SUN ES86PCIU+

使用手冊

/诲星斫电&科牧帝浪公司

第一章简介 1

[1.1实验仪介绍 1](#bookmark6)

1. [2实验仪配置方案 1](#bookmark8)

[1.3功能特点 2](#bookmark10)

[1.3. 1 软件 2](#bookmark12)

[1.3.2 硬件 3](#bookmark14)

第二章 硬件结构 5

1. 1 电路外观 5
2. 2 A1区：12864液晶显示模块电路 6
3. [3 A2 区：16X16 LED 实验电路 6](#bookmark24)
4. [4 A3区：CPU总线、片选区 7](#bookmark26)
5. 5 A4区：控制区 7
6. 6 B1区：语音模块ISD1420电路 7
7. 7 B2区：逻辑笔、单脉冲、频率发生器 8
8. [8 B3 区：8259 电路 9](#bookmark28)
9. 9 B4 区：简单 I/O、SRAM 电路 9
10. 10 B5、C5 区：扩展区 10
11. 11 C1区：电源区 10
12. [12 C2 区：ADC0809 模数转换 10](#bookmark30)
13. [13 C3 区：8251 11](#bookmark32)
14. [14 C4 区：8253 11](#bookmark34)
15. 15 D1区：步进电机 11

2. 16 D2 区：DAC0832 数模转换 12

2. 17 D3区：8255电路数码管驱动电路 12

2. 18 D4 区：8237 电路 13

[2. 19 D5 区：8250 电路 13](#bookmark36)

2. 20 D6 区：X5045 14

[2. 21 D7 区：RS232 14](#bookmark40)

1. [22 E1区：直流电机转速测量/控制 14](#bookmark42)
2. [23 E2区：光敏电阻、压力测量 15](#bookmark44)

[2. 23 E3区：继电器 15](#bookmark50)

[2. 24 E4 区：RS485 16](#bookmark52)

2. 25 F1区：红外通讯 16

2. 26 F2区：温度测量/控制 16

2. 27 F3区：PWM电压转换 16

1. 28 F4区：发光管、按键、开关 17

[2. 29 F5 区：键盘&LED 18](#bookmark54)

2. 30 F6区：0〜5V电压输出 18

2. 31 F7 区：138 译码器 18

1. 32 F8区：蜂鸣器 18

第三章星研集成环境软件 19

1. [1 软件安装 19](#bookmark56)
2. 1. 1安装星研集成环境软件 19
3. [1. 2 软件卸载 19](#bookmark58)
4. [1. 3 USB驱动程序 19](#bookmark60)
5. 1. 4软件启动 20
6. 1. 5编译器 21
7. [1. 6 README 文件 21](#bookmark66)
8. [2如何使用星研集成环境软件 22](#bookmark68)
9. [2. 1 数据传送程序（ASM） 22](#bookmark70)
10. 2. 2数据传送程序（C） 34
11. [3实验连线、演示实验、测试实验仪 41](#bookmark94)
12. [4 频率计（EMU86） 42](#bookmark98)
13. [5模拟波形发生器（EMU86） 43](#bookmark100)
14. [6 TDS2、TDS2A （EMU86）虚拟示波器 43](#bookmark102)

第四章软件实验 45

[实验一数据传送 45](#bookmark110)

[实验二 双字节BCD码（十进制数）加法 47](#bookmark122)

[实验三 双字节BCD码（十进制数）减法 50](#bookmark134)

[实验四四字节十六进制数转十进制数 53](#bookmark148)

[实验五散转 56](#bookmark162)

[实验六冒泡排序 58](#bookmark172)

[实验七二分查找法 60](#bookmark182)

第五章 基础硬件实验 62

[实验… 简单I/O （16位）实验 63](#bookmark194)

[实验二 静态存贮器（16位）读写实验 65](#bookmark204)

[实验三 8255控制交通灯实验 67](#bookmark214)

[实验四74HC138译码器实验 70](#bookmark228)

[实验五8253方波实验 72](#bookmark240)

[实验六8259A （1）中断控制器实验 74](#bookmark254)

[实验六 8259A （2）中断（优先级）实验 77](#bookmark260)

[实验七8251可编程通信实验（与微机） 81](#bookmark270)

[实验八8250可编程通信实验（与微机） 85](#bookmark276)

[实验九8255键盘显示实验 89](#bookmark290)

[实验十 并行DA实验 93](#bookmark306)

[实验十一 并行AD实验（数字电压表实验） 95](#bookmark314)

[实验十二红外通信实验 98](#bookmark324)

[实验十三图形点阵显示实验 102](#bookmark336)

[实验十四8237 DMA传输实验 111](#bookmark348)

[实验十五 RS485通信实验 115](#bookmark356)

[实验十六图形点阵显示实验(12864M) 119](#bookmark364)

[实验十七继电器控制实验 124](#bookmark374)

第六章综合实验 126

[实验一简易电子琴实验 126](#bookmark384)

[实验二 LED 16 \* 16点阵实验 136](#bookmark398)

[实验三数字式温度计实验(18B20) 141](#bookmark408)

[实验四步进电机实验 147](#bookmark412)

[实验五直流电机测速实验 154](#bookmark418)

[实验六旋转图形实验 160](#bookmark426)

[实验七ISD1420语音模块实验 164](#bookmark436)

实验九 X5045串行EEPROM读写实验 172

[实验十 电子钟(CLOCK) 179](#bookmark452)

[实验十一直流电机调速实验 185](#bookmark462)

[实验十二光敏电阻测量光照强度实验 188](#bookmark466)

附录A星研集成环境软件支持的软中断 190

iv

D简介

1.1实验仪介绍

SUN ES86PCIU+实验仪提供了全开放的80x86系统扩展总线，如：数据总线、地址总线、中 断请求信号、H0LD/HLDA,读写控制信号、字节使能信号等，全面支持开放式的微机原理、接口 技术各项实验。支持8/16/32位存贮器、I/O实验，可以满足各大专院校进行微机原理、微机 接口的开放式实验教学，它包含的丰富的功能模块、10多种的扩展模块也适合学生做课程设计、 毕业设计。

8086、80386EX模块与微机之间采用USB通信（真正的USB通信，而不是串口转USB）。 可选配“傻瓜式”**自检模块：**只需几根扁线，星研软件可对实验仪的所有模块自动检测， 自检结果…目了然，维护更方便。

星研软件提供一个库文件或提供对应的软中断，如果学生上机时间有限，只需编写最主要 的程序，其它调用库文件或软中断即可。提供详尽的C、汇编例子程序、使用说明。它布局合 理，清晰明了；模块化设计，可以无限升级，让您的选择永不落伍；兼容性强，可以轻松升级, 减少设备投资；使用方便，易于维护。

1.2实验仪配置方案

微机原理四种配置方案：

一、 实验仪主机、EMU86仿真模块，CPU： 8086, 16位数据总线。提供辅助功能：

1、 虚拟示波器：数字存贮虚拟示波器，画面与示波器类似，二个通道；触发极性分上升沿、 下降沿；触发通道分A、B二个通道；触发电平可选；触发方式：自动触发、单次触发、正常触 发；二个通道可随意选择垂直位置；10多种电压、时间分辨率选择；二个通道波形可加减；显 示每个通道的平均电压、峰峰值、频率。操作类似于泰克示波器。

2、 信号发生器：提供正弦波、方波、三角波。

3、 频率计：100MHz数字频率计功能

二、 实验仪主机、EMU386仿真模块，CPU： 80386EX,内部32位数据总线，外部16位数据 总线。

提供的辅助功能：双通道虚拟示波器；双通道信号发生器；数字频率计。

三、 上述任一种方案基础上，加配USB仿真功能：由微机CPU执行实验程序，涉及对实验仪 操作的指令，微机通过USB高速接口将指令发送给EMU86.EMU386产生时序，读写RAM、I/O接口， 将结果通过USB反馈给微机CPUo 32位数据总线。

因为实验程序由微机CPU运行，除了可能引起微机系统崩溃的指令外，实验程序几乎可由 任意指令组成，当然包括32位指令；支持软中断指令。

四、 实验仪主机、PCI板卡。

PCI板卡将PCI总线上的信号转换成局部总线信号，PCI板卡通常工作于ISA模式，PCI板 卡产生的的局部总线信号相当于一片8086 CPU产生的所有信号（它没有中断响应管脚INTA）o 支持8位、16位、32位数据总线。

允许学生在Windows XP/VISTA/Windows7等操作系统下，使用星研集成环境软件，编写、 编译、连接、错误定位、调试、观察、修改实验仪上提供的8位/16位/32位I/O、SRAM,而不 需要熟悉DOS环境，不需要使用早期的PWB、TD等DOS下的调试环境，也不用为无法查看硬件 内容而发愁。

支持二次开发：使用PCI模板，做PCI总线控制开发。例如：在PCI模块上加配数据采集 模块（8路12位AD、8路10位DA、8路光隔输入、8路光隔输出）；加配CAN模块，可做CAN 转 PCI。

采用了良好电路保护设计，星研产品从未损伤过客户微机。

方案三、四部分允许学生在Windows XP/Vista/Windows7等操作系统下直接操作硬件，因 为星研已替您解决了操作系统的限制，例如：INAL, DX； OUT DX, EAX。星研提供了多种软中 断功能，方便学生编写程序。

方案二、三、四可做保护模式下的全局描述符表实验、页目录表实验、页表实验、虚拟地 址转物理地址实验、物理地址转虚拟地址实验。方案二：三张表由您自行设计；方案三、四：三 张表由Windows XP/Vista/Windows7等操作系统为当前应用程序建立。通过对实验仪上16/32 位SRAM的读写，验证实验的正确性。

全面支持80x86实模式和保护模式微机原理及接口技术的实验教学

1.3功能特点

1.3.1软件

提供我公司自主版权的星研集成环境软件，2004年它已被认定为上海市高新技术成果转化 项目

O完全VC++风格。集编辑器、项目管理、启动编译、连接、错误定位、下载、调试于」 体，多种实验仪、仿真器、多类型CPU仿真全部集成在一个环境下，操作方法完全一样。不同 种类的CPU模块间采用软件切换。

◊在WindowsXP/Vista/Windows7的保护模式下，可使用单步、宏单步、全速断点（断点 数目没有限制）、全速运行等多种调试手段，调试所有实验程序，特别是32位汇编程序；任意 查看、修改实验仪上8/16/32位SRAM中数据、接口芯片内容。

◊支持ASM （汇编）、C、PLM语言，多种语言多模块混合调试：同时支持Ke订公司C51、 Franklin 公司 C51、IAR/Archimedes 公司的 C51、Intel C96> Tasking 的 C196、Borland 公司 的 TASM、 Turbo C。

O支持BIN、HEX、OMF、AUBROF等文件格式。可以直接转载ABS、0MF文件。

O支持所有数据类型观察和修改。自动收集变量于变量窗（自动、局部、模块、全局）。 O无须点击的感应式鼠标提示功能。

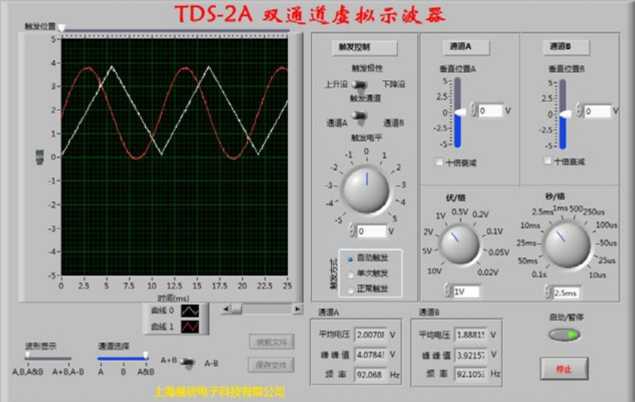
O功能强大的项目管理功能，含有调试该项目有关的仿真器、所有相关文件、编译软件、 编译连接控制项等所有的硬软件信息，下次打开该项目，无须设置，即可调试

◊支持USB、并口、串口通信。

O提供模拟调试功能：方便学生在实验前，编写程序；编译、连接，解决语法错误；初步 调试；在Windows XP/VISTA/Windows7环境中，提供软中断功能。

O软件提供一个窗口：显示当前实验程序对应的全局描述符表、页目录表、页表，方便教 学（全局描述符表、页目录表、页表均由操作系统自动产生）。支持实验仪的监控在线更新。

O符合编程语言语法的彩色文本显示，所有窗口的字体、大小、颜色可以随意设置。 O支持拖拉功能（源程序中快速移动、变量名拖至观察窗等）。调试极为快速方便。



1.3.2硬件

**仁主板组成**

|  |  |  |  |
| --- | --- | --- | --- |
| 显示部 分 | 液晶（四种选择）  1） 1602C字符型液晶  1） 12864图形点阵液晶（标配）；  2） 带汉字库的12864图形点阵液晶；  3） 带触摸屏的12864液晶 | 自定义8位、  16位总线 | 32K\*8 位\*2 片 SRAM |
| 简单 I/O （2 片 74HC273、2 片 74HC244） |
| 输入、输出 | 4\*4键盘 |
| 8个独立按键 |
| 8个拨动开关 |
| 16X16LED点阵显示模块 | 单脉冲 |
| 8位数码管 | 继电器控制 |
| 16个发光二极管（红、绿、黄色） | 可调电压 |
| 传统实 验 | 74HC138 译码 | 触摸屏（选配） |
| 8255、打印口 | AD、DA | 并行 AD ADC0809 |
| 8253/8254可编程定时器/计数器 | 并行 DA DAC0832 |
| 8251串行通信接口 | 传感器 | 压力传感器或光敏电阻电路 |
| 8250/16C550串行通信接口 | 光电开关电机测速 |
| 8259可编程中断控制器（可扩展级联） | 霍尔器件电机测速（选配） |
| 8237 DMA控制器 | 18B20测、控温（闭环） |
| 语音 | 录音、放音模块ISD1420 | 控制对象 | 直流电机控制（闭环） |
| 自带话筒、喇叭 | 步进电机控制（4相） |
| 蜂鸣器 | 控温模块 |
| 通信 | RS232和RS485接口电路 | 存贮器 | 串行EEPR0M及看门狗一 X5045 （SPI） |
| 红外通信实验 | 其它 | PWM脉宽调制输出接口 |
| 电源 | 5V 3A +12V 0. 4A -12V 0. 4A | 功率放大 |
| CPU | 8086： 128K仿真空间 | 逻辑笔 |
| 扩展区 | 三个扩展区；可根据客户需求，定制部件 | 74HC4040分频得到8种频率 |
| 辅助功 能 | 虚拟示波器、信号发生器、频率计 | | |

2、 扩展模块

实验主板右侧有三个万用扩展区（右上角是CPU扩展区），可根据客户需求，定制部件。

可扩展：GPS模块；GPRS模块；超声波模块；电流传感器模块；F/V、V/F模块；触模屏； USB1. 1模块；USB 2. 0模块；USB主控；CAN 2. 0模块；10M以太网模块；32位SRAM、32位10 模块；数据采集模块等。

3、 闭环控制

1） 旋转图形展现实验

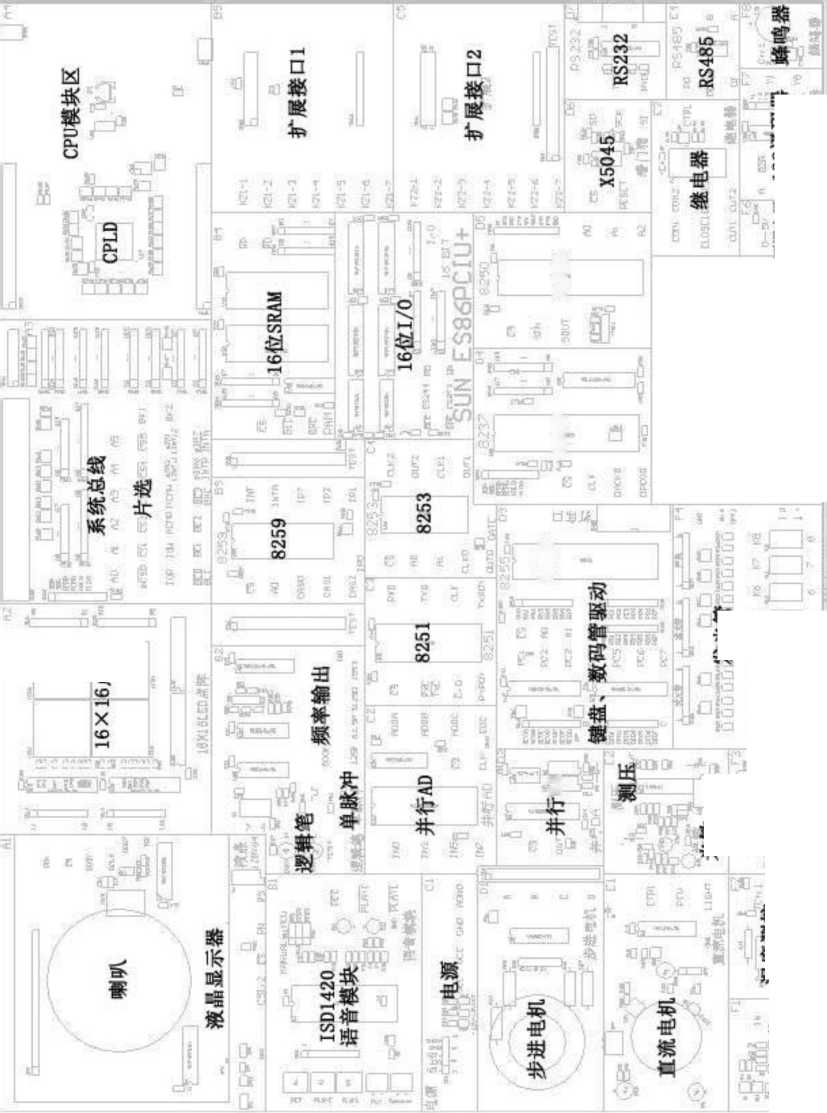
2） 直流电机转速控制，使用霍尔器件、光电开关精确控制电机转速

3） 数字式温度控制，通过该实验可较好认识控制在实际中的应用

4、 EDA——CPLD、FPGA可编程逻辑实验（选配）

可设计8255、8253、8237等实验；提供Verilog、VHDL语言编写的实验范例。

2硬件结构



电路外观

2. 1

0ZH8T

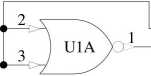
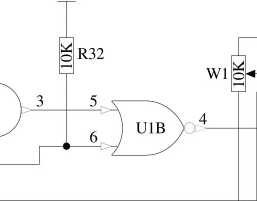
舉曲!fr8eT

|  |  |
| --- | --- |
|  |
|  |  |
|  |
|  |
|  |

|  | **8237 8250** |
| --- | --- |
|  |  |
| **to g z 00** |
|  |  |
|  |
|  |  |
|  |
| 苦 |
|  |  |

本章将逐一介绍实验仪的各个功能模块、相应的结构，读者在编写程序前，首先熟悉相应 的硬件电路。

**2. 2 A1**区:**12864**液晶显示模块电路



VCC

csO

RSO-

RWO—

JP2

VO 3

RS  
RW

c----

CS1/2O-

DO

PT

D2~  
pF

D4~

PF

D6 pF

CSf

\_7

「8

[9

10

Ti

T2

13

14

15

-

0

-

-

-

■｛二

--

CS2 16

17

18

19

VCC

20

VSS VDD VO

RS R/W

E

DB0

DB1

DB2

DB3

DB4

DB5

DB6

DB7

CS1

CS2

RST

VEE

LED1

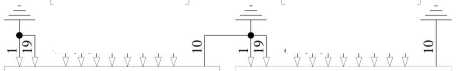
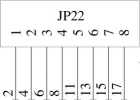
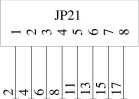
LED2

=12864J

gems

CS：片选信号，低电平有效。CS1/2：左右半屏使能选择，H：左半屏，L：右半屏。RS：选 择读写的是指令或数据，L：指令，H：为数据。RW：读写控制端，L：写操作，H：读操作。

1. 3 A2区：16X16 LED实验电路



SN74LS244N

iZE 寸《>9 卜 8  
AAAAAAAA

SN74LS244N  
iCME 寸 </">9卜00  
AXAAAAAA

——ZE 寸 S9f-8 Q S M VVVVVVVV z Qo U25 §

u

*>*

—<,Z —Zen 寸 S9L00 Q 枳田 <<<<<<<< g 60 U29 §

y

>

JP14

10

JP15

10

1

2

3

4

5

6

7

8

1

2

3

4

5

6

7

8

2  
f  
A

LL>

13  
E

17

|  |  |
| --- | --- |
| OEl  OE2 | VCC |
| Al | Y1 |
| A2 | Y2 |
| A3 | Y3 |
| A4 | Y4 |
| A5 | Y5 |
| A6 | Y6 |
| A7 | Y7 |
| A8 | Y8 |
| SN74LS244N | |
| GND |  |
| U20 | |
| OEl | VCC |
| OE2 |  |
| Al | Y1 |
| A2 | Y2 |
| A3 | Y3 |
| A4 | Y4 |
| A5 | Y5 |
| A6 | Y6 |
| A7 | Y7 |
| A8 | Y8 |
| SN74LS244N | |
| GND |  |

U19

VCC

20T

18

16

14

1

9\_

*匸*

5^  
I-

VCC

267

a

-——Q22

VCC

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 乂龙 | | £ | 6 r- | | m CO | |
| 9 | £ | 寸 | 6 |  | 0 | E |

OO

loTcnl

|  |  |
| --- | --- |
| R81 | 7 |
| k79 | 3 ' |
| R80 | 14 |
| R75 | *亠* |
| R77 |  |
| R78 | 11 |
| R76 | ■ 5 |

182 - 8

22

22

22

22

22

|  |  |
| --- | --- |
| R86 | 7 |
| R92 | 3 |
| R93 | ：14 |
| R91 | 2 |
| 專 | 如 |
| R85 | 11 |
| R84 | -5 |

L83

8

VCC

iZE 寸 V)9C~'8

R1 R2 R3 R4 R5 R6 R7 R8

LG7088BH

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 9 | ***9*** | 寸 | 6 | ID | O | E |

LED2\

R2

R3

R4

R5

R6

R7

R8

LG7088BH

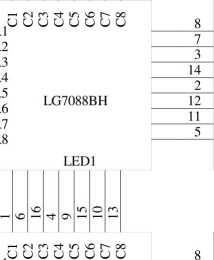
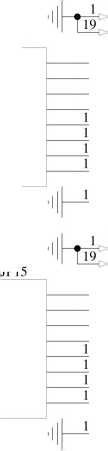
LED3

7  
亘  
14  
卫  
IT  
IT  
*~5*

iZE 寸 S9f-8 R]O u u u u u u u R2 R3 R4 R5 R6 R7 R8

LG7088BH

LED4



JP14、JP15组成16根行扫描线；JP21、JP22组成16根列扫描线。

**2. 4 A3**区：系统总线、片选区

JP28：地址线 AO. . A7；

JP32：地址线 A8.. A15；

JP29：地址线 AO. . A9；

根据数据总线宽度，选择合适的地址线

JP33：地址线 A8.. A17；

JP41、JP42：数据总线 DO. . D7

JP39、JP40：数据总线 D8. .D15

JP47、JP48：数据总线 D16. .D23

JP45、JP46：数据总线 D24. .D31

控制线：IOR、IOW, MEMR、MEMW, HOLD、HLDA、BLE、BHE、INTR、INTA

BK1、BK2：备用

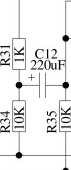
片选区

|  |  |  |
| --- | --- | --- |
| 片选 | 地址范围 | 说明 |
| mCSO | 80000H〜BFFFFH | 存贮器芯片的片选，16位数据总线 |
| CS1 | 0270H〜027FH | I/O芯片的片选，8位数据总线 |
| CS2 | 0260H〜026FH |
| CS3 | 0250H〜025FH |
| CS4 | 0240H〜024FH |
| CS5 | 0230H〜023FH | I/O芯片的片选，16位数据总线 |

2. 5 A4区：控制区

主控部分。

**2. 6 B1**区：语音模块**ISD1420**电路



VCC

Orec

DS5

R9

U2

Mic

Speaker

510

Cll

XCLK o NC 1F NC

NC

NC

VCCD

VCCA

28

16

VCC

27

REC v呀 PLAYE 丐寸 PLAYL U -

16R27R28

JP11 3

JP10 3

JP12 3

-o o

REC

「PLAYE

-o o >

PLAYE

J PLAYL

0.1 uF

4.7uF

[70K^

C7 I I 0.1 uF

fi

RECLED

AGC

ANA IN

ANA OUT

-o o

PLAYL

14

15

12

MIC REF

MIC

SP+

SP-

VSSD

VSSA

01234567

AAAAAAAA7

-1

-2

-3

-4

-5

-6

-7

-8

JP5

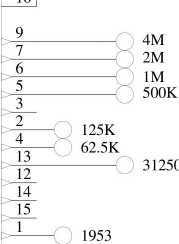
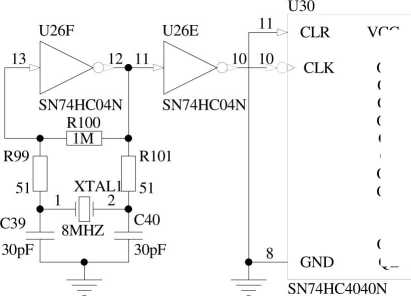
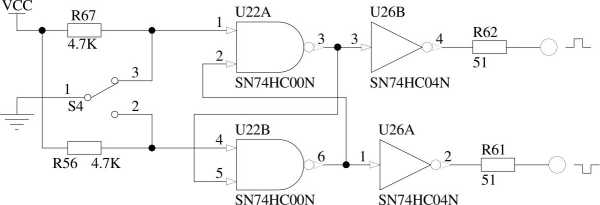
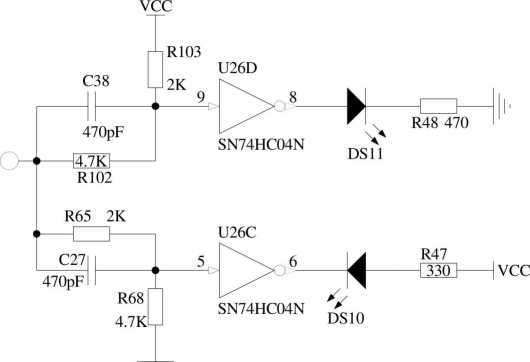
— ISD1420

JP10、JP11、JP12：设置操作模式，MCU： CPU 控制方式；MANUAL：手动（REC、PLAYL、PLAYE） 控制方式。

REC： 录音按键，低电平有效；

PLAYE：电平放音按键，低电平有效，直到放音内容结束停止放音

PLAYL：边沿放音按键，下降沿有效，并在下一个上升沿停止放音



JP5： 录、放音的起始地址

**2. 7 B2**区：逻辑笔、单脉冲、频率发生器

逻辑笔： 测试接口，输入 测量信号

绿灯(DS10):高电平点亮

红灯(DS11)：低电平点亮 两灯同时亮：频率信号

逻辑笔电路原理图

单脉冲电路原理图

S4： 脉冲发生开关

正脉冲：上凸符号端口输出正脉冲  
负脉冲：下凹符号端口输出负脉冲

XQAQBgQD雀QGQHOOQK如

频率发生器电路原理图

4M： 输出4MHZ频率信号

其他端口输出的信号频率与端口下标识的数值一致

2. 8 B3 区：8259 电路

csC

CASO O

CAS1 (J

|  |  |
| --- | --- |
| WR | 2 |
| RD | 3 |
| D7 | 4 |
| D6 | 5 |
| D5 | 6 |
| D4 | 7 |
| D3 | 8 |
| D2 | 9 |
| DI | 10 |
| DO | 11 |
| 12 | |
| 13 | |

14

U33

1

VCC

AO

INTA

IR7

CASO

CAS1

GND

6 5 4 3 2 10 IRIRIRIRIRIRIR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 24 |  | | |  | |
| 23 |
| 22 亠 | 1 | 1' | |
| 21 | 1 |
| 20 |  | |  |
| 19 | |  |  |  | |
| 18 . | |  |  |  | |
| 17 |  |  |  |  | |
| 16 |  |  |  |  | |
| 15 |  |  |  |  |  |
| i | e | cn | 寸 |  | |R109 |
|  |  |  |  |  |  |

INT

SP^N

CAS2

VCC

28 T

27

26

25

厂）IR2

IR1

-\_ IRO

■O5fr\_ <)SP/EN

-CAS2

AO

INTA

IR7

cs：片选信号，低电平有效;

A0： 地址信号

INRO.. INR7：中断输入

INTA： 中断响应

**2. 9 B4**区：简单**I/O**、**SRAM**电路

CS244 O 4

vccl

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| BLE | 9、 *、* | | | | BHE | 12. |
|  | **10** | j U37C) | 亠 | U40 | VCC |  |

U37B

U37D

SN74HC244N

BLE\_IO BHE\_IO

R142

RD C

R117 nos} nos}

R118 \_

riDKI ~~KS273~~

R141

1

2

3

4

5

6

7

8

|  |  |
| --- | --- |
| INI | 17 |
| IN2 | 4 |
| IN3 | 15 |
| IN4 | 6 |
| IN5 | 13 |
| IN6 | 8 |
| IN7 | 11 |

|  |  |
| --- | --- |
| 0E1  0E2 | VCC |
| A1 | Y1 |
| A8 | Y8 |
| A2 | Y2 |
| A7 | Y7 |
| A3 | Y3 |
| A6 | Y6 |
| A4 | Y4 |
| A5 | Y5 |
| GND |  |

|  |  |
| --- | --- |
| 3 | DI |
| 16 | D2 |
| 5 | D3 |
| 14 | D4 |
| 7 | D5 |
| 12 | D6 |
| 9 | D7 |

|  |  |
| --- | --- |
| IN2 | 17 |
| IN3 | 4' |
| IN4 | 15 - |
| IN5 | 6 |
| IN6 | 13 |
| IN7 | 8' |
| INO | 11' |

U45

OE1

OE2

VCC

VCC

20T

nRST

Al

A8

A2

A7

A3

A6

A4

A5

Y1

Y8

Y2

Y7

Y3

Y6

Y4

Y5

|  |  |
| --- | --- |
| 3 | D9 |
| 16 | DIO |
| 5 | Dll |
| 14 | D12 |
| 7 | D13 |
| 12 | D14 |
| 9 | D15 |

18 D8

GND

SN74HC244N

U46

|  |  |
| --- | --- |
| VCC | RST CLK |
| Q0 | DO |
| Q7 | D7 |
| Q1 | D1 |
| Q6 | D6 |
| Q2 | D2 |
| Q5 | D5 |
| Q3 | D3 |
| Q4 | D4 |
|  | GND |
| SN74HC273N | |

U41

1

IT

VCC

RST - ——

CLK —~

可做8位、16位简单I/O实验

CS244： 74HC244片选信号，低电平有效；

CS273： 74HC273片选信号，低电平有效；

RD： 读信号，低电平有效，读74HC244

1. DO

昌 18 DI

1. D2

'17 D3

,7 D4

L14 D5~

J 8 D6

\_\_DI

2

VCC I~^―

JP55

QO

Q7

Qi

Q6

Q2

Q5

Q3

Q4

DO

D7  
DI

D6

D2

D5

D3

D4

GND

SN74HC273N

D8

D9

DIO

Dll

D12

D13

D14

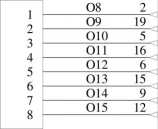
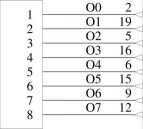
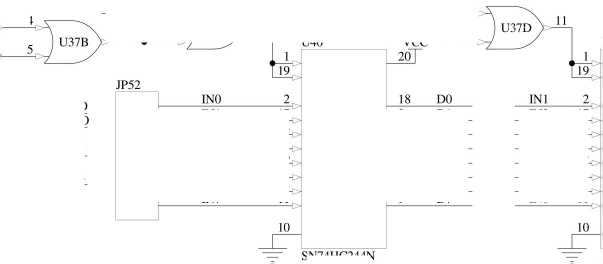
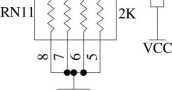
D15

10

BLE：低8位有效，允许访问U40、U41

BHE：高8位有效，允许访问U45、U46

WR：写信号，低电平有效，写74HC273



U38A

JP38

csC>

bleQ

AO

14

JP49

22

27

11 DO

U43

A15

RD

BHE O

5

JP57

12

13

RD WR

22

27

祚)

c

VC

1

2

3

4

5

6

7

8

A8 25

|  |  |
| --- | --- |
| Al |  |
| A2 | 8 |
| A3 |  |
| A4 | 6 |
| A5 | 5 |
| A6 | 4 |
| A7 | \_\_3： |

10,

A9 24

A10 21~

All 2于

A12 ，

Al 3 2&

A14 1

0 12 3 4 012345678911111 AAA.AAAAAAAAAAAA

-E 12345678 c QQQQQQQQ DDDDDDDD

9SZOZ9

11

|  |  |
| --- | --- |
| 12- | DI |
| 13~ | D2 |
| 15- | -D3 |
| 16- | -D4 |
| 17- | *D5* |
| 18- | D6 |
| 19- | D7 |

<E

<C

<K

<E

<E

<X

<X

——|vcc

|  |  |
| --- | --- |
| Al |  |
| A2 | 8 |
| A3 | 7 |
| A4 | 6, |
| A5 | 5, |
| A6 | 4 |
| A7 | 3 |
| A8 | 25, |
| A9 | 24 |
| A10 | 21, |
| All | 23 |
| A12 | 2, |
| A13 | 26' |
| A14 1： | |

A0

U44

AO

Al

A2

A3

A4

A5

A6

A7

A8

A9

A10

All

A12

A13

A14

CE

20

DQ1

DQ2

DQ3 gDQ4 RDQ5 ^DQ6 9DQ7

DQ8

VCC

GND

2

4

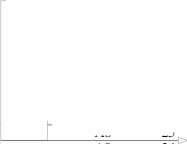
6

7

~~亦一~~|VCC "'6

|  |  |
| --- | --- |
| 12~ | D9 ] |
| 13~ | ~DW |
| 15~ | Dll [ |
| 16~ | ~D12 |
| 1厂 | -D13 |
| 18- | D14 |
| *19~* | ~D15 |

J\_\_D8



CS：片选信号，低电平有效;

RD：读信号，低电平有效

BLE：低8位有效，允许访问U43

BHE：高8位有效，允许访问U44

WR：写信号，低电平有效

**2. 10 B5**、**05**区：扩展区

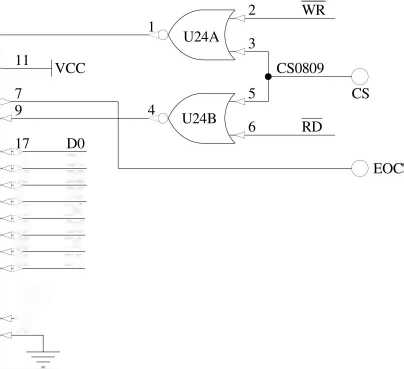
实验仪提供了二个扩展区，用来扩展USB1. 1、USB2.0、以太网、CAN总线、非接触式IC卡、 GPS、GPRS、双通道虚拟示波器、CPLD、FPGA等扩展模块，其它模块正在陆续推出中。

如果扩展模块较大，可以同时使用二个扩展区。

**2. 11 01**区：电源区

C1区为用户提供了+5V(2A)、+12V(400mA), -12V(400mA)等几种电源接口。DS& +5v指示 灯；DS7： +12v指示灯;DS6： T2v指示灯。

1. 12 C2区：ADC0809模数转换



ui6

clkQ

10

*~6*

CLK

START

IN0( ■

IN3C

IN7 C.:

ADDA

ADDB ]

ADDC =

ADDA ADDB ADDC ALE

25

24

23

1NTNININTNTNINTN

01234567

NOO6080OCIV

VCC

EOC

OE

DO

DI

D2

D3

D4

D5

D6

D7

VREF+

VREF-

GND

1. DI
2. D2

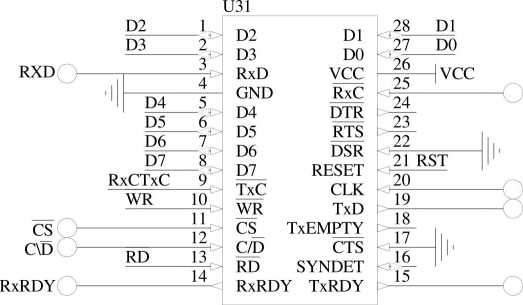
8 D3

1. D4
2. D5
3. D6
4. D7

VCC iir 16

|  |  |
| --- | --- |
| CS： | 片选，低电平有效； |
| CLK： | 输入时钟(10k- 1280kHZ)； |
| ADDA, ADDB, ADDC： | 通道地址输入口； |
| EOC： | 转换结束标志，高电平有效。 |
| INO、 IN3、 IN5、 IN7： | 模拟量输入 |

1. **13 C3** 区：**8251** 电路



8251

RxC TxC

TxRDY

CLK TXD

| CS：片选信号，低电平有效; | |
| --- | --- |
| RxC、 TxC： | 收发时钟； |
| C/D： | 命令/数据； |
| RXD、 TXD： | 串行收发。 |
| CLK： | 时钟。 |

1. 14 C4 区：8253 电路

| D0 | OUTO |
| --- | --- |
| D1 | GATEO |
| D2 | CLKO ] |
| D3 |  |
| D4 |  |
| D5 |  |
| D6 | OUT1 |
| D7 | GATE1 |
|  | CLK1 < |
| CS |  |
| RD |  |
| WR |  |
|  | OUT2 |
| AO | GATE2 |
| Al | CLK2 |

U34

8253

WR 23

DO 8

DI 7

D2 6

D3 *5*

D4 4

D5 3

D6 2

D7 f

| 10 | |
| --- | --- |
| H Rb | )8ji nv i |
| '9勺 |  |
| 13 |  |
| 15 |  |
|  |  |
| 16 |  |

O0UT1

OCLK1

OUT2

18

-OUTO

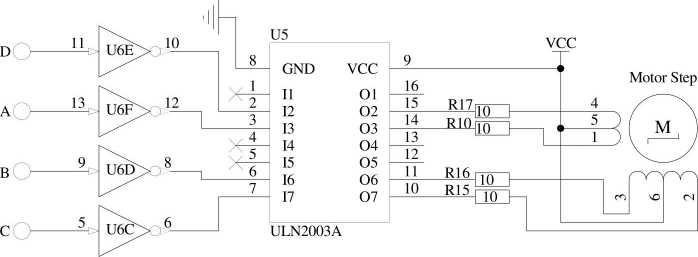
- GATE -Qclko

CS:片选信号,低电平有效;

AO、Al:地址信号；

CLK2

**2. 15 D1**区：步进电机



**2. 16 D2**区：**DAC0832**数模转换

U17

CS

WR

vcc}

DO

dT

D2

D3

D4

D5

D6

D7

|  |  |
| --- | --- |
| CS  WR1  WR2  ILE XFER | VDD |
| DIO | VREF |
| DI1 |  |
| DI2 | RFB |
| DI3 |  |
| DI4 | IOUT1 |
| DI5 | IOUT2 |
| DI6 |  |
| DI7 |  |
| AGND  DGND |  |

11

*~V2*

!

I

VCC

W5

9

2聲

|  |  |
| --- | --- |
|  | < |
|  |  |

+12V —

：C30

0.1 uF

H-12V

LM741CN

-12V

— DAC0832LCN

CS：片选，低电平有效；OUT：转换电压输出；电位器W5：调整基准电压。

**2. 17 D3**区：**8255**电路、数码管驱动电路

|  |  |
| --- | --- |
| D1 | 33 |
| D2 | 32 |
| D3 | 31 |
| D4 | 30 |
| D5 | 29 |
| D6 | 28 |
| D7 | 27 |
| RD | 5r |
| WR | 36 |
| 9； | |
|  | |
| RST | 35 |

D0 34

csO

|  |  |
| --- | --- |
| DO | PAO |
| DI | PAI |
| D2 | PA2 |
| D3 | PA3 |
| D4 | PA4 |
| D5 | PA5 |
| D6 | PA6 |
| D7 | PA7 |
| RD | PBO |
| WR | PB1 |
| AO | PB2 |
| Al | PB3 |
| RESET | PB4 |
| CS | PB5  PB6  PB7 |

}4 PAO

JP23

1

2

3

4

5

6

7

8

U27

OE1  
OE2

VCC

8255

VCC

01234567 pcpcpcpcpcpcpcpc

1

2

3

4

5

6

7

8

|  |  |
| --- | --- |
| PA1 | 17, |
| PA2 | 4. |
| PA3 | 15^ |
| PA4 | 6 |
| PA5 | 13 |
| PA6 | 8 |
| PA7 |  |

10

PAO

2,

Al

A8

A2

A7

A3

A6

A4

A5

Y1

Y8

Y2

Y7

Y3

Y6

Y4

Y5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **z** |  | 寸 | - | **z** | co | 寸 |  |
| 18 |  |  |  |  |  |  |  | SEGA |
| 3 |  |  |  |  |  |  |  | SEGB |
| 16 | | |  |  |  |  |  | SEGC |
| 5 | | |  |  |  |  |  | SEGD |
| 14 | | | | |  |  |  | SEGE |
| 7 | | | | | |  |  | SEGF |
| 12 | | | | | |  |  | SEGG |
| -9 Lh | | | | | | | | SEGDP |

VCC牛

JP16

1

2

3

1. B

5

6

7

8

GND

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | PCI | | | | | | |  |
| 6 | PC2 | | | | | |  |  |
| 17 | PC3 | | | |  |  |  |  |
| 3 | PC4 - | | | |  |  |  |  |
| 2 | ?C5 | | |  |  |  |  |  |
| 11 pc | | 6 |  |  |  |  |  |  |
| 1— |  | 7 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  | | | | |  | |
|  |  |  | | | | | | |
|  |  |  | | D | | | | |

J14 PCO

01234567

OOOOQHd

—SN74LS240N

U28

|  |  |
| --- | --- |
| PB1 | 17： |
| PB2 | *4.* |
| PB3 | 15 |
| PB4 | 6 |
| PB5 | 13 |
| PB6 | 8 |
| PB7 | 11 |

PBO

2

10

0E1

0E2

VCC

Y1

Y2

Y3

Y4

Y5

Y6

Y7

Y8

|  |  |
| --- | --- |
| 18 | DIGO |
| 16 | DIG1 |
| 14 | DIG2 |
| 12 | DIG3 |
| 9 | DIG4 |
| 7 | DIG5 |
| 5 | DIG6 |
| 3 | DIG7 |

JP17

1

2

3

1. C

5

6

7

S

GND

—SN74LS244N

|  |  |  |  |
| --- | --- | --- | --- |
| CS： | 片选信号，低电平有效； | AO、 A1： | 地址信号。 |
| JP24： | PC 口（键盘行）； | JP20： | PB 口（键盘列） |
| JP23： | PA 口； | JP16: | 数码管段码 |
| JP17: | 数码管段选 |  |  |

JP30

PB0

PB2

PB4

PB6

PC7

1 2

3 4

5 6

7 8

9 10

11 12

13 14

PB1

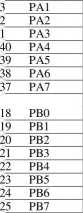
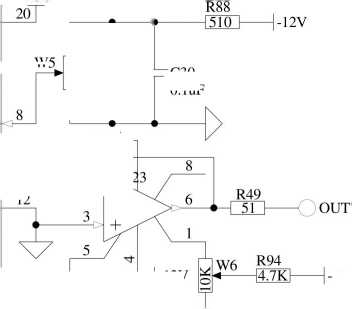
PB3

PB5

PB7

PCO

打印口



2. 18 D4 区：8237 电路

DREQO

csO

clkO

DACKOQ

DREQO

DREQ1

DREQ2

DREQ3

HLDA

READY

CS

> CLK RESET

DACKO

DACK1

DACK2

DACK3

HRQ EOP

GND

82C37A

IOR iow MEMR MEMW

NC

vcc

U397 i

2

RN17

10K

U35

17

16

HLDA8237 7

~~6

11

12

RST B

25

24

Jj

IT

HOLD8237 10 vcc I—rwi\_—

R122 20

01234567 01234567

AAAAAAAA DDDDDDDD

|  |  |
| --- | --- |
| 33 | Al |
| 34 | A2 |
| 35 | A3 |
| 37 | A4 |
| 38 | A5 |
| 39 | A6 |
| 40 | A7 |

32 AO

倾 IOW MEMR MEMW HOLD HLDA

CS： 片选信号，低电平有效;

DREQO： DMA请求信号；

HOLD： 总线请求信号，高电平有效；

I0R： I/O设备读信号，低电平有效;

MEMR： 存贮器读信号，低电平有效；

2. 19 D5 区：8250 电路

VCC

2

3

4

5

6

7

8

JP34

1

2

3

4

5

6

7

8

|  |  |  |
| --- | --- | --- |
| 29 | DI | 31 |
| 28 | D2 | 4 |
| 27 | D3 |  |
| 26 | D4 | 6 |
| 23 | D5 | *l'* |
| 22 | D6 |  |
| 21 | D7 |  |

30 DO

|  |  |
| --- | --- |
| 1 !or | |
| 2 | IOW |
| 3 | MEMR |
| 4 | MEMW |
| 5 |  |
|  |  |

1

2

3

4

5

6

7

8

9

RI

DCD DSR CTS

SIN SOUT DTR RTS

JP58

|  |  |
| --- | --- |
| DI | 2； |
| D2 |  |
| D3 | 4C |
| D4 | 5' |
| D5 | 6' |
| D6 | 7 |
| D7 | 8' |
|  | 9\_ |
| T | 2 |
| L\_\_13： | |

D0

^=C61

0.1 uF

1

csC>

SIN；

SOU1Q-

XTAL2

3.072MHZ

IM

R123

16

17

C56 50pF

30pF

—

R125 —

WR 18

20

宁 8250

|  |  |
| --- | --- |
| l§ | 严C  O |
| DI | Ql |
| D2 | Q2 |
| D3 | Q3 |
| D4 | Q4 |
| D5 | Q5 |
| D6 | Q6 |
| D7 | Q7 |
| D8 | Q8 |

—SN74HC573N

VCC

20T

|  |  |
| --- | --- |
|  | A9 |
| 17 | A10 |
| ~16~ | All |
| 15 | A12 |
| 14 |  |
|  | A14 |
| 12 | A15 |

JP50

19 A8 | .

CLK： 时钟宿号。控制8237内部定时和DMA

传输时的数据传送速率

DACKO： DMA响应信号；

HLDA: 总线响应信号，高电平有效；

I0W： I/O设备写信号，低电平有效；

MEMW： 存贮器写信号，低电平有效；

|  |  |
| --- | --- |
| DO | VCC |
| DI |  |
| D2 | RI |
| D3 | DCD |
| D4 | DSR |
| D5 | CTS |
| D6 | DTR |
| D7 | RTS |
| RCLK | MR |
| BAUDOUT |  |
|  | OUT1 |
| CSO | OUT2 |
| CS1 | INTR |
| CS2 | NC |
| SIN | AO |
|  | Al |
| SOUT | A2 |
|  | ADS |
|  | CSOUT |
| • XIN | DDIS |
| XOUT | RD |
| WR |  |
| WR | RD |
| GND |  |

10

11

U42

VCC

40T

39 RI

38 DCD

37 DSR

36 CTS

33 DTR 「32 RTS

^35 RST

29

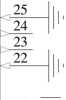
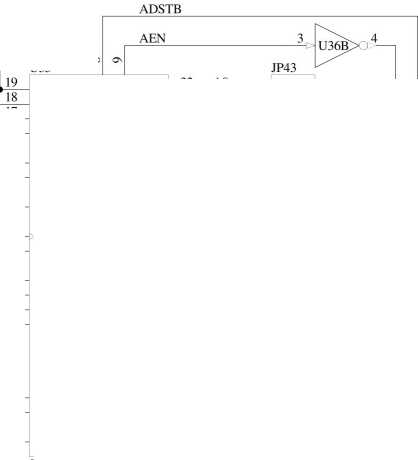
GA?

