

# 0 目录

第一章 简介.....	1
1.1 实验仪介绍.....	1
1.2 实验仪配置方案.....	1
1.3 功能特点.....	1
1.3.1 软件.....	1
1.3.2 硬件.....	2
第二章 硬件结构.....	4
2.1 电路外观.....	4
2.2 A1 区: 12864 液晶显示模块电路.....	5
2.3 A2 区: 16×16 LED 实验电路.....	5
2.4 A3 区: CPU 总线、片选区.....	6
2.5 A4 区: 控制区.....	6
2.6 B1 区: 语音模块 ISD1420 电路.....	6
2.7 B2 区: 逻辑笔、单脉冲、频率发生器.....	7
2.8 B3 区: 8259 电路.....	8
2.9 B4 区: 8155、8255 电路.....	8
2.10 B5、C6 区: 扩展区.....	9
2.11 C1 区: 电源区.....	9
2.12 C2 区: 93C46.....	9
2.13 C3 区: 138 译码器.....	9
2.14 C4 区: X5045.....	10
2.15 C5 区: 8253、8251.....	10
2.16 D1 区: 蜂鸣器.....	10
2.17 D2 区: 0~5V 电压输出.....	11
2.18 D3 区: 光敏电阻、压力测量.....	11
2.19 D4 区: 并串转换.....	12
2.20 D5 区: 串并转换.....	12
2.21 E1 区: 步进电机.....	12
2.22 E2 区: PWM 电压转换.....	12
2.23 E3 区: 继电器.....	13
2.24 E4 区: I <sup>2</sup> C 总线(包括 24C02A, PCF8563P, ZLG7290).....	13
2.25 E5 区: 8279 键盘/LED 控制器.....	14
2.26 E6 区: 8250.....	15
2.27 E7 区: RS232.....	15
2.28 E8 区: RS485.....	15
2.29 F1 区: 直流电机转速测量/控制.....	16
2.30 F2 区: 串行 AD.....	16
2.31 F3 区: DAC0832 数模转换.....	17
2.32 G1 区: 温度测量/控制.....	17

2. 33	G2 区: 红外通讯 .....	17
2. 34	G3 区: 串行 DA .....	18
2. 35	G4 区: ADC0809 模数转换 .....	18
2. 36	G5 区: 键盘&LED .....	19
2. 37	G6 区: 发光管、按键、开关 .....	19
2. 38	G7 区: 接触式 IC 卡 .....	20
第三章	星研集成环境软件 .....	21
3. 1	软件安装 .....	21
3. 1. 1	安装星研集成环境软件 .....	21
3. 1. 2	软件卸载 .....	21
3. 1. 3	USB 驱动程序 .....	21
3. 1. 4	软件启动 .....	22
3. 1. 5	编译器 .....	23
3. 1. 6	README 文件 .....	23
3. 2	如何使用星研集成环境软件 .....	23
3. 2. 1	数据传送程序 (ASM) .....	24
3. 2. 2	数据传送程序 (C) .....	37
第四章	软件实验 .....	43
实验一	数据传送 .....	43
实验二	双字节 BCD 码(十进制数)加法 .....	45
实验三	双字节 BCD 码(十进制数)减法 .....	48
实验四	四字节十六进制数转十进制数 .....	51
实验五	散转 .....	54
实验六	冒泡排序 .....	56
实验七	二分查找法 .....	58
第五章	基础硬件实验 .....	错误!未定义书签。
实验一	8255 控制交通灯实验 .....	错误!未定义书签。
实验二	74HC138 译码器实验 .....	错误!未定义书签。
实验三	8155 输入、输出、SRAM 实验 .....	错误!未定义书签。
实验四	8253 方波实验 .....	错误!未定义书签。
实验五	8259A 中断控制器实验 .....	错误!未定义书签。
实验六	8251 可编程通信实验(与微机) .....	错误!未定义书签。
实验七	8250 可编程通信实验(与微机) .....	错误!未定义书签。
实验八	8279 键盘显示实验 .....	错误!未定义书签。
实验九	并行 DA 实验 .....	错误!未定义书签。
实验十	并行 AD 实验(数字电压表实验) .....	错误!未定义书签。
实验十一	红外通信实验 .....	错误!未定义书签。
实验十二	图形点阵显示实验 .....	错误!未定义书签。
实验十三	8237 DMA 传输实验 .....	错误!未定义书签。
第六章	综合实验 .....	错误!未定义书签。

实验一	简易电子琴实验.....	错误!未定义书签。
实验二	LED16 * 16 点阵实验 .....	错误!未定义书签。
实验三	数字式温度计实验(18B20).....	错误!未定义书签。
实验四	步进电机实验.....	错误!未定义书签。
实验五	直流电机测速实验.....	错误!未定义书签。
实验六	旋转图形实验.....	错误!未定义书签。
实验七	ISD1420 语音模块实验 .....	错误!未定义书签。
实验九	电子钟(CLOCK).....	错误!未定义书签。
实验十	光敏电阻测量光照强度实验.....	错误!未定义书签。

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

# 简介

## 1.1 实验仪介绍

STAR ES598PCI 实验仪提供了几乎所有最实用、新颖的接口实验，提供详尽的 C、汇编例子程序、使用说明，不但可以满足各大专院校进行单片机、微机原理课程的开放式实验教学，也可以让参加电子竞赛的学生熟悉各种类型的接口芯片，做各种实时控制实验，轻松面对电子竞赛；也可以让刚参加工作的电子工程师迅速成为高手。

STAR ES598PCI 提供实验仪与微机同步演示功能，方便实验室老师的教学、演示。提供一个库文件，如果学生上机时间有限，只需编写最主要的程序，其它调用库文件即可。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

STAR ES8688 有二种配置方案：

- 1、实验仪主机、ES8688 模块、仿真模块（不含逻辑分析功能仪、实时跟踪仪功能）。
- 2、实验仪主机、ES8688 模块、仿真模块（带有逻辑分析功能仪、实时跟踪仪功能）。

逻辑分析功能：通过观察采样到的波形，可以让学生了解 CPU 执行指令的完整过程，加深对教科书上波形图的认识。

实时跟踪仪：记录程序运行轨迹。

## 1.3 功能特点

### 1.3.1 软件

1、提供我公司自主知识产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

◇ 集编辑器、项目管理、启动编译、连接、错误定位、下载、调试于一体，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。

◇ 完全 VC++ 风格。支持 C、PL/M、宏汇编：同时支持 Keil 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。

◇ 支持 ASM（汇编）、C、PLM 语言，多种语言多模块混合调试，文件长度无限制。

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

◇ 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）。

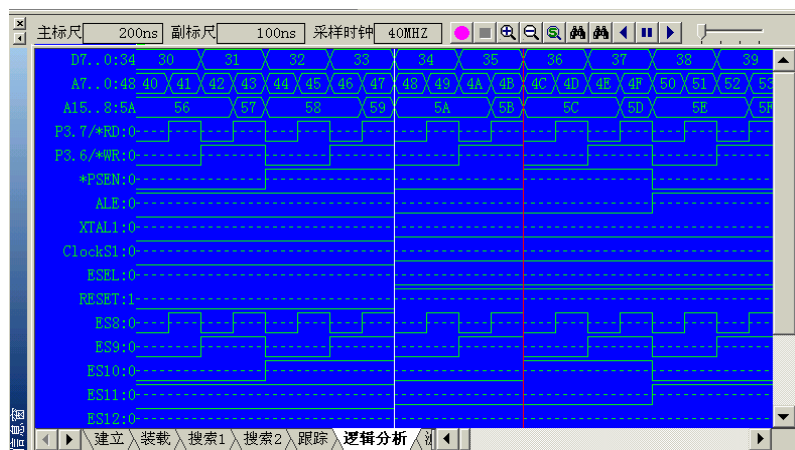
◇ 无须点击的感应式鼠标提示功能。

◇ 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。
- ◇ 提供模拟调试器。
- ◇ 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。

3、提供几十个实验的汇编、C 版本的源文件。提供一个库文件,如果学生上机时间有限,只需编写最主要的程序,其它调用库文件即可。

实验仪可提供以下软件实验:十进制数加法,十进制数减法,四字节二进制数转十进制数,数据传送,冒泡排序,二分查找法,散转等。



逻辑分析图

帧号	时间	反汇编	源文件	文件名, 行号
248	000175	290E D8FD DJNZ	R0, 290DH	
249	000176	290D F6 MOV	@R0, A	
250	000178	290E D8FD DJNZ	R0, 290DH	
251	000179	290D F6 MOV	@R0, A	
252	00017b	290E D8FD DJNZ	R0, 290DH	
253	00017c	290D F6 MOV	@R0, A	
254	00017e	290E D8FD DJNZ	R0, 290DH	
255	00017f	290D F6 MOV	@R0, A	
256	000181	290E D8FD DJNZ	R0, 290DH	
257	000183	2910 758121 MOV	SP, #21H	
258	000185	2913 02000E LJMP	000EH	
259	000187	000E 1228FD LCALL	28FDH	
260	000189	28FD 759852 MOV	SCON, #52H	c:\xingyan\examples\keil\whets\whets.c, 41
261	00018b	2900 758920 MOV	TMOD, #20H	c:\xingyan\examples\keil\whets\time.c, 28
262	00018d	2903 758869 MOV	TCON, #69H	c:\xingyan\examples\keil\whets\time.c, 30
263	00018f	2906 758DF3 MOV	TH1, #F3H	c:\xingyan\examples\keil\whets\time.c, 31
264	000191	2909 22 RET		c:\xingyan\examples\keil\whets\time.c, 32
				c:\xingyan\examples\keil\whets\time.c, 33

实时跟踪图

## 1.3.2 硬件

### 1、传统实验

74HC244、74HC273 扩展简单的 I/O 口;蜂鸣器驱动电路;74HC138 译码;8250 串行通讯实验;8251 串行通讯实验;RS232 和 RS485 接口电路;8155、8255 扩展实验;8253 定时、分频实验;128\*64 液晶点阵显示模块;16X16LED 点阵显示模块;键盘 LED 控制器 8279,并配置了 8 位 LED、4 \* 4 键盘;32K 数据 RAM 读写,使用 C 编制较大实验成为可能;并行 AD 实验;并行 DA 实验;光电耦合实验;直流电机控制;步进电机控制;继电器控制实验;逻辑笔;打印机实验;电子琴实验;74HC4040 分频得到十多种频率;另外提供 8 个拨码盘、8 个发光二极管、8 个独立按键;单脉冲输出。

### 2、新颖实验

录音、放音模块实验;光敏电阻、压力传感器实验;

#### 串行接口实验

1) 一线 DALLAS 公司的 DS18B20 测温实验

- 2) 红外通信实验
- 3) CAN CAN2.0 (扩展模块)
- 4) USB USB1.1、USB2.0 (扩展模块)
- 5) 以太网 10M 以太网模块 (扩展模块)
- 6) 蓝牙 (扩展模块)

### 3、闭环控制

- 1) 门禁系统实验
- 2) 光敏电阻、压力传感器实验
- 3) 旋转图形展现实验
- 4) 直流电机转速测量, 使用光电开关测量电机转速
- 5) 直流电机转速测量, 使用霍尔器件测量电机转速
- 6) 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
- 7) 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、实验扩展区, 提供扩展实验接口, 用户可自行设计实验

可以提供 USB1.1、USB2.0、USB 主控、10M 以太网接口的 TCP/IP 实验模块、CAN 总线、NAND FLASH 模块、FV\_VF 模块、触摸屏、非接触式 IC 卡、双通道虚拟示波器、虚拟仪器、读写优盘、CPLD、FPGA 模块。其它模块正在陆续推出中, 例如: 超声波测距、测速; GPS; GPRS; 蓝牙。

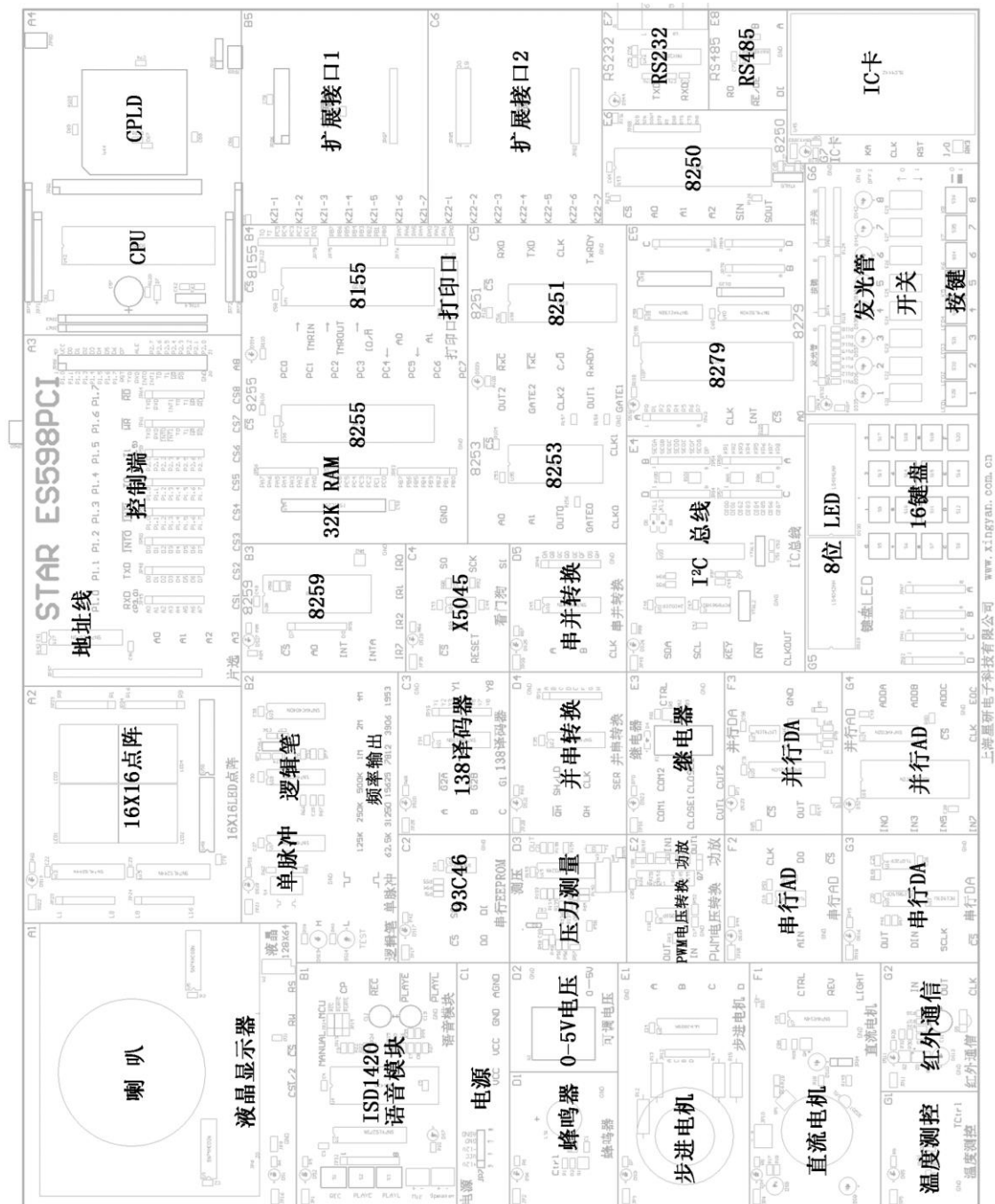
### 5、EDA —— CPLD、FPGA 可编程逻辑实验

逻辑门电路: 与门、或门、非门、异或门、锁存器、触发器、缓冲器等; 半加器、全加器、比较器、二、十进制计数器、分频器、移位寄存器、译码器; 常用的 74 系列芯片、接口芯片实验; 8 段数码块显示实验; 16x16 点阵式 LED 显示实验; 串行通信收发; I<sup>2</sup>C 总线等

提供 ABEL、VHDL 语言编写的实验范例

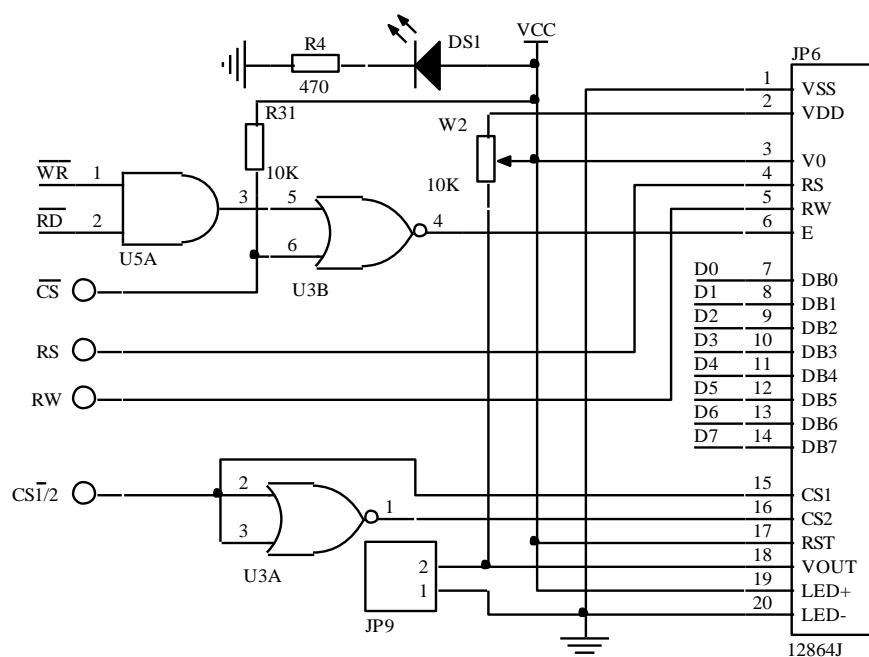
# 硬件结构

## 2. 1 电路外观



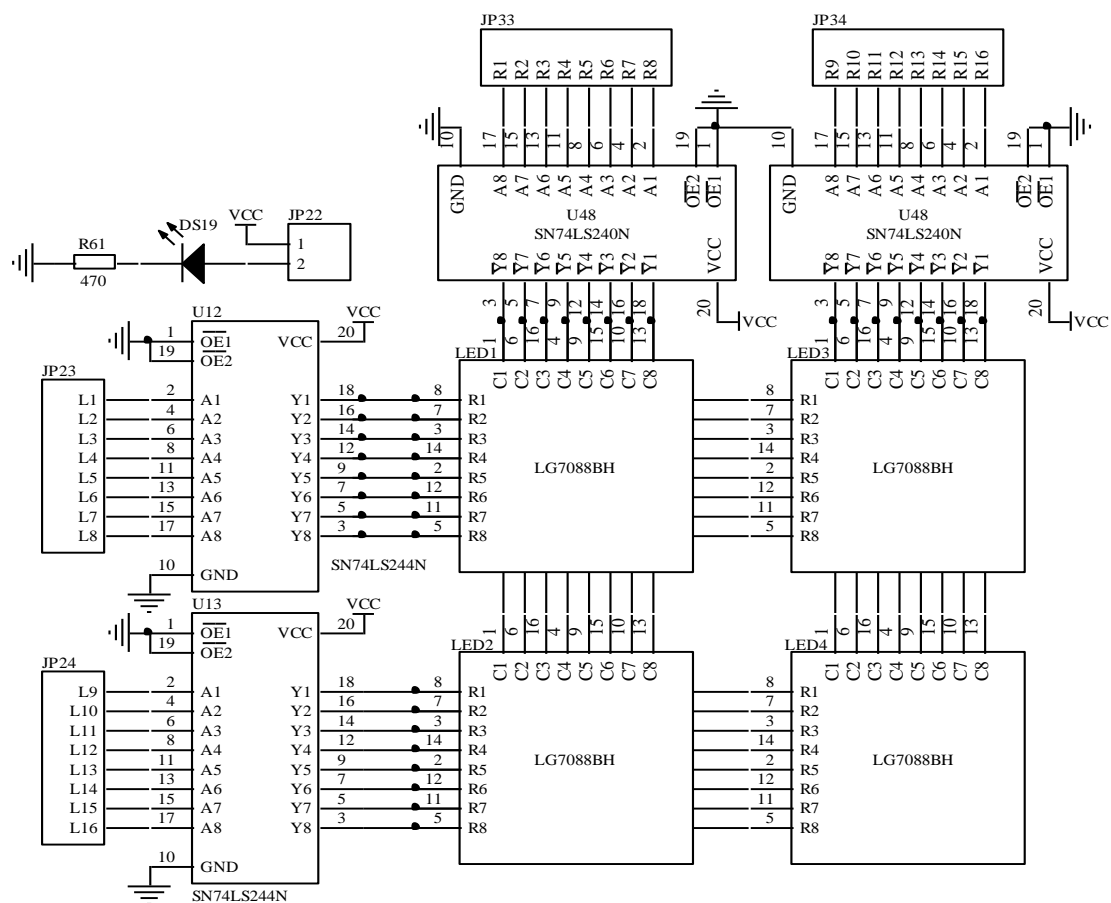
本章将逐一介绍实验仪的各个功能模块、相应的结构，读者在编写程序前，首先熟悉相应的硬件电路。