2] RD

CS：片选信号，低电平有效;

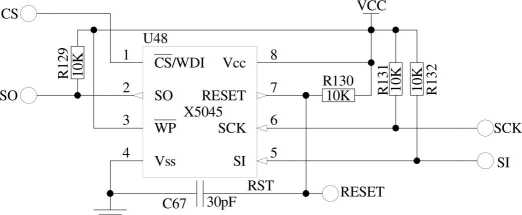
A0、Al、A2：地址信号； SIN : 串行输入

SOUT： 串行输出

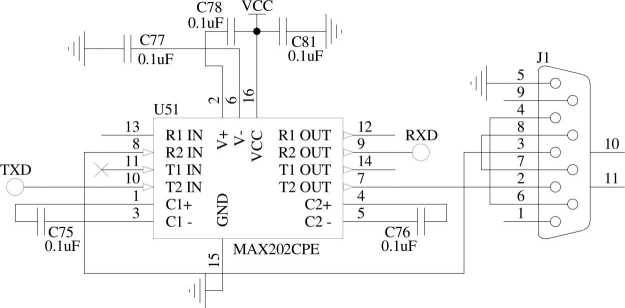


1. **20 D6** 区：**X5045**

|  |  |
| --- | --- |
| CS： | 片选，低有效 |
| SCK： | 时钟 |
| SI： | 数据输入 |
| S0： | 数据输出 |
| RESET： | 复位信号输出端，高电平有效 |



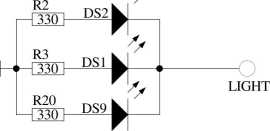
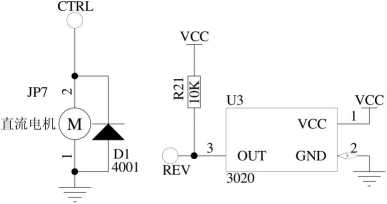
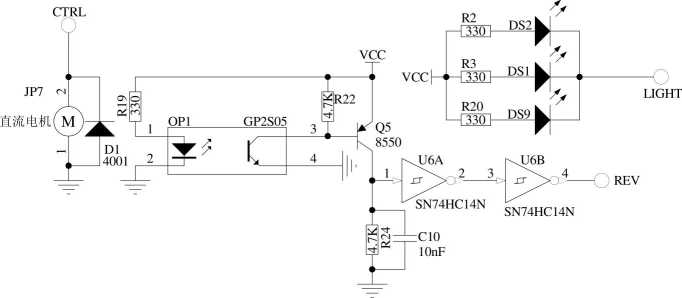
1. 21 D7 区：RS232



**2. 22 E1**区：直流电机转速测量/控制

使用光电开关测速

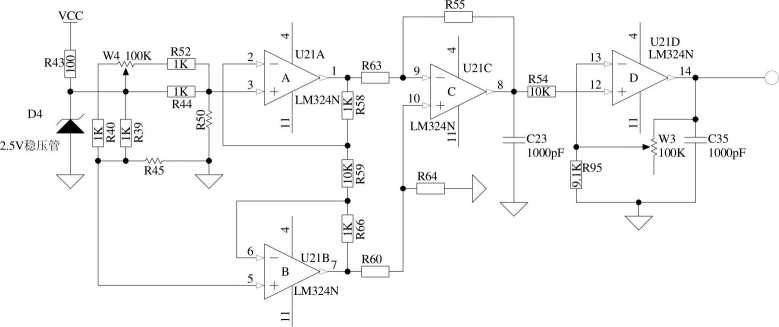
使用霍尔器件测速



CTRL：控制电压（DAC0832经功放电路提供）输入；REV：光电开关或霍尔器件脉冲输出（用 于转速测量）；LIGHT：低电平点亮发光管。

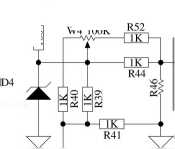
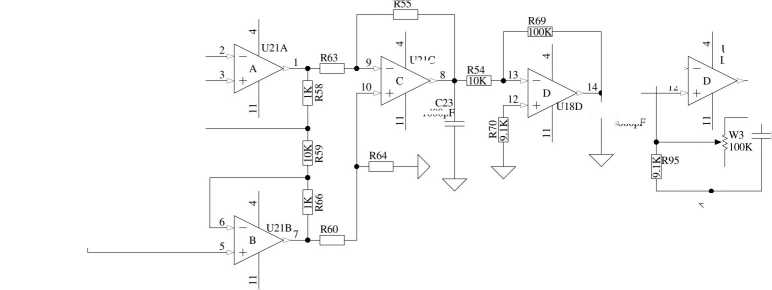
**2. 23 E2**区：光敏电阻、压力测量

光敏电路



R45、R50是光敏电阻；OUT：模拟电压信号输出端。

测压电路



2.5V稳压管

2 A

U21C

12

lOOOpF

R89

4TOK-

C35 lOOOpF

U21D

LM324N

14-

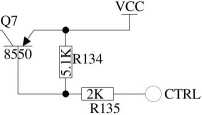
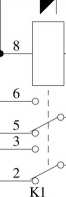
压力测量： R46：电阻应变片，阻值1K

W4 100K

lOOOpF

土C3]

■QOUT



CUT1O

CLOSE 1 -

CUT2O

CL0SE2O

D7

-:COM2

R133

nn——

OC0M1

OUT:压力模拟电压信号输出端;

2. 24 E3区：继电器

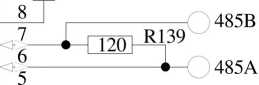
CTRL： 继电器开闭控制端

COM1、COM2：公共端 1、2

CLOSER CLOSE2：常闭端 1、2

CUT1、2： 常开端 1、2

1. 25 E4 区：RS485



U52

RO

RE/DE

DI

MAX485CPA

VCC

R0： 串行数据输出

DI： 串行数据输入

RE/DE：低电平，允许接收数据

:CB A Q INI V G s寸 xvw O-E E I R-R D D

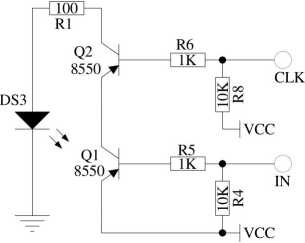
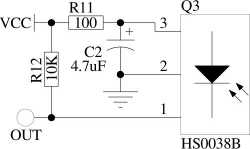
高电平，允许发送数据

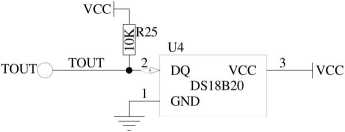
A、B： RS485 总线

**2. 26 F1**区：红外通讯

OUT: 串行数据输出

IN: 串行数据输入





CLK: 载波输入，可接31250 （B2区）频率输出

**2. 27 F2**区：温度测量/控制

TOUT： 数据线

TCtrl：温度控制端，向发热电阻RT1供电

**2. 28 F3**区：**PWM**电压转换、功率放大电路

**PWM电压转换电路**

IN： 信号输入

OUT： PWM转换电压输出

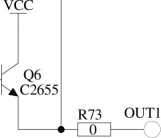
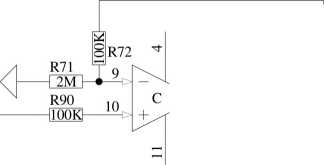
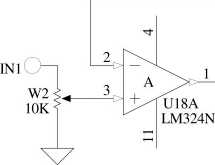
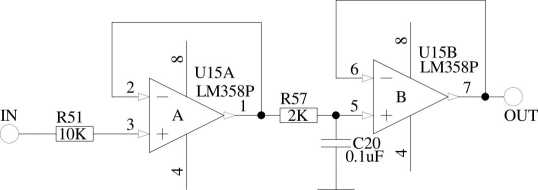
U18C

LM324N

**功率放大电路**

IN1：信号输入

0UT1：信号输出



**2. 29 F4**区：发光管、开关

18

JP1

JP19

iZE 寸 S9f-0O

OZSG

6ISG

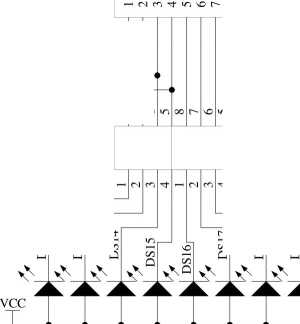
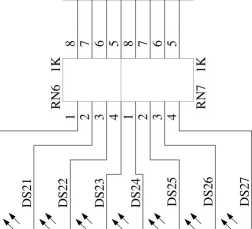
8

寸NH

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |
|  |  |  |
|  |  |  |
| OO |  | **9** |

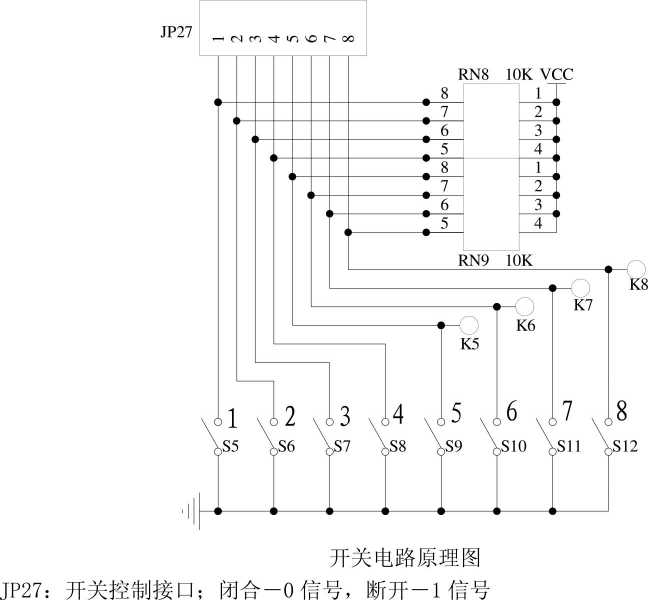
LEDLEDLEDLED4

ZISG

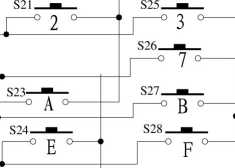
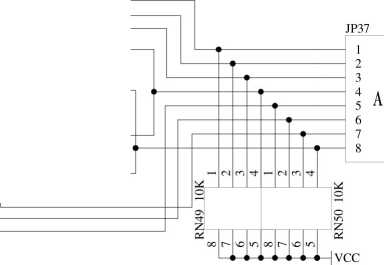
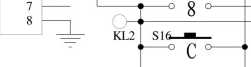
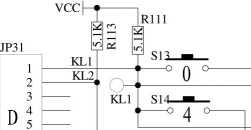
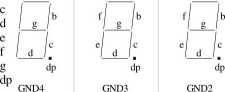
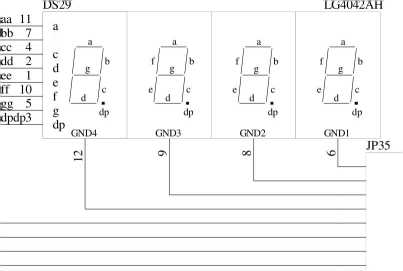
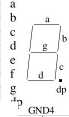
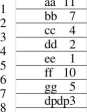


发光管电路原理图

JP18、JP19：发光管控制接口，0—灯亮，1—灯灭



1. 30 F5 区：键盘&LED



JP36

B

DS28

dp

**z**

| a | *b* | *力* |
| --- | --- | --- |
|  | c | *4丄1：* |
| GN,D3 | | GN,D2 |

LG4042AH

dp

GND I

1

2

3

1. C

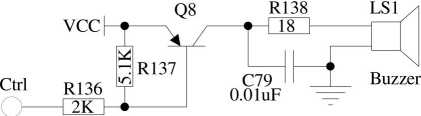
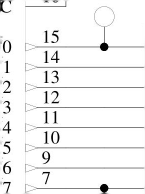
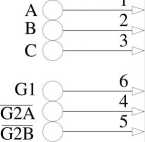
6

7

b

6

*8*



U49

A

B

C

OE1

OE2A

OE2B

8

GND

SN74HC138N

VCC

1

2

3

4

5

6

7

8

c 一 YYYYY一 YYY

***2. 33* F8**区：蜂鸣器

Ctrl：控制接口，0—蜂鸣

|  |  |  |  |
| --- | --- | --- | --- |
| A： | 按键的列线 | B： | 数码管段码 |
| C： | 数码管选择脚 | D： | 按键的行线 |

**2. 31 F6**区：**0**〜**5V**电压输出

v ' 0-5V：电压输出端

vcc| II'

W7

**2・32 F7**区：**138**译码器

JP63

A、B、C： 3位数据输入口

Y0〜Y7： 8位译码数据输出口

Gl、G2A > G2B：译码控制口

E5星研集成环境软件

USB接口的仿真器、实验仪客户：USB设备是即插即用的设备，在第一次安装时，Windows 将调用“添加新设备向导”扫描所有可用的INF文件，试图找到合适的驱动程序。为了避免USB 设备安装可能造成的麻烦，我们强烈的建议您先安装星研集成环境软件，安装程序将自动处理 USB设备安装所需的INF文件和驱动程序。

3. 1软件安装

**3. 1. 1**安装星研集成环境软件

1. 新用户安装步骤

使用光盘安装：

1. 将仿真器、实验仪所配CD插入CD-ROM驱动器。
2. 在“我的电脑”或“资源管理器”中选择CD-ROM驱动器\星研\ WIN32\星研，然后运行SETUP. EXE 文件即可进入安装界面。
3. 中文界面，用户只需按程序提示一步一步进行安装即可。

使用Internet下载文件的用户

1. 运行下载文件（XingYan. exe）,软件自动执行安装程序。
2. 安装程序为中文显示，用户只需按程序提示一步一步进行安装即可。
3. 已安装过低版本星研集成环境软件的用户的安装步骤：
4. 首先将原来的低版本软件进行卸载，具体步骤请参考“软件卸载”部分的内容。
5. 以后按新用户的安装步骤进行安装。

在安装过程中，如果用户没有指定安装目录，安装完成后会在C:盘建立一个C:\XINGYAN 目录（文件夹），结构如下：

XingYan 可执行文件、DLL文件、寄存器文件

EXAMPLES 例子程序

3. 1. 2软件卸载

1. 进入控制面板，运行“添加/删除程序”。
2. 进入“添加/删除程序”窗口，在“安装/卸载”页面上的列表中选择“星研集成环境软件”, 按“删除”按钮，之后按自动卸载程序的说明一步一步地操作即可。
3. 1. 3 USB驱动程序

1、USB驱动程序的安装

通过USB （通用串口总线）接口将微机与仿真器、实验仪相连，打开仿真器、实验仪电源。 仿真器、实验仪与微机的第…次连接引起驱动程序的安装会变得很简单，您只需等待安装过程的结束或按驱动程序的安装向导执行完即可。驱动程序的安装会出现如下界面:





实际的界面可能有些差别，请等待该过程的结束。驱动程序的安装过程中，请勿执行其它 应用程序。

2、如何解决连接不上的情况

如果仿真器、实验仪与微机连接不上是由于未按正确步骤造成的，可根据以下步骤解决：

Window98/Window Me：重新安装星研集成环境软件，关闭仿真器电源，稍等几秒钟，再打 开电源，等待操作系统安装新的驱动程序结束后，运行星研软件即可。

Windows2000/WinXP/WIN7：在仿真器或实验仪电源打开的情况下，使用控制面板中的'‘设 备管理器”，可以看到一个未安装好的USB设备：



上图中的“通用串行总线控制器”下有一个打问号的USB设备，选中后按鼠标右键，选择 菜单中的“卸载”项。重新安装星研集成环境软件，关闭仿真器、实验仪电源，稍等几秒钟， 再打开电源，等待操作系统安装新的驱动程序结束后，运行星研软件即可。

**注意：必须先安装星研集成环境软件；在WinXP/VISTA/WIN7中，驱动程序的安装会有选项， 按缺省的值选择即可。**

**3. 1. 4**软件启动

运彳亍Windows,进入桌面窗口。

鼠标单击“开始”按钮，在“程序”栏中打开“星研集成环境软件”菜单栏，在其中选择 “星研（SUPER、STAR系列仿真器）”，开始启动星研集成环境软件。

注意：当您使用低配置机器时，从星研集成环境软件退出后必须等待足够的时间，让系统完全 退出（硬盘停止工作）后，方可再次启动星研集成环境软件。

**3. 1. 5**编译器

**星研集成环境软件支持的编译器**

|  |  |  |
| --- | --- | --- |
| MCS51 | MCS96、 MCS196 | 80X86 |
| Keil A51、 C51  Franklin A51> C51  Intel ASM51、 PL/M51  Archimedes A8051> C-51 | Intel ASM96、 PLM96、 C96  Tasking ASM196、 C196 | TC、 TASM |

编译器请用户自备。

**设置工作环境**

您的编译器正确安装后，请设置星研集成环境软件的编译器工作环境。 打开［主菜单》项目》设置工作环境］:



例如：您使用的编译器是TASM、TC,安装在C:\xingyan\TASM, C:\xingyan\TC,

TASM 宏汇编路径： C:\xingyan\TASM；

Turbo C 路径： C:\xingyan\TC；

1. 1. 6 README 文件

使用通用的文本编辑器，打开星研集成环境软件安装目录下的README. DOC文件，可获得此 版本软件新增功能及最新的仿真器、实验仪安装、新增功能和使用信息，这些信息往往未及写 入本手册。

3. 2如何使用星研集成环境软件

下边几节，介绍如何使用星研集成环境软件：3. 2.1使用汇编语言，将数据段中3000H〜

30FFH单元的内容传送给6000H-60FF中；然后比较。3.2.2使用Turbo C,重新编写第一个实 验。

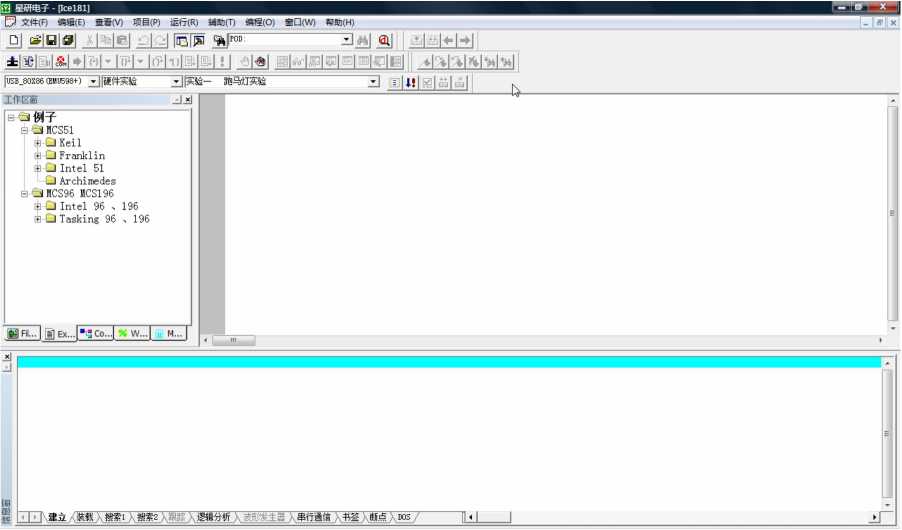
1. 2. 1数据传送程序（ASM）

星研集成环境软件推荐您使用项目为单位来管理您的程序。如果您做一个简单的实验，或 只希望看一个中间结果，您可以不建立项目文件，系统需要的各种设置，来源于“缺省项目”。 本节不使用项目文件。

本例子旨在通过建立一个具体的程序来介绍星研集成软件的使用方法以及它的强大的调试 功能。使用户很快的上手，体验到我们软件功能的强大和方便。

本实例是将数据段中3000H-30FFH单元的内容传送给6000H〜60FFH单元中；再将它读出 与3000H-30FFH单元中数据比较。程序是用汇编语言来编写。下面介绍相应的操作步骤：

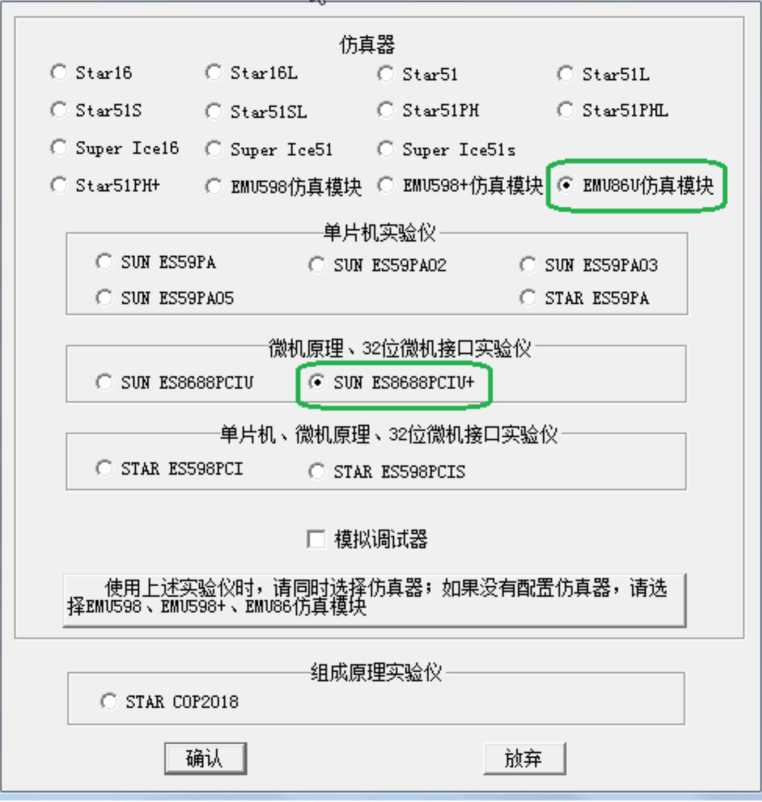
首先运行星研集成软件。启动画面如图：



1、选择仿真器或仿真模块

执行［主菜单》辅助》仿真器］,出现一个对话框:

"选择仿真器、实验仪 z Ih&T



请选择实验仪：SUN ES86PCIU+；仿真器：EMU86U仿真模块；选择“确认”。

**如果选择“模拟调试器”，实验仪电源不用开启，使用微机CPU模拟执行程序，可以调用 附件中的软中断，但无法对I/O接口操作。**

**学生在做实验前，可以选择“模拟调试器”，在星研集成环境中编写程序，对它编译、连 接，解决语法错误，使用模拟调试功能，初步调试；可以大幅度减少在实验室中做实验的时间。**

2、设置缺省项目

执行［主菜单》辅助》缺省项目］,出现一个对话框：

r MCS51

r STAR PCI

*C* 8086 (EMU86)

r USB\_80X86 CEMU86)

r STAR ES196仿真模块

请选择"8086 (EMU86)”。

点击进入下一步：“选择语言”

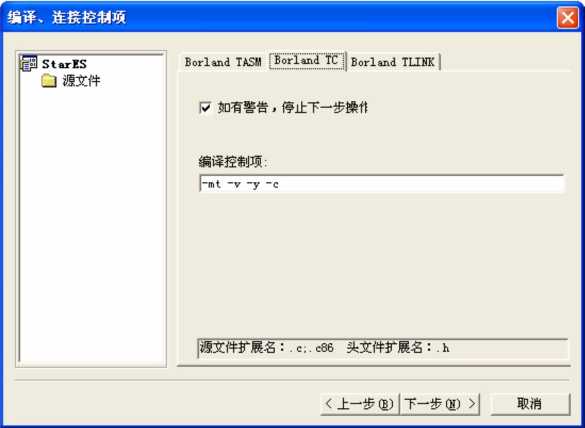
使用哪几种语言

选择语言

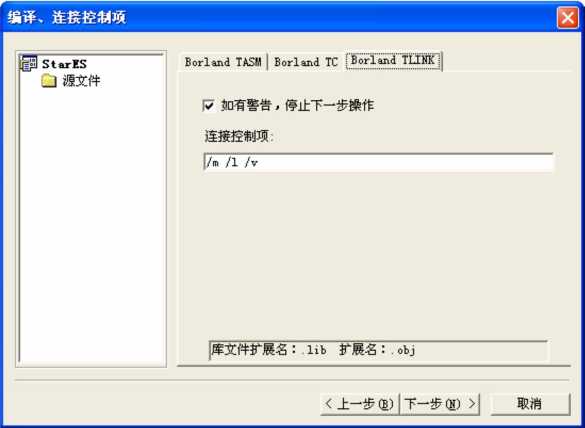
e蚤虽［五遡习日込卫◎卫匸了电匹區⑪

<上一步@) |下一步@) >| 取消

您可以根据自己的需要以及程序的类型作相应的选择，本实例选择Borland公司的Turbo C (3.0)、TASM (5.0)(请确定在选择语言之前已经安装好相应的编译软件)。然后再点击进入 下一步：“编译、连接控制项”

memory model 请选择 tiny,缩写为mt (也可以选择 其它模式)；如果需要源程序 级别调试，必须使用-v -y 控制项，为了支持多文件编 译、连接，必须使用-c控制项。

一般不必改变Turbo C的 编译控制项。

仿真模块EMU86提供 64K 仿 真 RAM （IS61LV6416）,作程序 段（CS）、数据段（DS）、 附加段（ES）、堆栈段（SS） 使用。

编译、连接控制项

**StarES**

□源文件

**Borland TASM | Borland TC | Borland TLEHK |**

**17**如有警告，停止下一步操作

编译控制项：

**|/zi /I-**

如果需要源程序级别调 试，必须使用/zi /I控制项。

一般不必改变Tasm的编 译控制项。

|源文件扩展名**：.asm;. %86** 头文件扩展名**：• inc**

<上一步|下一步@） >| 取消

如果需要源程序级别调 试，必须使用/m/1 /v控制项。

一般不必改变TLINK的 连接控制项。

然后再点击进入下一步：''存贮器出借方式”

B4区的SRAM （二片 62256）也可以作数据段 （DS）、附加段（ES）、堆 栈段（SS）使用，但需要 连接地址、数据总线、读 信号、写信号。

3、建立源文件 \_

下面我们建立源文件，执行［主菜单》文件》新建］,（或者点击图标°）打开窗口如下:



首先选择存放源文件的目录，输入文件名，注意：一定要输入文件名后缀。对源文件编译、 连接、生成代码文件时，系统会根据不同的扩展名启动相应的编译软件。比如：\*.asm文件， 使用TASM来对它编译。本实例文件名为move, asm。窗口如下：



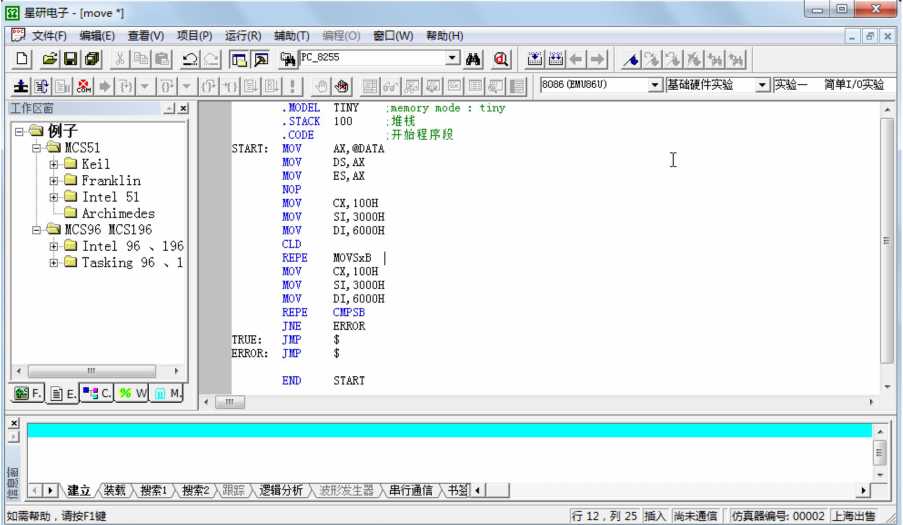
按“确定”即可。然后出现文件编辑窗口:

|  |  |  |
| --- | --- | --- |
| **r**  **iR1 Move, ass** | | E®区T |
|  |  |  |
| 1 |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 输入源程序，本实例的源程序如下： | | | |
|  | .MODEL | TINY | ;memory mode : tiny |
|  | .STACK | 100 | ;堆栈 |
|  | .CODE |  | ;开始程序段 |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |

|  |  |  |
| --- | --- | --- |
| MOV | | ES, AX |
|  | NOP |  |
|  | MOV | CX, 100H |
|  | MOV | SI,3000H |
|  | MOV | DI,6000H |
|  | CLD |  |
|  | REPE | MOVSxB |
|  | MOV | CX, 100H |
|  | MOV | SI,3000H |
|  | MOV | DI,6000H |
|  | REPE | CMPSB |
|  | JNE | ERROR |
| TRUE: | JMP | $ |
| ERROR: | JMP | $ |
|  | END | START |

输入源程序，如下图:



这样一个源文件就建立好了。

1. 编译、连接文件

首先选择-个源文件，然后可以编译、连接文件了。对文件编译，如果没有错误，再与库 文件连接，生成代码文件（DOB、EXE文件）。编译、连接文件的方法有如下二种：（1）使用［主 菜单》项目》编译、连接］或［主菜单》项目»重新编译、连接］”；（2）点击图标圍或窗 来“编译、连接”或“重新编译连接”。

“编译连接”与“重新编译、连接”区别：“重新编译、连接”不管源文件是否修改、编译 软件是否变化、编译控制项有无修改，对源文件编译，如果没有错误，再与库文件连接，生成 代码文件（DOB、EXE文件）。编译、连接过程中产生的信息显示在信息窗的“建立”视中。编

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 二  |6  w  和  e  保  e  忠  © B  i |  |  | [c:\XingYan\move.exe - 0 error (s), 0 warning (s) | RR 討討 |
| **▼** |  |  |  |
|  |  | **in** | 1 > |

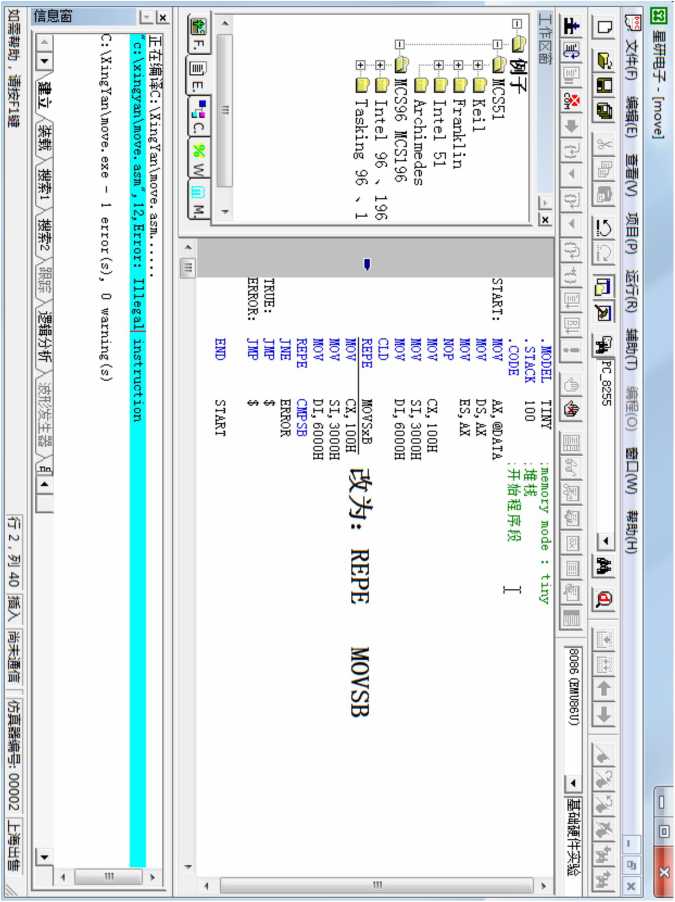
辐萍血韶沛sa&B苕

洲浴皿禅芦斗Mms圧離冷辛->M总屮圧sshk。

28

5.養

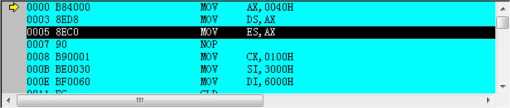
张申>蛊坯劳斛记tt錮m郵萍rtiaa口“弹h〔卄淵惟》議秤》涵a「錄斎茴笛吻“



仿真器、实验仪配套的通信线可以与微机USB 口相连，即为USB通信线，请选择USB。 对于最下面一行的校验，通常您不必选中它，可以提高传送DOB、HEX、BIN文件时的速度。 在进入调试状态以前，你还必须确定实验仪与微机的正确连接，电源接通，开关打开。

编译、连接正确后，可以开始调试程序。进入调试状态方法有:

1. 执行］主菜单》运行》进入调试状态］
2. 点击工具条的
3. 执行［主菜单》运行》装载DOB、HEX、BIN文件］ 进入后的窗口如下：



［51星硏电子-［move］

鬥文件(F)编辑(E)童看(V)项目(P)运行(R)靖助(T)扁程(Oi窗d(W)帮助(H)

□j国旧|创 对电i@i ©gi陌国 邑戶存

去|£| 團 |利▼ |别▼卜 |弋｝| 訓叫！ | | 刨 | |8086(EMU86U)

三］戲］@||團幽|.|\*||川％|%|廉皿皿||

三］|基础硬件实验 三］|实验一 简单I/O实验

REPE

Ml'V

MC'V

MOV REPE

START: MOV MOV

MOV

AX, ©DATA

DS, AX

CX,10OH

SI, 3000H

DI, 6000H

MOVSB

CX, 100H

SI, 3000H

DI,6000H

CMPSB

寄存窗

'ZF

* \_ofJF

AF | |CT 'JFj [0

* jlit

@ 催 9||DS：Memory 二| Address：|Q000

0000

0010

0020

0030

0040

B8 40

60 FC

02 EB

00 00

00 00

00 8E F3 A4 FE EB

00 00

00 00

D8

B9

FE

00

00

8E CO

00 01 00 00 00 00

00 00

| 00 | 01 | BE | 00 | 30 | BF | 00 |
| --- | --- | --- | --- | --- | --- | --- |
| 30 | BF | 00 | 60 | F3 | A6 | 75 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1 |  |  |  | | |  | ► |

n

B9 00 00 00 00

90 BE

00

00

00

| 正在装载：c: \x ingyan\move. dob | | **▲** |
| --- | --- | --- |
| 起始地址： | 000000H |  |
| 结束地址： | 00009fH |  |
| 已装载字节数： | OOOOaOH | = |
| 装载完毕 |  |  |
|  | | |
|  |  | - |
| >」'建立；装载［搜索1搜索2卜| | | *2T* |

如需帮助，请按F1键

习 c|y|

| 0000 | 00 | B8 | 00 | A2 | 40 | Fl C4 | 05 | 80 | CO | *62* | IB | 00 | 73 | 88 | 09 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0010 | 28 | D1 | 40 | 6C | 00 | 2B CO | 22 | 00 | 01 | *81* | 40 | 83 | CO | A0 | B2 |
| 0020 | 5E | 0C | 58 | 90 | 02 | 28 40 | 71 | E4 | A5 | FF | 01 | EC | 38 | 55 | 74 |
| 0030 | 82 | 35 | DF | 24 | E0 | 21 C2 | 86 | 14 | C2 | 12 | 58 | CO | 4B | 11 | 03 |
| 0040 | 63 | E4 | 12 | 08 | 47 | 58 72 | EE | 04 | 22 | C3 | *47* | 6D | 48 | 8A | 29 |
| 圈 | 存贮0工 | | |  | | 存贮2 >. | 存贝| **4** | | J H | | | | | | |

***w\***

▼ | Address ：|0000

\_d

通借主P鯉輕2也仿真器蕖号:00002上§出售

在整个图片中我们可以看到相对应的窗口信息。在“工作区窗”的“CommonRegister”中 我们可以了解通用寄存器的信息。中间的窗口为源程序窗口，用户可在此设置断点，设置光标 的运行处，编辑程序等。寄存器窗我们可以看到…些常用的寄存器的数值。存贮窗1、存贮窗2 显示相应的程序段(CS)、数据段(DS)、10设备区的数据，还有变量窗，自动收集变量显示其 中。反汇编窗显示对程序反汇编的信息代码、机器码、对应的源文件。在信息窗的“装载”视 中，显示装载的代码文件，装载的字节数，装载完毕后，显示启始地址，结束地址。这种船坞 化的窗口比通常的窗口显示的内容更多，移动非常方便。用鼠标左键点住窗口左边或上方的标 题条，移动鼠标，将窗口移到您认为合适的位置；将鼠标移到窗口的边上，鼠标的图标变成可 变化窗口时的形状，用鼠标左键点住，移动鼠标，变化一个或…组窗口的大小。在调试过程中， 可以根据您的需要，在［主菜单》查看］中打开：寄存器窗、存贮器窗1、2、3、观察窗、变量 窗、反汇编窗。您也可以通过［主菜单》辅助》设置》格式］,设置每一种窗口使用的字体、 大小、颜色。移动窗口到您喜欢的位置、大小。



设置

编辑辟I调试I工作区 格式I

种类（Q： 字休（E）：

崔窗器器窗窗窗绘 文作息存贮察量栈A 源工信寄存观变堆V

首先在“种类”中选择一个窗口，然

亦例：

星研集成环境软件

I宋体

土卩。

大小（£）:

后选择“字体”、“大小”，在“颜色”

中选择某一类，在“前景”、“背景”

中选择您喜欢的颜色。

确定 I 取消

对于高级语言，在您的程序前有…段库文件提供的初始化代码，O （当前可执行标志）不 会出现在您的文件行上，如果您使用C语言，可将光标移到main函数上，按F4功能键，让CPU 全速运行到main行上后停下；如果您使用PL/M语言，按F7功能键，让CPU “单步进入”，运 行到您的任何一个可执行后停下。

您可以使用以下命令调试您的程序：

0 设置或清除断点（功能键为F2）

在当前光标行上设置或清除一个断点 单步进入（功能键F7）

单步执行当前行或当前指令，可进入函数或子程序。

* 连续单步进入（功能键Ctrl + F7）

连续执行“单步进入”，用鼠标点击凰I或按任意键后，停止运行。

**H** 单步（功能键F8）

单步执行当前行或当前指令，将函数或子程序作为一条指令来执行。如果当前行中含 有函数、子程序或发生中断，CPU将执行完整个函数、子程序或中断，停止于当前行或当 前指令的下一有代码的行上。

* 连续单步（功能键Ctrl + F8）

连续执行“单步”，用鼠标点击咼I或按任意键后，停止运行。

%} 运行到光标行（功能键F4）

从当前地址开始全速运行用户程序，碰到光标行、断点或用鼠标点击同1,停止运行。

**13** 全速断点（功能键F9）

从当前地址开始全速运行用户程序，碰到断点或用鼠标点击咼I,停止运行。

■ 全速运行（功能键Ctrl + F10）

从当前地址开始全速运行用户程序，此时，按用户系统的复位键，CPU从头开始执行 用户程序，用鼠标点击咼I,停止运行。全速运行时，屏蔽了所有断点，即不会响应任何断点。

=11 停止运行

乐 终止微机与仿真器之间通信（功能键ESC）。

注意：欲终止微机与仿真器之间通信，功能键**ESC**是一个很方便的键，它的效果比点击相 应的图标的效果要好。建议用户多用**ESC**键。在系统运行“连续单步”或者“连续单步进入” 时**ESC**键被禁止，这时用户可以按键盘的其他任意键停止其运行。

1. 调试的方法及技巧

一般来说，用户的程序或多或少的会有一些逻辑错误，我们的仿真器、实验仪和星研集成 软件可以帮助用户很快的定位，很快的查出相应的错误。

在调试状态的窗口中我们可以看到很多的窗口，用户只要熟练地应用这些窗口来观察、分 析数据就会很快的调试好程序，达到事半功倍的效果。

进入调试界面后，由于我们本次操作需要观察二个数据块：数据段3000H-30FFH,数据段

6000H-60FFH,可以打开一到二个存贮器窗口，具体操作是：［主菜单＞＞查看］

***v***文件工具栏⑸

***v***建立工具栏(E)

***v***运行工具栏(助

***v***编辑工具栏(I)

▼状态栏⑸

***v***工作区窗

***v***信息窗(Q)

***v***寄存器窗(R)

***v***存贮器窗1 (1)

***v***存贮器窗2(2)

存贮器窗3 (3)

观察窗(I)

***v***变量窗(工)

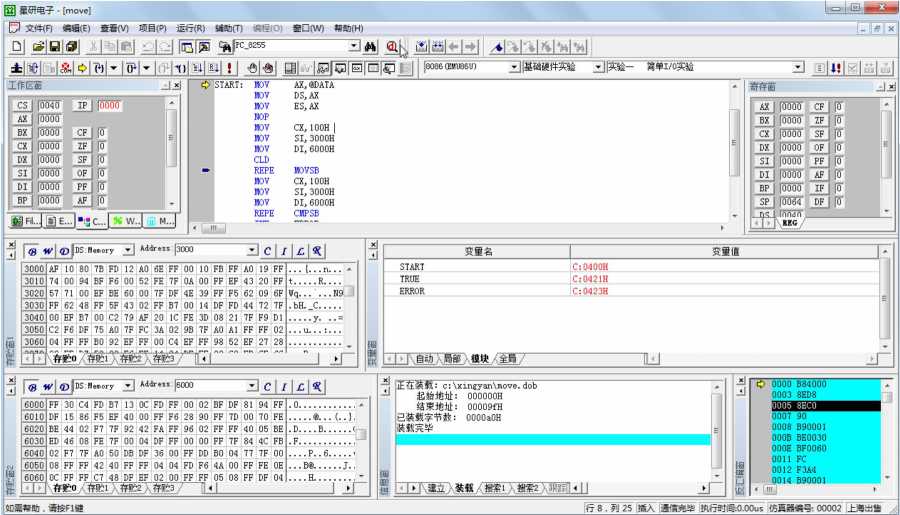
堆栈窗

断点窗(B) Alt+F2

***V***反汇编(D)

刷新

然后根据你的需要打开不同的窗口。调整后的调试界面为:



由于我们本次操作主要是观察存贮器窗口,所以我们拉大了这两个存贮器窗口的大小。每 个窗口设置了 4个分页项：空/輕巫空垂Z,我们可以在不同的分页项设置不同的 观察数据空间以及地址范围。在何：Memory二J中可以选择CS： Memory, DS： Memory, I/O,根据 需要可以做不同的选择。在A迪"S： 勺中可以直接输入地址，然后按回车，就可以直接

转到我们输入的地址的窗口上面观察数据。由于我们在此程序中的写入数据的RAM空间分别为 DS： 3000H〜30FFH、DS： 6000H〜60FFH,故我们建立的分页项如下：

存贮0分页项:

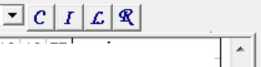


存贮1分页项:

莎 ***w \* ®]|DS:Memory Address:|6000**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6000 | 00 | FF | 00 | FF 00 FF 00 | FF | 00 | FF | 00 | FF | 00 | FF | 00 | FF |  |
| 6010 | 00 | FF | 00 | FF 00 FF 00 | FF | 00 | FF | 00 | FF | 00 | FF | 00 | FF |  |
| 6020 | 00 | FF | 00 | FF 00 FF 00 | FF | 00 | FF | 00 | FF | 00 | FF | 00 | FF |  |
| 6030 | 00 | FF | 00 | FF 00 FF 00 | FF | 00 | FF | 00 | FF | 00 | FF | 00 | FF |  |
| 6040 | 00 | FF | 00 | FF 00 FF 00 | FF | 00 | FF | 00 | FF | 00 | FF | 00 | FF |  |
| 6050 | 00 | FF | 00 | FF 00 FF 00 | FF | 00 | FF | 00 | FF | 00 | FF | 00 | FF |  |
| 6060 | 00 | FF | 00 | FF 00 FF 00 | FF | 00 | FF | 00 | FF | 00 | FF | 00 | FF |  |
| 6070 | 00 | FF | 00 | FF 00 FF 00 | FF | 00 | FF | 00 | FF | 00 | FF | 00 | FF |  |
| 小 | **1**存**SCO** ；［存疋**1/**存贮**2 h**存贮**3** | | | | | |  | | **q** | l> | |

我们这样设置界面的目的就是当用户要观察不同地址段的数据时，只要切换一下分页项就行了。 由于本次程序需要同时观察DS： 3000〜30FFH、DS： 6000H〜60FFH,所以打开二个存贮器窗。 如图：



| 3000 | AF | 10 80 | 7B | FD 12 | A0 | 6E FF | 00 | 10 | FB | FF A0 | 19 | FF | …L..n |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3010 | 74 | 00 94 | BF | F6 00 | 52 | FE IFF | 0A | 00 | FF | EF 43 | 20 | FF | t R C . |
| 3020 | 57 | 71 00 | EF | BE 60 | 00 | 7F DF | 4E | 39 | FF | F5 62 | 09 | 6F | Wq...... N9.. b. o |
| 3030 | FF | 62 48 | FF | 5F 43 | 02 | FF B7 | 00 | 14 | DF | FD 44 | 72 | 7F | .bH. \_C Dr・ |
| 3040 | 00 | EF B7 | 00 | C2 79 | AF | 20 1C | FE | 3D | 08 | 21 7F | F9 | DI | ……y. ..=• !... |
| 3050 | C2 | F6 DF | 75 | A0 7F | FC | 3A 02 | 9B | 7F | A0 | Al FF | FF | U2 | ••: |
| 3060 | 04 | FF FF | B0 | 92 EF | FF | 00 C4 | EF | FF | 98 | 52 EF | 27 | 28 | R・‘( |
|  | .，渝 1\*贮2 X 擬3 / | | | | | |  |  |  | |  |  | 1 ► |

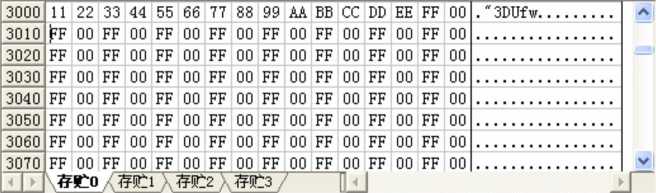
琲 @ **w|<p[|DS:Memory** 二]**Address:|3000**

| 6000 | FF | 30 | C4 | FD | B7 13 | OC | FD FF | 00 | 02 BF | DF 81 | 94 | FF | .0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6010 | DF | 15 | 86 | F5 | EF 40 | 00 | FF F6 | 28 | 90 FF | 7D 00 | 70 | FE | @・・・(・・}.p. |
| 6020 | BE | 44 | 02 | F7 | 7F 92 | 42 | FA FF | 96 | 02 FF | FF 40 | 05 | BE | .D. ...B . |
| 6030 | ED | 46 | 08 | FE | 7F 00 | 04 | DF FF | 00 | 00 FF | 7F 84 | 4C | FB | .F L. |
| 6040 | 02 | F7 | 7F | A0 | 50 DB | DF | 36 00 | FF | DD B0 | 04 77 | 7F | LIU | .…P.. 6 w.. |
| 6050 | 08 | FF | FF | 42 | 40 FF | FF | 04 04 | FD | F6 4A | 00 FF | FE | 0E | ...B@ J・・・・ |
| 6060 | OC | FF | FF | C7 | 48 DF | EF | 02 00 | FF | FF 05 | 08 FF | DF | U4 | ....H |

**2)@** 钳| **(DpS:Memory** 二]**Address：(6000**

软件中总共存在3个存贮器窗。可以同时观察三个不同的地址。

存贮器窗口支持数据的直接修改功能。本软件的所有窗口中的数据都支持直接修改功能。 用户可以根据自己的需要在窗口中直接修改数据。比如：执行程序前，将DS： 3000H-300FH中 的数据改为 11、22、33、44、55、66、77、88、99、AA、BB、CC、DD、EE、FF、00,在相对应 的地址中直接输入数据即可。如图：

一般刚刚写好的程序，在进入调试状态后，执行“单步”或者“单步进入”，我们推荐您能 记住这些操作的相对应的功能键，这样您就在调试程序的过程中很方便。

而 **fy|<D|DS:Memxy** 二| **Address:|3010** 二| **C | F | £ | % |**

在刚才的调试程序中我们多次执行“单步进入(F7)”命令，在工作区窗口的CommonRegister 视中查看通用的寄存器：

我们可以观察到在本程序中所使用的一些寄存器的变化，比 如CX、DI、SI的数值的变化，每一次MOVSB, CX减一，DI、 SI加一，从DS： [SI]取出的数值复制给ES： [DI]中：11、 22、 33、 44、 55、 66……。

IP | [0034

[oW foolo" |oooo, fOOFA

DX | [OOOCF SI I |30F7

bp | [ooocT SP I [0062

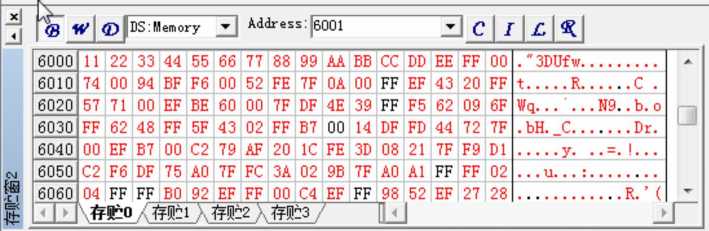
DS | [0040\*

ES | f0040" SS | [0044

1010-ofo-1 -110-1 CFIJdSFI-odPLlAFIIdil

笛| Fi“.[ [g E..]七 C... % W..

我们可以看到存贮器窗口中的相对应的RAM的数据的变化。比如



其中右边为相应数据的ASCII码。切换分页项我们可以观察到其它地址的数据。

把光标移动到第13行(MOV CX, 100H)上，点击图标％｝(功能键F4),全速运行到光标 行，检查DS： 6000H〜60FFH内容，是否与DS： 3000H〜30FFH相同，如果完全一样，说明以上 程序没有任何问题。

将光标移到第18行(TRUE： JMP $)的左边，鼠标变为①，点击鼠标，在该行上设置 了一个断点，也可以用鼠标点击该行，将光标移到鼠标处，点击图标电I (功能键为F2),设置 断点，重复操作，清除断点；在19行(ERROR： TMP $)上设置断点。

* **TRUE:**
* **ERROR:**

V V V V

MOCLRE恥MOMORE

NEMPMP

ND

**DI, 6000H**

**MOVSB**

**CX, 100H**

**SI, 3000H**

**DI,6000H CMP SB**

**ERROR**

**$**

**$**

**START**

点击图标冒1（功能键F9）, CPU全速断点运行到光标处停下，如果在第18行上CPU停止运 行，表示运行结果正确；如果CPU运行到第19行，停止运行，与期望值不符，表示程序有问题, 将光标移到第13行上（具体操作是：用鼠标点击该行，然后再点击图标自），使用单步进入命 令F7或连续单步进入命令Ctrl + F7,检查结果，判断程序出错原因。

**3. 2. 2**数据传送程序**（C）**

本例子使用项目文件来管理，旨在通过建立一个具体的项目来介绍星研集成软件的使用方 法。如果您的系统有几个文件组成，就必须使用项目文件。

1、建立项目文件 \_

执行［主菜单》文件》新建］,（或者点击图标D）打开窗口如下：

新建

新建文件I创建项目文件］

厂添加到项目文件中

文件名:

由于星研集成软件是以项目为单位来管理程序 的。所以我们在建立文件之前先要建立项目文件。 点击''创建项目文件”分页项，如图示：

位于哪个目录:

**|C:\WIHDOWS\Desktop**

确定 I 取消

新建文件 创逹项目文件|

项目文件

厂

位于哪个目

|c：\XingYan

我们可以输入项目文件名，以及选择目录，星 研集成软件在您输入一个项目文件名时，就建立了 以项目文件名为名的一个文件夹，以后您在编译、 调试过程中生成的所有文件都在此文件夹里。这体 现了星研集成软件的人性化设计。键入项目文件名 “move”,如下：

确定 I 取消



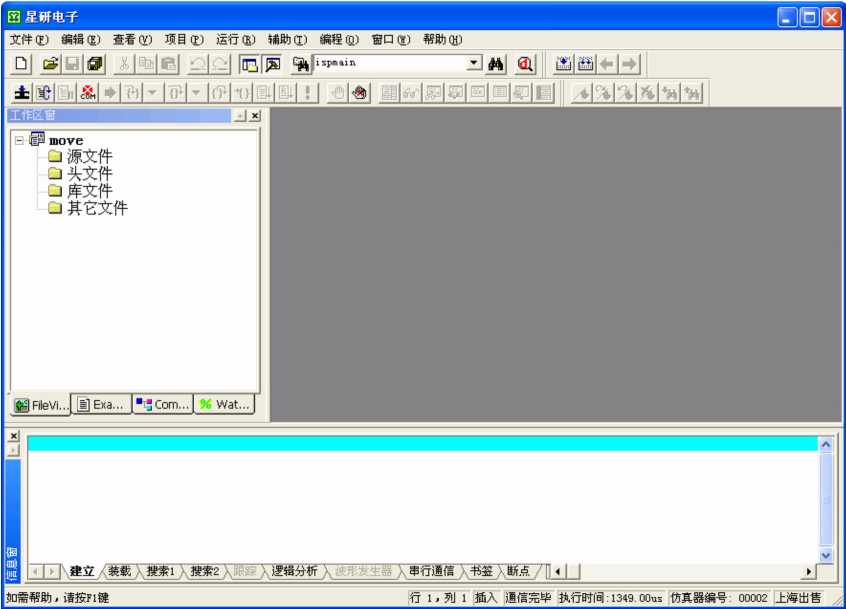
然后按确定，进入“设置项目文件”部分。

2、 设置项目文件

设置项目文件与设置缺省项目操作完全一样，请参阅上一节。

3、 建立源文件

建立好项目文件的窗口如下图所示：



工作区窗的项目视中，包含“源文件”、“头文件”、“库文件”、“其它文件”，“其它文件” 中通常包含对该项目用途作一些说明的文件。“库文件”通常包含编译软件自带的OBJ文件、LIB 等库文件。 \_

下面我们建立源文件，执行［主菜单》文件》新建］,（或者点击图标°）打开窗口如下：

选定刚才建立的项目文件的文件夹，输 入文件名，注意：一定要输入文件名后缀。 系统会根据不同的后缀名给文件归类。比 如：\*.asm文件系统会自动归类为源文件。 选中“添加到项目文件中”，系统自动将该 模块文件加入到项目中。本实例文件名为 move. Co

新建

按“确定”即可。然后即出现文件编辑窗口:

|  |  |  |  |
| --- | --- | --- | --- |
| r肆 | ***ve \_ c*** |  | |
| 1 |  |  |  |
| J |  |  |  |
|  |  |  |  |

程序清单：

#include 〃conio. h〃

void main(void)

{

unsigned char array1[0x100];

unsigned char array2[0xl00];

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| int i; | | | | |
| for | (i = 0; i | < | 0x100; | i++) |
|  | array1 [i] | 二 | i； |  |
| for | (i = 0; i | < | 0x100; | i++) |
|  | array2[i] | 二 | array1[i]; | |
| for | (i = 0; i | < | 0x100; | i++) |

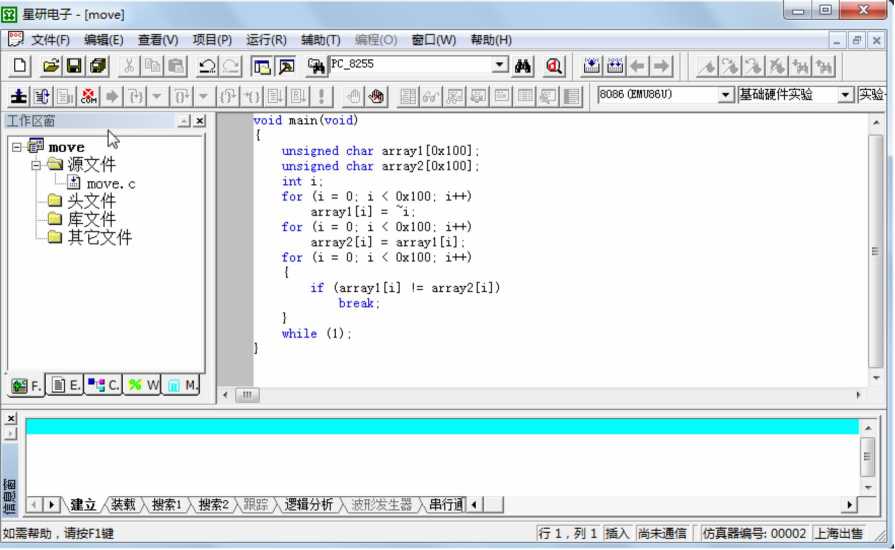
if (array1[i] != array2[i]) break;

}

while (1);

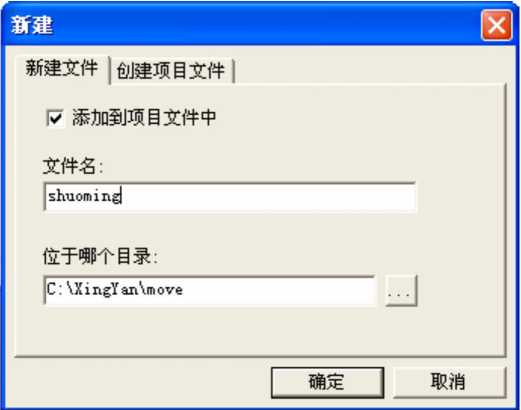
}

建立好文件的窗口如下：

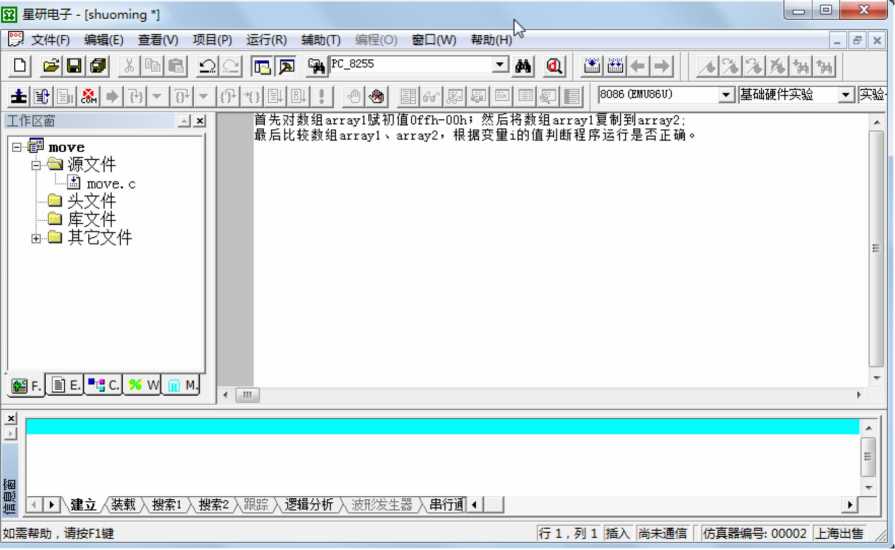


附：

注意：若在新建文件时不输入文件后缀，则其文件不会保存在源文件那一项，而是保存在其他 文件的文件夹中。一般我们建立对项目说明的文件即可用此方法。如图建立一个本程序的说明 文档 "shuoming"



然后编辑文档，如下图:



然后保存，就可以了。

4、 编译、连接文件

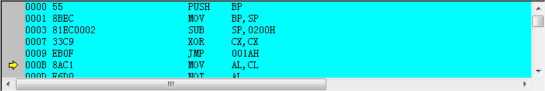
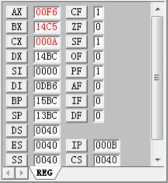
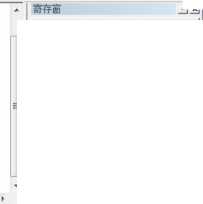
在建立好项目文件、源文件后，就可以编译、连接文件了。对工作区窗项目视的“源文件” 中所有模块文件编译，如果没有错误，再与“库文件”中所有库文件连接，生成代码文件（DOB、 EXE文件）。编译、连接文件的方法有如下三种：（1）在工作区窗的项目视中按鼠标右键，系统 弹出快捷菜单，选择“编译、连接”或“重新编译连接”。（2）使用［主菜单》项目》编译、 连接］或［主菜单》项目》重新编译、连接］”。（3）点击图标圜或窗来“编译、连接”或“重 新编译连接”。

“编译连接”与“重新编译、连接”区别：“重新编译、连接”不管项目中有无添加、删除 模块文件、编译软件是否变化、编译控制项有无修改、模块文件有无修改，对“源文件”中所 有模块文件编译，如果没有错误，再与“库文件”中所有库文件连接，生成代码文件（DOB、EXE 文件）。编译、连接过程中产生的信息显示在信息窗的“建立”视中。

详细请参阅上一节。

5、 调试项目文件

下面我们进入调试状态，调试C语言程序时，我们观察的比较多的是寄存器窗、观察窗和 变量窗。所以我们把这些窗口放在前台，并调整至适当的大小。如图所示：



\_D］国旧|0|紂屉|@| 2冋 陌国 邑吟^

三］鱼］旦］II创二"|\*| I /＜|%|%|廉|甸词I

TirFir><i

! |刨创 国"厢而画口原圍| |8086(flni86U)

三］|基础砸件实验~三］|实验一简单I/O实验

*% W..I*

unsigned char arrayl[0x100]: unsigned char array2[0x100]: int i;

for (i = 0; i < 0x100; i卄) arrayl [i] - \*i：| for (i = 0; i < 0x100;

array2[i] = arrayl[i]; for (i = 0; i < 0x100; i++)

if (arrayl[i] != array2[i]) break;

} while (1):

*1* i nr |

-nr

| 变里名 | 变里值 | • |
| --- | --- |
| i | 10 |
| E arrayl  El array2 | {FFH, FEH, FDH, FCH, FBH, FAH, F9H {00H, 00H, 00H, 00H, 00H, 00H, 00H |
| —[0] | 00H |
| -[1] | 00H |
| —[2] | 00H |
| -[3] | 00H |
| -[4] | 00H ° |
| i ► 1双察1 观察2人观察3上观察4「4丨 | ► | |

| 起始地址：000000H 结束地址：0015bfH 已装载宇节数：0015c0H 装载完毕 | fl |
| --- | --- |
|  | |
|  | ▼ |
| ＞撻立；1装载./；搜索1其搜索2「＜ | *1T* |

一IX1I-11理g

圈星硏电子-［move］

/文件(F)輛(E)查看(V)项目(P)运行(R)融CD嗝程Q 窗口㈣帮助(H)

|行8 ,列24插入 通信完毕 执行时间:0.Q0us仿真器编号:00002 亠出售

如需帮助，请按Fl键

我们执行“单步”命令时就会在变量窗口中看到相应的变量的变化:



变量窗包含“自动”、“局部”、“模块”、“全局”四个标签视。

自动： 星研自动搜集当前行(PC指针对应的文件行)及前二行上的变量。通常这三

行有您最关心的变量，也是星研集成环境的一大特色。

局部： 显示当前函数或当前过程中的所有变量。

模块： 显示当前模块文件中所有模块级变量。

全局： 显示所有全局变量。

经常查看的变量分别放入观察窗的4个标签视中，您会感觉非常方便、快捷。在观察窗口 中我们可以随意的添加我们想要观察的变量，具体方法是：(1)在文件窗中，用鼠标左键双击 变量名，按住鼠标左键，将该变量名拖至观察窗中，释放鼠标左键，星研自动将该变量添至观 察窗中。(推荐方法)(2)用鼠标左键双击观察窗中的虚线框，出现一个编辑框，在编辑框中 输入…个变量名，输入回车即可。观察窗中的四个页面项的作用完全-样，当您要观察的变量 很多时，用户可以在不同的页面项输入观察变量，这样观察时只要点击一下页面项就可以了。 这也是星研软件的人性化设计的-个方面。比如；我们在观察窗中添加变量i, arrayl, ar ray 2 观察，如图：

变量值

变量名

0

{OOH, FFH, FFH, OOH, FFH, OOH, FFH, OOH, FFH, OOH, FFF

array2

array!

| {FFH, OOH, OOH, FFH, OOH, FFH, OOH, FFH, OOH, FFH, OOl|

FFH

OOH

OOH

FFH

OOH

FFH

v nrnr 工 *I、* r ◄ I I ►!

这也就方便了用户的调试和分析。

在第…个循环中，连续执行单步操作，我们可以看到i的值在递增，OFFH〜00H赋给 arrayl[0]〜arrayl [255]；第二个循环中，连续执行单步操作，我们可以看到数组arrayl复制 给数组array2；第三个循环中，连续执行单步操作，比较数组arrayl、array2中数据，判断 程序是否正确（如果i=256,程序执行正确；其它值，程序执行错误。）

当然这些窗口中的数值也支持直接修改功能，双击要修改的数值，出现一个编辑框，在编 辑框中输入数据，输入回车即可。用户在调试很长的程序时，为了分段检测，若前段程序有问 题，我们可以在下一段程序前用修改变量的方法先使程序满足所需要运行的条件，这样就方便 了下…段程序的检测。

另外一种查看变量方法：鼠标移到文件窗、反汇编窗口中的变量、寄存器上，半秒钟后， 在它们的旁边，会显示相应的值。

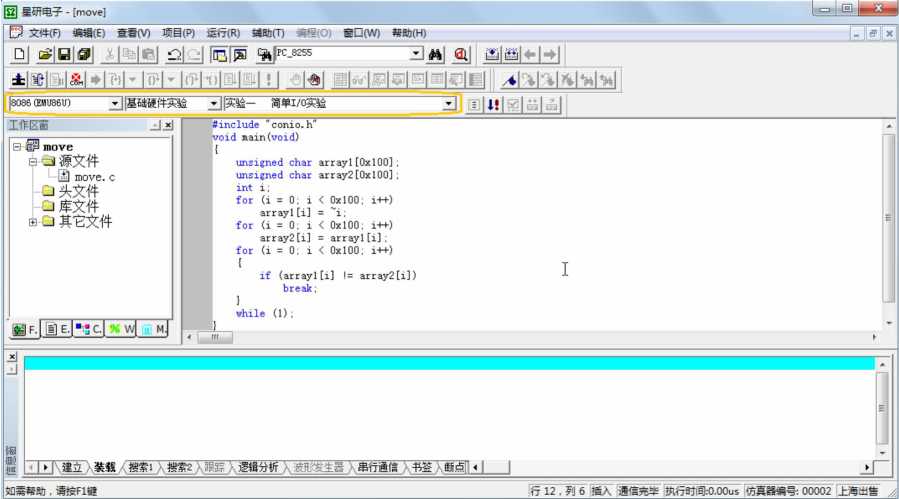
|  |  |  |  |
| --- | --- | --- | --- |
|  | {  unsigned char arrayl[0x100]; unsigned char array2[0x100]: int i;  for (i = 0: i < 0x100; i++) | | ▲ |
|  |  | arrjyl[i] = "i; |  |
|  | for | (i jarrayl = {FFH,FEH,FDH,FCH,FBH,FAH,F9H,F8H,F7H,F6H,F5H,00H,00H,00H,00H,00H,00H,0| |  |
|  |  | arr |  |
|  | for | (i = 0; i < 0x100; i++) | E |
|  | *\* | if (arrayl[i] != array2[i]) |  |
|  | i | break: |  |
|  | while (1):  } | |  |
| ***4*** 川 |  | **►** |  |

所以说星研集成软件为用户提供了许多实用和方便的观察、调试、分析的功能，其他的一 些功能我们会在后续的实例中作更详细的介绍。用户也可以依此类推，发掘出更多更好的为自 己调试程序方便的功能。

3. 3实验连线、演示实验、测试实验仪

演示实验

选择仿真器或仿真模块时，必须正确选择购买的实验仪，选择完毕后，会出现一个工具条



首先选择哪…类实验，例如：软件实验、基础硬件实验、综合实验；然后选择实验，点击回, 显示该实验对应的连线说明：



点击刨，星研软件自动将该实验的机器码装入实验仪，并全速运行该程序；如果按照上述 连线后，没有出现实验结果，可以怀疑与该实验相关的芯片出问题。

**完整自检**

SUN系列实验仪，如果配置了 Test Board测试模块，点击團，可以对整套实验仪检测，包 括测试模块：



首先，请正确选择串口，星研软件通过串口，向实验仪发出各种测试命令；然后，选择使 用的液晶，12864J： 12864图形点阵液晶；12864M：带汉字库的12864图形点阵液晶；12864C： 12864图形点阵液晶，通常同时配置触摸屏。

如果需要测试RS485,必须选择当前实验仪是主机还是从机；如果还需测试扩展模块，请 正确选择模块。

如果只需测试实验仪的一部分，请自行选择。

点击“开始测试”，根据对话框的下半部分提示，完成对实验仪的完整测试。(整个测试 过程，只需连接扁线、、二根连接线，非常适合新学期开始，对实验作-完整检测，了解实 验仪的实际状况。)

3. 4 频率计(EMU86)

如果您需要测试CPU的振荡频率、您电路中其它信号的频率，在调试状态，您可以选择频 率计功能：】主菜单》分析手段》频率计它可以测试100M以内的信号。

频率计

X]

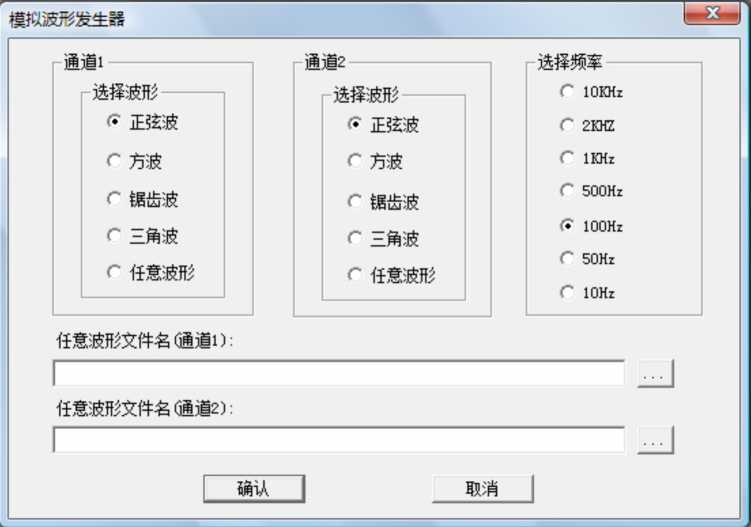
11,059,20

[ iltH(E)

EMU86仿真模块的FREQ与被测信号相连。 频率计、仿真部分可以并行运行。

3. 5模拟波形发生器(EMU86)

EMU86仿真模块可以提供2路模拟波形：正弦波、方波、锯齿波、三角波或自定义波形。 您可以选择模拟波形发生器功能：［主菜单》分析手段》模拟波形发生器］。



EMU86仿真模块的Wl、W2对应于通道1、通道2。

任意波形：首先创建一个BIN文件，包含100个字数据；每个字数据包含12bit二进制数, 即每个字数据的有效范围为0000H-0FFFH；根据您希望产生的波形，换算出100个字数据。

模拟波形发生器、仿真部分可以并行运行。

3. 6 TDS2、TDS2A (EMU86)虚拟示波器

TDS2虚拟示波器模块的安装软件在“实验仪\TDS2”文件夹；TDS2A虚拟示波器的安装软件 在"实验仪\TDS2A(EMU86) ”文件夹，运行SETUP. EXE文件即可进入安装界面，您只需按程序提 示一步一步进彳亍安装即可。

悔 H8.vi

文件(E)编辑(E)擾作9)工MCD窗口 (W)帮助但)

喝匾］

TDS-2A巩通道虚拟示欢赛

**22.5 25**

**10 12.5 15 17.5**

**WfflJ(ms)**

触雄制

验极性

Jb 下啣

诵道**A MB**

槪电平

**-1 ? 1**

**-2 x ， 2**

***3(* I**

〜广

曲线0空 曲线i **EM**

**1OA**

|  |  |
| --- | --- |
| 通道**A** | **i@ilB** |
| 垂直 | 垂直墮**B** |
| 5| | 5：| |
| **2.5**三［ | **2.5 el** |
|  | **oij >~v** |
| 25 1 |  |
| **-5-1** | 5 1 |
| E +«克减 | **i "HS**袞减 |
| 伏/格 | 秒/格 |
| **IV 0 5V 0.2V** | **2.5mslnJs5f)0250us** |
| 、'， |  |
| **2J**八\』**1V** 卜  **' A 25ms-** 、 **1 -50us**  **5V- *J* -0.05V “**  **50ms** j 一" **25us** | |
| **10V 0.02V** | **0.1s lOus** |
| 3^5 | **;^Z5msj|** |

MB

**A,B,A&B** A+b'a-B

**|~ I I A+B \* A-B A B A&B**

平均蚯**［o'V** 峰峰值**pi V** 频率『 **Hz**

平均蚯**［o' V**

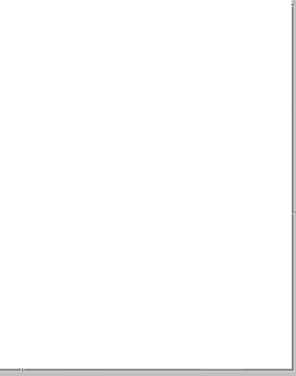
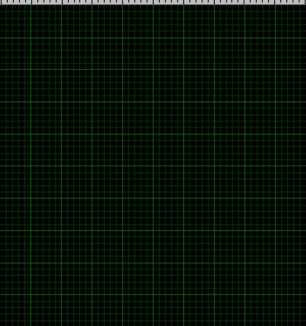
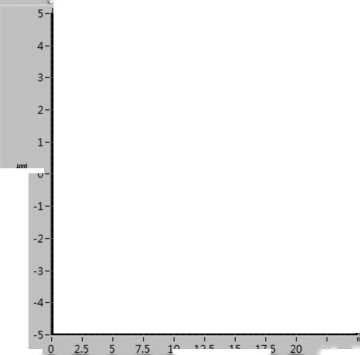
峰峰值**0 V**

频率**［o Hz**

启动停

**Cjs>**

点击“启动/暂停”按钮，可以启动或暂停虚拟示波器功能。操作与一般示波器类似。



41软件实验

软件实验部分共有七个实验组成,通过对这些实验程序的编写、调试，使学生熟悉8086/8088 的指令系统等，了解程序设计过程，掌握汇编程序、C程序设计方法以及如何使用实验系统提 供的各种调试、分析手段来排除程序错误。

实验一数据传送

一、实验目的

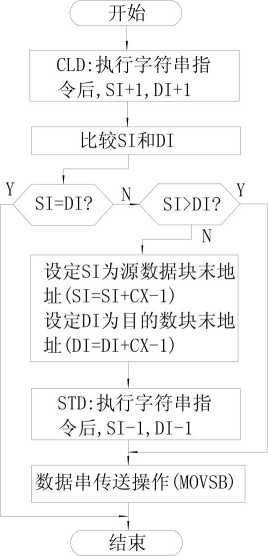
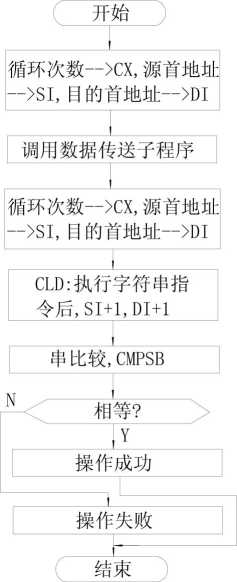
熟悉星研集成环境软件的使用方法。熟悉Borland公司的TASM编译器 熟悉8086汇编指令，能自己编写简单的程序，掌握数据传输的方法。

二、实验内容

1、 熟悉星研集成环境软件。

2、 编写程序，实现数据段的传送、校验。

三、程序框图



数据块传送比较程序框图

数据传送子程序

四、实验步骤

在DS段内3000H-30FFH中输入数据;使用单步、断点方式调试程序，检测DS段内6000H 60FFH中的内容。熟悉查看特殊功能寄存器、CS段、DS段的各种方法。

五、程序清单

.MODEL

TINY

|  |  |  |
| --- | --- | --- |
|  | .STACK  .DATA | 100 |
|  | .CODE |  |
| START: | MOV | AX, @DATA |
|  | MOV | DS, AX |
|  | MOV | ES, AX |
|  | NOP |  |
|  | MOV | CX, 100H |
|  | MOV | SI,3000H |
|  | MOV | DI,6000H |
|  | CALL | Move |
|  | MOV | CX, 100H |
|  | MOV | SI,3000H |
|  | MOV | DI, 6000H |
|  | CLD |  |
|  | REPE | CMPSB |
|  | JNE | ERROR |
| TRUE: | JMP $ |  |
| ERROR: | JMP $ |  |
| Move | PROC | NEAR |
|  | CLD |  |
|  | CMP | SI, DI |
|  | JZ | Return |
|  | JNB | Movel |
|  | ADD | SI, CX |
|  | DEC | SI |
|  | ADD | DI,CX |
|  | DEC | DI |
|  | STD |  |
| Movel: | REP | MOVSB |
| Return: | RET |  |
| Move | ENDP |  |
|  | END | START |

六、思考题

1、子程序Move中为什么比较SI、DI?

源数据块与目标范围有可能部分重叠，需要考虑从第一个字节开始复制（顺序复制），还是 从最后一个字节开始复制（倒序复制）。

实验二 双字节BCD码（十进制数）加法

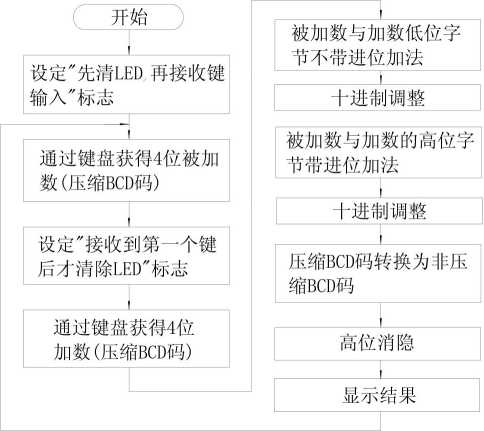
一、 实验目的

熟悉8086汇编指令，学会使用星研集成环境软件，能自己编写简单的程序，熟悉BCD码， 了解如何调用系统提供的子程序。

二、 实验内容

从键盘上输入4位被加数、加数，实现双字节BCD码（四位数）的加法，结果显示在数码管 上；熟悉使用断点、单步进入、单步、运行到光标处、修改PC指针、全速运行等各种调试手段; 熟悉查看特殊功能寄存器、CS段、DS段存贮器的各种方法。

三、实验框图



双字节BCD码加法程序框图

四、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS、AO、Al |  | A3 区：CS1、AO、Al |
| D3 区：PCO、PCI |  | F5 区：KL1、KL2 |
| D3 区：JP20、B、C |  | F5 区：A、B、C |

2、 在F5区的键盘上输入4位被加数、加数

3、 结果显示在F5区的数码管上

五、程序清单

.MODEL TINY

EXTRN InitKeyDisplay:NEAR, Display8:NEAR, GetBCDKey:NEAR

EXTRN Fl:BYTE

.STACK 100

.DATA

BUFFER DB 8 DUP(?)

|  |  |  |  |
| --- | --- | --- | --- |
| augend | DB | 2 DUP(?) | ;被加数 |
| addend | DB | 2 DUP(?) | ;加数 |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | InitKeyDisplay | ;对键盘、数码管扫描控制器8255初始化 |
|  | MOV | Fl,0 | ；先清除显示，再接收键输入 |
| START1: | LEA | DI, augend |  |
|  | MOV | CX, 4 | ；按键次数 |
|  | CALL | GetBCDKey | ；得到双字节十进制数（被加数） |
|  | MOV | Fl, 1 | ;接收到第-个键，才清除显示 |
|  | LEA | DI, addend |  |
|  | MOV | CX, 4 | ；按键次数 |
|  | CALL | GetBCDKey | ；得到双字节十进制数（加数） |
|  | MOV | AL,augend |  |
|  | ADD | AL,addend | ；低位 |
|  | DAA |  |  |
|  | XCHG | AL, AH |  |
|  | MOV | AL, augend + 1 |  |
|  | ADC | AL, addend + 1 | ；高位 |
|  | DAA |  |  |
|  | XCHG | AL, AH |  |
|  | MOV | BL, 0 |  |
|  | ADC | BL, 0 | ；进位 |
|  | CLD |  |  |
|  | LEA | DI, BUFFER | ；存放显示结果 |
|  | CALL | BltoB2 | ；低位 |
|  | MOV | AL, AH |  |
|  | CALL | BltoB2 | ；高位 |
|  | MOV | AL, BL |  |
|  | STOSB |  |  |
|  | MOV | AL, 10H | ；最高三位消隐 |
|  | STOSB |  |  |
|  | STOSB |  |  |
|  | STOSB |  |  |
|  | LEA | SI,BUFFER+4 |  |
|  | MOV | CX, 4 |  |
|  | CALL | BlackDisplay | ；将高位0消隐 |
|  | LEA | SI,BUFFER |  |
|  | CALL | Display8 |  |
|  | JMP | START1 |  |
| ；将一个字节压缩BCD码转换成二个字节非压缩BCD码 | | |  |

|  |  |  |
| --- | --- | --- |
| BltoB2 | PROC | NEAR |
|  | PUSH | AX |
|  | AND | AL, OFH |
|  | STOSB |  |
|  | POP | AX |
|  | AND | AL, OFOH |
|  | ROR | AL, 4 |
|  | STOSB |  |
|  | RET |  |
| BltoB2 | ENDP |  |
| BlackDisplay | PROC | NEAR |
|  | STD |  |
|  | MOV | DI, SI |
| BlackDisplayl: | LODSB | ；将高位0消隐 |
|  | CMP | AL, 0 |
|  | JNZ | Exit |
|  | MOV | AL, 10H |
|  | STOSB |  |
|  | LOOP | BlackDisplayl |
| Exit: | CLD |  |
|  | RET |  |
| BlackDisplay | ENDP |  |
|  | END | START |

实验三双字节**BCD**码什进制数）减法

一、 实验目的

熟悉8086汇编指令，学会使用星研集成环境软件，掌握汇编语言的设计和调试方法。能看 懂程序流程框图，能自己设计程序，熟悉BCD码、补码，了解如何调用系统提供的子程序。

二、 实验内容

从键盘上输入4位被减数、减数，实现双字节BCD码（四位数）的减法，结果显示在数码管 上；进一步熟悉使用断点、单步进入、单步、运行到光标处、修改PC指针、全速运行等各种调 试手段；熟悉查看特殊功能寄存器、CS段、DS段存贮器的各种方法。

三、程序框图



双字节十进制减法程序框图

四、实验步骤

**1**、连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS、AO、Al |  | A3 区：CS1、AO、Al |
| D3 区：PCO、PCI |  | F5 区：KL1、KL2 |
| D3 区：JP20、B、C |  | F5 区：A、B、C |

2、 在F5区的键盘上输入4位被减数、减数

3、 结果显示在F5区的数码管上

五、程序清单

.MODEL TINY

InitKeyDisplay:NEAR, Display8:NEAR, GetBCDKey:NEAR

EXTRN

EXTRN

Fl:BYTE

.STACK

100

.DATA

|  |  |  |  |
| --- | --- | --- | --- |
| BUFFER | DB | 8 DUP (?) | |
| minuend | DW | 1 DUP(?) | ;被减数 |
| subtrahend | DW | 1 DUP(?) | ;减数 |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | InitKeyDisplay | ;对键盘、数码管扫描控制器8255初始化 |
|  | MOV | Fl,0 | ；先清除显示，再接收键输入 |
| START1: | LEA | DI, minuend |  |
|  | MOV | CX, 4 | ；按键次数 |
|  | CALL | GetBCDKey | ；得到双字节十进制数（被减数） |
|  | MOV | Fl, 1 | ；接收到第一个键，才清除显示 |
|  | LEA | DI, subtrahend |  |
|  | MOV | CX, 4 | ；按键次数 |
|  | CALL | GetBCDKey | ；得到双字节十进制数（减数） |
|  | MOV | AX, minuend |  |
|  | MOV | BX, subtrahend |  |
|  | MOV | DL, 10H |  |
|  | CMP | AX, BX |  |
|  | JNB | START2 |  |
|  | XCHG | AX, BX |  |
|  | MOV | DL, 11H | ；负数 |
| START2: | SUB | AL, BL | ；低位 |
|  | DAS |  |  |
|  | XCHG | AL, AH |  |
|  | SBB | AL, BH | ；高位 |
|  | DAS |  |  |
|  | XCHG | AL, AH |  |
|  | CLD |  |  |
|  | LEA | DI, BUFFER | ；存放显示结果 |
|  | CALL | BltoB2 | ；低位 |
|  | MOV | AL, AH |  |
|  | CALL | BltoB2 | ；高位 |
|  | MOV | AL, 10H | ；最高三位消隐 |
|  | STOSB |  |  |
|  | STOSB |  |  |
|  | STOSB |  |  |
|  | STOSB |  |  |
|  | LEA | SI,BUFFER+3 |  |
|  | MOV | CX, 3 |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | CALL | BlackDisplay | ；将高位0消隐 |
|  | LEA | SI, BUFFER |  |
|  | CALL | Display8 |  |
|  | JMP | START1 |  |
| ；将一个字节压缩BCD码转换成二个字节非压缩BCD码 | | |  |
| BltoB2 | PROC | NEAR |  |
|  | PUSH | AX |  |
|  | AND | AL, OFH |  |
|  | STOSB |  |  |
|  | POP | AX |  |
|  | AND | AL, OFOH |  |
|  | ROR | AL, 4 |  |
|  | STOSB |  |  |
|  | RET |  |  |
| BltoB2 | ENDP |  |  |
| BlackDisplay | PROC | NEAR | ;将高位0消隐 |
| BlackDisplayl: | MOV | AL, [SI] |  |
|  | CMP | AL, 0 |  |
|  | JNZ | Exit |  |
|  | MOV | AL, 10H |  |
|  | MOV | [SI], AL |  |
|  | DEC | SI |  |
|  | LOOP | BlackDisplayl |  |
| Exit: | CLD |  |  |
|  | MOV | [SI+1],DL |  |
|  | RET |  |  |
| BlackDisplay | ENDP |  |  |
|  | END | START |  |

六、思考题

带符号的十进制数加法程序如何编写？

实验四四字节十六进制数转十进制数

一、实验目的

进一步熟悉8086汇编指令，了解十六进制数转十进制数的方法。

二、实验内容

从键盘上输入8位十六进制数，实现四字节十六进制数转8位十进制数，并在数码管上显 示。注意输入数据必须在00000000H-05F00000H范围，否则，结果超出8位十进制数，无法正 确显示。

**三、程序框图**



主程序 四字节十六转十进制子程序

四、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS、AO、Al |  | A3 区：CS1、AO、Al |
| D3 区：PCO、PCI |  | F5 区：KL1、KL2 |
| D3 区：JP20、 B、 C |  | F5 区：A、B、C |

2、 在F5区的键盘上输入8位十六进制数

3、 结果显示在F5区的数码管上

五、程序清单

.MODEL TINY

EXTRN InitKeyDisplay:NEAR, Display8:NEAR, GetKey:NEAR

EXTRN Fl:BYTE

.STACK 100

.DATA

BUFFER DB 8 DUP(?)

.CODE

|  |  |  |  |
| --- | --- | --- | --- |
| START: | MOV | AX, @DATA | |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP  CALL | InitKeyDisplay | ;对键盘、数码管扫描控制器8255初始化 |
|  | MOV | Fl,0 | ；先清除显示，再接收键输入 |
| START1: | LEA | DI, BUFFER |  |
|  | MOV | CX, 8 | ；按键次数 |
|  | CALL | GetKey | ；得到4字节十六进制数 |
|  | MOV | Fl,l | ；接收到第一个键，才清除显示 |
|  | MOV | SI, WORD PTR BUFFER | |
|  | MOV | DI,WORD PTR BUFFER + 2 | |
|  | CALL | B4toD4 | ；转换成十进制数 |
|  | LEA | DI,BUFFER | ；存放显示结果 |
|  | CALL | BltoB2 | ；低位 |
|  | MOV | AL, AH |  |
|  | CALL | BltoB2 |  |
|  | MOV | AL, BL |  |
|  | CALL | BltoB2 |  |
|  | MOV | AL, BH |  |
|  | CALL | BltoB2 |  |
|  | LEA | SI,BUFFER+7 |  |
|  | MOV | CX, 7 |  |
|  | CALL | BlackDisplay | ；将高位0消隐 |
|  | LEA | SI,BUFFER |  |
|  | CALL | Display8 |  |
|  | JMP | START1 |  |
| ；将一个字节压缩BCD码转换成二个字节非压缩BCD码 | | |  |
| BltoB2 | PROC | NEAR |  |
|  | PUSH | AX |  |
|  | AND | AL,OFH |  |
|  | STOSB  POP | AX |  |
|  | AND | AL, OFOH |  |
|  | ROR | AL, 4 |  |
| BltoB2  BlackDisplay | STOSB  RET  ENDP  PROC | NEAR |  |
|  | STD  MOV | DI, SI |  |
| BlackDisplayl: | LODSB |  | ；将高位0消隐 |
|  | CMP | AL, 0 |  |

|  |  |  |
| --- | --- | --- |
|  | JNZ  MOV  STOSB  LOOP | Exit  AL, 10H  BlackDisplayl |
| Exit: | CLD |  |
|  | RET |  |
| BlackDisplay | ENDP |  |
| ;四字节十六进制数转十进制数: | | DISI为十六进制，BXAX为压缩BCD码 |
| B4toD4 | PROC | NEAR |
|  | XOR | AX, AX |
|  | XOR | BX, BX |
|  | MOV | CX, 32 |
| B4toD4\_l: | RCL | SI, 1 |
|  | RCL | DI, 1 |
|  | ADC | AL, AL |
|  | DAA |  |
|  | XCHG | AL, AH |
|  | ADC | AL, AL |
|  | DAA |  |
|  | XCHG | AL, BL |
|  | ADC | AL, AL |
|  | DAA |  |
|  | XCHG | AL, BH |
|  | ADC | AL, AL |
|  | DAA |  |
|  | XCHG | AL, BH |
|  | XCHG | AL, BL |
|  | XCHG | AL, AH |
|  | LOOP | B4toD4\_l |
|  | RET |  |
| B4toD4 | ENDP |  |
|  | END | START |

六、思考题

如果不考虑在数码管上显示,

不限制数据范围，程序应如何编写。

实验五散转

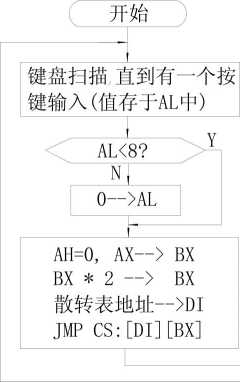
一、实验目的

熟悉使用8086指令，掌握汇编语言的设计和调试方法；理解并能运用散转指令。

**二、实验内容**

编写程序，根据接收到的键值，作不同的处理。

**三、程序框图**



| 0号键： | 显示8个1 |
| --- | --- |
| 1号键： | 显示8个2 |
| 2号键： | 显示8个3 |
| 3号键： | 显示8个4 |
| 4号键： | 显示8个5 |
| 5号键： | 显示8个6 |
| 6号键： | 显示8个7 |
| 7号键： | 显示8个8 |

散转程序流程框图

四、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS、AO、A1 |  | A3 区：CS1、A0、A1 |
| D3 区：PCO、PC1 |  | F5 区：KL1、KL2 |
| D3 区：JP20、 B、 C |  | F5 区：A、B、C |

2、 在F5区的键盘上输入1位数

3、 使用各种手段调试程序

3、结果显示在F5区的数码管上

五、程序清单

;调用GetKeyB返回键值，根据键值执行相应的程序

.MODEL TINY

EXTRN InitKeyDisplay:NEAR, Display&NEAR, GetKeyB:NEAR

|  |  |  |
| --- | --- | --- |
|  | .STACK  .DATA | 100 |
| BUFFER | DB | 8 DUP(?) |
|  | .CODE |  |
| START: | MOV | AX, @DATA |
|  | MOV | DS, AX |
|  | MOV | ES, AX |

|  |  |  |
| --- | --- | --- |
|  | NOP |  |
|  | CALL | InitKeyDisplay ;对键盘、数码管扫描控制器8255初始化 |
| START1: | CALL | GetKeyB ；键值存放在AL中 |
|  | CMP | AL, 8 |
|  | JB | START2 |
|  | XOR | AL, AL ；大于7,作0处理 |
| START2: | XOR | AH, AH |
|  | MOV | BX, AX |
|  | SHL | BX, 1 |
|  | LEA | DI, Table\_l |
|  | JMP | CS:[DI][BX] |
| Table\_l: | DW | KeyO, Keyl, Key2, Key3, Key4, Key5, Key6, Key7 |
| KeyO: | MOV | AL, 1 |
|  | JMP | Key |
| Keyl: | MOV | AL, 2 |
|  | JMP | Key |
| Key2: | MOV | AL, 3 |
|  | JMP | Key |
| Key3: | MOV | AL, 4 |
|  | JMP | Key |
| Key4: | MOV AL, 5 |  |
|  | JMP | Key |
| Key5: | MOV | AL, 6 |
|  | JMP | Key |
| Key6: | MOV | AL, 7 |
|  | JMP | Key |
| Key7: | MOV | AL, 8 |
|  | JMP | Key |
| Key: | MOV | CX, 8 |
|  | LEA | DI, BUFFER |
|  | REP | STOSB |
|  | LEA | SI, BUFFER |
|  | CALL | Display8 |
|  | JMP | START1 |
|  | END | START |

六、思考题

程序中为什么要把输入的值作乘以2处理?

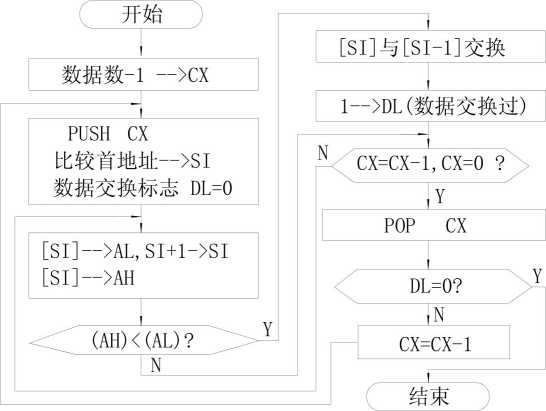
实验六冒泡排序

一、 实验目的

熟悉使用8086指令，掌握汇编语言的设计和调试方法，了解如何使用高效方法对数据排序。

二、 实验内容

编写并调试一个排序程序，要求使用冒泡法将一组数据从小到大重新排列。



三、

程序框图

冒泡排序程序框图

四、

| .MODEL | | TINY |
| --- | --- | --- |
|  | .STACK | 100 |
|  | .DATA |  |
| TAB\_1: | DB OH, 5H, 6H, 3H, 8H, 92H, 04H; | |
|  | .CODE |  |
| START: | MOV | AX, @DATA |
|  | MOV | DS, AX |
|  | NOP |  |
|  | MOV | CX, 16 - 1 |
| STAR2: | PUSH | ex |
|  | LEA | SI, TAB\_1 |
|  | MOV | DL,0 |
| STAR3: | LODSB |  |
|  | MOV | AH, [SI] |
|  | CMP | AH, AL |

**程序清单**

五、

:,57H, 46H, 01H, OFFH, 0A0H, 45H, 99H, 55H, 66H

；存放比较次数=数据个数-1

；0-＞交换过数据标志

实验步骤

使用断点方式调试程序，检查数据段中数据是否按从小到大的顺序排列。

|  |  |  |  |
| --- | --- | --- | --- |
|  | JNB  MOV  MOV  MOV | STAR5  [SI], AL  [SI - 1],AH  DL, 1 | ;1-〉交换过数据标志 |
| STAR5: | LOOP | STAR3 |  |
|  | POP | CX |  |
|  | CMP | DL, 0 |  |
|  | JZ | Exit |  |
|  | LOOP | STAR2 |  |
| Exit: | JMP | $ |  |
|  | END | START |  |

六、思考题

你还知道哪些排序方法？另外编写一个排序子程序。（数据结构方面的教材上有十几种排序 方法）。

实验七二分查找法

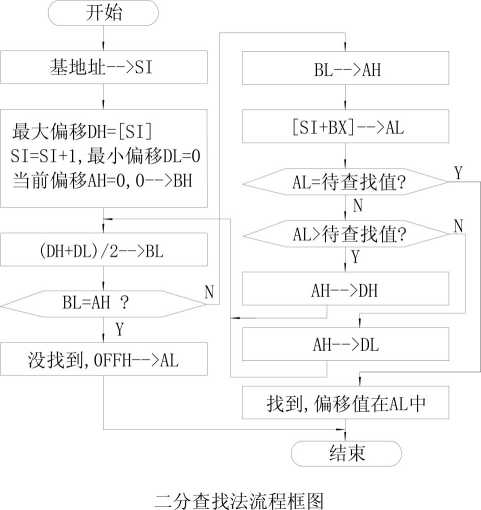
一、 实验目的

熟悉使用8086指令，掌握汇编语言的设计和调试方法。

二、 实验内容

编写并调试一个二分查找法程序，要求在一组从小到大排列的数据中查找一个数。

**三、程序框图**



四、实验步骤

在Search\_Data中定义一个需要查找的数据，运行程序，是否能找到指定的数据，结果是 否正确。

|  |  |  |  |
| --- | --- | --- | --- |
| **五、程序清单** | .MODEL | TINY | |
| SearchData | EQU | 60 | ;需要查找的数据 |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| TAB\_1: | DB | 32 | ；共有32个数 |
|  | DB | 01, 03, 05, 06, 07, 10, 11, 12, 13, 14, 15, 16, 20, 25, 26, 29 | |
|  | DB | 37, 3& 39, 42, 43, 44, 45, 50, 52, 53, 56, 59, 60, 62, 66, 68 | |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | NOP |  |  |
|  | LEA | SI, TAB\_1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| LODSB | | | |
|  | MOV | DH, AL | ;最大位置 |
|  | MOV | DL, 0 | ;最小位置 |
|  | MOV | AH, 0 | ;当前位置 |
|  | XOR | BH, BH |  |
| STAR1: | MOV | BL, DH |  |
|  | ADD | BL, DL |  |
|  | CLC |  |  |
|  | SHR | BL, 1 |  |
|  | CMP | BL, AH |  |
|  | JNE | STAR2 |  |
|  | MOV | AL, OFFH | ;没有找到 |
|  | JMP | NoFind |  |
| STAR2: | MOV | AH, BL |  |
|  | MOV | AL, [SI + BX] |  |
|  | CMP | AL, Search\_Data |  |
|  | JNZ | STAR3 |  |
|  | MOV | AL, AH |  |
|  | JMP | Find |  |
| STAR3: | JB | STAR4 |  |
|  | MOV | DH, AH |  |
|  | JMP | STAR1 |  |
| STAR4: | MOV | DL, AH |  |
|  | JMP | STAR1 |  |
| Find: | JMP | $ |  |
| NoFind: | JMP | $ |  |
|  | END | START |  |

六、思考题

1、程序只能实现256字节范围内的查找，请读者考虑，若查找范围大于256字节，程序该 怎么编写？

S基础硬件实验

基础硬件实验

本章和下一章将结合实验仪的所有单元电路（包括标准配置和可选各种模块）向读者逐一 介绍各个实验，由浅入深，从最基础的实验开始，直到读者学会使用各种外围电路，开发有」 定深度的项目。硬件实验分为基础实验和综合实验两部分，本章主要介绍常用外围电路；综合 实验介绍- ■些新颖外围电路，将各个单元电路灵巧组合、深入挖掘，生成-些具有实际意义的 工程。读者也可以根据自己的理解、需要，将各个单元电路自行组合而成具有实际意义的复杂 控制电路，在设计电路板前，在实验仪上作一认证。可见，SUNES86PCIU+适合于不同层次的学 者、工程师以及电子爱好者进行学习、实践，SUNES86PCIU+提供了完整的源代码，可以直接复 制到用户系统中，为你节省大量宝贵时间。

实验一简单I/O （16位）实验

一、实验目的与要求

1、 了解外设的扩展方法，掌握外设的读写时序。

2、 了解74HC273、74HC244的功能，掌握它们的使用方法。

3、 掌握CPU对16位外设的访问方法

4、 认真预习本节实验内容，尝试自行编写程序，填写实验报告。

二、 实验设备

SUN系列实验仪一套、PC机一台

三、 实验内容

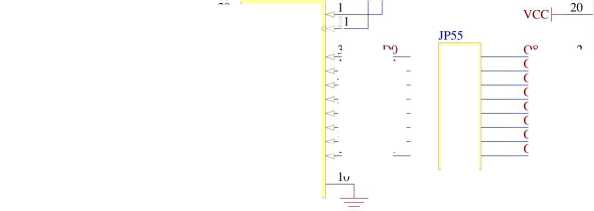
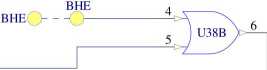
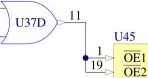
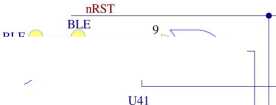
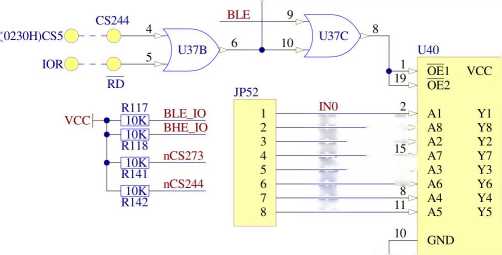
1、 说明：二片74HC244组成16位的只读外设，二片74HC273组成16位的只写外设，它们 都可以按字节或字方式操作。实验仪具有16位数据总线DO. .D15、BLE （低电平有效，选中挂 在低8位数据总线上外设）、BHE （低电平有效，选中挂在高8位数据总线上外设）；BLE、BHE 同时有效，对外设字方式读写，BLE或BHE有效，对外设字节方式读写。