## 2. 2 A1 区：12864 液晶显示模块电路



CS: 片选信号，低有效。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。

## 2. 3 A2 区：16×16 LED 实验电路





JP23、JP24 组成 16 根行扫描线；JP33、JP34 组成 16 根列扫描线。

2. 4 A3 区：CPU 总线、片选区

- JP45：地址线 A0..A7；
- JP48、JP50：低位地址/数据总线；
- JP51、JP55：MCS51 的 P1 口；
- JP59：高位地址线 A8..A15；
- JP61、JP64：MCS51 的 P3 口，P3.7、P3.6 作读、写信号线用；
- JP66：相当于一个 MCS51 类 CPU 座，使用 40 芯扁线与用户板相连，可仿真 P0、P2 口作地址/数据使用的 MCS51 类 CPU。

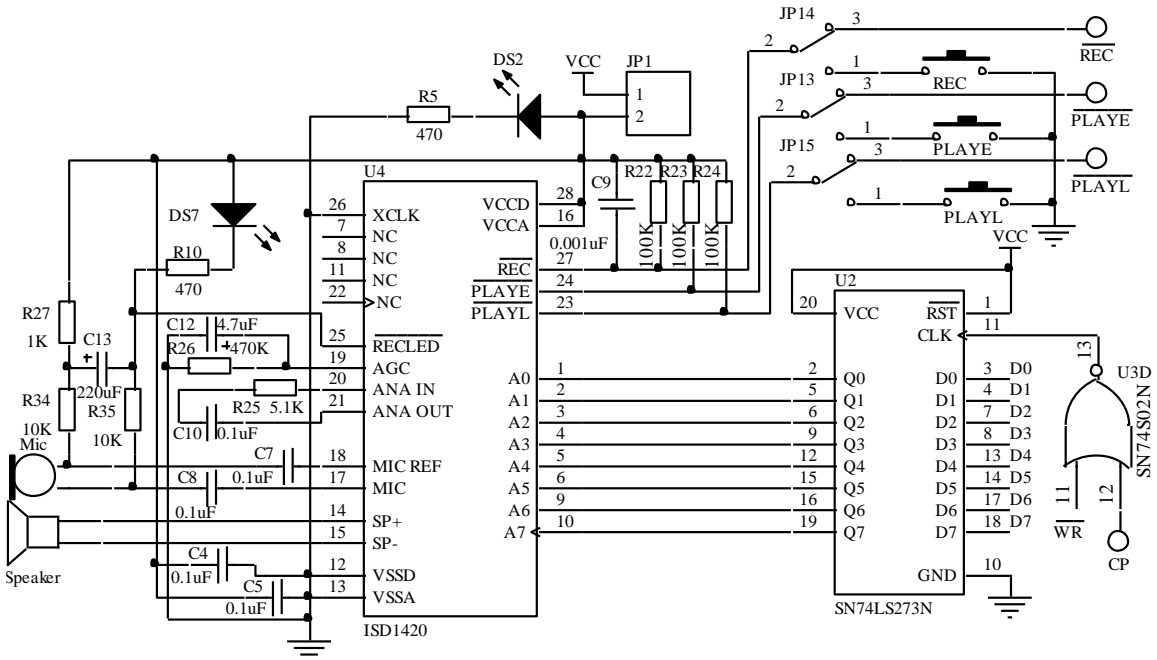
片选区

片选	地址范围	片选	地址范围
CS1	0F000H~0FFFFH	CS5	0B000H~0BFFFH
CS2	0E000H~0EFFFH	CS6	0A000H~0AFFFH
CS3	0D000H~0DFFFH	CS7	09000H~09FFFH
CS4	0C000H~0CFFFH	CS8	08000H~08FFFH

2. 5 A4 区：控制区

主控部分。

2. 6 B1 区：语音模块 ISD1420 电路

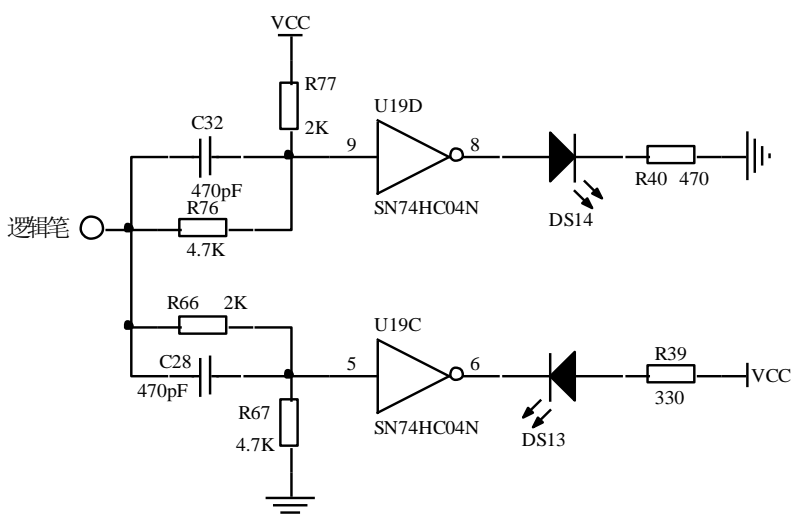


JP13、JP14、JP15：设置操作模式，MCU：CPU 控制方式；MANUAL：手动 (REC、PLAYL、PLAYE) 控制方式。

- 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

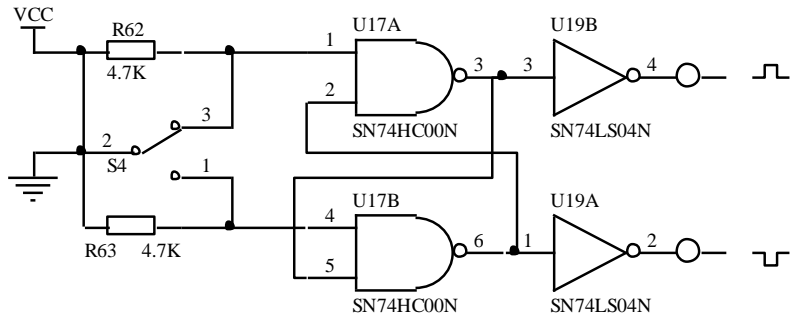
PLAYL: 边沿放音按键，下降沿有效，并在下一个上升沿停止放音

2. 7 B2 区：逻辑笔、单脉冲、频率发生器



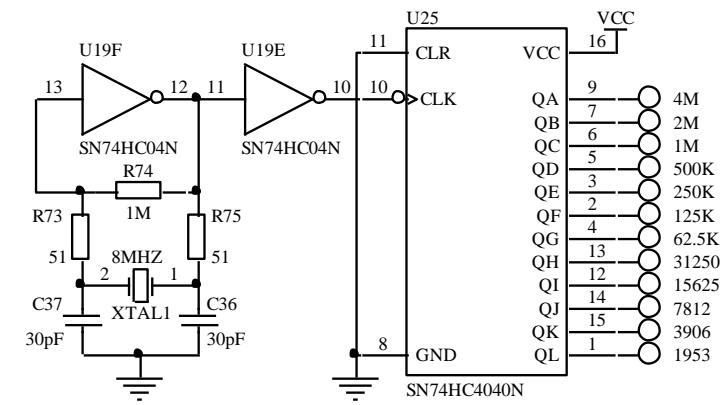
逻辑笔： 测试接口，输入  
测量信号  
绿灯 (DS13)： 高电平点亮  
红灯 (DS14)： 低电平点亮  
两灯同时亮： 频率信号

逻辑笔电路原理图



单脉冲电路原理图

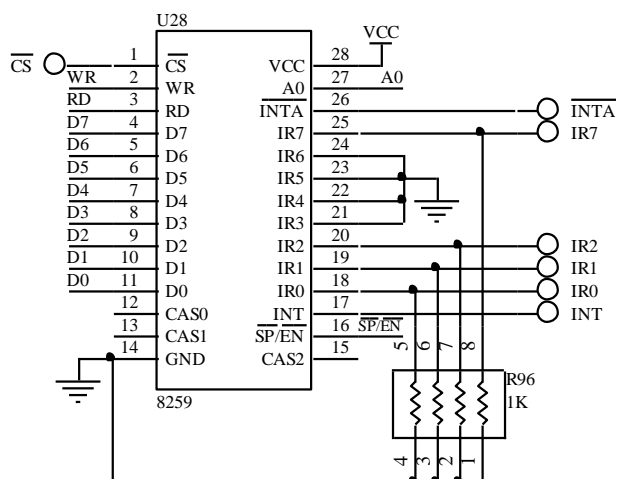
S4: 脉冲发生开关  
正脉冲： 上凸符号端口输出正脉冲  
负脉冲： 下凹符号端口输出负脉冲



频率发生器电路原理图

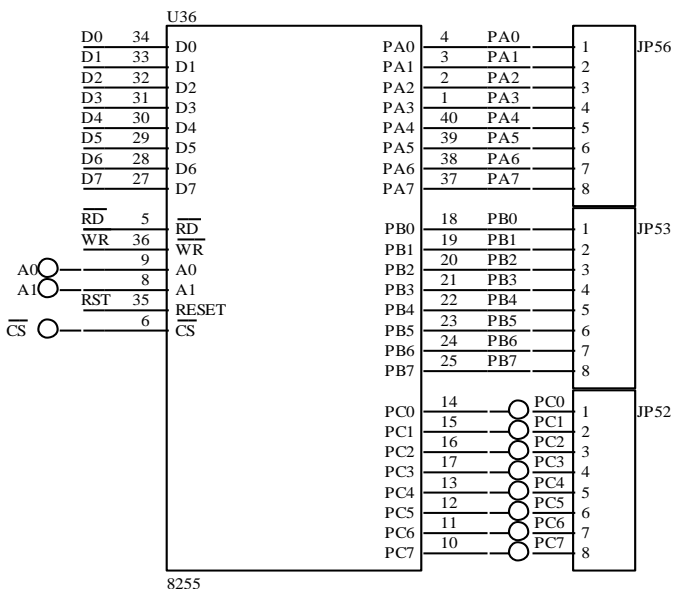
4M: 输出 4MHZ 频率信号  
其他端口输出的信号频率与端口下标识的数值一致

## 2. 8 B3 区: 8259 电路

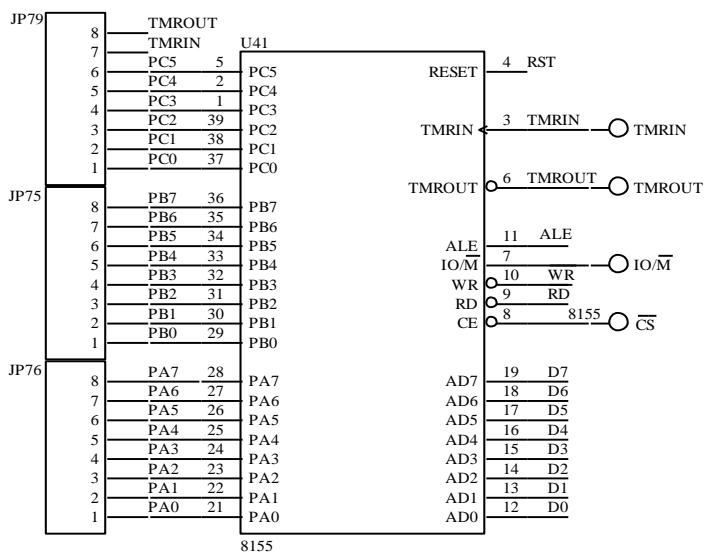


CS: 片选信号, 低电平有效;  
A0: 地址信号  
INTR0..INTR7: 中断输入  
INTA: 中断响应

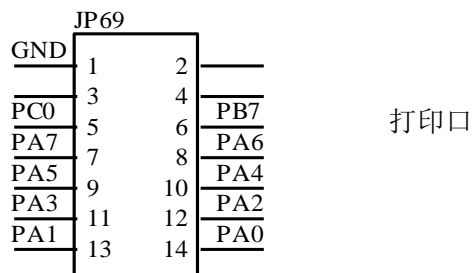
## 2. 9 B4 区: 8155、8255 电路



CS: 片选信号, 低电平有效;  
A0、A1: 地址信号。  
JP52: PC 口;  
JP53: PB 口;  
JP56: PA 口。



CS: 片选信号, 低电平有效。  
IO/M: 高电平, 选择 I/O 口;  
低电平, 选择数据 RAM。  
JP75: PB 口;  
JP76: PA 口;  
JP79: PC 口。



## 2. 10 B5、C6 区：扩展区

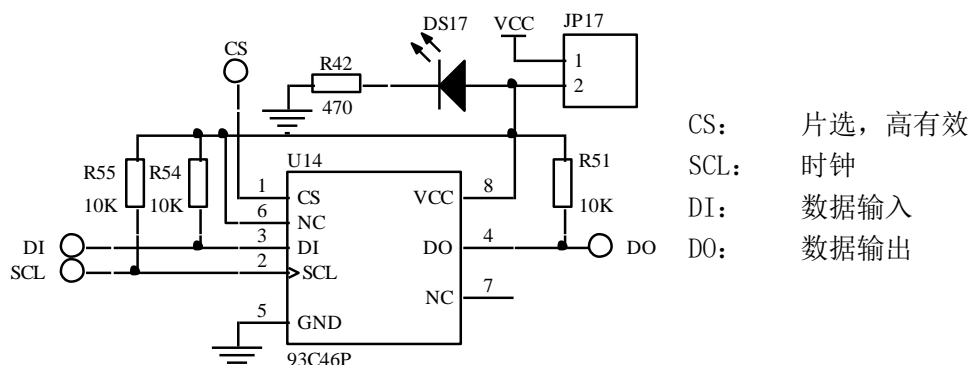
实验仪提供了二个扩展区，用来扩展 USB1.1、USB2.0、以太网、CAN 总线、非接触式 IC 卡、双通道虚拟示波器、CPLD、FPGA 等扩展模块，其它模块正在陆续推出中。

如果扩展模块较大，可以同时使用二个扩展区。

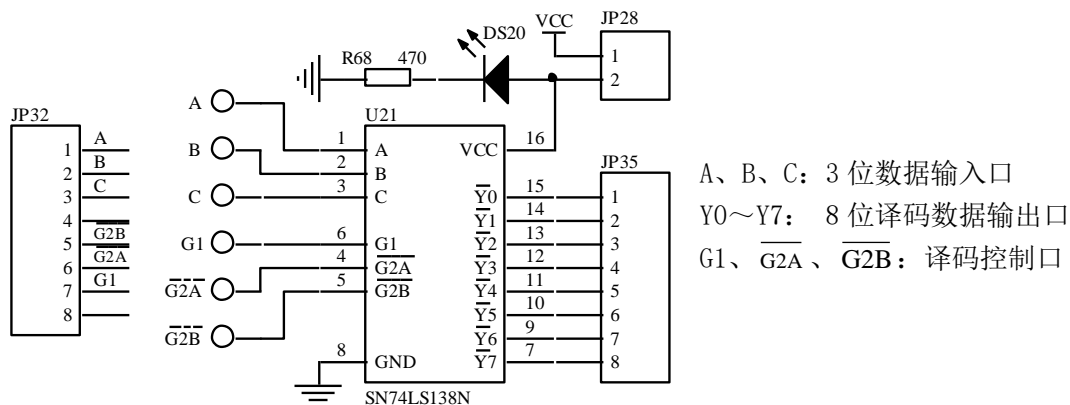
## 2. 11 C1 区：电源区

C1 区为用户提供了 5V(2A)、+12V(300mA)、-12V(300mA)等几种电源接口。

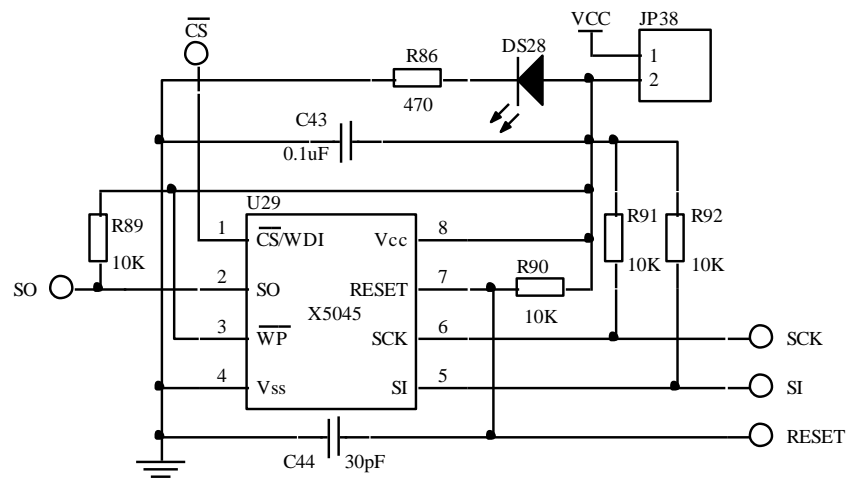
## 2. 12 C2 区：93C46



## 2. 13 C3 区：138 译码器

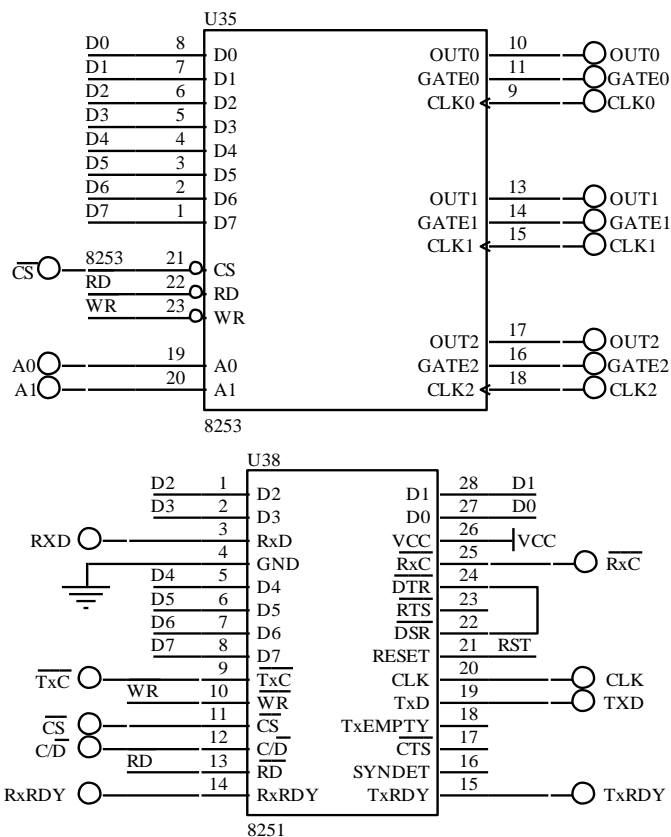


## 2. 14 C4 区: X5045



CS: 片选, 低有效  
SCK: 时钟  
SI: 数据输入  
SO: 数据输出  
RESET: 复位信号输出  
端, 高电平有效

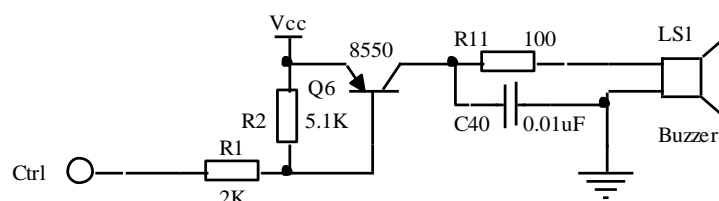
## 2. 15 C5 区: 8253、8251



CS: 片选信号, 低电平有效;  
A0、A1: 地址信号;

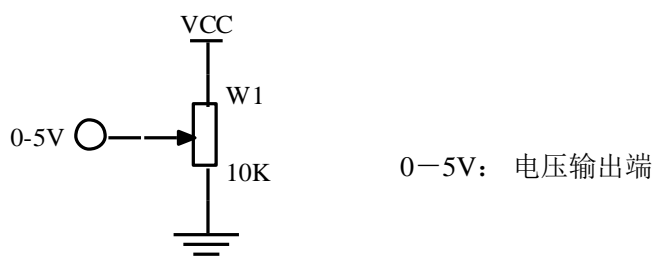
CS: 片选信号, 低电平有效;  
RxC、Tx C: 收发时钟;  
C/D: 命令/数据;  
RXD、TXD: 串行收发。  
CLK: 时钟。

## 2. 16 D1 区: 蜂鸣器



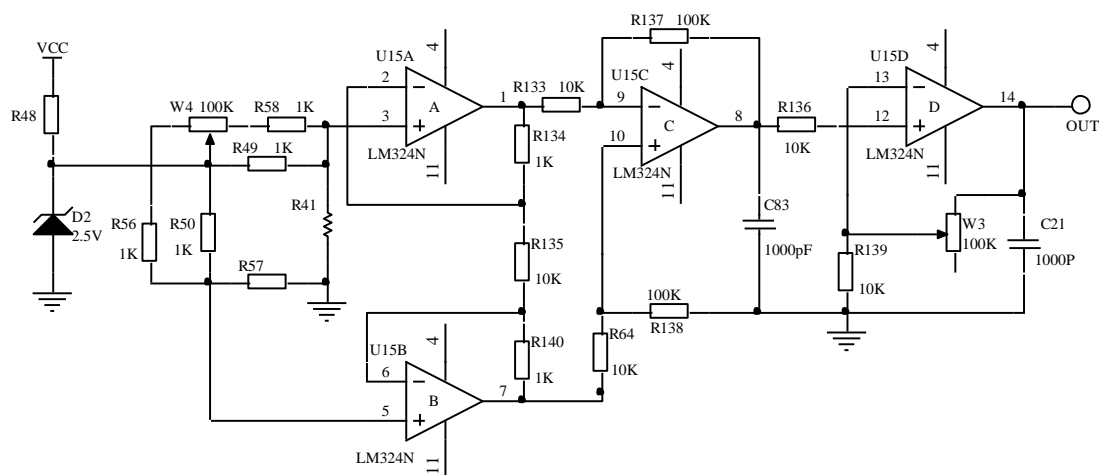
Ctrl: 控制接口, 0—蜂鸣

## 2. 17 D2 区: 0~5V 电压输出



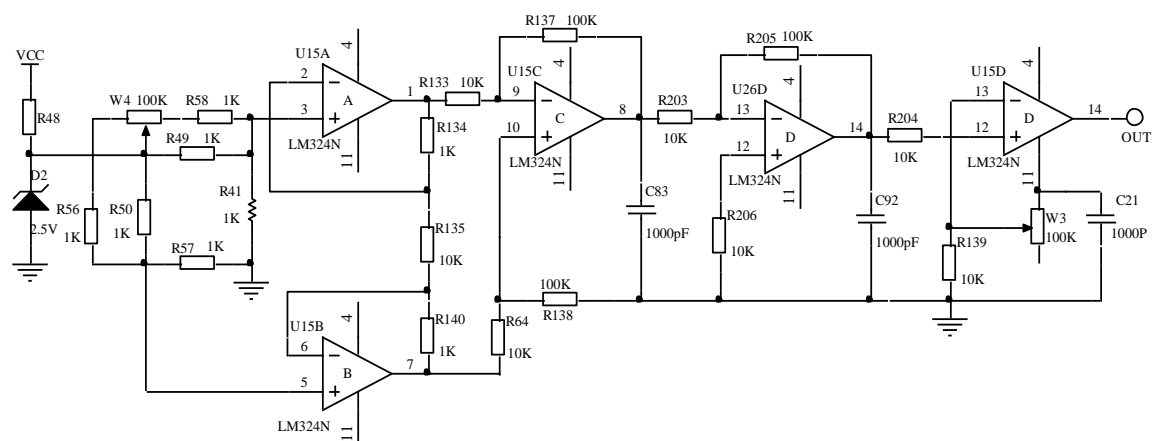
## 2. 18 D3 区: 光敏电阻、压力测量

### 光敏电路



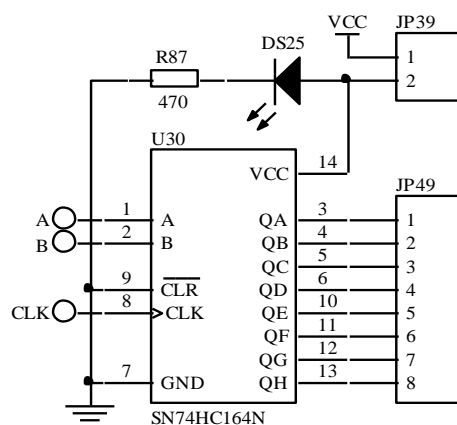
R41、R57 是光敏电阻；OUT：模拟电压信号输出端。

### 测压电路



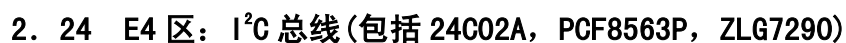
压力测量： R41： 电阻应变片，阻值 1K ；  
OUT： 压力模拟电压信号输出端；

## 2. 20 D5 区：串并转换



IN: 信号输入  
OUT: PWM 转换电压输出

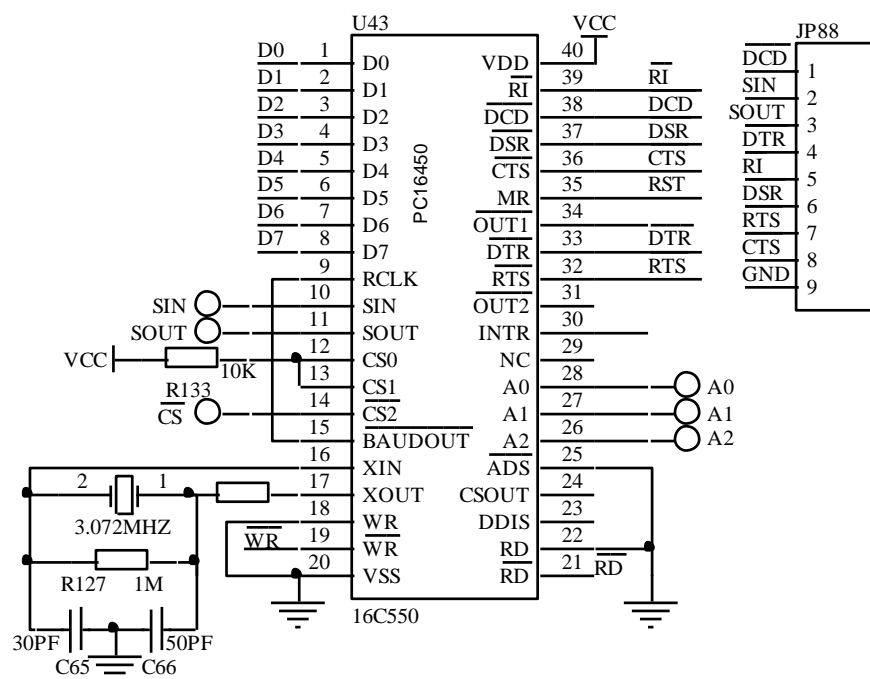
IN1: 信号输入  
OUT1: 信号输出





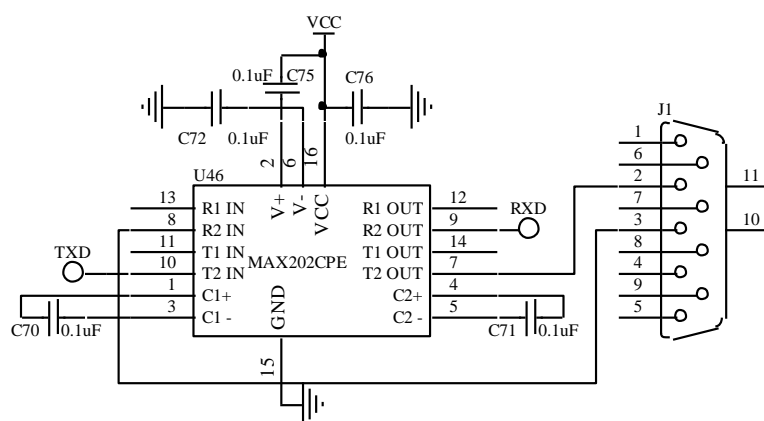


2. 26 E6 ☒: 8250

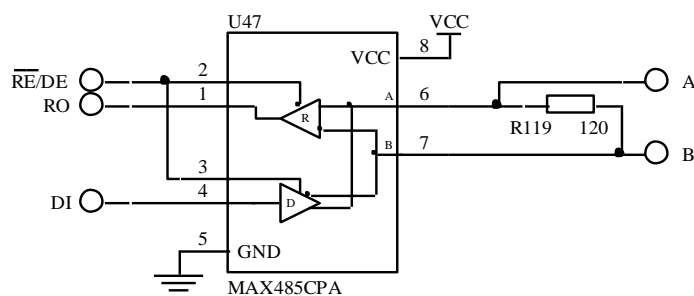


CS:	片选信号，低电平有效；	A0、A1、A2:	地址信号；
SIN :	串行输入	SOUT:	串行输出

## 2. 27 E7 区: RS232

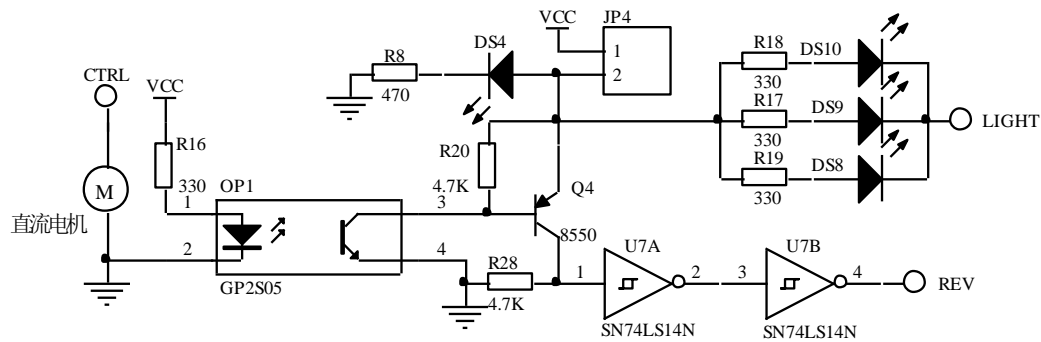


2. 28 E8 ☒: RS485

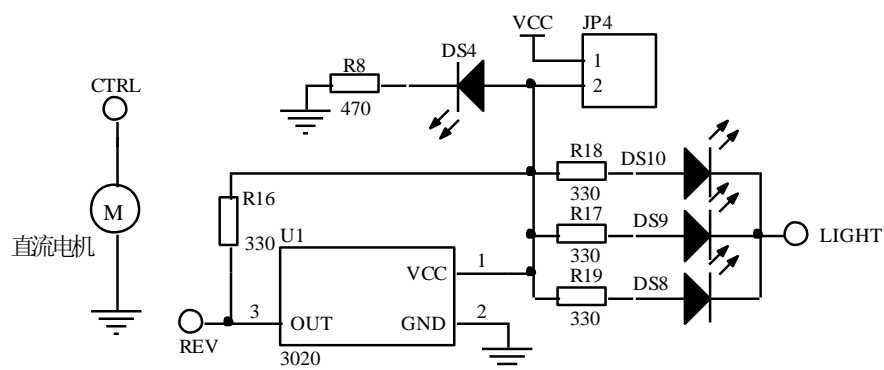


## 2. 29 F1 区：直流电机转速测量/控制

## 使用光电开关测速

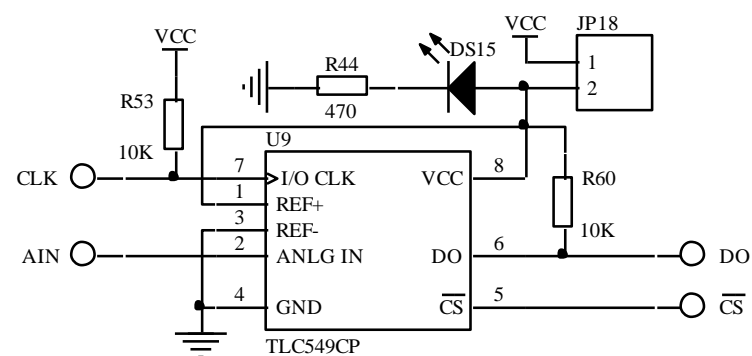


## 使用霍尔器件测速



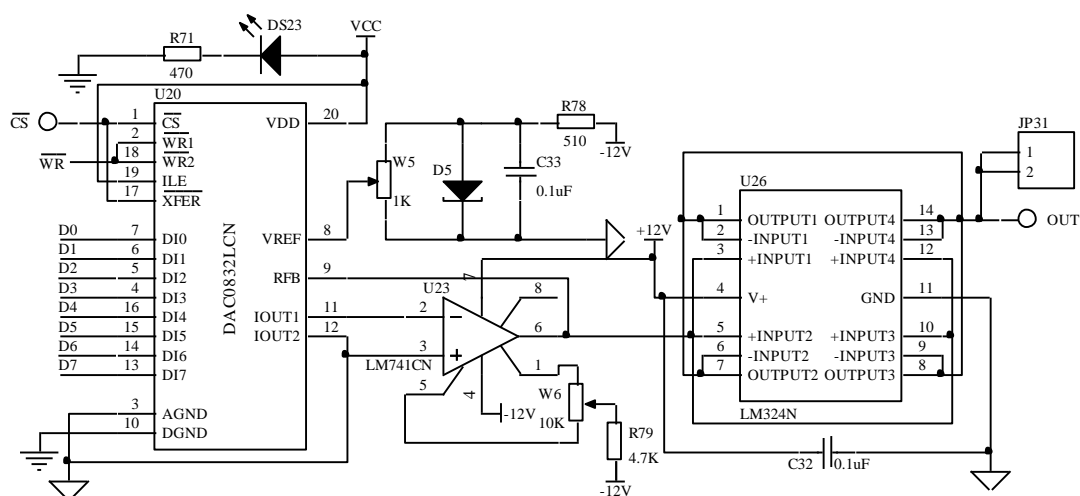
CTRL: 控制电压 (DAC0832 经功放电路提供) 输入; REV: 光电开关或霍尔器件脉冲输出 (用于转速测量); LIGHT: 低电平点亮发光管。

## 2. 30 F2 区：串行 AD



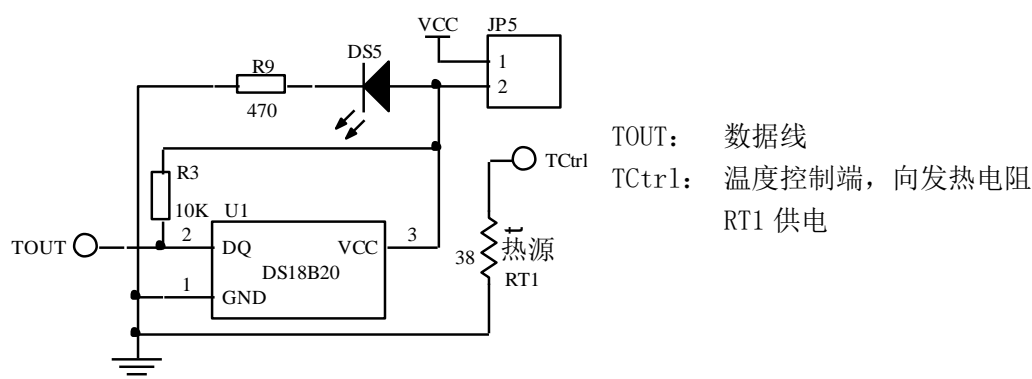
CS: 片选, 低电平有效;  
CLK: 时钟输入端;  
AIN: 模拟量输入端;  
DO: 数字量输出端。