二片74HC273的输出端与F4区的16个发光二极管相连；低位74HC244的输入端与F4区的 8个拨动开关相连，8个拨动开关循环左移一位后与高位74HC244的输入端相连。

2、 编写程序：将B4区的二片74HC244中数据读出、写入二片74HC273中；然后逐一点亮 挂在74HC273上的16个发光二级管；循环执行

3、 连接线路验证功能，熟悉它的使用方法。

四、 实验原理图



BHE

SN74HC244N

| IN] | I.. |
| --- | --- |
| IN2 | £ . |
| IN3 — |  |
| IN4 | ZZD |
| IN5 — | *耳* |
| IN6 |  |
| IN7 |  |

| 18 | DO |
| --- | --- |
| 3 | D1 |
| 16 | D2 |
| 5 | D3 |
| 14 | D4 |
| 7 | D5 |
| 12 | D6 |
| 9 | D7 |

13

VCC \_—

vcc|\_—

| IN2 | 17： |
| --- | --- |
| IN3 | *4* |
| IN4 | *15* |
| IN5 | *6* |
| IN6 | 13. |
| IN7 | 8 |
| IN0 | 11 |

IN1

—SN74HC244N

VCC

Al Y1

A8 Y8

A2 Y2

A7 Y7

A3 Y3

A6 Y6

A4 Y4

A5 Y5

VCC

| 3 | D9 |
| --- | --- |
| 16 | DIO |
| 5 | Dll |
| 14 | D12 |
| 7 | D13 |
| 12 | D14 |
| 9 | D15 |

D8

1

TT

GND

SN74HC273N

JP51

00 2

11

DO

10

2

亏

16

卫

TT

08

09~ oTo- olT 012" 013~ oTT 015-

WR

1

2

3

4

5

6

7

8

RSTCLKD0D7D1D6D2D5D3D4

C 07162534

V QQQQQQQQ

| 01 | 19 |
| --- | --- |
| 02 | 5 |
| 03 | 16 |
| 04 | 6 |
| 05 | 15 |
| 06 | 9 |
| 07 | 12 |

| 18 | D1 |
| --- | --- |
| *4* | D2 |
| 17 | D3 |
| *7* | D4 |
| 14 | D5 |
| 8 | D6 |
| *：13* | D7 |

3

| VCC | RST CLK |
| --- | --- |
| 00 | DO |
| Q7 | D7 |
| Q1 | D1 |
| Q6 | D6 |
| 02 | D2 |
| Q5 | D5 |
| Q3 | D3 |
| Q4 | D4 |
|  | GND |

SN74HC273N

| -4 | D10 |
| --- | --- |
| 17 | D11 |
| ；7 | D12 |
| 14 | D13 |
| 8 | D14 |
| ：13 | D15 |

3

；]8 D9

D8

**五、实验步骤**

1、连线说明:

|  |  |  |
| --- | --- | --- |
| B4(I/0)区：CS273、CS244 |  | A3 区：CS5、CS5 |
| B4(I/0)区：BLE、BHE |  | A3 区：BLE、BHE |
| B4(I/0)区：RD、WR |  | A3 区：IOR、IOW |
| B4(I/0)区：JP51、JP55 |  | F4 区:JP18、JP19（发光管） |
| B4(I/0)区：JP52 |  | F4区：JP27 （开关） |
| B4 区：JP57(D0. .D7) |  | A3 区：JP42 (DO.. D7) |
| B4 区：JP56 (D8.. D15) |  | A3 区：JP40(D&.D15) |

2、观察实验结果，拨动开关状态是否与点亮的发光二极管…致，是否循环点亮16个发光 二级管。

**六、演示程序**

|  |  |  |  |
| --- | --- | --- | --- |
|  | .MODEL | TINY | |
| 10244 | EQU | 0230H | ；244 （16位）片选 |
| 10273 | EQU | 0230H | ；273（16位）片选 |
|  | .STACK | 100 |  |
|  | .CODE |  |  |
| START: | MOV | DX, 10244 |  |
|  | IN | AX, DX | ；读取开关数据（16位，KO K7 K6 K5 K4 K3 |
|  |  |  | K2 KI K7 K6 K5 K4 K3 K2 KI K0) |
|  | MOV | DX, 10273 |  |
|  | OUT | DX, AX |  |
|  | CALL | Delay |  |
|  | CALL | Delay |  |
|  | CALL | Delay |  |
|  | CALL | Delay |  |
|  | CALL | Delay |  |
|  | CALL | Delay |  |
|  | MOV | DX, 10273 |  |
|  | MOV | AX, OFFFEH |  |
| START1: | OUT | DX, AX |  |
|  | CALL | Delay |  |
|  | TEST | AX, 8000H |  |
|  | JZ | START |  |
|  | ROL | AX, 1 |  |
|  | JMP | START1 |  |
| Delay | PROC | NEAR | ；延时 |
| Delayl: | XOR | CX, CX |  |
|  | LOOP | $ |  |
|  | RET |  |  |
| Delay | ENDP |  |  |
|  | END | START |  |
| **七、实验扩展及思考** | |  |  |
| 1、请按照字、 | 字节方式画出读（74HC244）写 | | （74HC273）的时序。 |

2、以上程序中，使用16位方式读写外设，请按照8位方式，重编程序。

实验二 静态存贮器（16位）读写实验

一、 实验目的与要求

1、 了解静态存贮器的特性、扩展方法，掌握存贮器的读写时序。

2、 掌握CPU对16位静态存贮器的访问方法

3、 认真预习本节实验内容，尝试自行编写程序，填写实验报告。

二、 实验设备

SUN系列实验仪一套、PC机一台

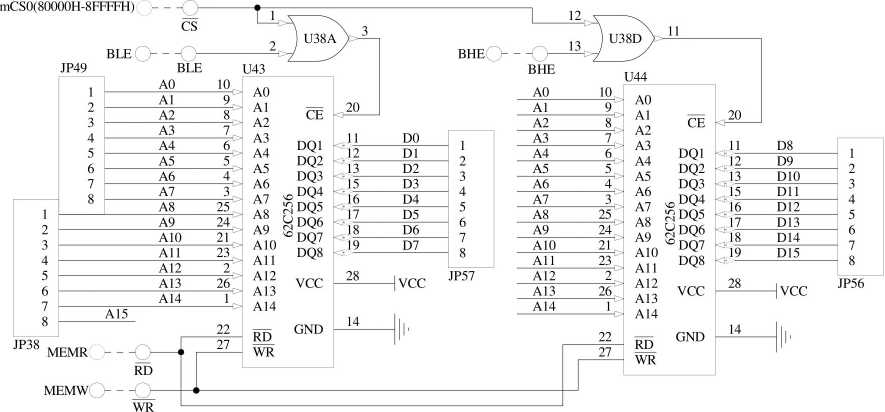
三、 实验内容

1、 说明：实验仪选用常用的静态存贮器芯片62256 （32KX8位），二片组成32KX16位， 共64K字节。实验仪具有16位数据总线DO. .D15、20位地址线AO. .A19、BLE （低电平有效， 选择低8位存贮器芯片）、BHE （低电平有效，选择高8位存贮器芯片）；BLE、BHE同时有效， 对存贮器字方式读写，BLE或BHE有效，对存贮器字节方式读写。扩展16位存贮器时，不使用 地址线AOo

2、 编写程序：将B4区的静态存贮器3000H-30FFH单元中数据复制到6000H-60FFH的单元 中，并校验，检测写入的数据是否正确。

3、 连接线路验证功能，熟悉它的使用方法。

**四、实验原理图**



**五、实验步骤**

1、连线说明:

|  |  |  |
| --- | --- | --- |
| B4(RAM)区：CS、BLE、BHE |  | A3 区：mCSO, BLE、 BHE |
| B4(RAM)区：RD、WR |  | A3 区：MEMR、 MEMW |
| B4(RAM)区：JP49 (AO. . A7) |  | A3 区:JP29(A1. ,A8) |
| B4(RAM)区：JP38 (A8. . A15) |  | A3 区:JP33(A9. .A16) |
| B4(RAM)区：JP57 (DO. . D7) |  | A3 区:JP42 (DO.. D7) |
| B4(RAM)区：JP56 (D &.D15) |  | A3 区:JP40 (D8. .D15) |

2、通过星研软件的存贮器窗、寄存器窗等，观察运行结果。

**六、演示程序**

；数据块移动（16位存贮器实验）

|  |  |  |
| --- | --- | --- |
| .MODEL | | TINY |
|  | .STACK | 100 |
|  | .CODE |  |
| START: | MOV | AX, 8000H |
|  | MOV | DS, AX |
|  | MOV | ES, AX |
|  | NOP |  |
|  | MOV | CX, 100H |
|  | MOV | SI, 3000H |
|  | MOV | DI, 6000H |
|  | CALL | Move |
|  | MOV | CX, 100H |
|  | MOV | SI, 3000H |
|  | MOV | DI, 6000H |
|  | CLD |  |
|  | REPE | CMPSB |
|  | JNE | ERROR |
| TRUE: | JMP | $ |
| ERROR: | JMP | $ |
| Move | PROC | NEAR |
|  | CLD |  |
|  | CMP | SI, DI |
|  | JZ | Return |
|  | JNB | Movel |
|  | ADD | SI, CX |
|  | DEC | SI |
|  | ADD | DI,CX |
|  | DEC | DI |
|  | STD |  |
| Movel: | REP | MOVSB |
| Return: | RET |  |
| Move | ENDP |  |
| END | START |  |

;存贮器扩展空间段地址

七、实验扩展及思考

1、 子程序Move中为什么比较SI、DI?

源数据块与目标范围有可能部分重叠，需要考虑从第一个字节开始复制（顺序复制），还是 从最后一个字节开始复制（倒序复制）。

2、 本实验与软件实验…的异同点。

实验三8255控制交通灯实验

一、实验目的与要求

1、 了解8255芯片的工作原理，熟悉其初始化编程方法以及输入、输出程序设计技巧。学 会使用8255并行接口芯片实现各种控制功能，如本实验（控制交通灯）等。

2、 熟悉8255内部结构和与8086的接口逻辑，熟悉8255芯片的3种工作方式以及控制字 格式。

3、 认真预习本节实验内容，尝试自行编写程序，填写实验报告。

二、实验设备

SUN系列实验仪一套、PC机一台

三、实验内容

1、 编写程序：使用8255的PA0.. 2、PA4.. 6控制LED指示灯，实现交通灯功能。

2、 连接线路验证8255的功能，熟悉它的使用方法。

四、实验原理图



CSl(0270H)

AO

Al

A0 aT

| DI | 33 |
| --- | --- |
| D2 | 32 |
| D3 | 31 |
| D4 | 30 |
| D5 | 29 |
| D6 | 28 |
| D7 | 27 |
| RD | 5r |
| WR | 31 |
| 9 | |
|  | |
| RST |  |

DO 34

| DO | PAO |
| --- | --- |
| DI | PAI |
| D2 | PA2 |
| D3 | PA3 |
| D4 | PA4 |
| D5 | PA5 |
| D6 | PA6 |
| D7 | PA7 |
| RD | PBO |
| WR | PB1 |
| AO | PB2 |
| Al | PB3 |
| RESET | PB4 |
| CS | PB5 |
|  | PB6 |
|  | PB7 |
|  | PCO |
|  | PCI |
|  | PC2 |
|  | PC3 |
|  | PC4 |
|  | PC5 |
|  | PC6 |
|  | PC7 |

U32

8255

<

-■

-

| 3 | 7 |
| --- | --- |
| 2 | 6 |
| 1 | 5 |
| 40 | 8 |
| 39 | 7 |
| 38 | 6 |
| 37 | 5 |
| 18 |  |
| 19 |  |
| 20 |  |
| 21 |  |
| *23-* |  |
| 24 |  |
| 25\_ |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 13 |  |
| 12 |  |
| 11 |  |
| 10 |  |

8

-----■

RN5 1K

-

*-:*

-■

-■

*:*

-■

-

-

-

<-

-

4

RN4 IK

丄

2

T

*±*

T

2

3

*~4*

DS12

DS13

DS14

DS15

DS16

DS17

DS18

DS19

fvcc

五、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS、AO、Al |  | A3 区：CS1、A0、Al |
| D3 区：JP23 (PA 口) |  | F4 区:JP18 |

2、观察实验结果，是否能看到模拟的交通灯控制过程。

**六、演示程序**

COM\_ADD

.MODEL

EQU

TINY

0273H

|  |  |  |  |
| --- | --- | --- | --- |
| PA\_ADD | EQU | 0270H | |
| PB\_ADD | EQU | 0271H |  |
| PC\_ADD | EQU | 0272H |  |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| LEDData | DB | 10111110B | ;东西绿灯，南北红灯 |
|  | DB | 10111111B | ;东西绿灯闪烁，南北红灯 |
|  | DB | 10111101B | ;东西黄灯亮，南北红灯 |
|  | DB | 11101011B | ;东西红灯，南北绿灯 |
|  | DB | 11111011B | ;东西红灯，南北绿灯闪烁 |
|  | DB | 11011011B | ;东西红灯，南北黄灯亮 |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | NOP |  |  |
|  | MOV | DX, COM\_ADD |  |
|  | MOV | AL, 80H | ;PA、PB、PC为基本输出模式 |
|  | OUT | DX, AL |  |
|  | MOV | DX, PA\_ADD | ；灯全熄灭 |
|  | MOV | AL, OFFH |  |
|  | OUT | DX, AL |  |
|  | LEA | BX, LED Data |  |
| START1: | MOV | AL, 0 |  |
|  | XLAT |  |  |
|  | OUT | DX, AL | ;东西绿灯，南北红灯 |
|  | CALL | DL5S |  |
|  | MOV | CX, 6 |  |
| START2: | MOV AL, 1 |  |  |
|  | XLAT |  |  |
|  | OUT | DX, AL | ；东西绿灯闪烁，南北红灯 |
|  | CALL | DL500ms |  |
|  | MOV | AL, 0 |  |
|  | XLAT |  |  |
|  | OUT | DX, AL |  |
|  | CALL | DL500ms |  |
|  | LOOP | START2 |  |
|  | MOV | AL, 2 | ；东西黄灯亮，南北红灯 |
|  | XLAT |  |  |
|  | OUT | DX, AL |  |
|  | CALL | DL3S |  |
|  | MOV | AL, 3 | ;东西红灯，南北绿灯 |
|  | XLAT |  |  |
|  | OUT | DX, AL |  |
|  | CALL | DL5S |  |

|  |  |  |
| --- | --- | --- |
|  | MOV | CX, 6 |
| START3: | MOV | AL, 4 |
|  | XLAT |  |
|  | OUT | DX, AL |
|  | CALL | DL500ms |
|  | MOV | AL, 3 |
|  | XLAT |  |
|  | OUT | DX, AL |
|  | CALL | DL500ms |
|  | LOOP | START3 |
|  | MOV | AL, 5 |
|  | XLAT |  |
|  | OUT | DX, AL |
|  | CALL | DL3S |
|  | JMP | START1 |
| DL500ms | PROC | NEAR |
|  | PUSH | ex |
|  | MOV | CX, 60000 |
| DL500msl: | LOOP | DL500msl |
|  | POP | ex |
|  | RET |  |
| DL500ms | ENDP |  |
| DL3S | PROC | NEAR |
|  | PUSH | ex |
|  | MOV | CX, 6 |
| DL3S1: | CALL | DL500ms |
|  | LOOP | DL3S1 |
|  | POP | ex |
|  | RET |  |
|  | ENDP |  |
| DL5S | PROC | NEAR |
|  | PUSH | ex |
|  | MOV | CX, 10 |
| DL5S1: | CALL | DL500ms |
|  | LOOP | DL5S1 |
|  | POP | ex |
|  | RET |  |
|  | ENDP |  |
|  | END | START |

;东西红灯，南北绿灯闪烁

;东西红灯，南北黄灯亮

七、实验扩展及思考

1、如何对8255的PC 口进行位操作？

实验四74HC138译码器实验

一、 实验目的与要求

1、 掌握74HC138译码器的工作原理，熟悉74HC138译码器的具体运用连接方法，了解74HC138 是如何译码的。

2、 认真预习本节实验内容，尝试自行编写程序，填写实验报告

二、 实验设备

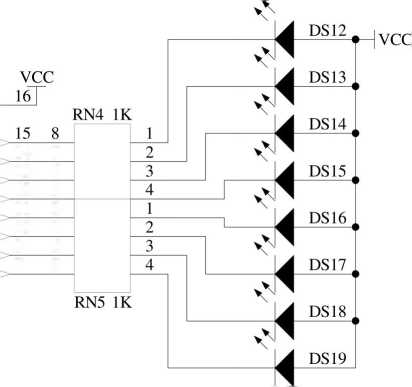
SUN系列实验仪一套、PC机一台

三、 实验内容

1、 编写程序：使用82C55的PCO、PCI、PC2控制74HC138的数据输入端，通过译码产生8 选1个选通信号，轮流点亮8个LED指示灯。

2、 运行程序，验证译码的正确性。

四、实验原理图



(8255)PC0Q 0

(8255)PC1

C

(8255)PC2

**(ci)vcco**

G1 r

(Cl)GNDQ CP

**c**

(Cl)GND O - <

U49

I

I

B

C

OE1

OE2A

OE2B

G2A

Q

G2B I

SN74HC138N

C

C

V

01234567

-Y-Y-Y-Y-Y-Y-Y-Y

| 14 | 7 |
| --- | --- |
| 13 | 6 |
| 12 | 5 |
| 11 | 8 |
| 10 | 7 |
| 9 | 6 |
| 7 | 5 |

**五、实验步骤**

|  |  |  |
| --- | --- | --- |
| 1、连线说明： | | |
| F7 区：A、B、C — | —D3 区：PCO、 | PCI、 PC2 |
| F7 区：Gl、G2A、G2B — | —C1 区：VCC、 | GND、 GND |
| F7 区：JP63 — | —F4 区：JP18 | （LED指示灯） |
| D3 区：CS、AO、A1 — | —A3 区：CS1、 | AO、A1 |

2、调试程序，查看运行结果是否正确。

**六、演示程序**

|  |  |  |  |
| --- | --- | --- | --- |
|  | .MODEL | TINY | |
| Con\_8255 | EQU | 0273H | ;8255控制口 |
| PC\_8255 | EQU | 0272H | ;8255 PC口 |
|  | ・STACK | 100 |  |

.CODE

|  |  |  |  |
| --- | --- | --- | --- |
| START: | MOV | DX, Con\_8255 | |
|  | MOV | AL, 80H |  |
|  | OUT | DX, AL | ；8255初始化,PC 口作输出用 |
|  | MOV | DX, PC\_8255 |  |
|  | MOV | AL, 0 |  |
| START1: | OUT | DX, AL |  |
|  | CALL | Delay |  |
|  | INC | AL |  |
|  | JMP | START1 |  |
| Delay | PROC | NEAR | ；延时 |
| Delayl: | XOR | CX, CX |  |
|  | LOOP | $ |  |
|  | RET |  |  |
| Delay | ENDP |  |  |
|  | END | START |  |

七、实验扩展及思考

请读者考虑-下，应如何处理?

在应用系统中，74HC138通常用来产生片选信号,

实验五8253方波实验

一、实验目的与要求

了解8253的内部结构、工作原理；了解8253与8086的接口逻辑；熟悉8253的控制寄存 器和初始化编程方法，熟悉8253的6种工作模式。

二、 实验设备

SUN系列实验仪一套、PC机一台

三、 实验内容

1、 编写程序：使用8253的计数器0和计数器1实现对输入时钟频率的两级分频，得到一 个周期为1秒的方波，用此方波控制蜂鸣器，发出报警信号，也可以将输入脚接到逻辑笔上来 检验程序是否正确。

2、 连接线路，验证8253的功能，熟悉它的使用方法。

四、实验原理图



A0

A1

A0

A1

| DO | 8 |
| --- | --- |
| DI | 7 |
| D2 | 6 |
| D3 | 5 |
| D4 | 4 |
| D5 | 3 |
| D6 | 2 |
| D7 | 1 |
| RD | 21。  22] |
| WR | 23<: |
|  | 19.. |
|  | |
|  | |

CS2(0260H)

a-o

cs

| D0 | OUTO |
| --- | --- |
| D1 | GATEO |
| D2 | CLKO < |
| D3 |  |
| D4 |  |
| D5 |  |
| D6 | OUT1 |
| D7 | GATE1 |
|  | CLK1 V |
| CS |  |
| RD |  |
| WR |  |
|  | OUT2 |
| AO | GATE2 |
| Al | CLK2 |

U34

8253

10

*JT*

17

16

18^

13 亘

*15*

Q VCC

2MHz(B2)

CLKO

-O——O CtrI(F8)

OUT1

Qclki

\_ OUT2

OCLK2

五、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| C4 区：CS、A0、Al |  | A3 区：CS2、AO、Al |
| C4 区：CLKO |  | B2 区：2M |
| C4 区：OUTO |  | C4 区：CLK1 |
| C4 区：0UT1 |  | F8区:Ctrl（蜂鸣器） |
| C4 区：GATE |  | Cl区的VCC |

2、测试实验结果：蜂鸣器发出时有时无的声音；用逻辑笔测试蜂鸣器的输入端口，红绿灯 交替点亮。

**六、演示程序**

|  |  |  |
| --- | --- | --- |
|  | .MODEL | TINY |
| COM\_ADDR | EQU | 0263H |
| TO\_ADDR | EQU | 0260H |
| T1\_ADDR | EQU | 0261H |
|  | ・ STACK | 100 |

| START: | MOV | DX, COM\_ADDR |
| --- | --- | --- |
|  | MOV | AL, 35H |
|  | OUT | DX, AL |
|  | MOV | DX, TO\_ADDR |
|  | MOV | AL, OOH |
|  | OUT | DX, AL |
|  | MOV | AL, 10H |
|  | OUT | DX, AL |
|  | MOV | DX, COM\_ADDR |
|  | MOV | AL, 77H |
|  | OUT | DX, AL |
|  | MOV | DX, T1\_ADDR |
|  | MOV | AL, OOH |
|  | OUT | DX, AL |
|  | MOV | AL, 10H |
|  | OUT | DX, AL |
|  | JMP | $ |
|  | END | START |

.CODE

;计数器TO设置在模式2状态,BCD码计数

:CLKO/1OOO

；计数器T1为模式3状态，输出方波,BCD码计数

:CLK1/1OOO

;OUT1输出1S的方波

七、实验扩展及思考

1、 8253还有其它五种工作方式，其它工作模式下，硬件如何设计？程序如何编写？

2、 使用8253,编写…个实时钟程序。

实验六**8259A (1)**中断控制器实验

一、 实验目的与要求

了解8259A的内部结构、工作原理；了解8259A与8086的接口逻辑；掌握对8259A的初始 化编程方法，了解8086是如何响应中断、退出中断的。

复习本节实验内容，可尝试自行编写程序，做好实验准备工作，填写实验报告。

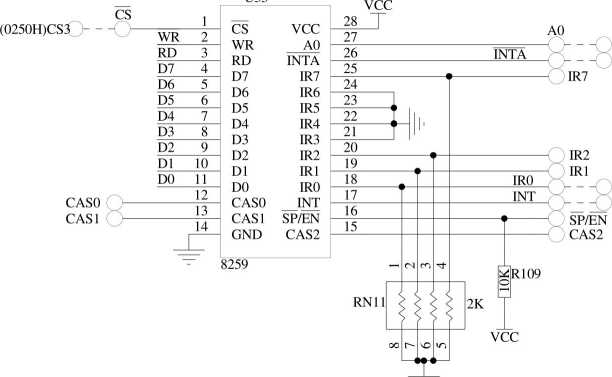
**二、 实验设备**

SUN系列实验仪一套、PC机一台

**三、 实验内容**

1、编制程序：拨动单脉冲开关，“JL”送给8259A的IR0,触发中断，8086计数中断次数, 显示于F5区的数码管上

**四、实验原理图**



TL(B2)

INTR

A0

INTA

**五、实验步骤**

1、连线说明:

|  |  |  |
| --- | --- | --- |
| B3 区：CS、AO |  | A3 区：CS3、A0 |
| B3 区：INT、INTA |  | A3 区:INTR、INTA |
| B3 区：I RO |  | B2区：单脉冲JT |
| D3 区：CS、AO、Al |  | A3 区：CS1、A0、Al |
| D3 区：PCO、PCI |  | F5 区：KL1、KL2 |
| D3 区：JP20、B、C |  | F5 区：A、B、C |

2、 运行程序

3、 上下拨动单脉冲开关，拨动二次，产生一个“JL”，观察结果，数码管上显示的次数与 拨动开关次数是否对应。

**六、演示程序**

.MODEL TINY

EXTRN InitKeyDisplay:NEAR, Display8:NEAR

EQU

I08259\_0

I08259\_l

0250H

0251H

EQU

・ STACK

.DATA

|  |  |  |  |
| --- | --- | --- | --- |
| BUFFER | DB | 8 DUP (?) | |
| Counter | DB | ? |  |
| ReDisplayFlag | DB | 0 |  |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | InitKeyDisplay | ;对键盘、数码管控制器8255初始化 |
|  | CALL | Init8259 |  |
|  | CALL | Wrilntver |  |
|  | MOV | Counter, 0 | ;中断次数 |
|  | MOV | ReDisplayFlag, 1 | ;需要显示 |
|  | STI | ；开中断 |  |
| START1: | LEA | SI, Buffer |  |
|  | CALL | Display8 |  |
|  | CMP | ReDisplayFlag, 0 |  |
|  | JZ | START1 |  |
|  | CALL | LedDisplay |  |
|  | MOV | ReDisplayFlag, 0 |  |
|  | JMP | START1 |  |
| Init8259 | PROC | NEAR |  |
|  | MOV | DX, I08259\_0 |  |
|  | MOV | AL, 13H |  |
|  | OUT | DX, AL |  |
|  | MOV | DX, IO8259\_1 |  |
|  | MOV | AL, 08H |  |
|  | OUT | DX, AL |  |
|  | MOV | AL, 09H |  |
|  | OUT | DX, AL |  |
|  | MOV | AL, OFEH |  |
|  | OUT | DX, AL |  |
|  | RET |  |  |
| Init8259 | ENDP |  |  |
| Wrilntver | PROC | NEAR |  |
|  | PUSH | ES |  |
|  | MOV | AX, 0 |  |
|  | MOV | ES, AX |  |
|  | MOV | DI,20H |  |
|  | LEA | AX, INT\_O |  |
|  | STOSW |  |  |
|  | MOV | AX, CS |  |
|  | STOSW |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | POP  RET | ES | |
| Wrilntver | ENDP |  |  |
| LedDisplay | PROC | NEAR |  |
|  | MOV | AL, Counter |  |
|  | MOV | AH, AL |  |
|  | AND | AL, OFH |  |
|  | MOV | Buffer, AL |  |
|  | AND | AH, OFOH |  |
|  | ROR | AH, 4 |  |
|  | MOV | Buffer + 1, AH |  |
|  | MOV | Buffer + 2, 10H | ;高六位不需要显示 |
|  | MOV | Buffer + 3, 10H |  |
|  | MOV | Buffer + 4, 10H |  |
|  | MOV | Buffer + 5, 10H |  |
|  | MOV | Buffer + 6, 10H |  |
|  | MOV | Buffer + 7, 10H |  |
|  | RET |  |  |
| LedDisplay | ENDP |  |  |
| INT\_0: | PUSH | DX |  |
|  | PUSH | AX |  |
|  | MOV | AL, Counter |  |
|  | ADD | AL, 1 |  |
|  | DAA |  |  |
|  | MOV | Counter, AL |  |
|  | MOV | ReDisplayFlag, 1 |  |
|  | MOV | DX, I08259\_0 |  |
|  | MOV | AL, 20H |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | POP | DX |  |
|  | IRET |  |  |
|  | END | START |  |

七、实验扩展及思考

1、从8259A收到上升沿,

到8086响应中断，试画这个过程的时序图。

实验六**8259A (2)**中断(优先级)实验

一、 实验目的与要求

了解8259A的内部结构、工作原理；了解8259A与8086的接口逻辑；掌握对8259A的初始 化编程方法，了解8086是如何响应中断、退出中断的；体会8259A的中断优先级处理方法。

复习本节实验内容，可尝试自行编写程序，做好实验准备工作，填写实验报告。

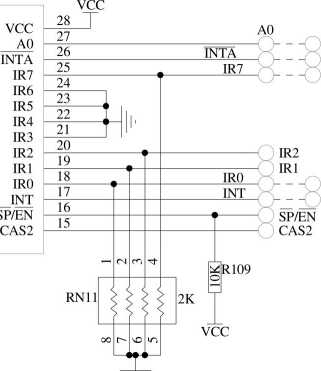
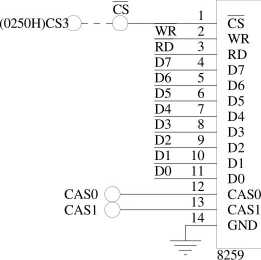
**二、 实验设备**

SUN系列实验仪一套、PC机一台

**三、 实验内容**

1、编制程序：拨动单脉冲开关，“JT”送给8259A的IRO, "IT”送给8259A的IR7, F5 区的最左边数码管对应IR7中断：没有触发中断或中断处理完毕，显示“0”，正在响应中断， 显示“1”； F5区的最右边数码管对应IR0中断：没有触发中断或中断处理完毕，显示“0”，正 在响应中断，显示“1”，如果IR0中断IR7,显示“2”

四、实验原理图



U33

-TL(B2)

INTR

AO

INTA h(B2)

五、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| B3 区：CS、A0 |  | A3 区：CS3、A0 |
| B3 区：INT、INTA |  | A3 区：INTR、INTA |
| B3 区：IR0 |  | B2区：单脉冲JT |
| B3 区：IR7 |  | B2区：单脉冲IT |
| D3 区：CS、AO、A1 |  | A3 区：CS1、AO、A1 |
| D3 区：PCO、PC1 |  | F5 区：KL1、KL2 |
| D3 区：JP20、B、C —— F5 区：A、B、C | | |

2、 运行程序

3、 拨动单脉冲开关，“JT”送给8259A的IRO, “IT”送给8259A的IR7, F5区的最左边 数码管对应IR7中断：没有触发中断或中断处理完毕，显示“0”，正在响应中断，显示“1”； F5区的最右边数码管对应IR0中断：没有触发中断或中断处理完毕，显示“0”，正在响应中断, 显示"1",如果IR0中断IR7,显示"2”

|  |  |  |  |
| --- | --- | --- | --- |
| **六、演示程序** | **（例子程序名8259\_3. asm）** | | |
| .MODEL | TINY | |
| EXTRN | In i tKeyD i s p1ay:NEAR, Display8:NEAR | | |
| 108259\_0 | EQU | 0250H |  |
| 108259\_1 | EQU | 0251H |  |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| BUFFER | DB | 8 DUP (?) |  |
| IRQO\_INT | DB | 0 |  |
| IRQ7\_INT | DB | 0 |  |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | InitKeyDisplay | ；对键盘、数码管扫描控制器8255初始化 |
|  | MOV | IRQO\_INT, OOH |  |
|  | MOV | IRQ7\_INT, OOH |  |
|  | CALL | Init8259 |  |
|  | CALL | Wrilntver |  |
|  | CALL | LedDisplay |  |
|  | STI |  | ；开中断 |
| START1: | LEA | SI,Buffer |  |
|  | CALL | Display8 |  |
|  | JMP | START1 |  |
| LedDisplay | PROC | NEAR |  |
|  | PUSH | AX |  |
|  | MOV | AL,IRQO\_INT |  |
|  | OR | AL,IRQO\_INT |  |
|  | JZ | LedDisplayl |  |
|  | ADD | AL,IRQ7\_INT | ;如果CPU正在处理IR7,显示2 |
| LedDisplayl: | MOV | Buffer, AL | ;显示1或2, IR0在中断 |
|  | MOV | Buffer + 1, 10H |  |
|  | MOV | Buffer + 2, 10H |  |
|  | MOV | Buffer + 3, 10H |  |
|  | MOV | Buffer + 4, 10H |  |
|  | MOV | Buffer + 5, 10H |  |
|  | MOV | Buffer + 6, 10H |  |
|  | MOV | AL,IRQ7\_INT |  |
|  | MOV | Buffer + 7, AL | ;显示1, IR7在中断 |
|  | POP | AX |  |
|  | RET |  |  |
| LedDisplay | ENDP |  |  |
| Init8259 | PROC | NEAR |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | MOV | DX, I08259\_0 |  |
|  | MOV | AL, 13H |  |
|  | OUT | DX, AL |  |
|  | MOV | DX,IO8259\_1 |  |
|  | MOV | AL, 08H |  |
|  | OUT | DX, AL |  |
|  | MOV | AL, 09H |  |
|  | OUT | DX, AL |  |
|  | MOV | AL, 7EH |  |
|  | OUT | DX, AL |  |
|  | RET |  |  |
| Init8259 | ENDP |  |  |
| DELAY | PROC | NEAR |  |
|  | PUSH | SI |  |
|  | PUSH | CX |  |
|  | PUSH | AX |  |
|  | MOV | CX, 100 |  |
| A3: | LEA | SI, BUFFER |  |
|  | CALL | DISPLAY8 |  |
|  | LOOP | A3 |  |
|  | POP | AX |  |
|  | POP | CX |  |
|  | POP | SI |  |
|  | RET |  |  |
| DELAY | ENDP |  |  |
| IRQO: | PUSH | DX |  |
|  | PUSH | AX |  |
|  | INC | IRQO\_INT | ;显示1, IRO在中断 |
|  | CALL | LedDisplay |  |
|  | STI |  |  |
|  | CALL | DELAY |  |
|  | DEC | IRQO\_INT | ;显示0, IRO中断处理完毕 |
|  | CALL | LedDisplay |  |
|  | MOV | DX, I08259\_0 |  |
|  | MOV | AL, 20H |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | POP | DX |  |
|  | I RET |  |  |
| IRQ7: | PUSH | DX |  |
|  | PUSH | AX |  |
|  | INC | IRQ7\_INT | ;显示1, IR7在中断 |
|  | CALL | LedDisplay |  |

|  |  |  |
| --- | --- | --- |
| STI | | |
|  | CALL | DELAY |
|  | DEC | IRQ7\_INT |
|  | CALL | LedDisplay |
|  | MOV | DX, I08259\_0 |
|  | MOV | AL, 20H |
|  | OUT | DX, AL |
|  | POP | AX |
|  | POP | DX |
|  | I RET |  |
| Wrilntver | PROC | NEAR |
|  | PUSH | ES |
|  | MOV | AX,0 |
|  | MOV | ES, AX |
|  | MOV | DI, 20H |
|  | LEA | AX,IRQO |
|  | STOSW |  |
|  | MOV | AX, CS |
|  | STOSW |  |
|  | MOV | DI, 20H+7\*4 |
|  | LEA | AX, IRQ7 |
|  | STOSW |  |
|  | MOV | AX, CS |
|  | STOSW |  |
|  | POP | ES |
|  | RET |  |
| Wrilntver | ENDP |  |
|  | END | START |

;显示0, IR7中断处理完毕

七、实验扩展及思考

1、从8259A收到上升沿,

到8086响应中断，试画这个过程的时序图。

实验七**8251**可编程通信实验（与微机）

一、 实验目的与要求

了解8251的内部结构、工作原理；了解8251与8086的接口逻辑；掌握对8251的初始化 编程方法，学会使用8251实现设备之间的串行通信。

复习本节实验内容，可尝试自行编写程序，做好实验准备工作，填写实验报告。

**二、 实验设备**

SUN系列实验仪一套、PC机一台

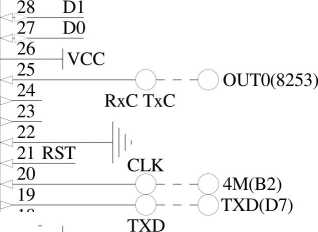
**三、 实验内容**

1、 编制程序：实现8251A与PC机的串行通讯，使用8253作分频器提供8251的收发时钟。

2、 连接线路，从微机接收一批数据，接收完毕，再将它们回送给微机。

**四、 实验原理图**

U31



CS

RxRDYQ

CS4(0240H)

AO

| D3 |  |
| --- | --- |
| 4 | |
| l| D4 | 5 |
| D5 | 6 |
| D6  D7 |  |

D2

D2

D3

RxD

RXD(D7)f> -

RXD

1

RxCTxC 9

| - WR  » | 10 |
| --- | --- |
| ll |
| C\D | 12 |
| RD | 13「 |

1£

DI

DO

VCC

RxC

DTR

RTS

DSR

RESET

CLK

TxD

TxEMPTY

TxRDY

C^D CTS

RD SYNDET RxRDY TxRDY

8 7 6-5

11 11 11 1±

8251

**五、实验步骤**

1、连线说明:

|  |  |  |
| --- | --- | --- |
| C4 区：CS (8253)、 A0、 Al |  | A3 区：CS2、A0、Al |
| C4 区：CLKO |  | B2 区：2M |
| C4 区：GATE |  | Cl区的VCC |
| C4 区：OUTO |  | C3 区：RxC TxC |
| C3 区：CS (8251)、C/D |  | A3 区：CS4、A0 |
| C3 区：CLK |  | B2 区：4M |
| C3 区：RXD、TXD |  | D7 区：RXD、TXD |

2、运行程序

3、 运行“串口助手（ComPort. EXE） ”，设置串口（波特率4800, 8个数据位，一个停止位， 偶校验）,打开串口，选择“HEX发送”、“HEX显示”，向8251发送10个字节数据（输入数据之 间用空格分隔），是否能接收到10个字节数据，接收到的数据是否与发送数据一致。

4、 改变传输数据的数目，重复实验，观察结果。

**六、演示程序**

.MODEL TINY

;使用8253的计数器0,外接2Mhz,经26分频后，送给8251,产生4800bps

CTL\_ADDR EQU 0241H ；控制字或状态字

DATA ADDR EQU 0240H ；读写数据

|  |  |  |  |
| --- | --- | --- | --- |
| W\_8253\_T0 | EQU | 0260H | ;计数器0地址 |
| W\_8253\_C | EQU | 0263H | ;控制字 |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| Receive\_Buffer | DB | 10 DUP(O) | ;接受缓冲器 |
| SendBuffer | EQU | Receive\_Buffer | ;发送缓冲器 |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | INIT\_8253 |  |
|  | CALL | INIT\_8251 |  |
| START1: | MOV | CX, 10 |  |
|  | CALL | Receive\_Group |  |
|  | MOV | CX, 10 |  |
|  | CALL | Send\_Group |  |
|  | JMP | START1 |  |
| INIT\_8253 | PROC | NEAR |  |
|  | MOV | DX, W\_8253\_C |  |
|  | MOV | AL, 37H | ；定时器0,方式3 |
|  | OUT | DX, AL |  |
|  | MOV | DX, W\_8253\_T0 |  |
|  | MOV | AL,26H | ;BCD码26(2000000/26)=16\*4800 |
|  | OUT | DX, AL |  |
|  | MOV | AL, 0 |  |
|  | OUT | DX, AL |  |
|  | RET |  |  |
| INIT\_8253 | ENDP |  |  |
| INIT\_8251 | PROC | NEAR |  |
|  | CALL | RESET\_8251 |  |
|  | MOV | DX, CTL\_ADDR |  |
|  | MOV | AL,7EH | ；波特率系数为16, 8个数据位 |
|  | OUT | DX, AL | ；-个停止位，偶校验 |
|  | CALL | DLTIME | ；延时 |
|  | MOV | AL, 15H | ；允许接收和发送发送数据，清错误标志 |
|  | OUT | DX, AL |  |
|  | CALL | DLTIME |  |
|  | RET |  |  |
| INIT\_8251 | ENDP |  |  |
| Reset\_8251 | PROC | NEAR |  |
|  | MOV | DX, CTL\_ADDR |  |
|  | MOV | AL, 0 |  |
|  | OUT | DX, AL | ;向控制口写入"0" |

|  |  |  |
| --- | --- | --- |
| CALL | DLTIME | ;延时，等待写操作完成 |
| OUT | DX, AL | ;向控制口写入"0" |
| CALL | DLTIME | ;延时 |
| OUT | DX, AL | ;向控制口写入"0" |
| CALL | DLTIME | ;延时 |
| MOV | AL,40H | ；向控制口写入复位字40H |

OUT DX, AL

CALL DLTIME

RET

Reset\_8251 ENDP

；接受一组数据,CX—接受数目

|  |  |  |  |
| --- | --- | --- | --- |
| Receive\_Group | PROC  LEA | NEAR  DI, Receive\_Buffer | |
| Receive\_Groupl: | CALL | Receive\_Byte |  |
|  | STOSB |  |  |
|  | LOOP | Receive\_Groupl |  |
|  | RET |  |  |
| Receive\_Group | ENDP |  |  |
| ；接受一个字节 |  |  |  |
| Receive\_Byte | PROC | NEAR |  |
|  | MOV | DX, CTL\_ADDR |  |
| Receive\_Bytel: | IN | AL, DX | ;读入状态 |
|  | TEST | AL, 2 |  |
|  | JZ | Receive\_Bytel | ;有数据吗？ |
|  | MOV | DX, DATA\_ADDR | ;有 |

IN AL, DX

RET

Receive\_Byte ENDP

；发送一组数据,CX—发送数目

|  |  |  |  |
| --- | --- | --- | --- |
| Send\_Group | PROC  LEA | NEAR  SI, Send\_Buffer | |
| Send\_Groupl: | LODSB |  |  |
|  | CALL | SendByte |  |
|  | LOOP | Send\_Groupl |  |
|  | RET |  |  |
| Send\_Group | ENDP |  |  |
| ;发送一个字节 |  |  |  |
| Sendbyte | PROC | NEAR |  |
|  | PUSH | AX |  |
|  | MOV | DX,CTL\_ADDR | ;读入状态 |
| Sendbytel: | IN | AL, DX |  |
|  | TEST | AL, 1 |  |
|  | JZ | Sendbytel | ;允许数据发送吗？ |
|  | POP | AX | ;发送 |

|  |  |  |
| --- | --- | --- |
| MOV | | DX, DATA\_ADDR |
|  | OUT | DX, AL |
|  | RET |  |
| Sendbyte | ENDP |  |
| 涎时 |  |  |
| DLTIME | PROC | NEAR |
|  | MOV | CX, 10 |
|  | LOOP | $ |
|  | RET |  |
| DLTIME | ENDP |  |
|  | END | START |

七、实验扩展及思考

1、 请读者思考，如何修改程序实现8251的自发自收功能?

2、 8251与8250有何区别？

实验八**8250**可编程通信实验（与微机）

一、实验目的与要求

了解8250的内部结构、工作原理；了解8250与8086的接口逻辑；掌握对8250的初始化 编程方法，学会使用8250实现设备之间的串行通信。

认真预习，做好实验前的准备工作，填写实验报告

二、实验设备

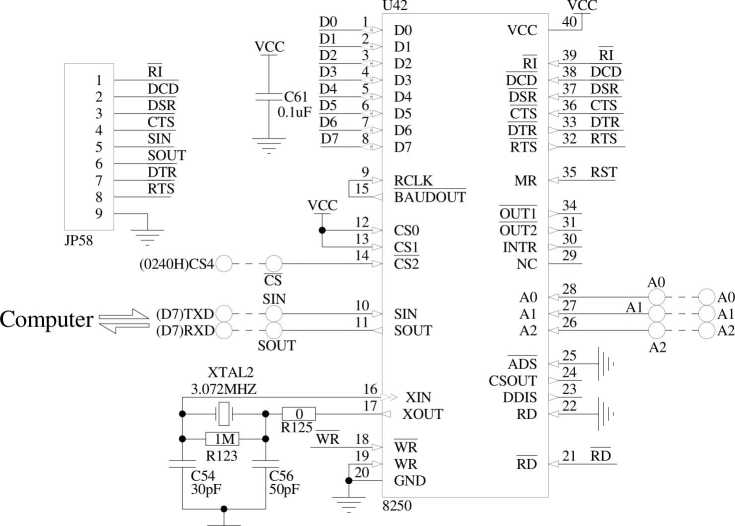
SUN系列实验仪一套、PC机一台

三、实验内容

1、 编写程序：通过初始化8250,设置波特率为4800bps （或其它，但与微机部分一致），数 据格式为8数据位，1停止位，偶校验；然后打开PC机的串行通信测试软件，向8250发送一 批数据，8250接收完数据之后，再将数据依次发送回去。

2、 按图连线，运行程序，观察实验结果，掌握8250的各项功能及编程方法。

**四、实验原理图**



五、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| D5 区：CS、A0、Al、A2 |  | A3 区：CS4、AO、Al、A2 |
| D5 区：SIN、S0UT |  | D7 区：RxD、TxD |

2、运行程序

3、运行“串口助手（ComPort. EXE） ” ,设置串口（波特率4800, 8个数据位，一个停止位， 偶校验）,打开串口，选择“HEX发送”、“HEX显示”，向8250发送10个字节数据（输入数据之 间用空格分隔），是否能接收到10个字节数据，接收到的数据是否与发送数据一致。

4、改变传输数据的数目，重复实验，观察结果。

六、演示程序

.MODEL TINY

;8250和PC机通信，需要在PC上运行一个串口软件，并设置与8250相同的波特率

|  |  |  |  |
| --- | --- | --- | --- |
| NS8250\_Base\_ | Address EQU | 0240H |  |
| RHR | EQU | NS8250\_Base\_Address | ;接收数据缓冲区 |
| THR | EQU | NS8250\_Base\_Address | ;发送数据缓冲区 |
| IER | EQU | NS8250\_Base\_Address+l | ;中断控制寄存器 |
| FCR | EQU | NS8250\_Base\_Address+2 | :FIFO控制寄存器 |
| ISR | EQU | NS8250\_Base\_Address+2 | ;中断状态寄存器 |
| LCR | EQU | NS8250\_Base\_Address+3 | ；串行口控制寄存器 |
| MCR | EQU | NS8250\_Base\_Address+4 | :MODEM控制寄存器 |
| LSR | EQU | NS8250\_Base\_Address+5 | ；串行口状态寄存器 |
| MSR | EQU | NS8250\_Base\_Address+6 | ;MODEM状态寄存器 |
| DLL | EQU | NS8250\_Base\_Address | ；波特率除数锁存器低位 |
| DLM | EQU | NS8250\_Base\_Address+l | ；波特率除数锁存器高位 |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| Receive\_Buffer DB | | 10 DUP（O） ；接受缓冲器 | |
| Send\_Buffer | EQU | Receive\_Buffer ；发送缓冲器 | |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | INIT8250 |  |
| START2: | MOV | ex, 10 ；接收数据（接收完设定的数据个数） | |
|  | CALL | RECEIVE\_GROUP |  |
|  | MOV | CX, 10 ;发送数据（发完设定的数据个数） | |
|  | CALL | SEND\_GROUP |  |
|  | JNC | START2 |  |
| WARNINGl: | JMP | $ |  |

；\*\*\*\*\*\*\*\*\*\*\*\*\*发送一组字符子程序，个数在CX中\*\*\*\*\*\*\*\*\*\*\*

|  |  |  |
| --- | --- | --- |
| Send\_Group | PROC | NEAR |
|  | LEA | SI, Send\_Buffer |
| Send\_Groupl: | LODSB |  |
|  | CALL | SendByte |
|  | JC | Send\_Group2 |
|  | LOOP | SendGroupl |
|  | CLC |  |
| Send\_Group2: | RET |  |
| Send\_Group | ENDP |  |

；\*\*\*\*\*\*\*\*\*\*\*\*\*接收一组字符子程序，存放首地址在DPTR中，个数在R6R7中\*\*\*\*\*\*\*\*\*\*\*

|  |  |  |
| --- | --- | --- |
| Receive\_Group | PROC | NEAR |
|  | LEA | DI, Receive\_Buffer |
| Receive\_Groupl: | :CALL | Receive\_Byte |
|  | STOSB |  |
|  | LOOP | Receive\_Groupl |
|  | CLC |  |
|  | RET |  |
| Receive\_Group | ENDP |  |
| INIT8250 | PROC | NEAR |
|  | MOV | DX, ISR |
|  | MOV | AL, 06H |
|  | OUT | DX, AL |
|  | MOV | DX, LCR |
|  | MOV | AL, 83H ;允许访问波特率因子寄存器 |

OUT DX, AL

MOV DX, DLL

MOV AL, 40

；除数低位寄存器，波特率设为4800二(3. 072\*1000000/16)/DLMDLL

|  |  |  |
| --- | --- | --- |
| OUT | DX, AL | |
| MOV | DX, DLM | ;00H送高字节寄存器 |
| MOV | AL, OOH |  |
| OUT | DX, AL |  |
| MOV | DX, LCR | ；不允许访问波特率因子寄存器 |
| MOV | AL,1BH | ；数据格式为8数据位,1停止位，偶校验 |
| OUT | DX, AL |  |
| RET |  |  |

INIT8250 ENDP

;\*\*\*\*\*\*\*\*\*发送一个字节子程序，发送A中的数，失败置CY\*\*\*\*\*\*\*\*\*

|  |  |  |
| --- | --- | --- |
| SendByte | PROC | NEAR |
|  | PUSH | CX |
|  | PUSH | AX |
|  | MOV | CX, 1000 |
|  | MOV | DX,LSR |
| REP11: | IN | AL, DX |
|  | TEST | AL, 20H |
|  | JNZ | 0UTP0RT1 |
|  | LOOP | REP11 |
|  | POP | AX |
|  | STC |  |
|  | JMP | EXIT8250 |
| 0UTP0RT1: | POP | AX |
|  | MOV | DX, RHR |
|  | OUT | DX, AL |

CLC

EXIT8250: POP CX

RET

SendByte ENDP

;\*\*\*\*\*\*\*\*\*接收一个字节子程序，接收字节在A中，接收失败置1CY\*\*\*\*\*\*\*\*\*

|  |  |  |
| --- | --- | --- |
| Receive\_Byte | PROC | NEAR |
|  | MOV | DX, LSR |
| Receivel: | IN | AL, DX |
|  | TEST | AL, 1 |
|  | JZ | Receivel |
| Receive2: | MOV | DX, RHR |
|  | IN | AL, DX |
| Receive3: | RET |  |
| Receive\_Byte | ENDP |  |
|  | END | START |

七、实验扩展及思考

1、 思考8250与8251有何异同之处？

2、 8250也可以做自发自收的实验，该如何连线及修改程序？

3、 如何通过中断处理方式实现8250串行接收,需要更改哪些线路?