## 2. 31 F3 区：DAC0832 数模转换

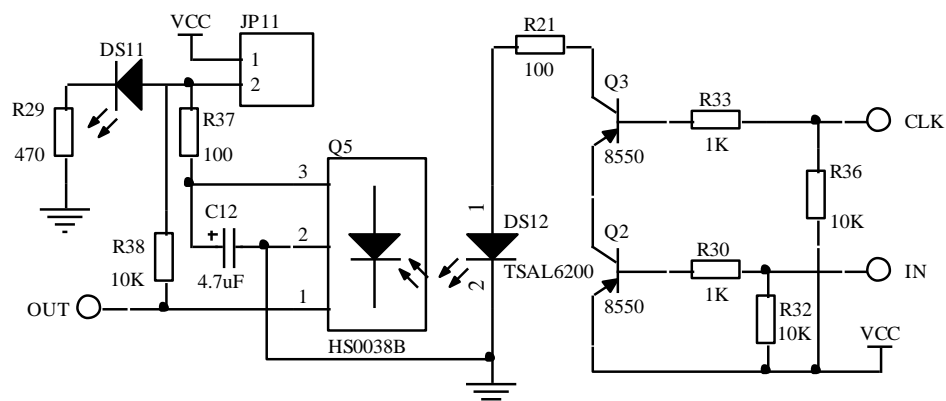


CS: 片选, 低有效; OUT: 转换电压输出; 电位器 W5: 调整基准电压。

## 2. 32 G1 区：温度测量/控制

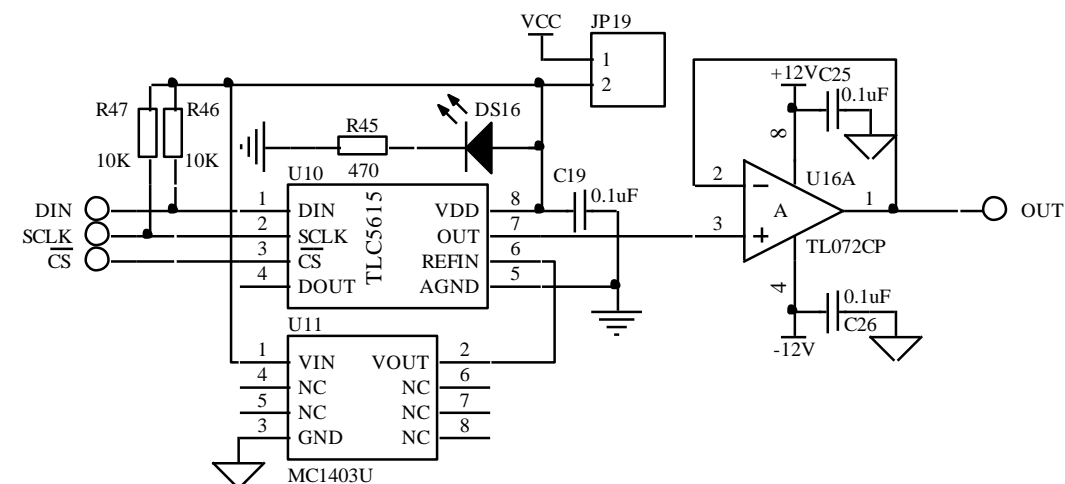


## 2. 33 G2 区：红外通讯



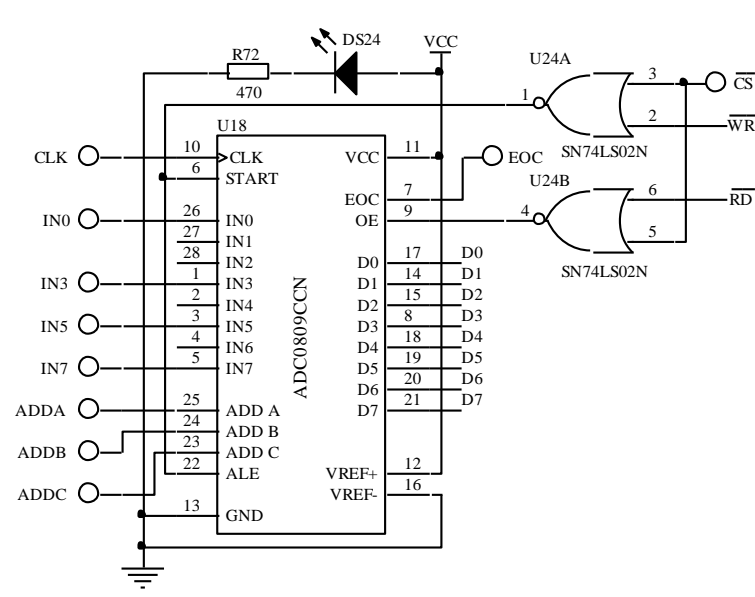
IN:      串行数据输入                          OUT:     串行数据输出  
CLK:     载波输入，可接 31250(B2 区)频率输出

2. 34 G3 区：串行 DA



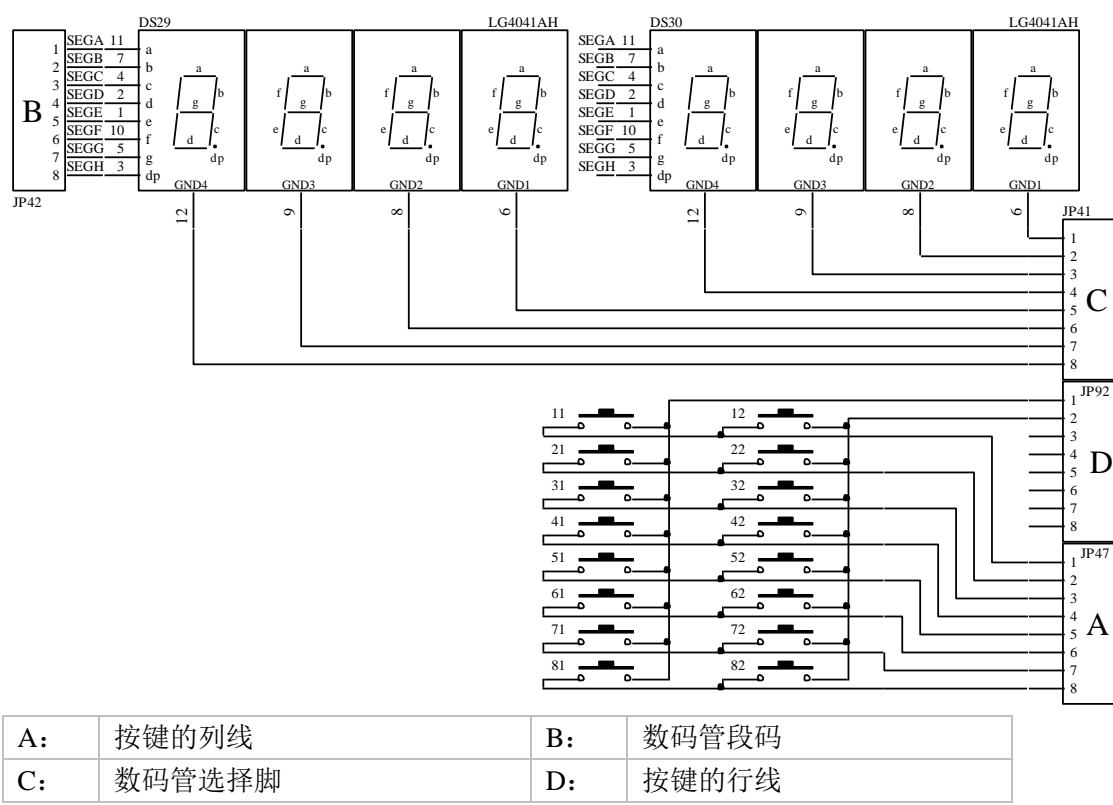
CS:	片选，低有效	DIN:	数字量输入端
SCLK:	时钟	OUT:	模拟量输出端

2. 35 G4 区：ADC0809 模数转换

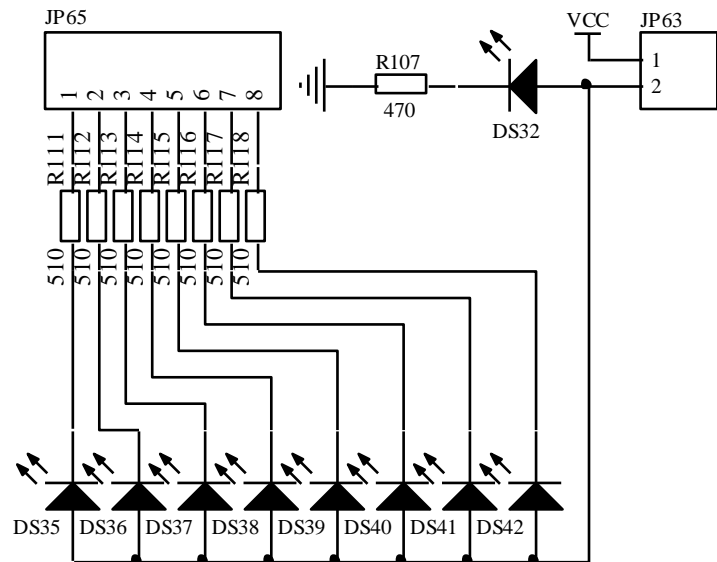


CS:	片选，低有效；
CLK:	输入时钟 (10k—1280kHz)；
ADDA, ADDB, ADDC:	通道地址输入口；
EOC:	转换结束标志，高有效。
IN0、IN3、IN5、IN7:	模拟量输入

2. 36 G5 区：键盘&LED

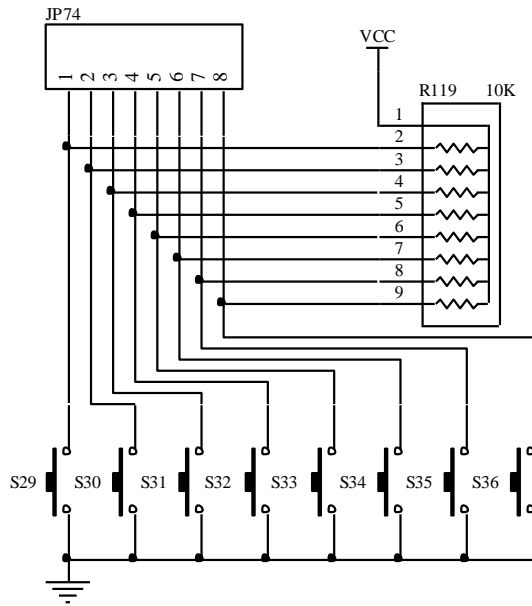


2. 37 G6 区：发光管、按键、开关



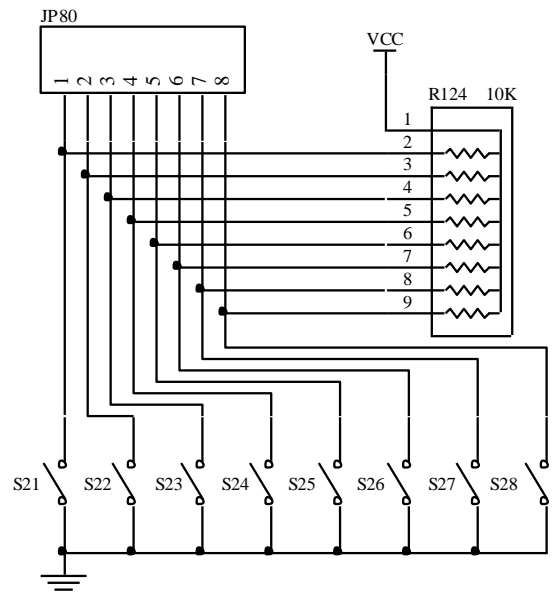
发光管电路原理图

JP65: 发光管控制接口, 0—灯亮, 1—灯灭



按键电路原理图

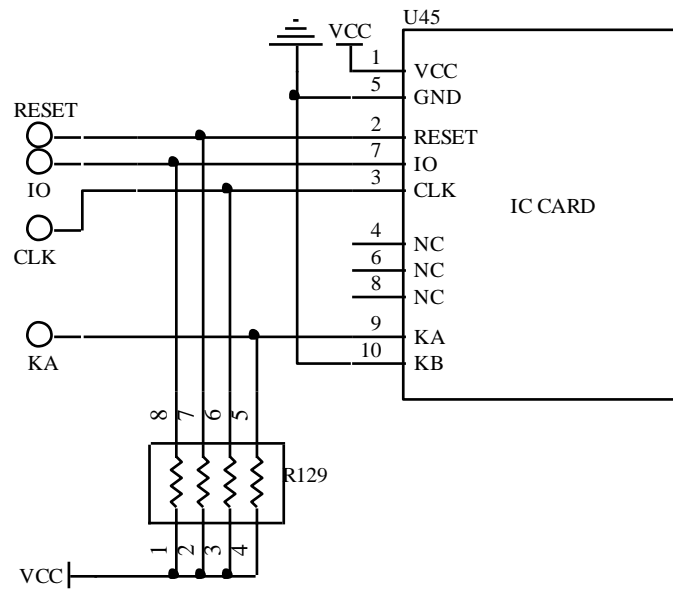
JP74: 按键控制接口; 按下—0 信号, 松开—1 信号



开关电路原理图

JP80: 开关控制接口; 闭合—0 信号, 断开—1 信号

## 2. 38 G7 区: 接触式 IC 卡





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

#### 一. 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\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。

#### 二. 已安装过低版本星研集成环境软件的用户安装步骤：

1. 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。
2. 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、实验仪与微机的第一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：





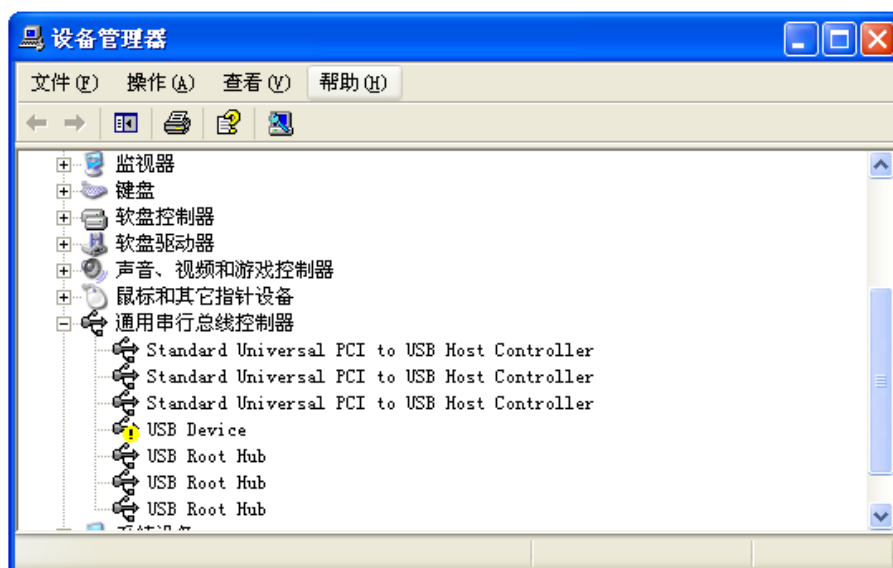
实际的界面可能有些差别，请等待该过程的结束。驱动程序的安装过程中，请勿执行其它应用程序。

## 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：在仿真器电源打开的情况下，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 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

运行 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；

3. 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 单元的内容传送给实验仪 B4 区的 61C256 的 2000H~20FF 中；B4 区的 61C256 在 I/O 设备区,使用  $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$  读写;再将它传送回数据段的 6000H~60FFH 中。3.2.2 使用 Turbo C, 重新编写第一个实验。

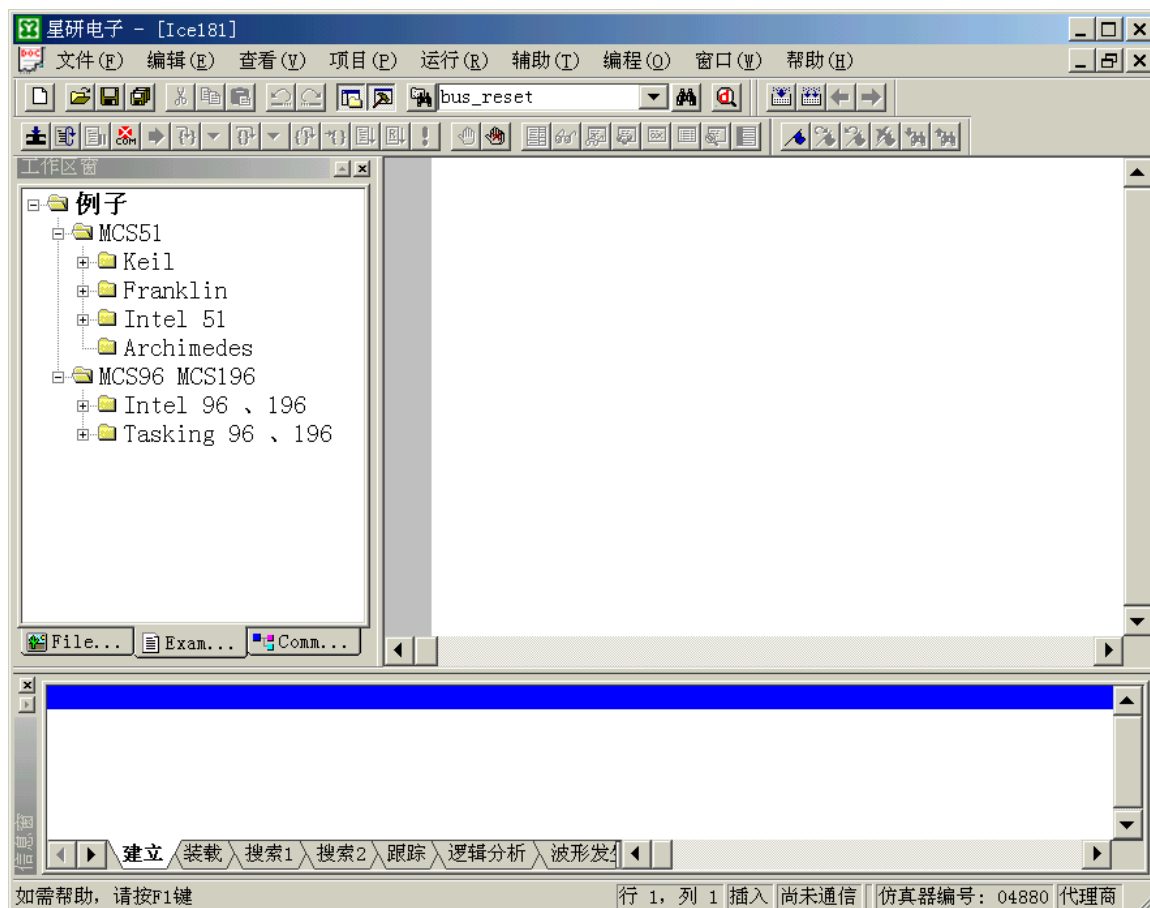
### 3. 2. 1 数据传送程序 (ASM)

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。

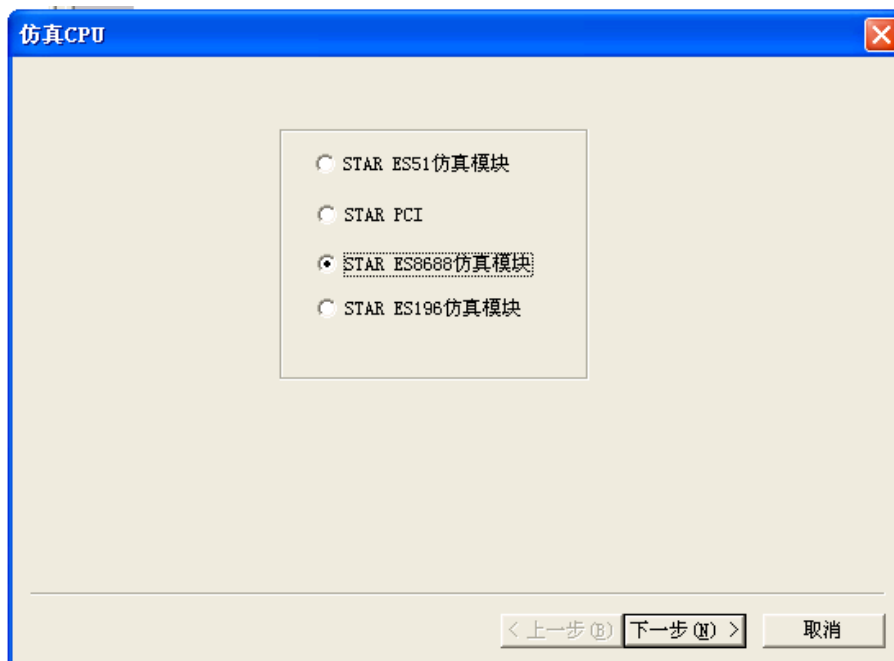
本实例是将数据段中 3000H~30FFH 单元的内容传送给实验仪 B4 区的 61C256 的 2000H~20FFH 中; B4 区的 61C256 在 I/O 设备区, 使用  $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$  读写; 再将它传送回数据段的 6000H~60FFH 中, 程序是用汇编语言来编写。下面介绍相应的操作步骤:

首先运行星研集成软件。启动画面如图:



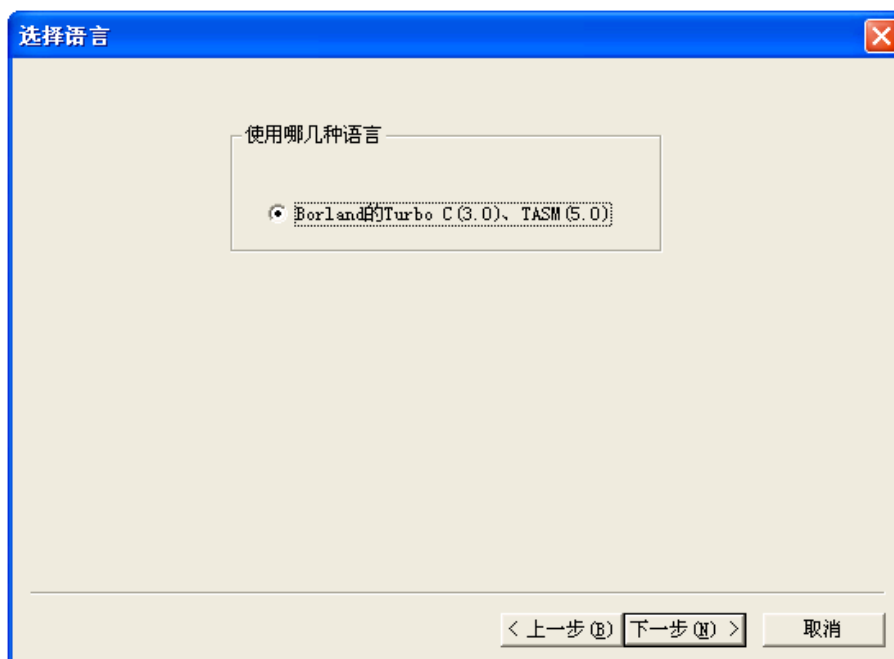
#### 1、设置缺省项目

执行 [主菜单 » 辅助 » 缺省项目], 出现一个对话框:

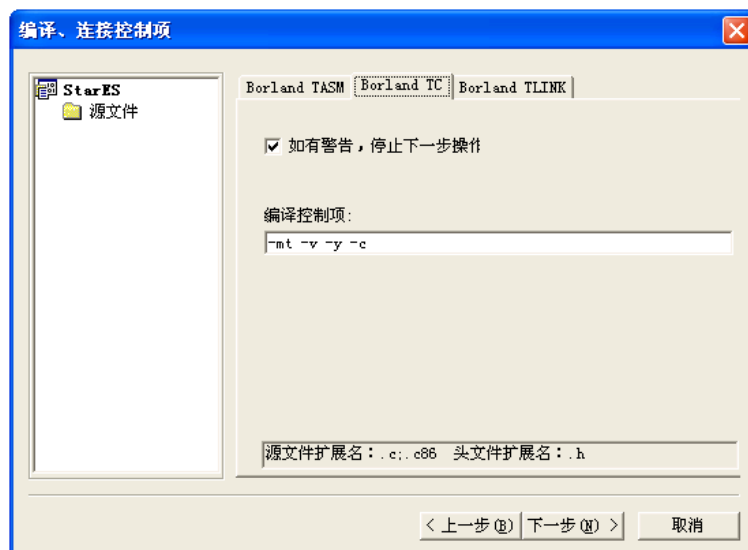


选择“STAR ES8688 仿真模块”。

点击进入下一步：“选择语言”

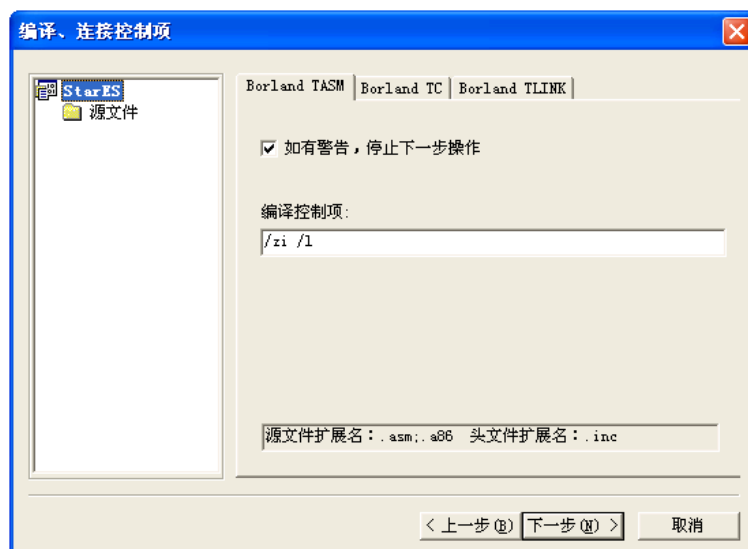


您可以根据自己的需要以及程序的类型作相应的选择，本实例选择 Borland 公司的 Turbo C (3.0)、TASM (5.0) (请确定在选择语言之前已经安装好相应的编译软件)。然后再点击进入下一步：“编译、连接控制项”



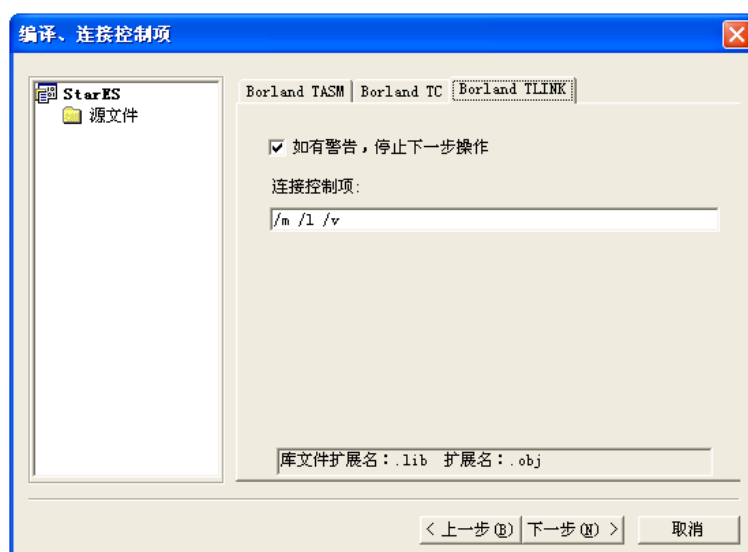
实验仪提供 64K memory 空间，memory model 请选择 tiny，缩写为 mt;如果需要源程序级别调试，必须使用 -v -y 控制项，为了支持多文件编译、连接，必须使用 -c 控制项。

一般不必改变 Turbo C 的编译控制项。



如果需要源程序级别调试，必须使用 /zi /l 控制项。

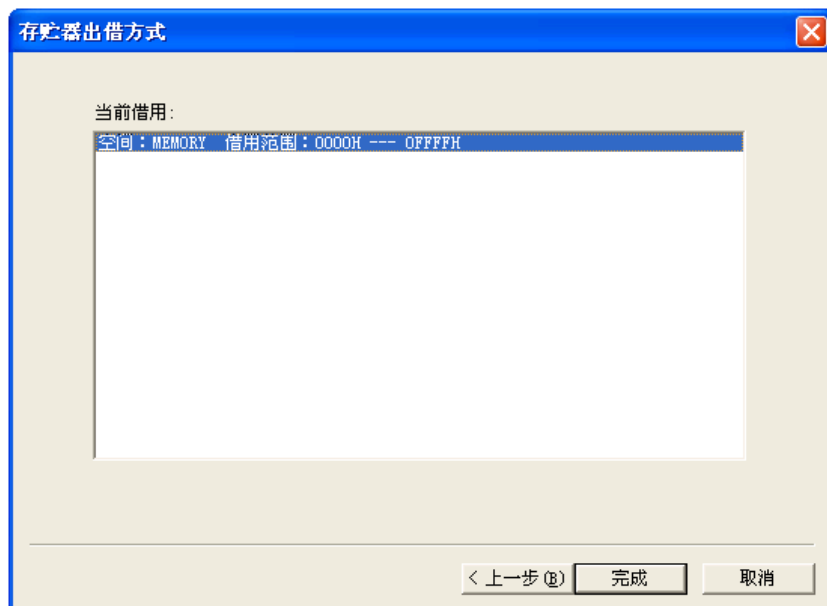
一般不必改变 Tasm 的编译控制项。



如果需要源程序级别调试，必须使用 /m /l /v 控制项。


一般不必改变 TLINK 的连接控制项。

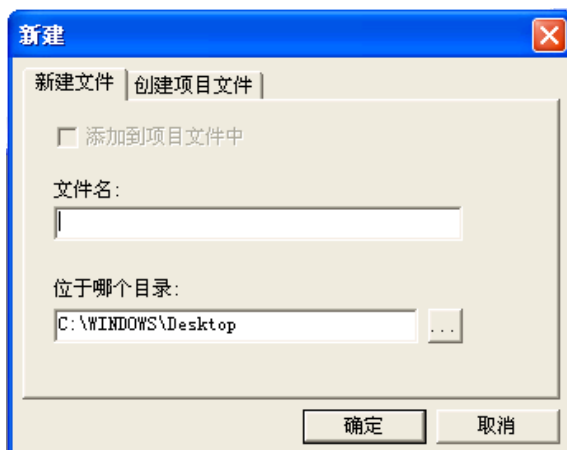
然后再点击进入下一步：“存储器出借方式”



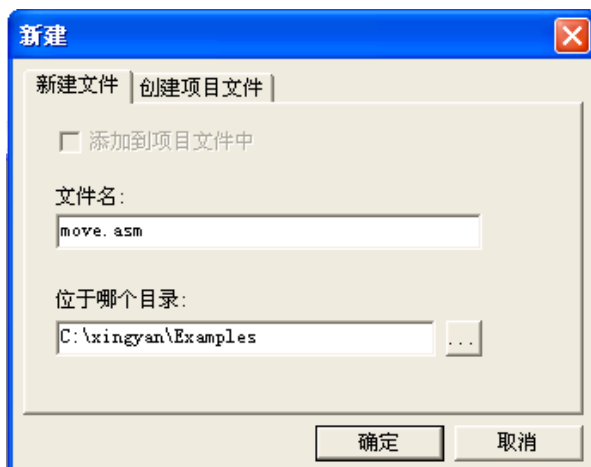
仿真模块 EMU598 提供 64K 仿真 RAM，作程序段（CS）、数据段（DS）、附加段（ES）、堆栈段（SS）使用。

## 2、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：

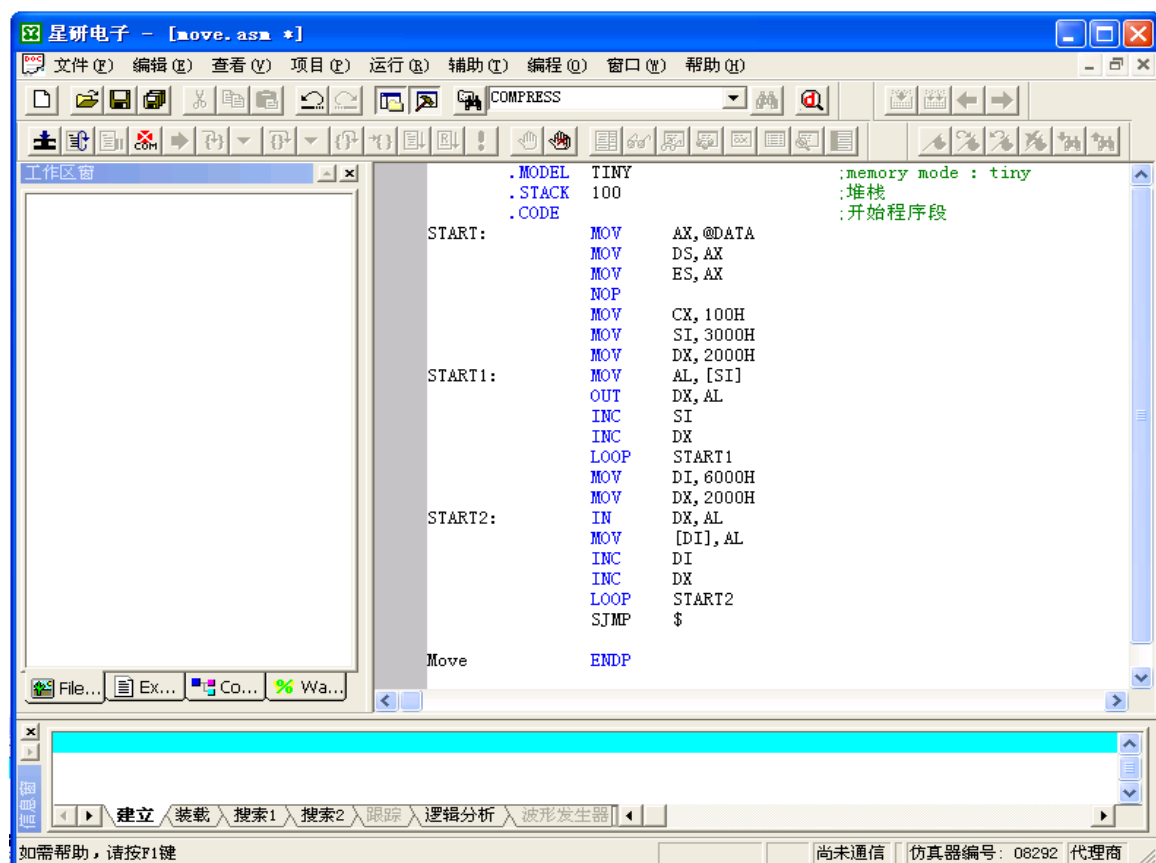


输入源程序，本实例的源程序如下：

```
.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      DX, 2000H
START1:  MOV      AL, [SI]
        OUT      DX, AL
        INC      SI
        INC      DX
        LOOP     START1
        MOV      DI, 6000H
        MOV      DX, 2000H
        MOV      CX, 100H
START2:  IN       DX, AL
        MOV      [DI], AL
        INC      DI
        INC      DX
        LOOP     START2
        SJMP     $



Move     ENDP
        END      START
```

输入源程序，如下图：

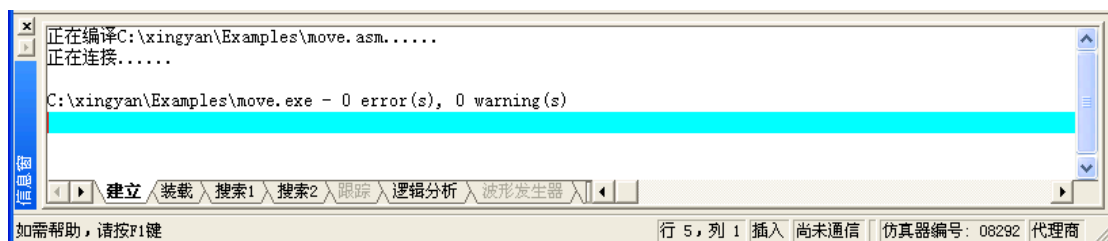


这样一个源文件就建立好了。

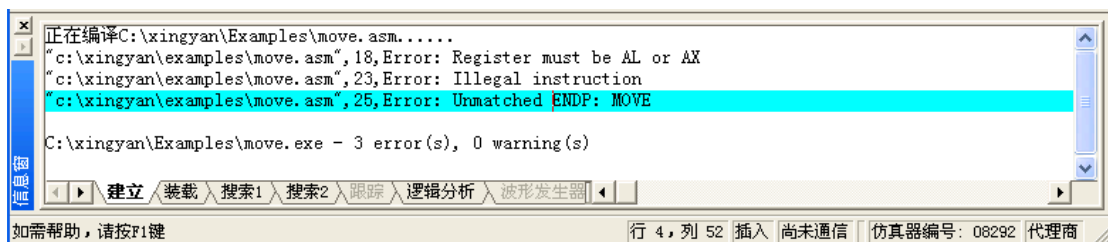
### 3. 编译、连接文件

首先选择一个源文件，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）点击图标或来“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。编译没有错误的信息如下：



若有错误则出现如下信息框：





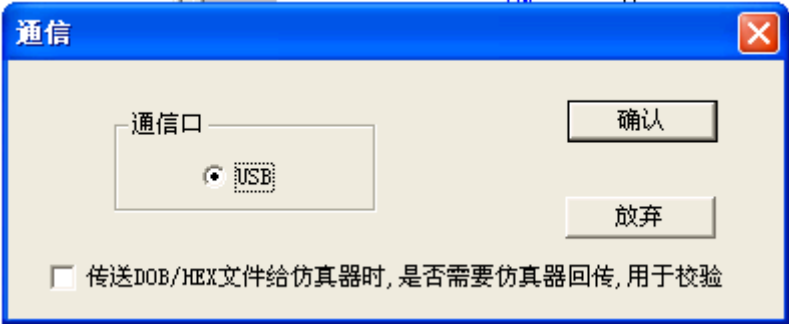
有错误、警告信息，用鼠标左键双击错误、警告信息或将光标移到错误、警告信息上，回车，系统自动打开对应的出错文件，并定位于出错行上。