实验九**8255**键盘显示实验

一、实验目的与要求

1、 进一步掌握8255的设计、编程方法。

2、 掌握矩阵键盘的扫描方法

3、 掌握动态扫描数码块的方法

4、 认真预习，做好实验前的准备工作，填写实验报告

二、实验设备

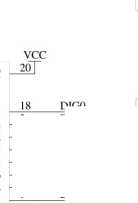
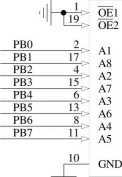
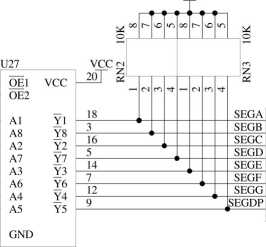
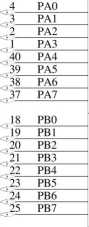
SUN系列实验仪一套、PC机一台

三、实验内容

1、 编写程序：扫描键盘，如有按键，键号显示于数码管。

2、 连接线路，验证8255的功能，熟悉它的使用方法。

四、实验原理图



A0

A1O—O-

A1

O—O-

CSl(0270H) CS

| Dl | 33 |
| --- | --- |
| D2 | 32 |
| D3 | 31 |
| D4 | 30 |
| 05 | 29 |
| D6 | 28 |
| D7 | 27 |
| RD | 5「 |
| WR | 36, |
| 9f | |
|  | |
| RST |  |

DO 34

DO

D1

D2

D3

D4

D5

D6

D7

RD

WR

AO

Al

RESET

CS

01234567 01234567 01234567 PAPAPAPAPAPAPAPAPBPBPBPBPBPBPBPBpcpcpcpcpcpcpcpc

0£

| PAI | 17： |
| --- | --- |
| PA2 | 4 |
| PA3 | 15 |
| PA4 | 6 |
| PA5 | 13] |
| PA6 8 | |
| PA7 | IL |

2

10

4567 3

12一

IT一

10\_

01234567

cccccccc

pppppppp

寸Zdf

VCC

SN74LS240N

U28 

D1GU

Y1

Y8

Y2

Y7

Y3

Y6

Y4

Y5

Jp

| 3 | DIG1 |
| --- | --- |
| 16 | DIG2 |
| 5 | DIG3 |
| 14 | DIG4 |
| 7 | DIG5 |
| 12 | DIG6 |
| 9 | DIG7 |

2

3

1. C

5

6

7

8

SN74LS244N

五、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区:CS、AO、Al |  | A3 区：CS1、AO、Al |
| D3 区：PCO、PCI |  | F5 区：KL1、KL2 |
| D3 区:JP20(PB 口)、JP16(B)、JP17(C) |  | F5 区：A、B、C |

2、运行程序，观察实验结果（任意按下F5区4X4键盘几个键，它上面的8个LED显示器 会将按键的编码从左至右依次显示出来），可依此验证对程序的正确性。

六、演示程序

• MODEL

TINY

|  |  |  |  |
| --- | --- | --- | --- |
| C0M\_8255 | EQU | 0273H | ;8255控制口 |
| PA\_8255 | EQU | 0270H |  |
| PB\_8255 | EQU | 0271H |  |
| PC\_8255 | EQU | 0272H |  |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| buffer | DB | 8 DUP(?) | ;8个字节显示缓冲区 |
| SEG\_TAB | DB | OCOH, 0F9H, 0A4H, OBOH, | 99H, 92H, 82H, 0F8H |
|  | DB | 080H, 90H, 88H, 83H, | 0C6H,0A1H, 86H, 8EH, OFFH |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CLD |  | ;0->DF,地址自动递增 |
|  | MOV | DX, C0M\_8255 |  |
|  | MOV | AL, 89H |  |
|  | OUT | DX, AL | ;PA、PB输出，PC输入 |
|  | LEA | DI, buffer |  |
|  | MOV | AL, 10H |  |
|  | MOV | CX, 08H |  |
|  | REP | STOSB |  |
|  | CALL | DIR |  |
| MAIN3: | LEA | DI, buffer |  |
| MAIN2: | CALL | keyi |  |
|  | STOSB |  |  |
|  | CALL | DIR |  |
|  | CMP | DI,offset buffer+8 |  |
|  | JNZ | MAIN2 |  |
|  | JMP | MAIN3 |  |
| DIR | PROC | NEAR |  |
|  | PUSH | AX |  |
|  | PUSH | BX |  |
|  | PUSH | DX |  |
|  | LEA | SI, buffer | ；置显示缓冲器初值 |
|  | MOV | AH, OFEH |  |
|  | LEA | BX, SEG\_TAB |  |
| LDO: | MOV | DX,PA\_8255 |  |
|  | LODSB |  |  |
|  | XLAT |  | ；取显示数据 |
|  | OUT | DX, AL | ；段数据->8255 PA 口 |
|  | INC | DX | ；扫描模式->8255 PB 口 |
|  | MOV | AL, AH |  |
|  | OUT | DX, AL |  |

|  |  |  |
| --- | --- | --- |
| CALL | | DL1 |
|  | MOV | DX, PB\_8255 |
|  | MOV | AL, OFFH |
|  | OUT | DX, AL |
|  | TEST | AH,80H |
|  | JZ | LD1 |
|  | ROL | AH, 01H |
|  | JMP | LDO |
| LD1: | POP | DX |
|  | POP | BX |
|  | POP | AX |
|  | RET |  |
| DIR | ENDP |  |
| DL1 | PROC | NEAR |
|  | PUSH | CX |
|  | MOV | CX, 500 |
|  | LOOP | $ |
|  | POP | CX |
|  | RET |  |
| DL1 | ENDP |  |
| KEYI | PROC | NEAR |
|  | PUSH | BX |
|  | PUSH | DX |
| LK: | CALL | AllKey |
|  | JNZ | LK1 |
|  | CALL | DIR |
|  | CALL | DIR |
|  | JMP | LK |
| LK1: | CALL | DIR |
|  | CALL | DIR |
|  | CALL | AllKey |
|  | JNZ | LK2 |
|  | CALL | DIR |
|  | JMP | LK |
| LK2: | MOV | BL, OFEH |
|  | MOV | BH,0 |
| LK4: | MOV | DX, PB\_8255 |
|  | MOV | AL, BL |
|  | OUT | DX, AL |
|  | INC | DX |
|  | IN | AL, DX |
|  | TEST | AL,01H |
|  | JNZ | LONE |
|  | XOR | AL, AL |

;延迟1ms

；延迟子程序

；调用判有无闭合键子程序

;调用显示子程序，延迟6ms

；调用判有无闭合键子程序

:R2

:R4

；0行有键闭合

|  |  |  |  |
| --- | --- | --- | --- |
|  | JMP | LKP | |
| LONE: | TEST | AL, 02H |  |
|  | JNZ | NEXT |  |
|  | MOV | AL, 08H | :1行有键闭合 |
| LKP: | ADD | BH, AL |  |
| LK3: | CALL | DIR | ;判断释放否 |
|  | CALL | AllKey |  |
|  | JNZ | LK3 |  |
|  | MOV | AL, BH | ;键号-〉AL |
|  | POP | DX |  |
|  | POP | BX |  |
|  | RET |  |  |
| NEXT: | INC | BH | ;列计数器加1 |
|  | TEST | BL, 80H |  |
|  | JZ | KND | ；判是否已扫到最后一列 |
|  | ROL | BL,O1H |  |
|  | JMP | LK4 |  |
| KND: | JMP | LK |  |
| KEYI | ENDP |  |  |
| AllKey | PROC | NEAR |  |
|  | MOV | DX, PB\_8255 |  |
|  | XOR | AL, AL |  |
|  | OUT | DX, AL | ;全"0"-＞扫描口 |
|  | INC | DX |  |
|  | IN | AL, DX | ；读键状态 |
|  | NOT | AL |  |
|  | AND | AL, 03H | ;取低二位 |
|  | RET |  |  |
| AllKey | ENDP |  |  |
|  | END | START |  |

七、实验扩展及思考

1、 显示程序中延时函数起什么作用？如何调节数码块亮度?

2、 重新编写软件实验二，自己编写键扫描、显示程序

实验十 并行**DA**实验

**一、实验目的**

了解数模转换的原理；了解0832与8086的接口逻辑，掌握使用DAC0832进行数模转换。

二、实验设备

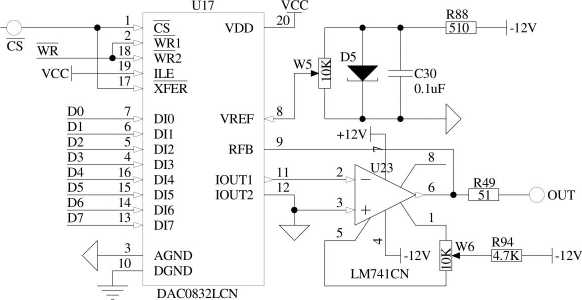
SUN系列实验仪一套、PC机一台、示波器一台。

三、实验内容

1、 编写程序：用0832输出正弦波

2、 按图连线，运行程序，使用示波器观察实验结果。

四、实验原理图



CS4(0240H)-

**五、实验步骤**

1、连线说明:

|  |  |  |
| --- | --- | --- |
| D2 区：CS |  | A3 区：CS4 |

2、运行程序，示波器的探头接D2区的OUT,观察实验结果，是否产生正弦波。

|  |  |  |  |
| --- | --- | --- | --- |
| **六、演示程序** | | .MODEL | TINY |
| ADDR\_0832 |  | EQU | 0240H ;0832输出口地址 |
|  |  | .STACK | 100 |
|  |  | .DATA |  |
| TAB\_1 | DB | 7FH, 8BH, 96H, 0A1H, OABH, 0B6H, OCOH, 0C9H, 0D2H | |
|  | DB | ODAH, 0E2H, 0E8H, OEEH, 0F4H, 0F8H, OFBH, OFEH, OFFH, OFFH | |
|  | DB | OFFH, OFEH, OFBH, 0F8H, 0F4H, OEEH, 0E8H, 0E2H, ODAH, 0D2H | |
|  | DB | 0C9H, OCOH, 0B6H, OABH, 0A1H, 096H, 08BH, 07FH | |
|  | DB | 74H, 69H, 5EH, 54H, 49H, 40H, 36H, 2DH, 25H, 1DH, 17H, 11H, OBH, 7, 4, 2, 0, 0 | |
|  | DB | 0, 2, 4, | 7, OBH, 11H, 17H, 1DH, 25H, 2DH, 36H, 40H, 49H, 54H, 5EH, 69H, 74H |
|  |  | .CODE |  |
| START: |  | MOV | AX, @DATA |
|  |  | MOV | DS, AX |
|  |  | NOP |  |

|  |  |  |
| --- | --- | --- |
|  | MOV | DX, ADDR\_0832 |
| START1: | LEA | SI, TAB\_1 |
|  | MOV | CX, 72 |
| START2: | LODSB |  |
|  | OUT | DX, AL |
|  | CALL | DELAY |
|  | LOOP | START2 |
|  | JMP | START1 |
| DELAY | PROC | NEAR |
|  | PUSH | ex |
|  | MOV | CX, 80 |
|  | LOOP | $ |
|  | POP | ex |
|  | RET |  |
| DELAY | ENDP |  |
|  | END | START |

实验■— 并行**AD**实验（数字电压表实验）

一、 实验目的与要求

1、 了解几种类型AD转换的原理；掌握使用ADC0809进行模数转换

2、 认真预习实验内容，做好准备工作，完成实验报告。

二、 实验设备

SUN系列实验仪一套、PC机一台、万用表一个。

三、 实验内容

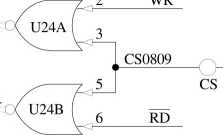
1、 ADC0809 （C2 区）

（1） 模数转换器，8位精度，8路转换通道，1路并行输出

（2） 转换时间lOOus,转换电压范围0〜5V

2、 编写程序：制作一个电压表，测量0〜5V,结果显示于数码管上。

四、实验原理图



500K(B2)O -

CLK

0-5v(F6)O——O

IN0

IN3

IN5

1N7

AOQ—O-

ADDA

AlO-<

ADDB

A2(>-O

ADDC

U16

10

26

\_>#>  
—工  
—亍  
*—三;*

22

■:--

13

CLK

START

NOO6080OCIV 01234567 [nn[n[nnn[nn

ADDA ADDB ADDC ALE

GND

vcc

EOC

OE

DO

DI

D2

D3

D4

D5

D6

D7

VREF+

VREF-

| vcc

*J*

,9

17 DO

14 Df

<x

1. DI
2. D2

8 D3

1. DT
2. D5
3. D6
4. D7

VCC

16

1

--O CS3(0250H)

**五、实验步骤**

1、连线说明:

|  |  |  |
| --- | --- | --- |
| C2 区：CS、ADDA、ADDB、ADDC |  | A3 区：CS3、AO、Al、A2 （选择通道） |
| C2 区：CLK |  | B2 区：500K |
| C2 区：INO |  | F6区：0〜5V |
| D3 区：CS、AO、Al |  | A3 区：CS1、AO、Al |
| D3 区：PCO、PCI |  | F5 区：KL1、KL2 |
| D3 区：JP20、B、C |  | F5 区：A、B、C |

2、调节0〜5V电位器（F6区）输出电压，显示在LED （最右边2位）上的电压数字量会随之 改变。用万用表验证AD转换的结果。

**六、演示程序**

.MODEL TINY

ADDR\_0809 EQU 0250H

EXTRN InitKevDisplay:NEAR, Display8:NEAR

.STACK

.DATA

100

|  |  |  |
| --- | --- | --- |
| BUFFER | DB | 8 DUP(?) |
| LastAD | DB  .CODE | 0 |
| START: | MOV | AX, @DATA |
|  | MOV  NOP | DS, AX |
|  | CALL | InitKeyDisplay |
|  | XOR | AL, AL |
|  | JMP | START6 |
| START1: | MOV | CX, 8 |
|  | MOV | BX, 0 |
| START2: | CALL | AD0809 |
|  | XOR | AH, AH |
|  | ADD | BX, AX |
|  | LOOP | START2 |
|  | MOV | AX, 8 |
|  | XCHG | AX, BX |
|  | DIV | BL |
|  | CMP | AL, LastAD |
|  | JZ | START3 |
| START6: | MOV | LastAD, AL |
|  | CALL | Display\_Data |
| START3: | CALL | DLTime |
|  | JMP | START1 |
| AD0809 | PROC | NEAR |
|  | PUSH | CX |
|  | MOV | AL, 0 |
|  | MOV | DX, ADDR\_0809 |
|  | OUT | DX, AL |
|  | MOV | CX, 100 |
|  | LOOP | $ |
|  | MOV | DX, ADDR\_0809 |
|  | IN | AL, DX |
|  | POP  RET | CX |
| AD0809 | ENDP |  |
| DISPLAY\_DATA | PROC | NEAR |
|  | MOV | AH, AL |
|  | AND | AL, OFH |
|  | MOV | BUFFER + 4, AL |
|  | MOV | AL, AH |
|  | AND | AL, OFOH |

;上一次AD转换值

;初始化键盘数码管控制器(8255)

；采样8次

；累计8次的采样值

；8次的平均值

；延时，等待AD转换完成

|  |  |  |  |
| --- | --- | --- | --- |
| ROR | | AL, 4 | |
|  | MOV | BUFFER + 5, AL |  |
|  | MOV | AL, AH |  |
|  | XOR | AH, AH |  |
|  | MOV | BL, 51 | ;255/51 (16进制的 1 |
|  | DIV | BL |  |
|  | OR | AL,80H | ;加上小数点 |
|  | MOV | BUFFER + 2, AL |  |
|  | MOV | AL, 10 |  |
|  | MUL | AH |  |
|  | DIV | BL |  |
|  | MOV | BUFFER + 1,AL | ；第一位小数 |
|  | MOV | AL, 10 |  |
|  | MUL | AH |  |
|  | DIV | BL |  |
|  | MOV | BUFFER, AL | ；第二位小数 |
|  | MOV | buffer+3, 10H |  |
|  | MOV | buffer+6, 10H |  |
|  | MOV | buffer+7, 10H | ；消隐 |
|  | RET |  |  |
| DISPLAY\_DATA | ENDP |  |  |
| DLTime | PROC | NEAR |  |
|  | MOV | CX, 10 |  |
|  | LEA | SI, buffer |  |
| XX: | CALL | Display8 |  |
|  | LOOP | XX |  |
|  | RET |  |  |
| DLTime | ENDP |  |  |
|  | END | START |  |
| **七、实验扩展及思考** | |  |  |
| 如何实现多路模拟量的数据采集、显示？ | | |  |

1/51V)

实验十二红外通信实验

一、 实验目的

1、理解红外通讯原理；2、掌握红外通讯；3、熟练使用8250

二、 实验设备

SUN系列实验仪一套、PC机一台。

三、 实验内容

1、 红外通讯原理

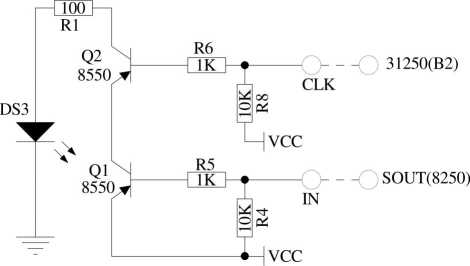
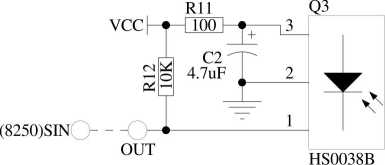
当红外接收器收到38kHZ频率的信号，输出电平会由1->0, 一旦没有此频率信号，输出电 平会由0 —〉1。因此，红外发射头控制通断发射38kHZ信号，就可以将数据发送出来

2、 实验过程

（1） 8250使用红外发送管和接收器进行数据自发自收

（2） 根据接收到的数据，通过8255的PA 口点亮8个发光管，会看到发光管不断变化

**四、实验原理图**



五、实验步骤

1、连线说明:

|  |  |  |
| --- | --- | --- |
| F1 区：IN、OUT |  | D5 区：SOUT, SIN |
| Fl 区：CLK |  | B2 区：31250 |
| D5 区：CS、AO、Al、A2 |  | A3 区：CS4、AO、Al、A2 |
| D3 区：CS、AO、Al |  | A3 区：CS1、A0、Al |
| D3 区：JP23 (PA) |  | F4 区:JP18 |

2、调试该程序时，使用较厚的白纸挡住红外发射管红外信号，使它反射到接收头。

说明：一般红外接收模块的解调频率为38KHz,当它接收到38KHz左右的红外信号时将输出低 电平，但连续输出低电平的时间是有限制的（如：100ms）,也就是说输出低电平的宽度是有限 制的。

3、发送数据，并接收，根据接收到的数据点亮8个发光管，程序运行之后，会看到8个发 光管（F4区）在闪烁，从第8个（最右边）向第1个逐一点亮过去。本实验通过红外通讯发送、接 收数据，发送的数据从00H开始+1,接收到该数据后用来点亮8个发光管。亮一1,熄一0。

**六、演示程序**

.MODEL TINY

;8250和PC机通信，需要在PC上运行一个串口软件，并设置与8250相同的波特率

|  |  |  |  |
| --- | --- | --- | --- |
| NS8250\_Base\_Address EQU | | 0240H | |
| RHR | EQU | NS8250\_Base\_Address | ;接收数据缓冲区 |
| THR | EQU | NS8250\_Base\_Address | ;发送数据缓冲区 |
| IER | EQU | NS8250\_Base\_Address+l ；中断控制寄存器 | |
| FCR | EQU | NS8250\_Base\_Address+2 ; FIFO 控制寄存器 | |
| ISR | EQU | NS8250\_Base\_Address+2 ；中断状态寄存器 | |
| LCR | EQU | NS8250\_Base\_Address+3 ;串行口控制寄存器 | |
| MCR | EQU | NS8250\_Base\_Address+4 ; MODEM 控制寄存器 | |
| LSR | EQU | NS8250\_Base\_Address+5 ；串行口状态寄存器 | |
| MSR | EQU | NS8250\_Base\_Address+6 ; MODEM 状态寄存器 | |
| DLL | EQU | NS8250\_Base\_Address | ;波特率除数锁存器低位 |
| DLM | EQU | NS8250\_Base\_Address+l ;波特率除数锁存器高位 | |
| C0M\_8255 | EQU | 0273H | ;8255控制口 |
| PA\_8255 | EQU | 0270H | ;8255PA口 |
| .STACK | 100 |  |  |
| .DATA |  |  |  |
| Receive\_Buffer DB | | 10 DUP(O) | ；接受缓冲器 |
| SendBuffer | EQU | Receive\_Buffer | ;发送缓冲器 |
| .CODE |  |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | MOV | DX, C0M\_8255 |  |
|  | MOV | AL, 80H |  |
|  | OUT | DX, AL | ;PA、PB、PC为基本输出模式 |
|  | INC | DX |  |
|  | MOV | AL, OFFH |  |
|  | OUT | DX, AL | ；熄灭指示灯 |
|  | CALL | INIT8250 |  |
|  | MOV | BX,0 |  |
| START1: | CALL | InfraredTest | ；调用自收自发红外通讯子程序 |
|  | CALL | DelayTime | 涎时 |
|  | JMP | START1 | ;循环进行红外通讯 |
| INIT8250 | PROC | NEAR |  |
|  | MOV | DX, ISR |  |
|  | MOV | AL, 06H |  |
|  | OUT | DX, AL |  |
|  | MOV | DX, LCR |  |
|  | MOV | AL, 83H | ；允许访问波特率因子寄存器 |
|  | OUT | DX, AL |  |
|  | MOV | DX, DLL |  |

INIT8250

| MOV | AL, 80 | ；波特率为2400= (3. 072\*1000000/16) /DLMDLL |
| --- | --- | --- |
| OUT | DX, AL |  |
| MOV | DX, DLM | ;00H送高字节寄存器 |
| MOV | AL, OOH |  |
| OUT | DX, AL |  |
| MOV | DX, LCR | ；不允许访问波特率因子寄存器 |
| MOV | AL, 1BH | ；数据格式为8数据位,1停止位，偶校验 |
| OUT | DX, AL |  |
| RET |  |  |
| ENDP |  |  |

;\*\*\*\*\*\*\*\*\*发送一个字节子程序，发送A中的数，失败置1CY\*\*\*\*\*\*\*\*\*

|  |  |  |
| --- | --- | --- |
| SendByte | PROC | NEAR |
|  | PUSH | AX |
|  | MOV | CX, 1000 |
|  | MOV | DX, LSR |
| REP11: | IN | AL, DX |
|  | TEST | AL, 20H |
|  | JNZ | 0UTP0RT1 |
|  | LOOP | REP 11 |
|  | POP | AX |
|  | STC |  |
|  | JMP | EXIT8250 |
| 0UTP0RT1: | POP | AX |
|  | MOV | DX, RHR |
|  | OUT | DX, AL |
|  | CLC |  |
| EXIT8250: | RET |  |
| SendByte | ENDP |  |

；\*\*\*\*\*\*\*\*\*接收一个字节子程序，接收字节在A中,接收失败置1CY\*\*\*\*\*\*\*\*\*

|  |  |  |
| --- | --- | --- |
| Receive\_Byte | PROC | NEAR |
|  | MOV | CX, 1000 |
|  | MOV | DX, LSR |
| Receivel: | IN | AL, DX |
|  | TEST | AL, 1 |
|  | JNZ | Receive2 |
|  | LOOP | Receivel |
|  | STC |  |
|  | JMP | Receive3 |
| Receive2: | MOV | DX, RHR |
|  | IN | AL, DX |
|  | CLC |  |
| Receive3: | RET |  |
| Receive\_Byte | ENDP |  |
| ；红外通讯 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| InfraredTest | PROC | NEAR | |
|  | MOV | AL, BL |  |
|  | CALL | Send\_Byte |  |
|  | JNB | InfraredTestl |  |
|  | MOV | AL, OFFH | ;无法发送 |
|  | JMP | InfraredTest2 |  |
| Infrared Testl: | CALL | Receive\_Byte |  |
|  | JNB | InfraredTest2 |  |
|  | MOV AL, 0 | ；无法接收 |  |
| Infrared\_Test2: | CALL | Light | ；根据收接到的数据点亮8个红色发光管 |
|  | INC | BX | ;发送数据逐步递增 |
|  | RET |  |  |
| InfraredTest | ENDP |  |  |
| ；点亮8个发光管 |  |  |  |
| Light | PROC | NEAR |  |
|  | NOT | AL | ；0-亮，1-灭 |
|  | MOV | DX, PA\_8255 |  |
|  | OUT | DX, AL |  |
|  | RET |  |  |
| Light | ENDP |  |  |
| ；延时程序 |  |  |  |
| DelayTime | PROC | NEAR |  |
|  | MOV | CX, 60000 | ;延时0.5s |
|  | LOOP | $ |  |
|  | RET |  |  |
| DelayTime | ENDP |  |  |
|  | END | START |  |

七、实验扩展及思考

实验名称：红外遥控器实验

实验目的：了解日常所用的家电红外遥控器是如何工作的

实验内容：结合按键模拟4路红外遥控器，遥控发光管或电机转动快慢

实验十三图形点阵显示实验

一、实验目的与要求

了解图形液晶模块的控制方法；了解它与8086的接口逻辑；掌握使用图形点阵液晶显示字 体和图形。

二、实验设备

SUN系列实验仪一套、PC机一台。

三、实验内容

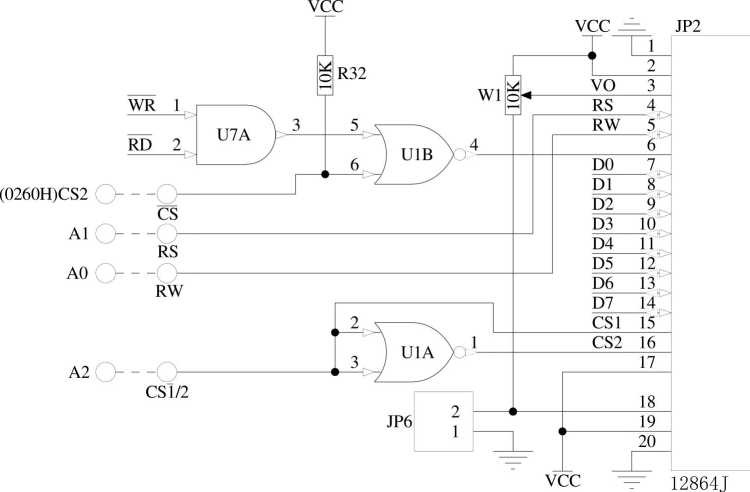
1> 12864J液晶显示器

1. 图形点阵液晶显示器，分辨率为128X64„可显示图形和8X4个(16X16点阵)汉字。
2. 采用8位数据总线并行输入输出和4条控制线。
3. 指令简单，7种指令

2、实验过程

在12864J液晶上显示一段字，包括汉字和英文：“星研电子”、“STAR ES51PRO”、“欢迎使 用”，三行字。

**四、实验原理图**



VEE

LED1

LED2

寸 98ZIOWS

S)D>y012345671 2-T SDOS/VBBBBBBBBSszsl VVVRR7EDDDDDDDDC C-R

**五、实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| Al 区：CS、RW、RS、CS1/2 |  | A3 区:CS2、AO、Al、A2 |

2、运行程序，验证显示结果。

六、演示程序

.MODEL

WR COM AD L

EQU

TINY

0264H

；写左半屏指令地址

|  |  |
| --- | --- |
| Line3\_2 DB  DB  文字： | 40H, 41H, OCEH, 04H, OOH, OFCH, 04H, 02H, 02H, OFCH, 04H, 04H, 04H, OFCH, OOH, OOH  40H, 20H, 1FH, 20H, 40H, 47H, 42H, 41H, 40H, 5FH, 40H, 42H, 44H, 43H, 40H, OOH  使一 |

Line3\_3 DB

40H, 20H, OFOH, 1CH, 07H, 0F2H, 94H, 94H, 94H, OFFH, 94H, 94H, 94H, 0F4H, 04H, OOH

| WR\_COM\_AD\_R | EQU | 0260H | ;写右半屏指令地址 |
| --- | --- | --- | --- |
| WR\_DATA\_AD\_L | EQU | 0266H | ;写左半屏数据地址 |
| WR\_DATA\_AD\_R | EQU | 0262H | ;写右半屏数据地址 |
| RD\_BUSY\_AD | EQU | 0261H | ;查忙地址 |
| RD\_DATA\_AD | EQU | 0263H | ;读数据地址 |
| X | EQU | 0B8H | ;起始显示行基址 |
| Y | EQU | 040H | ;起始显示列基址 |
| FirstLine | EQU | OCOH | ;起始显示行 |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| 文字：星 | 一一 |  |  |

Linel\_l DB

OOH, OOH, OFCH, 82H, 82H, OAAH, 2AH, OAAH, OAAH, OAAH, 02AH, 02H, 02H, OFCH, OOH, OOH

DB OOH, OEEH, 9BH, 90H, 98H, 94H, 95H, 80H, 80H, 80H, 95H, 95H, 95H, 95H, OFFH, OOH

；— 文字：研 --

Linel\_2 DB

| DB  文字： | 03H, 3CH, 40H, 40H, 46H, 40H, 0F1H, 8EH, 80H, 40H, 7CH, 80H, 80H, 80H, OFEH, 03H 电一一 |
| --- | --- |

9EH, 62H, 02H, 02H, 02H, 32H, OFEH, 62H, 02H, 02H, 32H, 02H, 02H, 02H, 62H, ODCH

OOH, 0F8H, 04H, 04H, 44H, 44H, 06H, 02H, 02H, 46H, 44H, 04H, 04H, 0F8H, OOH, OOH

Linel\_3 DB

DB OOH, OFH, 10H, 10H, 11H, 11H, OFOH, 80H, 90H, 91H, 91H, 8CH, 84H, 87H, 0C8H, 78H

；-文字：子一

Linel\_4 DB

80H, 40H, 5EH, 52H, 52H, 52H, 32H, 72H, 82H, 82H, 42H, 62H, 52H, 4CH, OCOH, OOH

DB 07H, 04H, 04H, 04H, OFCH, 8CH, 8CH, 80H, 80H, 7CH, 04H, 04H, 04H, 04H, 07H, OOH

;” STARES51PRO”

Line2 1

DB

Line2 2

DB

Line2 3

DB

Line2\_4

Line2 5

DB

DB

Line2 6

DB

Line2 7

DB

Line2 8

DB

Line2 9

DB

Line2 10

DB

Line2 11

DB

DB

；—文字：

Line3 1 DB

Line2 12

DB 文字:

OOH, 70H, 88H, 08H, 08H, 08H, 38H, OOH, OOH, 38H, 20H, 21H, 21H, 22H, 1CH, OOH

18H, 08H, 08H, 0F8H, 08H, 08H, 18H, OOH, OOH, OOH, 20H, 3FH, 20H, OOH, OOH, OOH OOH, OOH, OCOH, 38H, OEOH, OOH, OOH, OOH, 20H, 3CH, 23H, 02H, 02H, 27H, 38H, 20H 08H, 0F8H, 88H, 88H, 88H, 88H, 70H, OOH, 20H, 3FH, 20H, OOH, 03H, OCH, 30H, 20H OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH 08H, 0F8H, 88H, 88H, 0E8H, 08H, 10H, OOH, 20H, 3FH, 20H, 20H, 23H, 20H, 18H, OOH OOH, 70H, 88H, 08H, 08H, 08H, 38H, OOH, OOH, 38H, 20H, 21H, 21H, 22H, 1CH, OOH

OOH, 0F8H, 08H, 88H, 88H, 08H, 08H, OOH, OOH, 19H, 21H, 20H, 20H, 11H, OEH, OOH OOH, 10H, 10H, 0F8H, OOH, OOH, OOH, OOH, OOH, 20H, 20H, 3FH, 20H, 20H, OOH, OOH 08H, 0F8H, 08H, 08H, 08H, 08H, OFOH, OOH, 20H, 3FH, 21H, 01H, 01H, 01H, OOH, OOH 08H, 0F8H, 88H, 88H, 88H, 88H, 70H, OOH, 20H, 3FH, 20H, OOH, 03H, OCH, 30H, 20H

OEOH, 10H, 08H, 08H, 08H, 10H, OEOH, OOH, OFH, 10H, 20H, 20H, 20H, 10H, OFH, OOH 欢--

14H, 24H, 44H, 84H, 64H, 1CH, 20H, 18H, OFH, 0E8H, 08H, 08H, 28H, 18H, 08H, OOH

20H, 10H, 4CH, 43H, 43H, 2CH, 20H, 10H, OCH, 03H, 06H, 18H, 30H, 60H, 20H, OOH 迎 --

DB OOH, OOH, 7FH, OOH, 40H, 41H, 22H, 14H, OCH, 13H, 10H, 30H, 20H, 61H, 20H, OOH

|  |  |  |  |
| --- | --- | --- | --- |
| 文字： | 用-- | | |
| Line3\_4 DB | OOH, OOH, OOH, OFEH, 22H, 22H, 22H, 22H, OFEH, 22H, 22H, 22H, 22H, OFEH, OOH, OOH | | |
| DB | 80H, 40H, 30H, OFH, 02H, 02H, 02H, 02H, OFFH, 02H, 02H, 42H, 82H, 7FH, OOH, OOH | | |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | NOP |  |  |
| START1: | CALL | LCD\_INIT | ;液晶初始化 |
|  | CALL | DelayTime |  |
|  | CALL | DisLinel | ;第2行显示”星研电子” |
|  | CALL | DelayTime |  |
|  | CALL | DisLine2 | ;第3行显示” STAR ES51PR0” |
|  | CALL | DelayTime |  |
|  | CALL | DisLine3 | ;第4行显示”欢迎使用” |
|  | CALL | DelayTime |  |
|  | JMP | START1 |  |
| ；延时程序 |  |  |  |
| DelayTime | PROC | NEAR |  |
|  | MOV | CX, 0 |  |
|  | LOOP | $ |  |
|  | LOOP | $ |  |
|  | RET |  |  |
| DelayTime | ENDP |  |  |
| ;第2行显示” | 星研电子” |  |  |
| DisLinel | PROC | NEAR |  |
|  | LEA | SI,Linel\_l |  |
|  | MOV | AL, 2 | ;A-起始显示行地址，第2行 |
|  | MOV | AH, 32 | ；B-起始显示列地址，第32列，以下同 |
|  | CALL | WordDISL | ;左半屏，显示一个字子程序 |
|  | LEA | SI,Linel\_2 |  |
|  | MOV | AL, 2 |  |
|  | MOV | AH, 48 |  |
|  | CALL | WordDISL |  |
|  | LEA | SI,Linel\_3 |  |
|  | MOV | AL, 2 |  |
|  | MOV | AH, 0 |  |
|  | CALL | WordDISR | ;右半屏，显示一个字子程序 |
|  | LEA | SI,Linel\_4 |  |
|  | MOV | AL, 2 |  |
|  | MOV | AH, 16 |  |
|  | CALL | WordDISR |  |

RET

DisLinel ENDP

；第3行显示” STAR ES51PR0”

DisLine2

|  |  |
| --- | --- |
| PR0C | NEAR |
| LEA | SI, Line2\_l |
| MOV | AL, 4 |
| MOV | AH, 16 |
| CALL | ByteDISL |
| LEA | SI,Line2\_2 |
| MOV | AL, 4 |
| MOV | AH, 24 |
| CALL | ByteDISL |
| LEA | SI, Line2\_3 |
| MOV | AL, 4 |
| MOV | AH, 32 |
| CALL | ByteDISL |
| LEA | SI,Line2\_4 |
| MOV | AL, 4 |
| MOV | AH, 40 |
| CALL | ByteDISL |
| LEA | SI,Line2\_5 |
| MOV | AL, 4 |
| MOV | AH, 48 |
| CALL | ByteDISL |
| LEA | SI,Line2\_6 |
| MOV | AL, 4 |
| MOV | AH, 56 |
| CALL | ByteDISL |
| LEA | SI,Line2\_7 |
| MOV | AL, 4 |
| MOV | AH, 0 |
| CALL | ByteDISR |
| LEA | SI,Line2\_8 |
| MOV | AL, 4 |
| MOV | AH, 8 |
| CALL | ByteDISR |
| LEA | SI,Line2\_9 |
| MOV | AL, 4 |
| MOV | AH, 16 |
| CALL | ByteDISR |
| LEA | SI,Line2\_10 |
| MOV | AL, 4 |
| MOV | AH, 24 |
| CALL | ByteDISR |
| LEA | SI,Line2\_ll |
| MOV | AL, 4 |
| MOV | AH, 32 |

；A-起始显示行地址，第4行

;B-起始显示列地址，第16列，以下同

;左半屏，显示一个字节子程序

；右半屏字节显示数据

CALL

LEA

MOV

MOV

CALL

RET

ByteDISR

SI,Line2\_12

AL, 4

AH, 40

ByteDISR

DisLine2 ENDP

;第4行显示”欢迎使用”

|  |  |
| --- | --- |
| DisLine3 | PROC |
| DisLine3 | LEA  MOV  MOV  CALL  LEA  MOV  MOV  CALL  LEA  MOV  MOV  CALL  LEA  MOV  MOV  CALL  RET  ENDP |
| ；液晶初始化  LCD\_INIT | PROC |
| LCD\_INIT | MOV CALL  MOV  CALL  MOV  CALL  MOV  CALL  CALL  MOV  CALL  MOV  CALL  RET  ENDP |

NEAR

SI, Line3\_l

AL, 6

AH, 32 WordDISL

SI,Line3\_2

AL, 6

AH, 48

WordDISL

SI, Line3\_3

AL, 6

AH, 0

WordDISR

SI,Line3\_4

AL, 6

AH, 16

WordDISR

NEAR

AL, 3EH

WRComL

AL, FirstLine

WRComL

AL, 3EH

WRComR

AL, FirstLine

WRComR

LCDClear

AL, 3FH

WRComL

AL, 3FH

WRComR

;A-起始显示行地址，第6行

;沪起始显示列地址，第32列，以下同

;左半屏，显示一个字子程序

；右半屏，显示一个字子程序

;初始化左半屏，关显示

;写指令子程序

；设置起始显示行，第0行

;初始化右半屏，关显示

;写指令子程序

；设置起始显示行，第0行

;清屏

;开显示

;开显示

|  |  |  |  |
| --- | --- | --- | --- |
| ;清屏 | | | |
| LCDClear | PROC | NEAR |  |
|  | ；清左半屏 |  |  |
|  | MOV | AL, 0 | ；起始行，第o行 |
|  | MOV | AH, 0 | ；起始列，第0列 |
| LCDClearLl: | PUSH | AX |  |
|  | MOV | CX, 64 |  |
|  | CALL | SETXYL | ;设置起始显示行列地址 |
| LCDClearL2: | MOV | AL, 0 |  |
|  | CALL | WRDATAL |  |
|  | LOOP | LCDClearL2 |  |
|  | POP | AX |  |
|  | INC | AX |  |
|  | CMP | AL, 8 | ；共8行 |
|  | JNZ  ；清右半屏 | LCDClearLl |  |
|  | MOV | AL, 0 | ；起始行，第0行 |
|  | MOV | AH, 0 | ；起始列，第0列 |
| LCDClearRl: | PUSH | AX |  |
|  | MOV | CX, 64 |  |
|  | CALL | SETXYR | ;设置起始显示行列地址 |
| LCDClearR2: | XOR | AL, AL |  |
|  | CALL | WRDATAR |  |
|  | LOOP | LCDClearR2 |  |
|  | POP | AX |  |
|  | INC | AL |  |
|  | CMP | AL, 8 | ；共8行 |
|  | JNZ | LCDClearRl |  |
|  | RET |  |  |
| LCDClear | ENDP |  |  |

；显示字体，显示一个数据要占用X行两行位置

;左半屏显示一个字节/字：AL-起始显示行序数X(0-7) ； AH-起始显示列序数Y (0-63) ； SI-显示字

|  |  |  |  |
| --- | --- | --- | --- |
| 数据首地址 | | | |
| ByteDisL | PROC | NEAR |  |
|  | MOV | CX, 8 | ；显示8个字节数据，用于显示一个英文/符号 |
|  | CALL | DispL |  |
|  | RET |  |  |
| ByteDisL | ENDP |  |  |
| WordDisL | PROC | NEAR |  |
|  | MOV | CX, 16 | ;显示16字节数据，用于显示一个汉字 |
|  | CALL | DispL |  |
|  | RET |  |  |
| WordDisL | ENDP |  |  |
| DispL | PROC | NEAR |  |

|  |  |  |  |
| --- | --- | --- | --- |
| PUSH | | AX | |
|  | PUSH | CX |  |
|  | CALL | SETXYL | ;设置起始显示行列地址 |
|  | CALL | DisplayL | ;显示上半行数据 |
|  | POP | CX |  |
|  | POP | AX |  |
|  | INC | AL |  |
|  | CALL | SETXYL | ;设置起始显示行列地址 |
|  | CALL | DisplayL | ;显示下半行数据 |
|  | RET |  |  |
| DispL | ENDP |  |  |
| ;右半屏显示一个字节/字： | | AL-起始显示行序数X(0-7)； | AH-起始显示列序数Y(0-63)； SI-显示字 |
| 数据首地址 |  |  |  |
| ByteDisR | PROC | ENAR |  |
|  | MOV | CX, 8 | ；显示8个字节数据，用于显示一个英文/符号 |
|  | CALL | DispR |  |
|  | RET |  |  |
| ByteDisR | ENDP |  |  |
| WordDisR | PROC | NEAR |  |
|  | MOV | CX, 16 | ；显示16字节数据，用于显示一个汉字 |
|  | CALL | DispR |  |
|  | RET |  |  |
| WordDisR | ENDP |  |  |
| DispR | PROC | NEAR |  |
|  | PUSH | AX |  |
|  | PUSH | CX |  |
|  | CALL | SETXYR | ；设置起始显示行列地址 |
|  | CALL | DisplayR | ；显示上半行数据 |
|  | POP | CX |  |
|  | POP | AX |  |
|  | INC | AL |  |
|  | CALL | SETXYR | ；设置起始显示行列地址 |
|  | CALL | DisplayR | ；显示下半行数据 |
|  | RET |  |  |
| DispR | ENDP |  |  |

；显示图形

；显示左半屏一行图形,AL-X起始行序数(0-7), AH-Y起始列地址序数(0-63)

|  |  |  |  |
| --- | --- | --- | --- |
| LineDisL | PROC | NEAR | |
|  | MOV | CX, 64 |  |
|  | CALL | SETXYL | ;设置起始显示行列 |
|  | CALL | DisplayL | ;显示数据 |
|  | RET |  |  |
| LineDisL | ENDP |  |  |

;显示右半屏一行图形,AL-X起始行地址序数(0-7), AH-Y起始列地址序数(0-63)

|  |  |  |  |
| --- | --- | --- | --- |
| LineDisR | PROC | NEAR | |
|  | MOV | CX, 64 |  |
|  | CALL | SETXYR | ;设置起始显示行列 |
|  | CALL | DisplayR | ;显示数据 |

RET

LineDisR

ENDP

;基本控制

;显示左半屏数据，R7-显示数据个数

|  |  |  |  |
| --- | --- | --- | --- |
| DisplayL | PROC | NEAR | |
|  | LODSB |  |  |
|  | CALL | WRDataL | ;写左半屏数据 |
|  | LOOP | DisplayL |  |
|  | RET |  |  |
| DisplayL | ENDP |  |  |
| ；显示右半屏数据，R7-显示数据个数 | | |  |
| DisplayR | PROC | NEAR |  |
|  | LODSB |  |  |
|  | CALL | WRDataR | ;写左半屏数据 |

LOOP DisplayR

RET

DisplayR ENDP

；设置左半屏起始显示行列地址,AL-X起始行序数(0-7), AH-Y起始列序数(0-63)

|  |  |  |
| --- | --- | --- |
| SETXYL | PROC | NEAR |
|  | OR | AL, X ;行地址二行序数+行基址 |
|  | CALL | WRComL |
|  | MOV | AL, AH |
|  | OR | AL, Y ;列地址二列序数+列基址 |

CALL WRComL

RET

SETXYL ENDP

；设置右半屏起始显示行列地址,AL-X起始行序数(0-7), AH-Y起始列序数(0-63)

|  |  |  |
| --- | --- | --- |
| SETXYR | PROC | NEAR |
|  | OR | AL, X ;行地址二行序数+行基址 |
|  | CALL | WRComR |
|  | MOV | AL, AH |
|  | OR | AL, Y ;列地址二列序数+列基址 |

CALL WRComR

RET

SETXYR ENDP

；写左半屏控制指令，A-写入指令

|  |  |  |
| --- | --- | --- |
| WRComL | PROC | NEAR |
|  | MOV | DX, WR\_C0M\_AD\_L |
|  | OUT | DX, AL |
| WRComLl: | MOV | DX, RD\_BUSY\_AD |

IN AL, DX

TEST AL, 80H

;检查液晶显示是否处于忙状态

JNZ WRComLl

RET

WRComL ENDP

;写右半屏控制指令，A-写入指令

|  |  |  |  |
| --- | --- | --- | --- |
| WRComR | PROC  MOV  OUT | NEAR  DX, WR\_COM\_AD\_R  DX, AL | |
| WRComRl: | MOV | DX, RD\_BUSY\_AD |  |
|  | IN | AL, DX |  |
|  | TEST | AL, 80H | ;检查液晶显示是否处于忙状态 |
|  | JNZ | WRComRl |  |
|  | RET |  |  |
| WRComR | ENDP |  |  |
| ；写左半屏数据， | A-写入数据 |  |  |
| WRDataL | PROC | NEAR |  |
|  | MOV | DX, WR\_DATA\_AD\_L |  |
|  | OUT | DX, AL |  |
| WRDataLl: | MOV | DX, RD\_BUSY\_AD |  |
|  | IN | AL, DX |  |
|  | TEST | AL, 80H | ;检查液晶显示是否处于忙状态 |
|  | JNZ | WRDataLl |  |
|  | RET |  |  |
| WRDataL | ENDP |  |  |
| ;写右半屏数据， | A-写入数据 |  |  |
| WRDataR | PROC | NEAR |  |
|  | MOV | DX, WR\_DATA\_AD\_R |  |
|  | OUT | DX, AL |  |
| WRDataRl: | MOV | DX, RD\_BUSY\_AD |  |
|  | IN | AL, DX |  |
|  | TEST | AL, 80H | ;检查液晶显示是否处于忙状态 |
|  | JNZ | WRDataRl |  |
|  | RET |  |  |
| WRDataR | ENDP |  |  |
|  | END | START |  |

七、实验扩展及思考

实验内容：显示一幅图画,

进一步熟练液晶显示的操作。

实验十四**8237 DMA**传输实验

一、 实验目的与要求

了解8237的内部结构、工作原理；了解8237与8086的接口逻辑；掌握使用8237,实现 DMA传输数据。

复习本节实验内容，可尝试自行编写程序，做好实验准备工作，填写实验报告。

**二、 实验设备**

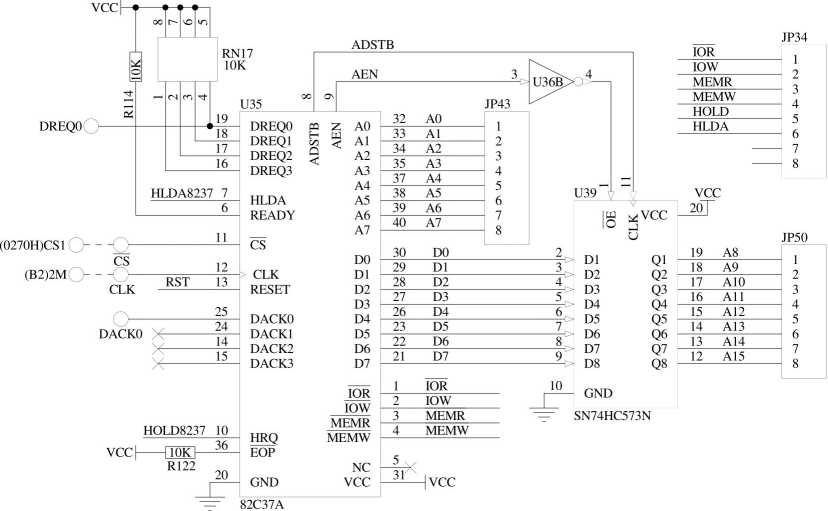
SUN系列实验仪一套、PC机一台

**三、 实验内容**

1、编制程序:将B4区SRAM中83000H〜837FFH内数据通过DMA方式传输给86000H-867FFH, 并对 83000H〜837FFH 与 86000H〜867FFH 作比较。

由于DMA传输期间，8237只能生成地址线A15. .A0,地址总线的高四位A19.. A16由实验仪 强制置位为“ 1000”，从而存贮器片选信号mCSl有效，选中B4区的16位SRAM。

**四、实验原理图**



**五、实验步骤**

1、连线说明:

|  |  |  |
| --- | --- | --- |
| B4 (RAM)区：CS、BLEs BHE |  | A3 区：mCSO、 BLE、 BHE |
| B4 (RAM)区:RD、WR |  | A3 区：MEMR> MEMW |
| B4 (RAM)区：JP49 (AO.. A7) |  | A3 区:JP29 (Al. . A8) |
| B4 (RAM)区：JP38 (A8.. A15) |  | A3 区:JP33(A9. . A16) |
| B4 (RAM)区：JP57 (DO.. D7) |  | A3 区：JP42 (DO. . D7) |
| B4 (RAM)区：JP56(D8. .D15) |  | A3 区:JP40 (D &.D15) |
| D4 区:CS |  | A3 区：CS1 |
| D4 区;CLK |  | B2 区：2M |
| D4 区:JP34 |  | A3 区：JP26 |

|  |  |  |
| --- | --- | --- |
| D4 区:JP43 (AO. . A7) |  | A3 区:JP28 (AO. . A7) |
| D4 区:JP50(A8. . A15) |  | A3 区:JP32 (A8. . A15) |

2、编写、调试程序

3、存贮器的83000H-837FFH与86000H〜867FFH二块数据是否完全相同

六、演示程序

DMAaddr

STAR:

STAR3:

|  |  |  |
| --- | --- | --- |
| .MODEL | TINY | |
| EQU | 0270H | ;8237基地址 |
| .STACK | 100 |  |
| .CODE |  |  |
| MOV | AX, 8000H |  |
| MOV | DS, AX |  |
| mov | si,3000h |  |
| mov | al, Offh |  |
| mov | ex, 0800h |  |
| mov | [si], al |  |
| inc | Si |  |
| dec | Al |  |
| loop | STAR3 |  |
| mov | al, 04h |  |
| mov | dx, DMAaddr+8 |  |
| out | dx, al | ;禁止DMA操作 |
| mov | al, 00 |  |
| mov | dx, DMAaddr+Odh |  |
| out | dx, al | ；复位 |
| mov | dx, DMAaddr+Och |  |
| out | dx, al | ；清除先/后寄存器 |
| mov | dx, DMAaddr+0 | ;源启始地址3000H |
| mov | al, 0 |  |
| out | dx, al |  |
| mov | al, 30h |  |
| out | dx, al |  |
| mov | dx, DMAaddr+Och |  |
| out | dx, al | ；清除先/后寄存器 |
| mov | dx, DMAaddr+2 | ；目的启始地址600( |
| mov | al, 0 |  |
| out | dx, al |  |
| mov | al, 60h |  |
| out | dx, al |  |
| mov | dx, DMAaddr+Och |  |
| out | dx, al | ；清除先/后寄存器 |
| mov | dx, DMAaddr+03h |  |
| mov | al, Offh | ;长度0800H |
| out | dx, al |  |
| mov | al, 07h |  |

|  |  |  |  |
| --- | --- | --- | --- |
| out | | dx, al | |
|  | mov | dx, DMAaddr+Obh |  |
|  | mov | al, 88h |  |
|  | out | dx, al | ;通道0方式字 |
|  | mov | al, 85h |  |
|  | out | dx, al | ;通道1方式字 |
|  | mov | dx,DMAaddr+8 |  |
|  | mov | al,41H |  |
|  | out | dx, al | ;允许8237工作、存贮器方式传送 |
| starl: | mov | dx, DMAaddr+Ofh |  |
|  | mov | al, Och |  |
|  | out | dx, al | ;允许通道0、通道1 |
|  | mov | dx, DMAaddr+09h |  |
|  | mov | al, 04h |  |
|  | out | dx, al | ;允许DMA操作 |
|  | mov  NOP  NOP | dx, DMAaddr+08h |  |
| star2: | in | al, dx |  |
|  | TEST | AL, 03H |  |
|  | jz | st ar 2 | ;等待DMA结束 |
|  | mov | dx, DMAaddr+Och |  |
|  | out | dx, al |  |
|  | mov | dx, DMAaddr+03h | ；清除先/后寄存器 |
|  | IN | AL, DX |  |
|  | MOV | AH, AL |  |
|  | IN | AL, DX |  |
|  | CMP | AX, OFFFFH |  |
|  | JNZ | st arl |  |
|  | mov | dx, DMAaddr+09h |  |
|  | mov | al, OOh |  |
|  | out | dx, al | ;清除DMA请求 |
|  | mov | dx, DMAaddr+08h |  |
|  | mov | al, 04h |  |
|  | out | dx, al | ;禁止DMA操作 |
|  | mov | si,3000h |  |
|  | mov | bx,6000h |  |
|  | mov | ex, 0800h |  |
| STAR5: | mov | al, [si] |  |
|  | cmp | al, ds: [bx] |  |
|  | jne | False |  |
|  | inc | si |  |
|  | inc | Bx |  |
|  | loop | STAR5 |  |

t rue:

jmp

jmp

false:

END STAR

七、实验扩展及思考

1、8237的DMA传输有多种工作方式，请尝试一下。

实验十五**RS485**通信实验

**一、实验目的与要求**

掌握RS485串行通讯；初步了解远程控制方法

二、实验设备

SUN系列实验仪二套、PC机二台。

三、 实验内容

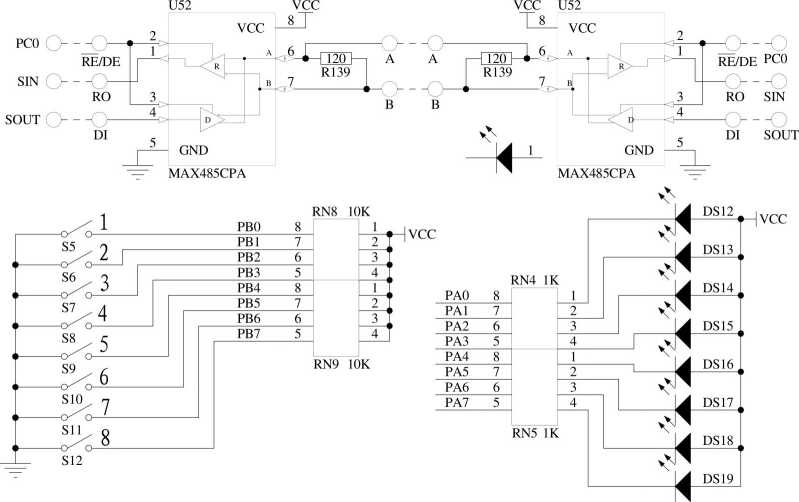
1、 RS485

1. 传输距离1.5km,最大传输率^2. 5Mbps
2. 半双工工作方式

2、 实验过程

1. 主机通过RS485发出数据给从机，等待接收从机回传的数据
2. 从机收到数据后，点亮相应的发光管，并回传数据给主机

**四、 实验原理图**



**五、实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS （8255）、AO、A1 |  | A3 区：CS1、AO、A1 |
| D3 区：JP23 （PA 口） |  | F4 区:JP18 |
| D3 区：JP20 （PB 口） |  | F4 区：JP27 |
| D3 区：PCO |  | E4 区：RE/DE |
| D5 区：CS、AO、Al、A2 |  | A3 区：CS4、AO、Al、A2 |
| D5 区：SIN、SOUT |  | E4 区:RO、DI |

从机连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS (8255)、 A0、 A1 |  | A3 区：CS1、A0、A1 |
| D3 区：JP23 (PA 口) |  | F4 区：JP18 |
| D3 区：PCO |  | E4 区：RE/DE |
| D5 区：CS、A0、Al、A2 |  | A3 区：CS4、A0、Al、A2 |
| D5 区：SIN、SOUT |  | E4 区：R0、DI |

2、运行程序，拨动F4区的拨动开关，主机读取拨动开关数据，发送给从机；从机接收到 数据后，点亮相应发光管，回传数据；主机接收到数据后，点亮相应发光管；一直循环。

六、演示程序

;主机程序(RS485\Send\Send\_485. ASM)

.MODEL TINY

;8250和PC机通信，需要在PC上运行一个串口软件，并设置与8250相同的波特率

| RHR | EQU | NS8250\_ | Base\_ | Address |
| --- | --- | --- | --- | --- |
| THR | EQU | NS8250 | Base | Address |
| IER | EQU | NS8250\_ | Base\_ | \_Address+l |
| FCR | EQU | NS8250 | Base | \_Address+2 |
| ISR | EQU | NS8250 | Base | \_Address+2 |
| LCR | EQU | NS8250\_ | Base\_ | Address+3 |
| MCR | EQU | NS8250 | Base | \_Address+4 |
| LSR | EQU | NS8250\_ | Base\_ | Address+5 |
| MSR | EQU | NS8250 | Base | \_Address+6 |
| DLL | EQU | NS8250\_ | Base\_ | Address |
| DLM | EQU | NS8250\_ | Base\_ | \_Address+l |

NS8250\_Base\_Address EQU 0240H

Address

Base

；接收数据缓冲区

；发送数据缓冲区

；中断控制寄存器

:FIFO控制寄存器

；中断状态寄存器

;串行口控制寄存器

;MODEM控制寄存器

；串行口状态寄存器

:MODEM状态寄存器

；波特率除数锁存器低位 ；波特率除数锁存器高位

NEC8255

NEC8255\_C0M\_ADDR

NEC8255\_PA\_ADDR

NEC8255\_PB\_ADDR

NEC8255 PC ADDR

EQU

EQU

EQU

EQU

EQU 0270H

NEC8255\_Base

NEC8255\_Base\_Address NEC8255\_Base\_Address NEC8255 Base Address

Address

；控制口

;PA 口 ;PB 口

;PC 口

・ STACK .DATA

Receive\_Buffer

SendBuffer EQU

100

10 DUP (0)

START:

.CODE

MOV

MOV

MOV

NOP

CALL

CALL

DB

Receive Buffer

；接受缓冲器

；发送缓冲器

AX, @DATA

DS, AX

ES, AX

INIT8250

INIT8255

| START2: CALL | Send\_Ready |
| --- | --- |
| MOV | DX, NEC8255\_PB\_ADDR |
| IN | AL, DX |
| CALL | Send\_Byte |
| JC | ERROR |
| CALL | Receive\_Ready |
| CALL | Receive\_Byte |
| MOV | DX, NEC8255\_PA\_ADDR |
| OUT | DX, AL |

;允许RS485发送

;读出PB 口(读拨动开关)

；发送一个字节

；发送出错

;允许RS485接收

；接收一个字节

;通过PA 口，将接收到数据显示于发光二极管

JMP START2

ERROR: JMP $

| INIT8255 PROC | NEAR |
| --- | --- |
| MOV | DX, NEC8255\_C0M\_ADDR |
| MOV | AL, 82H ;PA、PC 输出，PB 输入 |
| OUT | DX, AL |
| RET |  |
| INIT8255 ENDP |  |

Send\_Ready PROC NEAR

MOV DX, NEC8255\_PC\_ADDR

MOV AL, OFFH

OUT DX, AL

MOV CX, 1OOH

LOOP $

;8255的PCO控制RS485收发

;允许RS485发送

;等待RS485稳定

|  |  |  |
| --- | --- | --- |
| INIT8250 PROC | NEAR | |
| MOV | DX, ISR |  |
| MOV | AL, 06H |  |
| OUT | DX, AL |  |
| MOV | DX, LCR |  |
| MOV | AL, 83H | ;允许访问波特率因子寄存器 |
| OUT | DX, AL |  |
| MOV | DX, DLL |  |
| MOV | AL, 40 | ；除数低位寄存器，波特率设为**4800=(3. 072\*1000000/16)/DLMDLL** |
| OUT | DX, AL |  |
| MOV | DX, DLM | ;00H送高字节寄存器 |
| MOV | AL, OOH |  |
| OUT | DX, AL |  |
| MOV | DX, LCR | ；不允许访问波特率因子寄存器 |
| MOV | AL, 1BH | ；数据格式为8数据位,1停止位，偶校验 |
| OUT | DX, AL |  |
| RET |  |  |
| INIT8250 ENDP |  |  |

RET

| ;\*\*\*\*\*\*\*\*\*发送一个字节子程序, | |
| --- | --- |
| Send\_Byte PROC | NEAR |
| PUSH | CX |
| PUSH | AX |
| MOV | CX,1000 |
| MOV | DX, LSR |

Send\_Ready ENDP

|  |  |  |
| --- | --- | --- |
| MOV | DX, LSR | |
| Receive\_Readyl: |  |  |
| \_IN | AL, DX |  |
| TEST | AL, 40H |  |
| JZ | Receive\_Readyl | ;发送寄存器、移位寄存器均空 |
| MOV | DX, NEC8255 PC ADDR |  |
| MOV | AL,OFEH |  |
| OUT | DX, AL | ;允许RS485接收 |
| MOV | CX, 80H |  |
| LOOP | $ | ;等待RS485稳定 |
| RET |  |  |
| Receive\_Ready | ENDP |  |

Receive\_Ready PROC NEAR

发送A中的数，失败置ICY\*\*\*\*\*\*\*\*\*

|  |  |
| --- | --- |
| REP11: IN  TEST JNZ  LOOP  POP  STC  JMP  OUTPORT1:POP  MOV  OUT  CLC  EXIT8250:P0P  RET | AL, DX  AL, 20H ;发送寄存器空，有可能移位寄存器不空  OUTPORT1  REP11  AX  EXIT8250  AX  DX, RHR  DX, AL  CX |

Send\_Byte ENDP

；\*\*\*\*\*\*\*\*\*接收一个字节子程序，接收字节在a中，接收失败置1CY\*\*\*\*\*\*\*\*\*

Receive\_Byte PROC NEAR

|  |  |
| --- | --- |
| MOV | DX, LSR |
| Receivel:IN TEST JZ  Receive2:M0V  IN  Receive3:RET | AL, DX  AL, 1  Receivel  DX, RHR  AL, DX |

Receive\_Byte ENDP

END START

从机程序(RS485\Receive\Rece\_485. ASM)

;从机的主程序与主机不同，其它完全一样

|  |  |
| --- | --- |
| START: MOV  MOV  MOV  NOP  CALL  CALL | AX, @DATA  DS, AX  ES, AX  INIT8250  INIT8255 |
| START2: CALL  CALL  MOV  OUT  PUSH  CALL  POP  CALL  JC  JMP  ERROR: JMP | Receive\_Ready ;允许 RS485 接收  Receive\_Byte ;接皈一个字节  DX, NEC8255 PA ADDR  DX, AL ；通过PA 口，将接收到数据显示于发光二极管  AX  Send\_Ready ;允许 RS485 发送  AX  Send Byte ;回发一个字节  ERROR ；发送出错  START2  $ |

**七、实验扩展及思考**

1、 RS485通信如何实现既接收又发送的？

2、 在掌握RS485串行通讯和基本的远程控制方法的基础上，进行其他方面的远程控制：如 远程控制电机转速、语音、温度测量、显示等等，可以尽情发挥自己的想法。

实验十六图形点阵显示实验(12864M)

一、 实验目的与要求

了解图形液晶模块的控制方法；了解它与8086的接口逻辑；掌握使用图形点阵液晶显示字 体和图形。

二、 实验设备

SUN系列实验仪一套、PC机一台。

三、 实验内容

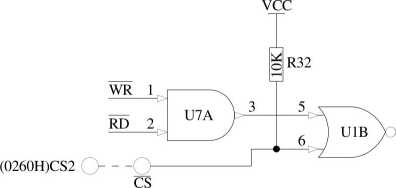
1> 12864M液晶显示器

1. 图形点阵液晶显示器，分辨率为128X64„可显示图形和8X4个(16X16点阵)汉字。
2. 与CPU连接，可采用并行或串行接口方式。
3. 内置8192个中文汉字(16\*16点阵)、128个字符、64 \* 256点阵显示RAM。

2、实验过程

使用并行接口方式，在12864M液晶上画一个矩形，显示一段字，包括汉字和英文：“星研 电子”、"STAR ES59PA”。

四、 实验原理图



VCC 二

V0 3

RS 4

6

A

JP2

(Cl)VCC，——-

CS1/2

JP6

| DO | 7 |
| --- | --- |
| DI | 8 |
| D2 9 | |
| D3 | 10 |
| D4 | 11 |
| D5 | *12* |
| D6 | 13 |
| D7 | 14 |

1. "

16

VSSVDDVORSRAVEDBODB1DB2DB3DB4DB5DB6DB7PSBNCRST

18

19

20

VEE

LED1

LED2

vcc宁

12864M

**五、实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| Al 区：CS、RW, RS |  | A3 区：CS1、AO、Al |
| Al 区:CS1/2 (PSB) |  | Cl 区：VCC |

2、运行程序，验证显示结果。

**六、演示程序(完整程序见目录12864M)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1、 12864M | | | | |
| #define | Y12864 | \_W\_CON | 0x270 | //写指令地址 |
| #define | Y12864 | \_R\_CON | 0x271 | //读取忙状态地址 |
| ^define | Y12864 | W Data | 0x272 | //写数据地址 |

ftdefine Y12864\_R\_Data 0x273

〃读数据地址

extern void outportb( unsigned int, char); //写 I/O

extern char inportb( unsigned int ); //读 I/O

//写指令

void WRCon(u8 data)

{

outportb(Y12864WC0N, data):

wh订e (inportb(Y12864\_R\_CON) & 0x80) {:}

//写控制命令

//检查液晶显示是否处于忙状态

u8 RD Data ()

{

return inportb(Y12864\_R\_Data):

}

//写数据

void WRData(u8 data)

{

outportb(Y12864\_W\_Data, data):

wh订e (inportb(Y12864\_R\_CON) & 0x80) {:}

//写数据

//检查液晶显示是否处于忙状态

//液晶初始化 void initLCD()

{

WR\_Con(0x30);

WR\_Con(0x04);

WR\_Con(0x0f);

WR\_Con(0x0c);

WR\_Con(0x01);

asm nop

WR\_Con(0x02);

//不调用扩充指令

//点设定，游标向右移

//开光标

//关光标

//清屏

//地址归位，光标移到第1行第1列

//清屏

void Clear\_LCD()

{

WR\_Con(0x01);

}

/ /关光标

void Close Cursor ()

{

WR\_Con(0x0c);

}

〃设置光标pos—光标位置

void SetCursor(u8 pos)

{

WR\_Con(0x0e);

WR\_Con(pos);

}

//基本模式

void BaseMode()

{

WR\_Con(0x30);

}

〃扩展模式+允许绘图

void ExpandModeDrawOn()

{

WR\_Con(0x36);

}

//扩展模式+关闭绘图

void ExpandModeDrawOff()

{

WR\_Con(0x34);

}

//显示-行

void Disp\_Line(u8 pos, u8 \*pBuffer)

{

BaseMode ();

WR\_Con(pos) : //定位，第一个数据显示的位置

wh订e(\*pBuffer) //判断是否到了显示结束标志

{

WR\_Data(\*pBuffer++);

//设置DDRAM、GDRAM地址到地址计数器，参数在pos中 void Set\_DDRAM\_Addr(u8 pos)

WR\_Con(pos | 0x80);

void Set\_GDRAM\_Addr(u8 pos) {

WR\_Con(pos | 0x80);

//写GDRAM(一次二个字节)

//first:第一个字节，second:第二个字节，col:X, row: Y void Write\_GDRAM\_X\_Y(u8 col, u8 row, u8 first, u8 second)

Set\_GDRAM\_Addr(col & Oxdf); //行

if (col & 0x20)

Set\_GDRAM\_Addr(row | 0x8); //列

else

Set\_GDRAM\_Addr(row); //列

WR\_Data(first);

WR\_Data(second);

〃读GDRAM(一次二个字节)

//col:X, row:Y, pFirst:第一个字节，pSecond:第二个字节

void Read\_GDRAM\_X\_Y(u8 col, u8 row, u8\* pFirst, u8\* pSecond) {

Set\_GDRAM\_Addr(col & Oxdf); //行

if (col & 0x20)

Set\_GDRAM\_Addr(row | 0x8); //列

else

Set\_GDRAM\_Addr(row); //列

RD DataO ;

\*pFirst 二 RD Data (); \*pSecond 二 RDData();

//画一幅图画(128X64), pBuffer指向数据区 void Draw\_A\_Picture(u8\* pBuffer)

{

u8 i, j;

u8 il,i2;

ExpandModeDraw\_0ff();

i 二 0;

do

j = 0；

do

{

1. 二 \*pBuffer++;
2. = \*pBuffer++;

Write\_GDRAM\_X\_Y(i, j, il, i2);

}while(++j != 8);

}wh订e(++i != 64);

ExpandModeDraw On();

}

3、因主程序、其它子程序较长，请您参阅光盘上完整程序或自己编写：

Draw\_one\_row:画一条竖线子程序 A: Y F0 = 0,清除；F0 = 1,画一条竖线

Draw\_one\_line:画一条横线子程序 A: X F0 = 0,清除；F0 = 1,画一条横线

Draw\_0ne\_dot: 画一个点子程序 R6: X, R7: Y F0 = 0 :清除，F0 二 1 :画一个点 Fill\_Y12864: 填充整个液晶屏 CY二0,清屏；CY二1,全屏显示

实验十七继电器控制实验

一、 实验目的与要求

掌握使用继电器控制外设的基本方法和编程。

二、 实验设备

SUN系列实验仪一套、PC机一台。

三、 实验内容

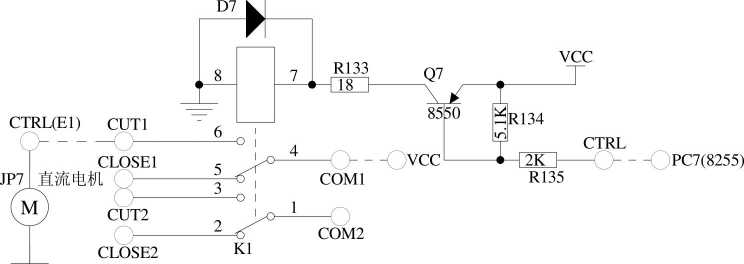
1、 预备知识

自动化控制设备中，存在一个电子与电气电路的互连问题，一方面，电子电路需要控制电 气电路的执行元件，如：电动机、电磁铁、电灯等；一方面又要为电子电路提供良好的电隔离, 以保护电子电路和人身安全，减少干扰源。继电器就起这个桥梁作用。

2、 实验过程

使用F4区的拨动开关，通过继电器控制直流电机转动、停止。

**四、实验原理图**



五、实验步骤

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS、AO、A1 |  | A3 区：CS1, AO、A1 |
| D3 区：JP23 (8255) |  | F4 区:JP27 |
| D3 区：PC7 |  | E3 区：CTRL |
| E3 区：C0M1 |  | Cl 区：VCC |
| E3 区：CUT1 |  | El 区：CTRL |

2、运行程序，F4区的K1拨动开关控制直流电机转动、停止。

**六、演示程序**

|  |  |  |
| --- | --- | --- |
|  | ・MODEL | TINY |
| C0N\_8255 | EQU | 0273H |
| PA\_8255 | EQU | 0270H |
|  | .STACK | 100 |
|  | .CODE |  |
| START: | CALL | INIT8255 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | CALL | ReadPA | ;读戸尤0 |
| START1: | MOV | BL, AL |  |
|  | CMP | AL, 0 |  |
|  | JZ | START2 |  |
|  | CALL | hOutput | ;输出高电平 |
|  | JMP | START3 |  |
| START2: | CALL | lOutput | ;输出低电平 |
| START3: | CALL | DLTime |  |
|  | CALL | ReadPA | ;读戸尤0 |
|  | CMP | AL, BL |  |
|  | JNZ | START1 | ;拨动开关是否变化 |
|  | JMP | START3 |  |
| hOutput | PROC | NEAR | ;输出高电平 |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, OFH |  |
|  | OUT | DX, AL |  |
|  | RET |  |  |
| hOutput | ENDP |  |  |
| lOutput | PROC | NEAR | ;输出低电平 |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, OEH |  |
|  | OUT | DX, AL |  |
|  | RET |  |  |
| lOutput | ENDP |  |  |
| INIT8255 | PROC | NEAR | ；8255初始化 |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, 90H | ;8255. PA. 0 -输入，8255. PC. 7-输出 |
|  | OUT | DX, AL |  |
|  | RET |  |  |
| INIT8255 | ENDP |  |  |
| ReadPA | PROC | NEAR | ；读卩九0 |
|  | MOV | DX, PA\_8255 |  |
|  | IN | AL, DX |  |
|  | AND | AL, O1H |  |
|  | RET |  |  |
| ReadPA | ENDP |  |  |
| DLTime | PROC | NEAR | ;延时及检测开关状态 |
|  | MOV | CX, 1OOOO |  |
|  | LOOP | $ |  |
|  | RET |  |  |
| DLTime | ENDP |  |  |
|  | END | START |  |

*(b*综合实验

实验一简易电子琴实验

一、 实验目的与要求

掌握蜂鸣器的使用方法；掌握蜂鸣器的不同发音的方法。

二、 实验设备

SUN系列实验仪一套、PC机一台。

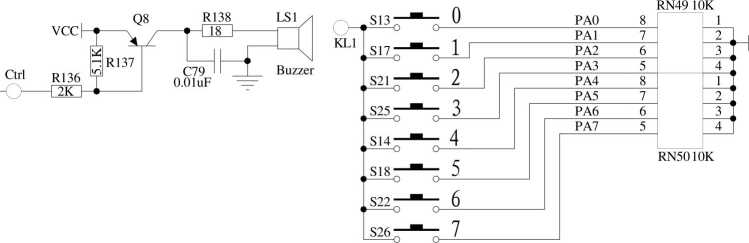
三、 实验内容

1、 简易电子琴原理：

1. 蜂鸣器输入不同频率的方波，会发出不同的声音；
2. 通过按键，由CPU控制产生不同频率的方波，从而发出不同的声音。
3. 实验过程

(1)通过8255的PAD,使F5区的1〜7号键由低到高发出1-7的音阶。

四、实验原理图



(8255)PC7

**五、实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS、AO、Al |  | A3 区：CS1、AO、A1 |
| D3 区：PC7 |  | F8 区：Ctrl |
| D3 区：JP23 (PA 口) |  | F5 区：JP37 (A) |
| Cl 区：GND |  | F5 区：KL1 |
| 2、运行程序，按F5区的1〜7号铤 | | L输出7种音阶 |

3、使用F5区的1〜7号键，弹一首《生日快乐》。

六、演示程序

C8255

.MODEL

EQU

TINY

0273H

|  |  |  |  |
| --- | --- | --- | --- |
| PA8255 | EQU  .STACK  .DATA | 0270H  100 | |
| Music | DW Ml, M2, M3, M4, M5, M6, M7, M7, M7, M6, M5, M4, M3, M2, Ml | | |
|  | DW Ml, M2, Ml, M2, M3, M2, M3, M4, M3, M4, M5, M4, M5, M6, M5 | | |
|  | DW M6, M7, M6, M7, M7, M6, M6, M6 | |  |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | CALL | INIT8255 | ;8255初始化 |
|  | CALL | Demo | ；播放…段音乐 |
| START1: | MOV | DX, PA8255 | ；按键查询 |
|  | IN | AL, DX | ；读键值 |
|  | CMP | AL, OFFH |  |
|  | JZ | START1 | ；无键 |
|  | XOR | AL,OFFH | ；有键 |
|  | TEST | AL, 2 |  |
|  | JZ | START2 |  |
|  | CALL | Musicl | ；1号键，调1号键输出 |
|  | JMP | START1 |  |
| START2: | TEST | AL, 4 |  |
|  | JZ | START3 |  |
|  | CALL | Music2 | ；2号键 |
|  | JMP | START1 |  |
| START3: | TEST | AL, 8 |  |
|  | JZ | START4 |  |
|  | CALL | Music3 | ；3号键 |
|  | JMP | START1 |  |
| START4: | TEST | AL, 10H |  |
|  | JZ | START5 |  |
|  | CALL | Music4 | ；4号键 |
|  | JMP | START1 |  |
| START5: | TEST | AL, 20H |  |
|  | JZ | START6 |  |
|  | CALL | Music5 | ；5号键 |
|  | JMP | START1 |  |
| START6: | TEST | AL, 40H |  |
|  | JZ | START7 |  |
|  | CALL | Music6 | ；6号键 |
|  | JMP | START1 |  |
| START7: | TEST | AL, 80H |  |
|  | JZ | START1 |  |
|  | CALL | Music7 | ；7号键 |
|  | JMP | START1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Demo | PROC | NEAR |  |
|  | MOV | CX, 38 | ；共38拍 |
|  | LEA | BX, Music |  |
| DemolO: | PUSH | CX |  |
|  | CALL | [BX] | ;播放该音调声音 |

INC BX

INC BX

POP CX

LOOP Demo10

RET

Demo ENDP

；节拍1（手动按键时用）

|  |  |  |  |
| --- | --- | --- | --- |
| Musicl | PROC | NEAR | |
|  | CALL | W\_L | ；写0,蜂鸣器响 |
|  | CALL | T10 | ;延时100us |
|  | CALL | T10 |  |
|  | CALL | T5 | ;延时50us |
|  | CALL | T1 | ;延时10us |
|  | CALL | T1 | ;延时10us |
|  | CALL | W\_H | ；写1,蜂鸣器不响 |
|  | CALL | T10 | ；延时 |
|  | CALL | T10 |  |
|  | CALL | T5 |  |
|  | CALL | T1 | ;延时10us |
|  | CALL | T1 | ;延时10us |
|  | RET |  |  |
| Musicl | ENDP |  |  |
| ;节拍2,同上 |  |  |  |
| Music2 | PROC | NEAR |  |
|  | CALL | W\_L |  |
|  | CALL | T10 |  |
|  | CALL | T10 |  |
|  | CALL | T5 |  |
|  | CALL | T1 |  |
|  | CALL | W\_H |  |
|  | CALL | T10 |  |
|  | CALL | T10 |  |
|  | CALL | T2 |  |
|  | CALL | T2 |  |
|  | CALL | T1 |  |
|  | RET |  |  |
| Music2 | ENDP |  |  |
| ;节拍3,同上 |  |  |  |
| Music3 | PROC | NEAR |  |

|  |  |  |
| --- | --- | --- |
| CALL | | W\_L |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T2 |
|  | CALL | T2 |
|  | CALL | W\_H |
|  | CALL | T10 |
|  | CALL | T1O |
|  | CALL | T2 |
|  | CALL | T2 |
| Music3 | RET  ENDP |  |
| ;节拍4,同上 |  | |
| Music4 | PROC | NEAR |
|  | CALL | W\_L |
|  | CALL | T1O |
|  | CALL | T1O |
|  | CALL | T2 |
|  | NOP  NOP  NOP  NOP  CALL | W\_H |
|  | CALL | T1O |
|  | CALL | T1O |
|  | CALL | T2 |
| Music4 | NOP  NOP  NOP  NOP  RET  ENDP |  |
| ;节拍5,同上 |  | |
| Music5 | PROC | NEAR |
|  | CALL | W\_L |
|  | CALL | T1O |
|  | CALL | T1O |
|  | CALL | T1 |
|  | CALL | W\_H |
|  | CALL | T1O |
|  | CALL | T1O |
|  | CALL | T1 |
| Music5 | RET  ENDP |  |

|  |  |
| --- | --- |
| Music6 | PROC |
|  | CALL  CALL  CALL  CALL  CALL  CALL  NOP  NOP  CALL  CALL  CALL  CALL  CALL  CALL  NOP  NOP  RET |
| Music6  ；节拍7,同上 | ENDP |
| Music7 | PROC  CALL  CALL  CALL  CALL  CALL  CALL  CALL  CALL  CALL  CALL  RET |

;节拍6,同上

Music7

ENDP

NEAR

W\_L

T10

T5

T2

T2

T1

W\_H

T10

T5

T2

T2

T1

NEAR

W\_L

T10

T5

T2

T2

W\_H

T10

T5

T2

T1

|  |  |  |
| --- | --- | --- |
| Ml | PROC | NEAR |
|  | MOV | CX, 1100 |
| MIO: | CALL | W\_L |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T5 |
|  | CALL | T2 |
|  | CALL | T2 |
|  | NOP |  |

；节拍1（自动放音时用，时间约0. 2s）

|  |  |  |
| --- | --- | --- |
| NOP | | |
|  | NOP | |
| LOOP | Mil |
| Mil: | CALL | W\_H |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T5 |
|  | CALL | T2 |
|  | CALL | T1 |
|  | NOP  NOP  NOP  LOOP | MIO |
| Ml | RET  ENDP |  |
| ;节拍2,同上 |  | |
| M2 | PROC | NEAR |
|  | MOV | CX, 1150 |
| M20: | CALL | W\_L |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T2 |
|  | CALL | T2 |
|  | CALL | T1 |
|  | CALL | T1 |
|  | LOOP | M21 |
| M21: | CALL | W\_H |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T2 |
|  | CALL | T2 |
|  | CALL | T1 |
|  | LOOP | M20 |
| M2 | RET  ENDP |  |
| ;节拍3,同上 |  | |
| M3 | PROC | NEAR |
|  | MOV | CX, 1200 |
| M30: | CALL | W\_L |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T5 |
|  | LOOP | M31 |
| M31: | CALL | W\_H |

|  |  |  |
| --- | --- | --- |
| CALL | | T10 |
|  | CALL | T1O |
|  | CALL | T2 |
|  | CALL | T2 |
|  | NOP  NOP  LOOP | M30 |
| M3 | RET  ENDP |  |
| ;节拍4,同上 |  | |
| M4 | PROC | NEAR |
|  | MOV | CX, 1250 |
| M40: | CALL | W\_L |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T2 |
|  | CALL | T1 |
|  | PUSH | AX |
|  | POP | AX |
|  | PUSH | AX |
|  | POP | AX |
|  | LOOP | M41 |
| M41: | CALL | W\_H |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T2 |
|  | CALL | T1 |
|  | PUSH | AX |
|  | POP | AX |
|  | PUSH | AX |
|  | POP | AX |
|  | NOP  NOP  LOOP | M40 |
| M4 | RET  ENDP |  |
| ;节拍5,同上 |  | |
| M5 | PROC | NEAR |
|  | MOV | CX, 1300 |
| M50: | CALL | W\_L |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T2 |
|  | CALL | T1 |

|  |  |  |
| --- | --- | --- |
| PUSH | | AX |
|  | POP | AX |
|  | LOOP | M51 |
| M51: | CALL | W\_H |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T2 |
|  | NOP  NOP  NOP  NOP  LOOP | M50 |
| M5 | RET  ENDP |  |
| ；节拍6,同上 |  | |
| M6 | PROC | NEAR |
|  | MOV | CX, 1350 |
| M60: | CALL | W\_L |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T1 |
|  | NOP  NOP  LOOP | M61 |
| M61: | CALL | W\_H |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T2 |
|  | PUSH | AX |
|  | POP | AX |
|  | LOOP | M60 |
| M6 | RET  ENDP |  |
| ;节拍7,同上 |  | |
| M7 | PROC | NEAR |
|  | MOV | CX, 1420 |
| M70: | CALL | W\_L |
|  | CALL | T10 |
|  | CALL | T10 |
|  | CALL | T1 |
|  | LOOP | M71 |
| M71: | CALL | W\_H |
|  | CALL | T10 |
|  | CALL | T10 |

|  |  |  |
| --- | --- | --- |
| CALL | | T1 |
|  | LOOP | M70 |
|  | RET |  |
| M7 | ENDP |  |
| ;写 0(8255. PC. 7=0) | |  |
| W\_L | PROC | NEAR |

MOV DX, C8255

MOV AL, OEH

OUT DX, AL

RET

W\_L ENDP

;写 1 (8255. PC. 7=1)

|  |  |  |
| --- | --- | --- |
| W\_H | PROC | NEAR |
|  | MOV | DX, C8255 |
|  | MOV | AL, OFH |
|  | OUT | DX, AL |
| W\_H | RET  ENDP |  |
| ；8255初始化 |  | |
| INIT8255 | PROC | NEAR |
|  | MOV | DX, C8255 |
|  | MOV | AL, 90H |
|  | OUT | DX, AL |
|  | MOV | DX, C8255 |
|  | MOV | AL, OFH |
|  | OUT | DX, AL |
| INIT8255 | RET  ENDP |  |
| ；延时lOus |  | |
| T1 | PROC | NEAR |
| T1 | RET  ENDP |  |
| ;延时20us |  | |
| T2 | PROC | NEAR |
|  | CALL | T1 |
| T2 | RET  ENDP |  |
| ;延时50us |  | |
| T5 | PROC | NEAR |
|  | CALL | T2 |
|  | CALL | T2 |
| T5 | RET  ENDP |  |
| ;延时100s |  | |

；PC.7输出,PA输入

|  |  |  |
| --- | --- | --- |
| T10 | PROC | NEAR |
|  | CALL | T2 |
|  | CALL | T2 |
|  | CALL | T5 |
|  | RET |  |
| T10 | ENDP |  |
|  | END | START |

1. 实验扩展及思考题

设计一个简易电子播放器实验程序，使用蜂鸣器，回放…段音乐。

实验二LED16 \* 16点阵实验

一、实验目的与要求

1、 熟悉8255的功能，了解点阵显示的原理及控制方法；

2、 学会使用LED点阵，通过编程显示不同字符；

3、 认真预习本节实验内容，可尝试自行编写程序，做好实验准备工作，填写实验报告。

二、 实验设备

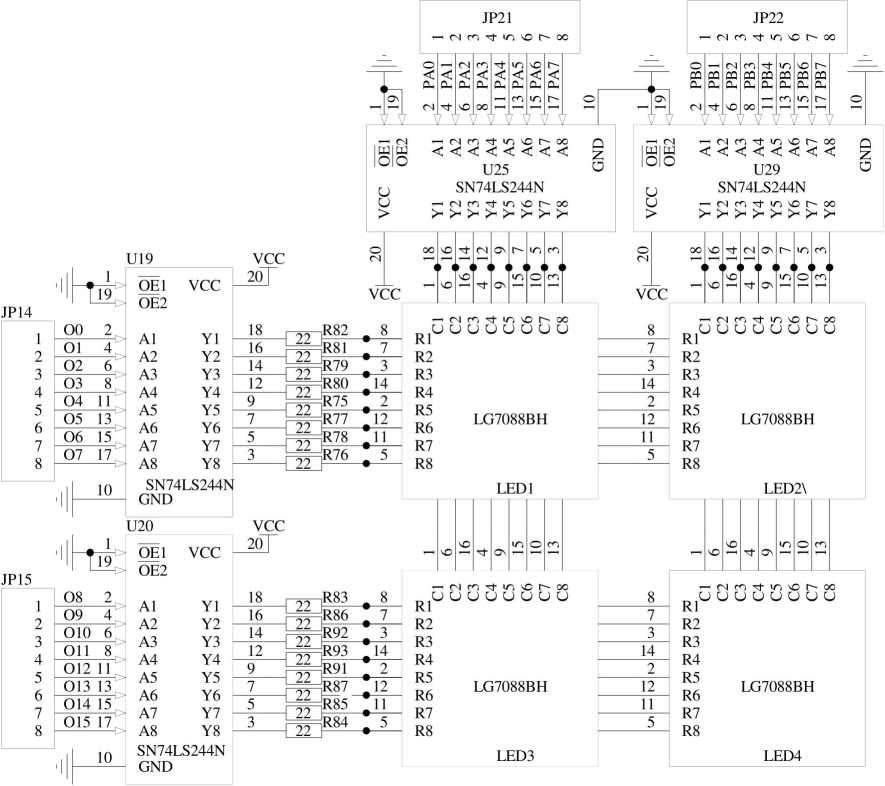
SUN系列实验仪一套、PC机一台。

三、 实验内容

1、 编写程序，用B4区的二片74HC273控制16X16点阵的行；8255的PA、PB 口控制16X16 点阵的列；显示字符。

2、 按图连接线路；运行程序，观察实验结果，学会编程控制LED点阵显示字符。

**四、实验原理图**



**五、实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS (8255)、 AO、 Al |  | A3 区：CS1、AO、Al |
| D3 区:JP23(PA)、JP20 (PB) |  | A2区：JP21、JP22 （列输出线） |
| B4 区:JP57 (DO. . D7) |  | A3 区:JP42 (DO.. D7) |
| B4 区:JP56 (D &.D15) |  | A3 区：JP40(D&.D15) |
| B4(I/0)区：CS273、BLE、BHE |  | A3 区：CS5、BLE、BHE |
| B4(I/0)区：RD、WR |  | A3 区：I OR、IOW |
| B4(I/0)区：JP51、JP55 |  | A2区：JP14、JP15 （行输出线） |

**（注意连线方向）**

2、运行程序，观察实验结果。运行演示程序将会看到字符“欢迎使用星研实验仪”在点阵 上自下而上循环移动显示。

|  |  |  |
| --- | --- | --- |
|  | ・MODEL | TINY |
| ADDR\_8255\_PA | EQU | 270H ;8255 PA口 |
| ADDR\_8255\_PB | EQU | 271H ;8255 PB口 |
| ADDR\_8255\_C | EQU | 273H ;8255控制口 |
| ADDR\_273 | EQU | 230H ;I0区74HC273(16位I/O) |
| LINE | EQU | ADDR\_273 ；行线1,行线2 |
| R0W1 | EQU | ADDR\_8255\_PA ;列线］ |
| R0W2 | EQU | ADDR\_8255\_PB ;列线2 |
|  | .STACK | 100 |
|  | .DATA |  |

**六、演示程序**

HUAN

YING

SHI

YONG

XING

YAN

SHIO

YANO

YI

NONE

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

DB

START:

OOH, OCOH, OOH, OCOH, OFEH, OCOH, 07H, OFFH, 0C7H, 86H, 6FH, 6CH, 3CH, 60H, 18H, 60H 1CH, 60H, 1CH, 70H, 36H, OFOH, 36H, 0D8H, 61H, 9CH, 0C7H, OFH, 3CH, 06H, OOH, OOH 60H, OOH, 31H, OCOH, 3FH, 7EH, 36H, 66H, 06H, 66H, 06H, 66H, 0F6H, 66H, 36H, 66H 37H, 0E6H, 37H, 7EH, 36H, 6CH, 30H, 60H, 30H, 60H, 78H, OOH, OCFH, OFFH, OOH, OOH OOH, OOH, 06H, 30H, 07H, 30H, OFH, OFFH, OCH, 30H, 1FH, OFFH, 3BH, 33H, 7BH, 33H 1BH, OFFH, 1BH, 33H, 19H, OBOH, 18H, OEOH, 18H, 60H, 18H, OFCH, 19H, 8FH, 1FH, 03H 00, 0, 1FH, OFEH, 18H, 0C6H, 18H, 0C6H, 18H, 0C6H, 1FH, OFEH, 018H, 0C6H, 18H, 0C6H 18H, 0C6H, 1FH, OFEH, 18H, 0C6H, 18H, 0C6H, 30H, 0C6H, 30H, 0C6H, 60H, ODEH, OCOH, OCCH

OOH, OOH, 1FH, OFCH, 18H, OCH, 1FH, OFCH, 18H, OCH, 1FH, OFCH, 01H, 80H, 19H, 80H 1FH, OFEH, 31H, 80H, 31H, 80H, 6FH, OFCH, 01H, 80H, 01H, 80H, 7FH, OFFH, OOH, OOH 0, 0, OFFH, OFFH, 18H, OCCH, 18H, OCCH, 30H, OCCH, 30H, OCCH, 7FH, OFFH, 7CH, OCCH OFCH, OCCH, 3CH, OCCH, 3CH, OCCH, 3DH, 8CH, 3DH, 8CH, 33H, OCH, 06H, OCH, OCH, OCH 01H, 80H, OOH, OCOH, 3FH, OFFH, 3CH, 06H, 67H, OCCH, 06H, OCOH, OCH, OCOH, 07H, OCOH 06H, OCOH, 7FH, OFFH, OOH, OCOH, 01H, OEOH, 03H, 30H, 06H, 18H, 1CH, 1CH, 70H, 18H OOH, OOH, OFCH, 60H, OCH, 60H, 6CH, OFOH, 6CH, 0D8H, 6DH, 8FH, 6FH, 0F8H, 7EH, OOH 06H, 0C6H, 07H, 66H, 3FH, OECH, 0E7H, OECH, 06H, 18H, 1FH, OFFH, OCH, OOH, OOH, OOH OCH, OCOH, OCH, 60H, 18H, 7CH, 1BH, 6CH, 33H, OCH, 73H, 18H, 0F1H, 98H, 31H, 98H 30H, OFOH, 30H, OFOH, 30H, 60H, 30H, OFOH, 31H, 98H, 33H, OFH, 3EH, 06H, 30H, OOH OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH, OOH

.CODE

MOV

AX, @DATA

MOV DS, AX

MOV ES, AX

NOP

CALL INIT\_IO

CALL TEST\_LED

；调用测试子程序，测试LED是否全亮

CALL CLEAR

；滚动显示多个字符

|  |  |  |
| --- | --- | --- |
| CHS\_SHOW: | MOV | CX, 9 |
|  | LEA | SI, HUAN |
| CHS\_1: | PUSH | CX |
|  | MOV | CX, 16 |
| CHS\_2: | CALL | DISP\_CH |

INC SI

INC SI

LOOP CHS\_2

POP CX

LOOP CHS\_1

JMP CHS\_SHOW

;显示一个16\*16点阵字子程序，字型码放在DPTR指出的地址

DISP\_CH

DISP\_CH\_1:

| PROC | NEAR |
| --- | --- |
| PUSH | CX |
| MOV | CX, 8 |
| CALL | DISP1 |
| LOOP | DISP\_CH\_1 |
| POP | CX |
| RET |  |

DISP\_CH ENDP

;显示一个16\*16点阵字子程序，字型码放在显示缓冲区XBUFF

|  |  |  |  |
| --- | --- | --- | --- |
| DISP1 | PROC | NEAR | |
|  | PUSH | SI |  |
|  | PUSH | CX |  |
|  | MOV | CX, 16 | ；计数器,16列依次被扫描 |
|  | MOV | BL, OFEH | ；上边列输出值 |
|  | MOV | BH, OFFH | ;下边列输出值 |
| REPEAT: | MOV | DX, LINE |  |
|  | MOV | AX, BX |  |
|  | OUT | DX, AX | ；列输出 |
|  | LODSB |  |  |
|  | CALL | ADJUST | ；调整AL,将AL中二进制数旋转180度 |
|  | MOV | DX, ROW1 |  |
|  | OUT | DX, AL | ；左边行输出 |
|  | LODSB |  |  |
|  | CALL | ADJUST | ；调整AL,将AL中二进制数旋转180度 |
|  | MOV | DX, ROW2 |  |

DISP1

INIT\_IO

INIT\_IO

CLEAR

CLEAR

|  |  |
| --- | --- |
| OUT | DX, AL |
| CALL | DL1OMS |
| CALL | CLEAR |
| STC |  |
| RCL | BL, 1 |
| RCL | BH, 1 |
| LOOP | REPEAT |
| POP | CX |
| POP | SI |
| RET |  |
| ENDP |  |
| PROC | NEAR |
| MOV | DX, ADDR\_8255\_C |
| MOV | AL, 80H |
| OUT | DX, AL |
| RET |  |
| ENDP |  |
| PROC | NEAR |
| MOV | AX, OFFFFH |
| MOV | DX, LINE |
| OUT | DX, AX |
| MOV | AL, 0 |
| MOV | DX, ROW1 |
| OUT | DX, AL |
| MOV | DX, ROW2 |
| OUT | DX, AL |
| RET |  |
| ENDP |  |

；右边行输出

；循环移位BX,行线扫描输岀0

;8255控制字地址

;设置8255的PA、PB、PC口为输出口

；写控制字

;测试LED子程序，点亮LED并延时IS

TEST\_LED

TEST\_LED

|  |  |
| --- | --- |
| PROC | NEAR |
| MOV | DX, LINE |
| XOR | AX, AX |
| OUT | DX, AX |
| MOV | AL, OFFH |
| MOV | DX, ROW1 |
| OUT | DX, AL |
| MOV | DX, ROW2 |
| OUT | DX, AL |
| CALL | DL500ms |
| CALL | DL500ms |
| RET |  |
| ENDP |  |

；调整AL中取到的字型码的一个字节，将最高位调整位最低位,最低位调整为最高位

ADJUST PROC NEAR

|  |  |  |
| --- | --- | --- |
|  | PUSH  MOV | CX  CX, 8 |
| ADJUST1: | RCL | AL, 1 |
|  | XCHG | AL, AH |
|  | RCR | AL, 1 |
|  | XCHG | AL, AH |
|  | LOOP | ADJUST1 |
|  | MOV | AL, AH |
|  | POP | CX |
|  | RET |  |
| ADJUST | ENDP |  |
| DLlOms | PROC | NEAR |
|  | PUSH | CX |
|  | MOV | CX, 133 |
|  | LOOP | $ |
|  | POP | CX |
|  | RET |  |
| DLlOms | ENDP |  |
| DL500ms | PROC | NEAR |
|  | PUSH | CX |
|  | MOV | CX, OFFFFH |
|  | LOOP | $ |
|  | POP | CX |
|  | RET |  |
| DL500ms | ENDP |  |
|  | END | START |

七、实验扩展及思考

1、修改程序，使显示的字符从左至右动态循环显示。

实验三 数字式温度计实验(18B20)

一、 实验目的

掌握一线串行接口的读写操作；掌握数字温度计DS18B20的使用

**二、 实验设备**

SUN系列实验仪一套、PC机一台。

**三、 实验内容**

1、 DS18B20：

1. 一线串行接口数字式温度计
2. 温度测量范围-55°C-125°C, -10°C-85°C内误差±0.5°C
3. 9-12位转换精度，转换时间100ms-750ms,通常为500ms

2、 实验过程

(1)应用DS18B20制作一个数字温度计，通过DS18B20测量温度，B4区74HC273控制LED (F5区)动态显示温度

**四、 实验原理图**

vcch

囱R25 阜U4

(8255)PC0「； *"-L-2- -* DQ VCC ———VCC

TOUT 1 DS18B20



**五.实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| F2 区：TOUT |  | D3 区：PCO (8255) |
| D3 区：CS (8255)、 AO、 Al |  | A3 区：CS1、A0、A1 |
| B4 区:JP57 (DO.. D7) |  | A3 区:JP42 (DO. . D7) |
| B4 区:JP56 (D &.D15) |  | A3 区：JP40(D&.D15) |
| B4(I/0)区：CS273、BLE、BHE |  | A3 区：CS5、BLE、BHE |
| B4(I/0)区：RD、WR |  | A3 区：I0R、I0W |
| B4(I/0)区：JP51、JP55 |  | F5 区：B、C |
| 2、使用DS18B20测量温度，将读上 | B的十六进制温度值转换为十进制数 | |