这时用户可以作相应的修改，直到编译、连接文件通过。

#### 4. 调试

在进入调试状态以前，请正确设置通信口：执行[ 主菜单 » 辅助 » 通信 ]，对话框如下：



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，你还必须确定仿真器、实验仪与微机的正确连接，如果使用仿真器，仿真头正确地连接在仿真器上。电源接通，开关打开。

在软件中选择对应的仿真器、实验仪型号，具体设置如下：执行[ 主菜单 » 辅助 » 仿真器、实验仪 ]，对话框如下：

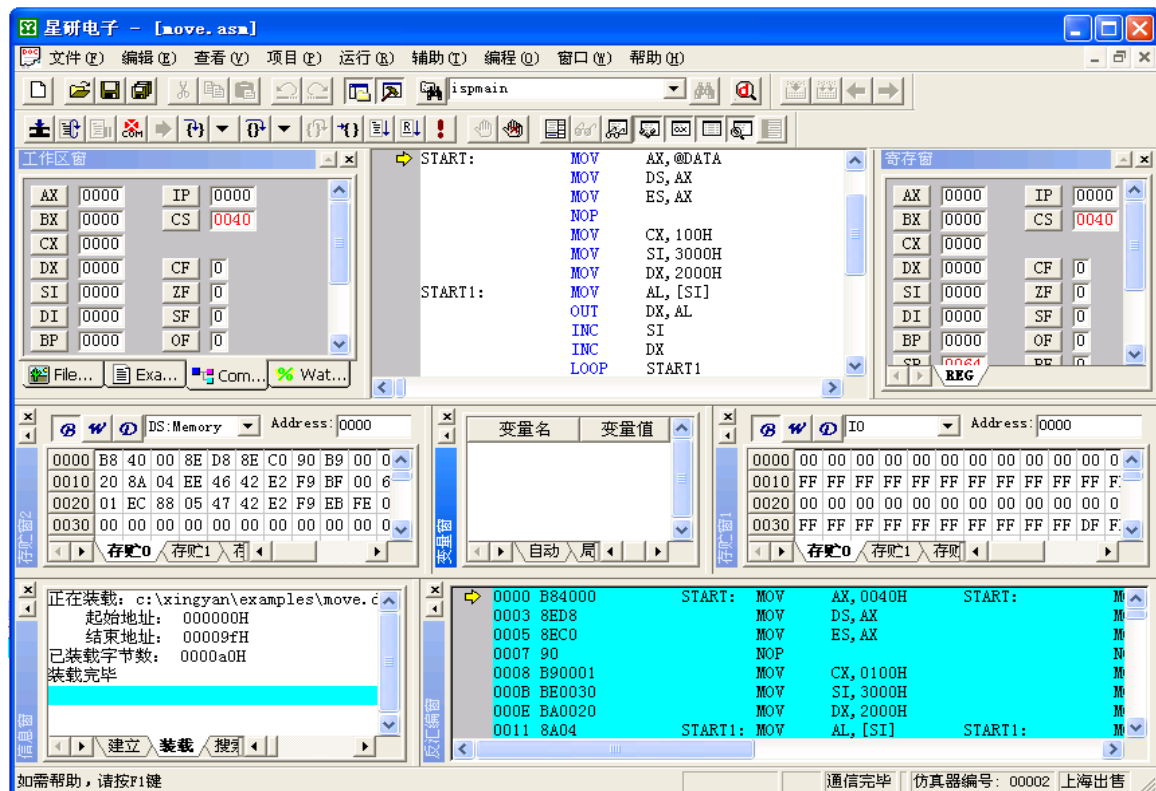


根据你所使用的机型作相应选择。

如果编译、连接正确后，可以开始调试程序。进入调试状态方法有：

- 执行[ 主菜单 » 运行 » 进入调试状态]
- 点击工具条的
- 执行[ 主菜单 » 运行 » 装载 DOB、HEX、BIN 文件]

进入后的窗口如下：




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一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）、IO 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，显示起始地址，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，变化一个或一组窗口的大小。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、存储器窗 1、2、3、观察窗、变量窗、反汇编窗。您可以通过[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。



首先在“种类”中选择一个窗口，然后选择“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。

对于高级语言，在您的程序前有一段库文件提供的初始化代码，（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行后停下。

您可以使用以下命令调试您的程序：

-  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”，用鼠标点击或按任意键后，停止运行。
-  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，停止运行。
-  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，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，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。



停止运行

终止微机与仿真器之间通信（功能键 ESC）。

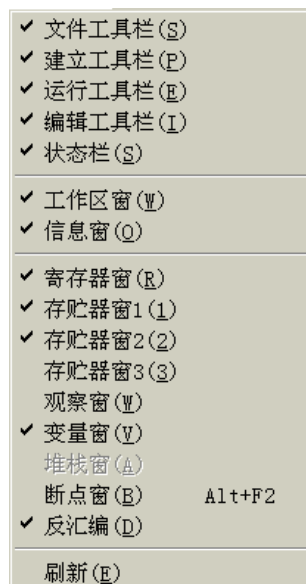
注意：欲终止微机与仿真器之间通信，功能键 **ESC** 是一个很方便的键，它的效果比点击相应的图标的效果要好。建议用户多用 **ESC** 键。在系统运行“连续单步”或者“连续单步进入”时 **ESC** 键被禁止，这时用户可以按键盘的其他任意键停止其运行。

## 5. 调试的方法及技巧

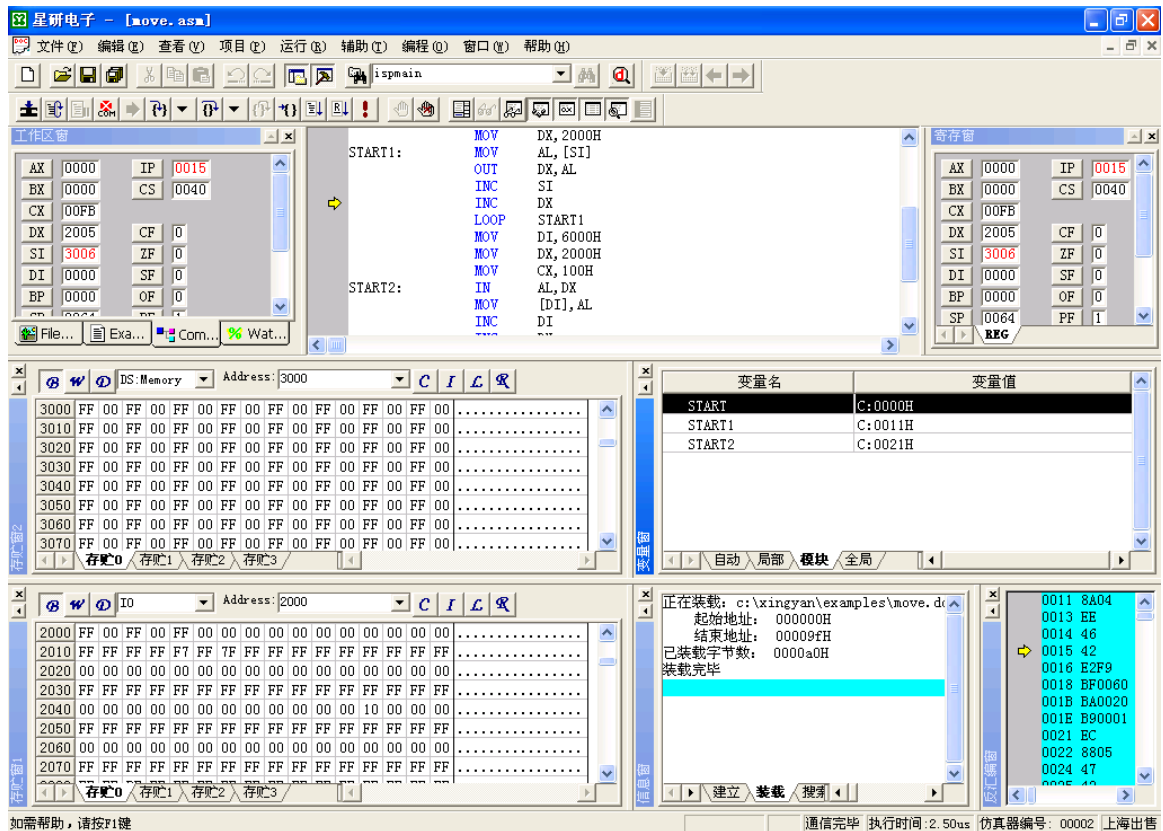
一般来说，用户的程序或多或少的会有一些逻辑错误，我们的仿真器、实验仪和星研集成软件可以帮助用户很快的定位，很快的查出相应的错误。

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三个数据块：数据段 3000H~30FFH，数据段 6000H~60FFH，I/O 区 2000H~20FFH，可以打开二到三个存储器窗口，具体操作是：[主菜单>>查看]

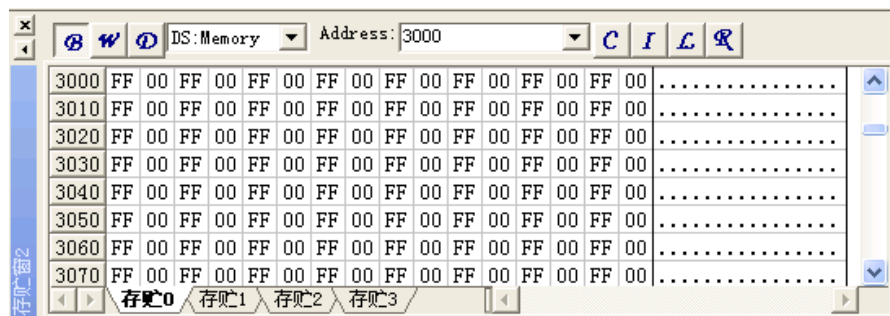


然后根据你的需要打开不同的窗口。调整后的调试界面为：

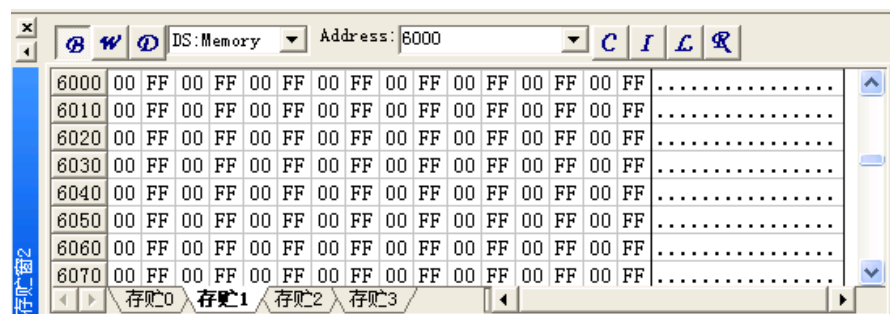


由于我们本次操作主要是观察存储器窗口，所以我们拉大了这两个存储器窗口的大小。每个窗口设置了4个分页项：**存储0** **存储1** **存储2** **存储3**，我们可以在不同的分页项设置不同的观察数据空间以及地址范围。在**DS: Memory** 中可以选择 CS: Memory, DS: Memory, I/O，根据需要可以做不同的选择。在**Address: 0000** 中可以直接输入地址，然后按回车，就可以直接转到我们输入的地址的窗口上面观察数据。由于我们在此程序中的写入数据的 RAM 空间分别为 DS: 3000H~30FFH、DS: 6000H~60FFH、I/O 区 2000H~20FFH，故我们建立的分页项如下：

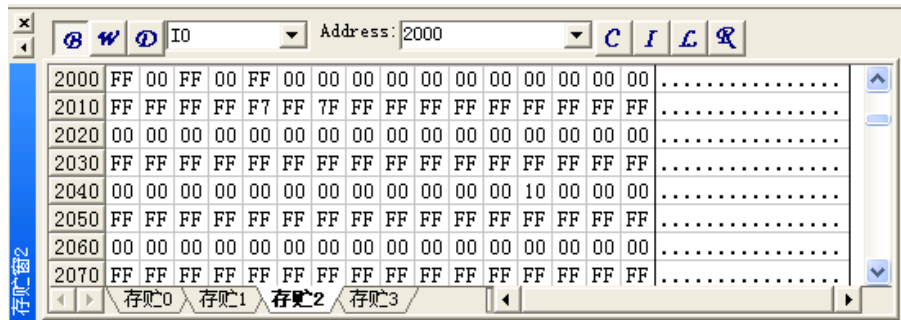
存储 0 分页项：



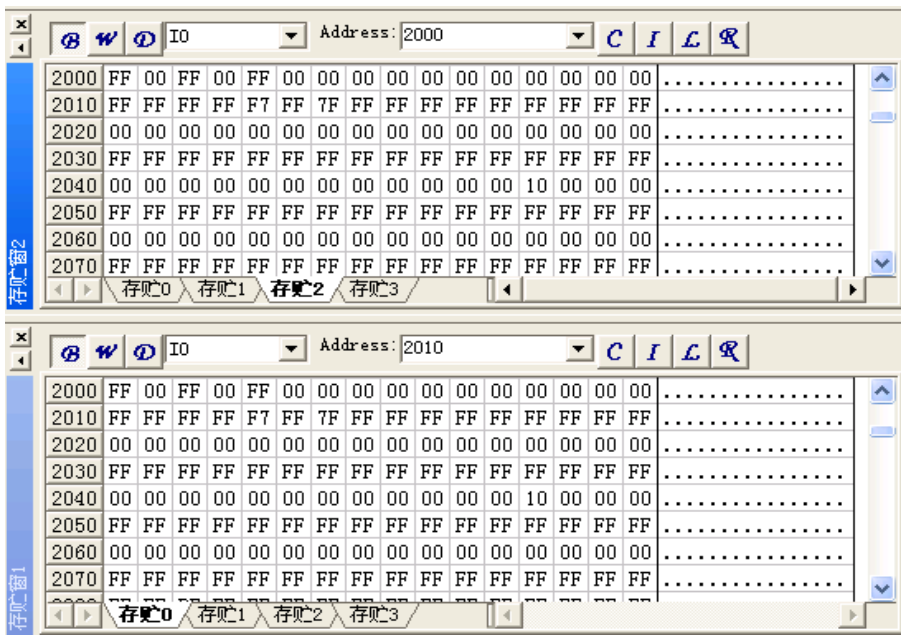
存储 1 分页项：



存贮 2 分页项:

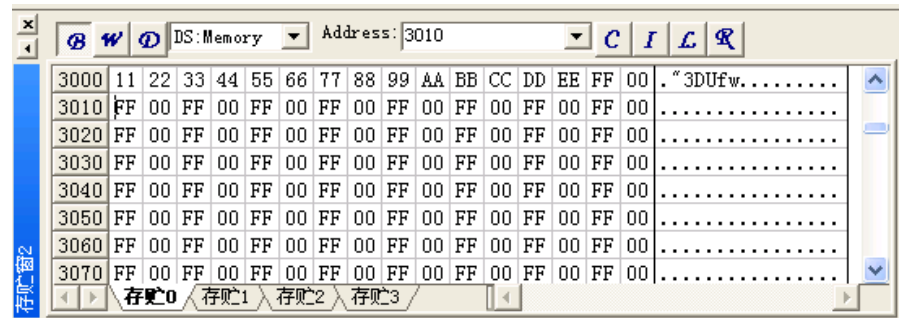


我们这样设置界面的目的就是当用户要观察不同地址段的数据时，只要切换一下分页项就行了。由于本次程序需要同时观察 DS: 3000~30FFH、I/O: 2000~20FFH 和 I/O: 2000H~20FFH、DS: 6000H~60FFH，所以打开二个存贮器窗。如图：



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：

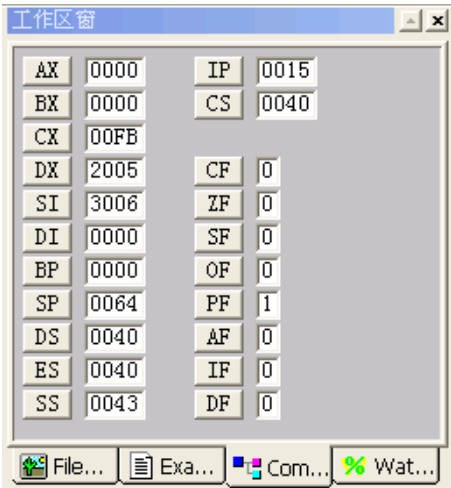


一般刚刚写好的程序，在进入调试状态后，执行“单步”或者“单步进入”，我们推荐您能记住这些操作的相对应的功能键，这样您就在调试程序的过程中很方便。

在刚才的调试程序中我们多次执行“单步（F8）”命令，在工作区窗口的 CommonRegister

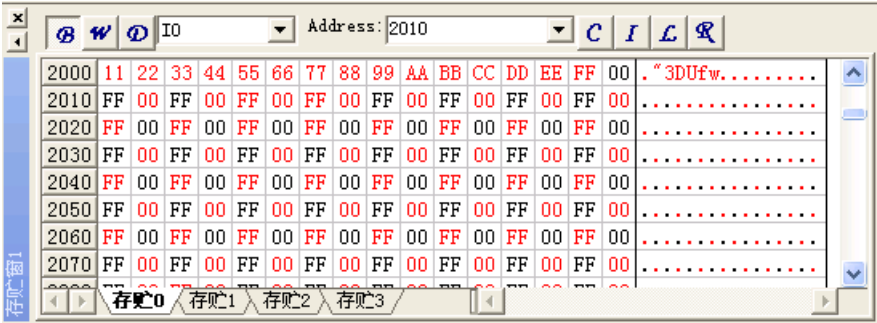


视中查看通用的寄存器:



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 AX、CX、DX、SI 的数值的变化，每一次循环，CX 减一，DX、SI 加一，AX 寄存器的低字节 AL 暂存从 DS: [SI]取出的数值：11、22、33、44、55、66……。

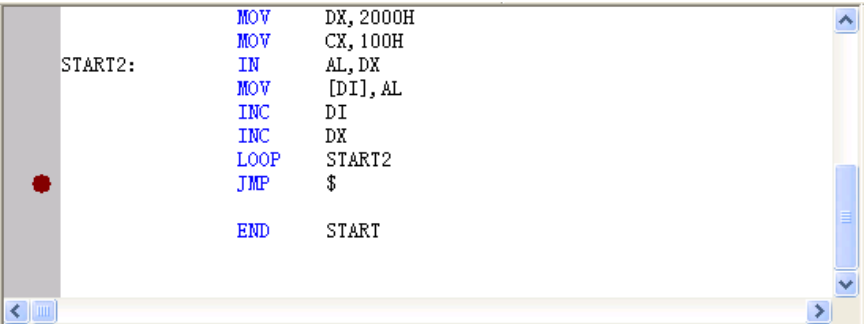
我们可以看到存储器窗口中的相对应的 RAM 的数据的变化。比如



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 MOV DI, 6000H 行上，点击图标 (功能键 F4)，全速运行到光标行，检查 IO: 2000H~20FFH 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。

切换分页项，存储器窗显示 DS: 6000H 开始的单元内容，将光标移到 JMP \$ 行的左边，鼠标变为 ，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 (功能键为 F2)，设置断点，重复操作，清除断点。




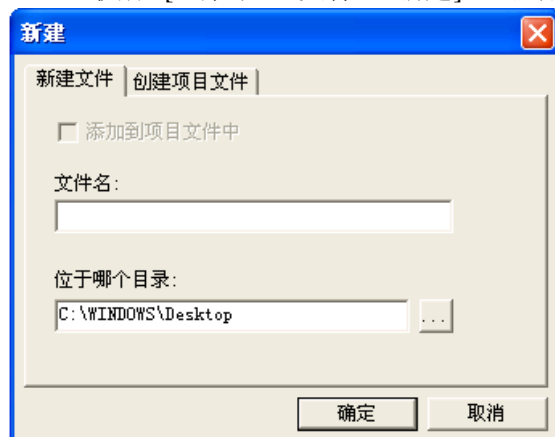
点击图标 (功能键 F9)，CPU 全速断点运行到光标处停下，检查 DS: 6000H~60FFH 内容，与 I/O: 2000H~20FFH 内容是否一样，相同表示程序没问题；如果不相同，将光标移到 MOV DI, 6000H 行上 (具体操作是：用鼠标点击该行，然后再点击图标 )，使用单步进入命令 F7 或连续单步进入命令 Ctrl + F7，检查结果，判断程序出错原因。

### 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

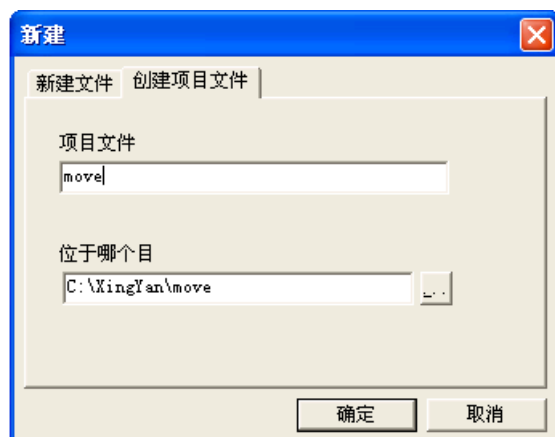
执行 [主菜单 » 文件 » 新建]，（或者点击图标）打开窗口如下：



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：



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：



然后按确定，进入“设置项目文件”部分。

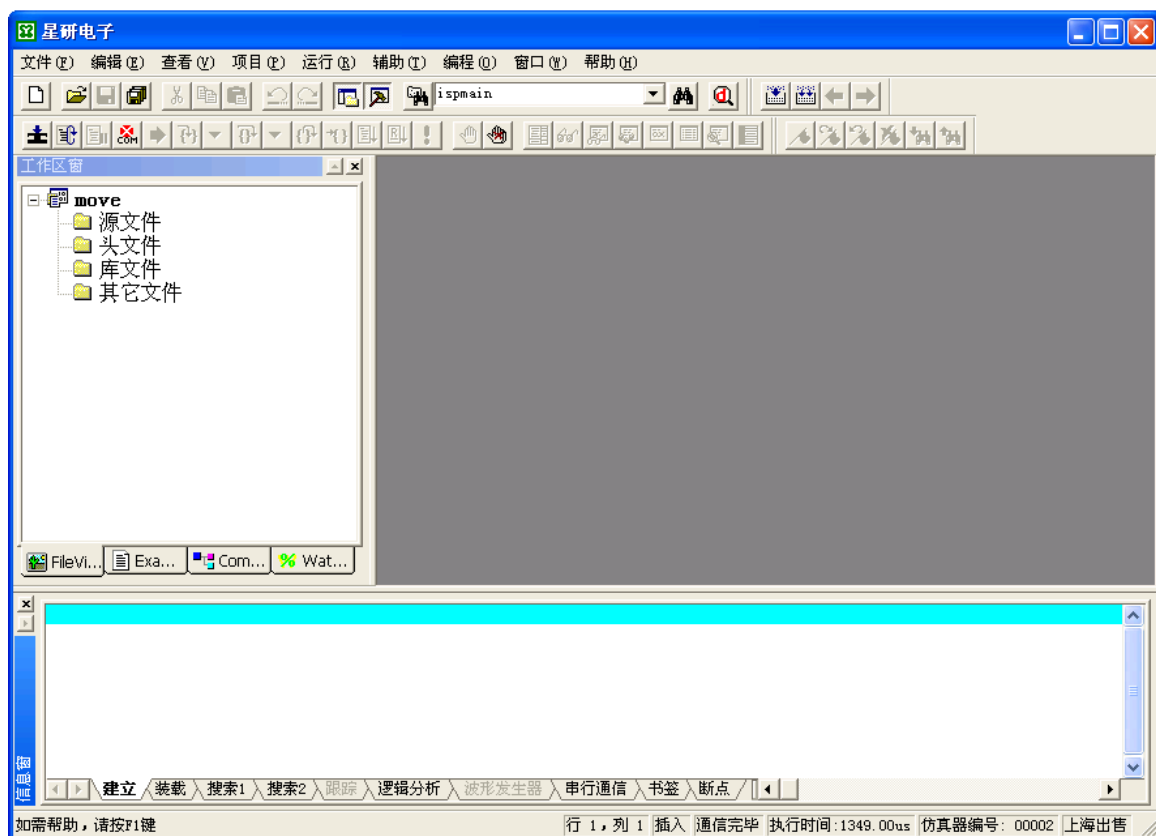
#### 2、设置项目文件

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

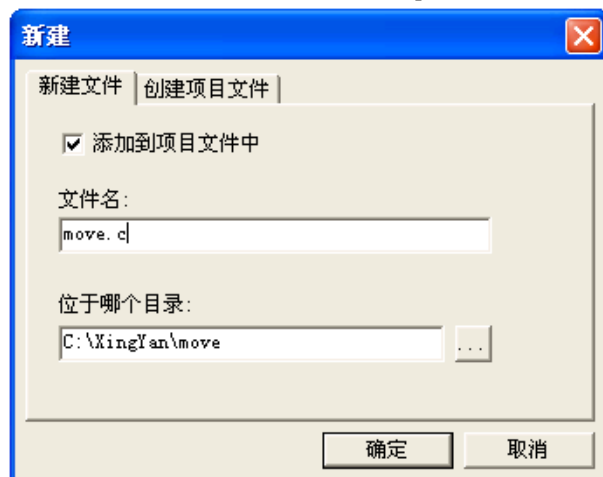
建立好项目文件的窗口如下图所示：





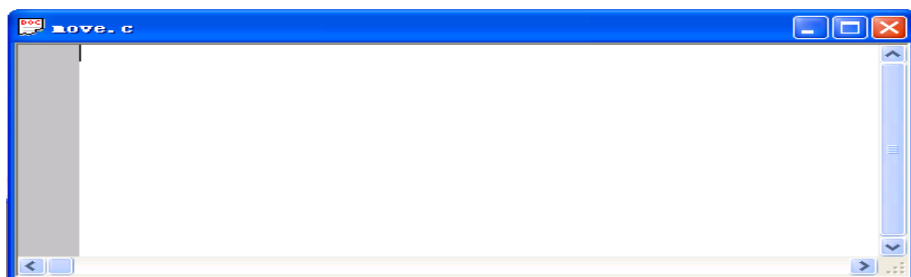
工作区窗的项目视图中，包含“源文件”、“头文件”、“库文件”、“其它文件”，“其它文件”中通常包含对该项目用途作一些说明的文件。“库文件”通常包含编译软件自带的 OBJ 文件、LIB 等库文件。

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.c。

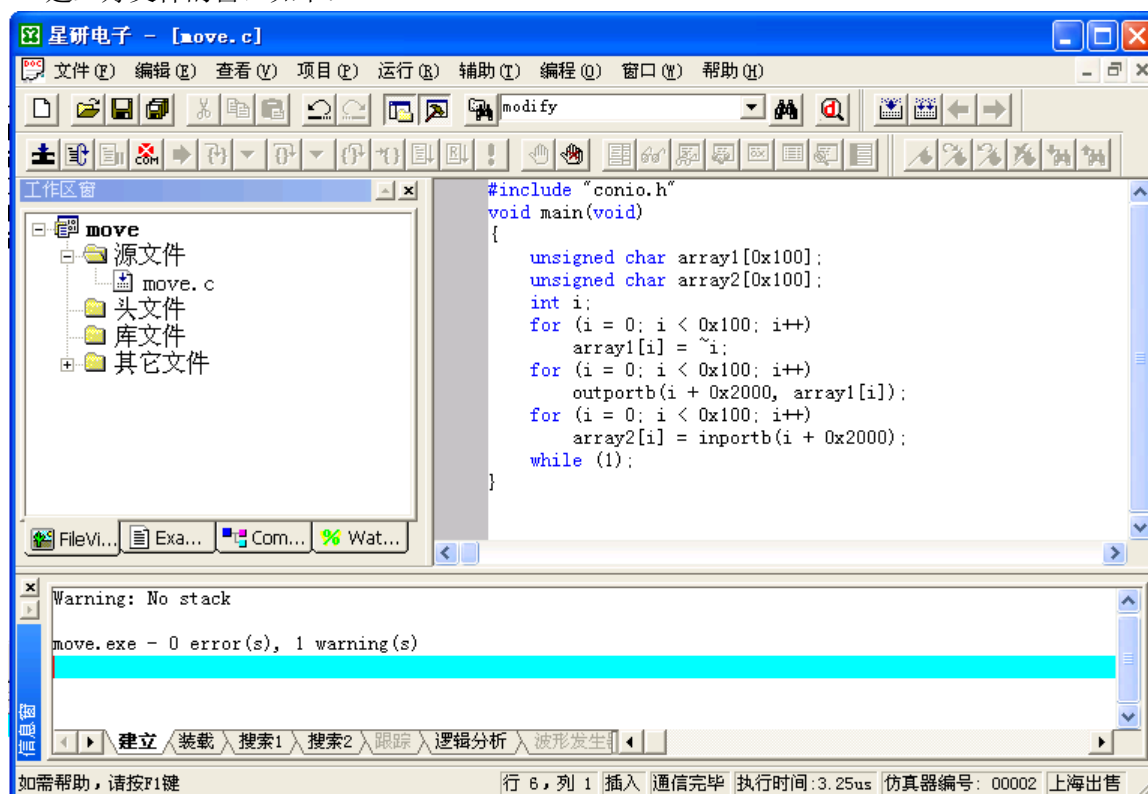
按“确定”即可。然后即出现文件编辑窗口：



程序清单：

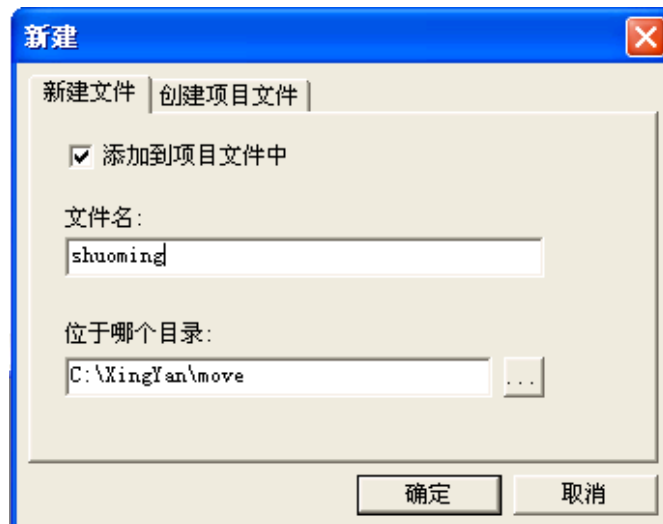
```
#include "conio.h"
void main(void)
{
    unsigned char array1[0x100];
    unsigned char array2[0x100];
    int i;
    for (i = 0; i < 0x100; i++)
        array1[i] = ~i;
    for (i = 0; i < 0x100; i++)
        outportb(i + 0x2000, array1[i]);
    for (i = 0; i < 0x100; i++)
        array2[i] = inportb(i + 0x2000);
    while (1);
}
```

建立好文件的窗口如下：

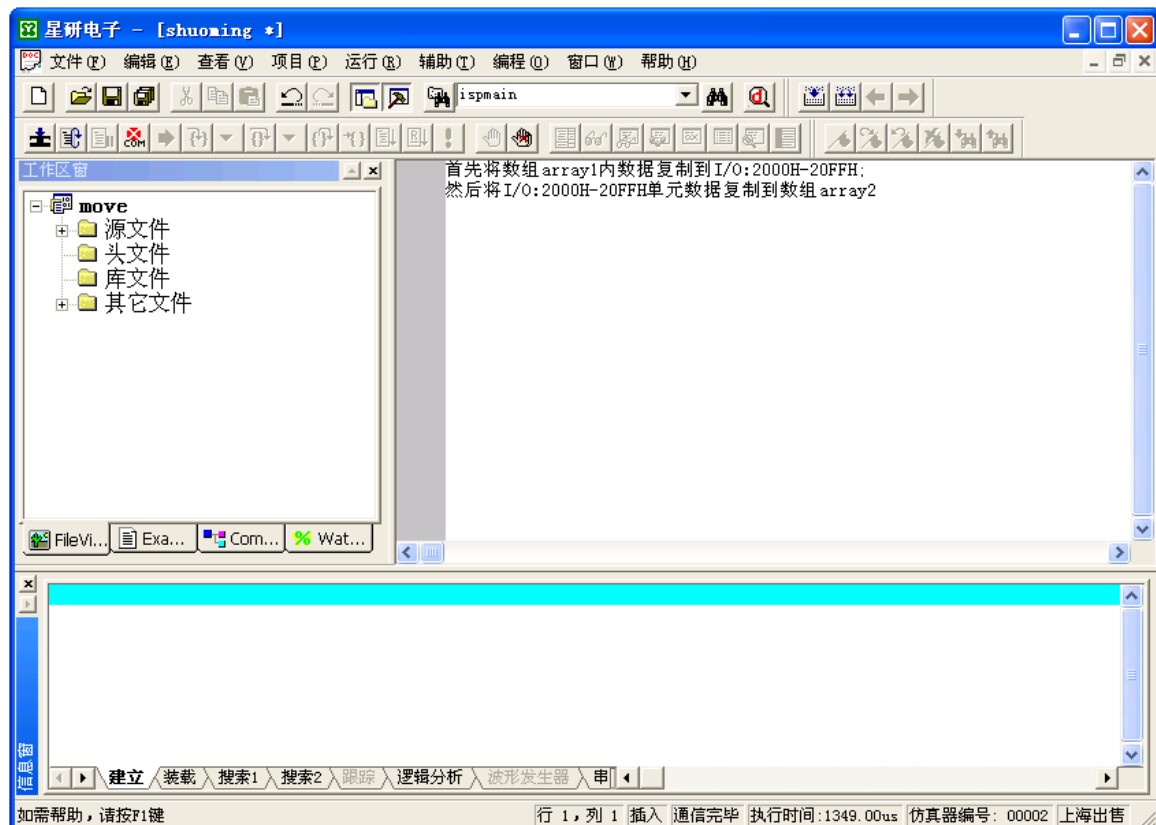


附：

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）点击图标或来“编译、连接”或“重新编译连接”。

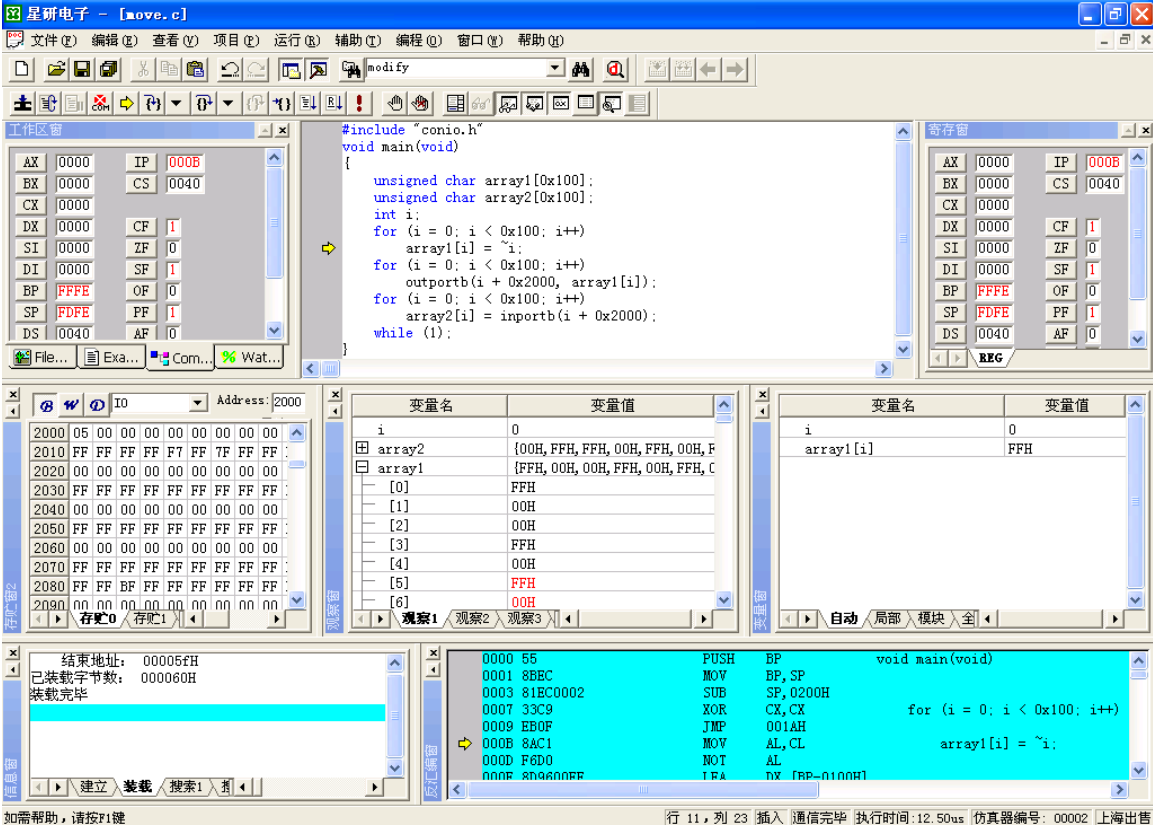
“编译连接”与“重新编译、连接”区别：“重新编译、连接”不管项目中有无添加、删除

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。所以把这些窗口放在前台，并调整至适当的大小。如图所示：