3、通过LED(F5区)动态显示温度，温度数据通过DS18B20获取。可用手指贴住DS18B20 (G1

区)，温度显示会随之上升。

**六.演示程序**

.MODEL TINY

Con\_8255 EQU

PC\_8255 EQU

0273H

0272H

EXTRN InitKeyDisplay:NEAR, Display8A:NEAR

.STACK 300

.DATA

buffer DB 8 DUP (0)

；温度临时存放区

.CODE

|  |  |  |  |
| --- | --- | --- | --- |
| START: | MOV | AX, @DATA | |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | InitDisBuffer | ;给显示缓冲区赋初值，消隐 |
|  | CALL | Init8255 |  |
| MAIN: | CALL | START Temperature ;向DS18B20发送读温度指令 | |
|  | JB | MAIN |  |
|  | CALL | DelayTime |  |
|  | CALL | RD\_Temperature | ；读出温度值，并转换为BCD码 |
|  | CALL | DIS\_BCD ;提取温度数据，转换为非压缩型BCD码,并显示 | |
|  | JMP | MAIN |  |
| ；温度转换/显示 |  |  |  |
| DIS\_BCD | PROC | NEAR |  |
|  | MOV | BX, AX |  |
|  | LEA | DI, buffer+7 |  |
|  | STD |  |  |
|  | MOV | AL,10H | ;10H表示不需要显示 |
|  | STOSB |  |  |
|  | STOSB |  |  |
|  | STOSB |  |  |
|  | STOSB |  |  |
|  | TEST | AH, 08H |  |
|  | JNZ | DIS\_BCD1 |  |
|  | STOSB | ；正数 |  |
|  | JMP | DIS\_BCD2 |  |
| DIS\_BCD1: | MOV | AL, 11H |  |
|  | STOSB | ；负数 |  |
|  | NEG | BX |  |
| DIS\_BCD2: |  |  | ；将温度整数位转换为ASCI |
|  | SHL | BX, 1 | ；将温度的个位与十位合在BH中 |
|  | SHL | BX, 1 |  |
|  | SHL | BX, 1 |  |
|  | SHL | BX, 1 |  |
|  | MOV | AX, 10 |  |
|  | XCHG | AL, BH |  |
|  | DIV | BH |  |
|  | CMP | AL, 0 |  |
|  | JNZ | DIS\_BCD3 | ；判断温度的十位是否为0进行相应处理 |
|  | MOV | AL, 10H | ；十位为0 |
|  | XCHG | AL, [DI+1] |  |
|  | STOSB |  |  |
|  | JMP | DIS\_BCD4 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| DIS\_BCD3: | STOSB | | |
| DIS\_BCD4: | MOV | AL, AH |  |
|  | OR | AL, 80H | ;小数点 |
|  | STOSB  XOR | AL, AL | ;转换小数部分 |
|  | TEST | BL, 1OH |  |
|  | JZ | DIS\_BCD5 |  |
|  | MOV | AL, 6 |  |
| DIS\_BCD5: | TEST | BL, 20H |  |
|  | JZ | DIS\_BCD6 |  |
|  | ADD | AL, 12H |  |
| DIS\_BCD6: | DAA  TEST | BL, 40H |  |
|  | JZ | DIS\_BCD7 |  |
|  | ADD | AL, 25H |  |
| DIS\_BCD7: | DAA  TEST | BL, 80H |  |
|  | JZ | DIS\_BCD8 |  |
|  | ADD | AL, 50H |  |
| DIS\_BCD8: | DAA  MOV | CL, 4 |  |
|  | ROR | AL, CL |  |
|  | AND | AL, OFH |  |
| DIS\_BCD | STOSB  RET  ENDP |  |  |
| ；延时程序  DelayTime | PROC | NEAR |  |
|  | MOV | CX, 70 |  |
| LI: | LEA | SI, buffer |  |
|  | CALL | Display8A |  |
|  | LOOP | LI |  |
| DelayTime | RET  ENDP |  |  |
| ；写o  W\_L | PROC | NEAR |  |
|  | PUSH | AX |  |
|  | MOV | DX, Con\_8255 |  |
|  | MOV | AL, 80H |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
| W\_L | RET  ENDP |  |  |
| ；写1 |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| W\_H | PROC | NEAR | |
|  | PUSH | AX |  |
|  | MOV | DX, Con\_8255 |  |
|  | MOV | AL, O1H |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | RET |  |  |
| W\_H | ENDP |  |  |
| :DS18B20复位初始化子程序 | |  |  |
| INIT\_18B20 | PROC | NEAR |  |
|  | CALL | W\_L | ;主机发出501us复位低脉冲 |
|  | MOV | CX, 136 |  |
|  | LOOP | $ |  |
|  | MOV | DX, Con\_8255 |  |
|  | MOV | AX, 89H |  |
|  | OUT | DX, AL | ；PC输入状态 |
|  | DEC | DX |  |
|  | MOV | CX, 15 |  |
| INIT\_18B20\_l: | IN | AL, DX |  |
|  | TEST | AL, O1H |  |
|  | JZ | INIT\_18B20\_2 |  |
|  | LOOP | INIT\_18B20\_l |  |
|  | STC |  | ；置位标志位，表示DS18B20不存在 |
|  | RET |  |  |
| INIT\_18B20\_2: | MOV | CX, 136 |  |
|  | LOOP | $ |  |
|  | CLC |  | ;复位标志位，表示DS18B20存在 |
|  | RET |  |  |
| INIT\_18B20 | ENDP |  |  |
| ；写操作 |  |  |  |
| WRITE\_18B20 | PROC | NEAR |  |
|  | MOV | CX, 8 | ；一共8位数据 |
| WRI: | PUSH | AX | ;O->PCO CALL W\_L |
|  | MOV | DX, Con\_8255 |  |
|  | MOV | AL, 80H |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | ROR | AL, 1 |  |
|  | JNB | WRI1 |  |
|  | PUSH | AX | ;1->PCO CALL W\_H |
|  | MOV | DX, Con\_8255 |  |
|  | MOV | AL, 01H |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |

|  |  |  |  |
| --- | --- | --- | --- |
| WRI2: | PUSH | CX | ;延时55us |
|  | MOV | CX, 7 |  |
|  | LOOP | $ |  |
|  | POP | CX |  |
|  | CALL | W\_H |  |
|  | LOOP | WRI |  |
|  | RET |  |  |
| WRI1: | PUSH | CX |  |
|  | NOP |  |  |
|  | POP | CX |  |
|  | JMP | WRI 2 |  |
| WRITE\_18B20 | ENDP |  |  |
| ；读操作 |  |  |  |
| READ\_18B20 | PROC | NEAR |  |
|  | MOV | CX, 8 | ；数据一共有8位 |
| Read: | MOV | DX, Con\_8255 |  |
|  | MOV | AL, 80H |  |
|  | OUT | DX, AL | ;O->PCO |
|  | MOV | AL, 89H |  |
|  | OUT | DX, AL | ；输入状态 |
|  | NOP |  |  |
|  | NOP |  |  |
|  | NOP |  |  |
|  | NOP |  |  |
|  | NOP |  |  |
|  | MOV | DX, PC\_8255 |  |
|  | IN | AL, DX |  |
|  | ROR | AL, 1 |  |
|  | RCR | BL, 1 |  |
|  | PUSH | CX |  |
|  | MOV | CX, 11 |  |
|  | LOOP | $ |  |
|  | POP | CX |  |
|  | LOOP | Read |  |
|  | MOV | AL, BL |  |
|  | RET |  |  |
| READ\_18B20 | ENDP |  |  |
| ；判断DS18B20是否存在， | | 启动 DS18B20 | ；CY为判断标志 |
| START\_Temperature: CALL | | INIT\_18B20 | ;先复位DS18B20 |
|  | JB | GET\_T |  |
|  | MOV | AL, OCCH | ;跳过ROM匹配 |
|  | CALL | WRITE\_18B20 |  |
|  | MOV | AL, 44H | ；发出温度转换命令 |
|  | CALL | WRITE\_18B20 |  |

CLC

GET\_T: RET

|  |  |  |  |
| --- | --- | --- | --- |
| ;读出转换后的温度值， | | 存在AX | |
| RDTemperature: | :CALL | INIT\_18B20 | ;准备读温度前先复位 |
|  | MOV | AL, OCCH | ;跳过ROM匹配 |
|  | CALL | WRITE\_18B20 |  |
|  | MOV | AL, OBEH | ；发出读温度命令 |
|  | CALL | WRITE\_18B20 |  |
|  | CALL | READ\_18B20 | ；读出温度 |
|  | MOV | AH, AL | ;存放到AX |
|  | CALL | READ\_18B20 |  |
|  | XCHG | AL, AH |  |
|  | RET |  |  |
| Init8255 | PROC | NEAR |  |
|  | MOV | DX, Con\_8255 |  |
|  | MOV | AL, 80H |  |
|  | OUT | DX, AL |  |
|  | DEC | DX |  |
|  | DEC | DX |  |
|  | MOV | AL, OFFH |  |
|  | OUT | DX, AL |  |
|  | RET |  |  |
| Init8255 | ENDP |  |  |
| InitDisBuffer | PROC | NEAR |  |
|  | PUSH | DS |  |
|  | POP | ES |  |
|  | LEA | DI, buffer |  |
|  | MOV | AX, 1010H |  |
|  | MOV | CX, 4 |  |
|  | CLD |  |  |
|  | REP | STOSW |  |
|  | RET |  |  |
| InitDisBuffer | ENDP |  |  |
|  | END | START |  |
| **七、实验扩展及思考题** | |  |  |

实验内容：读取DS18B20内部64位识别码，了解多个DS18B20协同工作原理

实验四步进电机实验

一、 实验目的与要求

1、 了解步进电机的基本原理，掌握步进电机的转动编程方法

2、 了解影响电机转速的因素有那些

**二、 实验设备**

STAR系列实验仪一套、PC机一台。

**三、 实验内容**

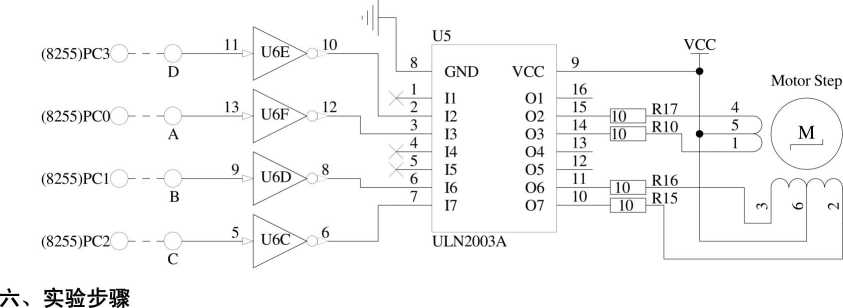
编写程序：使用F5区的键盘控制步进电机的正反转、调节转速，连续转动或转动指定步数; 将相应的数据显示在F5区的数码管上。

**四、 控制原理**

步进电机的驱动原理是通过它每相线圈的电流的顺序切换来使电机作步进式旋转，驱动电 路由脉冲来控制，所以调节脉冲的频率便可改变步进电机的转速，微控制器最适合控制步进电 机。另外，由于电机的转动惯量的存在，其转动速度还受驱动功率的影响，当脉冲的频率大于 某一值（本实验为f. >100hz）时，电机便不再转动。

实验电机共有四个相位（A,B,C,D）,按转动步骤可分单4拍（A-〉B->C-〉D->A），双4拍 （AB->BC->CD->DA->AB）和单双 8 拍（A-〉AB-〉B-〉BC->C->CD-〉D-〉DA-〉A）.

**五、 实验原理图**



1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| DI 区：A、B、C、D |  | D3 区：PC4、 PC5、 PC6、 PC7 |
| D3 区：CS、AO、Al |  | A3 区：CS1、AO、Al |
| D3 区：PCO、PCI |  | F5 区：KL1、KL2 |
| D3 区:JP20(PB)、B、C |  | F5 区：A、B、C |
| B3 区：CS、AO |  | A3 区：CS3、AO |
| B3 区：INT、INTA |  | A3 区：INTR、INTA |
| C4 区:CS (8253) > AO、Al |  | A3 区：CS2、AO、Al |
| C4 区：GATE |  | Cl 区：VCC |
| C4 区：CLKO |  | B2 区：IM |
| C4 区：OUTO |  | B3 区：IRO |

2、调试程序，查看运行结果是否正确

**七、演示程序**

.MODEL TINY

EXTRN InitKeyDisplay:NEAR, GetKeyA:NEAR, DisPlay8:NEAR

|  |  |  |  |
| --- | --- | --- | --- |
| I08259\_0 | EQU | 0250H | |
| I08259\_l | EQU | 0251H |  |
| Con\_8253 | EQU | 0263H |  |
| T0\_8253 | EQU | 0260H |  |
| I08255\_Con | EQU | 0273H | :CS3 |
| I08255\_PC | EQU | 0272H |  |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| StepControl | DB | 0 | ;下一次送给步进电机的值 |
| buffer | DB | 8 DUP (0) | ;显示缓冲区，8个字节 |
| SpeedNo | DB | 0 | ；选择哪一级速度 |
| StepDelay | DB | 0 | ；转动一步后，延时常数 |
| StartStepDelay | DB | 0；若选择速度过快,延时由长到短，最终使用对应延时常数 | |
| StartStepDelayl | DB | 0 | ;StartStepDelay |
| bFirst | DB | 0 | ;有没有转动过步进电机 |
| bClockwise | DB | 0 ； | ;=1顺时针方向=0逆时针方向转动 |
| bNeedDisplay | DB | 0 | ;已转动一步，需要显示新步数 |
| StepCount | DW | 0 | ;需要转动的步数 |
| StepDelayTab: | DB | 250, 125, 83, 62, 50, 42, 36, 32, 2& 25, 22, 21 | |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | InitKeyDisplay | ；初始化键盘、数码管控制器（8255） |
|  | MOV | bFirst, 1 | ；有没有转动过步进电机 |
|  | MOV | bClockwise, 1 | ；顺时针方向 |
|  | MOV | StepControl, 33H | ；下一次送给步进电机的值 |
|  | MOV | SpeedNo, 5 | ；第五级速度 |
|  | CALL | Init8255 |  |
|  | CALL | Init8253 |  |
|  | CALL | Init8259 |  |
|  | CALL | Wrilntver |  |
|  | MOV | buffer,0 | ；显示缓冲器初始化 |
|  | MOV | buffer+1, 0 |  |
|  | MOV | buffer+2, 0 |  |
|  | MOV | buffer+3, 0 |  |
|  | MOV | buffer+4, 10H |  |
|  | MOV | AL, SpeedNo |  |
|  | MOV | buffer+5, AL |  |
|  | MOV | buffer+6, 10H |  |
|  | MOV | buffer+7, 0 |  |
| STAR2: | LEA | SI,buffer |  |
|  | CALL | Display8 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| STAR3: | CALL | GetKeyA | |
|  | JB | STAR5 |  |
|  | CMP | bNeedDisplay, 0 |  |
|  | JZ | STAR3 |  |
|  | MOV | bNeedDisplay, 0 |  |
|  | CALL | Step\_SUB\_l |  |
|  | JMP | STAR2 |  |
| STAR5: | CLI |  | ;终止步进电机转动 |
|  | CMP | AL, 10 |  |
|  | JNB | STAR1 |  |
|  | MOV | AH, buffer+2 |  |
|  | MOV | buffer+3, AH |  |
|  | MOV | AH, buffer+1 |  |
|  | MOV | buffer+2, AH |  |
|  | MOV | AH, buffer |  |
|  | MOV | buffer+1, AH |  |
|  | MOV | buffer, AL |  |
|  | JMP | STAR2 |  |
| STAR1: | CMP | AL, 14 |  |
|  | JNB | STAR2 |  |
|  | LEA | SI, DriverTab |  |
|  | SUB | AL, 10 |  |
|  | SHL | AL, 1 |  |
|  | XOR | AH, AH |  |
|  | MOV | BX, AX |  |
|  | JMP | CS:[SI+BX] |  |
| DriverTab: | DW | Direction | ;转动方向 |
|  | DW | Speed\_up | ;提高转速 |
|  | DW | Speed\_Down | ;降低转速 |
|  | DW | Exec | ;步进电机根据方向、转速、步数开始转动 |
| Direction: | CMP | bClockwise, 0 |  |
|  | JZ | Clockwise |  |
|  | MOV | bClockwise, 0 |  |
|  | MOV | buffer+7, 1 |  |
| AntiClockwise: | CMP | bFirst, 0 |  |
|  | JZ | AntiClockwisel |  |
|  | MOV | StepControl, 91H |  |
|  | JMP | Directionl |  |
| AntiClockwisel: | MOV | AL, StepControl |  |
|  | ROR | AL, 2 |  |
|  | MOV | StepControl, AL |  |
|  | JMP | Directionl |  |
| Clockwise: | MOV | bClockwise, 1 |  |
|  | MOV | buffer+7, 0 |  |

|  |  |  |
| --- | --- | --- |
| CMP | | bFirst, 0 |
|  | JZ | Clockwisel |
|  | MOV | StepControl, 33H |
|  | JMP | Directionl |
| Clockwisel: | MOV | AL, StepControl |
|  | ROL | AL, 2 |
|  | MOV | StepControl, AL |
| Directionl: | JMP | STAR2 |
| Speed\_up: | MOV | AL, SpeedNo |
|  | CMP | AL, 11 |
|  | JZ | Speed\_up2 |
| Speed\_upl: | INC | AL |
|  | MOV | SpeedNo, AL |
|  | MOV | buffer+5, AL |
| Speed\_up2: | JMP | STAR2 |
| SpeedDown: | MOV | AL, SpeedNo |
|  | CMP | AL, 0 |
|  | JZ | SpeedDownl |
|  | DEC | AL |
|  | MOV | SpeedNo, AL |
|  | MOV | buffer+5, AL |
| SpeedDownl: | JMP | STAR2 |
| Exec: | MOV | bFirst, 0 |
|  | CALL | TakeStepCount |
|  | LEA | BX, StepDelayTab |
|  | MOV | AL,SpeedNo |
|  | XLAT |  |
|  | MOV | StepDelay, AL |
|  | CMP | AL, 50 |
|  | JNB | Execl |
|  | MOV | AL, 50 |
| Execl: | MOV | StartStepDelay, AL |
|  | MOV | StartStepDelayl, AL |
|  | STI |  |
|  | JMP | STAR2 |
| TIMERO: | PUSH | AX |
|  | PUSH | DX |
|  | DEC | StartStepDelay |
|  | JNZ | TIMERO\_1 |
|  | MOV | AL, StartStepDelayl |
|  | CMP | AL, StepDelay |
|  | JZ | TIMER0\_2 |
|  | DEC | AL |
|  | MOV | StartStepDelayl, AL |

|  |  |  |
| --- | --- | --- |
| TIMER0\_2: | MOV | StartStepDelay, AL |
|  | MOV | AL, StepControl |
|  | MOV | DX, IO8255\_PC |
|  | OUT | DX, AL |
|  | CMP | bClockwise, 0 |
|  | JNZ | TIMER0\_3 |
|  | ROR | AL, 1 |
|  | JMP | TIMER0\_4 |
| TIMER0\_3: | ROL | AL, 1 |
| TIMER0\_4: | MOV | StepControl, AL |
|  | CMP | StepCount, 0 |
|  | JZ | TIMERO\_1 |
|  | MOV | bNeedDisplay, 1 |
|  | DEC | StepCount |
|  | JNZ | TIMERO\_1 |
|  | add popf cli pushf | sp, 8 ；小写部分不允许使用单步、单步进入命令 |
|  | sub  nop | sp, 8 |
| TIMERO\_1: | MOV | DX, 108259\_0 |
|  | MOV | AL, 20H |
|  | OUT | DX, AL |
|  | POP | DX |
|  | POP  I RET | AX |
| Step\_SUB\_l | PROC | NEAR |
|  | MOV | CX,4 |
|  | LEA | BX, buffer |
| Step\_SUB\_l\_l: | DEC | BYTE PTR [BX] |
|  | CMP | BYTE PTR [BX],0FFH |
|  | JNZ | Step\_SUB\_l\_2 |
|  | MOV | BYTE PTR [BX],9 |
|  | INC | BX |
|  | LOOP | Step\_SUB\_l\_l |
| Step\_SUB\_l\_2: | RET |  |
| Step\_SUB\_l | ENDP |  |
| TakeStepCount | PROC | NEAR |
|  | MOV | AL, buffer+3 ；转动步数送入StepCount |
|  | MOV | BX, 10 |
|  | MUL | BL |
|  | ADD | AL,buffer+2 |
|  | MUL | BL |

TakeStepCount

Init8255

Init8255

Init8253

Init8253

Init8259

Init8259

Wrilntver

|  |  |
| --- | --- |
| ADD | AL, buffer+1 |
| ADC | AH, 0 |
| MUL | BX |
| ADD | AL, buffer |
| ADC | AH,0 |
| MOV | StepCount, AX |
| RET |  |
| ENDP |  |
| PROC | NEAR |
| MOV | DX, I08255\_Con |
| MOV | AL, 81H |
| OUT | DX, AL |
| DEC | DX |
| MOV | AL, OFFH |
| OUT | DX, AL |
| RET |  |
| ENDP |  |
| PROC | NEAR |
| MOV | DX, Con\_8253 |
| MOV | AL, 35H |
| OUT | DX, AL |
| MOV | DX, T0\_8253 |
| MOV | AL, 10H |
| OUT | DX, AL |
| MOV | AL, 02H |
| OUT | DX, AL |
| RET |  |
| ENDP |  |
| PROC | NEAR |
| MOV | DX, I08259\_0 |
| MOV | AL, 13H |
| OUT | DX, AL |
| MOV | DX, IO8259\_1 |
| MOV | AL, 08H |
| OUT | DX, AL |
| MOV | AL, 09H |
| OUT | DX, AL |
| MOV | AL, OFEH |
| OUT | DX, AL |
| RET |  |
| ENDP |  |
| PROC | NEAR |
| PUSH | ES |
| MOV | AX, 0 |

;8255 PC输出

;0FFH->8255 PC

;计数器TO设置在模式2状态,BCD码计数

:CLK0/210

Wrilntver

|  |  |
| --- | --- |
| MOV | ES, AX |
| MOV | DI, 20H |
| LEA | AX, TIMERO |
| STOSW |  |
| MOV | AX, CS |
| STOSW |  |
| POP | ES |
| RET |  |
| ENDP |  |
| END | START |

八、实验扩展及思考

1、 怎样改变电机的转速？

2、 通过实验找出电机转速的上限，如何能进一步提高最大转速?

3、 怎样能使电机反转？

实验五直流电机测速实验

实验目的

了解直流电机工作原理；了解光电开关的原理；掌握使用光电开关测量直流电机转速。

实验设备

SUN系列实验仪一套、PC机一台。

三.实验内容

1、转速测量原理:

JK无艾

图3转盘

本转速测量实验采用反射式光电开关，通过计数转盘通断光电开关产生的脉冲，计算出转速

1. 反射式光开关工作原理：光电开关发射光，射到测量物体上，如果强反射，如图h光电 开关接收到反射回来的光，则产生高电平1；弱反射，如图2,光电开关接收不到反射回来的光, 则产生弱电平0。
2. 实验方法：本实验转速测量用的转盘在下表面做成如图3样子的转盘，白部分为强反射区, 黑部分为弱反射区，转盘每转一圈，产生4个脉冲，每1/4秒计数出脉冲数， 速。(演示程序中，LED显示的是每秒钟转速)

2、实验过程

1. 由DAC0832给电机供电，使用光电开关，测量电机转速，再经调整， 在LED上。
2. 通过按键调节电机转速，随之变化的转速动态显示LED上

**实验原理图**

图1强反射

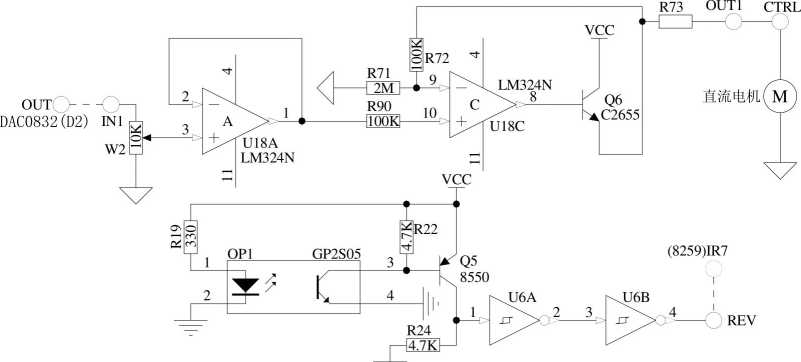
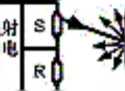
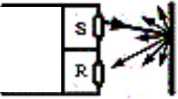
图2弱反射

即得到每秒的转

最终将转速显示

四.

**五、实验步骤**



1、连线说明:

|  |  |  |
| --- | --- | --- |
| B3 区：CS、A0 |  | A3 区：CS3、A0 |
| B3 区：INT、INTA |  | A3 区：INTR、INTA |
| C4 区：CS、AO、A1 |  | A3 区：CS2、AO、A1 |
| C4 区：GATE |  | C1 区：VCC |
| C4 区:CLK0 |  | B2 区：31250Hz |
| C4 区:CLK1 |  | B2 区：1M |
| C4 区：OUTO |  | B3 区：IR0 |
| D2 区：CS |  | A3 区：CS4 |
| D2 区：OUT |  | F3 区：IN1 |
| F3 区：0UT1 |  | E1 区：CTRL |
| E1 区：REV |  | B3 区：IR7 |
| D3 区：CS、A0、A1 |  | A3 区：CS1、AO、A1 |
| D3 区:JP20(PB)、B、C |  | F5 区：A、B、C |
| D3 区：PCO、PC1 |  | F5 区：KL］、KL2 |

2、 由DAC0832经功放电路驱动直流电机，计数光电开关通关次数并经过换算得出直流电机 的转速，并将转速显示在LED上。

3、 F5区的0、1号按键控制直流电机转速快慢，（最大转速~96r/s, 5V,误差±lr/s）

|  |  |  |  |
| --- | --- | --- | --- |
| **六、演示程序** | .MODEL | TINY | |
| EXTRN | InitKeyDisplay:NEAR, DisPlay8:NEAR, GetKeyA:NEAR | | |
| 108259\_0 | EQU | 0250H |  |
| 108259\_1 | EQU | 0251H |  |
| Con\_8253 | EQU | 0263H |  |
| T0\_8253 | EQU | 0260H |  |
| Tl\_8253 | EQU | 0261H |  |
| DA0832 | EQU | 0240H |  |
| VoltageOffset | EQU | 5 | ;0832调整幅度 |
|  | .STACK | 200 |  |
|  | .DATA |  |  |
| buffer | DB | 8 DUP(O) | ；显示缓冲区，8个字节 |
| buffer1 | DB | 8 DUP(O) | ；显示缓冲区，8个字节 |
| VOLTAGE | DB | 0 | ;转换电压数字量 |
| Cou nt | DW | 0 | ；一秒转动次数 |
| NowCount | DW | 0 | ；当前计数值 |
| kpTime | DW | 0 | ;保存上一次采样时定时器的值 |
| bNeedDisplay | DB | 0 | ；需要刷新显示 |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | InitKeyDisplay | ;对键盘、数码管扫描控制器8255初始化 |

|  |  |  |  |
| --- | --- | --- | --- |
| MOV | | bNeedDisplay, 1 | ;显示初始值 |
|  | MOV | VOLTAGE, 99H | ；初始化转换电压输入值，99H-3.0V |
|  | MOV | Count, 0 | ；一秒转动次数 |
|  | MOV | NowCount, 0 | ；当前计数值 |
|  | MOV | kpTime, 0 | ;保存上…次采样时定时器的值 |
|  | CALL | DAC0832 | ；初始D/A |
|  | CALL | Init8253 |  |
|  | CALL | Init8259 |  |
|  | CALL | Wrilntver |  |
|  | STI |  |  |
| MAIN: | CALL | GetKeyA | ；按键扫描 |
|  | JNB | Mainl |  |
|  | CMP | AL, OOH |  |
|  | JNZ | Keyl |  |
| KeyO: | MOV | AL, VoltageOffset; 0号键按下，转速提高 | |
|  | ADD | AL, VOLTAGE |  |
|  | CMP | AL, VOLTAGE |  |
|  | JNB | KeyO\_l |  |
|  | MOV | AL, OFFH | ；最大 |
| KeyO\_l: | MOV | VOLTAGE, AL |  |
|  | CALL | DAC0832 | :D/A |
|  | JMP | Main2 |  |
| Keyl: | MOV | AL, VOLTAGE | ；1号键按下，转速降低 |
|  | SUB | AL,VoltageOffset |  |
|  | JNB | Keyl\_l |  |
|  | XOR | AL, AL | ；最小 |
| Keyl\_l: | MOV | VOLTAGE, AL |  |
|  | CALL | DAC0832 | ;D/A |
|  | JMP | Main2 |  |
| Mainl: | CMP | bNeedDisplay, 0 |  |
|  | JZ | MAIN |  |
|  | MOV | bNeedDisplay,0 | :Is定时到刷新转速 |
| Main2: | CALL | RateTest | ；计算转速/显示 |
|  | JMP | MAIN | ;循环进行实验内容介绍与测速功能测试 |
| ;转速测量/显示 |  |  |  |
| Ra/t eTes t: | MOV | AX, Count |  |
|  | MOV | BL, 10 |  |
|  | DIV | BL |  |
|  | CMP | AL, 0 |  |
|  | JNZ | RateTestl |  |
|  | MOV | AL, 10H | ；高位为0,不需要显示 |
| Ra/t eTes tl: | MOV | buffer, AH |  |
|  | MOV | buffer+1, AL |  |
|  | MOV | AL, VOLTAGE | ;给0832送的数据 |

|  |  |  |  |
| --- | --- | --- | --- |
| AND | | AL, OFH | |
|  | MOV | buffer+4, AL |  |
|  | MOV | AL,VOLTAGE |  |
|  | AND | AL, OFOH |  |
|  | ROR | AL, 4 |  |
|  | MOV | buffer+5, AL |  |
|  | MOV | buffer+2, 10H | ;不显示 |
|  | MOV | buffer+3, 1OH |  |
|  | MOV | buffer+6, 1OH |  |
|  | MOV | buffer+7, 10H |  |
|  | LEA | SI,buffer |  |
|  | CALL | Display8 | ；显示转换结果 |
|  | RET |  |  |
| TimerOInt: | PUSH | AX |  |
|  | PUSH | DX |  |
|  | MOV | bNeedDisplay, 1 |  |
|  | MOV | AX, NowCount |  |
|  | SHR | AX, 1 |  |
|  | SHR | AX, 1 |  |
|  | MOV | Count, AX | ;转一圈产生4个脉冲,Count二NowCount/4 |
|  | MOV | NowCount, 0 |  |
|  | MOV | DX, I08259\_0 |  |
|  | MOV | AL,20H |  |
|  | OUT | DX, AL |  |
|  | POP | DX |  |
|  | POP | AX |  |
|  | I RET |  |  |
| Countlnt: | PUSH | AX |  |
|  | PUSH | DX |  |
|  | MOV | DX, Con\_8253 |  |
|  | MOV | AL, 40H |  |
|  | OUT | DX, AL | ;锁存 |
|  | MOV | DX, Tl\_8253 |  |
|  | IN | AL, DX |  |
|  | MOV | AH, AL |  |
|  | IN | AL, DX |  |
|  | XCHG | AL, AH | ；T1的当前值 |
|  | XCHG | AX, kpTime |  |
|  | SUB | AX, kpTime |  |
|  | CMP | AX,100 |  |
|  | JB | CountInti | ;前后二次采样时间差小于100,判断是干扰 |
|  | INC | NowCount |  |
| CountInti: | MOV | DX, I08259\_0 |  |
|  | MOV | AL, 20H |  |

|  |  |  |  |
| --- | --- | --- | --- |
| OUT | | DX, AL | |
|  | POP | DX |  |
|  | POP  IRET | AX |  |
| Init8253 | PROC | NEAR |  |
|  | MOV | DX, Con\_8253 |  |
|  | MOV | AL, 34H |  |
|  | OUT | DX, AL | ;计数器TO设置在模式2状态,HEX计数 |
|  | MOV | DX, T0\_8253 |  |
|  | MOV | AL, 12H |  |
|  | OUT | DX, AL |  |
|  | MOV | AL, 7AH |  |
|  | OUT | DX, AL | :CLK0=31250Hz, Is 定时 |
|  | MOV | DX, Con\_8253 |  |
|  | MOV | AL, 74H |  |
|  | OUT | DX, AL | ；计数器T1设置在模式2状态,HEX计数 |
|  | MOV | DX, Tl\_8253 |  |
|  | MOV | AL, OFFH |  |
|  | OUT | DX, AL |  |
|  | MOV | AL, OFFH |  |
|  | OUT  RET | DX, AL | ；作定时器使用 |
| Init8253 | ENDP |  |  |
| Init8259 | PROC | NEAR |  |
|  | MOV | DX, I08259\_0 |  |
|  | MOV | AL, 13H |  |
|  | OUT | DX, AL |  |
|  | MOV | DX, IO8259\_1 |  |
|  | MOV | AL,08H |  |
|  | OUT | DX, AL |  |
|  | MOV | AL, 09H |  |
|  | OUT | DX, AL |  |
|  | MOV | AL, 7EH |  |
|  | OUT  RET | DX, AL |  |
| Init8259 | ENDP |  |  |
| Wrilntver | PROC | NEAR |  |
|  | PUSH | ES |  |
|  | MOV | AX, 0 |  |
|  | MOV | ES, AX |  |
|  | MOV | DI, 20H |  |
|  | LEA  STOSW | AX, TimerOInt |  |
|  | MOV | AX, CS |  |

|  |  |  |
| --- | --- | --- |
| ST0SW | | |
|  | LEA | AX, Countlnt |
|  | MOV | DI, 3CH |
|  | STOSW |  |
|  | MOV | AX, CS |
|  | STOSW |  |
|  | POP | ES |
|  | RET |  |
| Wrilntver | ENDP |  |
| ；数模转换， | A-转换数字量 |  |
| DAC0832 | PROC | NEAR |
|  | MOV | DX, DA0832 |
|  | MOV | AL, VOLTAGE |
|  | OUT | DX, AL |
|  | RET |  |
| DAC0832 | ENDP |  |
|  | END | START |
| **七.实验扩展及思考题** | |  |

实验内容：在日光灯或白炽灯下，将转速调节到25、50、75,观察转盘有什么现象出来

实验六旋转图形实验

实验目的

了解旋转图形如何呈现的原理；加深了解控制在实际中的应用。

实验设备

SUN系列实验仪一套、PC机一台。

1. 实验内容

1、 实验原理

一个旋转中的图形，当每转到一定角度时，被照亮一下，不断重复此过程，就可以看到图 形稳定的图像（虽然该图形在旋转中）。如果推迟照亮（推迟时间很短），就会出现图形在慢慢旋 转。

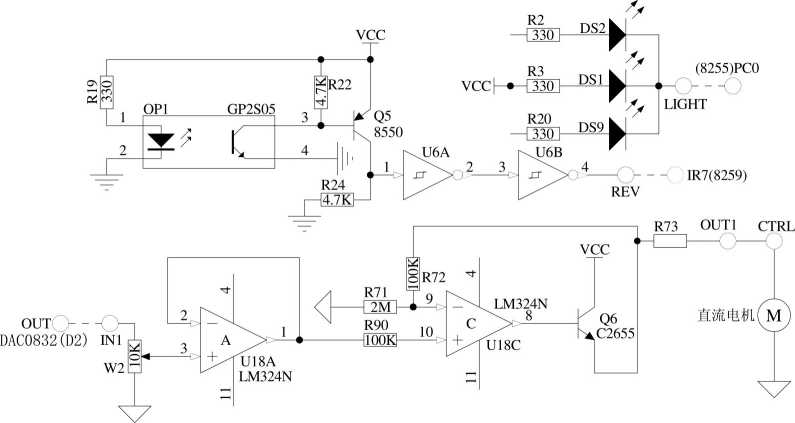
本实验利用光电开关来判断转盘转到某个固定角度的时刻，给出中断信号

2、 实验过程

（1）由光电开关结合中断给出控制信号，通过CPU控制直流电机边上的小灯闪烁，将直流 电机驱动的旋转转盘上的图形呈现出来，并令图形也在旋转（缓慢的）。

\*该实验类似于：夏天，快速旋转的风扇里能看到缓慢旋转的扇叶的假像。

1. **实验原理图**



**五.实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| B3 区：CS、A0 |  | A3 区：CS3、A0 |
| B3 区：INT、INTA |  | A3 区：INTR、INTA |
| B3 区：IR7 |  | El 区：REV |
| D2 区：CS |  | A3 区：CS4 |
| D2 区：OUT |  | F3 区：IN1 |
| F3 区：0UT1 |  | E1 区：CTRL |
| D3 区:CS(8255), AO、A1 |  | A3 区：CS1、AO、A1 |
| D3 区：PCO |  | E1 区：LIGHT |

2、 检测光电开关产生的中断信号，判断转盘转到固定的角度，计算得出延时，延时过后闪 亮（几十微秒）直流电机上的小灯一次（确保小灯照在转盘上）。

3、 每转一圈产生4个中断，演示程序运行后，可以看到转盘中的图形在转动。

|  |  |  |  |
| --- | --- | --- | --- |
| **六.演示程序** |  |  |  |
|  | ・MODEL | TINY |  |
| 108259\_0 | EQU | 0250H |  |
| 108259\_1 | EQU | 0251H |  |
| Io8255\_Con | EQU | 0273H | ;8255控制口 |
| Io8255\_PC | EQU | 0272H | ;PC.O控制小灯 |
| DA0832 | EQU | 0240H | ;0832输出口地址 |
| Voltage | EQU | 085H | ；直流电机输入电压，85—>50r/s |
|  | .STACK | 100 |  |
|  | .DATA |  |  |
| Cou nt | DW | 0 | ；计数值缓冲器 |
| FlashFlag | DB | 0 | ；是否点亮小灯标志 |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | nop |  |  |
|  | MOV | Count,1 |  |
|  | MOV | FlashFlag, 0 | ；不点亮小灯 |
|  | CALL | Init8255 | ;初始化8255,熄灭小灯 |
|  | CALL | Init8259 |  |
|  | CALL | Wrilntver |  |
|  | CALL | DAC0832 |  |
|  | STI |  |  |
| MAIN: | CMP | FlashFlag, 0 | ；是否点亮小灯 |

JZ MAIN

MOV FlashFlag, 0

CALL FlashLED

JMP MAIN

；点灯中断标志服务程序

|  |  |  |  |
| --- | --- | --- | --- |
| Int\_LED: | PUSH | AX | |
|  | PUSH | DX |  |
|  | MOV | FlashFlag, 1 | ；发光标志，1有效 |
|  | MOV | DX, I08259\_0 |  |
|  | MOV | AL, 20H |  |
|  | OUT | DX, AL |  |
|  | POP | DX |  |
|  | POP | AX |  |
|  | I RET |  |  |
| ；点亮小灯 |  |  |  |
| FlashLED | PROC | NEAR |  |

|  |  |  |
| --- | --- | --- |
|  | CALL | DelayLED |
|  | MOV | DX, Io8255\_PC |
|  | MOV | AL, 0 |
|  | OUT | DX, AL ；点亮小灯 |
|  | CALL | DelayTime |
|  | MOV | DX, Io8255\_PC |
|  | MOV | AL, OFFH |
|  | OUT | DX, AL |
|  | RET | ;熄灭小灯 |
| FlashLED | ENDP |  |
| ;小灯点亮前要延时，产生旋转中的图形在旋转的效果 | | |
| DelayLED | PROC | NEAR |
|  | MOV | CX, Count ;每次发光前的延时，由Count内的值决定 |
| DelayLEDl: | CALL | Delay ；延时 |
|  | LOOP | DelayLEDl |
|  | INC | Cou nt |
|  | MOV | AX, Count |
|  | CMP | AX, 200 |
|  | JNZ | DelayLED2 |
|  | MOV | Count, 1 ;延时达到转盘转过1/4圈的时间，要重新设定 |
| DelayLED2: | RET |  |
| DelayLED | ENDP |  |
| ；延时 |  |  |
| Delay | PROC | NEAR |
|  | PUSH | CX |
|  | MOV | CX, 1 |
|  | LOOP | $ |
|  | POP | CX |
|  | RET |  |
| Delay | ENDP |  |
| ;小灯闪亮的时间，控制图形的亮度和清晰度 | | |
| DelayTime | PROC | NEAR |
|  | PUSH | CX |
|  | MOV | CX, 48 |
|  | LOOP | $ |
|  | POP | CX |
|  | RET |  |
| DelayTime | ENDP |  |
| Init8259 | PROC | NEAR |
|  | MOV | DX, I08259\_0 |
|  | MOV | AL,13H |
|  | OUT | DX, AL |
|  | MOV | DX, I08259\_l |
|  | MOV | AL, 08H |

|  |  |  |
| --- | --- | --- |
| OUT | | DX, AL |
|  | MOV | AL, 09H |
|  | OUT | DX, AL |
|  | MOV | AL, 7FH |
|  | OUT | DX, AL |
|  | RET |  |
| Init8259 | ENDP |  |
| Wrilntver | PROC | NEAR |
|  | PUSH | ES |
|  | MOV | AX, 0 |
|  | MOV | ES, AX |
|  | MOV | DI, 3CH |
|  | LEA | AX, Int\_LED |
|  | STOSW |  |
|  | MOV | AX, CS |
|  | STOSW |  |
|  | POP | ES |
|  | RET |  |
| Wrilntver | ENDP |  |
| ;设置直流电机供电电压 | |  |
| DAC0832 | PROC | NEAR |
|  | MOV | DX, DA0832 |
|  | MOV | AL,Voltage |
|  | OUT | DX, AL |
|  | RET |  |
| DAC0832 | ENDP |  |
| Init8255 | PROC | NEAR |
|  | MOV | DX, Io8255\_Con |
|  | MOV | AL,80H |
|  | OUT | DX, AL |
|  | MOV | AL, OFFH |
|  | DEC | DX |
|  | DEC | DX |
|  | OUT | DX, AL |
|  | RET |  |
| Init8255 | ENDP |  |
|  | END | START |

:PC 口输出

;1->PC.0

七.实验扩展及思考题

实验内容：上面实验中的图形的旋转方向是同向的，通过控制小灯闪烁时机，可以实现旋 转中的图形反向旋转，有兴趣者可自行尝试

实验七ISD1420语音模块实验

一、 实验目的

了解ISD1420的性能；与8086的接口逻辑；掌握手动和CPU控制两种录音、放音的基本功 能。

**二、 实验设备**

STAR系列实验仪一套、PC机一台。

**三、 实验内容**

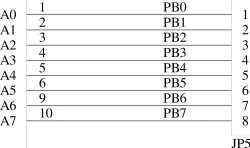
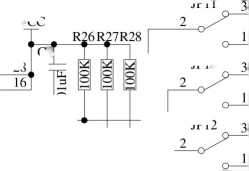
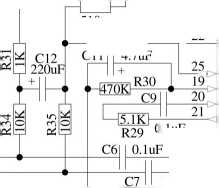
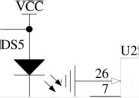
1、 ISD1420语音模块(B1区)：

(1) 20秒录放音长度，具有不掉电存储功能；(2)可分1 — 160段录放音片段

2、 具体操作

1. 手动控制方式，通过B1区按键REC录音和按键PLAYE、PLAYL放音
2. MCU控制方式，通过F5区8个按键控制录、放音：0~3号键录音各5秒；然后通过4〜 7号键放音，放音内容顺序对应0〜3号键的录音内容

**四、 实验原理图**



Mic

/

Speaker

8

11

R9

【).1 uF

O.luFg

510

Cl 1 I I 4.7uF

XCLK

NC NC NC

NC

RECLED

AGC

ANA IN

ANA OUT

SP+

SP-

VSSD

VSSA

.TP11

VCC

C8

27

.TP 12

VCCD

VCCA

o

**o**

6

-o o

PLAYE

-o

REC

REC PLAYE PLAYL

28

JP10

MIC REF

MIC

-o o-

PLAYL

REC

~o—o

PC0(8255)

PLAYL

—Q——Q

PC2(8255)

= ISD1420

1. 实验步骤

**六.演示程序**

ISD1420\_AD0

.MODEL

・ STACK

EQU

TINY

100

00H

；堆栈段

；0号键录放音起始地址，每次录音5s

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区:CS(8255)、A0、Al |  | A3 区：CS1、A0、Al |
| D3 区:JP23 (PA 口) |  | F5 区：A |
| F5 区：KL1 |  | Cl 区：GND |
| Bl 区：REC |  | D3区:PC0(8255)录音控制 |
| Bl 区：PLAYE |  | D3区:PCI (8255)电平放音控制 |
| Bl 区：PLAYL |  | D3区：PC2(8255)触发放音控制，下降沿触发 |
| Bl 区：JP5 |  | D3 区：JP20 (PB 口) |

2、将JP10, JP11, JP12跳向“MCU”，8086控制，运行演示程序，0〜3号键录音，4〜7号

键放音。

|  |  |  |  |
| --- | --- | --- | --- |
| ISD142O\_AD1 | EQU | 28H | ;1号键录放音起始地址 |
| ISD1420\_AD2 | EQU | 50H | ;2号键录放音起始地址 |
| ISD1420\_AD3 | EQU | 78H | ;3号键录放音起始地址 |
| ISDCOMM | EQU | 0271H ;录放音地址/操作模式输入地址，由8255的PB口控制 | |
| I8255\_Ctr | EQU | 0273H | ;8255控制端口地址 |
| I8255\_PA | EQU | 0270H | ；键盘数据输入口 |
| I8255\_PC | EQU | 0272H | ;ISD1420控制输出口 |
| .DATA |  |  |  |
| KeepMode | DB | 7 | ;保存REC、PLAYE、PLAYL当前值 |
| bNewKey | DB | 0 | ；有键按下标志位，清0-无键按下 |
| KEYno | DB | 0 |  |
| KeyTab | DW KEYO, | KEY1, KEY2, KEY3, KEY4, KEY5, KEY6, KEY7；录音键放音键子程序入口 | |
| .CODE |  |  | ；程序段 |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | NOP |  |  |
|  | CALL | MainInit | ；主程序初始化 |
| Main: | CALL | ScanKey | ；扫描按键 |
|  | JNB | Main |  |
| Mainl: | CALL | KeyRun | ;按键处理 |
|  | CMP | bNewKey, 0 | ；是否有新的键按下 |
|  | JZ | Main |  |
|  | MOV | bNewKey, 0 | ；清按键标志 |
|  | JMP | Mainl | ；循环进行实验内容介绍与ISD1420功能测试 |
| ;主程序初始化 |  |  |  |
| MainInit | PROC | NEAR |  |
|  | MOV | bNewKey, 0 | ；有键按下标志位，清0-无键按下 |
|  | MOV | DX, I8255\_Ctr ;8255初始化 | |
|  | MOV | AL, 90H | ；PA为输入，PC的低四位为输出 |
|  | OUT | DX, AL |  |
|  | CALL | ISD\_INIT |  |
|  | RET |  |  |
| MainInit | ENDP |  |  |
| ；录放音子程序 |  |  |  |
| KEYO | PROC | NEAR |  |
|  | MOV | AL, ISD1420\_ | \_ADO ;0号键录音首地址 |
|  | CALL | KEY\_REC |  |
|  | RET |  |  |
| KEYO | ENDP |  |  |
| KEY1 | PROC | NEAR |  |
|  | MOV | AL, ISD1420\_ | \_AD1 ; 1号键录音首地址 |
|  | CALL | KEY\_REC |  |
|  | RET |  |  |
| KEY1 | ENDP |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| KEY2 | PROC  MOV  CALL  RET | NEAR | |
| AL, ISD1420\_AD2  KEY\_REC | ;2号键录音首地址 |
| KEY2 | ENDP |  |  |
| KEY3 | PROC | NEAR |  |
|  | MOV | AL,ISD1420\_AD3 | ；3号键录音首地址 |
|  | CALL | KEY\_REC |  |
|  | RET |  |  |
| KEY3 | ENDP |  |  |
| ；录音子程序 |  |  |  |
| KEY\_REC | PROC | NEAR |  |
|  | MOV | CX, 20 | ；录音时间长度,5s |
|  | CALL | ISD\_REC | ;调用录音子程序 |
| KEY\_REC1: | CALL | Delay\_025S | ;延时 |
|  | CMP | bNewKey, 0 | ；检测按键是否有键按下 |
|  | JNZ | KEY\_REC2 |  |
|  | LOOP | KEY\_REC1 | ；录音时间，根据CX的值决定 |
|  | CALL | ISD\_STOP | ;停止录音 |
| KEY\_REC2: | RET |  |  |
| KEY\_REC | ENDP |  |  |
| ；放音子程序 |  |  |  |
| KEY4 | PROC | NEAR |  |
|  | MOV | AL,ISD1420\_AD0 | ；4号键放音首地址 |
|  | CALL | KEY\_PLAY |  |
|  | RET |  |  |
| KEY4 | ENDP |  |  |
| KEY5 | PROC | NEAR |  |
|  | MOV | AL, ISD142O\_AD1 |  |
|  | CALL | KEY\_PLAY |  |
|  | RET |  |  |
| KEY5 | ENDP |  |  |
| KEY6 | PROC | NEAR |  |
|  | MOV | AL, ISD1420\_AD2 | ;6号键放音首地址 |
|  | CALL | KEY\_PLAY |  |
|  | RET |  |  |
| KEY6 | ENDP |  |  |
| KEY7 | PROC | NEAR |  |
|  | MOV | AL, ISD1420\_AD3 | ；7号键放音首地址 |
|  | CALL | KEY\_PLAY |  |
|  | RET |  |  |
| KEY7 | ENDP |  |  |
| KEY\_PLAY | PROC | NEAR |  |
|  | MOV | CX, 20 |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | CALL | ISD\_PLAY | ;调用录音子程序 |
| KEY\_PLAY1: | CALL | Delay\_025S | ;用于进度显示的时间参照 |
|  | CMP | bNewKey, 0 |  |
|  | JNZ | KEY\_PLAY2 | ;检测按键是否有键按下 |
|  | LOOP | KEY\_PLAY1 |  |
| KEY\_PLAY2: | RET |  |  |
| KEY\_PLAY | ENDP |  |  |
| KeyRun | PROC | NEAR |  |
|  | LEA | BX, KeyTab | ;有键按下，跳到相应处理程序 |
|  | MOV | AL, KEYno | ;KEYno—按键值 |
|  | SHL | AL, 1 | ；X2倍 |
|  | XOR | AH, AH |  |
|  | ADD | BX, AX |  |
|  | CALL | :BX] | :[BX]=对应按键子程序入口地址 |
|  | RET |  |  |
| KeyRun | ENDP |  |  |
| ;按键扫描 |  |  |  |
| ScanKey | PROC | NEAR |  |
|  | MOV | DX, I8255\_PA | ;8255.PA——检测按键输入 |
|  | IN | AL, DX | ；键扫描 |
|  | CMP | AL, OFFH |  |
|  | JNZ | ScanKeyl |  |
| ScanKey4: | CLC |  | ；无按键按下 |
|  | RET |  |  |
| ScanKeyl: | CALL | ScanKey2 | ；有按键，取抖动处理 |
|  | JNB | ScanKey4 |  |
| ScanKey3: | MOV | BL, KEYno |  |
|  | CALL | Delay20ms | ；消抖动 |
|  | CALL | Delay20ms |  |
|  | CALL | ScanKey2 |  |
|  | JNB | ScanKey4 |  |
|  | CMP | BL, KEYno |  |
|  | JNZ | ScanKey3 |  |
| ScanKey5: | MOV | DX, I8255\_PA |  |
| ScanKey6: | IN | AL, DX |  |
|  | CMP | AL, OFFH |  |
|  | JNZ | ScanKey6 |  |
|  | STC |  |  |
|  | RET |  |  |
| ScanKey | ENDP |  |  |
| ；按下的键 | （1〜8）转化为对应的键值（0〜7）, | | ，便于多分支程序处理 |
| ScanKey2 | PROC | NEAR |  |
|  | PUSH | BX |  |
|  | XOR | BL, BL |  |

|  |  |  |
| --- | --- | --- |
|  | MOV | DX, I8255\_PA |
|  | IN | AL, DX |
|  | TEST | AL, 01H |
|  | JZ | ScanKey21 |
|  | INC | BL |
|  | TEST | AL, 02H |
|  | JZ | ScanKey21 |
|  | INC | BL |
|  | TEST | AL, 04H |
|  | JZ | ScanKey21 |
|  | INC | BL |
|  | TEST | AL, 08H |
|  | JZ | ScanKey21 |
|  | INC | BL |
|  | TEST | AL, 10H |
|  | JZ | ScanKey21 |
|  | INC | BL |
|  | TEST | AL, 20H |
|  | JZ | ScanKey21 |
|  | INC | BL |
|  | TEST | AL, 40H |
|  | JZ | ScanKey21 |
|  | INC | BL |
|  | TEST | AL, 80H |
|  | JZ | ScanKey21 |
|  | CLC |  |
|  | JMP | ScanKey22 |
| ScanKey21: | STC | ；有键按下，置有键按下标志 |
|  | MOV | KEYno, BL ；获得键值 |
| ScanKey22: | POP | BX |
|  | RET |  |
| ScanKey2 | ENDP |  |
| 涎时 |  |  |
| Delay20ms | PROC | NEAR |
|  | PUSH | CX |
|  | MOV | CX, 2640 |
|  | LOOP | $ |
|  | POP | CX |
|  | RET |  |
| Delay20ms | ENDP |  |
| ;延时0. 25s | （兼有键盘检测功能） | |
| Delay\_025S | PROC | NEAR |
|  | PUSH | AX |
|  | PUSH | CX |