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。

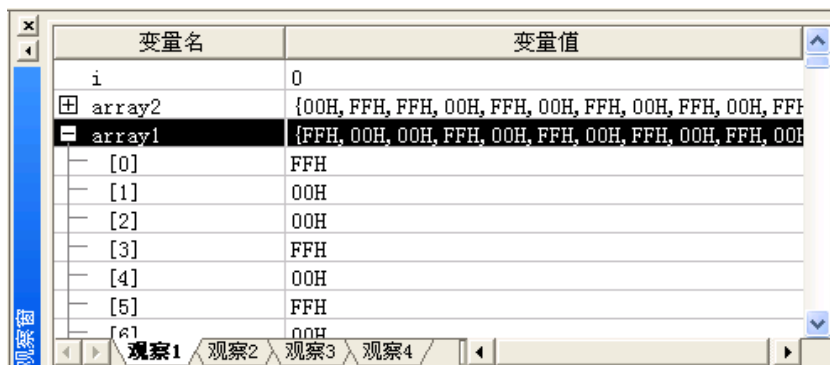
局部：显示当前函数或当前过程中的所有变量。

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，在编辑框中输入一个变量名，输入回车即可。观察窗中的四个页面项的作用完全一样，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一个方面。比如：我们在观察窗中添加变量 i, array1, array2 观察，如图：

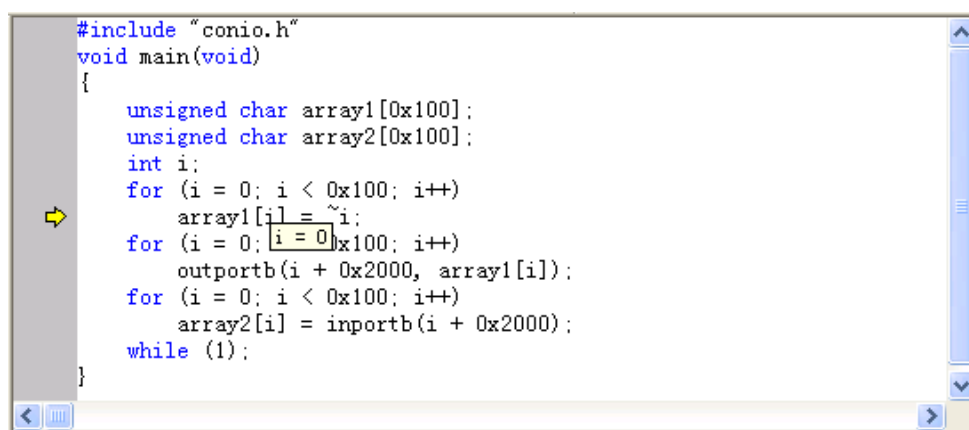


这也就方便了用户的调试和分析。

在第一个循环中，连续执行单步操作，我们可以看到 i 的值在递增，0FFH~00H 赋给 array1[0]~array1[255]；第二个循环中，连续执行单步操作，我们可以看到数组 array1 复制给 I/O: 2000H~20FFH；第三个循环中，连续执行单步操作，我们可以看到 I/O: 2000H~20FFH 单元数据复制给数组 array2；比较数组 array1、array2 中数据，判断程序是否正确。

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，这样就方便了下一段程序的检测。

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。



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。

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

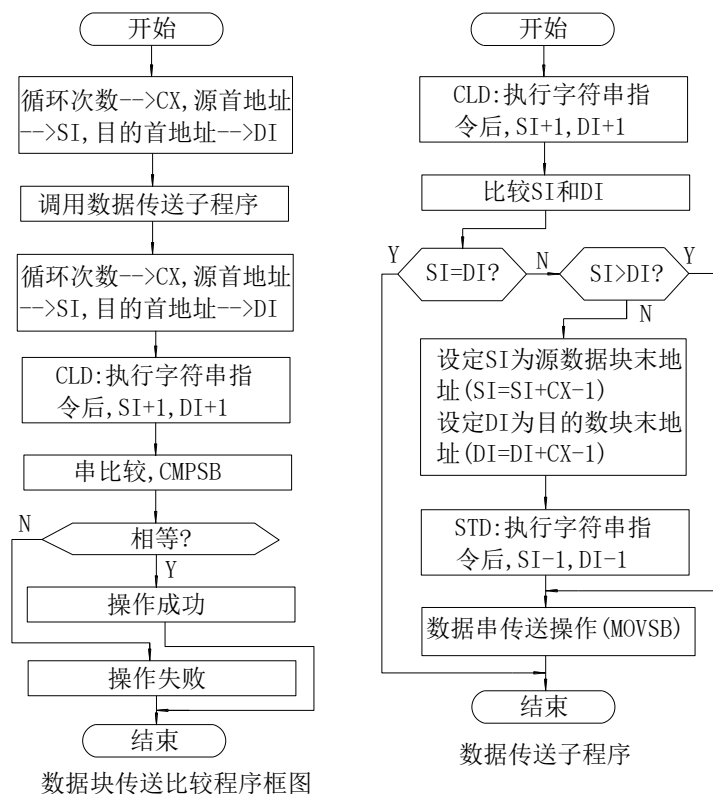
### 一、实验目的

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



### 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            100
                . DATA

                . 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                Move1
                ADD                SI, CX
                DEC                SI
                ADD                DI, CX
                DEC                DI
                STD
Move1:          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）。

2、编写一个程序，将 DS 段中的数据传送到实验仪 B4 区的 61C256 中。

说明：B4 区的 61C256 在 I/O 设备区，使用  $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$  读写。

## 实验二 双字节 BCD 码(十进制数)加法

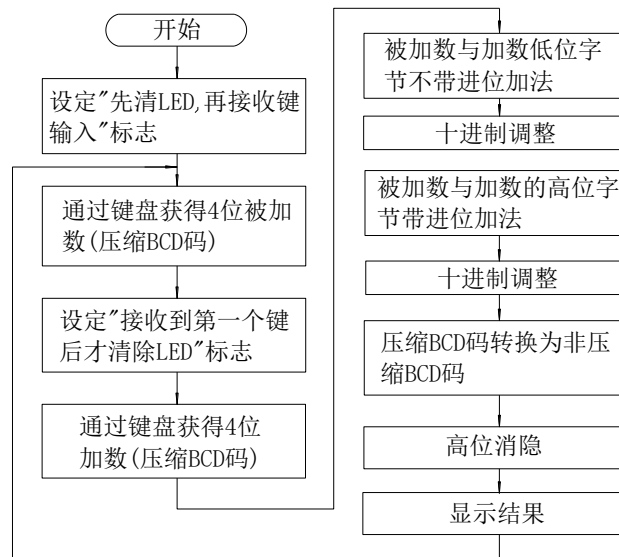
### 一、实验目的

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 段存取器的各种方法。

### 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:

E5 区 : CLK	——	B2 区: 2M
E5 区 : CS	——	A3 区: CS5
E5 区 : A0	——	A3 区: A0
E5 区 : A、B、C、D	——	G5 区: A、B、C、D

2、在 G5 区的键盘上输入 4 位被加数、加数

3、结果显示在 G5 区的数码管上

### 五、程序清单

```
.MODEL            TINY
EXTRN             Display8:NEAR,GetBCDKey:NEAR
EXTRN             F1: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		
	MOV	F1, 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	F1, 1	;接收到第一个键，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	B1toB2	;低位
	MOV	AL, AH	
	CALL	B1toB2	;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

B1toB2	PROC	NEAR	
	PUSH	AX	
	AND	AL, 0FH	
	STOSB		
	POP	AX	
	AND	AL, 0F0H	
	ROR	AL, 4	
	STOSB		
	RET		
B1toB2	ENDP		
BlackDisplay	PROC	NEAR	
	STD		
	MOV	DI, SI	
BlackDisplay1:	LODSB		;将高位0消隐
	CMP	AL, 0	
	JNZ	Exit	
	MOV	AL, 10H	
	STOSB		
	LOOP	BlackDisplay1	
Exit:	CLD		
	RET		
BlackDisplay	ENDP		
	END	START	

## 实验三 双字节 BCD 码(十进制数)减法

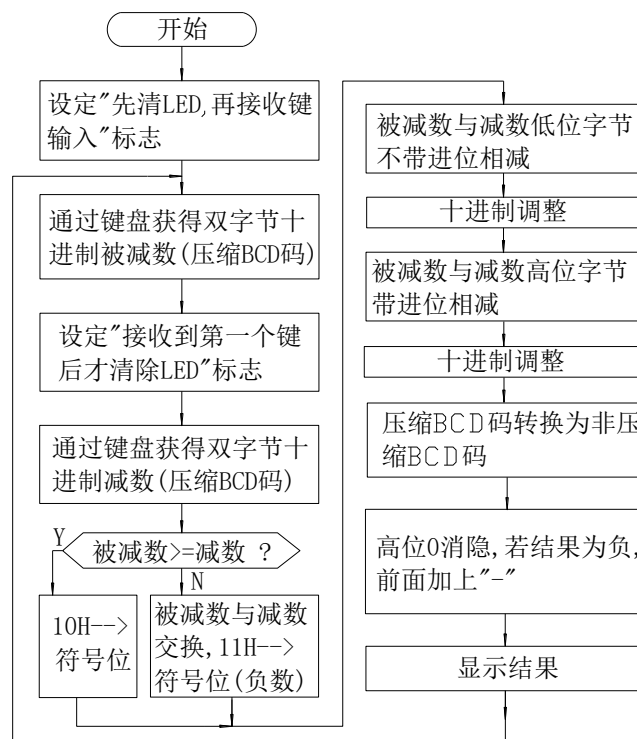
### 一、实验目的

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 段存储器的各种方法。

### 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:

E5 区 : CLK	——	B2 区: 2M
E5 区 : CS	——	A3 区: CS5
E5 区 : A0	——	A3 区: A0
E5 区 : A、B、C、D	——	G5 区: A、B、C、D

#### 2、在 G5 区的键盘上输入 4 位被减数、减数

#### 3、结果显示在 G5 区的数码管上

### 五、程序清单

.MODEL

TINY

```

EXTRN      Display8:NEAR, GetBCDKey:NEAR
EXTRN      F1: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
           MOV             F1,0          ;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             F1,1          ;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            B1toB2        ;低位
           MOV             AL,AH
           CALL            B1toB2        ;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
B1toB2      PROC          NEAR
PUSH         AX
AND          AL, 0FH
STOSB
POP          AX
AND          AL, 0F0H
ROR          AL, 4
STOSB
RET
B1toB2      ENDP
BlackDisplay PROC          NEAR      ;将高位0消隐
BlackDisplay1: MOV        AL, [SI]
CMP          AL, 0
JNZ          Exit
MOV          AL, 10H
MOV          [SI], AL
DEC          SI
LOOP         BlackDisplay1
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

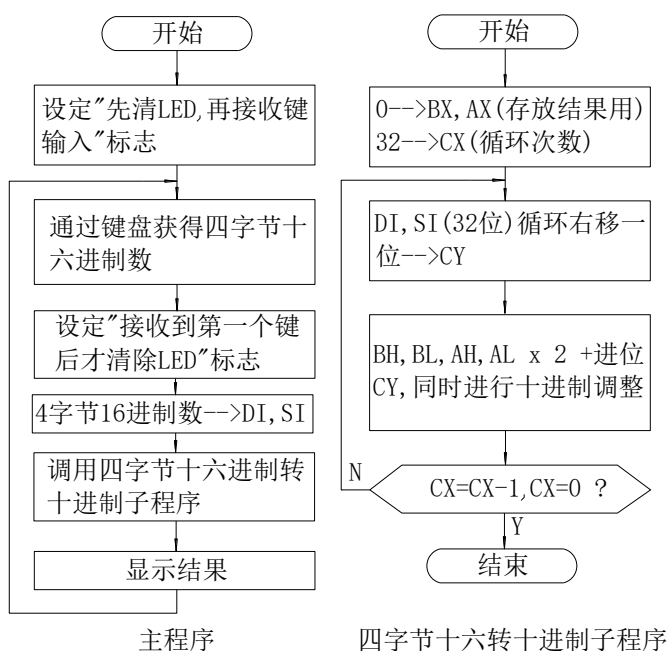
### 一、实验目的

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



### 四、实验步骤

1、连线说明：

E5 区：CLK	——	B2 区：2M
E5 区：CS	——	A3 区：CS5
E5 区：A0	——	A3 区：A0
E5 区：A、B、C、D	——	G5 区：A、B、C、D

2、在 G5 区的键盘上输入 8 位十六进制数

3、结果显示在 G5 区的数码管上

### 五、程序清单

```

.MODEL            TINY
EXTRN             Display8:NEAR, GetKey:NEAR
EXTRN             F1:BYTE
.STACK            100
  
```

```

        . DATA
BUFFER  DB          8 DUP (?)

        . CODE
START:  MOV          AX, @DATA
        MOV          DS, AX
        MOV          ES, AX
        NOP
        MOV          F1, 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          F1, 1          ;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         B1toB2         ;低位
        MOV          AL, AH
        CALL         B1toB2
        MOV          AL, BL
        CALL         B1toB2
        MOV          AL, BH
        CALL         B1toB2
        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
B1toB2  PROC          NEAR
        PUSH         AX
        AND          AL, 0FH
        STOSB
        POP          AX
        AND          AL, 0F0H
        ROR          AL, 4
        STOSB
        RET
B1toB2  ENDP
BlackDisplay  PROC      NEAR
        STD
        MOV          DI, SI
BlackDisplay1: LODSB                ;将高位0消隐

```

```

                                CMP            AL, 0
                                JNZ            Exit
                                MOV            AL, 10H
                                STOSB
                                LOOP           BlackDisplay1
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_1:                      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_1
                                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

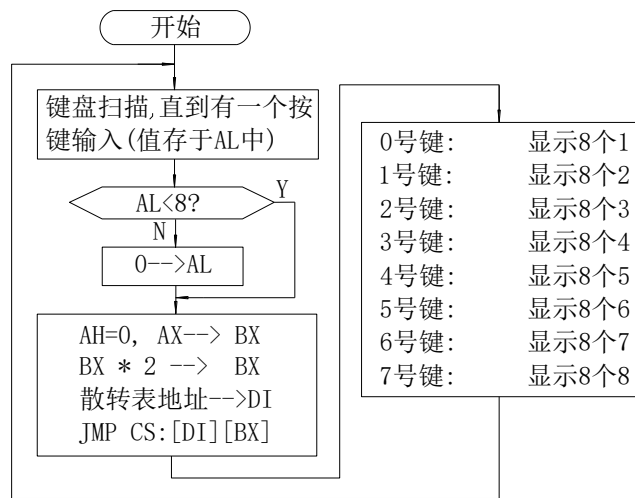
### 一、实验目的

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



散转程序流程框图

### 四、实验步骤

1、连线说明：

E5 区 : CLK	——	B2 区: 2M
E5 区 : CS	——	A3 区: CS5
E5 区 : A0	——	A3 区: A0
E5 区 : A、B、C、D	——	G5 区: A、B、C、D

2、在 G5 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 G5 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              Display8:NEAR, GetKeyB:NEAR
. STACK            100
. DATA
BUFFER            DB              8 DUP(?)
. CODE
START:            MOV              AX, @DATA
```

```

MOV DS, AX
MOV ES, AX
NOP
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_1
JMP CS:[DI][BX]
Table_1: DW Key0, Key1, Key2, Key3, Key4, Key5, Key6, Key7
Key0: MOV AL, 1
JMP Key
Key1: 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
MOV CX, 60000
LOOP $ ;延时
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

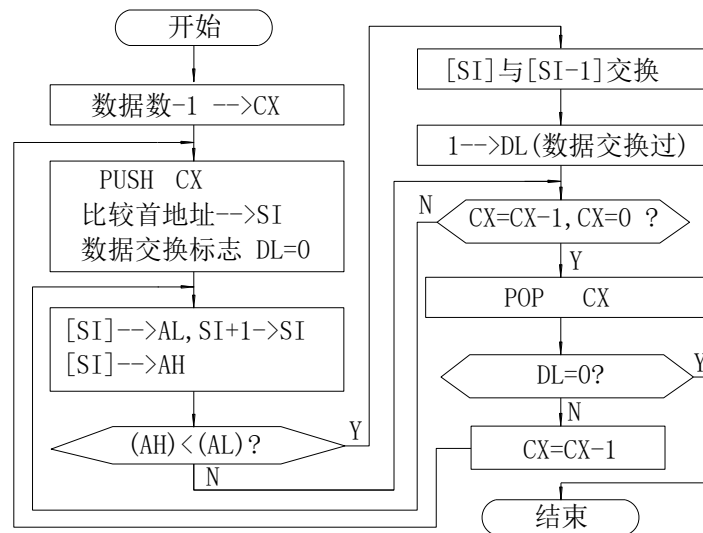
### 一、实验目的

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、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。

### 五、程序清单

```

.MODEL            TINY
.STACK            100
.DATA
TAB_1:            DB  0H, 5H, 6H, 3H, 8H, 92H, 04H, 57H, 46H, 01H, 0FFH, 0A0H, 45H, 99H, 55H, 66H
.CODE
START:            MOV     AX, @DATA
                  MOV     DS, AX
                  NOP
                  MOV     CX, 16 - 1          ;存放比较次数 = 数据个数 - 1
STAR2:            PUSH    CX
                  LEA     SI, TAB_1
                  MOV     DL, 0              ;0->交换过数据标志
STAR3:            LODSB
                  MOV     AH, [SI]
                  CMP     AH, AL

```

	JNB	STAR5	
	MOV	[SI], AL	
	MOV	[SI - 1], AH	
	MOV	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

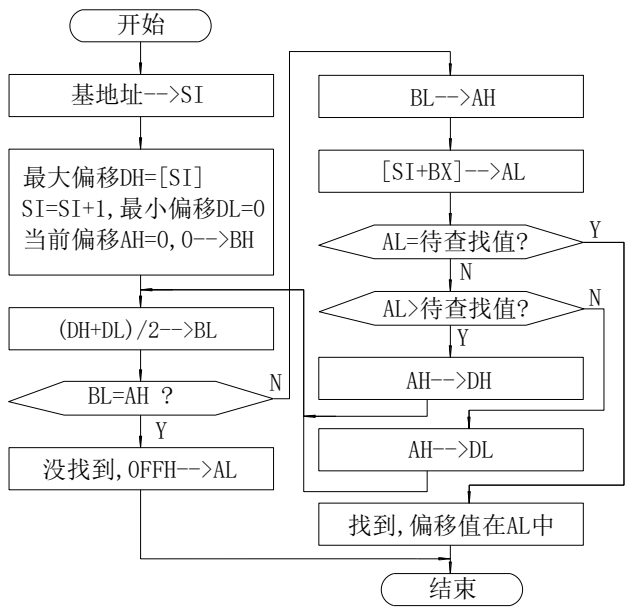
## 一、实验目的

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



二分查找法流程框图

## 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
Search_Data      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, 38, 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, 0FFH       ;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？