DL1S\_2:

Delay\_025S

；录音子程序

|  |  |
| --- | --- |
| PUSH | DX |
| MOV | CX, 28000 |
| LOOP | $ |
| MOV | CX, 28000 |
| LOOP | $ |
| CALL | ScanKey |
| JNB | DL1S\_2 |
| MOV | bNewKey, 1 |
| POP | DX |
| POP | CX |
| POP | AX |
| RET |  |
| ENDP |  |

；AL—存放操作方式设置值，CX—录几秒

|  |  |  |
| --- | --- | --- |
| ISD\_INIT | PROC | NEAR |
|  | MOV | DX, I8255\_PC |
|  | MOV | AL, KeepMode |
|  | OR | AL, 7 |
|  | OUT | DX, AL |
|  | MOV | KeepMode, AL |
|  | MOV | DX, ISDCOMM |
|  | XOR | AL, AL |
|  | OUT | DX, AL |
|  | RET |  |
| ISD\_INIT | ENDP |  |
| ；操作模式, | AL-操作模式设置值 | |
| ISD\_MODE | PROC | NEAR |
|  | PUSH | AX |
|  | CALL | ISD\_STOP |
|  | MOV | DX, ISDCOMM |
|  | POP | AX |
|  | OUT | DX, AL |
|  | MOV | DX, I8255\_PC |
|  | MOV | AL,KeepMode |
|  | AND | AL, OFBH |
|  | OUT | DX, AL |
|  | OR | AL, 4 |
|  | OUT | DX, AL |
|  | MOV | KeepMode, AL |
|  | RET |  |
| ISD\_MODE | ENDP |  |
| ；录音 |  |  |
| ISD\_REC | PROC | NEAR |

；语音模块初始化，关闭录放音功能

；允许手动录放音，当A6, A7为高时，无法手动放音

;初始化,REC, PLAYE, PLAYL置位，设置操作模式

；设置操作模式:分段录音

；设置操作模式命令在AL中

;给卩1^丫1 一个上升沿，锁存命令

|  |  |  |  |
| --- | --- | --- | --- |
|  | MOV | DX, ISDCOMM | |
|  | OUT | DX, AL | ;设置录音起始地址 |
|  | MOV | DX, I8255\_PC |  |
|  | MOV | AL, KeepMode |  |
|  | AND | AL, OFEH |  |
|  | OUT | DX, AL | ;REC变低，即开始录音 |
| ISD\_REC  ；放音子程序  ；AL—放哪段音 | MOV  RET  ENDP | KeepMode, AL |  |
| ISD\_PLAY | PROC | NEAR |  |
|  | PUSH | AX |  |
|  | CALL | ISD\_STOP | ;暂停之前的录放音操作 |
|  | POP | AX |  |
|  | MOV | DX, ISDCOMM | ;设置放音起始地址 |
|  | OUT | DX, AL |  |
|  | MOV | DX, I8255\_PC |  |
|  | MOV | AL,KeepMode |  |
|  | AND | AL, OFDH |  |
|  | OUT | DX, AL | ；0->PLAYE开始放音,边沿放音模式 |
|  | OR | AL, 2 |  |
|  | OUT | DX, AL | :1->PLAYE |
| ISD\_PLAY  ;停止录放音 | MOV  RET  ENDP | KeepMode, AL |  |
| ISD\_ST0P | PROC | NEAR |  |
|  | MOV | DX, I8255\_PC |  |
|  | MOV | AL, KeepMode |  |
|  | AND | AL, OFBH |  |
|  | OUT | DX, AL | :PLAYL:-个负脉冲停止放音 |
|  | OR | AL, 4 |  |
|  | OUT | DX, AL |  |
|  | CALL | Delay50ms |  |
|  | OR | AL, 3 | ;1->REC,PLAYE |
|  | OUT | DX, AL | ；关闭所有操作指令 |
|  | MOV | KeepMode, AL |  |
|  | MOV | DX, ISDCOMM |  |
|  | XOR | AL, AL |  |
| ISD STOP | OUT  RET  ENDP | DX, AL | ;允许手动录放音，当A6, A7为高时，无法手动放音 |

涎时

|  |  |  |
| --- | --- | --- |
| Delay50ms | PROC | NEAR |
|  | PUSH | CX |
|  | MOV | CX, 13200 |
|  | LOOP | $ |
|  | POP | CX |
|  | RET |  |
| Delay50ms | ENDP |  |
|  | END | START |

1. 实验扩展及思考题

实验名称：公交车的报站功能

实验内容：利用掌握分段录音和放音控制，实现公交车的报站功能，有兴趣者可自行尝试

实验九X5045串行EEPROM读写实验

一、 实验目的与要求

了解SPI总线；掌握SPI总线的数据读写操作；掌握对X5045单字节读写、页写模式和连 续读取模式的操作

学会建立项目，使用多个模块文件完成一个功能，便于移植。

二、 实验设备

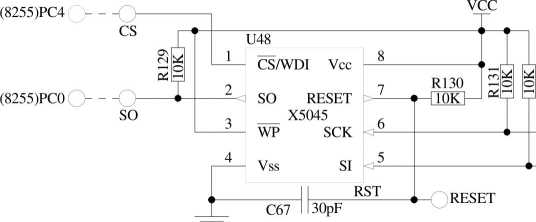
SUN系列实验仪一套、PC机一台。

三、 实验内容

1. X5045 (D6 区):
2. 内置4K bit三线串行接口 EEPROM,支持16字节页写模式和连续读取功能。
3. 内置看门狗和低电压检测功能。

2、 实验过程

1. 写满X5045内部整个4Kbit串行EEPROM,起始写入地址为00H,之后地址与数据都以 +1递增，直到写满整个EEPROM
2. 检验写入数据是否正确并显示结果，正确：点亮F4区的发光管(DS12-DS19)，错误： 熄灭发光管。

**四、 实验原理图**

O O PC5(8255)

SCK

O O PC6(8255)

SI

**五、实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| D3 区：CS、AO、A1 |  | A3 区：CS1、AO、A1 |
| D3 区：PC4、PC5、PC6 |  | D6 区：CS、SCK、SI |
| D3 区：PCO |  | D6 区:SO |
| D3 区：JP23 (PA) |  | F4 区：JP18 |

2、运行程序，正确：点亮F4区的发光管(DS12-DS19),错误：熄灭发光管。

**六、演示程序**(完整程序见目录X5045)

|  |  |  |
| --- | --- | --- |
| ；cs | PC4 | 片选，低有效 |
| ;SCLK | PC5 | 时钟输入 |
| ；SI | PC6 | 数据输入 |
| ;S0 | PCO | 数据输出 |

C0N\_8255 EQU 0273H

PC\_8255 EQU 0272H

;］、X5045 子程序(X5045. ASM)

;(1)对。5、SCLK、SI置1、清零、读SO子程序

|  |  |  |  |
| --- | --- | --- | --- |
| CS\_H | PROC | NEAR | 输出高电平 |
|  | PUSH | AX |  |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, 09H |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | RET |  |  |
| CS\_H | ENDP |  |  |
| CS\_L | PROC | NEAR | ;输出低电平 |
|  | PUSH | AX |  |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, 08H |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | RET |  |  |
| CS\_L | ENDP |  |  |
| SCLK\_H | PROC | NEAR | ;输出高电平 |
|  | PUSH | AX |  |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, OBH |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | RET |  |  |
| SCLK\_H | ENDP |  |  |
| SCLK\_L | PROC | NEAR | ;输出低电平 |
|  | PUSH | AX |  |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, OAH |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | RET |  |  |
| SCLK\_L | ENDP |  |  |
| SI\_H | PROC | NEAR | ;输出高电平 |
|  | PUSH | AX |  |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, ODH |  |
|  | OUT | DX, AL |  |
|  | POP | AX |  |
|  | RET |  |  |
| SI\_H | ENDP |  |  |
| SI\_L | PROC | NEAR | ;输出低电平 |
|  | PUSH | AX |  |
|  | MOV | DX, CON\_8255 |  |
|  | MOV | AL, OCH |  |

|  |  |  |
| --- | --- | --- |
|  | OUT  POP  RET | DX, AL  AX |
| SI\_L | ENDP |  |
| R\_SO | PROC | NEAR |
|  | PUSH | AX |
|  | MOV | DX, PC\_8255 |
|  | IN | AL, DX |
|  | TEST | AL, 01H |
|  | POP | AX |
|  | RET |  |
| R\_SO | ENDP |  |
| ；(2)串行发送8位数据， | | 用于指令/数据的发送 |
| XTRAN | PROC | NEAR |
| X\_WMid | : PUSH | DX |
|  | PUSH | CX |
|  | PUSH | AX |
|  | MOV | CX, 08H |
|  | MOV | AH, 80H |
| XTRAN1 | : CALL | SCLK\_L |
|  | TEST | AH, AL |
|  | JZ | XTRAN2 |
|  | CALL | SI\_H |
|  | JMP | XTRAN3 |
| XTRAN2 | : CALL | SI\_L |
| XTRAN3 | : CALL | SCLK\_H |
|  | ROR | AH, 1 |
|  | LOOP | XTRAN1 |
|  | POP | AX |
|  | POP | CX |
|  | POP | DX |
|  | RET |  |
| XTRAN | ENDP |  |
| ；(2) > | 串行接收8位数据, | 用于读取数据 |
| XRECE | PROC | NEAR |
| X\_RMid | : PUSH | DX |
|  | PUSH | CX |
|  | MOV | CX, 08H |
|  | XOR | AL, AL |
| XRECE1 | : ROL | AL, 01H |
|  | CALL | SCLK\_H |
|  | CALL | SCLK\_L |
|  | CALL | R\_SO |
|  | JZ | XRECE2 |

;上升沿写入

;下降沿读出

|  |  |  |
| --- | --- | --- |
|  | OR | AL, 01H |
| XRECE2: | LOOP | XRECE1 |
|  | POP | CX |
|  | POP | DX |
|  | RET |  |
| XRECE | ENDP |  |
| ；(3)读状态 |  |  |
| X\_RSDR | PROC | NEAR |
|  | CALL | SCLK\_L |
|  | CALL | CS\_L |
|  | MOV | AL, 05H |
|  | CALL | XTRAN |
|  | CALL | XRECE |
|  | CALL | CS\_H |
|  | RET |  |
| X\_RSDR | ENDP |  |

(4)写状态，控制看门狗，存储内容的保护范围

;读状态，用于查忙

|  |  |  |
| --- | --- | --- |
| X\_WRSR | PROC | NEAR |
|  | PUSH | AX |
|  | CALL | X\_WREN |
|  | CALL | SCLK\_L |
|  | CALL | CS\_L |
|  | MOV | AL, 01H |
|  | CALL | XTRAN |
|  | POP | AX |
|  | CALL | XTRAN |
|  | CALL | CS\_H |
| X\_WRSR1: | CALL | X\_RSDR |

；必须先开启允许写

；写状态寄存器命令01H

:A中放设置命令

TEST AL, 01H

JNZ X\_WRSR1

RET

X\_WRSR ENDP

；(5)允许写操作

|  |  |  |
| --- | --- | --- |
| X\_WREN | PROC | NEAR |
|  | CALL | SCLK\_L |
|  | CALL | CS\_L |
|  | MOV | AL, 06H |
|  | CALL | XTRAN |
|  | CALL | CS\_H |
|  | RET |  |
| X\_WREN | ENDP |  |
| ;(6)禁止写操作 | |  |
| X\_WRDI | PROC | NEAR |
|  | CALL | SCLK L |

；允许写指令06H

;发送8位数据子程序

X\_WBegl:

X\_WBeg

|  |  |
| --- | --- |
| CALL | CS\_L |
| MOV | AL, 04H |
| CALL | XTRAN |
| CALL | CS\_H |
| RET |  |
| ENDP |  |
| PROC | NEAR |
| PUSH | AX |
| MOV | AL, 02H |
| JNB | X\_WBegl |
| MOV | AL, OAH |
| PUSH | AX |
| CALL | X\_WREN |
| CALL | SCLK\_L |
| CALL | CS\_L |
| POP | AX |
| CALL | XTRAN |
| POP | AX |
| CALL | XTRAN |
| RET |  |

X\_WRDI

；⑺开始写

X\_WBeg

ENDP

；禁止写指令04H

；发送8位数据子程序

;写低256个字节数据

；每次要写入数据，都必须开允许写操作

；写操作

；起始地址

|  |  |  |
| --- | --- | --- |
| ;(9)结束写 | | |
| X\_WEnd | PROC | NEAR |
|  | CALL | CS\_H |
| X\_WEndl: | CALL | X\_RSDR |
|  | TEST | AL, 01H |
|  | JNZ | X\_WEndl |
|  | RET |  |
| X\_WEnd | ENDP |  |
| ；(10)开始读 |  |  |
| X\_RBeg | PROC | NEAR |
|  | CALL | SCLK\_L |
|  | CALL | CS\_L |
|  | PUSH | AX |
|  | MOV | AL, 03H |
|  | JNB | X\_RBegl |
|  | MOV | AL, OBH |
| X\_RBegl: | CALL | XTRAN |
|  | POP | AX |
|  | CALL | XTRAN |

；(8)写数据

X\_WMid与XTRAN完全…样

RET

；写数据结束后，关闭允许写

;读低2K Bit数据

;读高2K Bit数据

；起始地址

X\_RBeg ENDP

；(11)读数据

;X\_RMid 与 XRECE 完全一样

|  |  |  |
| --- | --- | --- |
| ;(12)结束读 | | |
| X\_REnd | PROC | NEAR |
| X\_REnd  ;测试X5045 | CALL  RET | CS\_H |

;写数据，X5045内的串行EEPROM分为高、低各256个字节，要分别写入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| WLOOP1 | | EQU | 10H | ;写内循环次数 |
| WLOOP2 |  | EQU | 20H | ;写外循环次数 |
| RLOOP1 |  | EQU | 10H | ;读内循环次数 |
| RL00P2 |  | EQU | 20H | ;读外循环次数 |
| TData |  | EQU | OFEH | ；启始数据FEH |
| T\_X5045\_ | WR | PROC | NEAR |  |
|  |  | MOV | AH, T\_Data | ；启始数据，之后不数据值会不断增加 |
|  |  | MOV | BL, OOH | ;串行EEPROM地址起始地址 |
|  |  | MOV | CX, WLOOP2 | ;外循环次数20H |
|  |  | MOV | AL, 30H |  |
|  |  | CALL | X\_WRSR | ；写状态寄存器 |
| XWR: |  | CLC |  | ;低256字节 |
|  |  | CMP | CX, 11H |  |
|  |  | JNB | XWR1 |  |
|  |  | STC |  | ；高256字节 |
| XWR1: |  | PUSH | ex |  |
|  |  | MOV | CX, WLOOP1 | ；内循环次数10H |
|  |  | MOV | AL, BL |  |
|  |  | CALL | X\_WBeg |  |
| XWR2: |  | MOV | AL, AH | ；欲写数据放入A |
|  |  | INC | AH |  |
|  |  | CALL | X\_WMid | ；调用写操作子程序 |
|  |  | LOOP | XWR2 |  |
|  |  | CALL | X\_WEnd | ；结束本次写操作 |
|  |  | ADD | BL,WLOOP1 |  |
|  |  | POP | ex |  |
|  |  | LOOP | XWR |  |
|  |  | RET |  |  |
| T\_X5045\_ | WR | ENDP |  |  |
| ；读数据, | 可以…次连续地读出高、低256字节串行EEPROM里的数据，并检验 | | | |
| T\_X5045\_ | RD | PROC | NEAR |  |
|  |  | LEA | DI, Buffer5045 |  |
|  |  | MOV | AH, T\_Data | ;启始检验数据 |
|  |  | MOV | BL, OOH | ;串行EEPROM地址起始地址 |

|  |  |  |  |
| --- | --- | --- | --- |
| MOV | | CX, RL00P2 | ；外循环次数20H |
|  | CLC |  |  |
|  | MOV | AL, BL | ;从低256字节的OOH开始读 |
|  | CALL | X\_RBeg | ；开始读 |
| XRD1: | PUSH | ex |  |
|  | MOV | CX, RLOOP1 | ；内循环次数10H |
| XRD2: | CALL | X\_RMid | ；调用读操作子程序 |
|  | STOSB |  |  |
|  | CMP | AL, AH |  |
|  | JNZ | XRD3 | ；比较判断写入的数据是否正确 |
|  | INC | AH |  |
|  | LOOP | XRD2 |  |
|  | POP | ex |  |
|  | LOOP | XRD1 |  |
|  | CALL | X\_REnd | ；读数据结束 |
|  | CLC |  | ;为测试结果的判断标志，C=0正确；C=1错误 |
|  | RET |  |  |
| XRD3: | CALL | X\_REnd |  |
|  | POP | ex |  |
|  | STC |  |  |
|  | RET |  |  |
| T X5045 RD | ENDP |  |  |

七、实验扩展及思考

实验内容：上面实验采用连续读写方式操作，现在对X5045中的串行EEPROM进行单字节的 写操作和读操作，进一步熟练串行数据的读写，有兴趣者可尝试。

实验十电子钟(CLOCK)

实验目的

进一步熟悉 8253、8259、8279

二、 实验设备

STAR系列实验仪一套、PC机一台。

三、 实验内容

1、 使用8253定时功能，产生0.5S的定时中断给8259

2、 在F5区的数码管上显示时间

3、 允许设置时钟初值

四、 实验步骤

1、主机连线说明:

|  |  |  |  |
| --- | --- | --- | --- |
| D3区 | :CS、 AO、 A1 |  | A3 区：CS1、AO、A1 |
| D3区 | :PCO、 PC1 |  | F5 区：KL1、KL2 |
| D3区 | :JP20 (PB)、 B、 C |  | F5 区：A、B、C |
| B3区 | :CS、 A0 |  | A3 区：CS3、A0 |
| B3区： | INT、 INTA |  | A3 区：INTR、INTA |
| B3区: | IR0 |  | C4 区：OUTO |
| C4区 | :CS (8253)、 AO、 A1 |  | A3 区：CS2、AO、A1 |
| C4区 | :GATE |  | C1 区：VCC |
| C4区 | :CLKO |  | B2 区：62. 5K |
| 2、运行程序，按F5区的F键，设置时钏 | | | P初值； |

3、观察F5区数码管上显示的时间是否正确。

**五、演示程序**

|  |  |  |  |
| --- | --- | --- | --- |
|  | .MODEL | TINY | |
| EXTRN | In i tKeyDisplay:NEAR, | | Display8:NEAR, GetKeyA:NEAR, GetKeyB:NEAR |
| 108259\_0 | EQU | 0250H |  |
| 108259\_1 | EQU | 0251H |  |
| Con\_8253 | EQU | 0263H |  |
| T0\_8253 | EQU | 0260H |  |
|  | .STACK | 200 |  |
|  | .DATA |  |  |
| halfsec | DB | 0 | ;0. 5秒计数 |
| Sec | DB | 0 | ；秒 |
| Min | DB | 0 | ；分 |
| hour | DB | 0 | ；时 |
| buffer | DB | 8 DUP(O) | ；显示缓冲区，8个字节 |
| buffer1 | DB | 8 DUP(O) | ；显示缓冲区，8个字节 |
| bNeedDisplay | DB | 0 | ；需要刷新显示 |

|  |  |  |  |
| --- | --- | --- | --- |
| number | DB | 0 | ;设置哪一位时间 |
| bFlash | DB | 0 | ;设置时是否需要刷新 |
|  | .CODE |  |  |
| START: | MOV | AX, @DATA |  |
|  | MOV | DS, AX |  |
|  | MOV | ES, AX |  |
|  | NOP |  |  |
|  | CALL | InitKeyDisplay | ；对键盘、数码管扫描控制器8255初始化 |
|  | mov | sec, 0 | ；时分秒赋初值23:58:00 |
|  | mov | min, 58 |  |
|  | mov | hour, 23 |  |
|  | MOV | bNeedDisplay, 1 | ；显示初始值 |
|  | CALL | Init8253 |  |
|  | CALL | Init8259 |  |
|  | CALL | Wrilntver |  |
|  | STI |  |  |
| MAIN: | CALL | GetKeyA | ；按键扫描 |
|  | JNB | Mainl |  |
|  | CMP | AL, OFH | ；设置时间 |
|  | JNZ | Mainl |  |
|  | CALL | SetTime |  |
| Mainl: | CMP | bNeedDisplay, 0 |  |
|  | JZ | MAIN |  |
|  | CALL | Display\_LED | ；显示时分秒 |
|  | MOV | bNeedDisplay, 0 | ；Is定时到刷新转速 |
| Main2: | JMP | MAIN | ;循环进行实验内容介绍与测速功能测试 |
| SetTime | PROC | NEAR |  |
|  | LEA | SI, buffer1 |  |
|  | CALL | TimeToBuffer |  |
|  | MOV | Number, 0 |  |
| Key: | CMP | bFlash, 0 |  |
|  | JZ | Key2 |  |
|  | LEA | SI, buffer1 |  |
|  | LEA | DI,buffer |  |
|  | MOV | CX, 8 |  |
|  | REP | MOVSB |  |
|  | CMP | halfsec, 0 |  |
|  | JNZ | FLASH |  |
|  | MOV | BL, number |  |
|  | NOT | BL |  |
|  | AND | BX, 07H |  |
|  | LEA | SI, buffer |  |
|  | MOV | BYTE PTR [SI+BX], 10H | ；当前设置位置产生闪烁效果 |
| FLASH: | LEA | SI, buffer |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | CALL  MOV | Display8  bFlash, 0 | |
| Key2: | CALL | GetKeyA |  |
|  | JNB | Key |  |
|  | CMP | AL, OEH | ;放弃设置 |
|  | JNZ | Keyl |  |
|  | JMP | Exit |  |
| Keyl: | CMP | AL, OFH |  |
|  | JZ | SetTime8 |  |
| SetTimel: | CMP | AL, 10 |  |
|  | JNB | Key | ;无效按键 |
|  | CMP | number, 0 |  |
|  | JNZ | SetTime2 |  |
|  | CMP | AL, 3 | ;调整时的十位数 |
|  | JNB | Key |  |
|  | MOV | bufferl + 7, AL |  |
|  | JMP | SetTime7 |  |
| SetTime2: | CMP | number, 1 |  |
|  | JNZ | SetTime3 |  |
|  | CMP | bufferl + 7, 1 | ;调整时的个位数 |
|  | JZ | SetTime2\_l |  |
|  | CMP | AL, 4 |  |
|  | JNB | Key |  |
| SetTime2\_l: | MOV | bufferl + 6, AL |  |
|  | INC | number |  |
|  | JMP | SetTime7 |  |
| SetTime3: | CMP | number, 3 |  |
|  | JNZ | SetTime4 |  |
|  | CMP | AL, 6 | ;调整分的十位数 |
|  | JNB | Key |  |
|  | MOV | bufferl + 4, AL |  |
|  | JMP | SetTime7 |  |
| SetTime4: | CMP | number, 4 |  |
|  | JNZ | SetTime5 |  |
|  | MOV | bufferl + 3, AL | ;调整分的个位数 |
|  | INC | number |  |
|  | JMP | SetTime7 |  |
| SetTime5: | CMP | number, 6 |  |
|  | JNZ | SetTime6 |  |
|  | CMP | AL, 6 | ;调整秒的十位数 |
|  | JB | SetTime5\_l |  |
|  | JMP | Key |  |
| SetTime5\_l: | MOV | bufferl + 1, AL |  |
|  | JMP | SetTime7 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| SetTime6: | MOV | bufferl, AL | ;调整秒的个位数 |
| SetTime7: | INC | number |  |
|  | CMP | number, 8 |  |
|  | JNB | SetTime8 |  |
|  | MOV | bFlash, 1 | ;需要刷新 |
|  | JMP | Key |  |
| SetTime8: | MOV | AL, bufferl + 1 | ;确认 |
|  | MOV | BL, 10 |  |
|  | MUL | BL |  |
|  | ADD | AL, bufferl |  |
|  | MOV | sec, AL | ;秒 |
|  | MOV | AL, bufferl + 4 |  |
|  | MUL | BL |  |
|  | ADD | AL, bufferl + 3 |  |
|  | MOV | min, AL | ;分 |
|  | MOV | AL, bufferl + 7 |  |
|  | MUL | BL |  |
|  | ADD | AL, bufferl + 6 |  |
|  | MOV | hour, AL | ;时 |
|  | JMP | Exit |  |
| Exit: | RET |  |  |
| SetTime | ENDP |  |  |
| ;hour min | sec转化成可显示格式 | |  |
| TimeToBuffer PROC | | NEAR |  |
|  | MOV | AL, sec |  |
|  | XOR | AH, AH |  |
|  | MOV | BL, 10 |  |
|  | DIV | BL |  |
|  | MOV | [SI], AH |  |
|  | MOV | [SI + 1], AL | ;秒 |
|  | MOV | BYTE PTR [SI + 2],10H | ;这位不显示 |
|  | MOV | AL, min |  |
|  | XOR | AH, AH |  |
|  | DIV | BL |  |
|  | MOV | [SI + 3], AH |  |
|  | MOV | [SI + 4], AL | ;分 |
|  | MOV | BYTE PTR [SI + 5],10H | ;这位不显示 |
|  | MOV | AL, hour |  |
|  | XOR | AH, AH |  |
|  | DIV | BL |  |
|  | MOV | [SI + 6], AH |  |
|  | MOV | [SI + 7], AL | ;时 |

RET

TimeToBuffer ENDP

|  |  |  |
| --- | --- | --- |
| ;显示时分秒 | | |
| Display LED | PROC | NEAR |
|  | LEA | SI, buffer |
|  | CALL | TimeToBuffer |
|  | LEA | SI, buffer |
|  | CALL | Display8 |
|  | RET |  |
| Display\_LED | ENDP |  |
| ;0. 5s产生一次中断 | |  |
| TimerOInt: | PUSH | AX |
|  | PUSH | DX |
|  | MOV | bFlash, 1 |
|  | INC | halfsec |
|  | CMP | halfsec, 2 |
|  | JNZ | TimerOIntl |
|  | MOV | bNeedDisplay,1 |
|  | MOV | halfsec, 0 |
|  | INC | sec |
|  | CMP | sec, 60 |
|  | JNZ | TimerOIntl |
|  | MOV | sec, 0 |
|  | INC | min |
|  | CMP | min, 60 |
|  | JNZ | TimerOIntl |
|  | MOV | min, 0 |
|  | INC | hour |
|  | CMP | hour,24 |
|  | JNZ | TimerOIntl |
|  | MOV | hour, 0 |
| TimerOIntl: | MOV | DX, I08259\_0 |
|  | MOV | AL, 20H |
|  | OUT | DX, AL |
|  | POP | DX |
|  | POP | AX |
|  | IRET |  |
| Init8253 | PROC | NEAR |
|  | MOV | DX, Con\_8253 |
|  | MOV | AL, 34H |
|  | OUT | DX, AL |
|  | MOV | DX, T0\_8253 |
|  | MOV | AL, 12H |
|  | OUT | DX, AL |
|  | MOV | AL, 7AH |
|  | OUT | DX, AL |

；计数器TO设置在模式2状态,HEX计数

;CLKO二62. 5kHz, 0. 5s定时

|  |  |  |
| --- | --- | --- |
| RET | | |
| Init8253 | ENDP |  |
| Init8259 | PROC | NEAR |
|  | MOV | DX, I08259\_0 |
|  | MOV | AL, 13H |
|  | OUT | DX, AL |
|  | MOV | DX, I08259\_l |
|  | MOV | AL, 08H |
|  | OUT | DX, AL |
|  | MOV | AL, 09H |
|  | OUT | DX, AL |
|  | MOV | AL, OFEH |
|  | OUT | DX, AL |
|  | RET |  |
| Init8259 | ENDP |  |
| Wrilntver | PROC | NEAR |
|  | PUSH | ES |
|  | MOV | AX, 0 |
|  | MOV | ES, AX |
|  | MOV | DI, 20H |
|  | LEA | AX, TimerOInt |
|  | STOSW |  |
|  | MOV | AX, CS |
|  | STOSW |  |
|  | POP | ES |
|  | RET |  |
| Wrilntver | ENDP |  |
|  | END | START |

实验■一直流电机调速实验

一、 实验目的

了解光电开关测速原理；掌握使用光电开关测量直流电机转速。

**二、 实验设备**

SUN系列实验仪一套、PC机一台。

**三、 实验内容**

1、 转速控制原理

将设置的转速与当前测量的转速比较，得出差值用于控制DAC0832的输出电压，从而控制 直流电机的转速，使转速逐渐达到设置转速。

2、 实验过程

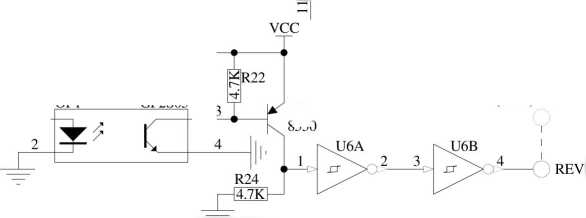
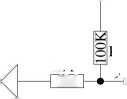
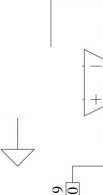
（1） 将当前转速与设置转速（要求达到的转速）相比较，得出差值来调整DAC0832的输出电 压，逐步将转速控制到设置转速。

（2） 在LED上显示设置转速（左4位LED）和当前转速（右4位LED）,转速显示采用十进制。

控制过程中，当前转速显示不断变化。

**四、实验原理图**

直流电机转速范围3—99r/s,误差±lr/s



CO

| outQ O~i | |  |
| --- | --- | --- |
| DAC0832(D2) |  |  |
|  | w2rT |  |

A —

"U18A

LM324N

10

R72 寸

R71

2M

R90 -TO

直流电机（M

(8259)IR7

GP2S05

3

Q5

8550

Q6

SC2655

7 LM324N

-

^1J18C

Qpi

**五、实验步骤**

1、主机连线说明:

|  |  |  |
| --- | --- | --- |
| B3 区：CS、AO |  | A3 区：CS3、AO |
| B3 区：INT、INTA |  | A3 区:INTR、INTA |
| C4 区：CS、AO、Al |  | A3 区：CS2、AO、Al |
| C4 区：GATE |  | Cl 区：VCC |
| C4 区：CLKO |  | B2 区：31250Hz |
| C4 区:CLK1 |  | B2 区：IM |
| C4 区：OUTO |  | B3 区：IRO |
| D2 区：CS |  | A3 区：CS4 |
| D2 区：OUT |  | F3 区：INI |
| F3 区：0UT1 |  | El 区：CTRL |
| El 区：REV |  | B3 区:IR7 |
| D3 区：CS、AO、Al |  | A3 区：CS1、AO、Al |

D3 区：JP20(PB)、B、C —— F5 区：A、B、C

|  |  |  |
| --- | --- | --- |
| D3 区：PCO、PC1 |  | F5 区：KL1、KL2 |

2、 设置要求达到的转速，显示在LED上（左4位）；测量当前转速，显示在LED上（右4位）

3、 比较设置转速与测量的当前转速，得出差值，用于调整DAC0832的输出电压，控制电机 转速达到设置的转速。可以看到LED上显示的当前转速迅速靠近设置转速。

1. **演示程序**（完整程序RateCtrl. asm）

1、 基本控制程序

（1） DAC0832程序请参阅“并行DA实验”；（2）键盘、数码管扫描可参阅“基础实验九”，也可 以调用库函数。

2、 主程序（MAIN. ASM）

（1） 转速测量

转速测量请参阅“直流电机测速实验”。（0.25s测速一次，控制一次）

（2） 转速控制

|  |  |  |  |
| --- | --- | --- | --- |
| REVControl | PR0C | NEAR | |
|  | MOV | AL, Count | ;当前转速 |
|  | CMP | AL, RevSet | ;设置转速 |
|  | JZ | REVControl1 |  |
|  | JNB | RevDEC |  |
| ；提高转速 |  |  |  |
| RevINC: | MOV | Count500ms, 1 |  |
|  | MOV | AL, RevSet | ;设置转速 |
|  | SUB | AL, Count | ;当前转速 |
| RevINCl: | ADD | AL, Data\_0832 | ;转速差值+上一次DAC0832输入值=DAC0832输入值 |
|  | JNB | RevINC2 | ；判断是否超过DAC0832最大输入值 |
|  | MOV | AL, OFFH |  |
| RevINC2: | MOV | Data\_0832, AL |  |
|  | CALL | DAC0832 | :D/A,调整DAC0832输出电压 |
|  | JMP | REVControl1 |  |
| ；降低转速 |  |  |  |
| RevDEC: | MOV | AL, Count | ；当前转速 |
|  | SUB | AL, RevSet | ；设置转速 |
|  | CMP | AL, 40 |  |
|  | JNB | RevDECl |  |
|  | MOV | AH, Count500ms |  |
|  | OR | AH, AH |  |
|  | JZ | RevDEC3 |  |
|  | MOV | Count500ms, 0 |  |
| RevDECl: | XCHG | AL, Data\_0832 | ;上一次DAC0832输入值-转速差值=DAC0832输入值 |
|  | SUB | AL,Data\_0832 |  |
|  | JNB | RevDEC2 |  |
|  | MOV | AL, 10 |  |
| RevDEC2: | MOV | Data\_0832, AL |  |
|  | CALL | DAC0832 | :D/A,调整DAC0832输出电压 |
|  | JMP | REVControl1 |  |
| RevDEC3: | INC | Count500ms |  |

REVControll: RET

REVControl ENDP

1. **实验扩展及思考题**

实验内容：本实验采用差值法控制转速，现请使用其他的方法控制转速，实现更精确、快 速的转速控制。

实验十二光敏电阻测量光照强度实验

一、 实验目的

1、 了解如何利用光敏电阻测量光照强度。

2、 掌握使用光敏电阻和ADC0809制作简易的光照强度测量仪。

**二、 实验设备**

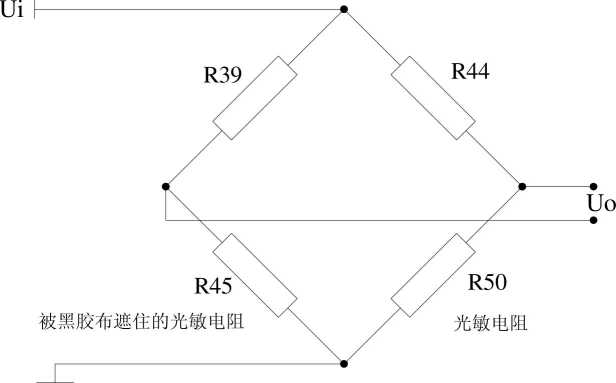
SUN系列实验仪一套、PC机一台。

**三、 实验内容**

1、光照强度测量原理：

1. 在实验仪上有两个光敏电阻，其中有一个光敏电阻表面贴上黑胶布，让其不能对光照 的变化产生明显的电阻值改变。另外一个光敏电阻没有贴上黑胶布能迅速地感应光照强度的变 化。本实验利用光敏电阻测量光照强度，外界光照强度发生变化引起了光敏电阻的阻值改变。
2. 光敏电阻R41与完全相同的光敏电阻R57 (被黑胶布遮住)组成的电桥(图1), ARx 为照度变大时电阻的增加值。将电阻变化转换为电桥产生的压差U。的变化，经放大电路将产生 的压差U。放大，得到可以被A/D转换的电压，整个系统的实际电路图如图2所示。
3. 本实验假设光照强度与产生的压差U。成正比，设定某一标准光照强度与某一压差对应, 以此对应关系换算产生其它压差的照度。从而达到测量光照强度的目的。

如果你需要精确测量光照强度，先将电压值与光照强度对应关系制成表，根据测得电压值 通过查表，得出光照强度值。

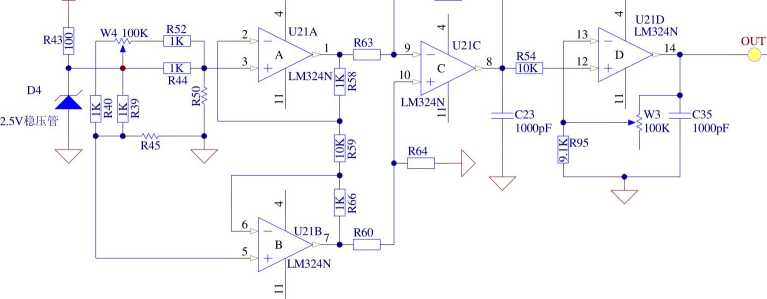


**Uo =** Vi X { Rx/(Rx + l)-(Rx + ARx)/ (Rx + ARx + 1)}

图1电压差产生原理图

2、实验过程

1. 由光照强度产生的模拟电压信号转换为数字信号，然后转换为照度(单位是勒克斯)显 示在LED±；
2. 校准照度测量器：在一定的光强度下，产生200数字量的电压，以此对应关系(照度 —电压)将其它光强度转换为勒克斯值，显示在LED上。



vcc

R55

IN0(ADC0809)

图2实验的电路图

**四、实验步骤**

1、连线说明：

|  |  |  |
| --- | --- | --- |
| C2 区：CS、ADDA、ADDB、ADDC |  | A3 区：CS3、AO、Al、A2 （选择通道） |
| C2 区：CLK |  | B2 区：500K |
| C2 区：IN0 |  | E2 区：OUT |
| D3 区：CS、AO、A1 |  | A3 区：CS1、A0、A1 |
| D3 区：PCO、PC1 |  | F5 区：KL1、KL2 |
| D3 区：JP20、B、C |  | F5 区：A、B、C |

2、 两个光敏电阻都处于遮住状态，通过调节电位器W4,使显示为零。

3、 调节电位器W3 （放大系数），核准室内某一光照强度一200电压数字量的对应关系。

4、 用其他物体在光敏电阻上方改变它的光强度，可以看到LED上（最右边3位）显示照度 的大小（在此以勒克斯表示）。随着光照强度的变化，LED显示数值跟着变化。

说明：光敏电阻与其它半导体光电器件相比有很多优点，例如所测的光电强度范围宽、灵 敏度高、无选择极性之分等等，但是也有其不足的地方，特别是在强光的条件下光电线性度较 差、光电迟豫过程较长、频率特性不够好等。所以这个实验要避免在强光下进行操作，譬如要 避免在太阳光直射下操作。

大多数室内照度都在300Lux（勒克斯）以下，一般的教室内部照度大约在150Lux左右，黄 昏室内照度大约为lOLux,而在夏日太阳强光之下可以达到100, OOOLux。

本实验电路采用两个完全相同的光敏电阻来组成电桥，为的是减小误差，选用差放电路也 是基于这个目的。然而实际上并不能完全消除零点漂移，差放电路在达到一定的放大值的时候, 在最后的电压输出端产生了 100mv左右的零点漂移值，这也是实际电路都无法避免的。

1. 演示程序

略

附录A星研集成环境软件支持的软中断

|  |  |  |
| --- | --- | --- |
| **输入** | **功能** | **输出** |
| INT 21H软中断 | | |
| AH 二 01H, AL = 00H | 带回显的字符输入  （无字符输入时将等待输入） | AL =键值 |
| AH 二 01H, AL = 01H | 带回显的16进制数输入  （无输入时将等待输入） | AL =键值 |
| AH 二 01H, AL = 02H | 带回显的10进制数输入  （无输入时将等待输入） | AL =键值 |
| AH 二 02H,  DL二8位数据（通常是ASCII代码） | 字符输出（输出一字符到信息窗的  Dos标签视中） |  |
| AH 二 06H, DL二0FFH | 请求输入  （它与INT 21H, AH=01H区别，它 不会等待键输入） | 如果之前有键按下： 零标志二清除，AL二键值； 如果之前没有键按下， 零标志二置1 |
| AH 二 06H,  DL二 00H-0FEH（通常是 ASCII 代码） | 字符输出（输出一字符到信息窗的  Dos标签视中） |  |
| AH = 09H  DS： DX二段：偏移地址 | 输出字符串（送一字符串到信息窗 的Dos标签视中，字符串以'字 符（24H）结尾 |  |
| AH = OAH AL = 0（接收任意字符） DS： DX =段：偏移地址 缓冲区的第一个字节存放它能保 存的最大字符数（1至255） | 缓冲输入（从键盘读一行并放入用 户定义的缓冲区） | 缓冲区的第一个字节说明它能保 存的最大字符数（1至255）,该值 由用户设置，第二个字节返回实际 输入的字符数（回车除外） |
| AH = OAH AL = 0（接收16进制数） DS： DX =段：偏移地址 缓冲区的第一个字节存放它能保 存的最大字符数（1至255） | 缓冲输入（从键盘读一行并放入用 户定义的缓冲区） | 缓冲区的第一个字节说明它能保 存的最大字符数（1至255）,该值 由用户设置，第二个字节返回实际 输入的字符数（回车除外） |
| AH = OAH AL = 2 （接收10进制数） DS： DX =段：偏移地址 缓冲区的第一个字节存放它能保 存的最大字符数（1至255） | 缓冲输入（从键盘读一行并放入用 户定义的缓冲区） | 缓冲区的第一个字节说明它能保 存的最大字符数（1至255）,该值 由用户设置，第二个字节返回实际 输入的字符数（回车除外） |
| AH *二* 0BH | 取输入状态（检查是否可以从键盘 缓冲区取一个字符） | AL = 00H （没有输入），  AL = 0FFH （有字符输入） |
| AH 二 0CH | 先清键盘缓冲区，  然后，如果 AL = 01H、06H、07H、  08H、0AH,相当于 INT21H, AH = AL |  |
| AH = 25H, AL =中断号  DS： DX =中断处理过程段：位移 | 置中断向量 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| AH 二 | 35H, AL =中断号 | 取中断向量（得到当前中断处理程 序地址） | ES： BX =中断处理程序段：位移 |
| AH 二 | 4CH | 带返回码结束程序 |  |
| AH 二  EDI | OFEH, AL = 0  二存放缓冲区首地址 | 读取GDTR寄存器 | EDI：缓冲区首地址 |
| AH 二 OFEH, AL 二 1  EBX =全局描述符表首地址 ex二读取长度  EDI二存放缓冲区首地址 | | 读取全局描述符表 | EDI：缓冲区首地址 |
| AH 二 | OFEH, AL = 2 | 如果操作系统是Winxp SP2或更新， 返回：AL=1；操作系统比Winxp SP2 来得早，返回：AL二0 | 操作系统是Winxp SP2或更新，返 回：AL=1；操作系统比Winxp SP2 来得早，返回：AL二0 |
| AH = OFEH, AL 二 3,  EDI二存放缓冲区首地址 | | 读CR3 | EDI：缓冲区首地址（存放CR3） |
| AH = OFEH, AL 二 4  CX二读取字节数，  EBX =物理地址，  EDI二存放缓冲区首地址 | | 读内存 | CY = 1,读取成功；  CY = 0,读取失败 |
| AH 二 OFEH, AL 二 5  EDI二存放缓冲区首地址 | | 读PCI9052板卡三块Memory对应的 虚拟地址、物理地址 | 第一个双字：虚拟地址—— PCI9052 MEMORY基地址（用于访问 局部配置寄存器）  第二个双字：物理地址  PCI9052 MEMORY 基地址  第三个双字：虚拟地址—— PCI9052板卡上8位Memory空间 基地址  第四个双字： 物理地址—— PCI9052板卡上8位Memory空间 基地址  第五个双字：虚拟地址  PCI9052板卡上32位Memory空间 基地址  第六个双字：物理地址—— PCI9052板卡上32位Memory空间 基地址 |
| AH 二 | OFEH, AL 二 6 | 读当前系统的DS | DS->EAX |
| AH 二 | OFFH, DX二多少毫秒 | 延时DX毫秒 |  |
| INT 10H软中断 | | | |
| AH 二 | 0 | 清屏 |  |
| AH 二 | 2, DH 二行，DL 二列 | 设置光标位置  （用字符坐标确定光标位置） |  |
| AH 二 | 3 | 读光标位置 | DH 二行，DL =列 |

附录B数码管显示、键扫描库程序说明

1、InitKeyDisplay初始化键盘、LED控制器8255

|  |  |  |
| --- | --- | --- |
| 2、 Display8 | 显示子程序（8255对数码管扫描） | |
| 输入参数： | SI—指向8字节显不缓冲区；  如果需要显示小数点，8位16进制数的最高位为1, 如果某位不需要显示，符值10H；  如果需要显示负号符值11H | 例如：80H； |
| 例子： | 10H, 10H, 03H, 82H, OOH, OOH, OOH, 00H 显示为：“ | 32.0000” |
| 3、 Display8A  输入参数： | 显示子程序（简单I/O的二片74HC273对数码管扫描）  SI—指向8字节显示缓冲区；  如果需要显示小数点，8位16进制数的最高位为1,例如：80H；  如果某位不需要显示，符值10H；  如果需要显示负号符值11H | |
| 例子： | 10H, 10H, 03H, 82H, OOH, OOH, OOH, 00H 显示为：“ | 32.0000” |

接收一组压缩BCD码键值

4、 GetBCDKey

输入参数:

DI —指向接收缓冲区

CX -接收键值数量

F1 - 0 :先清除显示，再接收键输入

-1 :接收到第一个按键后，先清除原来显示的内容，再显示键值

|  |  |  |  |
| --- | --- | --- | --- |
| 变量F1已在库文件中定义 | | | |
| 例子： | EXTRN | Fl:BYTE | :Fl已在库文件中定义 |
|  | MOV | Fl,0 | ;先清除显示，再接收键输入 |
|  | LEA | DI, augend |  |
|  | MOV | CX,4 | ；按键次数 |
|  | CALL | GetBCDKey | ；得到双字节十进制数 |

5、GetKey 接收一组压缩16进制键值

输入参数： DI —指向接收缓冲区

CX -接收键值数量

F1 - 0 :先清除显示，再接收键输入

-1 :接收到第一个按键后，先清除原来显示的内容，再显示键值 变量F1已在库文件中定义

接收一个16进制键值，如果没有按键，立即返回

6、 GetKeyA

输出:

CY — 0,没有按键；

CY — 1, AL—键值

7、 GetKeyB

接收一个16进制键值，如果没有按键，一直等待

AL —键值  
以太网、USB1.1、USB2.0、CAN、GPS、GPRS等模块说明请参阅光盘中说明

输出: