

**伟福®**

**伟福 Lab8000 系列  
单片机仿真实验系统**

**使用说明书**

**南京伟福实业有限公司**

## 目 录

第一章 概述.....	1
第二章 伟福实验系统组成和结构.....	3
1. 实验系统的硬件组成.....	3
2. 实验系统的 仿真板简介.....	15
3. 实验系统的调试方法.....	16
第三章 MCS51 系列单片机实验 .....	17
MCS96 系列单片机实验 .....	18
8088/86 系列CPU实验 .....	19
ARM LPC2103 MCU实验 .....	20
PIC5X系列CPU实验 .....	20
软件实验	
1. 存储器块清零(51/96/88/PIC).....	21
2. 二进制到 BCD 码转换(51/96/88/PIC).....	22
3. 二进制到 ASCII 码转换(51/96/88/PIC).....	23
4. 内存块移动(51/96/88/PIC).....	24
5. 程序跳转表(51/96/88/PIC).....	25
6. 数据排序(51/96/88/PIC).....	26
硬件实验	
1. IO 口输入输出(51/96/PIC/ARM) .....	27
2. 继电器控制(51/96/PIC/ARM) .....	29
3. 用 74HC245 读入数据(51/96/88/ARM) .....	30
4. 用 74HC273 输出数据(51/96/88/ARM) .....	31
5. PWM转换电压实验(51/96/PIC/ARM) .....	32
6. 音频控制(51/96/PIC/ARM) .....	33
7. 用 8255 输入、输出(51/96/88/ARM) .....	34
8. 串行数转换并行数(51/96/PIC/ARM) .....	35
9. 并行数转换串行数(51/96/PIC/ARM) .....	37
10. 计数器实验(51/PIC/ARM) .....	39
11. 外部中断实验(51/96/ARM) .....	40
12. 定时器实验(51/96/PIC/ARM) .....	42
13. D/A数模转换实验(51/96/88/ARM) .....	44
14. A/D模数转换实验(51/96/88/ARM) .....	46
15. 外部中断实验(急救车与交通灯) (51/96/ARM).....	48
16. 八段数码管显示(51/96/88/PIC/ARM).....	50

---

17. 键盘扫描显示实验(51/96/88/ARM).....	52
18. 电子时钟(51/96/88/PIC/ARM).....	54
19. 单片机串行口通讯实验(51/96/ARM).....	56
20. 1-Wire总线实验(51/96/PIC/ARM).....	58
21. 直流电机控制实验(51/96/88/ARM).....	60
22. 步进电机控制实验(51/96/88/PIC/ARM).....	62
23. 温度传感器实验(51/96/88/ARM).....	65
24. 液晶显示屏控制实验(51/96/88/ARM).....	67
25. 电子琴实验(51/96/88/ARM).....	68
26. 空调温度控制实验(51/96/88/ARM).....	70
27. 计算器实验(51/96/88/ARM).....	73
28. 用 HSO 方式输出 PWM 波形(96).....	75
29. 用 HSI 方式测量脉冲宽度(96).....	76
30. 用 HSI 中断方式统计脉冲个数(96).....	77
31. 计数器实验(96).....	79
32. 用片内 A/D 做模数转换实验(96).....	80
32. PWM 转换电压实验(88).....	81
34. 8253 计数器实验(88).....	82
35. 8259 外部中断实验(88).....	83
36. 8253 定时器实验(88).....	85
37. 8251A 串行口通讯实验(88).....	87
38. 8237 DMA 实验(88).....	89
39. 压力传感器实验(51/96/88/ARM).....	91
40. 红外通讯实验(51/96/88/ARM).....	92
41. 16x16 点阵显示实验(51/96/88/ARM).....	96
42. I2C总线实验(51/96/PIC/ARM).....	98
43. SPI总线实验(51/96/PIC/ARM) .....	100
 第四章 ARM LPC2103 仿真板说明.....	101
在KEIL和ADS开发环境中安装LAB8000 的驱动.....	103
在KEIL开发环境中安装LAB8000 的驱动.....	106
调试时可能出现的错误信息及原因.....	110
第五章 逻辑分析工具.....	111
第六章 系统自检功能.....	114

本实验说明书包括 8051、80C196、8088/86、ARM、PIC57 五种 MCU 的实验说明(MCS51 有 6 个软件实验、31 个硬件实验, MCS96 有 6 个软件实验、35 个硬件实验, 8088/86 有 6 个软件实验、25 个硬件实验, PIC57 有 6 个软件实验、14 个硬件实验, ARM 提供了 32 个硬件实验)。其中 6 个软件实验说明, 适合所有三种 CPU。有 42 个硬件实验说明, 分别适合不同的 CPU, CPU 类型见实验标题, 标有“51/96/88/ARM/PIC”适合五种 MCU、标有“51/96”适合 MCS51 和 MCS96、标有“51”只适合 MCS51 单片机、标有“96”只适合 MCS96 单片机, 标有“88”只适合 8088/86 CPU, 只标“ARM”只适合 ARM。

实验演示程序见软件光盘, 分汇编和 C 两种语言。MCS51 实验的汇编语言程序在“EX51ASM”目录下, C 语言程序在“EX51C”目录下, MCS96 实验的汇编语言程序在“EX96ASM”目录下, C 语言程序在“EX96C”目录下, 8088 实验汇编语言程序在“EX86ASM”目录下, C 语言程序在“EX86C”目录下。对于 PIC57 的实验只提供了汇编语言程序在“PIC5X”目录下, 对于 ARM LPC2103 开发板, 提供了 KEIL 和 ARM ADS 两种开发环境的 C 语言程序的样例。分别在“ARM\_KEIL”和“ARM\_ADS”两个目录下。

对于软件实验, 其项目名为 S1. PRJ... S6. PRJ。对于硬件实验, 其项目名为 H1. PRJ... H43. PRJ。若该 CPU 无此实验, 则相应的目录下无此项目。例硬件实验一“IO 口输入输出实验”不适合 8088/86, 那么在“EX86ASM”“EX86C”目录下就没有 H1. PRJ。有的实验有两种实验方法, 则分为 A、B 两个项目, 例如硬件实验 8“串行数换并行数”, 有 P1 口和串口两种实验方法, 分成 H8A. PRJ 和 H8B. PRJ 两个项目。对于 ARM LPC2103, 我们在做实验时, 除了用 IO 口, 用模拟的总线以外, 也考虑到 MCU 内部的资源, 比如 AD 变换, 我们有总线实验的例子, 也有用片内 ADC 进行转换的实验例子, 分别用不同的项目表示。例如硬件实验 5“PWM 转换电压实验”, 用片内 PWM 功能的实验以 H5B 表示, 例子保存在 H5B 文件夹下。还有硬件实验 14“ADC 模数转换实验”, 用片 ADC 模块的实验例子保存在 H14B 文件夹下。

### 关于 ARM LPC2103 开发板的说明:

当做 IO 口模拟总线的实验时, 都需要将开发板上的 GPIO/BUS 开关拨到“BUS”一侧; 而在做不用总线的实验时, 建议将开关拨到“GPIO”一侧, 以避免 MCU 的 IO 信号跟实验仪上的总线信号有冲突。(详细可见后面的 LPC2103 开发板说明章节)。

我们在 KEIL 环境下和 ARM ADS 环境下都有实验的样例程序, 每个项目中都有 Simulator(软件模拟)、DebugInOnchipFlash(在片上 Flash 内调试)、DebugInOnchipRAM(在片上 RAM 内调试)三个设置组, 默认为 DebugInOnchipFlash。为了避免片上 Flash 反复擦写而影响芯片寿命, 用户可以选择“在片内 RAM 内调试”方式。每次切换设置组必须重新编译。在片内 RAM 中调试时, 程序代码和数据不能超过 0x2000 (代码和数据空间我们分别设置为 0x1800 和 0x800, 你可以根据具体程序做相应的设置调节)。给出的所有实验程序中都共用的两个包含文件 LPC210X.H 和 LPC210X\_BUS.H 我们放在了一个共用文件夹下。而其它与各个实验程序相关的包含文件我们放在了项目所在的文件夹下。

用 ADS 调试, 首先要求安装 keil ARM 组件 RealView, 因为 Flash 的下载算法是直接利用 keil 的算法(位于 x:\keil\ARM\Flash\), 建议安装在 keil 默认路径(C:\KEIL), 因为默认的编程算法搜索路径是 c:\keil\ARM\Flash\, 如安装在其他路径, 需要修改编程算法路径。

更详细的说明可见后面的 LPC2103 开发板说明章节

## 第一章 概述

为了更好的发展教育，提高学生的计算机应用能力，根据本公司对市场的调研，现推出 LAB8000 改进型伟福通用微控制器仿真实验系统。本仿真实验系统由板上仿真器、实验仪、伟福仿真软件、开关电源构成。实验仪提供强大的逻辑分析、波形输出和程序跟踪功能，可以让学生直观地观察到单片机内部及外部电路工作的波形。

### 1.1 系统实验板

本实验板提供以下实验电路和模块

- (1) 逻辑电平输入开关
- (2) 逻辑电平显示电路
- (3) 单脉冲电路
- (4) 扬声器驱动电路
- (5) 继电器控制电路
- (6) 逻辑笔电路
- (7) 1MHz 和 10MHz 时钟脉冲信号。
- (8) PWM 转换电压电路
- (9) 模拟量电压(电位器)电路
- (10) 串口通信实验电路
- (11) 六位 8 段码 LED 数字显示器
- (12) 4x6 键盘
- (13) 存储器
- (14) 8255 端口扩展电路
- (15) 模数变换电路，可接入两路模拟量。
- (16) 数模变换电路，提供 0~-5V，-5V~+5V，-8V~+8V 三路输出
- (17) 液晶屏显示电路
- (18) 直流电机实验模块
- (19) 步进电机实验控制模块
- (20) 温度传感器实验模块
- (21) 压力传感器实验模块
- (22) 红外通讯实验模块
- (23) 16x16 点阵显示实验模块
- (24) I<sup>2</sup>C 总线实验模块
- (25) SPI 总线实验模块
- (26) 1-Wire 实验模块
- (27) 8251A 串行口扩展电路

- (28) 8253 定时器扩展电路
- (29) 8259 中断扩展电路
- (30) 8237 DMA 扩展电路
- (31) 地址译码输出模块
- (32) 逻辑分析仪数字采样和可编程数字脉冲信号输出模块
- (33) 虚拟示波器

## 1.2 仿真器系统构成

本仿真实验系统具有两种使用方法：

- (1) 有 PC 机，用 PC 机上的集成调试软件驱动板上仿真器进行仿真和实验。
- (2) 无实验仪、无仿真器，仅在 PC 机上采用软件模拟方式进行仿真。

- 1.3 实验仪可以直接进行 MCS51 系列实验；升级后可以进行 8088/86 实验和 PIC57 实验；配 EX96C 仿真板，可进行 80C196 的实验；配 ARM 仿真板，可以进行 ARM 实验。也可以根据学校要求设计其它 MCU/CPU 的控制板，进行该 MCU 的实验
- 1.4 配备 PC 机集成调试软件，在有系统机的情况下，通过板上仿真器实现 64K 全空间的硬件断点和仿真。
- 1.5 PC 机和系统机软件具有全集成化仿真环境，中、英文两种界面，软件仿真与硬件仿两种模式，软件仿真可以在无仿真仪的情况下进行。
- 1.6 配有 MCS51 系列、80C196 系列和 8088 系列的学生实验指导书，并提供了丰富的实验实例及实验程序，实验程序采用了机器码、汇编、C 等三种语言编写。

综上所述，本实验仪可以方便灵活地构成各种实验方案，在有无系统机和实验仪的情况下，都能进行相应的编程实验，从而具有极为广泛的应用范围，板上提供了基本的实验电路，减少繁琐的连接线过程，实验程序采用多种语言适应不同层次的学生需要。高级语言编写应用程序，是一种时代的需要，通过应用高级语言的编程和实验，可使学生掌握高级语言的编程方法，为今后进入社会实践打下坚实的基础，而汇编语言又能让学生了解机器深层的原理。

各个学校可以根据自身的具体情况，选择相应的实验项目。如果需要进行实验指导书以外的实验，本公司可以代编程序，并尽可能的提供各种技术支持。

## 第二章 实验系统组成和结构

实验系统可根据教学实践的需要实现 MCS51/MCS196 单片机原理与接口、8088/8086 微机原理与接口的一系列实验，并在硬件上预留了自主开发实验的空间。对基本实验仅需少量连接线就可完成，减轻学员工作量。同时也提供了需较多连线的扩展性实验，以进一步锻炼学员的实践开发能力。此外，系统还为学员们提供了强大的软、硬件调试手段。

### § 2.1 实验系统主机的硬件组成

本实验系统主机上有丰富的实验电路模块和灵活的组成方法，既可以和 MCS51、MCS96 CPU 也可以和 8088 / 8086CPU 组合完成各种实验。本实验仪成功高档通用仿真器所具有的逻辑分析仪、波形发生器和程序跟踪器等强大的分析功能，让学生在实验时不仅能了解程序的执行过程，更能直观地看到程序运行时的时序或者电路上的信号。

#### 2.1.1 逻辑电平开关电路

实验仪上有 8 只开关 K0—K7，并有与之相对应的 K0—K7 引线孔为逻辑电平输出端。开关向上拨相应插孔输出高电平“1”，向下拨相应插孔输出低电平“0”。

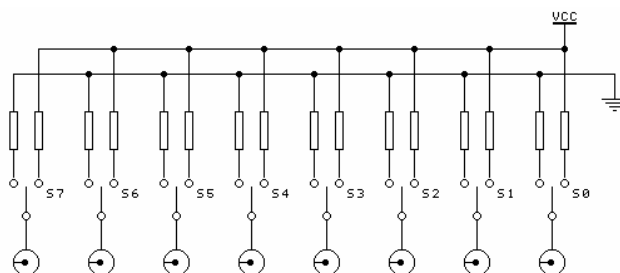


图 1： 逻辑电平开关电路

#### 2.1.2 LED 电平显示电路

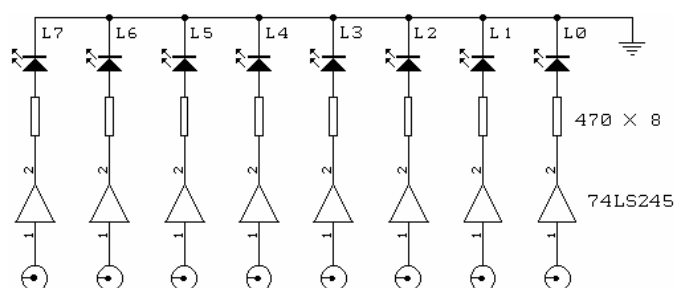




图 2： LED 电平显示电路

实验仪上装有 8 只发光二极管及相应驱动电路。见图 2，L0—L7 为相应发光二极管驱动

信号输入端，该输入端为高电压电平“1”时发光二极管点亮。我们可以通过 P1 口对其直接进行控制，点亮或者熄灭发光二极管。

### 2.1.3 单脉冲电路

单脉冲电路由按键 (PULSE) 和去抖动处理组成, 每按一次 (PULSE) 键 产生一个单脉冲。板上有单脉冲的输出信号插孔, 图为 “” 和 “”, 分别为正脉冲和负脉冲。

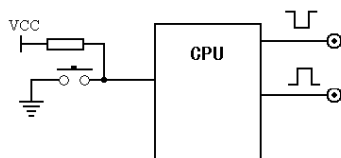


图 3: 单脉冲发生电路

### 2.1.4 音频放大电路

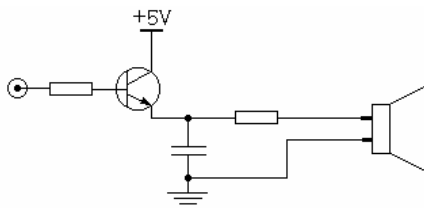


图 4: 音频放大滤波电路

### 2.1.5 继电器输出电路

当控制端电平置高，公共触点与常开端吸合。我们可以将常开端接入一发光二极管，公共端接+5V 电平，通过对控制端进行控制，观察发光二极管的状态。见图 5。

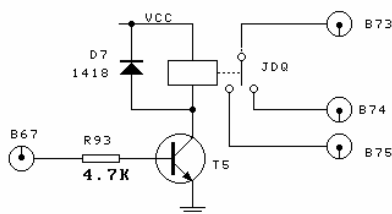


图 5: 继电器控制电路



### 2.1.7 逻辑测量（逻辑笔）电路

本实验仪上有逻辑测量电路，如图 11。可用于测量各种电平，其中红灯亮表示高电平，绿灯亮表示低电平。如果两灯同时闪动，表示有脉冲信号；两灯都不亮时，表示浮空（高阻态）。

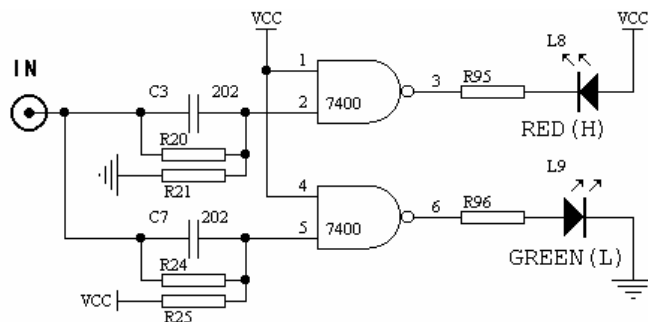


图 7： 逻辑笔电路

### 2.1.8 脉冲信号 10 MHz 和 1MHz

下图是 10MHz 脉冲信号和 1MHz 输出电路。

### 2.1.9 PWM 转换电

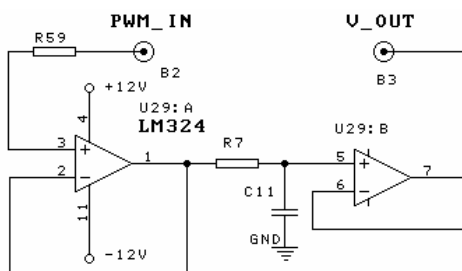


图 9： PWM 转换电路

### 2.1.10 可调模拟量输入电路

电位器电路用于产生可变的模拟量（0-5V）。

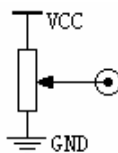


图 10： 电位器

### 2.1.11 串口通信程序实验插孔

做串行通信时，如果不需要将 TTL 电平转到 RS232 电平，可直接将 POD51/96 仿真板或 POD8086 仿真板上的 TXD、RXD 与通信对方交叉对接，并且共地即可。

单片机与标准的串行设备通信，需要将 TTL 电平转到 RS232 电平或将 RS232 电平转成 TTL 电平。本实验仪提供用户串行通信接口，可以用这两个插孔进行 RS232 通信程序实验，经电平转换后，再通过实验仪的“用户串口”接到 PC 机或其它 RS232 设备，实现数据互传。

### 2.1.12 六位 LED 数码显示器

#### 2.1.13 4×6 键盘电路

本实验仪的 LED 显示电路和键盘电路如图 1。显示电路和键盘电路可以工作在内驱和外驱两种方式，内驱是用 CPU 总线方式驱动，通过总线读写外部设备的地址来控制显示和读入键盘码。外部驱动方式是直接用 I/O 方式驱动八段显示的段码、位码和键盘按键信息，这里的 I/O 控制可以用 CPU 的 I/O 口来实现控制，也可通过 8255 等 I/O 扩展电路来控制。内驱、外驱由板上的拨动开关控制。

**内驱方式：**将拨动开关拨到“内驱”位置，显示和键盘工作于内驱方式，显示控制的位码通过总线由 74HC374 输出，经 ULN2003 反向驱动后，做 LED 的位选通信号。位选通信号也可做为键盘列扫描码，键盘扫描的行数据从 74HC245 读回，374 输出的列扫描码经 245 读入后，用来判断是否有键被按下，以及按下的是什么键。如果没有键按下，由于上拉电阻的作用，经 245 读回的值是高，如果有键按下，374 输出的低电平经过按键被接到 245 的端口上，这样从 245 读回的数据就会有低位，根据 374 输出的列信号和 245 读回的行信号，就可以判断哪个键被按下。LED 显示的段码由另一个 74HC374 输出。

键盘和 LED 显示电路的地址译码见图，做键盘和 LED 实验时，需将 KEY/LED CS 接到相应的地址译码上。位码输出地址为 0X002H，段码输出地址为 0X004H，键盘行码读回地址为 0X001H，此处 X 是地址高四位，由 KEY/LED CS 决定。例如将 KEY/LED CS 接到地址译码的 CS0 上，那么位码输出地址就为 08002H，段码输出地址就是 08004H，键盘行码读回地址为 08001H。

**外驱方式：**将拨动开关拨到“外驱”位置，八段显示和键盘工作于外驱方式，八段管的段码控制由输出端口输出到 A~H 插孔。G0~G5 是八段管的位码，同时也是键盘列扫描信号，I/O 口输出的信号一方面可以点亮一位八段管，另一方面向键盘输出列扫描信号。K0~K3 是键盘行信号，I/O 口可以从这里读到键盘按下的信息，与列扫描一起可以判断是哪个键被按下。

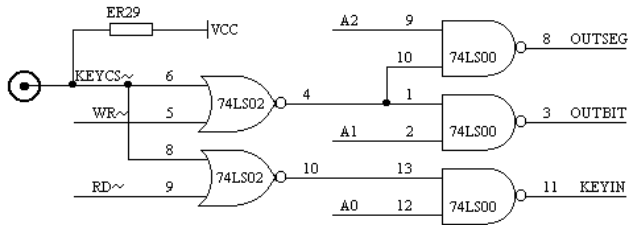


图 12-13: 键盘及 LED 显示电路

### 2.1.14 存储器电路

本实验仪上有一片 32K 存储器 61256。提供给学生做存储器实验，由于地址译码为 4K 一段，所以只能提供 4K 容量使用，地址从 0000H~0FFFH。用 RAM CS 来选择不同的地址段，以适应不同的应用电路。

### 2.1.15 8255 端口扩展电路

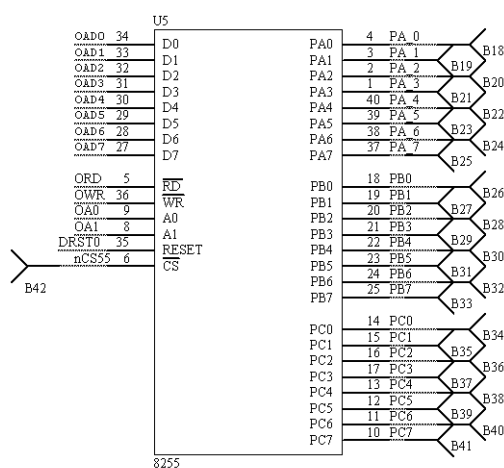


图 15: 8255 端口扩展电路

### 2.1.16 A/D 转换电路

实验仪上有一个 0~5V 的可调电位器，将可变电电压输出端接入 A/D 转换电路的输入端，通过 CPU 软件处理，读进 A/D 转换值，再将转换值送数码管显示。我们可以调节电位器，使之输出不同电压值，通过数码管的显示，检验 A/D 转换正确与否。

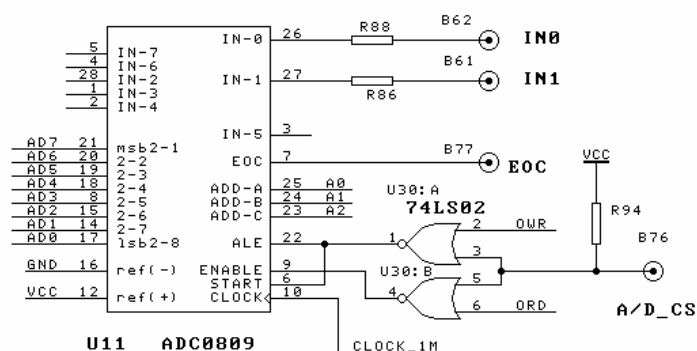


图 16: A/D 转换电路

### 2.1.17 D/A 转换电路

实验仪上提供了 D/A 转换电路如下图所示。我们可以通过软件编程控制 D/A 转换芯片 DAC0832，输出相应电流值，经过采样电路取出模拟量电压值，用电压表测量电压输出端子，读出电压值。

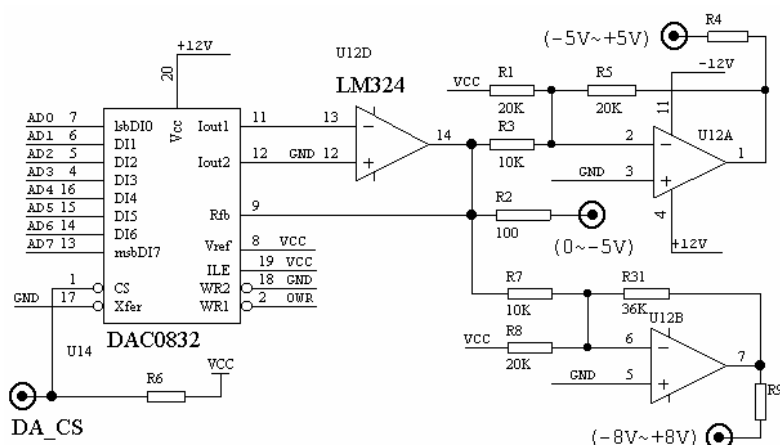


图 17: D/A 转换电路

### 2.1.18 液晶屏显示控制电路

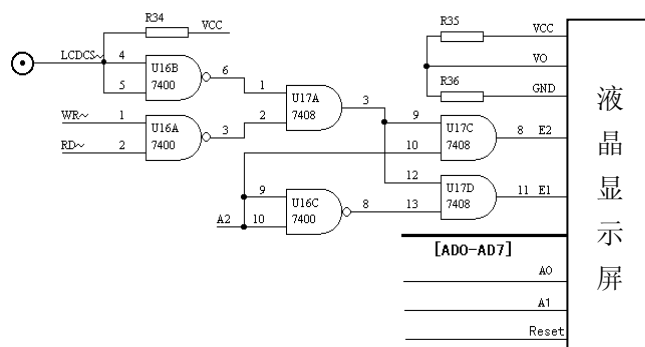


图 18: 液晶屏接口电路

## 2.1.19 直流电机电路

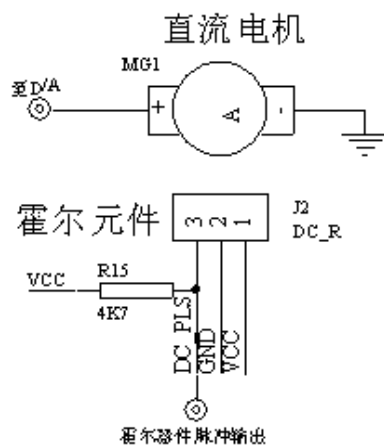


图 19: 直流电机/霍尔器件电路

## 2.1.20 步进电机电路

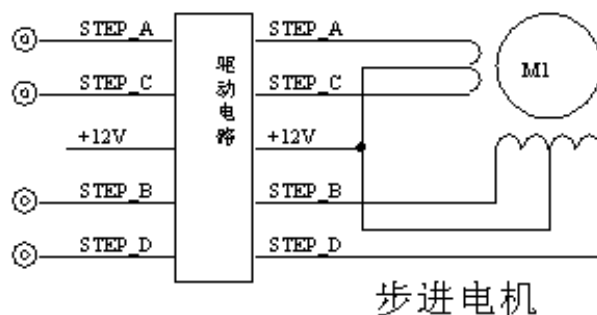


图 20: 步进电机驱动电路

## 2.1.22 温度传感器电路

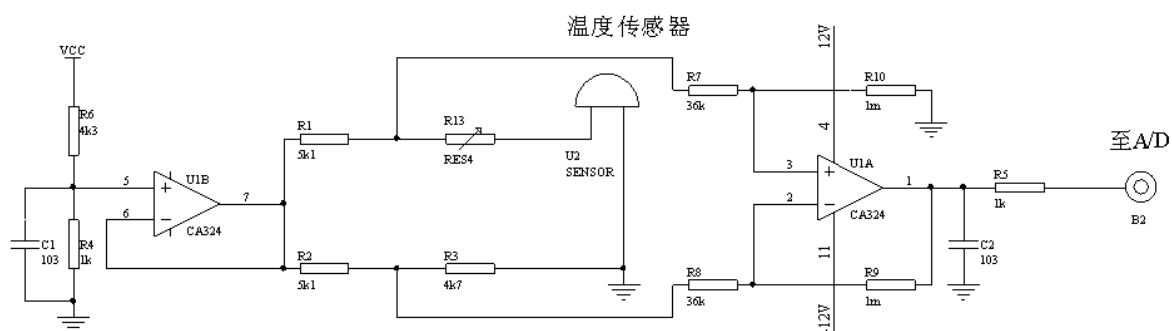


图 22: 温度传感器电路

## 2.1.23 压力传感器电路

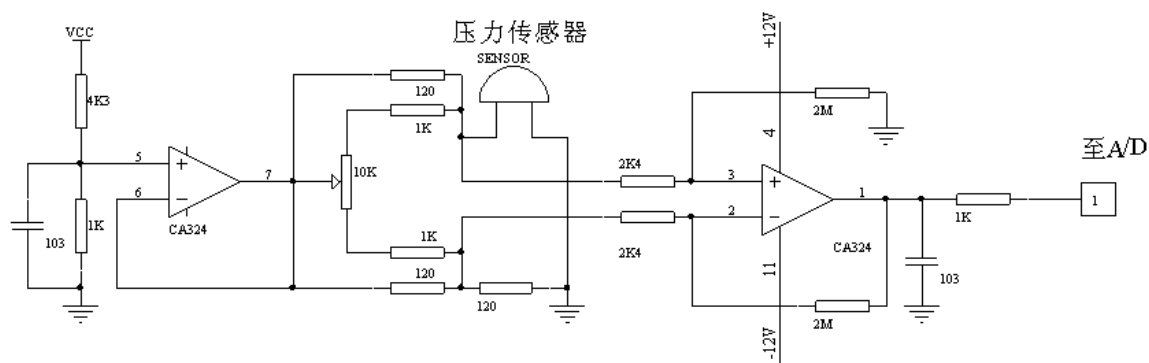


图 23: 压力传感器电路

## 2.1.24 红外通信电路

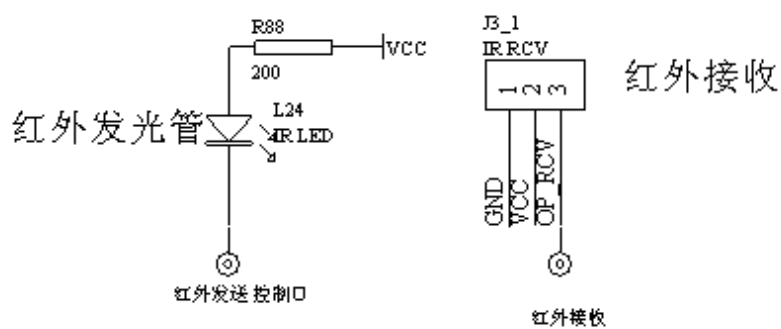


图 24: 红外通信电路

## 2.1.25 16×16 LED 点阵电路

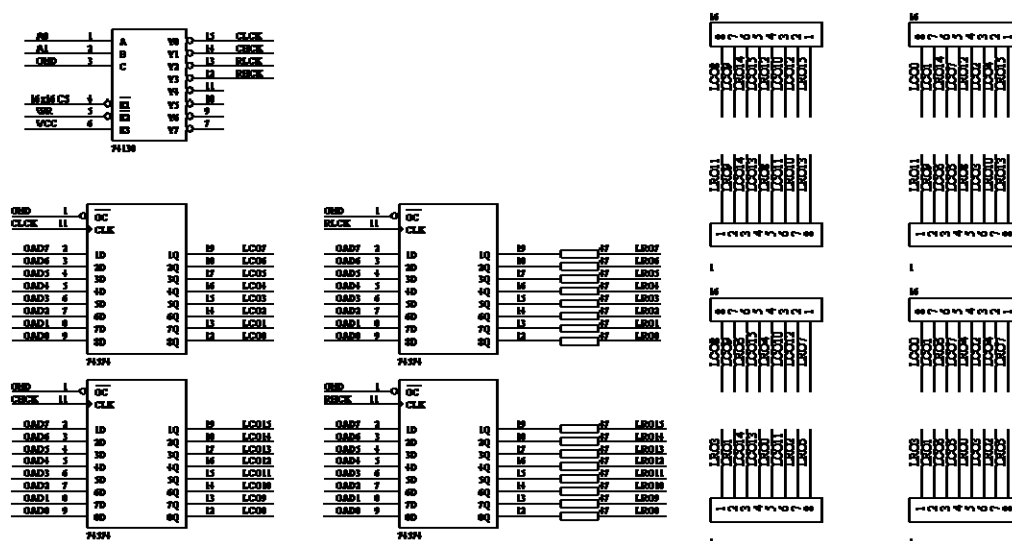
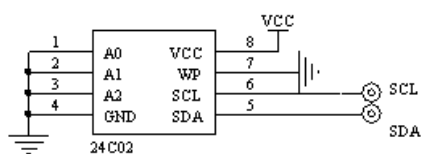


图 25: LED 点阵电路

2.1.26 I<sup>2</sup>C 总线实验电路图 26: I<sup>2</sup>C 总线电路

## 2.1.27 8251A 串行口扩展电路

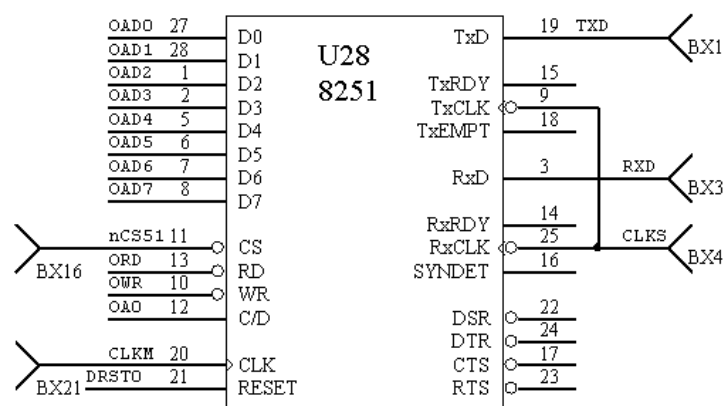


图 27: 8251A 串行口扩展电路



## 2.1.28 8253 定时器/计数器扩展电路

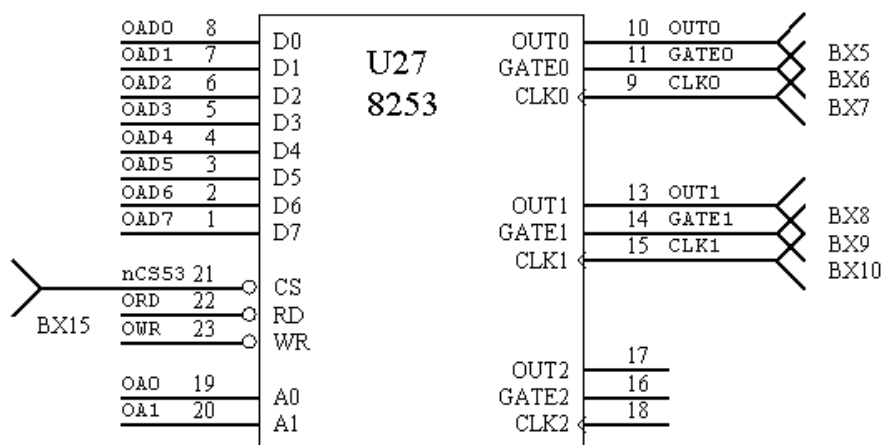


图 27: 8253 定时器/计数器扩展电路

## 2.1.29 8259 中断扩展电路

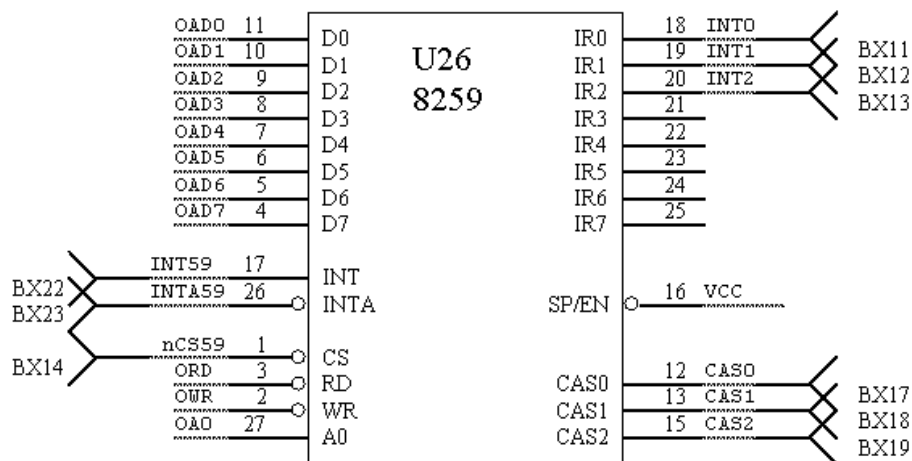


图 29: 8259 中断扩展电路

## 2.1.30 8237 DMA 扩展电路

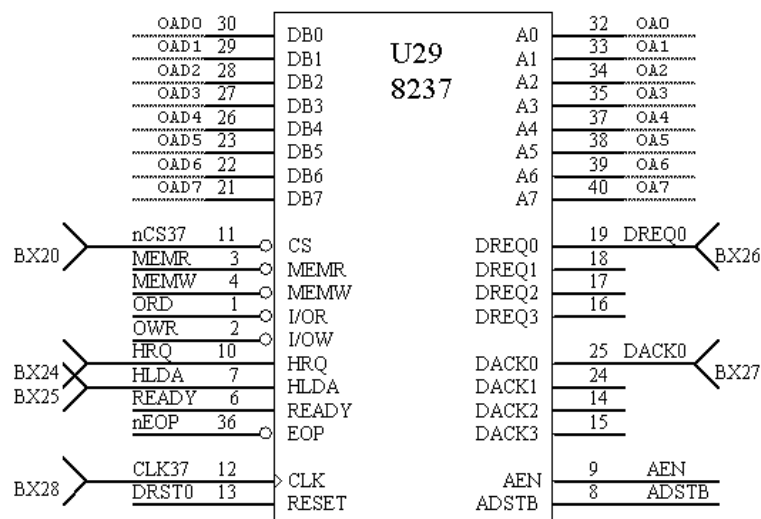


图 30: 8237 DMA 扩展电路

## 2.1.33 插孔

本实验仪上有 MCS51 芯片的 P0~P3 端口的插孔，端口功能与 MCS51 芯片一样，P0，P2 既可以当 I/O 口，又可以当总线使用，P3 可以当 I/O 口，也可以工作于第二功能，比如串口，中断，定时，读写等等到。如果是进行 PIC57 实验，相关的插孔还可以做为 PIC57 的管脚，实现 PIC57 的相关功能（PIC 的管脚用稍小的字体标出）。除些之外，如果实验仪支持 8088/86 实验，板上还提供了 8251 串行通信芯片的管脚插孔，8253 定时器/计数器芯片的管脚插孔，8259 外部中断处理芯片及 8237DMA 芯片的管脚接出插座，在做实验时，将信号接到相关芯片的管脚插孔上即可。

## 2.1.32 地址译码插孔

片选号	地址范围
CS0	08000H~08FFFH
CS1	09000H~09FFFH
CS2	0A000H~0AFFFH
CS3	0B000H~0BFFFH
CS4	0C000H~0CFFFH
CS5	0D000H~0DFFFH
CS6	0E000H~0EFFFH
CS7	0F000H~0FFFFH

## § 2.2 实验系统的仿真板简介

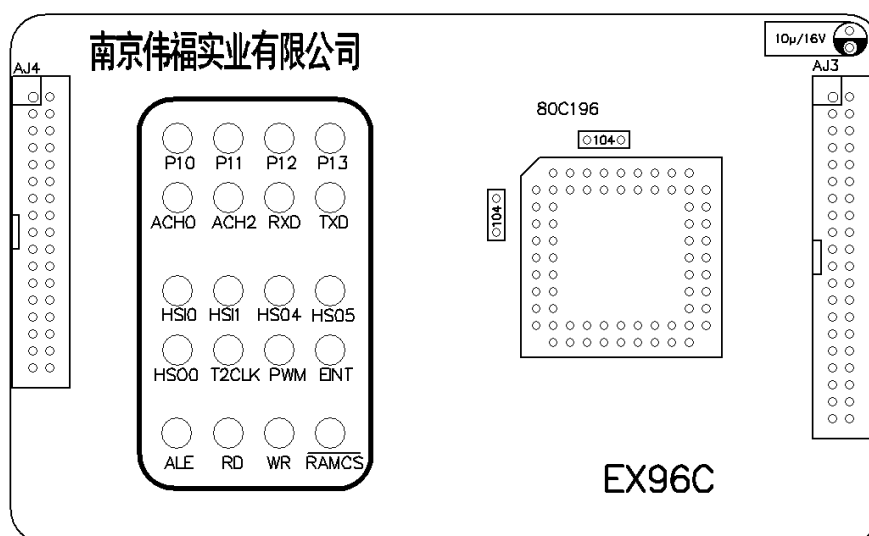
### 2.2.1 MCS51 仿真板

MCS51 的仿真板已经集成到实验仪的仿真电路中了。仿真板的总线已经接到实验仪的总线器件上，做总线实验时，连接好 CS 即可，在实验仪上也接出了 MCS51 的 P0~P3 输入输出端口，I/O 实验时只要接到相应的接线柱即可。

本实验仪具有外接仿真功能，将实验仪附带的 40 芯仿真插头的电缆插到 AJ3 插座上，就是一台 MCS51 的仿真器，对你自己的设计用户板进行仿真。

### 2.2.2 EX96C 仿真板

进行 MCS80196 系列实验时，需将 EX96C 仿真板插在实验仪上，EX96C 仿真板如下图：



仿真板插上后，MCS96 的总线也已经接到实验仪的总线器件上，做总线实验时接上 CS 即可，做其它实验时（片内 AD，I/O，HSI，等）板上留出了接线柱，接好相应的连线即可。

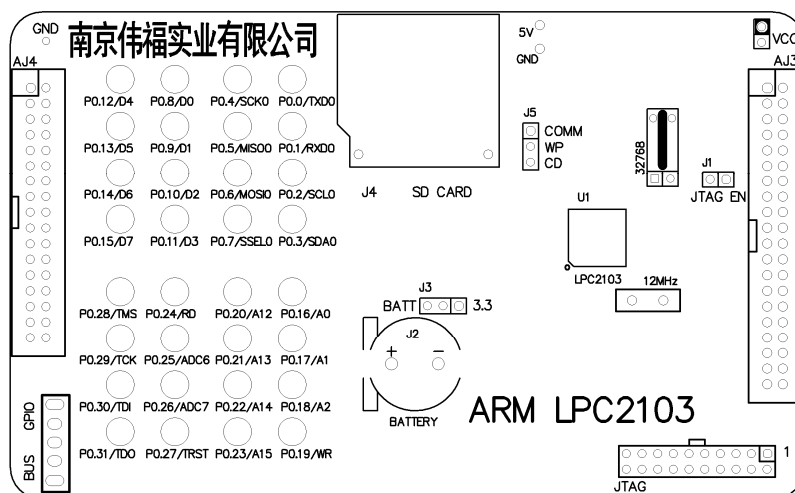
### 2.2.3 8088/86 仿真板

将实验仪的仿真电路软件硬件升级，不用外加仿真板就可以同时支持 i8088/86 的实验。同时仿真电路还提供了 8251 串行通信，8253 定时器，8259 外部中断，8237DMA 等电路功能，并在实验仪留出这些芯片的接口，用来进行相关实验。i8088/86 的总线也已经接到实验仪上的总线器件，实验时根据器件地址接好 CS 即可。

### 2.2.4 PIC5X 仿真板

PIC5X 的仿真板已经集成到实验仪的仿真电路中，不用外加仿真板就可以支持 PIC5X 的实验。在实验仪上接出了 PIC57 的 PA, PB, PC 端口，在实验仪以稍小的字体标出，实验时只需要在伟福开发环境中只要将仿真头选成“PIC5X”，将连线接到相应的接线柱即可。因为 PIC5X 没有外扩总线，所以所有跟总线有关的实验都暂未提供。

### 2.2.5 ARM LPC2103 仿真板



LPC2103 有 32 个 IO 端口已经全部接到接线柱，当做总线的实验时，都需要将开发板上的 GPIO/BUS 开关拨到“BUS”一侧，这样用 IO 口模拟的总线就会接到实验仪的总线器件，实验时按器件地址接好 CS，程序模拟的总线就会控制器件进行工作。而在做不用总线的实验时，建议将开关拨到“GPIO”一侧，以避免 MCU 的 IO 信号跟实验仪上的总线信号有冲突。

## § 2.3 实验系统的调试方法

- 1、使用 WAVE 集成调试软件进行联机仿真，有关 WAVE 集成调试软件的使用方法，参见《伟福仿真器使用说明书》。
- 2、当使用 ARM LPC2103 仿真板做实验时，要用到 KEIL 或 ADS 开发环境，如何在这两个开发环境下添加 LAB8000 实验的驱动，如何进行项目的设置，如何进行程序的开发和调试，本说明书都有介绍。如果要更详细有关开发环境的说明，请参考这两个开发环境附带的使用手册。

## 第三章 实验说明

### ▣ MCS51 系列单片机实验说明

#### § 3.1 系统的安装和启动

- 1、 仿真开发系统集成调试软件的安装和使用见 WAVE 仿真开发系统使用手册。
- 2、 用配套的 USB 通讯电缆将 PC 机和实验仪相联接。
- 3、 将实验台的电源线与 220V 电源相连。（实验结束后应拔下）
- 4、 打开实验台电源开关，红色电源指示灯亮。
- 5、 打开计算机电源，执行 WAVE 集成调试软件。

注意：

- 1、 论是集成电路的插拔、通讯电缆的连接、跳线器的设置还是实验线路的连接，都应确保在断电情况下进行，否则可能造成对设备的损坏。
- 2、 实验线路连接完成后，应仔细检查无误后再接通电源。

#### § 3.2 MCS51 系列单片机实验软件设置

WAVE 集成调试环境应设置如下：

仿真器型号：伟福 Lab8000 实验仪

仿真头型号：MCS51 实验 （8031 / 32）

## 📌 MCS96 系列单片机实验说明

### § 3.3 系统的安装和启动

- 1、仿真开发系统集成调试软件的安装和使用见 WAVE 仿真开发系统使用手册。
- 2、用户根据实验要求，进行 MCS96 单片机实验时，应插上 EX96C 仿真板。
- 3、用配套的 USB 通讯电缆将 PC 机和实验仪相联接。
- 4、将实验台的电源线与 220V 电源相连（实验结束后应拔下）。
- 5、打开实验台电源开关，红色电源指示灯亮。打开计算机电源，执行 WAVE 集成调试软件。

注意：

- 1、论是集成电路的插拔、通讯电缆的连接、跳线器的设置还是实验线路的连接，都应确保在断电情况下进行，否则可能造成对设备的损坏。
- 2、实验线路连接完成后，应仔细检查无误后再接通电源。

### § 3.4 MCS96 系列单片机实验软件设置

WAVE 集成调试环境应设置如下：

仿真器型号：伟福 Lab8000 实验仪

仿真头型号：MCS96 实验 （80C196）

如果用 C 语言调试验程序，请将‘LINK 命令行’设置为：

CSTART.OBJ, KC\_SFRS.OBJ, C96.LIB ROM(2000H-7FFFH) RAM(8000H-0FFFFH)

## 8088/86 CPU 系列实验说明

### § 3.5 系统的安装和启动

- 1、仿真开发系统集成调试软件的安装和使用见 WAVE 仿真开发系统使用手册。
- 2、用配套的 USB 通讯电缆联接好 PC 机和实验仪。
- 3、将实验台的电源线与 220V 电源相连。(实验结束后应拔下)
- 4、打开实验台电源开关, 红色电源指示灯亮。打开计算机电源, 执行 WAVE 集成调试软件。

注意:

- 1、无论是集成电路的插拔、通讯电缆的连接、跳线器的设置还是实验线路的连接, 都应确保在断电情况下进行, 否则可能造成对设备的损坏。
- 2、实验线路连接完成后, 应仔细检查无误后再接通电源。

### § 3.6 8088/8086 CPU 系列实验软件设置

WAVE 集成调试环境应设置如下:

仿真器型号: 伟福 Lab8000 实验仪

仿真头型号: 8088/86 实验 (8088/86)

语言设置对话框各命令行应设置如下:

编译器路径: 一般情况下为 C:\COMP86, 也可设为你自己的安装路径。

ASM 命令行: '/zi /l'

C 命令行: '-mt -v -y -c'

LINK 命令行: '/m /l /v'

## ✎ ARM LPC2103 MCU 实验说明

### § 3.7 系统的安装和启动

1. 进行 ARM LPC2103 相关实验，要求用户已经安装了 KEIL uV3 或 ARM ADS 集成开发环境。
2. 户根据实验要求，进行 ARM LPC2103 单片机实验时，应插上 ARM LPC2103 仿真板。
3. 配套的 USB 通讯电缆将 PC 机和实验仪相联接。
4. 实验台的电源线与 220V 电源相连（实验结束后应拔下）。
5. 开实验台电源开关，红色电源指示灯亮。打开计算机电源，打开 KEIL uV3 或 ARM ADS 集成调试软件。

注意：

1. 是集成电路的插拔、通讯电缆的连接、跳线器的设置还是实验线路的连接，都应确保在断电情况下进行，否则可能造成对设备的损坏。
2. 实验线路连接完成后，应仔细检查无误后再接通电源。

### § 3.8 LPC2103MCU 实验软件设置

如何在 KEIL uV3 或 ARM ADS 集成开发环境中添加伟福 LAB8000 实验仪的驱动，如何设置项目，如何编程，如何调试程序，请参见第四章“ARM LPC2103 仿真板的说明”

## ✎ PIC5X 单片机实验说明

### § 3.9 系统的安装和启动

1. 开发系统集成调试软件的安装和使用见 WAVE 仿真开发系统使用手册。
2. 用配套的 USB 通讯电缆将 PC 机和实验仪相联接。
3. 将实验台的电源线与 220V 电源相连。（实验结束后应拔下）
4. 打开实验台电源开关，红色电源指示灯亮。
5. 打开计算机电源，执行 WAVE 集成调试软件。

注意：

1. 论是集成电路的插拔、通讯电缆的连接、跳线器的设置还是实验线路的连接，都应确保在断电情况下进行，否则可能造成对设备的损坏。
2. 验线路连接完成后，应仔细检查无误后再接通电源。

### § 3.10 PIC5X 系列单片机实验软件设置

WAVE 集成调试环境应设置如下：

仿真器型号：伟福 Lab8000 实验仪

仿真头型号：PIC5X 实验 （PIC57）



## 软件实验一 存储器块清零

### 一、实验要求

指定存储器中某块的起始地址和长度，要求能将其内容清零。

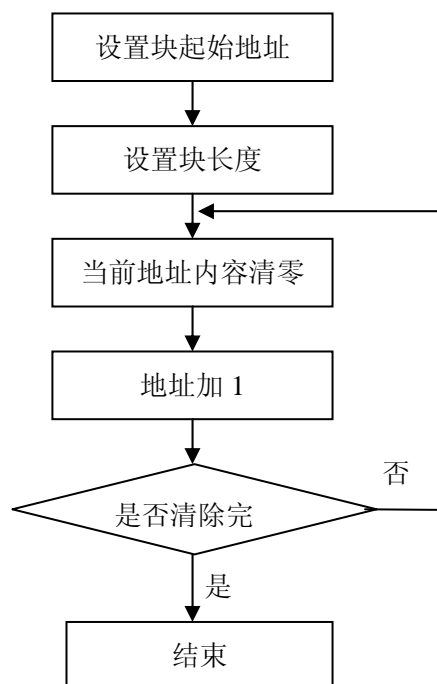
### 二、实验目的

1. 握存储器读写方法
2. 解存储器的块操作方法

### 三、实验说明

通过本实验，学生可以了解单片机读写存储器的读写方法，同时也可以了解单片机编程，调试方法。如何将存储器块的内容置成某固定值(例全填充为 0FFH)? 请学生修改程序，完成此操作。

### 四、程序框图



## 软件实验二 二进制到 BCD 转换

### 一、实验要求

将给定的一个二进制数，转换成二十进制（BCD）码

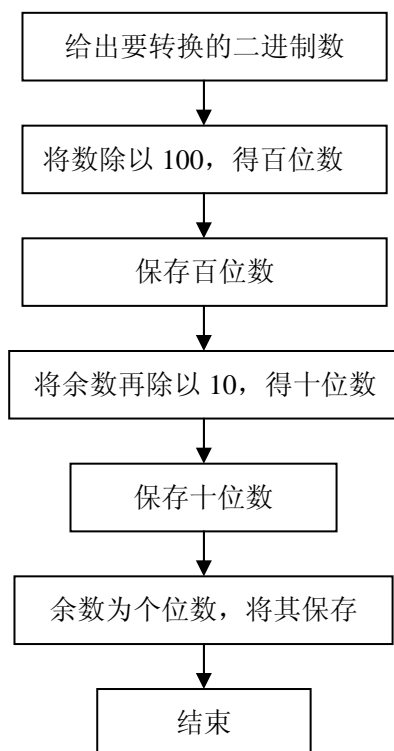
### 二、实验目的

1. 掌握简单的数值转换算法
2. 基本了解数值的各种表达方法

### 三、实验说明

计算机中的数值有各种表达方式，这是计算机的基础。掌握各种数制之间的转换是一种基本功。有兴趣的同学可以试试将 BCD 转换成二进制码。

### 四、程序框图



## 软件实验三 二进制到 ASCII 码转换

### 一、实验要求

给出一个 BCD 数，将其转换成 ASCII 值。

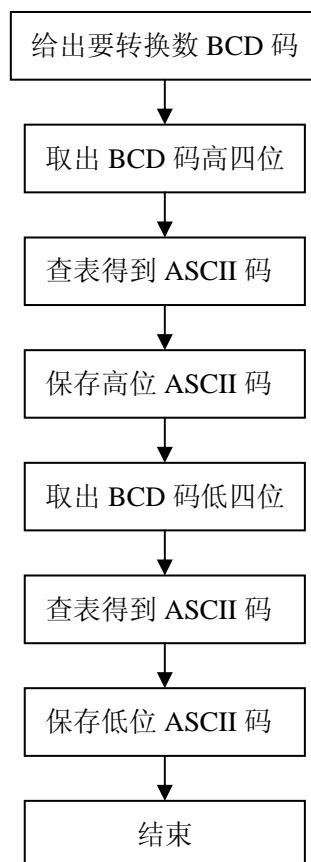
### 二、实验目的

1. 了解 BCD 值和 ASCII 值的区别
2. 了解如何将 BCD 值转换成 ASCII 值
3. 了解如何查表进行数值转换及快速计算。

### 三、实验说明

此实验主要让学生了解数值的 BCD 码和 ASCII 码的区别，利用查表功能可能快速地进行数值转换。进一步了解数值的各种表达方式。

### 四、程序框图



## 软件实验四 内存块移动

### 一、实验要求

将指定源地址和长度的存储块移到指定目标位置

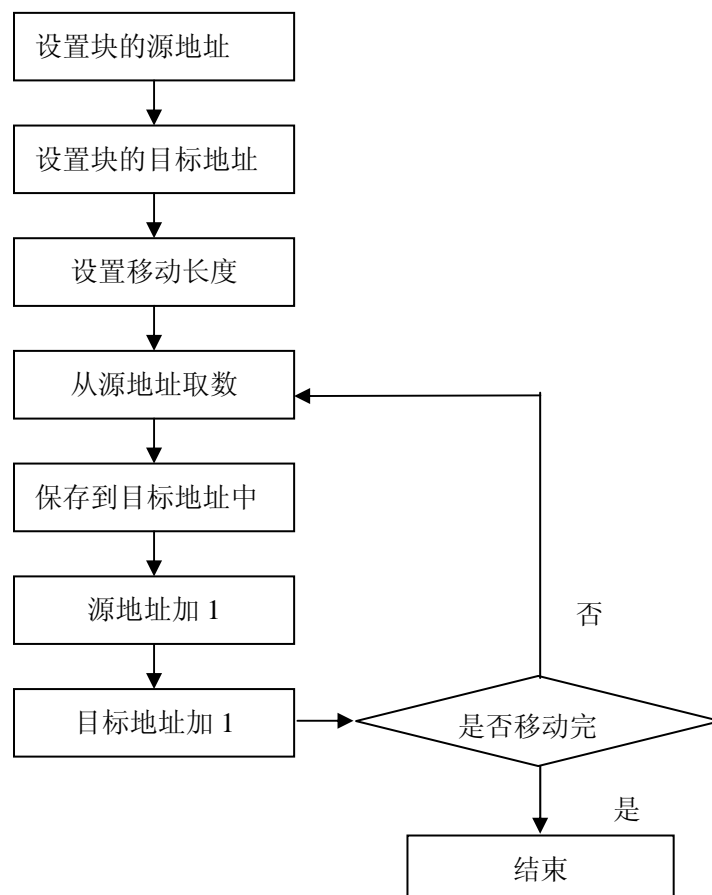
### 二、实验目的

1. 了解内存的移动方法。
2. 加深对存储器读写的认识。
- 3.

### 三、实验说明

块移动是计算机常用操作之一，多用于大量的数据复制和图象操作。本程序是给出起始地址，用地址加一方法移动块，请思考：给出块结束地址，用地址减一方法移动块的算法。另外，若源块地址和目标块地址有重叠，该如何避免？

### 四、程序框图



## 软件实验五 程序跳转表

### 一、实验要求

在多分支结构的程序中，能够按调用号执行相应的功能，完成指定操作。

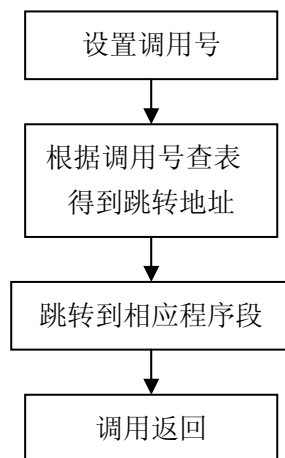
### 二、实验目的

1. 了解程序的多分支结构
2. 了解多分支结构程序的编程方法

### 三、实验说明

多分支结构是程序中常见的结构，若给出调用号来调用子程序，一般用查表方法，查到子程序的地址，转到相应子程序。

### 四、程序框图



## 软件实验六 数据排序

### 一、实验要求

给出一组随机数，将此组数据排序，使之成为有序数列

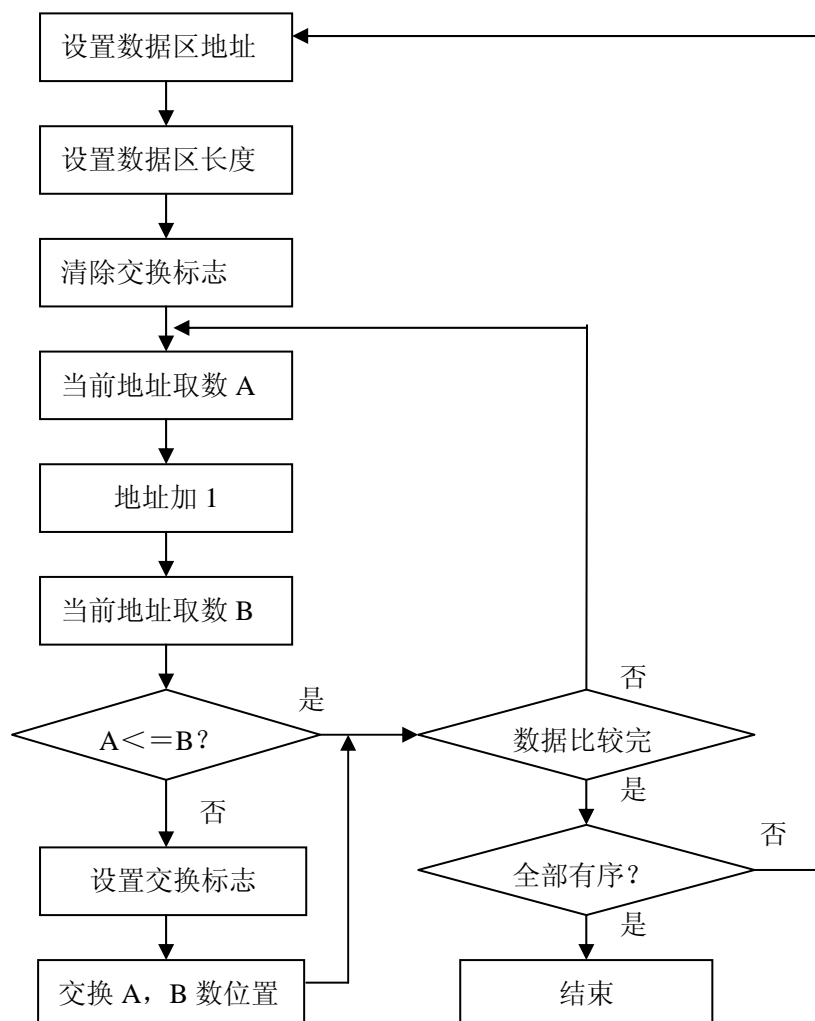
### 二、实验目的

1. 了解数据排序的简单算法。

### 三、实验说明

有序的数列更有利于查找。本程序用的是“冒泡排序”法，算法是将一个数与后面的数相比较，如果比后面的数大，则交换，如此将所有的数比较一遍后，最大的数就会在数列的最后面。再进行下一轮比较，找出第二大数据，直到全部数据有序。

### 四、程序框图



## 硬件实验一 I/O 口输入、输出实验 (51/96/PIC/ARM)

### 一、实验要求

1. P1 口做输出口，接八只发光二极管，编写程序，使发光二极管循环点亮。
2. P1.0、P1.1 作输入口接两个拨动开关，P1.2、P1.3 作输出口，接两个发光二极管，编写程序读取开关状态，将此状态，在发光二极管上显示出来。编程时应注意 P1.0、P1.1 作为输入口时应先置 1，才能正确读入值。
3. 对于 PIC 或 ARM MCU，选出四个 I/O 完成以上实验。

### 二、实验目的

1. 学习单片机的 I/O 口的使用方法。
2. 学习延时子程序的编写和使用。
3. 学习用‘与’‘或’运算对 MCS96 系列 CPU 的变量进行位操作。

### 三、实验电路及连线

连线	连接孔 1	连接孔 2
1	P1.0	L0
2	P1.1	L1
3	P1.2	L2
4	P1.3	L3

MCS51/96 的 P1 口循环点灯

连线	连接孔 1	连接孔 2
1	S0	P1.0
2	S1	P1.1
3	P1.2	L4
4	P1.3	L5

MCS51/96 的 P1 口输入输出

连线	连接孔 1	连接孔 2
1	PB0	L0
2	PB1	L1
3	PB2	L2
4	PB3	L3

PIC5X 的 PB 口循环点灯

连线	连接孔 1	连接孔 2
1	S0	PB0
2	S1	PB1
3	PB2	L4
4	PB3	L5

PIC5X 的 PB 口输入输出

连线	连接孔 1	连接孔 2
1	P0.4	L0
2	P0.5	L1
3	P0.6	L2
4	P0.7	L3

LPC2103 的 I/O 口循环点灯

连线	连接孔 1	连接孔 2
1	S0	P0.4
2	S1	P0.5
3	P0.6	L4
4	P0.7	L5

LPC2103 的 I/O 口输入输出

### 四、实验说明

1. 对于 MCS51/96MCU，P1 口是准双向口。它作为输出口时与一般的双向口使用方法相同。由准双向口结构可知当 P1 口用为输入口时，必须先对它置“1”。若不先对它置“1”，读入的数据是不正确的。
2. 由于 80C196 系列 CPU 没有位操作，所以要对 P1.0，P1.1 进行与运算，以判断该

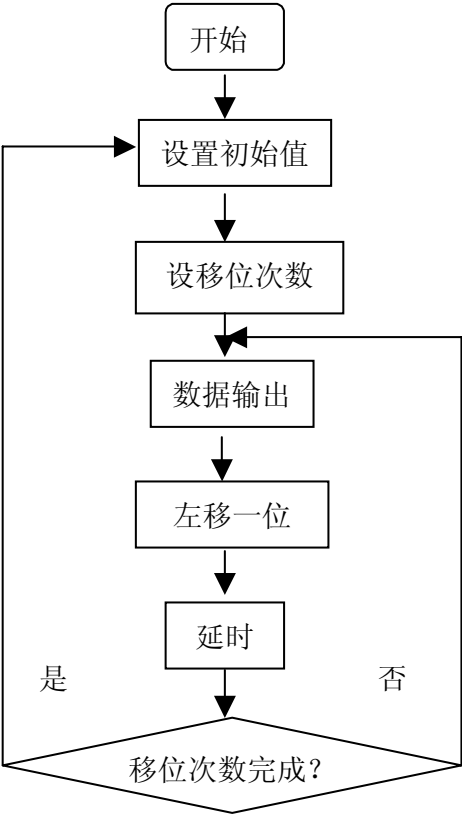
- 位为高还是为低，然后再用‘与’和‘或’运算将 P1.2, P1.3 的相应位置高或低，这与 80C51 系列 CPU 不同。80C51 可以将位变量通过 C 标志位赋值给其它位。
3. 对于 PIC 和 ARM MCU 输入输出端口需要设置方向，以确定端口工作于输入还是输出状态。
4. 8051 延时子程序的延时计算问题，对于程序

查指令表可知 MOV, DJNZ 指令均需用两个机器周期，在 6MHz 晶振时，一个机器周期时间长度为 12/6MHZ，所以该段程序执行时间为：

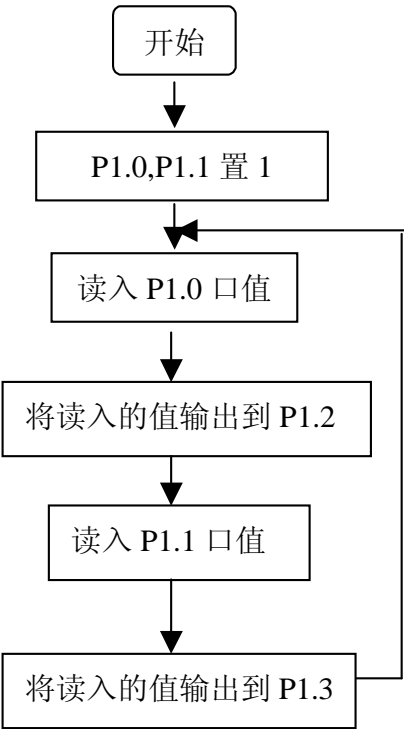
$$(256 \times 255 + 2) \times 2 \times 12 \div 6 \approx 261\text{ms}$$

```
Delay:
    MOV    R6, #0H
    MOV    R7, #0H
DelayLoop:
    DJNZ   R6, DelayLoop
    DJNZ   R7, DelayLoop
    RET
```

五、实验框图



(A) P1 口循环点灯程序框图



(B) P1 口输入输出程序框图



## 硬件实验二 继电器控制实验（51/96/PIC/ARM）

### 一、实验要求

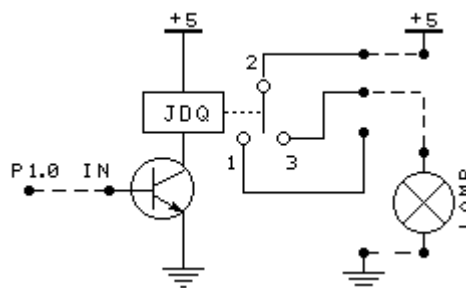
用单片机的端口，输出电平控制继电器的吸合和断开，实现对外部装置的控制。

### 二、实验目的

1. 学习 I/O 端口的使用方法
2. 掌握继电器的控制的基本方法
3. 了解用弱电控制强电的方法

### 三、实验电路及连线

连线	连接孔 1	连接孔 2
1	P1.0	继电器输入
2	5V	继电器常闭输入
3	L0	继电器中间输入



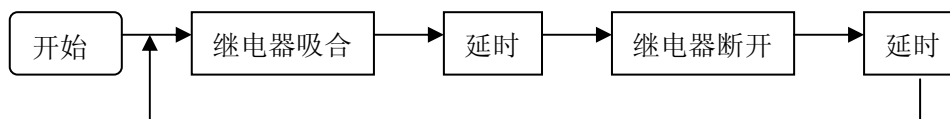
对于 PIC5X MCU，将 PB0 接到继电器输入，  
对于 LPC2103 MCU，将 P0.0 接到继电器输入。

### 四、实验说明

现代自动控制设备中，都存在一个电子电路与电气电路的互相连接问题，一方面要使电子电路的控制信号能够控制电气电路的执行元件（电动机，电磁铁，电灯等），另一方面又要为电子线路的电气电路提供良好的电气隔离，以保护电子电路和人身的安全。继电器便能完成这一桥梁作用。

本实验采用的继电器其控制电压是 5V，控制端为高电平时，继电器工作常开触点吸合，连触点的 LED 灯被点亮。当控制端为低电平时，继电器不工作。执行时，对应的 LED 将随继电器的开关而亮灭。

### 五、实验框图



硬件实验三 用 74HC245 读入数据（51/96/88/ARM）

一、实验要求

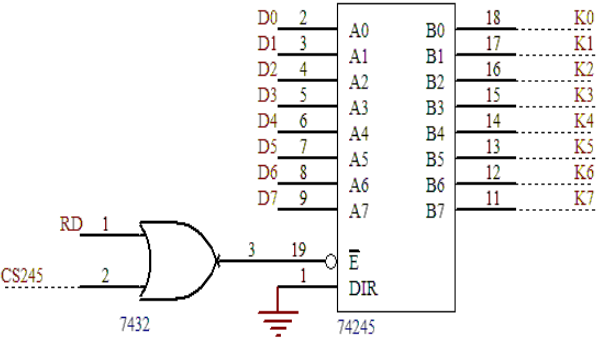
利用板上的 245 输入电路，用总线方式读入开关状态。

二、实验目的

- 1、了解 CPU 常用的端口连接总线的方法。
- 2、掌握 74LS245 进行数据读入或输出。

三、实验电路及连线

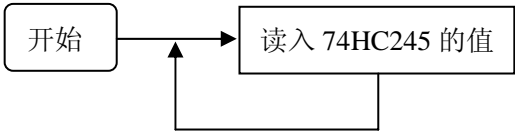
连线	连接孔 1	连接孔 2
1	K0	245-I0
2	K1	245-I1
3	K2	245-I2
4	K3	245-I3
5	K4	245-I4
6	K5	245-I5
7	K6	245-I6
8	K7	245-I7
9	CS0	CS245



四、实验说明

一般情况下，CPU 的总线会挂有很多器件，如何使这些器件不造成冲突，这就要使用一些总线隔离器件，例如 74LS245 就是一种。74LS245 是三态总线收发器，利用它既可以输出也可输入数据。本实验 74LS245 的片选地址为 CS0，即 8000H，读这个地址，就是从 74LS245 读回开关的值。可以用单步的方式执行程序，改变开关状态，观察读回的值。

五、实验程序框图



## 硬件实验四 用 74HC273 输出数据（51/96/88/ARM）

### 一、实验要求

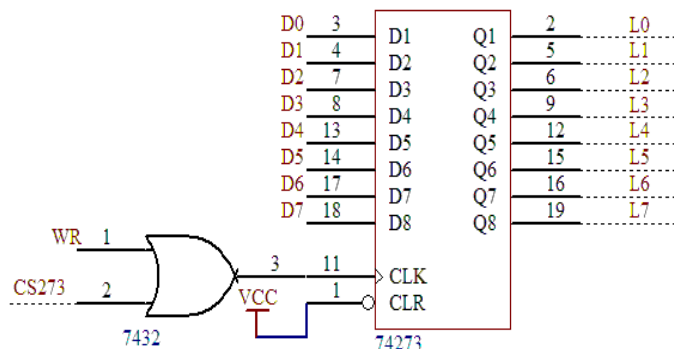
利用板上的 273 输出电路,从总线上输出 I/O 信号,控制八个 LED 灯。

### 二、实验目的

- 1、学习在单片机系统中扩展简单 I/O 接口的方法。
- 2、学习数据输出程序的设计方法。
- 3、了解数据锁存的概念和方法。

### 三、实验电路及连接

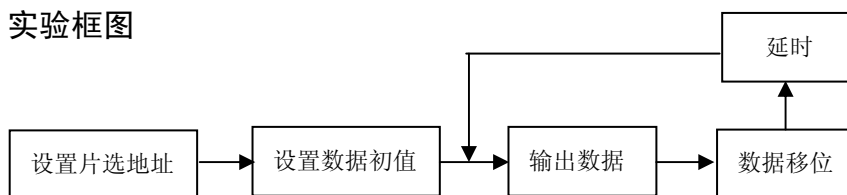
连线	连接孔 1	连接孔 2
1	L0	273-00
2	L1	273-01
3	L2	273-02
4	L3	273-03
5	L4	273-04
6	L5	273-05
7	L6	273-06
8	L7	273-07
9	CS0	CS273



### 四、实验说明

因为本实验是用 74HC273 扩展 I/O 端口。方法是：通过片选信号和写信号将数据总线上的值锁存在 273 中，同时在 273 的输出端品输出，当数据总线上的值撤销以后，由于 273 能够锁存信号，所以 273 的输出端保持不变，直到下次有新的数据被锁存。本实验中，在数据输出同时输出片选信号和写信号。

### 五、实验框图



## 硬件实验五 PWM 转换电压实验（51/96/PIC/ARM）

### 一、实验要求

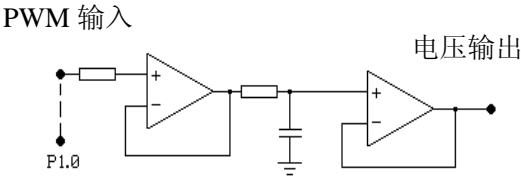
用 P1 端口输出不同占空比的脉冲，通过 PWM 转换电压电路转换成电压。

### 二、实验目的

1. 了解脉宽调制(PWM)的原理
2. 学习用 PWM 输出模拟量

### 三、实验电路及连接

连线	连接孔 1	连接孔 2
1	P1.0	PWM 输入
2	PWM 输出	电压表

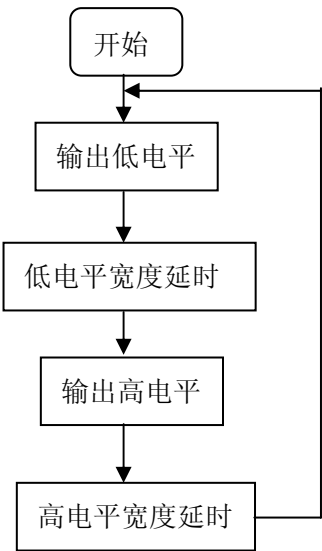


对于 PIC5X MCU，将 PB0 接到 PWM 输入，对于 LPC2103 MCU，如果用 IO 方式输出 PWM 脉冲，将 P0.0 接到 PWM 输入。做 LPC2103 片内 PWM 模块实验时，将 P0.7 接 PWM 输入，程序参见 H5B 文件夹。

### 四、实验说明

PWM 是单片机上常用的模拟量输出方法，通过外接的转换电路，可以将脉冲的占空比变成电压。程序中通过调整占空比来输出模拟电压。占空比就是脉冲中高电平与低电平的宽度比。用万用表测量电压。

### 五、程序框图



## 硬件实验六 音频控制实验（51/96/PIC/ARM）

### 一、实验要求

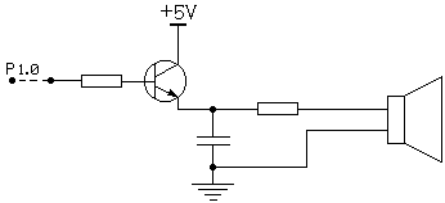
用端口输出不同频率的脉冲，控制喇叭发出不同音调

### 二、实验目的

- 1. 学习输入 / 输出端口控制方法。
- 2. 了解音频发声原理。

### 三、实验线路及连线

连线	连接孔 1	连接孔 2
1	P1.0	喇叭输入

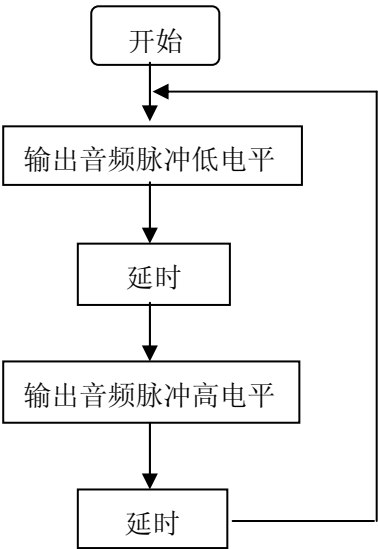


对于 PIC5X MCU，将 PB0 接到喇叭输入，  
对于 LPC2103 MCU，将 P0.0 接到喇叭输入。

### 四、实验说明

端口输出的方波经放大滤波后，驱动扬声器发声。声音的频率由端口输出时延时控制。本实验只给出发出单频率的声音的程序，请同学们思考如何修改程序，可以让扬声器发出不同频率，不同长短的声音。

### 五、程序框图



## 硬件实验七 8255 输入、输出实验（51/96/88/ARM）

### 一、实验要求

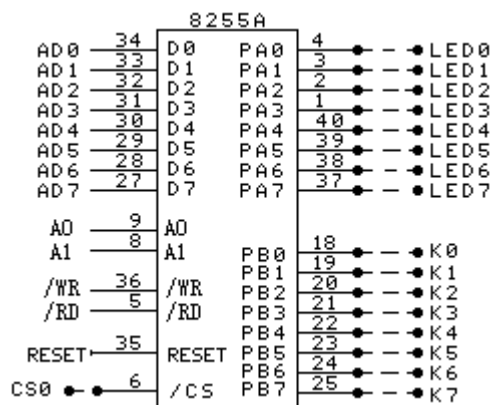
利用 8255 可编程并行口芯片，实现输入/输出实验，实验中用 8255PA 口作输出，PB 口作输入。

### 二、实验目的

- 1、了解 8255 芯片结构及编程方法。
- 2、了解 8255 输入/输出实验方法。

### 三、实验电路及连线

8255 的 CS/接地址译码/CS0，则命令字地址为 8003H，PA 口地址为 8000H，PB 口地址为 8001H，PC 口地址为 8002H。PA0-PA7（PA 口）接 LED0-LED7（LED）PB0-PB7（PB 口）接 K0-K7（开关量）。数据线、读/写控制、地址线、复位信号板上已接好。

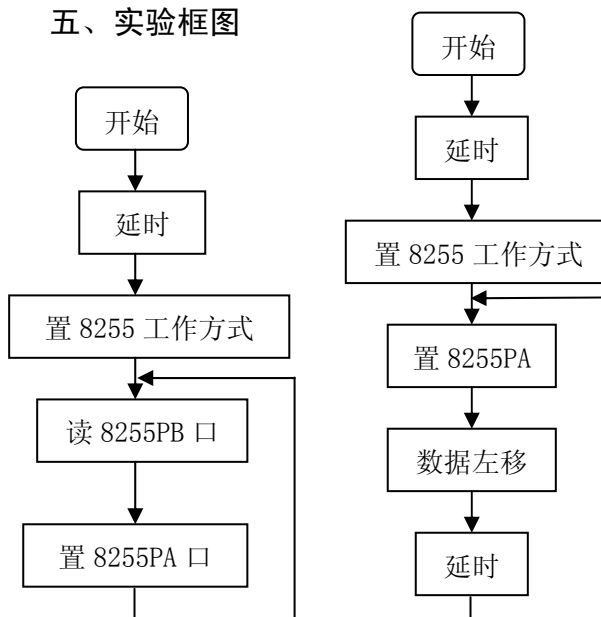


连线	连接孔 1	连接孔 2
1	CS0	8255CS
2	L0	8255-PA0
3	L1	8255-PA1
4	L2	8255-PA2
5	L3	8255-PA3
6	L4	8255-PA4
7	L5	8255-PA5
8	L6	8255-PA6
9	L7	8255-PA7
10	K0	8255-PB0
11	K1	8255-PB1
12	K2	8255-PB2
13	K3	8255-PB3
14	K4	8255-PB4
15	K5	8255-PB5
16	K6	8255-PB6
17	K7	8255-PB7

### 四、实验说明

可编程通用接口芯片 8255A 有三个八位的并行 I/O 口，它有三种工作方式。本实验采用的是方式 0：PA，PC 口输出，PB 口输入。很多 I/O 实验都可以通过 8255 来实现。

### 五、实验框图



## 硬件实验八 串行数转换并行数实验（51/96/PIC/ARM）

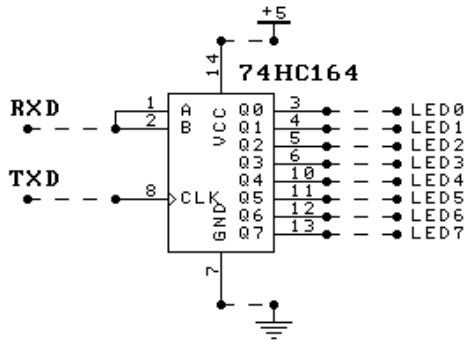
### 一、实验要求

利用单片机的串行口和 I/O 端口串行输出,利用板上的 74HC164 串转并电路,移位转换成并行数据,接在 LED 灯上显示。

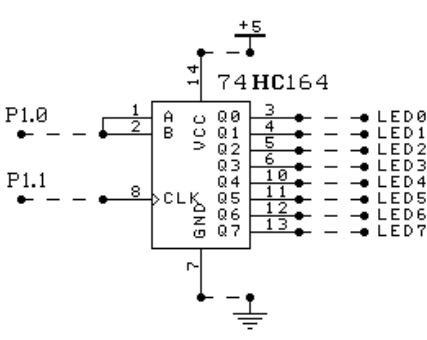
### 二、实验目的

- 1、掌握 MCS51/MCS96 串行口方式 0 工作方式及编程方法。
- 2、掌握用 MCS51/MCS96 的 P1 口的 I/O 功能,输出串行数据。
- 3、掌握利用串行口入 I/O 口,扩展 I/O 通道的方法。

### 三、实验电路及连线



连线	连接孔 1	连接孔 2
1	L0	164-Q0
2	L1	164-Q1
3	L2	164-Q2
4	L3	164-Q3
5	L4	164-Q4
6	L5	164-Q5
7	L6	164-Q6
8	L7	164-Q7
9	RXD	164-AB
11	TXD	164-CLK



连线	连接孔 1	连接孔 2
1	L0	164-Q0
2	L1	164-Q1
3	L2	164-Q2
4	L3	164-Q3
5	L4	164-Q4
6	L5	164-Q5
7	L6	164-Q6
8	L7	164-Q7
9	P1.0	164-AB
11	P1.1	164-CLK

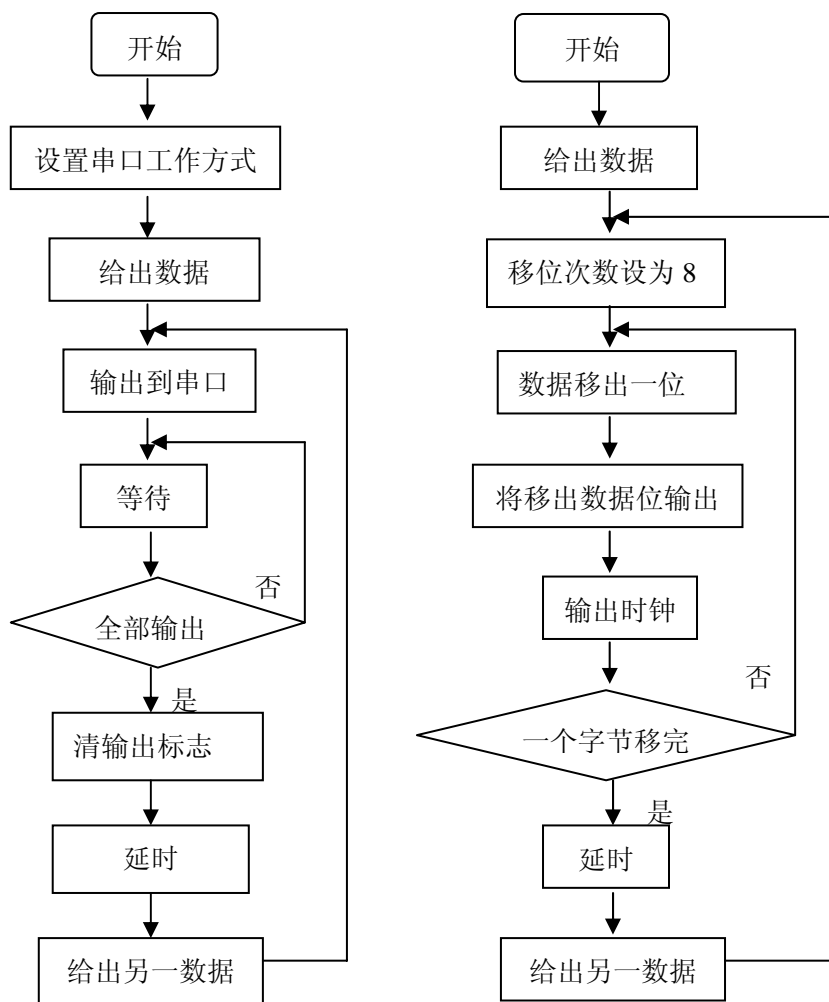
- 1、用串口方式 0, 串行输出数据/时钟      2、用 I/O 口串行输出数据/时钟

对于 PIC5X 将 PA0 接 164-AB, 将 PA1 接 164-CLK; 对于 LPC2103 将 P0.0 接 164-CLK, 将 P0.1 接 164-AB, 用 IO 方式输出数据和时钟信号, 控制移位输出。

#### 四、实验说明

串行口工作在方式 0 时，可通过外接移位寄存器实现串并行转换。在这种方式下，数据为 8 位，只能从 RXD 端输入输出，TXD 端总是输出移位同步时钟信号，其波特率固定为  $F_{osc}/12$ 。对于 80C196CPU 波特率为  $F_{osc}/2(B+1)$ ， $F_{osc}$  为晶振频率，B 为波特率常数。在 CPU 将数据写入 SBUF 寄存器后，立即启动发送。待 8 位数据输完后，硬件将状态寄存器的 TI 位置 1，TI 必须由软件清零。用串行口工作方式 0 输出数据/时钟，是自动移位输出，用 P1 端口串行输出数据时，要编程移位数据，输出数据/时钟。对于 80C196 用串口发数据时要将 IOCl.5 置 1，选择 P2.0 脚为 TXD 功能。同时注意用 P1 口输出位信号时，需要用到‘与’‘或’运算得到相应位。

#### 五、实验框图



(A) 串口方式 0 输出 程序框图

(B) P1 口输出数据/时钟 程序框图



## 硬件实验九 并行数转换串行数实验（51/96/PIC/ARM）

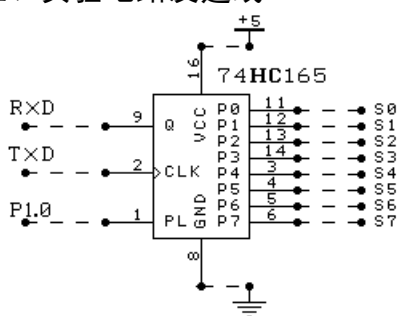
### 一、实验要求

利用板上的 74HC165 并转串电路, 读入外部的并行数据, 移位转换成串行数, 利用单片机串行口和 P1 口串行读入。

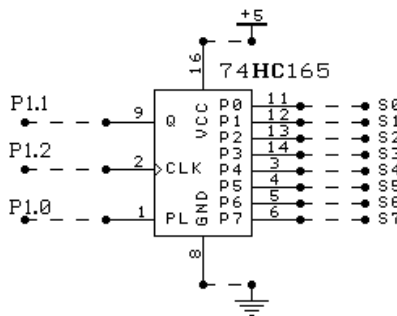
### 二、实验目的

- 1、掌握 MCS51/MCS96 串行口方式 0 工作方式及编程方法。
- 2、掌握用 MCS51/MCS96 的 P1 口的 I/O 功能, 读入串行数据。
- 3、掌握利用串行口及 I/O 口, 扩展 I/O 通道的方法。

### 三、实验电路及连线



连线	连接孔 1	连接孔 2
1	K0	165-P0
2	K1	165-P1
3	K2	165-P2
4	K3	165-P3
5	K4	165-P4
6	K5	165-P5
7	K6	165-P6
8	K7	165-P7
9	RXD	165-Q7
10	TXD	165-CLK
11	P1.0	165-LOAD



连线	连接孔 1	连接孔 2
1	K0	165-P0
2	K1	165-P1
3	K2	165-P2
4	K3	165-P3
5	K4	165-P4
6	K5	165-P5
7	K6	165-P6
8	K7	165-P7
9	P1.1	165-Q7
10	P1.2	165-CLK
11	P1.0	165-LOAD

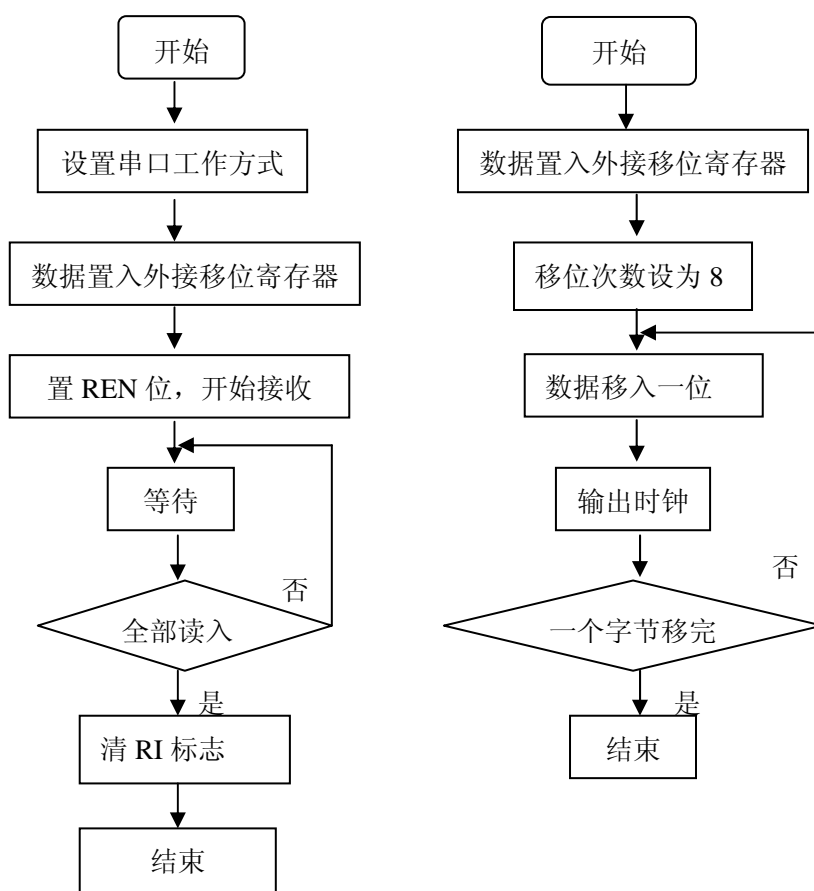
- 1、用串口方式 0, 串行输入数据/时钟    2、用 I/O 口串行输入数据/时钟

对于 PIC5X 将 PB0 接 165-LOAD, 将 PA0 接 165-Q7, 将 PA1 接 164-CLK; 对于 LPC2103 将 P0.0 接 165-CLK, 将 P0.1 接 165-Q7, 将 P0.4 接 165-LOAD, 用 IO 方式输出数据和时钟信号, 控制移位。

#### 四、实验说明

与上个实验一样，这个实验主要是用串并转换方法扩展 I/O 口。串行口工作在方式 0 时，可通过外接移位寄存器实现串并行转换。在这种方式下，数据为 8 位，只能从 RXD 端输入输出，TXD 端总是输出移位同步时钟信号，其波特率固定为晶振频率  $F_{osc}/12$ 。对于 80C196CPU 波特率为  $F_{osc}/2(B+1)$ ， $F_{osc}$  为晶振频率，B 为波特率常数。由软件置位串行控制寄存器的允许接收位 (REN) 后，才启动串行接收。待 8 位数据收完后，硬件将状态寄存器的 RI 位置 1，RI 必须由软件清零。用串行口工作方式 0 读入数据，是自动移位完成的，用 P1 端口串行读入数据时，要编程输出时钟信号，移位读入数据。对于 80C196 用串口发数据时要将 I0C1.5 置 1，选择 P2.0 脚为 TXD 功能。同时注意用 P1 口输出位信号时，需要用到‘与’‘或’运算得到相应位。

#### 五、实验框图



(A) 串口方式 0 读入 程序框图

(B) P1 口读入数据 程序框图

硬件实验十 计数器实验（51/PIC/ARM）

一、实验要求

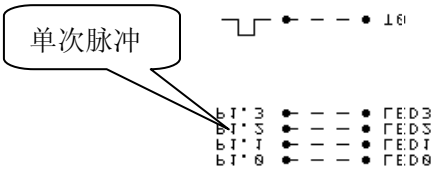
MCS51 内部定时计数器 T0，按计数器模式和方式 1 工作，对 P3.4 (T0) 引脚进行计数。将其数值按二进制数在 P1 口驱动 LED 灯上显示出来。

二、实验目的

- 1、学习单片机内部定时/计数器使用方法。

三、实验电路及连线

连线	连接孔 1	连接孔 2
1	P1.0	L0
2	P1.1	L1
3	P1.2	L2
4	P1.3	L3
5	单脉冲输出	P3.4(T0)

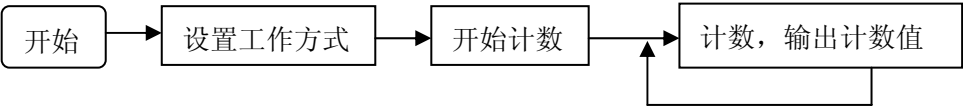


对于 PIC5X MCU, 将单脉冲输出接到 TOCKI 接线柱, 对于 ARM LPC2103 MCU 将单脉冲输出接到 P0.2, 将 P0.8~P0.15 接 L0~L7。

四、实验说明

- 1. 本实验中内部计数器起计数器的作用。外部事件计数脉冲由 P3.4 引入定时器 T0。单片机在每个机器周期采样一次输入波形, 因此单片机至少需要两个机器周期才能检测到一次跳变。这就要求被采样电平至少维持一个完整的机器周期, 以保证电平在变化之前即被采样。同时这就决定了输入波形的频率不能超过机器周期频率。

五、实验框图



## 硬件实验十一 外部中断实验（51/96/ARM）

### 一、实验要求

用单次脉冲申请中断，在中断处理程序中对输出信号进行反转。

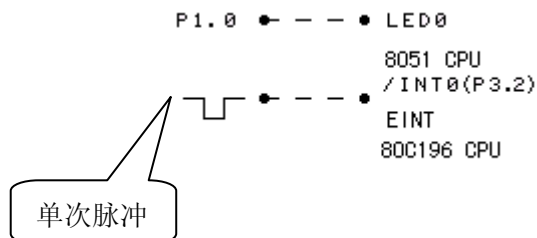
### 二、实验目的

- 1、学习外部中断技术的基本使用方法。
- 2、学习中断处理程序的编程方法。

### 三、实验电路及连线

连线	连接孔 1	连接孔 2
1	P1.0	L0
2	单脉冲输出	P3.2 (MCS51 的 INT0)
2	单脉冲输出	EINT (MCS96 系列)

对于 LPC2103 MCU，将单脉冲（低有效）输出接到 P0.16，将 P0.0 接到发光管 L0。



### 四、实验说明

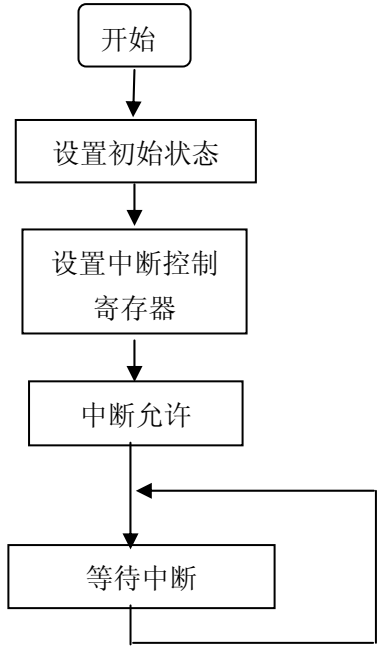
中断服务程序的关键是：

1. 保护进入中断时的状态，并在退出中断之前恢复进入时的状态。
2. 必须在中断程序中设定是否允许中断重入，即设置 EX0 位。
3. 对于 80C196，要选择相应的中断源，并设置中断屏蔽寄存器的相应位。

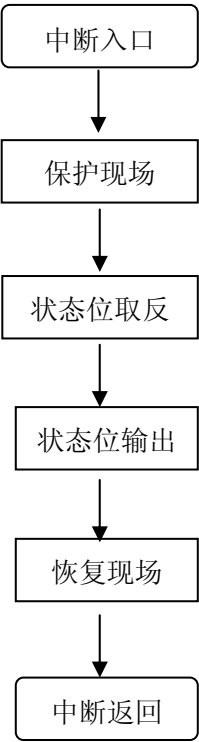
本例中使用了 INT0 中断（80C196 为 EXTINT 中断），一般中断程序进入时应保护 PSW，ACC 以及中断程序使用但非其专用的寄存器。本例的中断程序保护了 PSW，ACC 等三个寄存器并且在退出前恢复了这三个寄存器。另外中断程序中涉及到关键数据的设置时应关中断，即设置时不允许重入。本例中没有涉及这种情况。

INT0（P3.2）端（80C196 为 EINT 端）接单次脉冲发生器。P1.0 接 LED 灯，以查看信号反转。

五、实验框图



主程序框图



外部中断子程序框图

## 硬件实验十二 定时器实验（51/96/PIC/ARM）

### 一、实验要求

用 CPU 内部定时器中断方式计时, 实现每一秒钟输出状态发生一次反转.

### 二、实验目的

- 1、学习单片机内部计数器/定时器的使用和编程方法。
- 2、进一步掌握中断处理程序的编程方法。

### 三、实验电路及连线

连线	连接孔 1	连接孔 2
1	P1.0	L0

P1.0 ● — — ● LED0

对于 PIC5X MCU, 将 PB0 接 L0 发光二极管, 对于 ARM LPC2103 MCU, 将 P0.0 接到 L0 发光二极管。

### 四、实验说明

1、关于内部计数器的编程主要是定时常数的设置和有关控制寄存器的设置。内部计数器在单片机中主要有定时器和计数器两个功能。本实验使用的是定时器。

2、定时器有关的寄存器有工作方式寄存器 TMOD 和控制寄存器 TCON。TMOD 用于设置定时器/计数器的工作方式 0-3, 并确定用于定时还是用于计数。TCON 主要功能是为定时器在溢出时设定标志位, 并控制定时器的运行或停止等。

3、内部计数器用作定时器时, 是对机器周期计数。每个机器周期的长度是 12 个振荡器周期。因为实验系统的晶振是 6MHZ, 本程序工作于方式 2, 即 8 位自动重装方式定时器, 定时器 100us 中断一次, 所以定时常数的设置可按以下方法计算:

机器周期=12÷6MHZ=2uS

(256-定时常数)×2uS=100us

定时常数=206. 然后对 100us 中断次数计数 10000 次, 就是 1 秒钟.

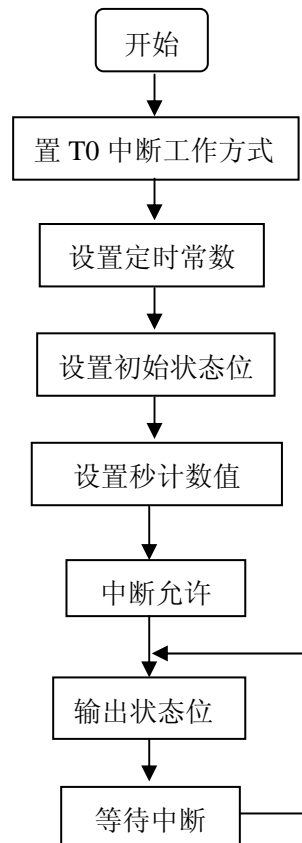
4、在例程的中断服务程序中, 因为中断定时常数的设置对中断程序的运行起到关键作用, 所以在置数前要先关对应的中断, 置数完之后再打开相应的中断。

5、对于 80C196, 与定时器有关的寄存器为 IOC1.2 和 INT-MASK。IOC1.2 为定时器 1 溢出中断允许/禁止位。INT-MASK 的第 0 位为定时器溢出屏蔽位。

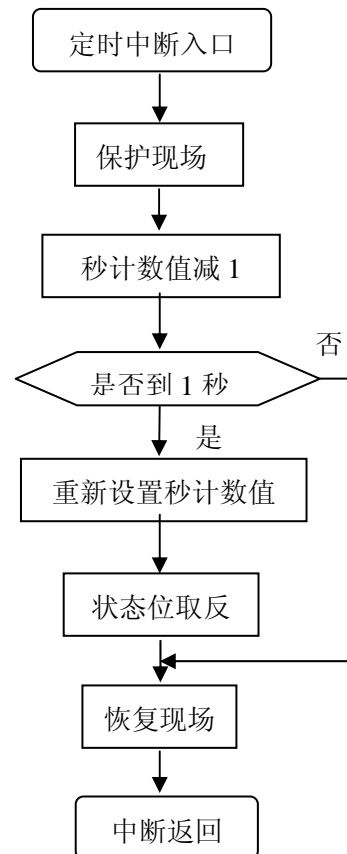
6、对于 80C196, 在设置中断定时常数时, 要注意先设置窗口寄存器 WSR, 设置完常数后, 再恢复原 WSR 值。

7、对于 PIC5X 和 LPC2103 内部定时器/计数器的编程, 参阅相关 MCU 的数据手册。

## 五、实验框图



主程序框图



定时中断子程序框图

硬件实验十三  
D/A 数模转换实验（51/96/88/ARM）

一、实验要求

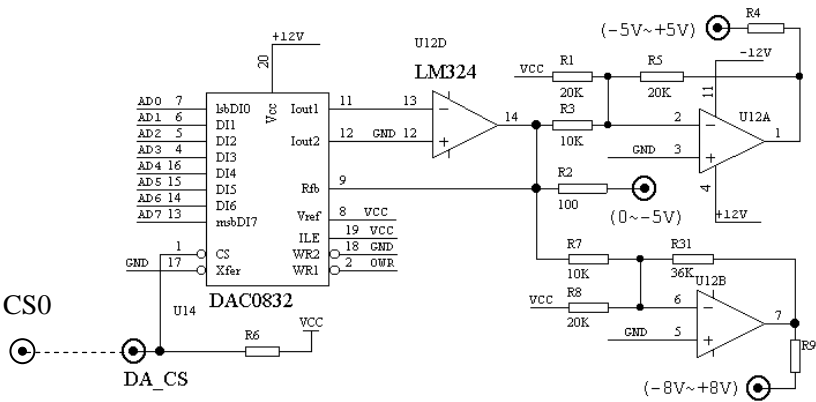
利用 DAC0832，编制程序产生锯齿波、三角波、正弦波。三种波轮流显示，用示波器观看。

二、实验目的

- 1、了解 D/A 转换的基本原理。
- 2、了解 D/A 转换芯片 0832 的性能及编程方法。
- 3、了解单片机系统中扩展 D/A 转换的基本方法。

三、实验电路及连线

连线	连接孔 1	连接孔 2
1	DA_CS	CS2
2	-5V~+5V	电压表



用电压表或示波器探头接-5V~+5V 输出，观察显示电压或波形。

四、实验说明

这是用总线器件做的 ADC 实验，对于 MCS196，LPC2103，芯片已经带有 ADC 转换电路，也可以用片上的 ADC 进行模数转换。

1、D/A 转换是把数字量转换成模拟量的变换，实验台上 D/A 电路输出的是模拟电压信号。要实现实验要求，比较简单的方法是产生三个波形的表格，然后通过查表来实现波形显示。

2、产生锯齿波和三角波的表格只需由数字量的增减来控制，同时要注意三角波要分



段来产生。

要产生正弦波，较简单的方法是造一张正弦数字量表。即查函数表得到的值转换成十六进制数填表。

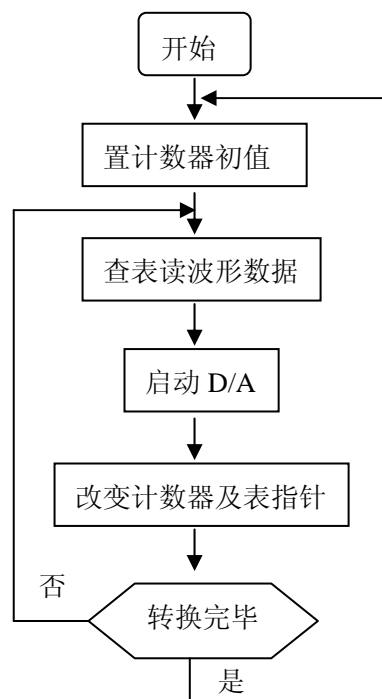
D/A 转换取值范围为一个周期，采样点越多，精度越高些。本例采用的采样点为 256 点/周期。

3、8 位 D/A 转换器的输入数据与输出电压的关系为

$$U(0 \sim -5V) = U_{ref} / 256 \times N$$

$$U(-5V \sim +5V) = 2 \cdot U_{ref} / 256 \times N - 5V \quad (\text{这里 } U_{ref} \text{ 为 } +5V)$$

## 五、实验框图



## 硬件实验十四

### A/D 模数转换实验 (51/96/88/ARM)

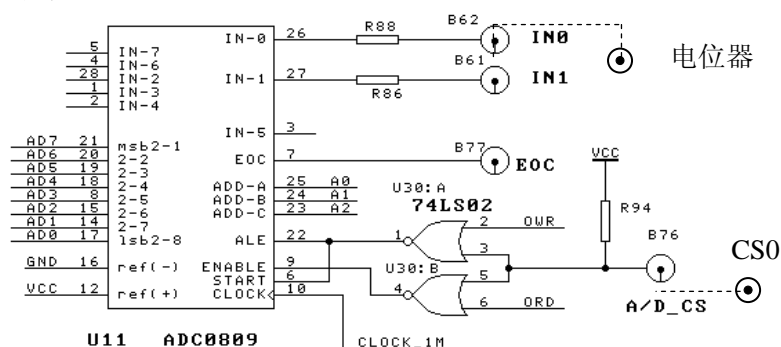
## 一、实验要求

利用实验板上的 ADC0809 做 A/D 转换器，实验板上的电位器提供模拟量输入，编制程序，将模拟量转换成二进制数字量，用 8255 的 PA 口输出到发光二极管显示。

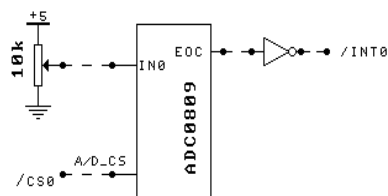
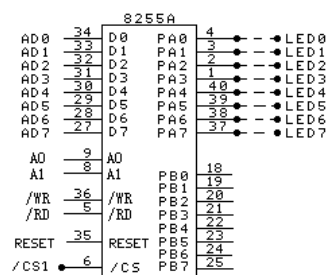
## 二、实验目的

- 1、掌握 A/D 转换与单片机的接口方法。
- 2、了解 A/D 芯片 ADC0809 转换性能及编程。
- 3、通过实验了解单片机如何进行数据采集。

### 三、实验电路及连线



连线	连接孔 1	连接孔 2
1	IN0	电位器输出
2	AD_CS	CS0
3	EOC	INT0
4	8255_CS	CS1
5	PA0	L0
6	PA1	L1
7	PA2	L2
8	PA3	L3
9	PA4	L4
10	PA5	L5
11	PA6	L6
12	PA7	L7



对于 LPC2103 MCU, 如果用模拟总线来控制实验上的 ADC0809 进行 ADC 实验, 接线见上表格 (E0C 不接)。如果用片内的 ADC 模块来进行 ADC 实验, 将 KEY/LED CS 接 CS0, 将

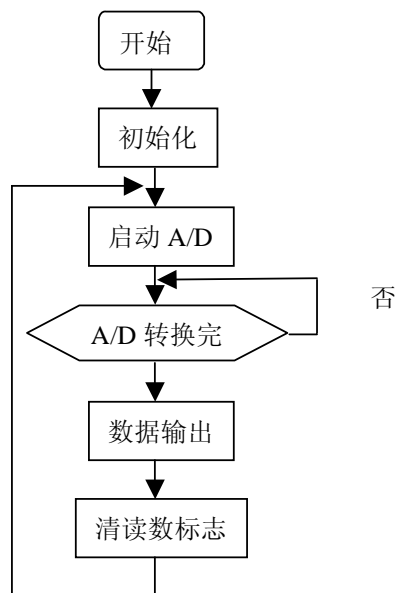
P0.26 (ADC7) 接电位器。实验程序参见 H14B 文件夹。

#### 四、实验说明

A/D 转换器大致有三类：一是双积分 A/D 转换器，优点是精度高，抗干扰性好；价格便宜，但速度慢；二是逐次逼近 A/D 转换器，精度，速度，价格适中；三是并行 A/D 转换器，速度快，价格也昂贵。

实验用的 ADC0809 属第二类，是八位 A/D 转换器。每采集一次一般需 100us。本程序是用延时查询方式读入 A/D 转换结果，也可以用中断方式读入结果，在中断方式下，A/D 转换结束后会自动产生 EOC 信号，将其与 CPU 的外部中断相接，有兴趣的同学可以试试编程用中断方式读回 A/D 结果。

#### 五、实验框图



主程序框图

## 硬件实验十五

### 外部中断(急救车与交通灯) (51/96/ARM)

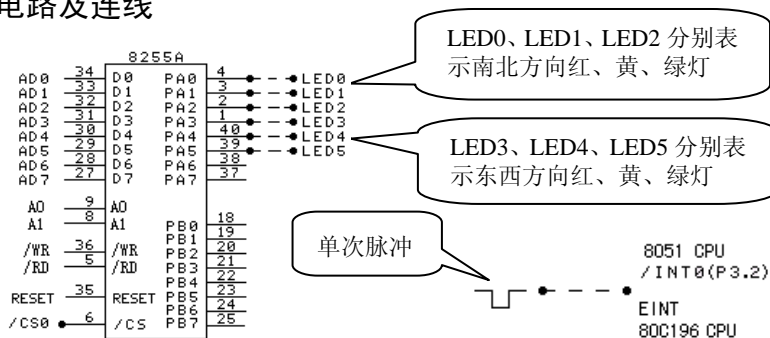
#### 一、实验要求

本实验模拟交通信号灯控制，一般情况下正常显示，有急救车到达时，两个方向交通信号灯全红，以便让急救车通过。设急救车通过路口时间为 10 秒，急救车通过后，交通恢复正常，本实验用单次脉冲申请外部中断，表示有急救车通过。

#### 二、实验目的

- 1、学习外部中断技术的基本使用方法。
- 2、学习中断处理程序的编程方法。

#### 三、实验电路及连线



连线	连接孔 1	连接孔 2
1	8255_CS	CS0
2	PA0	L0
3	PA1	L1
4	PA2	L2
5	PA3	L3
6	PA4	L4
7	PA5	L5
8	单脉冲输出	EINT (96)

连线	连接孔 1	连接孔 2
1	P1.0	L0
2	P1.1	L1
3	P1.2	L2
4	P1.3	L3
5	P1.4	L4
6	P1.5	L5
7	单脉冲输出	INT0 (51)

对于 LPC2103 的实验，将单脉冲(低有效)输出接到 P0.16，P0.8~P0.13 接 L0~L5。

#### 四、实验说明

中断服务程序的关键是：

- 1、保护进入中断时的状态，并在退出中断之前恢复进入时的状态。
- 2、必须在中断程序中设定是否允许中断重入，即设置 EX0 位。

本例中使用了 INT0 中断(80C196 为 EINT 中断)，一般中断程序进入时应保护 PSW, ACC

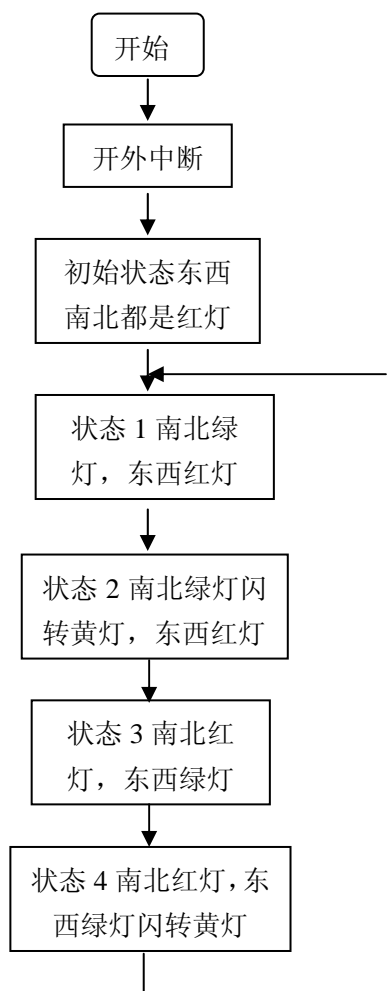
以及中断程序使用但非其专用的寄存器。本例的中断程序保护了 PSW, ACC 等三个寄存器并且在退出前恢复了这三个寄存器。另外中断程序中涉及到关键数据的设置时应关中断, 即设置时不允许重入。本例中没有涉及这种情况。

对于 MCS51CPU 外部中断由 INT0 (P32) 端接入。对于 80C196CPU 外部中断由 EINT 接入。中断信号由单次脉冲发生器产生。

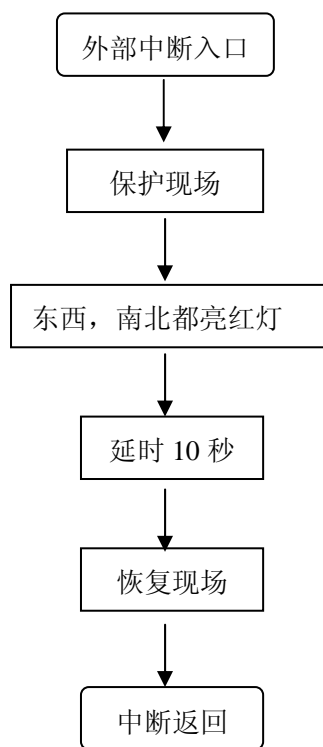
对中断的处理可参见硬件实验十一

本实验提供了用 8255 的 PA 口和用单片机的 I/O 端口控制交通信号灯两种方法, 其中 196 用 8255, 51 用 P1 口, 可以互相参考 51 和 196 的程序。

## 五、实验框图



主程序框图



外部中断子程序框图

硬件实验十六 八段数码管显示（51/96/88/PIC/ARM）

一、实验要求

利用实验仪提供的显示电路, 动态显示一行数据.

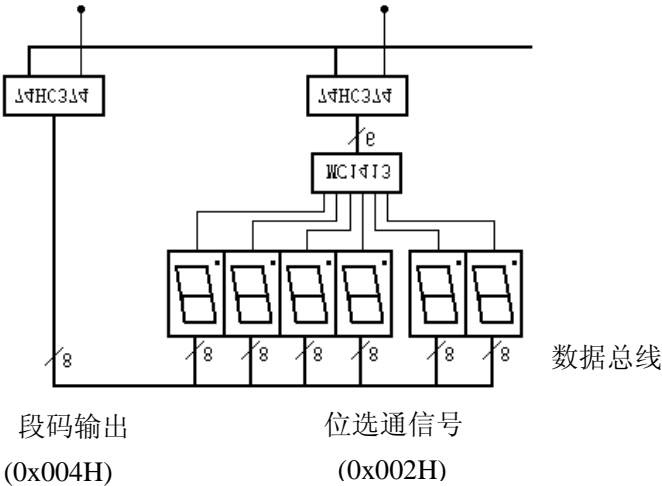
二、实验目的

- 1. 了解数码管动态显示的原理。
- 2. 了解用总线方式控制数码管显示

三、实验线路及连线

连线	连接孔 1	连接孔 2
1	KEY/LED_CS	CS0

当用 PIC5X 驱动八段管时, 是用 I/O 方式驱动, 所以, 驱动方式开关拨到“外驱”方式, PB0~PB7 接八段的 A~H 段, PC0~PC5 接 G0~G5。当用 LPC2103 驱动时, 用的是模拟总线方式, 所以只需将八段管的 KEY/LED\_CS 片选接到 CS0 即可。



四、实验说明

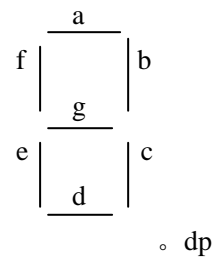
注意: 当用总线方式驱动八段显示管时, 请将八段的驱动方式选择开关拨到“内驱”位置; 当用 I/O 方式驱动八段显示管时, 请将开关拨到“外驱”位置。

本实验仪提供了 6 位 8 段码 LED 显示电路, 学生只要按地址输出相应数据, 就可以实现对显示器的控制。显示共有 6 位, 用动态方式显示。8 位段码、6 位位码是由两片 74LS374 输出。位码经 MC1413 或 ULN2003 倒相驱动后, 选择相应显示位。

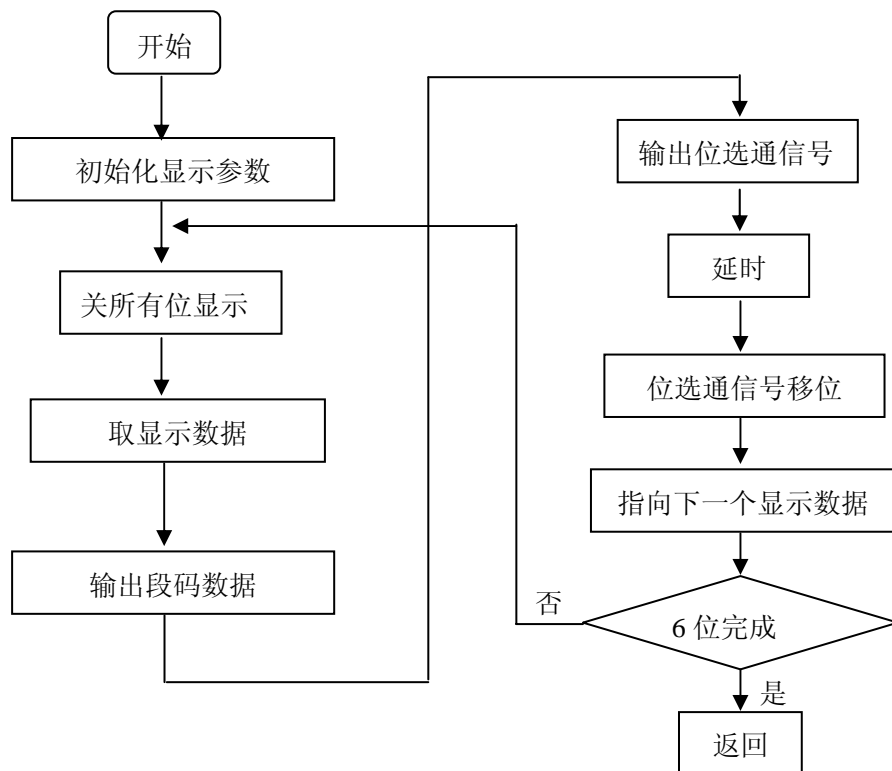
本实验仪中 8 位段码输出地址为 0X004H, 位码输出地址为 0X002H。此处 X 是由 KEY/LED CS 决定, 参见地址译码。做键盘和 LED 实验时, 需将 KEY/LED CS 接到相应的地址译码上。以便使用相应的地址来访问。例如, 将 KEY/LED CS 接到 CS0 上, 则段码地址为 08004H, 位码地址为 08002H。

七段数码管的字型代码表如下表：

显示字形	g	f	e	d	c	b	a	段码
0	0	1	1	1	1	1	1	3fh
1	0	0	0	0	1	1	0	06h
2	1	0	1	1	0	1	1	5bh
3	1	0	0	1	1	1	1	4fh
4	1	1	0	0	1	1	0	66h
5	1	1	0	1	1	0	1	6dh
6	1	1	1	1	1	0	1	7dh
7	0	0	0	0	1	1	1	07h
8	1	1	1	1	1	1	1	7fh
9	1	1	0	1	1	1	1	6fh
A	1	1	1	0	1	1	1	77h
b	1	1	1	1	1	0	0	7ch
C	0	1	1	1	0	0	1	39h
d	1	0	1	1	1	1	0	5eh
E	1	1	1	1	0	0	1	79h
F	1	1	1	0	0	0	1	71h



## 五、程序框图



## 硬件实验十七 键盘扫描显示实验（51/96/88/ARM）

### 一、实验要求

在上一个实验的基础上, 利用实验仪提供的键盘扫描电路和显示电路, 做一个扫描键盘和数码显示实验, 把按键输入的键码在六位数码管上显示出来。

实验程序可分成三个模块。

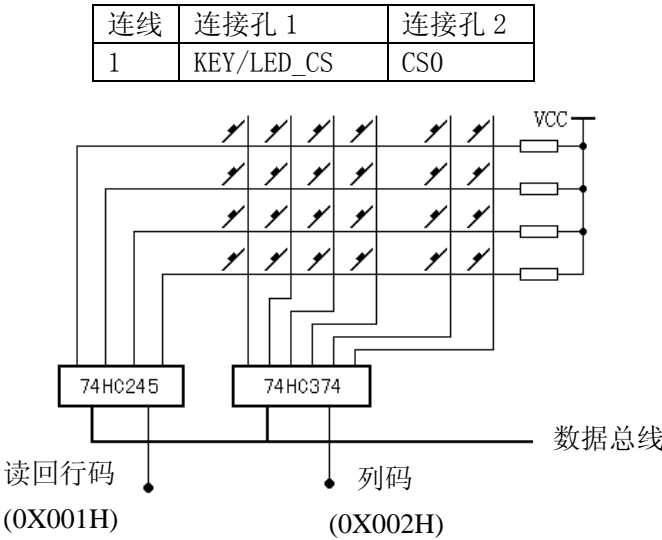
- ①键输入模块：扫描键盘、读取一次键盘并将键值存入键值缓冲单元。
- ②显示模块：将显示单元的内容在显示器上动态显示。
- ③主程序：调用键输入模块和显示模块。

### 二、实验目的

- 1、掌握键盘和显示器的接口方法和编程方法。
- 2、掌握键盘扫描和 LED 八段码显示器的工作原理。

### 三、实验电路及连线

这里只是键盘草图，详细原理参见图 1

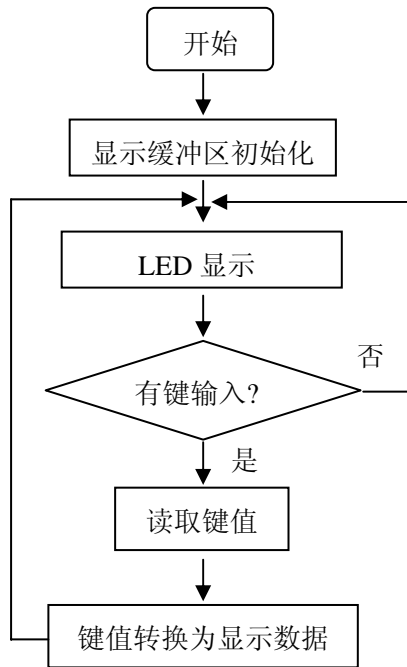


### 三、实验说明

本实验仪提供了一个 6×4 的小键盘，向列扫描码地址 (0X002H) 逐列输出低电平，然后从行码地址 (0X001H) 读回。如果有键按下，则相应行的值应为低，如果无键按下，由于上拉的作用，行码为高。这样就可以通过输出的列码和读取的行码来判断按下的是什么键。在判断有键按下后，要有一定的延时，防止键盘抖动。地址中的 X 是由 KEY/LED CS 决定，参见地址译码。做键盘和 LED 实验时，需将 KEY/LED CS 接到相应的地址译码上。以便使用相应的地址来访问。例如将 KEY/LED CS 信号接 CS0 上，则列扫描地址为 08002H，行码地址为 08001H。列扫描码还可以分时用作 LED 的位选通信号。

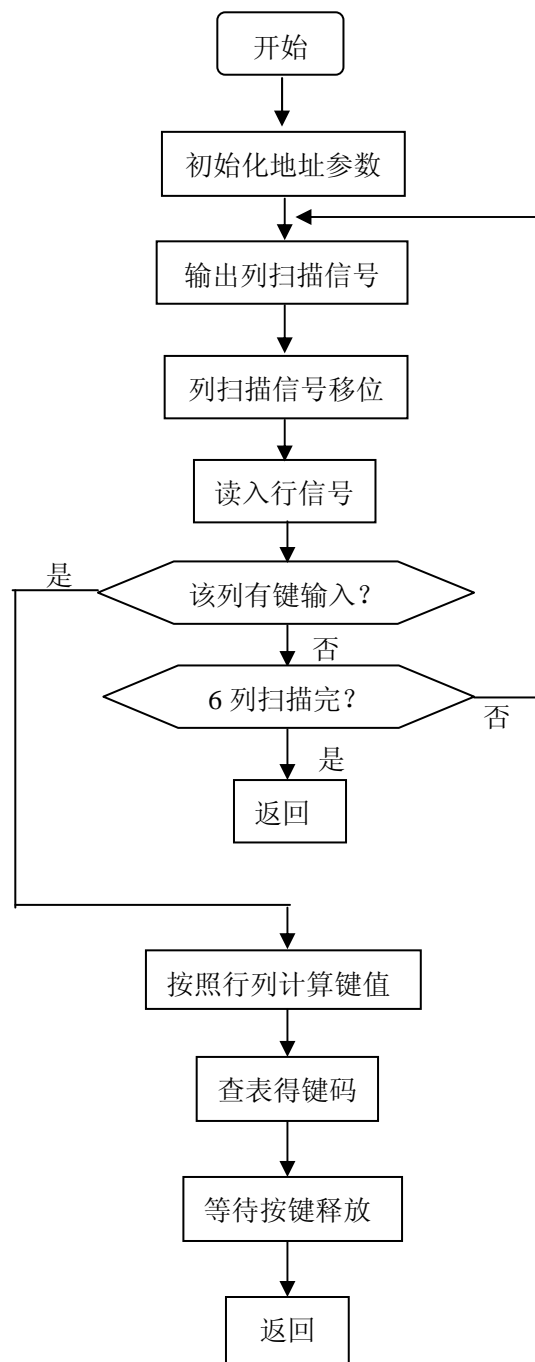


## 五、实验框图



主程序框图

显示程序框图见前个实验



读键输入子程序框图

## 硬件实验十八 电子时钟（51/96/88/ARM）

### 一、实验要求

利用 CPU 的定时器和实验仪上提供的数码显示电路，设计一个电子时钟。格式如下：

XX XX XX 由左向右分别为：时、分、秒

### 二、实验目的

- 1、进一步掌握定时器的使用和编程方法。
- 2、进一步掌握中断处理程序的编程方法。
- 3、进一步掌握数码显示电路的驱动方法。

### 三、实验电路及连线

连线	连接孔 1	连接孔 2
1	KEY/LED_CS	CS0

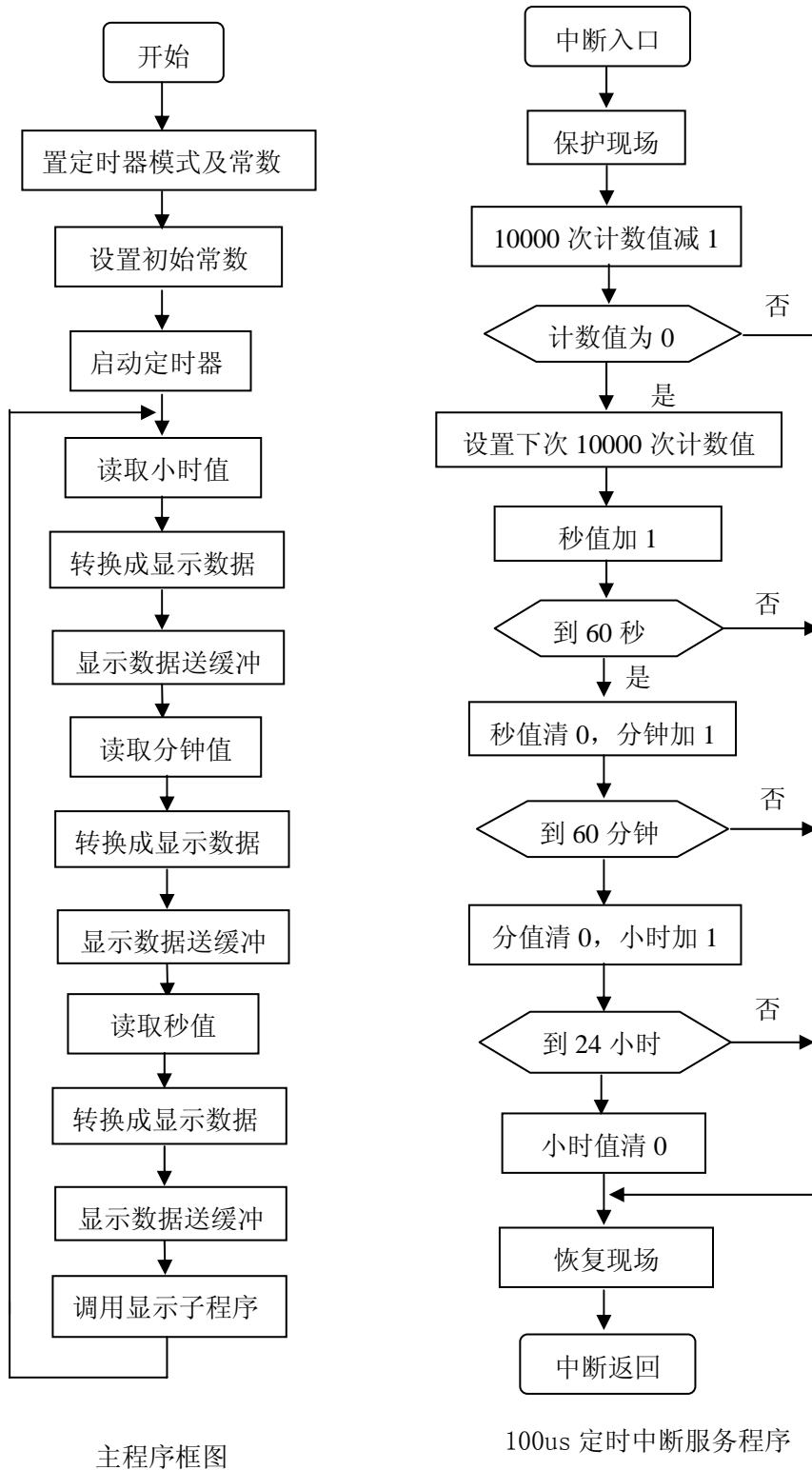
本实验连线只需接上显示/键盘的选择信号即可。显示电路见第 4 页的图 1.

对于 LPC2103 的时钟实验，除了用定时器来实验电子时钟功能，还可以用片上 RTC（实时时钟）模块来进行时钟实验，实验程序参见 H18B 文件夹。

### 四、实验说明

定时器每 100us 中断一次，在中断服务程序中，对中断次数进行计数，100us 计数 10000 次就是 1 秒。然后再对秒计数得到分和小时值，并送入显示缓冲区。显示子程序模块可参照硬件实验十六和硬件实验十七。

## 五、实验框图



## 硬件实验十九

### 单片机串行口通讯实验（51/96/PIC/ARM）

#### 一、实验要求

利用单片机串行口，实现两个实验台之间的串行通讯。其中一个实验台作为发送方，另一侧为接收方。发送方读入按键值，并发送给接收方，接收方收到数据后在 LED 上显示。

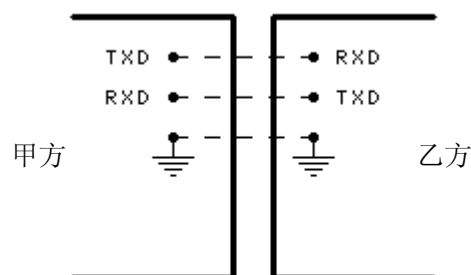
#### 二、实验目的

- 1、掌握单片机串行口工作方式的程序设计，及简易三线式通讯的方法。
- 2、了解实现串行通讯的硬环境、数据格式的协议、数据交换的协议。
- 3、学习串口通讯的中断方式的程序编写方法。

#### 三、实验电路

显示电路和键盘电路见硬件实验十六和硬件实验十七。

连线	连接孔 1	连接孔 2
1	甲方 TXD	乙方 RXD
2	甲方 RXD	乙方 TXD
3	甲方 GND	乙方 GND
4	KEY/LED_CS	CS0



对于 PIC5X MCU，将 PA0 接 RXD，PA1 接 TXD，用 I/O 方式模拟 UART 通信；对于 LPC2103，将 P0.0 接 TXD，P0.1 接 RXD，程序内把该端口设成 UART 功能即可。

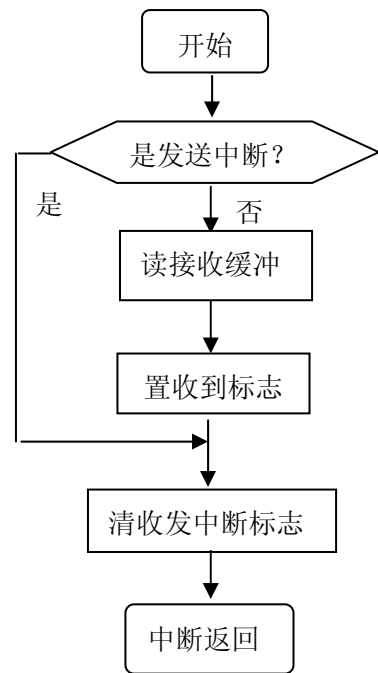
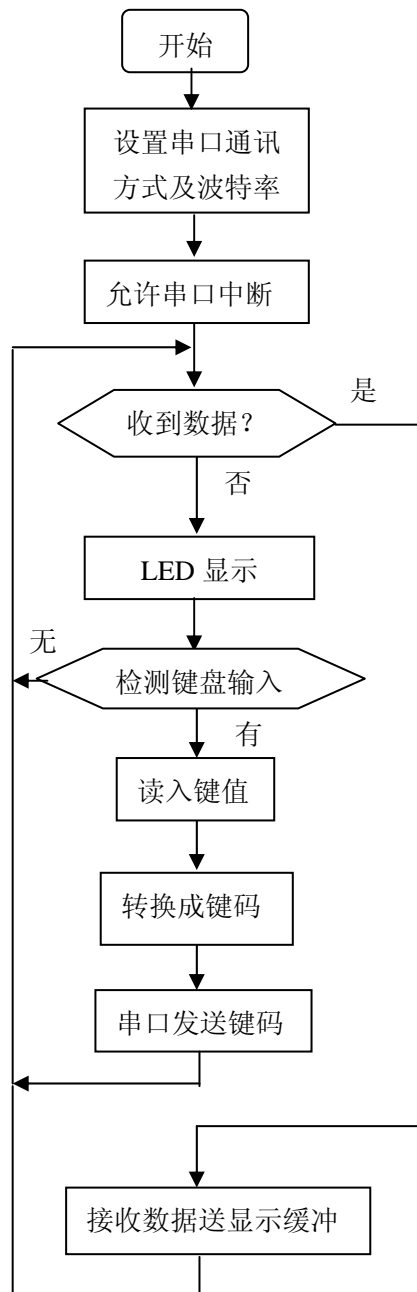
#### 四、实验说明

1、MCS51、80C196 的 RXD、TXD 接线柱在 POD51/96 仿真板上，8088/86 的 TXD、RXD 在 POD8086 仿真板上的 8251 芯片旁边。

2、通讯双方的 RXD、TXD 信号本应经过电平转换后再行交叉连接，本实验中为减少连线可将电平转换电路略去，而将双方的 RXD、TXD 直接交叉连接。也可以将本机的 TXD 接到 RXD 上，这样按下的键，就会在本机 LED 上显示出来。

3、若想与标准的 RS232 设备通信，就要做电平转换，输出时要将 TTL 电平换成 RS232 电平，输入时要将 RS232 电平换成 TTL 电平。可以将仿真板上的 RXD、TXD 信号接到实验板上的“用户串口接线”的相应 RXD 和 TXD 端，经过电平转换，通过“用户串口”接到外部的 RS232 设备。可以用实验仪上的逻辑分析仪采样串口通信的波形

## 五、实验框图



硬件试验二十 1Wire 总线实验（51/96/PIC/ARM）

一、实验要求

利用实验仪上的 1Wire 总线器件 DS18B20，编写 1Wire 总线读写程序并自行验证程序的正确性。

二、实验目的

- 1. 了解 1Wire 总线基本原理
- 1. 掌握典型 1Wire 总线器件读写操作

三、实验电路连线框图

MCS51/96	连线	连接孔 1	连接孔 2
	1	DQ	P1.0
	2	KEY/LED CS	CS0
PIC57	连线	连接孔 1	连接孔 2
	1	DQ	PORB0
	2		
ARM LPC2103	连线	连接孔 1	连接孔 2
	1	DQ	P0.7
	2	KEY/LED CS	CS0

四、实验说明

1Wire 单总线是Maxim全资子公司Dallas的一项专有技术。与目前多数标准串行数据通信方式不同，它采用单根信号线，既传输时钟又传输数据，而且数据传输是双向的。它具有节省I/O 口

线资源、结构简单、成本低廉、便于总线扩展和维护等诸多优点。

1Wire 单总线适用于单个主机系统，能够控制一个或多个从机设备。当只有一个从

机位于总线上时，系统可按照单节点系统操作，而当多个从机位于总线上时，则系统

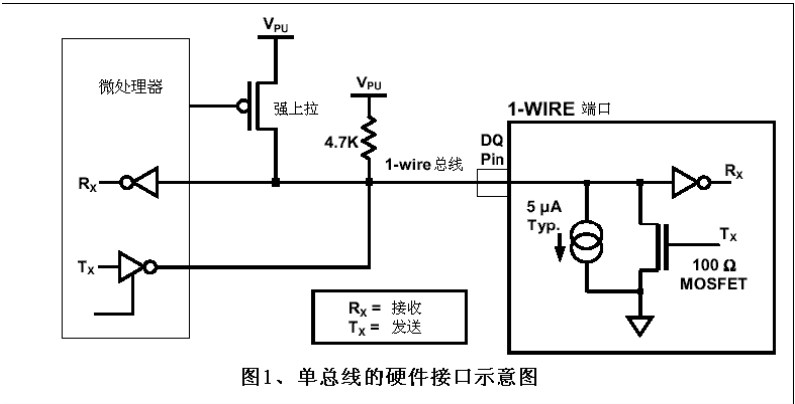


图1、单总线的硬件接口示意图

按照多节点系统操作。

顾名思义，单总线只有一根数据线。设备（主机或从机）通过一个漏极开路或三态端口连接至该数据线，这样允许设备在不发送数据时释放数据总线，以便总线被其它设备使用。单总线端口为漏极开路，其内部等效电路如图所示。

单总线要求外接一个约5k的上拉电阻，这样单总线的闲置状态为高电平，不管什么原因，如果传输过程需要暂时挂起，且要求传输过程还能够继续的话，则总线必须处于空闲状态。位传输之间的恢复时间没有限制，只要总线在恢复期间处于空闲状态（高电平）。如果总线保持低电平超过480us，总线上的所有器件将被复位。另外，在寄生方式供电时，为了保证单总线器件在某些工作状态下（如温度转换期间、EEPROM 写入等）具有足够的电源电流，必须在总线上提供强上拉（如图所示的MOSFET）。

典型的单总线命令序列如下：第一步初始化、第二步ROM命令（跟随需要交换的数据）、第三步功能命令（跟随需要交换的数据）。每次访问单总线器件，必须严格遵守这个命令序列。如果出现序列混乱，则单总线器件不会响应主机。但是，这个准则对于搜索ROM命令和报警搜索命令例外，在执行两者中任何一条命令之后，主机不能执行其后的功能命令，必须返回至第一步。

所有的单总线器件要求采用严格的通信协议，以保证数据的完整性。该协议定义了几种信号类型：复位脉冲、应答脉冲、写0、写1、读0、读1。所有这些信号，除了应答脉冲以外，都由主机发出同步信号。并且发送所有的命令和数据都是字节的低位在前，这一点与多数串行通信格式不同（多数为字节的高位在前）。

本实验提供一个单结点访问D18B20的程序样例，关于1Wire更详细的说明及通讯协议可以参照DALLAS相关技术资料及1Wire器件资料。

## 硬件试验二十一 直流电机控制实验（51/96/88/ARM）

### 一、实验要求

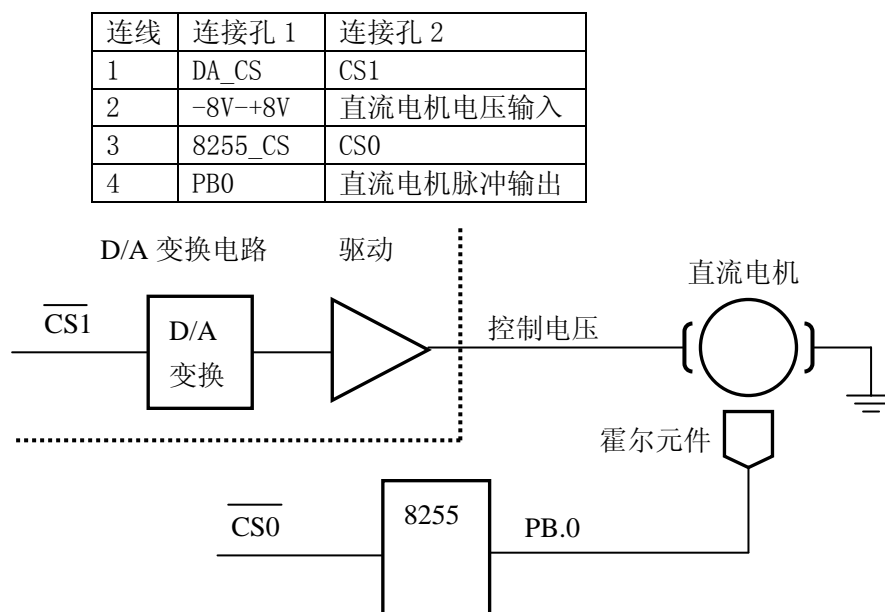
利用实验仪上的 D/A 变换电路，输出-8V 至+8V 电压，控制直流电机。改变输出电压值，改变电机转速，用 8255 的 PB.0 读回脉冲计数，计算电机转速。

### 二、实验目的

1. 了解直流电机控制原理。
2. 学习单片机控制直流电机的编程方法。
3. 了解单片机控制外部设备的常用电路。

### 三、实验电路连线框图

D/A 变换电路可参见 D/A 变换实验。8255 电路可参见 8255 实验。



### 四、实验说明

在电压允许范围内，直流电机的转速随着电压的升高而加快，若加上的电压为负电压，则电机反向旋转。本实验仪的 D/A 变换可输出-8V 到+8V 的电压，将电压经驱动后加在直流电机上，使其运转。通过单片机输出数据到 D/A 变换电路，控制电压的高低和正负，观察电机的旋转情况。

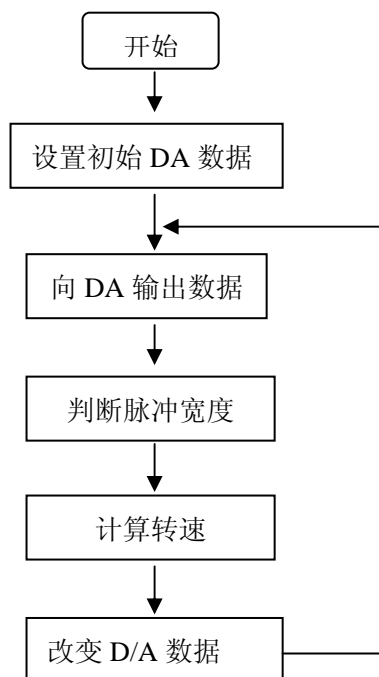
在电机转盘上安装一个小磁芯，用霍尔元件感应电机转速，用单片机控制 8255 读



回感应脉冲，从而测算出电机的转速。

有兴趣的同学，可以做一个恒速的试验，即让电机转速保持一定。若电机转速偏低，则提高输出电压，若电机转速偏高，则降低输出电压。首先给电机一定的阻力，让转速保持一定，然后稍微给加大阻力，观察 D/A 输出的电压是否能做出反应，再减小阻力，也观察 D/A 电压，有何变化。注意所加的阻力不能过大，以免电机烧毁。

## 五、 实验程序框图



硬件试验二十二  
步进电机控制实验（51/96/88/PIC/ARM）

一、实验要求

用 8255 扩展端口控制步进电机，编写程序输出脉冲序列到 8255 的 PA 口，控制步进电机正转、反转，加速，减速。

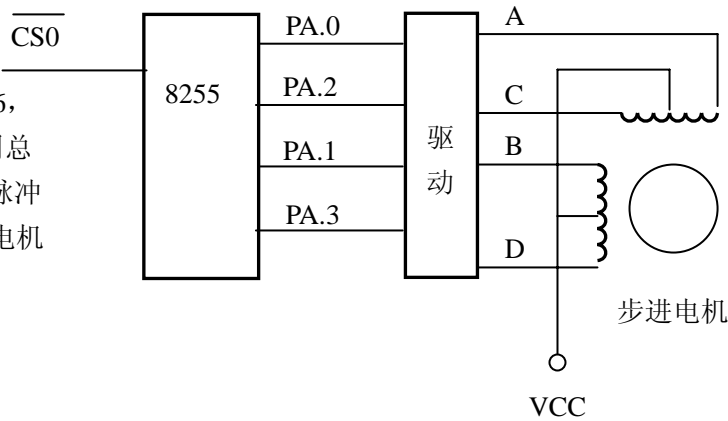
二、实验目的

- 1. 了解步进电机控制的基本原理。
- 2. 掌握控制步进电机转动的编程方法。
- 3. 了解单片机控制外部设备的常用电路。

三、实验电路连线框图

8255 控制的原理图，  
参见 8255 实验。

此实验中，MCS51，MCS96，  
i8088/86，ARM LPC2103 用总  
控制 8255 输出步进电机的脉冲  
PIC5X 用 IO 方式输出步进电机  
的控制脉冲。



用总线方式控制 8255 输出脉冲

连线	连接孔 1	连接孔 2
1	8255_CS	CS0
2	PA0	步进电机 A 项
3	PA1	步进电机 B 项
4	PA2	步进电机 C 项
5	PA3	步进电机 D 项

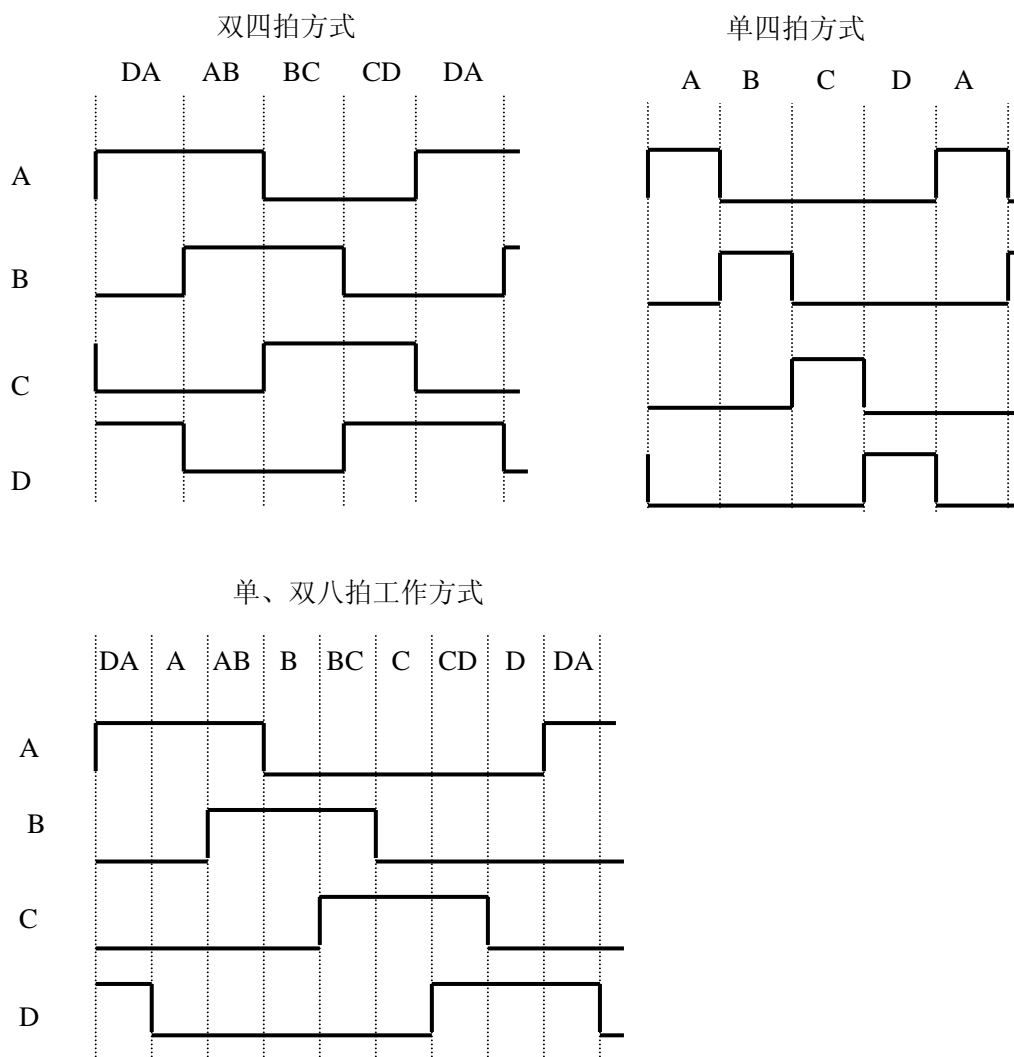
PIC5X 用 IO 控制方式输出脉冲

连线	连接孔 1	连接孔 2
1	PIC5X 的 PA0	步进电机 A 项
2	PIC5X 的 PA1	步进电机 B 项
3	PIC5X 的 PA2	步进电机 C 项
4	PIC5X 的 PA3	步进电机 D 项

四、实验说明

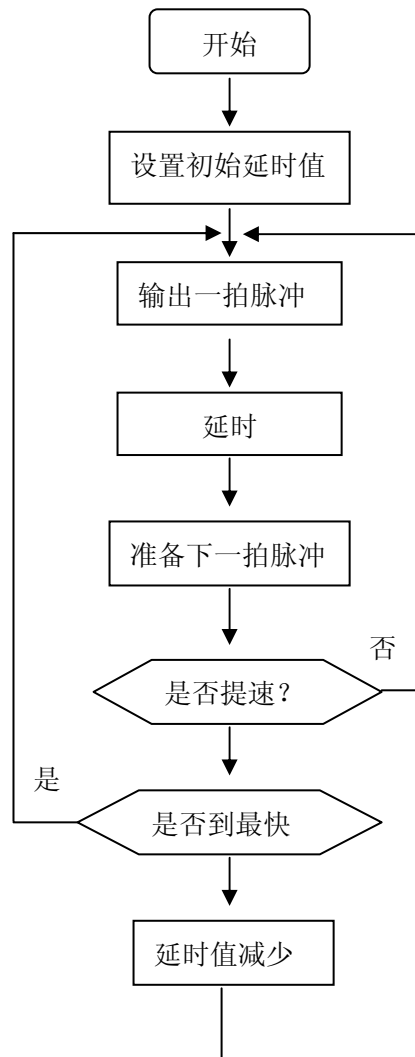
步进电机驱动原理是通过每相线圈中的电流的顺序切换来使电机作步进式旋转。切换是通过单片机输出脉冲信号来实现的。所以调节脉冲信号的频率便可以改变步进电机的转速，改变各相脉冲的先后顺序，可以改变电机的旋转方向。步进电机的转速应由慢到快逐步加速。

电机驱动方式可以采用双四拍( $AB \rightarrow BC \rightarrow CD \rightarrow DA \rightarrow AB$ )方式,也可以采用单四拍( $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$ )方式,或单、双八拍( $A \rightarrow AB \rightarrow B \rightarrow BC \rightarrow C \rightarrow CD \rightarrow D \rightarrow DA \rightarrow A$ )方式。各种工作方式的时序图如下:(高电平有效)



上图中示意的脉冲信号是高有效,但实际控制时公共端是接在 VCC 上的,所以实际控制脉冲是低有效。8255 的 PA 口输出的脉冲信号经 (MC1413 或 ULN2003A) 倒相驱动后,向步进电机输出脉冲信号序列。MCS51 或 80C196 单片机也可以通过 P1 口输出脉冲信号控制步进电机的运转。也可以通过实验仪上的波形发生器来产生脉冲序列来控制步进电机的运转,参见“第五章 逻辑分析工具”的波形发生器部分。

## 五、实验程序框图



硬件试验二十三  
温度传感器实验（51/96/88/ARM）

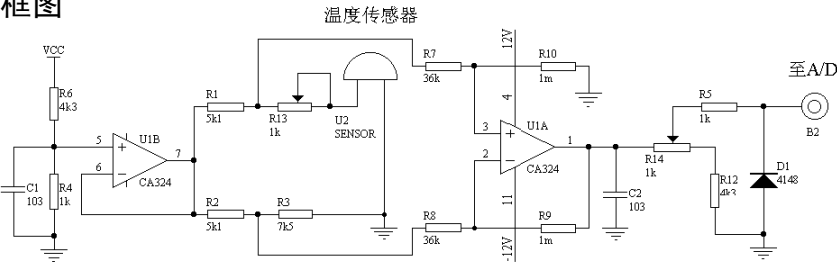
一、实验要求

利用试验板上提供的温度传感器电路，完成温度的采样和显示。

二、实验目的

- 1. 了解温度传感器电路的工作原理。
- 2. 了解弱信号传感器的常见电路。
- 3. 进一步熟悉 A/D 变换电路的工作原理。

三、实验接线框图



连线	连接孔 1	连接孔 2
1	KEY/LED_CS	CS0
2	模数转换 AD_CS	CS2
3	温度传感输出	模数转换器 IN0

四、实验说明

上图是一个较常用的温度测量电路，大致分成电源，电阻电桥，运放，输出四部分。电源由 R4、R6、C1、U1B 组成，R4、R6 为分压电路，C1 主要滤除 VCC 中纹波，U1B 为 LM324 运算放大器，工作于电压跟随器方式，其特点是具有高输入阻抗低输出阻抗，为后级电桥提供较稳定的电流。电桥由 R1、R2、R3、R13 及热敏电阻组成，通过调节 R13 使电桥平衡，当温度发生变化时，热敏电阻阻值变化，电桥产生电压差。运放电路由 R7、R8、R9、R10 及 U1A 组成，这是一种灵敏度较高的电桥放大电路，放大倍数由 R9/R8 得到。输出电路由 R4、R12、R14、D1 组成，调节 R14 可以调整输出电压幅度。D1 主要用于防止输出负电压，保护后级 A/D 电路。

其它电阻类传感器的工作原理与此相似，也可分成以上几部分，具体电路可能有所不同，但原理是相同的。

五、实验程序框图

本实验主要介绍温度传感器的电路，具体程序可参见 A/D 变换实验和显示实验。

# 硬件实验二十四 液晶显示控制实验 (51/96/88/ARM)

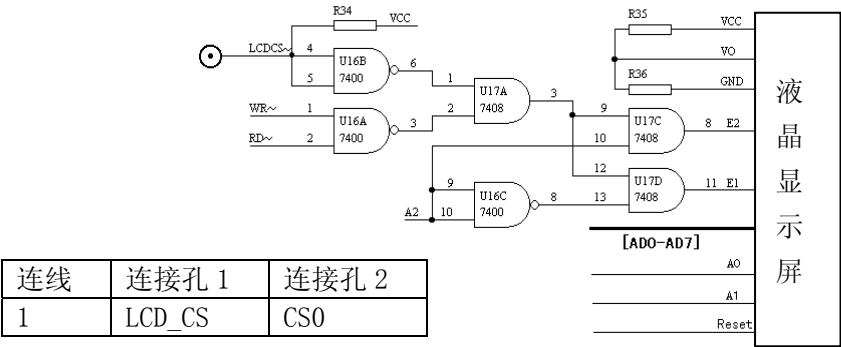
## 一、实验要求

利用实验上的液晶显示屏电路，编写程序控制显示，输出汉字。

## 二、实验目的

1. 了解液晶显示屏的控制原理及方法。
2. 了解点阵汉字的显示原理。

## 三、实验电路及连接



## 四、实验说明

本实验仪采用的液晶显示屏内置控制器为 SED1520，点阵为 122x32, 需要两片 SED1520 组成，由 E1、E2 分别选通，以控制显示屏的左右两半屏。图形液晶显示模块有两种连接方式。一种为直接访问方式，一种为间接控制方式。本实验仪采用直接控制方式。

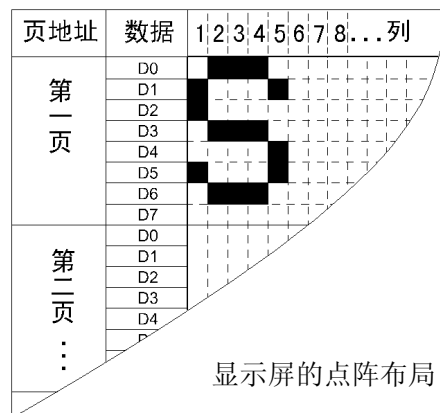
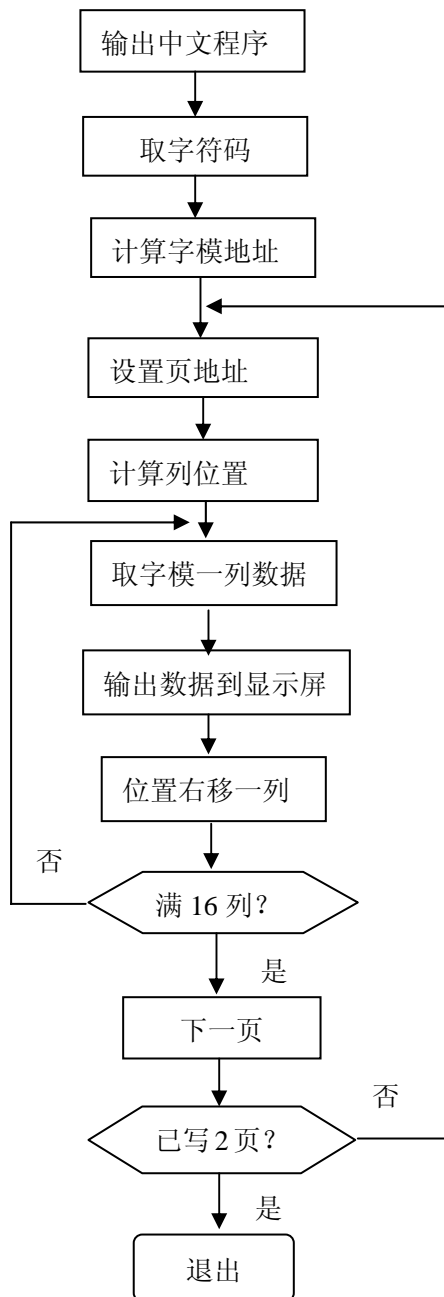
直接控制方式就是将液晶显示模块的接口作为存储器或 I / O 设备直接挂在计算机总线上。计算机通过地址译码控制 E1 和 E2 的选通；读 / 写操作信号 R / W 由地址线 A1 控制；命令/数据寄存器选择信号 A0 由地址线 A0 控制。实际电路如上图所示。地址映射如下（地址中的 X 由 LCD CS 决定，可参见地址译码部分说明）

0X000H	0X001H	0X002H	0X003H	0X004H	0X005	0X006H	0X007H
写 E1 指令	写 E1 数据	读 E1 状态	读 E1 数据	写 E2 指令	写 E2 数据	读 E2 状态	读 E2 数据

间接控制方式是计算机通过自身的或系统的并行接口与液晶显示模块连接，如 MCS51 的 P1 口和 P3 口， 8255 等并行接口芯片。计算机通过对该并行接口输出状态的编程操作，完成对液晶显示模块所需时序的操作和数据的传输。这种间接控制方式的电路简单，控制时序通过编程来实现。

有关图形液晶显示屏的命令和详细原理，可参见 SED1520 的芯片资料。

## 五、实验框图



显示屏的点阵布局

图示的是一个 5x7 的英文字符，用 6 列 x 1 页来显示。一个 16x16 点阵的汉字，可以看成是 16 列 x 2 页的数据。那么同理，一个 24x24 点阵的汉字，可以用 24 列 x 3 页来显示。

## 硬件实验二十五 电子琴实验（51/96/88/ARM）

### 一、实验要求

利用实验仪上提供的键盘，使数字键 1、2、3、4、5、6、7 作为电子琴按键，按下即发出相应的音调。用 8255 的 PA.0 口发出音频脉冲，驱动喇叭。

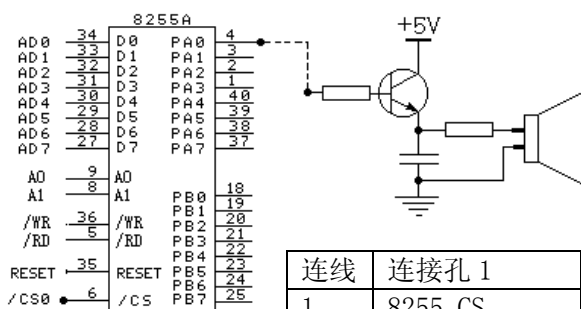
### 二、实验目的

1. 了解计算机发声原理。
2. 进一步熟悉定时器编程方法。
3. 进一步熟悉键盘扫描电路工作原理及编程方法

### 三、实验电路及连接

喇叭发声电路与硬件实验六相同，

键盘电路无需连线，原理图可参考图 1 和硬件实验十七



连线	连接孔 1	连接孔 2
1	8255_CS	CS0
2	KEY/LED_CS	CS1
3	PA0	喇叭脉冲输入

### 四、实验说明

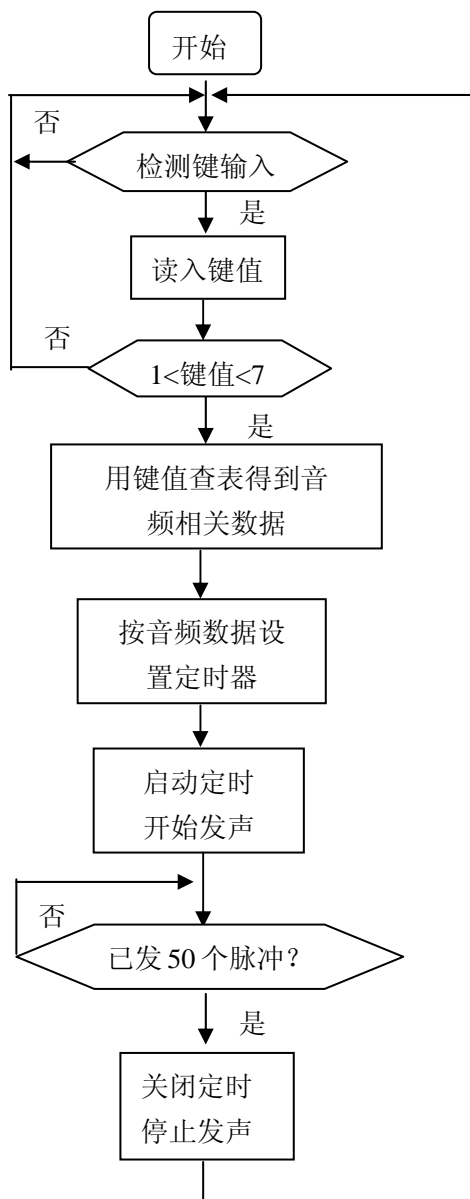
1. 利用定时器，可以发出不同频率的脉冲，不同频率的脉冲经喇叭驱动电路放大滤波后，就会发出不同的音调。
2. 定时器按设置的定时参数产生中断，这一次中断发出脉冲低电平，下一次反转发出脉冲高电平。由于定时参数不同，就发出了不同频率的脉冲。本实验中按键一次，会发 50 个脉冲。发完后继续检测键盘，如果键还按下，继续发音。

各音阶标称频率值：

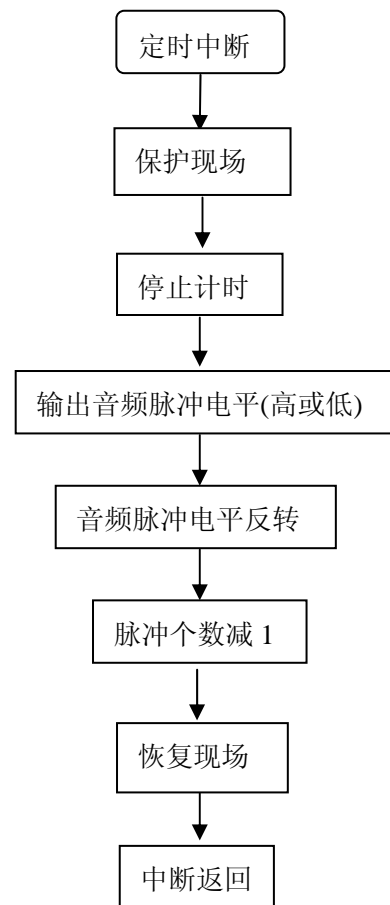
音 阶	1	2	3	4	5	6	7
频率 (HZ)	440.00	493.88	554.37	587.33	659.26	739.99	830.61



## 五、实验框图



主程序框图



定时中断程序框图

## 硬件实验二十六 空调温度控制实验 (51/96/88/ARM)

## 六、实验要求

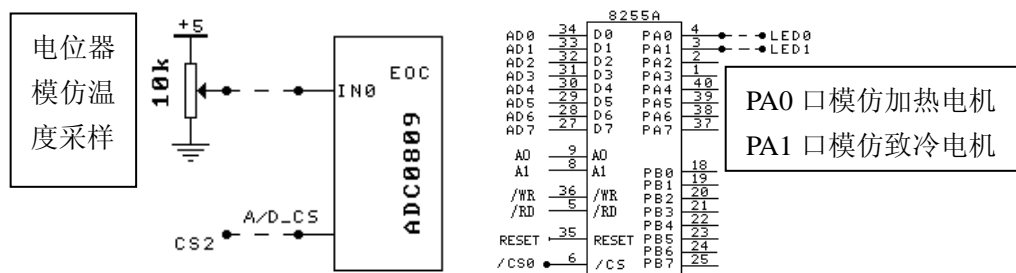
利用实验仪上显示电路，键盘电路，A/D 变换电路，完成类似空调恒温控制实验，可以利用实验仪上的电位器模仿温度变化，加热和致冷电机可以用发光管代替。要求可以用键盘设定恒温温度，当外界温度超过设定温度 $\pm 2^{\circ}\text{C}$ 时，就要启动加热或致冷电机。

## 二、实验目的

1. 了解闭环控制的基本原理。
2. 进一步熟悉 A/D 变换原理和编程方法。
3. 进一步了键盘扫描和 LED 显示原理和编程方法。

### 三、实验电路及连接

LED 显示电路和键盘电路实验仪上已接好。原理图见图 1。A/D 变换电路只要接上模拟量输入和地址选择信号即可。

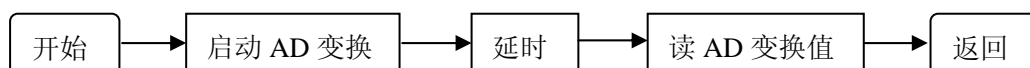


#### 四、实验说明

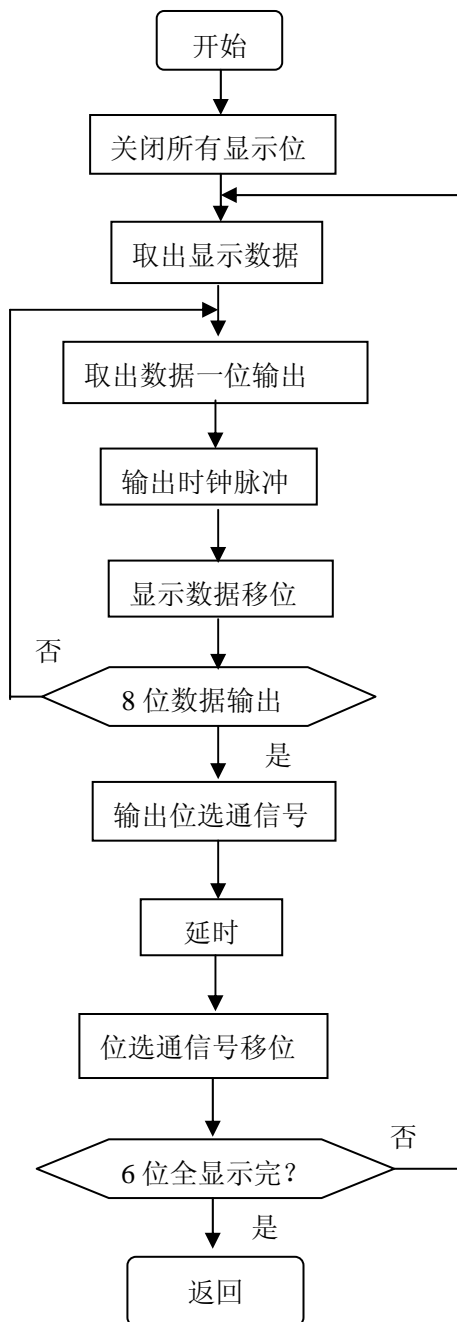
这是一个综合硬件实验,其中各部分实验已单独做过,现联合起来形成一个控制系统。其中 LED 显示实验和键盘扫描实验可参见硬件实验十六、十七。A/D 变换实验可参见硬件实验十三。I/O 口输入输出可参见 8255 硬件实验。

连线	连接孔 1	连接孔 2
1	AD_CS	CS2
2	IN0	电位器输出
3	8255_CS	CS0
4	PA0	L0
5	PA1	L1
6	KEY/LED_CS	CS1

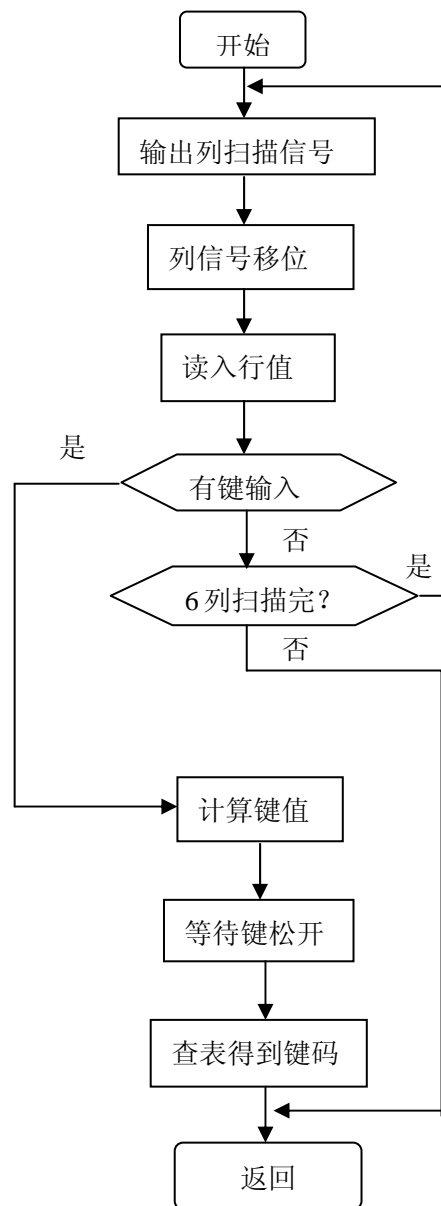
## 五、实验程序框图



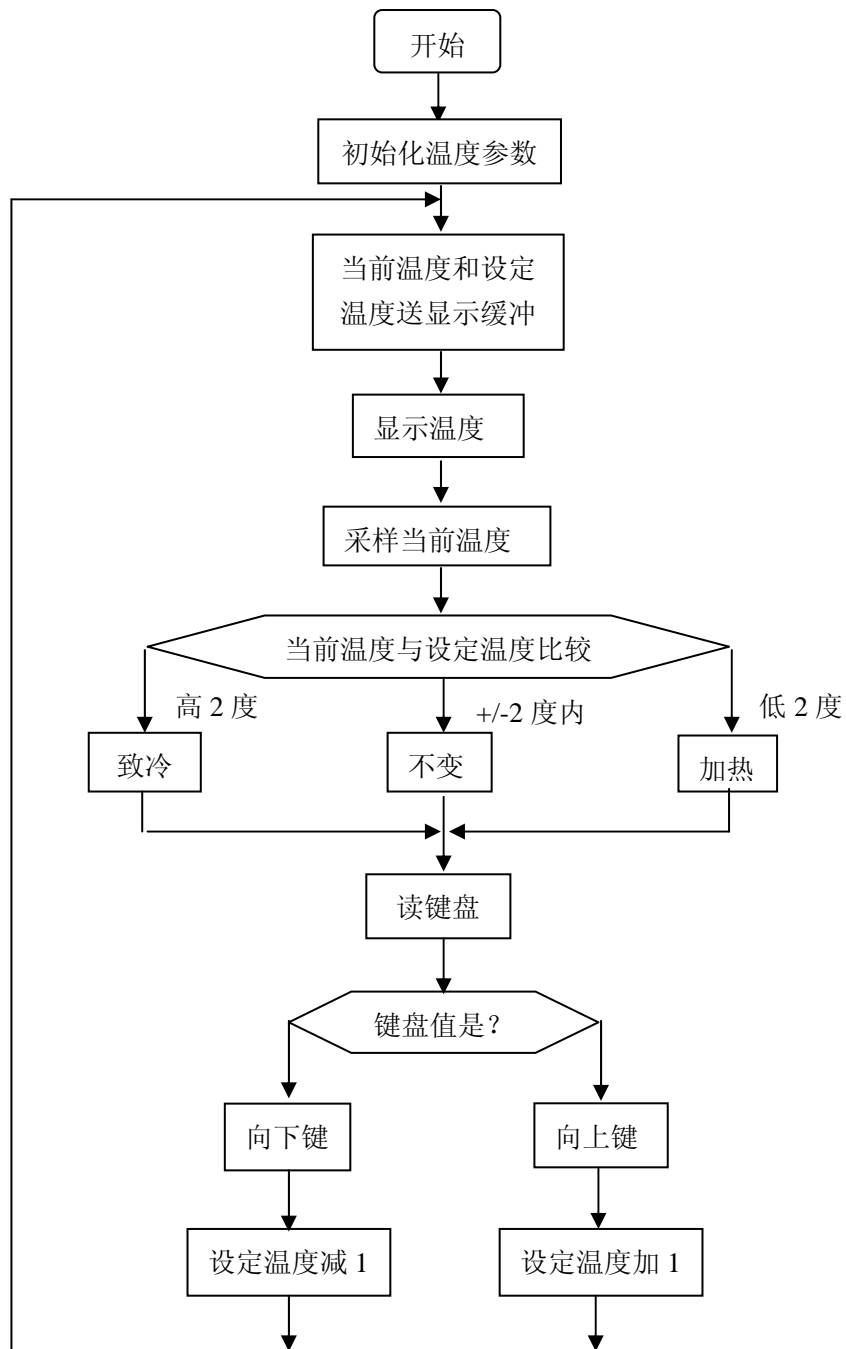
### A/D 采样子程序框图



LED 显示子程序框图



键盘扫描子程序框图



温度控制主程序框图

## 硬件实验二十七 计算器实验（51/96/88/ARM）

### 一、实验要求

利用实验仪上提供的显示电路和键盘电路，做一个简单的计算器。

用十六进制 A 键 = ‘+’、B 键 = ‘-’、C 键 = ‘x’、D 键 = ‘/’、E 键 = ‘=’、F 键 = ‘C’。

### 二、实验目的

1. 进一步熟悉 LED 显示电路和键盘扫描电路的工作原理和编程方法。
2. 了解数据计算的基本方法。

### 三、实验电路和连线

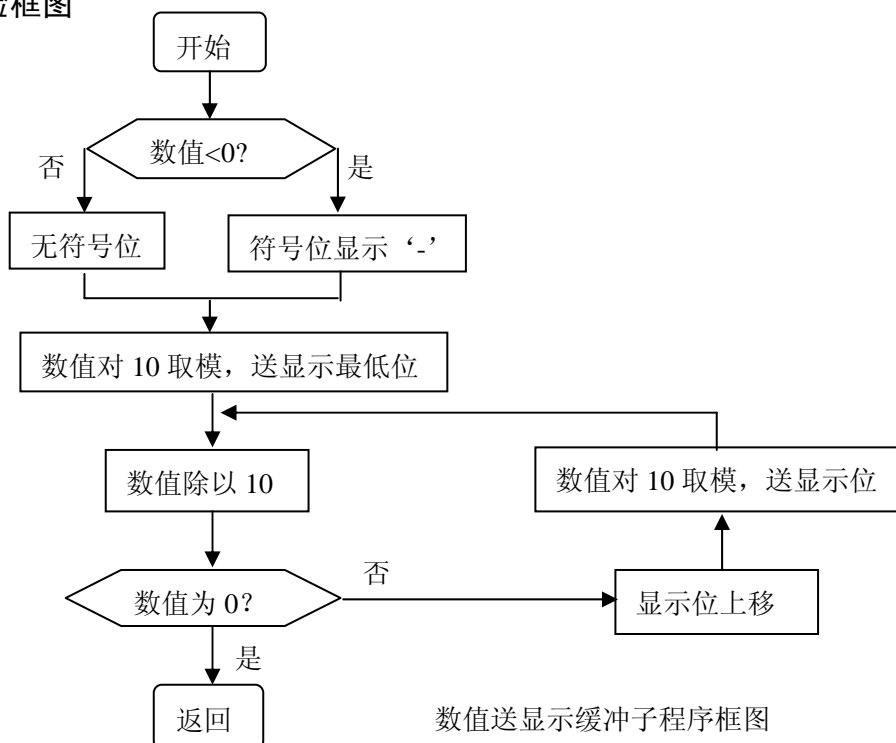
本实验的显示电路和键盘扫描电路已经连接好。具体原理图见第 4 页图 1。用户只需接上键盘/显示的选择信号。

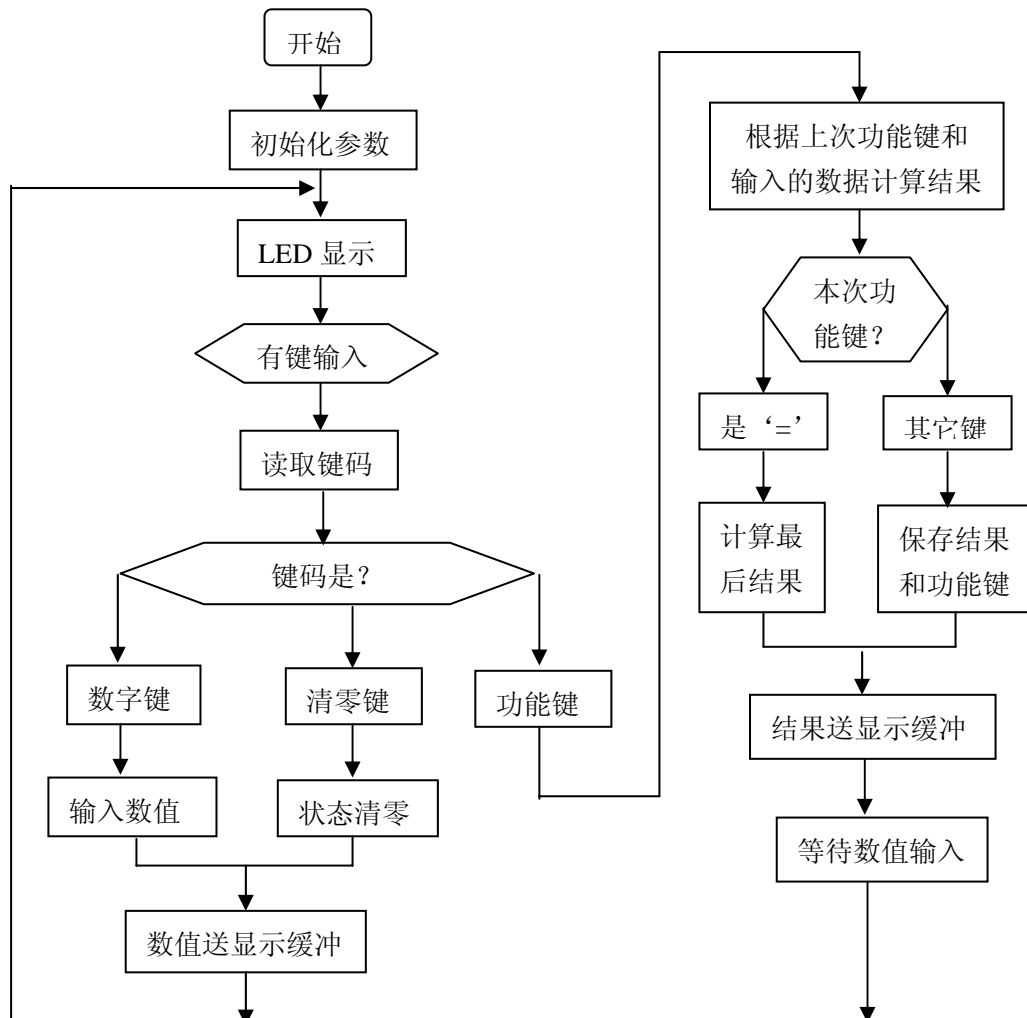
连线	连接孔 1	连接孔 2
1	KEY/LED_CS	CS1

### 四、实验说明

本实验的 LED 显示电路和键盘扫描电路见硬件实验十六、十七。

### 五、实验框图





硬件实验二十八 用 HS0 方式输出 PWM 波形（96）

一、实验要求

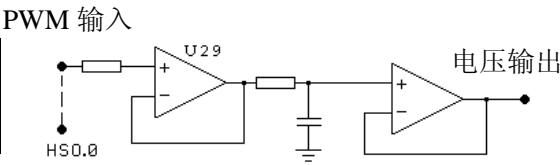
用 80C196 的高速输出（HS0）端口输出不同占空比的脉冲，通过 PWM 转换电压电路转换成电压。

二、实验目的

- 1. 了解脉宽调制(PWM)的原理
- 2. 学习 80C196 的 HS0 编程方法。
- 3. 学习用 PWM 输出模拟量

三、实验电路及连接

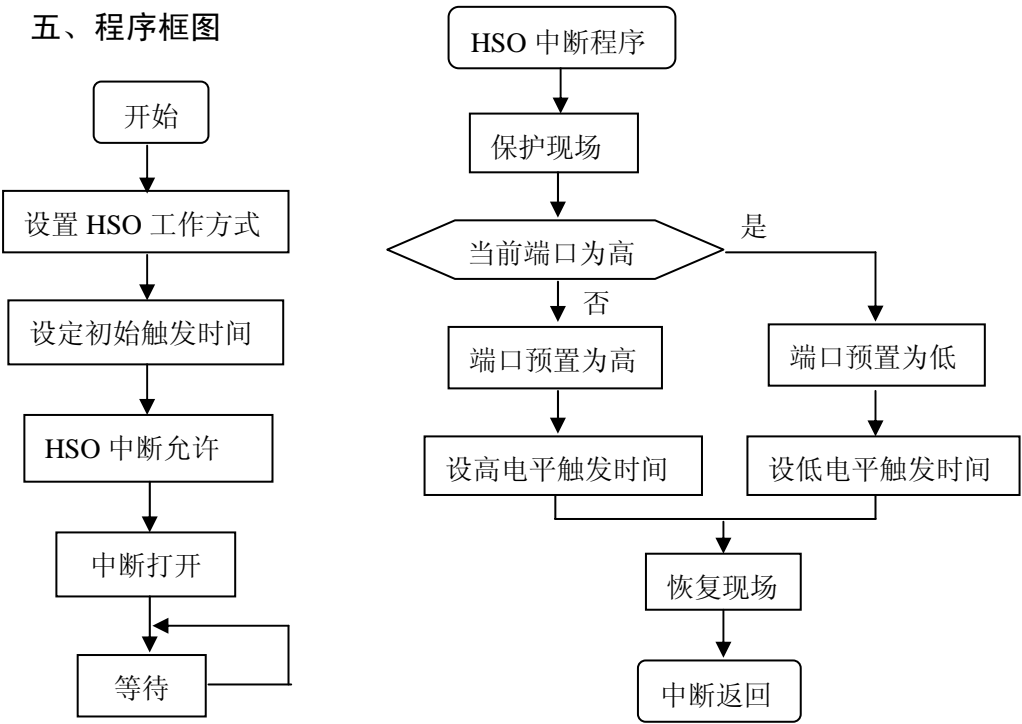
连线	连接孔 1	连接孔 2
1	HS0.0	PWM 输入
2	PWM 输出	电压表



四、实验说明

HS0 就是按特定的时间去触发一些事件。本实验就是在指定的时间后，使 HS0 脚上的电平翻转。通过改变触发时间来改变脉冲的占空比，从而实现 PWM 的输出。通过外接的转换电路，可以将脉冲的占空比变成电压。占空比就是脉冲中高电平与低电平的宽度比。

五、程序框图



硬件实验二十九 用 HSI 方式测量脉冲宽度（96）

一、实验要求

利用 80C196 的高速输入（HSI）功能测量输入的脉冲宽度，输入的脉冲可以用 P1.0 输出。

二、实验目的

- 1. 了解 80C196 的 HSI 的原理
- 2. 学习 80C196 的 HSI 编程方法
- 3. 掌握一种测量脉冲宽度的方法

三、实验电路及连线

连线	连接孔 1	连接孔 2
1	P1.0	HS0.0

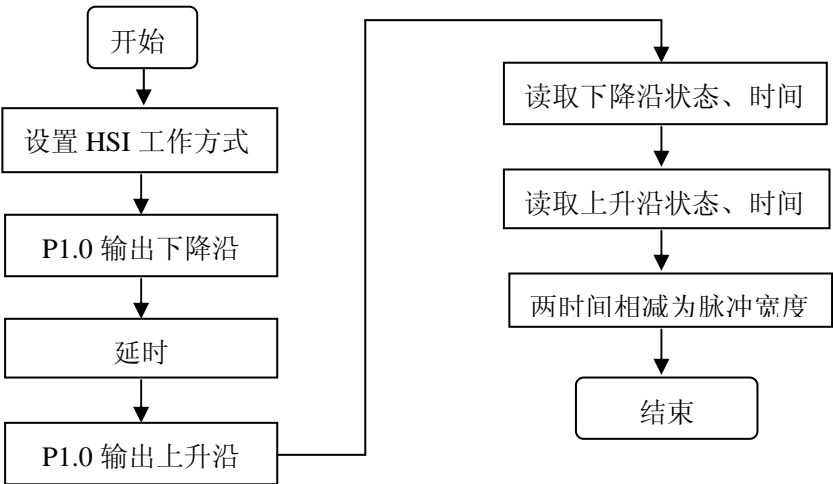
P1.0 ● - - - ● HSI0

四、实验说明

80C196 的 HSI 功能是用来捕捉 HSI 管脚上的状态变化及变化时间。HSI 有一个事件队列，用于存储多个事件。可以在事件发生后，用中断方式或查询方式读出。

本实验用 P1.0 输出一个下降沿，经过延时后再输出一个上升沿。这两个事件都被捕捉到，并且把时间存在事件队列中，在事件之后读出当时的时间，在程序的最后设一断点，观察 BX，CX，DX 寄存器，就可以看到两个事件的时间和时间差。

五、实验程序框图





## 硬件实验三十 用 HSI 中断统计脉冲个数 (96)

### 一、实验要求

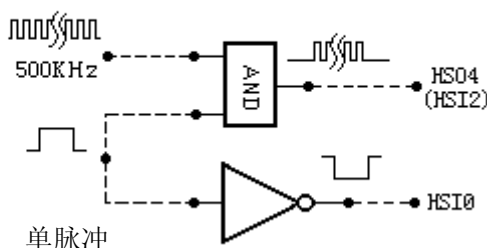
利用 80C196 的 HSI 功能统计输入的脉冲个数。可以用板上 500KHz 信号和单脉冲的输出相‘与’得到一定个数的输入脉冲。

### 二、实验目的

1. 了解 80C196CPU 的 HSI 原理
2. 学习 HSI 的编程方法及中断处理方法
3. 区分 HSI 的事件中断与 HSI.0 管脚中断的不同。

### 三、实验电路及连线

连线	连接孔 1	连接孔 2
1	4MHz	Fin
2	F/8	AND 门输入
3	单脉冲	AND 门输入
4	AND 门输出	HS04 (HSI2)
5	单脉冲	NOT 门输入
6	NOT 门输出	HSI0



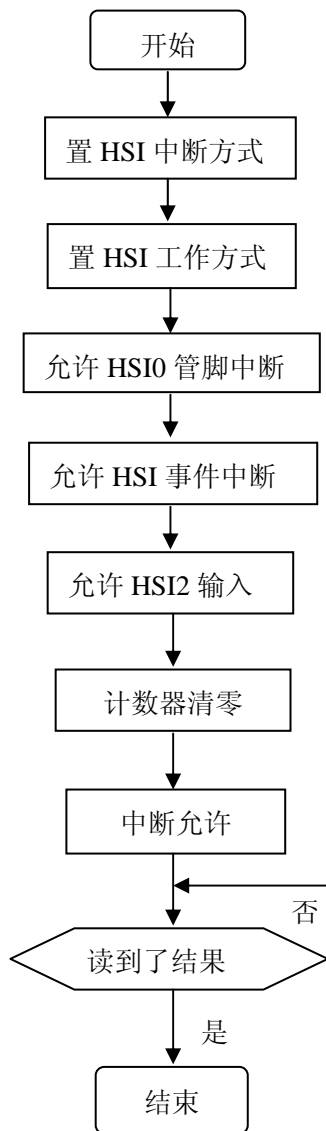
### 四、实验说明

本实验用单脉冲通过与门选择一串脉冲输出到 HSI2 脚,用 HSI 的中断方式对输入的脉冲进行计数,用 HSI.0 的管脚中断功能捕捉单脉冲的下降沿,也就是脉冲序列的结束位置,将统计结果取回。

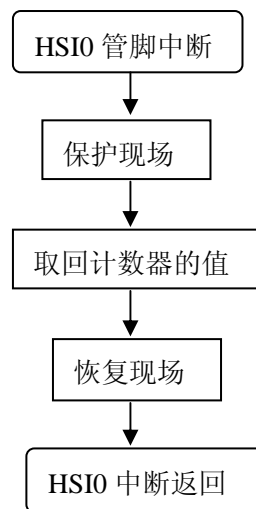
实验采用 HSI 的方式 0,也就是 HSI2 脚上每 8 个脉冲计数一次。500KHz 的信号周期为 2μs,这样也就是 16μs 计数一次。这样 16 位的计数器可计数约 1 秒钟时间的脉冲序列,实验时可以快速地按一下单脉冲键,输入一串脉冲到 HSI2,按键时间不能太长,以免 16 位计数器溢出。如果想加长统计时间,可以将计数器改成 32 位的。有兴趣的同学可以试试。HSI0 管脚中断可以捕捉接到 HSI.0 脚上的上升沿,以判断脉冲序列的结束。

将分频器的 Fin 接到 4MHz 脉冲发生器,经过分频后可得到 500KHz 时钟脉冲。

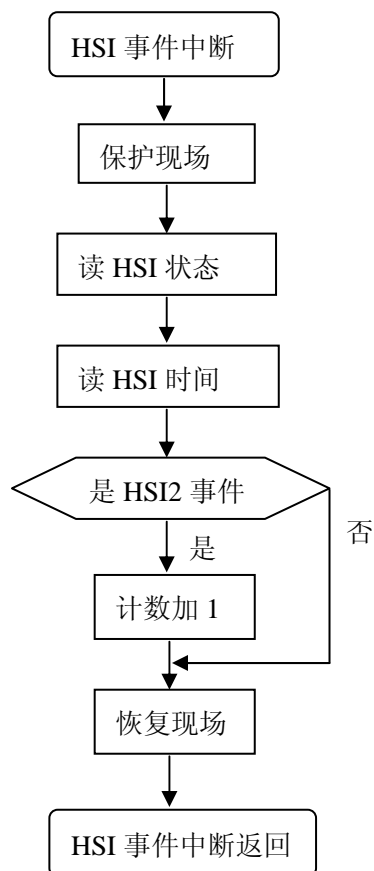
## 五、实验程序框图



主程序框图



HSIO 管脚中断程序框图



HSI 事件中断程序框图

硬件实验三十一 计数器实验（96）

一、实验要求

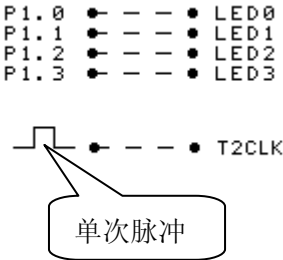
利用 80C196 内部定时计数器 T2，按计数器模式工作，对 T2CLK(P2.3) 引脚进行计数。将其数值按二进制数在 P1 口驱动 LED 灯上显示出来。

二、实验目的

- 1、学习 80C196 内部定时/计数器使用方法。

三、实验电路及连线

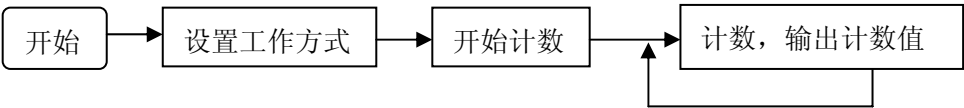
连线	连接孔 1	连接孔 2
1	P1.0	L0
2	P1.1	L1
3	P1.2	L2
4	P1.3	L3
5	单脉冲	T2CLK



四、实验说明

- 2. 本实验中内部计数器起计数器的作用。外部事件计数脉冲由 T2CLK(P2.3) 引入定时器 T2。单片机对外部脉冲的每个边沿（包括上升沿和下降沿）进行计数。可以从 P1 输出的结果看出每次按键，计数器都加 1。

五、实验框图



## 硬件实验三十二 片内 A/D 转换实验（96）

### 一、实验要求：

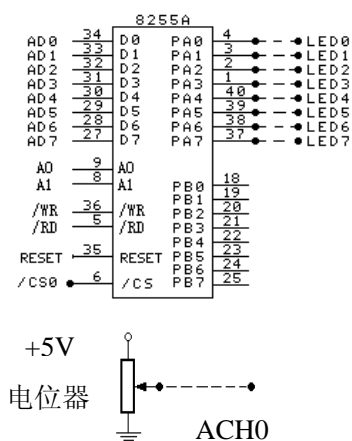
利用 80C196 片内的 A/D 转换模块,进行 A/D 转换实验,模拟量输入选用板提供的电位器,改变电位器值,观察 AX 寄存器中每次转换结果。将高 8 位 AD 结果用 8255 的 PA 口输出到 LED 灯上。

### 二、实验目的：

- 1、了解 80C196 片内的 A/D 转换电路的原理及编程方法。
- 2、进一步了解单片机采集数据的方法。

### 三、实验电路及接线：

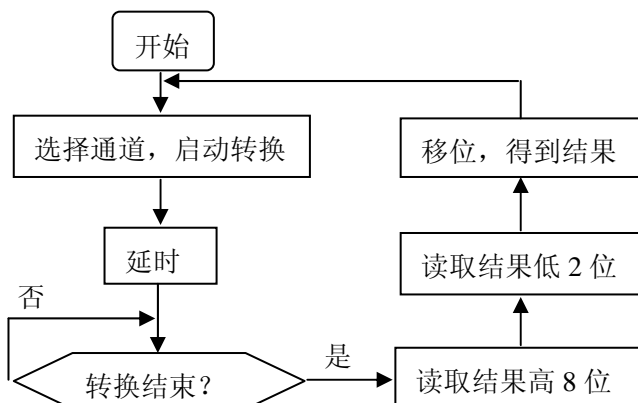
连线	连接孔 1	连接孔 2
1	电位器输出	ACH0
2	8255_CS	CS0
3	PA0	L0
4	PA1	L1
5	PA2	L2
6	PA3	L3
7	PA4	L4
8	PA5	L5
9	PA6	L6
10	PA7	L7



### 四、实验说明：

80C196 片内集成了一个 8 通道的 A/D 转换系统。包括模拟多路开关、采样和保持电路及 10 位逐次逼近的 A/D 转换器。通过 AD\_COMMAND 的 0~2 位选择转换通道。对 4 (GO) 位置 1, 就可以启动 A/D 转换。经过延时后, 对 AD\_RESULT\_LO 寄存器的第 3 (S) 位进行查询, 以判断 A/D 转换是否结束。如果结束, 可以分次读取 AD\_RESULT 的高 8 位和低 2 位。有兴趣的同学可以试试用中断方式读回 A/D 转换结果。中断向量地址为 2002H。

### 五、实验程序框图：



硬件实验三十三 PWM 转换电压实验（88）

一、实验要求

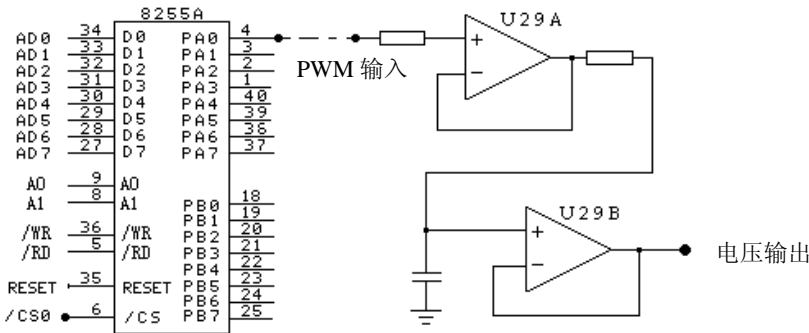
用 8255 扩展端口输出不同占空比的脉冲，通过 PWM 转换电压电路转换成电压。

二、实验目的

- 3. 了解脉宽调制(PWM)的原理
- 4. 学习用 PWM 输出模拟量

三、实验电路及连接

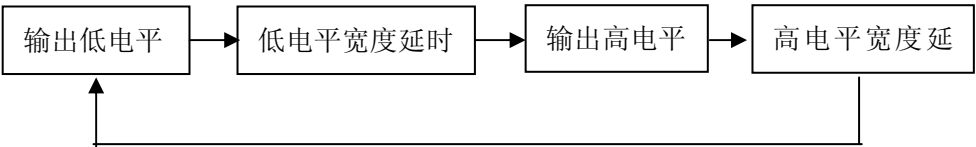
连线	连接孔 1	连接孔 2
1	8255_CS	CS1
2	PA0	PWM 输入
3	PWM 输出	电压表



四、实验说明

PWM 是单片机上常用的模拟量输出方法，通过外接的转换电路，可以将脉冲的占空比变成电压。8088/86 可用 8255 扩展输出输出不占空比的脉冲来实现 PWM。程序中通过调整占空比来输出模拟电压。占空比就是脉冲中高电平与低电平的宽度比。注意本实验是采用软件延时控制时间，如果对占空时间要求精确，可用定时器控制时间。

五、程序框图



硬件实验三十四 8253 计数器实验（88）

一、实验要求

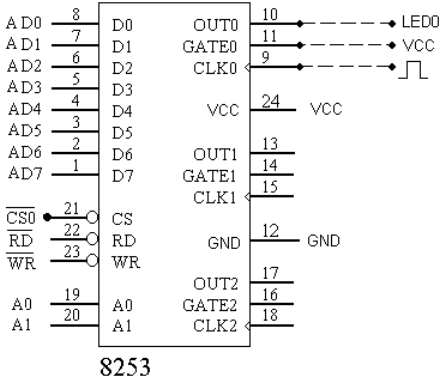
利用 8088/86 外接 8253 可编程定时器/计数器，可以实现对外部事件进行计数。设置断点读回计数器的值。

二、实验目的

- 1、学习 8088/86 与 8253 的连接方法。
- 2、学习 8088/86 对 8253 的控制方法。

三、实验电路及连线

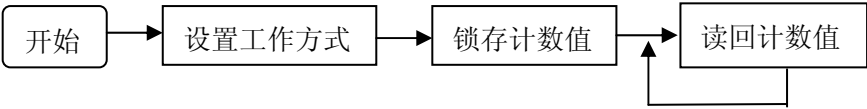
连线	连接孔 1	连接孔 2
1	8253_CS	CS0
2	8253_OUT0	L0
3	8253_GATE0	VCC
4	8253_CLK0	单脉冲



四、实验说明

- 3. 本实验中计数器按方式 0 工作。即十六位二进制计数器。当计数设置好后，计数器就开始计数。如果要读入计数器的值，要先锁存计数值，才能读到计数值。本实验所设计数值为 5，也就是外部 5 个脉冲，计数器值加 1。同时 OUT 脚输出一个高电平。实验时，可以将 OUT0 接到 LED 上，观察计数器是否工作。

五、实验框图



## 硬件实验三十五 8259 外部中断实验（88）

### 一、实验要求

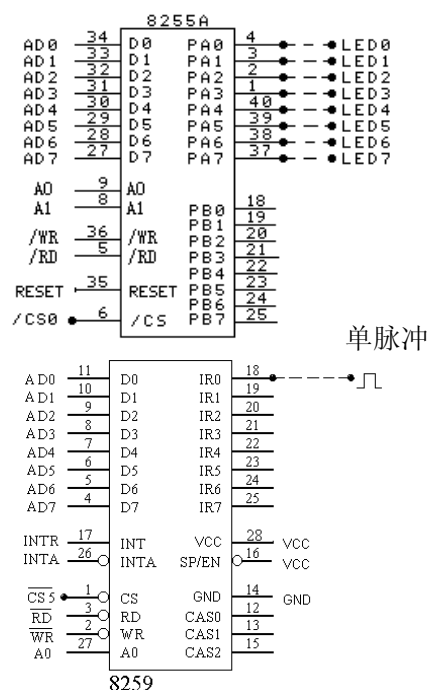
用 8088/86 控制 8259 可编程中断控制器，实现对外部中断的响应和处理。要求程序中对每次中断进行计数，并将计数结果用 8255 的 PA 口输出到 LED。

### 二、实验目的

- 1、学习 8088/86 与 8259 的连接方法。
- 3、学习 8088/86 对 8259 的控制方法。

### 三、实验电路及连线

连线	连接孔 1	连接孔 2
1	8259_CS	CS5
2	8259_INT0	单脉冲
3	8255_CS	CS0
4	PA0	L0
5	PA1	L1
6	PA2	L2
7	PA3	L3
8	PA4	L4
9	PA5	L5
10	PA6	L6
11	PA7	L7



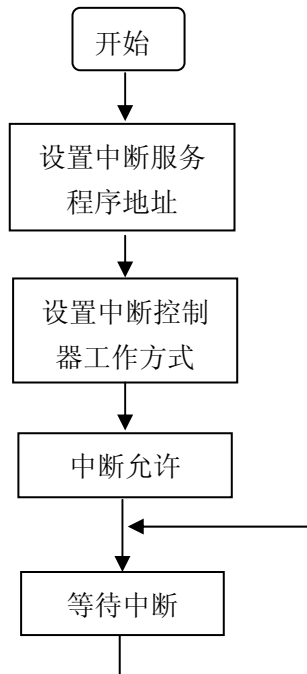
### 四、实验说明

8088/86 需要外接中断控制器才能对外部中断进行处理。在编程时应注意

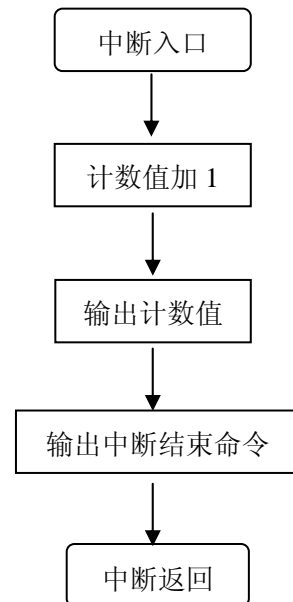
1. 正确地设置可编程中断控制和工作方式。
2. 必须正确地设置中断服务程序地址。

8259 可外接 8 个中断源，本实验只响应 INT0 中断，8259 也可以多级连接以响应多个中断源。将单脉冲信号接到 8259 的 INT0 脚。每次中断时，可以看到 LED 显示会加 1。

## 五、实验框图



主程序框图



中断子程序框图



## 硬件实验三十六 8253 定时器实验 (88)

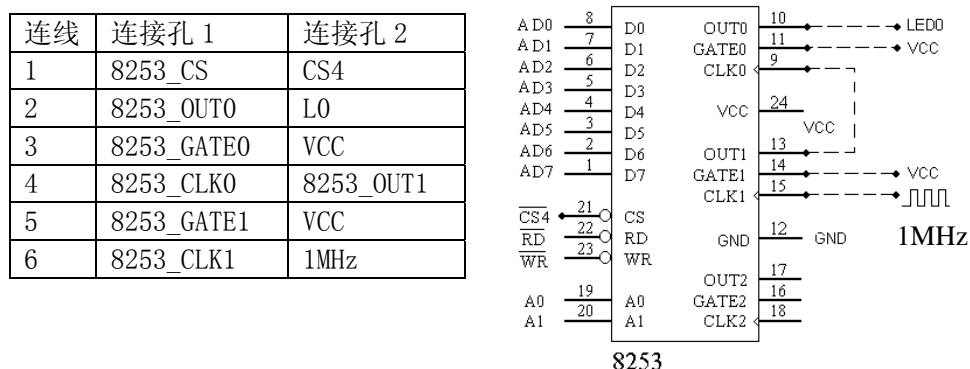
### 一、实验要求

用 8253 对标准脉冲信号进行计数, 就可以实现定时功能。用板上的 1MHz 做为标准信号, 将 8253 可编程计数器/定时器的时间常数设在 1000000 次, 就可以在定时器的管脚上输出 1 秒钟高/1 秒钟低的脉冲信号。因为 8253 每个计数器只有十六位, 要用两个计数器才能实现一百万次的计数, 实现每一秒钟输出状态发生一次反转。

### 二、实验目的

- 1、学习 8253 可编程定时器/计数器定时方法。
- 2、学习 8253 多级串联实现大时间常数的定时方法。
- 3、学习 8088/86 控制 8253 可编程定时器的方法。

### 三、实验电路及连线

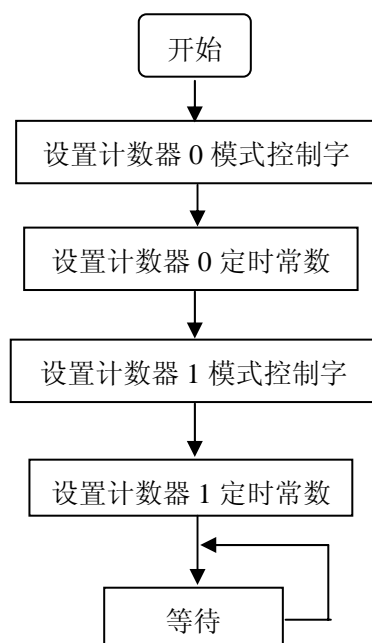


### 四、实验说明

1、本实验与实验五相似, 实验五用的工作方式 0, 计数值减完后输出一个脉冲宽度的高电平。而本实验在计数值减完后, 管脚状态产生变化 (从高到低或从低到高)。直到下一次计数值减完。这样输出的波形为方波。

2、由于定时常数过大, 就要用多级串联方式。本实验采用两级计数器。定时常数分别为 100 和 10000。将计数器的输出接到计数器 0 输入。计数器 0 的输出接到 LED0。

## 五、实验框图



主程序框图

## 硬件实验三十七 8251A 串行通讯实验（88）

### 一、实验要求

利用 8088/86CPU 控制 8251A 可编程串行通信控制器,实现两个实验台之间的串行通讯。其中一个实验台作为发送方,另一侧为接收方。发送方读入按键值,并发送给接收方,接收方收到数据后在 LED 上显示。

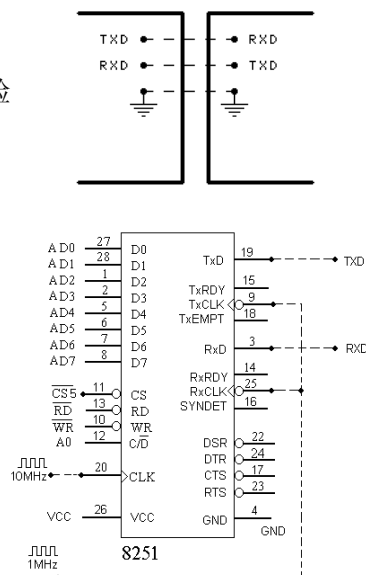
### 二、实验目的

- 1、掌握 8088/86 实验串行口通信的方法。
- 2、了解实现串行通讯的硬环境、数据格式的协议、数据交换的协议。
- 3、学习串行口通讯程序编写方法。

### 三、实验电路

显示电路和键盘电路见硬件实验十六和硬件实验十七。

连线	连接孔 1	连接孔 2
1	KEY/LED_CS	CS0
2	8251_CS	CS5
3	8251_TxCLK	1MHz
4	8251_RxCLK	1MHz
5	8251_CLK	10MHz
7	甲方 8251_TXD	乙方 8251_RXD
8	甲方 8251_RXD	乙方 8251_TXD
9	甲方 GND	乙方 GND



### 四、实验说明

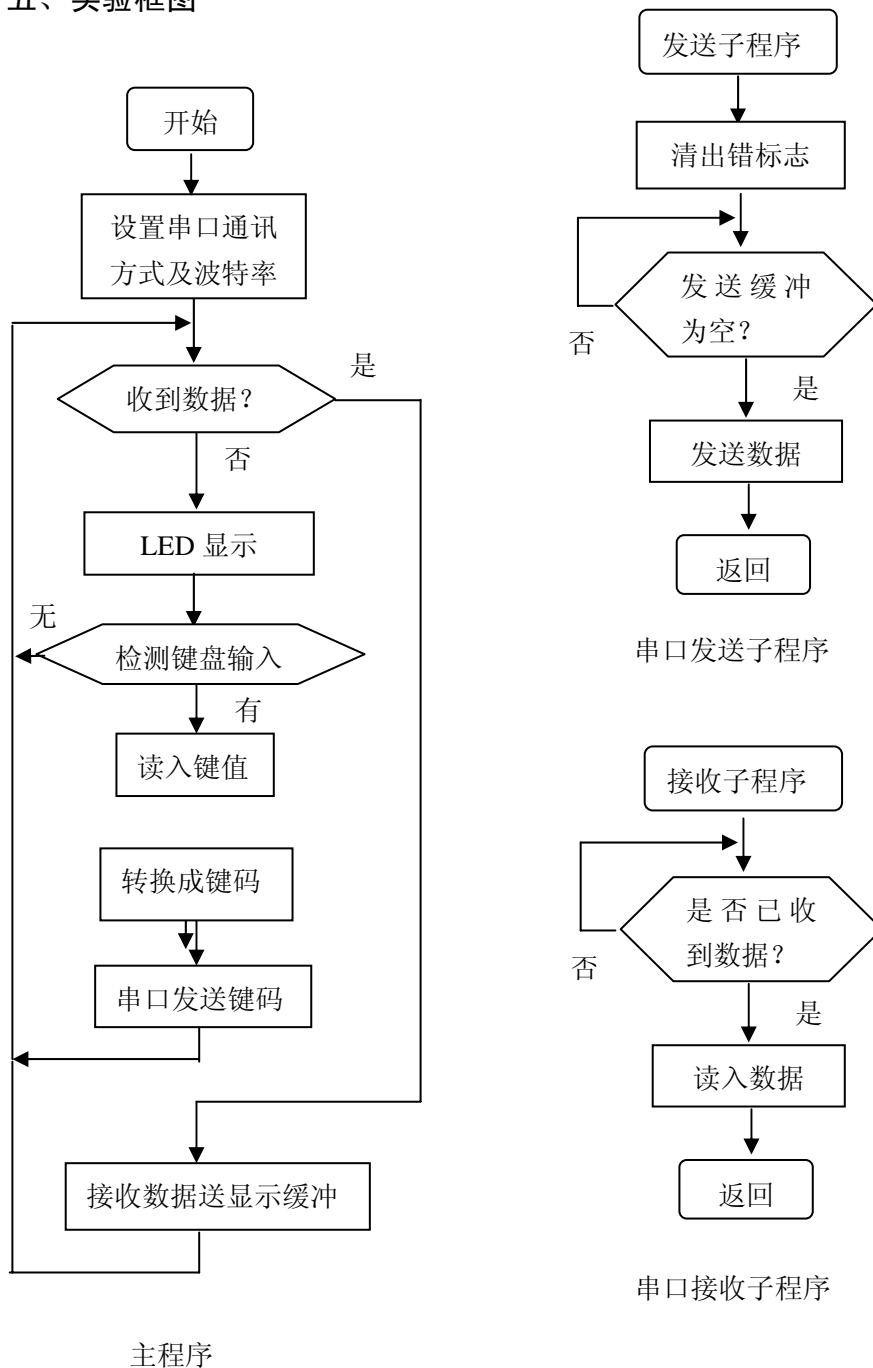
1、8088/86CPU 通过外接的 8251 可编程串行控制器实现串口通信。通过 8251 的控制口写入其工作方式或读入当前状态,通过数据口发送或接收串行口数据。本实验是通过查询方式读写状态和输入输出串行数据。

2、通讯双方的 RXD、TXD 信号本应经过电平转换后再行交叉连接,本实验中为减少连线可将电平转换电路略去,而将双方的 RXD、TXD 直接交叉连接。也可以将本机的 TXD 接到 RXD,这样按下的键就会在本机的 LED 上显示。

3、若想与标准的 RS232 设备通信,就要做电平转换,输出时要将 TTL 电平换成 RS232 电平,输入时要将 RS232 电平换成 TTL 电平。可以将仿真板上的 RXD、TXD 信号接到实验板上的“用户串口接线”的相应 RXD 和 TXD 端,经过电平转换,通过“用户串口”接到外部的 RS232 设备。

4、芯片工作所需时钟信号可利用板上提供的 10MHz 和 1MHz 时钟信号。

## 五、实验框图



## 硬件实验三十八 8237 DMA 实验 (88)

### 一、实验要求

利用 8088/86CPU 控制 8237 可编程 DMA 控制器, 实现两个存储器之间 DMA 块传输。

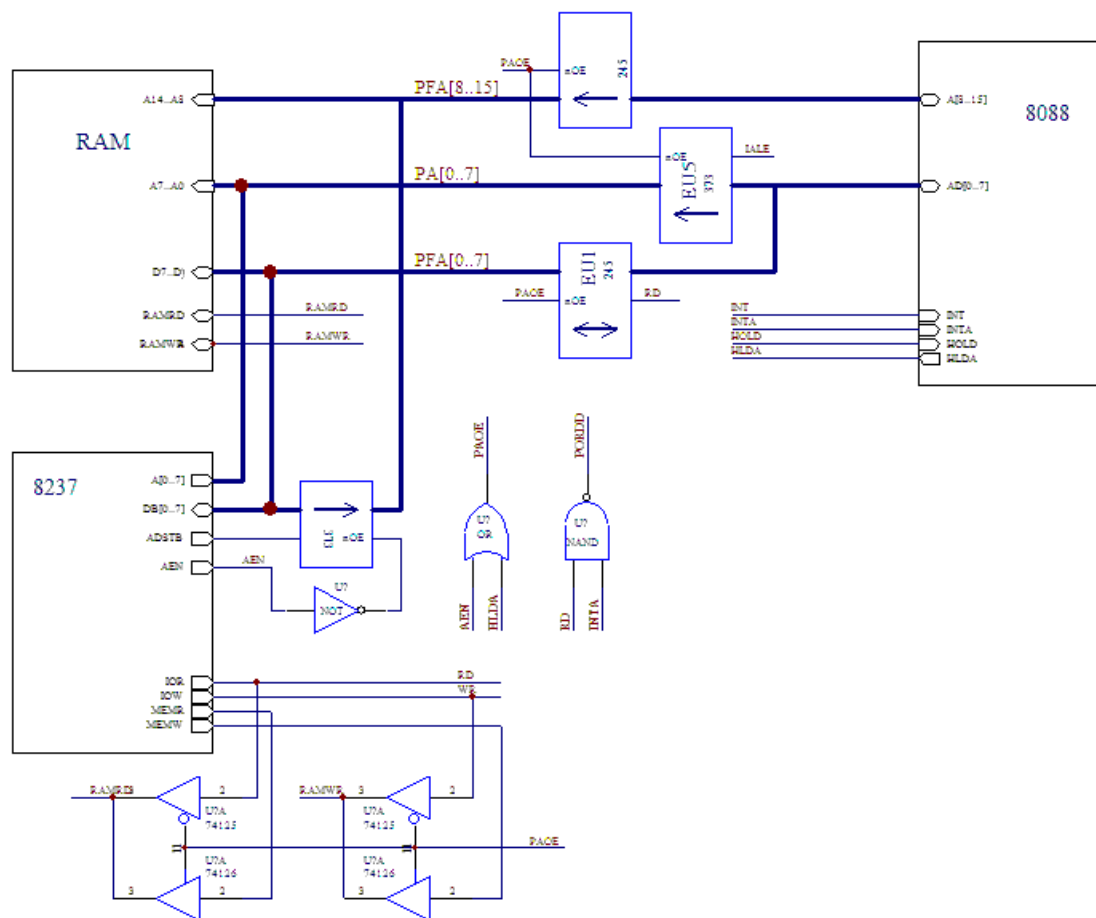
### 二、实验目的

1. 掌握 8237 DMA 控制器。
2. 学习 8237 DMA 块传输的编程方法。

### 三、实验接线框图

连线	连接孔 1	连接孔 2
1	RAM CS	CS0
2	8237_CS	CS1
3	8237_CLK	10MHz

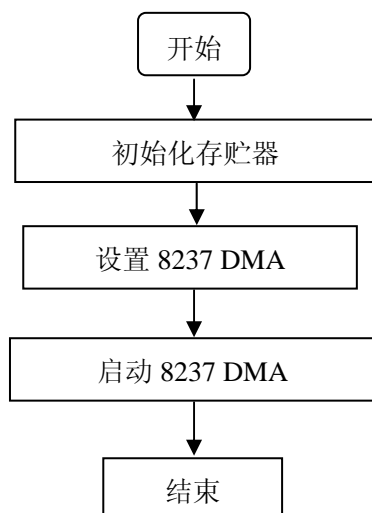
实验仪 8237 模块原理图如下: 其中大部分信号已连接, 只需接 RAMCS、CLK、8237CS。



#### 四、实验说明

1. 8088/86CPU 通过外接的 8237 可编程 DMA 控制器实现 DMA 传输。
2. 首先将存储器 8000H-80FFH 初始化
3. 设置 8237DMA, 设定源地址为 8000H, 设定目标地址为 8800H, 设定块长度为 100H
4. 启动 8237 DMA
5. 8237 DMA 工作后 8088 CPU 暂停运行程序, 总线由 8237 DMA 控制, 在 DMA 传输完 100H 个单元后, 8237A 将控制权还给 8088CPU, CPU 执行 RET 指令, 我们可以在 RET 指令处设断点。程序暂停后检查 8800H-88FFH。

#### 五、实验框图



## 硬件试验三十九 压力传感器实验（51/96/88/ARM）

### 一、实验要求

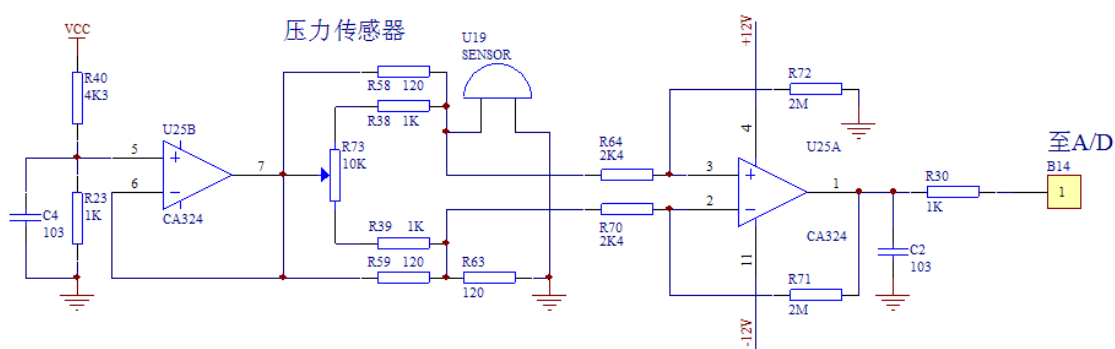
利用试验板上提供的压力传感器电路，完成压力的采样和显示。

### 二、实验目的

1. 了解压力传感器电路的工作原理。
2. 了解弱信号传感器的常见电路。
3. 进一步熟悉 A/D 变换电路的工作原理。

### 三、实验接线框图

连线	连接孔 1	连接孔 2
1	KEY/LED_CS	CS0
2	模数转换 AD_CS	CS2
3	压力传感输出	模数转换器 IN0



### 四、实验说明

上图是一个较常用的温度测量电路，大致分电源，电阻电桥，运放，输出四部分。具体的原理可参见温度传感器实验的说明。

### 五、实验程序框图

本实验主要介绍压力传感器的电路，具体程序可参见 A/D 变换实验和显示实验。

## 硬件试验四十 红外遥控实验（51/96/PIC/ARM）

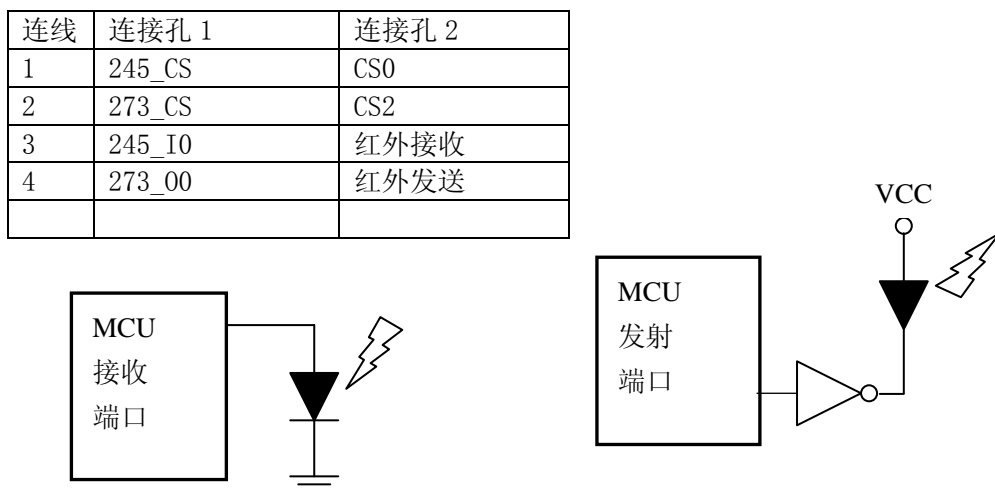
### 一、实验要求

利用实验仪上的红外器件，编写发送和接红外信号程序，实现近距离的无线通信。

### 二、实验目的

1. 了解红外遥控电路的原理，及编码方法。
2. 了解远程控制的一般原理和方法。
3. 学习如何编写红外发射和接收程序。
4. 了解单片机控制外部设备的常用电路。

### 三、实验接线框图



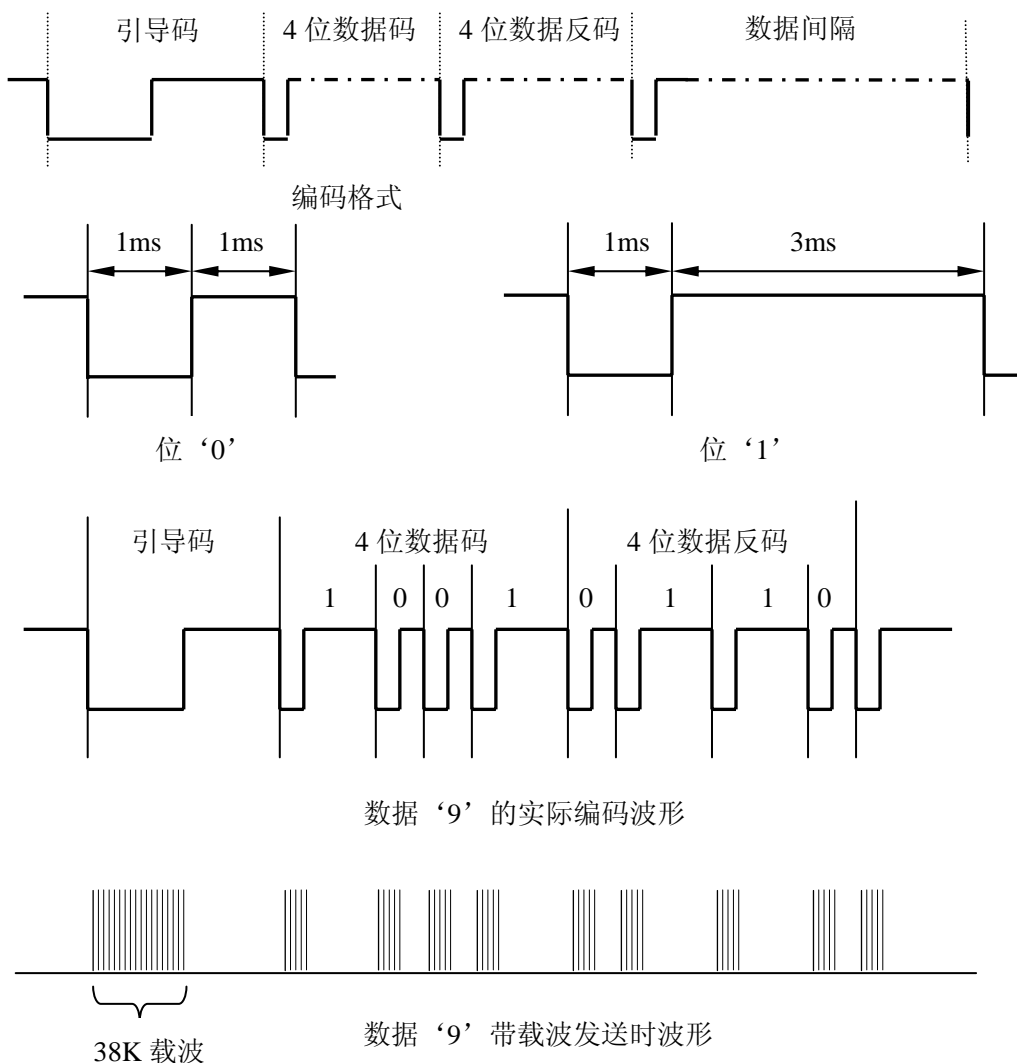
### 四、实验说明

红外遥控为现在最常用的近距离无线通信方式，它是将数字信号用红外线发送出去。为了能让受控设备能识别信号，要将数字信号编码，现今红外有很多编码标准，常见有 PHILIPS 的 RC5 格式和 NEC 格式。本实验为了简化，采用我们自己设计的一种编码方式。下面将详细说明。在普通场合，有很多红外发射源，有白炽灯、日光、发热体，这些都会干扰红外信号，所以在发射时，还要数字将脉冲信号调制在 30K-40K 的载波上，以抑制这些红外干扰。本实验采用最常用的 38K 载波。为了抗干扰，还可以在接收处适当地加一些隔离。本实验接收部分采用的是一体化接收头，将信号解调和放大全部做在一起，提高了可靠性。这样，接收头送到单片机的就是编码的数据信号，而不是调制信号，数据的解码通过单片机来完成。



本实验使用的编码包括四部分，引导码，4 位数据码，4 位数据反码和数据间隔。引导码用于标识一个数据的开始，数据码为有效数据，数据反码是将有效数据取反后编码，用于提高数据的识别率。

引导码由 5ms 低电平和 5ms 的高电平组成，数字位 ‘0’ 由 1ms 低电平和 1ms 高电平组成，数字位 ‘1’ 由 1ms 电平和 3ms 高电平组成。数据间隔为 20ms。

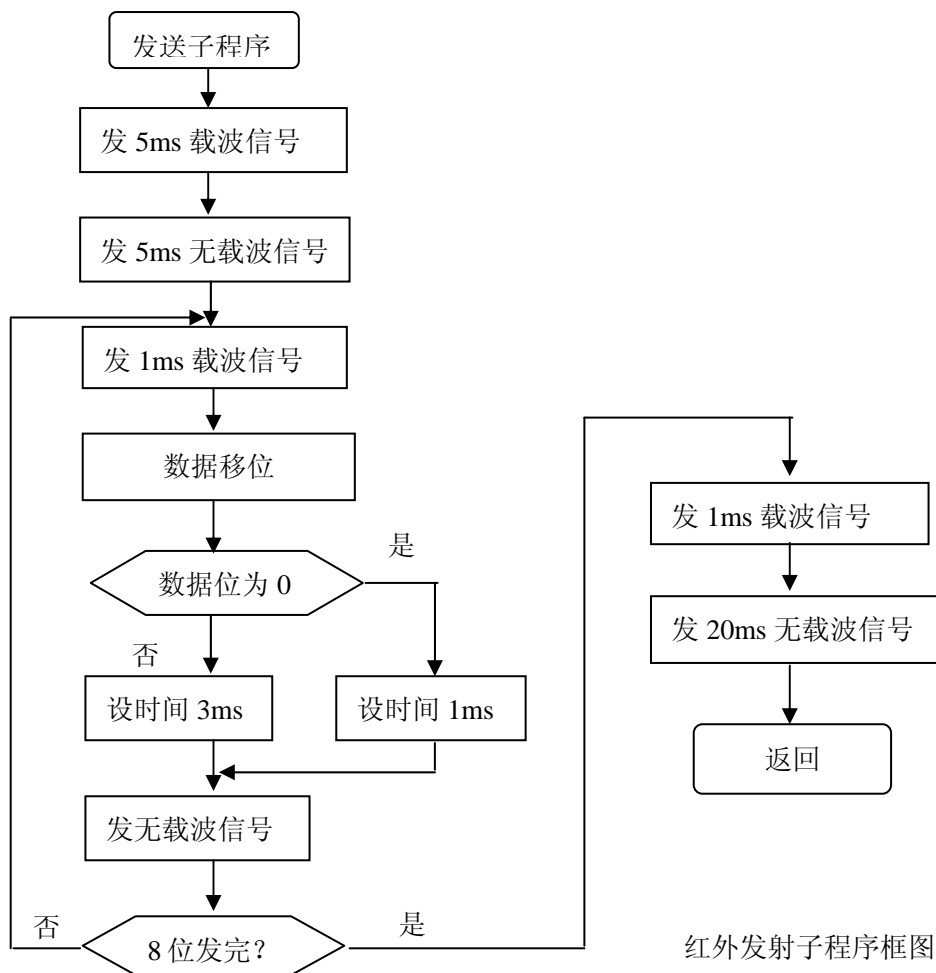


在用脉冲控制红外发射管时，是低有效。即当输出低电平时，红外管导通发光。单片机输出的脉冲信号被锁存在 74273 中，经 ULN2003A 反向驱动后，驱动红外管产生脉冲信号。

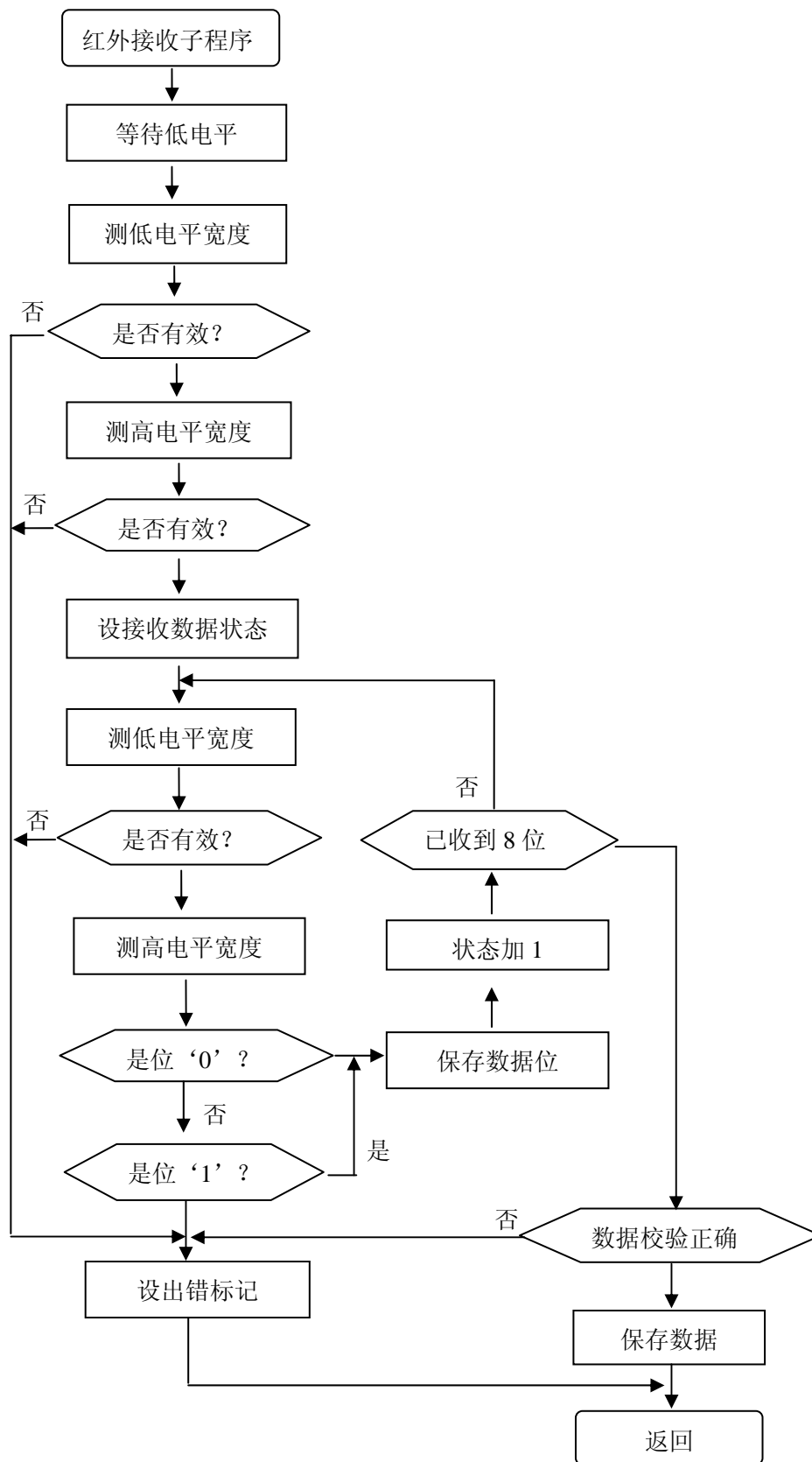
接收红外编码信号时，信号是从 74245 读回的。从图可以看出。红外信号是从总线的第 0 位读回的，这样，从 74245 读回数据,再判断第 0 位的高低，而得知信号状态。判断信号变化时间长短，就可以对信号进行译码，得到对方发过来的数据。

## 五、实验程序框图

这里只给出红外发送和接收的子程序框图，有关键盘和显示的程序和框图请参见本实验仪的键盘和显示实验的相关部分。



红外发射子程序框图



## 硬件试验四十一 16x16 点阵显示实验（51/96/88/ARM）

### 一、实验要求

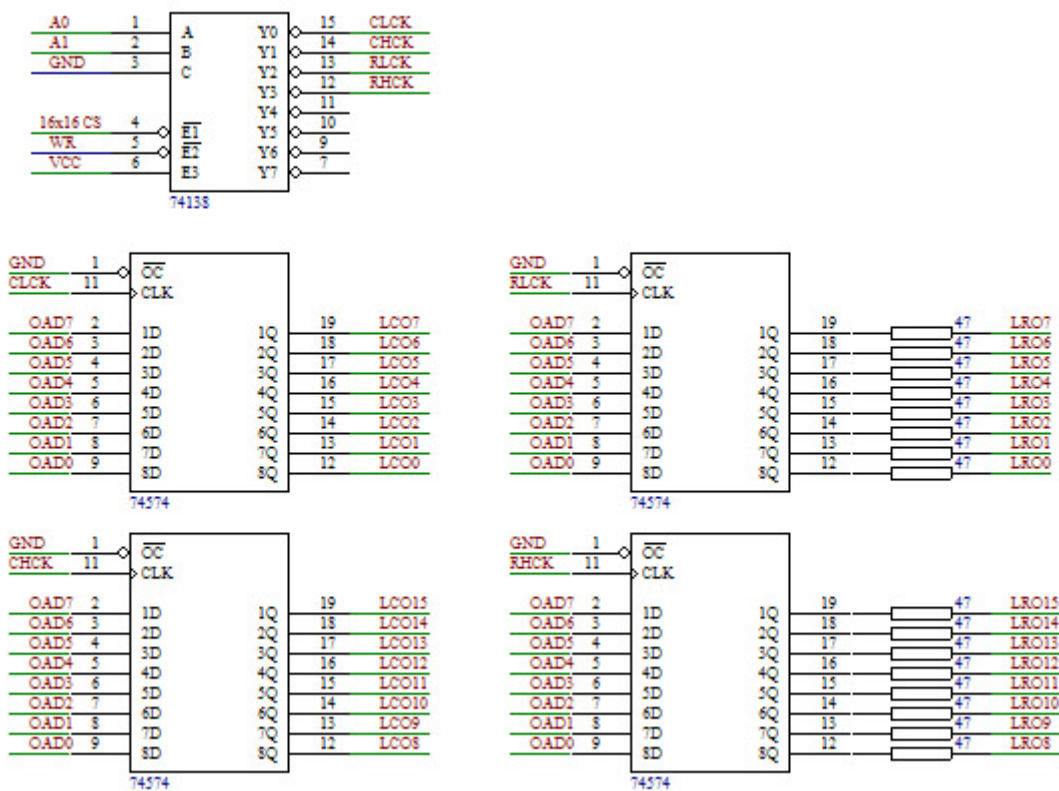
利用实验仪上的 16x16 LED 点阵显示器，编写显示英文、汉字字符程序。

### 二、实验目的

2. 了解 16x16 点阵电路的原理。

### 三、实验电路连线框图

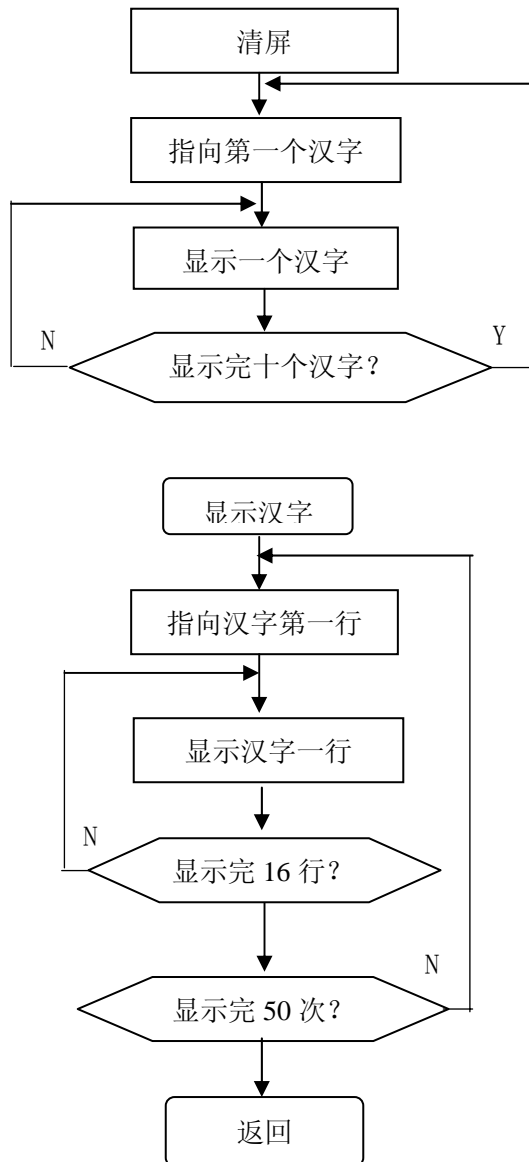
连线	连接孔 1	连接孔 2
1	16x16_CS	CS3



### 四、实验说明

16x16 点阵需要 32 个驱动，分别为 16 个列驱动及 16 个行驱动。每个行与每个列可以选中一个发光管，共有 256 个发光管，采用动态驱动方式。每次显示一行，10ms 后再显示下一行。

## 五、 实验程序框图



## 硬件试验四十二 I<sup>2</sup>C 总线实验（51/96/PIC/ARM）

### 一、实验要求

利用实验仪上的 I<sup>2</sup>C 总线器件 AT24C01A/AT24C02，编写 I<sup>2</sup>C 总线读写程序并自行验证程序的正确性。

### 二、实验目的

1. 了解 I<sup>2</sup>C 总线基本原理
2. 掌握典型 I<sup>2</sup>C 总线器件 AT24C01A/AT24C02 的读写操作

### 三、实验电路连线框图

连线	连接孔 1	连接孔 2
1	SCL	P1.0
2	SDA	P1.1

MCS51 / MCS96 接线表

连线	连接孔 1	连接孔 2
1	SCL	PB0
2	SDA	PB1

PIC5X 接线表

连线	连接孔 1	连接孔 2
1	SCL	P0.2
2	SDA	P0.3

LPC2103 接线表

对于 LPC2103，有两种实验方式，一是用 I/O 端口模拟 I<sup>2</sup>C 总线时序对外部器件进行读写，二是用片内的 I<sup>2</sup>C 模块对外部器件进行读写，用片内 I<sup>2</sup>C 的实验程序在 H42B 文件夹内

### 四、实验说明

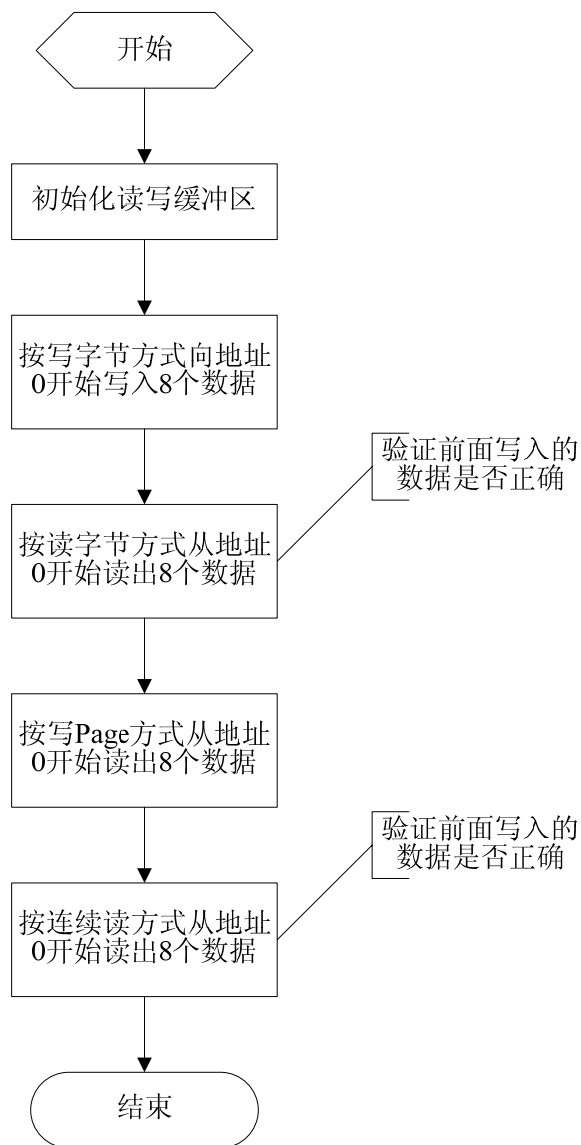
AT24C01A/AT24C02 为 I<sup>2</sup>C 总线型 EEPROM 存储器，容量为 1K/2K 位(128/256\*8)，前读/写时序遵循 I<sup>2</sup>C 总线协议标准。AT24C01A/AT24C02 内部设有一个控制寄存器，其每一位的含义如下：

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	0	1	0	A2	A1	A0	R/W

其中 A2/A1/A0 用于选择总线上待访问的 I<sup>2</sup>C 器件，R/W=1 读操作，R/W=0 写操作；从上述时序可以看出，I<sup>2</sup>C 总线上最多可以扩展  $2^3=8$  片同样的 1K/2K 容量 EEPROM 存储器或者可以扩展 1 片容量为 16K Bits 的 EEPROM 存储器。如果扩展 8 片 2K 以内容量的 EEPROM 存储器，每片存储器将对应一个地址，这个地址由芯片的 A2/A1/A0 决定，因此在同一个 I<sup>2</sup>C 总线上扩展多片同样的 EEPROM 存储器时，需要注意任意两片的地址不能相同。

由于实验仪上的 AT24C01A/AT24C02 的 A2/A1/A0 引脚全部接地，等效为实验仪上的 AT24C01A/AT24C02 的地址为 0，所以在实验中读写控制字分别为：0xa1/0xa0

## 五、实验程序框图



## 硬件试验四十三 SPI 总线实验（51/96/PIC/ARM）

### 一、实验要求

利用实验仪上的 SPI 总线器件 AT93C46，编写 SPI 总线读写程序并自行验证程序的正确性。

### 二、实验目的

1. 了解 SPI 总线基本原理
2. 读写 AT93C46，编写典型的 SPI 总线器件控制程序。
3. 了解带 SPI 接口的 MCU 的编程方法。

### 三、实验电路连线框图

MCS51/MCS96 接线表

连线	连接孔 1	连接孔 2
1	CS	P1.0
2	SK	P1.1
3	DI	P1.2
4	DO	P1.3

LPC2103 接线表

连线	连接孔 1	连接孔 2
1	CS	扩展板上 SSEL0(P0.7)
2	SK	扩展板上 SCK0(P0.4)
3	DI	扩展板上 MOSI(P0.6)
4	DO	扩展板上 MISO(P0.5)

做 PIC5X MCU 实验时, PB0 连接 CS, PB1 连接 SK , PB2 连接 DI , PB.3 连接 DO。

### 四、实验说明

AT93C46 为 EEPROM 存储器，容量为 1K 位(128\*8)，微控制器（MCU）通过四条线对其进行读写操作。其读/写时序与 SPI 总线协议标准基本相同，可以通过编写 93C46 的读写程序，了解一般的 SPI 总线器件的控制方法。

串行外设接口（SPI）协议由 Motorola 制定，被很多流行的处理器和微控制器所采用，SPI 接口需要两条控制线（CS 和 SCLK）和两条数据线（DIN/SDI 和 DOUT/SDO）。为了叙述的清晰，在这里从器件的角度出发说明三线接口的数据线。DI 是器件的数据输入线，DO 是器件的数据输出线。这里我们使用 CS，CK，DI 和 DO 来定义三线接口中的各条线。

SPI 接口可以工作在全双工模式（数据可以在同一时间发送和接收）。而且三线接口是边沿触发的而不是电平触发，因此具有更强的抗干扰能力。三线接口的主要缺点是它要为每一个器件提供一条片选线，除非将从设备用菊链形式连接。另外一个缺点是三线接口没有应答机制去判断数据的收发是否正确。

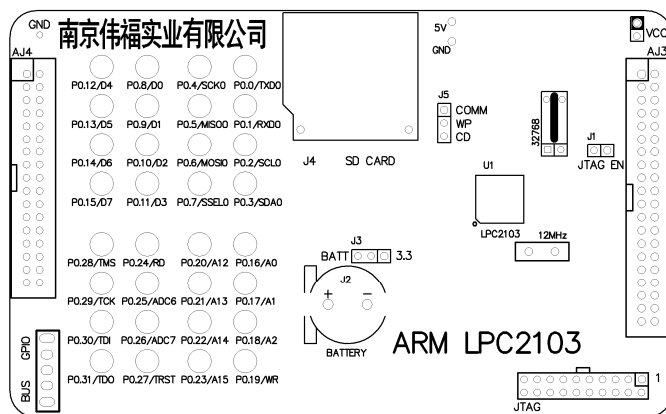
更详细的 SPI 控制时序请参照相应器件的资料和数据手册。



## 第四章 ARM LPC2103 仿真板的说明

伟福(WAVE)的 LAB8000 通用微控制器实验系可配各种 MCU,除了系统自带的 MCS51, i8088/86 以及 PIC57 系列,还可以通过扩充方式来增加实验微控制器的种类,如 MCS96 及 ARM 系列。对于 ARM 系列微控制器,我们选用了容易入手的 NXP 公司的 LPC2103,其内核是 16/32 位 ARM7TDMI-S CPU,内嵌实时仿真功能,内置 32K 高速 FLASH 存储器,8K RAM,片内还带有两个 UART, SPI, I2C 接口,8 路 10 位 A/D 转换器。ARM LPC2103 的扩展板如上图。LPC2103 的扩展板上配有 32 个 IO 端口的接线柱,方便与 LAB8000 系统中的各种实验电路连接;配有 JTAG 仿真调试接口,这样既可以用 LAB8000 上的仿真系统来调式程序,也可用独立的 JTAG 仿真器调试程序;配有 SD 卡的插座并与 LPC2103 的 SPI0 四个端口连接。因为 LAB8000 上有很多总线方式访问的实验电路和设备,但 LPC2103 本身没有总线外扩的端口,我们用它的 IO 口模拟了 MCU 的总线,除了模拟出数据总线外,还模拟了访问外设所需的地址总线和读写控制信号,使用时只要将扩展板上的控制开关拨到 BUS 方式即可。在用 IO 口模拟总线同时,我们还保留了 LPC2103 端口的复用功能,例如 UART0, I2C0, SPI0, ADC6, ACD7 等,以便这些设备可与总线控制的设备同时工作。

当用配用 ARM LPC2103 扩展板时,LAB8000 可以在 KEIL 和 ADS 两种集成开发环境下工作,两种环境下程序略有不同。我们对于两种开发环境都做了样例程序,便于同学们参考。



**AJ3、AJ4**, 是 ARM LPC2103 与 LAB8000 实验仪的连接插座,通过这两个插座将扩展板的 CPU 的模拟总线接到实验仪上的总线控制器件,并将扩展板上的 JTAG 控制信号与实验仪上的仿真器相连接。

**32 个接线柱**,是从 CPU 的 PORT0 口接出,其中 **P00~P07** 为原来端口功能不变,这 8 个端口的第二功能分别可做为 UART0 的 TXD, RXD, I2C0 总线的 SDA, SCK, SPI0 的 SCK, MISO, MOSI, SSEL; **P08~P15** 用来模拟总线的数据总线 D0~D7,这是一个程序控制双向的总线; **P16~P18** 用来模拟总线的地址总线 A0~A2, **P20~P23** 用来模拟总线的地

址总线 A12~A15, 地址总线的高四位 A12~A15 用于译码片选, 选中总线上的器件, 地址总线低三位 A0~A2 用于选择器件内部地址; **P19** 用来模拟总线的写信号 **WR**; **P24** 用来模拟总线的读信号 **RD**; **P25~P31** 为原来端口功能不变, **P25, P26** 的第二功能为模数变换 ADC 的 ADC6 通道, ADC7 通道。**P27~P31** 在 LPC2103 扩展板连接到实验仪上时, 只能做为 JTAG 仿真端口使用。

**BUS/GPIO 选择开关**, 当开关拨向 **BUS** 位置时, 扩展板上用 CPU 的 IO 口模拟的总线就连接到实验仪的总线器件上。这时 CPU 相应的端口就不能作为 I/O 口使用, 否则会造成系统工作不正常。当开关拨到 **GPIO** 位置时, 模拟的总线与实验仪上的总线断开, CPU 用来模拟总线的端口恢复原来的功能, 可以做 I/O 口用, 也可以做端口本来的第二、第三功能使用。

**JTAG 仿真插座**, 当 LPC2103 扩展板不连接到 LAB8000 实验仪上单独使用时, 可以用独立的 JTAG 仿真器接到此插座对 CPU 进行调试、仿真、编程。因为扩展板的电源是由 LAB8000 实验仪提供的, 如果扩展板脱离实验仪单独使用, 需要另外加上 +5V 电源。

**J1**, JTAG 功能允许跳线, 短接这个跳线, **P27~P31** 才能工作于 JTAG 仿真方式, 无论是扩展接到 LAB8000 实验仪还是用独立的 JTAG 仿真器, 如果要对 CPU 进行调试、仿真、编程, 必须短接 **J1** 跳线。

**J3**, RTC 电源选择跳线, 当你需要用电池为 RTC 供电时, 将跳线跳到 **BATT** 一边 (这时你需要安装上电池), 一般实验时我们用板上的 3.3V 电源供电, 所以通常情况下跳线跳到 3.3V 一方。(本扩展板不包含电池及电池座)。

**J4**, SD 卡插槽, SD 卡的 SPI 接口接到 CPU 的 **P04~P07** 端口, 也就是 **SPI0** 接口上, 当程序把这四个端口设为 SPI 工作方式时, CPU 就可以用 SPI 方式操作 SD 卡了。如果想用 SPI1/SSP 控制 SD 卡, 首先要将 **BUS/GPIO** 拨到 **GPIO** 方式, 禁止用 I/O 口模拟总线, 然后程序中将 **SPI0** 的四个端口 (**P04~P07**) 设为 I/O 方式, 并置成输入态 (一般情况下, LPC2103 上电后, 这些端口就自动为 I/O 输入状态), 将 **SPI1** 的端口的四根接线柱用连线接到 **SPI0** (**P04~P07**) 的接线柱上, 因为 **SPI0** 为输入方式, 所以 **SPI1** 的控制信号不会与其冲突。

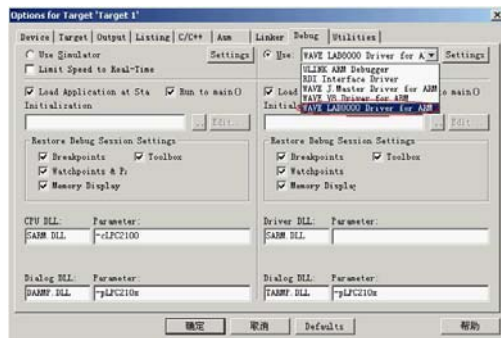
**J5**, SD 卡的写保护开关状态和卡插入状态检测信号, 当检测到 **CD** 与 **COMM** 短接时, 说明有 SD 卡插到卡槽中, 当检测到 **WP** 与 **COMM** 短接时, 说明 SD 是写保护的。如果你程序中有要求检测写保护状态和卡插入状态, 就将这三个信号接到 CPU 的 I/O 上, 在程序进行检测。

**RTC 晶振**, LPC2103 的 RTC 可以在程序中选择系统时钟作为 RTC 时钟源, 也可选择外接 32768Hz 晶振作为时钟源。本扩展板不包含 32768Hz 的 RTC 晶振, 在 RTC 实验中, 给出的样例程序是选择系统时钟作为 RTC 时钟源。

## 在开发环境中添加伟福 LAB8000 实验仪的驱动

### 一. 在 KEIL 开发环境下添加 LAB8000 驱动

在 KEIL 开发环境中，选择项目的“Options for Target”对话框，在“Debug”栏中，选择硬件仿真方式，在仿真器选择栏中，根据仿真器型号，选择“WAVE LAB8000 Driver for ARM”，然后按“Settings”按钮，弹出仿真器设置的对话框。



右图为伟福 LAB8000 for ARM 的设置对话框。这里我们可以看到仿真器的型号，编号，版本号，用户板电压，JTAG 的工作频率，这些参数在 LAB8000 实验仪上是固定的，不可人为设置。

“JTAG 设备”是根据读出的芯片 ID 码，自动显示的芯片型号，芯片类型。如果不能自动显示芯片型号和芯片，可以选在“JTAG 设定”中选择“手工设定”方式，然后选择芯片类型及芯片名称，对于某些具体芯片，可以根据你手中的芯片资料，在“更多设置”里对其进行详细的手工设置。



“缓存选项”中是将芯片的存储器、寄存器在电脑中做相应的缓存，如果打勾选中该项，就会建立缓存，如果不选中该项，就不会在电脑建立缓存，需要显示的数据都要从芯片里读回。一般情况下，都打勾选中，以提高仿真速度。

“闪存编程选项”中，是在编译完程序后，要执行的操作。编译完成后，要将程序代码下载到用户板上，一般步骤是：擦除、编程、校验、复位并运行。一般情况下，全部打勾选中，如果有某项没被选中，就不会执行相应的操作，可能会导致调试失败。（由于 KEIL 的 FLASH 算法有问题，LPC2103 在擦除里只能选“扇区擦除”，而不能选“全片擦除”）



“编程算法”是对不同的地址空间选择不同的编程算法。不同型号的芯片内部 FLASH 的编程算法有可能不同，芯片内部 FLASH，外部 FLASH 的编程算法也不同，所以需要不同的芯片，不同的地址空间要实行不同的编程算法。KEIL 提供了这些 FLASH 的具体实现方法，这些算法的描述文件是按一定的规则编写，被保存在某个文件夹下，在“编程算法目录”里，可以用“浏览”找到文件所在的文件夹，然后用“添加算法”找到你所用器件的算法，并加到“编程描述”栏内。LPC2103 内部带 32K FLASH，选用的算法是“LPC2000 IAP2 32KB Flash”“编程算法占用 RAM”是指对 FLASH 编程时编程程序运行空间，很多芯片在外部地址空间还没有初始化时不能定，一般芯片内部的 RAM 在上电后就可以直接使用，编程程序可以运行在这个地址空间里，不同的芯片内部 RAM 的地址不一样，需要在“编程算法占用 RAM”栏里设定一下内部 RAM 的起址和空间大小。LPC2103 的内部 RAM 起始地址为“0X4000000”，片上 RAM 空间大小为 8K，编程时只要用到其中 2K 即可，所以使用 RAM 大小填“0X800”就可以。“程序算法运行速度”填 CPU 晶振的速度，以 Hz 为单位。我们这里填的是“1000000”

按“好”确认退出。软件会对仿真器进行初始化，如果不成功，会弹出对话框提示仿真器初始错误，如果是手动设置的芯片类型，就要检查是不是手工设置错误。还有硬件连接和开发板是不是有错。

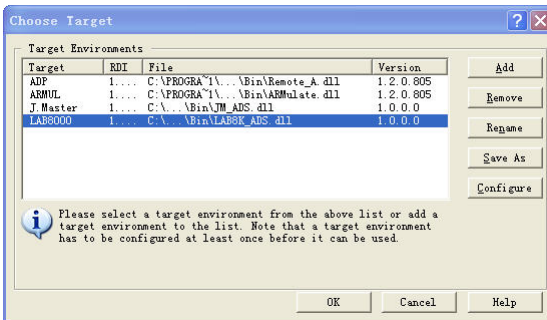
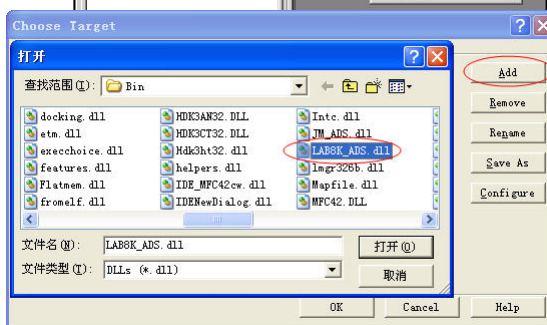
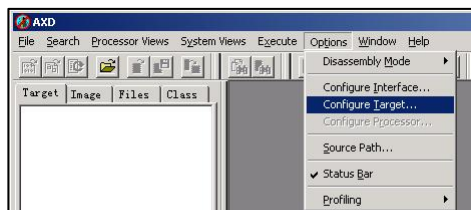
## 二. 在 ARM ADS 开发环境下添加 LAB8000 实验仪驱动

在 ARM 的 ADS 调试环境中，添加 LAB8000 实验仪驱动是在 AXD 中完成的，AXD 是一个独立的调试程序，可以直接运行，也可以在 CodeWarrior 中启动。

在实验仪软件的安装过程中，已经将 ADS 环境下实验仪的驱动程序“LAB8K\_ADS.DLL”安装在“..\ARM\ADSv1.2”文件夹下。

在 AXD 中选择菜单“Options”的“Configure Target...”，弹出仿真器的设置对话框，用“Add”在“..\ARM\ADSv1.2”找到 LAB8000 for ARM 的驱动程序“LAB8K\_ADS.dll”，点“打开”加到驱动列表里，然后用“Configure”打开驱动设置对话框，对实验仪进行驱动设置。

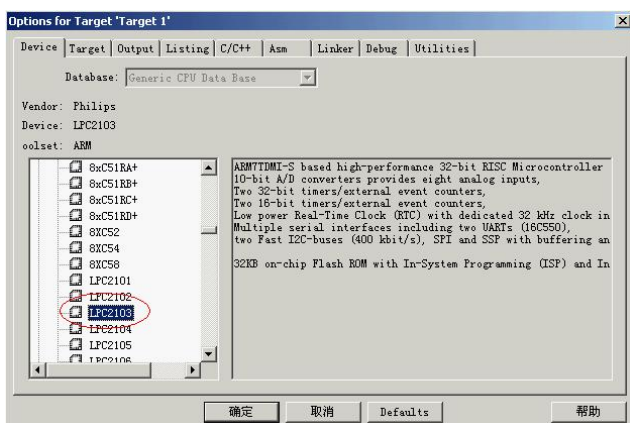
ADS 开发环境下的“Configure”实验仪驱动设置与 KEIL 开发环境下的“Settings”驱动设置完全相同，这里就不多说。可以参照前面的设置说明对仿



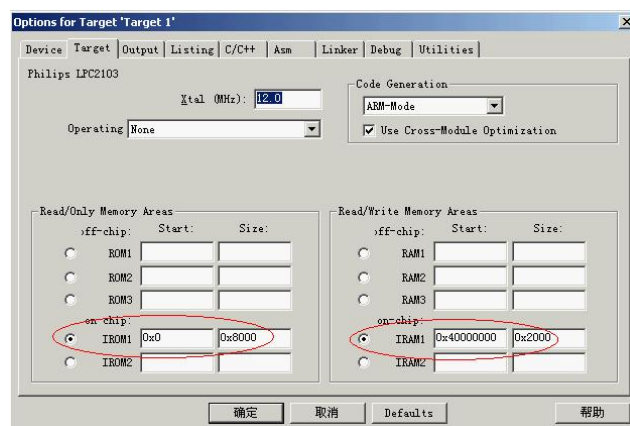
真进行设置。“Configure”设置里包括对 CPU 的编程，当启动 AXD 准备对程序进行调试时，AXD 会自动根据编译的结果，将代码下载到相应空间，下载时根据设置里的编程算法对芯片进行编程。这里有一点要注意，如果在编译时把程序空间放在了芯片内部 RAM 上，而不是片内 FLASH，就不能用 FLASH 的编程算法对 RAM 编程，所以就必须要要在“Configure”里把编程算法删掉不用。AXD 会直接把程序下载片内 RAM 里。与此不同的是，在 KEIL 开发环境里，只有按了“Download”键，才会对 FLASH 进行编程，而不像 AXD 是自动下载的，所以即使编译时将程序空间放在片内 RAM 里，仿真时也不用删掉“Settings”的编程算法。但是，在 KEIL 环境里，如果你程序空间在片内 FLASH 上，编译后就必须按“Download”对芯片进行编程，否则无法进行 DEBUG 调试。

## 在 KEIL 开发环境下的项目设置

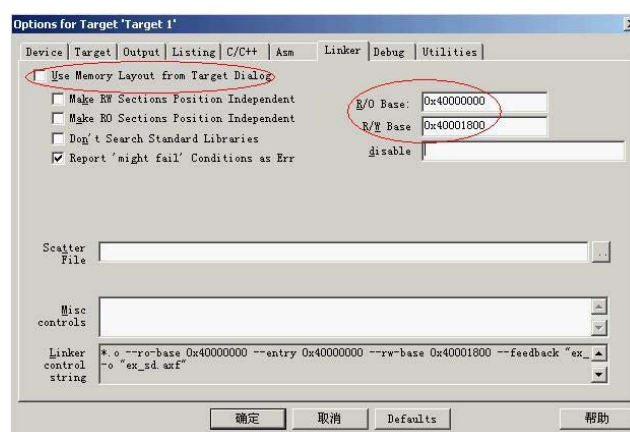
器件的选择，在“Options for Target”对话框中“Device”页面，找到并选择 NXP 公司的 LPC2103 芯片（Philips 公司的半导体公司现改为 NXP 公司，所以在早期的 KEIL 开发环境里，LPC2103 不是 NXP 公司而是 Philips 公司的产品）。



在“Target”页面，对目标器件进行设置，填上扩展板所用的 CPU 晶振频率 12MHz，选中 on-chip ROM 和 on-chip RAM 的地址范围（片上 ROM 和片上 RAM），LPC2103 的片上 FLASH 是从 0X00 开始，空间大小为 32KB，即 0X8000；LPC2103 片上 RAM 起始地址从 0X40000000 开始，空间大小为 8KB，即 0X2000。



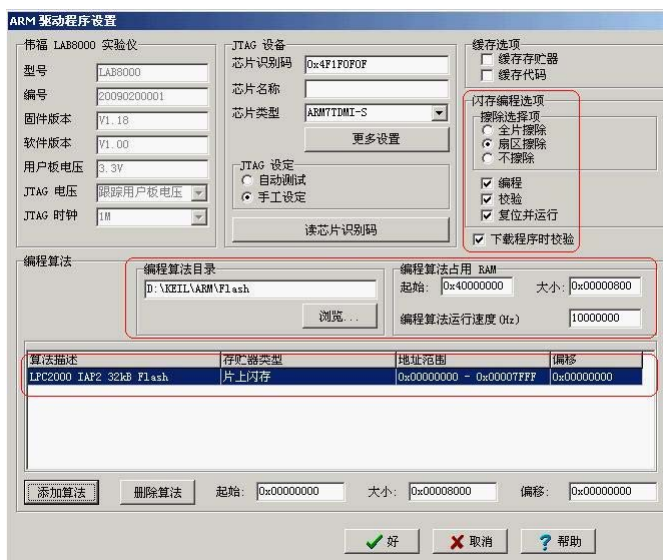
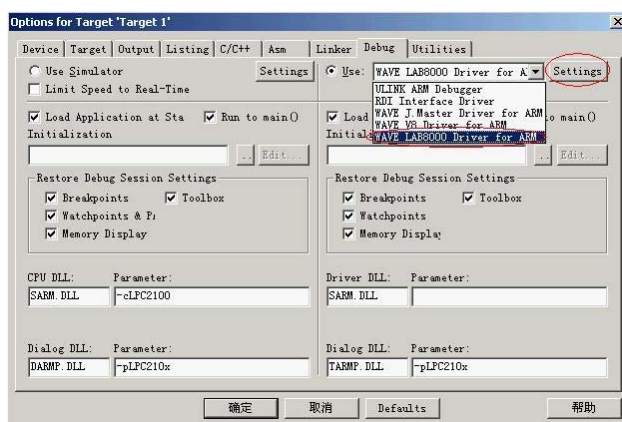
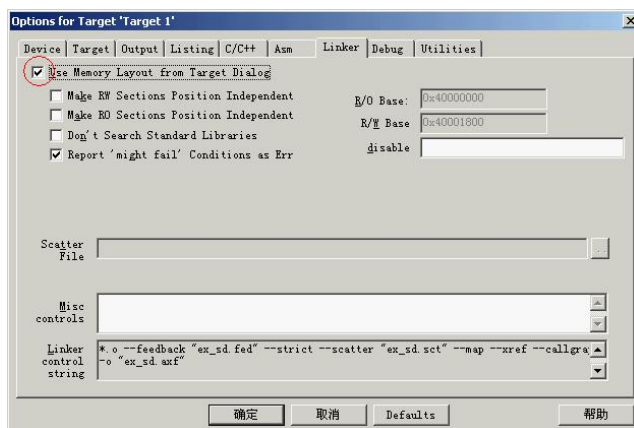
在“Linker”页面，可以选择程序代码的运行空间，链接器（LINKER）根据这个设置来产生代码。当选中“Use Memory Layout from Target Dialog”前面的选择框时，程序运行空间在片内 FLASH 中，当没有选中这条（前面框内没勾中）时，要自行设置程序运行的 ROM，RAM 空间，这样我们可以将程序放片内 RAM 里来运行，根据 LPC2103 的片内 RAM 容量，我们把 0X40000000 ~ 0X400017FF 这段空间做程序空间，把剩下的 0X40001800 ~ 0X40001FFF 做为数据空间。如果代码的程序空间加上数据空间





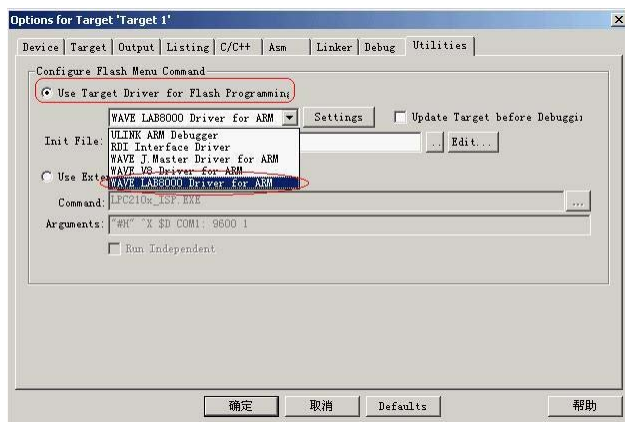
不超过 8K, 就可以把代码下载到片内 RAM 里运行。一般做实验, 可以把代码放到片内 RAM 上运行, 对于真正产品, 程序还是要下载到片内 FLASH 上, 这样才不会因断电丢失程序。

在“Debug”页面, 选择仿真器, 并对仿真器进行设置, 在下拉框里找到并选择“WAVE LAB8000 Driver for ARM”, 按“Settings”对实验仪上的仿真器进行设置, 如果电脑与实验仪的 USB 线连接正确, 可以在设置对话框里看到从 CPU 里读回的芯片识别码, LPC2103 的识别码为“0X4F1F0F0F”。有关对片内 FLASH 编程下载的设置如右图, 有关编程下载设置的说明可参见前面内容。如果要把程序下载到片内 FLASH 里, 除了编程算法法要选择正确外, 在“Utilities”页面还要选择一下编程所用的方式。

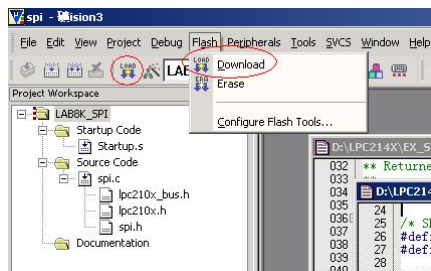


在“Utilities”页面,选中“Use Target Driver for Flash Programming”,并在下拉框里选择“WAVE LAB8000 Driver for ARM”,再按“Settings”对编程算法进行设置。这里按“Settings”调出的设置对话框与在“Debug”页面调出的设置对话框的相同。

如此设置好以后,在开发环境中如果程序编译无误,就可以下载到片内 FLASH 里,然后启动 DEBUG 进行调试了。



前面说了,根据空间设置的不同,程序可以在片内 FLASH 里或片内 RAM 里运行,这两种运行方式下载程序方式稍有不同,在 FLASH 中运行或调试程序,首先要将程序下载到片内 FLASH 里去,根据前面的说明,选择正确的 FLASH 编程方式,在程序编译无误后可以选菜单上的“Flash -> Download”,也可以用快捷按钮。当下载并校验完成后就可以按“d”快捷键进行调试。



如果程序运行空间被设置成在片内 RAM 里,除了前面所说的 LINK 的时候要选择程序空间,还需要加一个初始化文件,让 PC 定位到内部 RAM 的地址上,在“Options for Target”设置对话框的“Debug”页面,将“Load Application at Start”前面选中,点“Initialization”后面的“...”按钮添加初始文件,初始文件一般以 INI 为后缀,最简单的初始化文件如下:

```
FUNC void Setup(void) {
    PC = 0x40000000;
}

LOAD SPI.AXF INCREMENTAL

Setup();
```

这个初始文件将程序调入片内 RAM,并将 PC 定位在 RAM 最开始的地方 0X40000000,也就程序最开始的地方。初始文件可以自己编写保存成文本格式,也可复制其它项目的





初始文件，修改后使用。当程序被装入 RAM 后，就可以运行调试了。在片内 RAM 上运行程序可以不用对片内 FLASH 编程，特别是在做实验时，这种方法很方便快捷，但因为 LPC2103 片内只有 8KB RAM，这个空间里既有程序，又有数据，如果程序加上数据超过了 8KB，就不能在片内 RAM 上运行你的程序了。

注意，这样将 PC 定位到片内 RAM 的初始化文件，只适合程序在 RAM 里运行的例子，如果程序是运行在 FLASH 里，要在“Initialization”里删掉这个初始化文件，或加以修改以适合 FLASH 运行程序的需要。更详细的初始化文件编写方法，可以参考 KEIL 开发环境的说明和帮助。

在 ADS 开发环境下，当启动 AXD 对程序进行调试时，AXD 会根据编译的结果，自动将代码下载到相应空间，下载时根据设置里的编程算法对芯片进行编程。这里有一点要注意，如果在编译时把程序空间放在了芯片内部 RAM 上，而不是片内 FLASH，就不能用 FLASH 的编程算法对 RAM 编程，所以就必须要要在“Configure”里把编程算法删掉不用。AXD 会直接把程序下载片内 RAM 里。与此不同的是，在 KEIL 开发环境里，只有按了“Download”键，才会对 FLASH 进行编程，而不像 AXD 是自动下载的，所以即使编译时将程序空间放在片内 RAM 里，仿真时也不用删掉“Settings”的编程算法。但是，在 KEIL 环境里，如果你程序空间在片内 FLASH 上，编译后就必须按“Download”对芯片进行编程，否则无法进行 DEBUG 调试。

## ARM 仿真调试时可能出现的错误信息及原因

### 未找到仿真器

原因：仿真器电源未打开、USB 电缆没有连接

### 未知芯片

原因：没有找到用户板上的芯片，可能是仿真电缆没有连接或是用户板没有上电

### 只支持 2 个硬件断点

原因：设置的断点数大于 2 个。ARM 芯片只能支持 2 个断点，请减少断点数量

### 不是 ARM 系列芯片

原因：仿真器找到芯片，但不是 ARM 芯片。可能是用户板不对或 JTAG 时钟过高

### JTAG 通信错

原因：仿真器与芯片通信错。可能是连接有问题

### 编程算法运行错误

原因：在进行 FLASH 编程时发生错误。可能是 FLASH 损坏或 FLASH 连接有问题

### 编程算法重复，地址：xxxx

原因：所指出的地址有 2 个以上的编程算法。是用户在设定编程算法时产生了地址重复

### 未找到编程算法，地址：xxxx

原因：所指出的地址处没有相应的编程算法。请加入相应的编程算法

### 编程失败。编程中的芯片无应答

原因：收不到 FLASH 芯片的应答信号。可能是 FLASH 芯片有问题

### 编程失败。不支持这种芯片

原因：选择的编程算法有错

### 闪存擦除失败

原因：未能成功擦除 FLASH

### 编程算法文件错误

原因：所指定的编程算法有问题。可能是软件未安装好或是用户自己编写的编程算法不对

### 编程算法类型不对

原因：驱动程序不支持该种编程算法类型

### 闪存编程失败

原因：FLASH 编程失败

### 闪存校验失败

原因：FLASH 校验失败

### 将算法程序调入 RAM 时出错

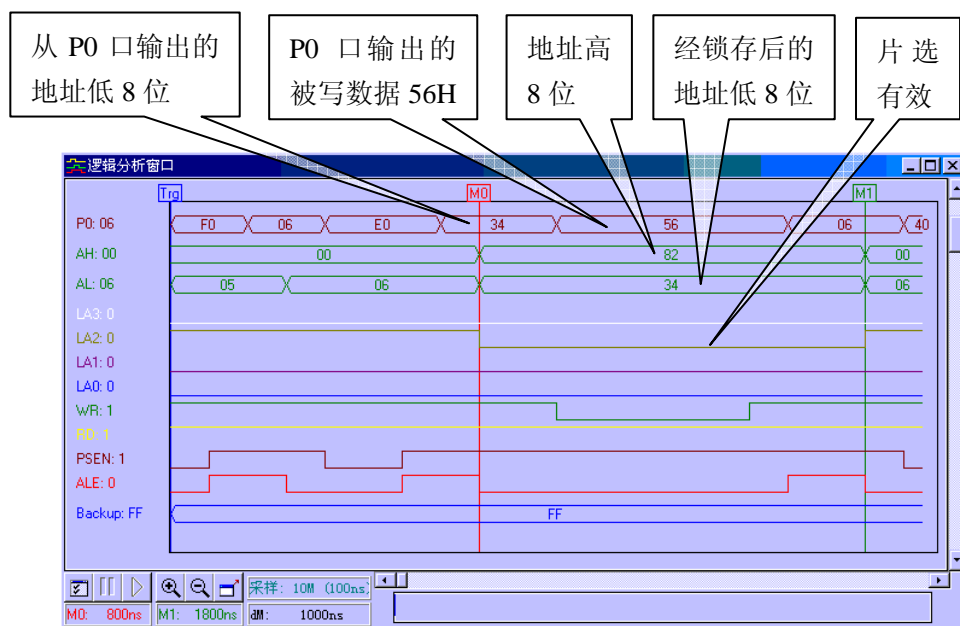
原因：将编程算法调入 RAM 时出错。可能是设定的编程算法的 RAM 地址及大小出错

## 第五章 逻辑分析工具

### 一、逻辑分析仪

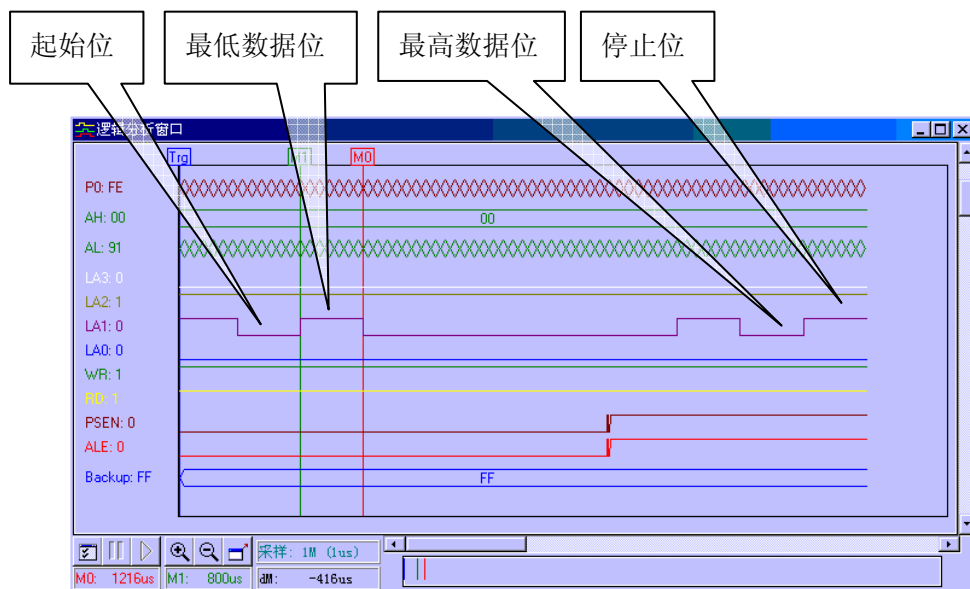
本实验仪带有 32 路逻辑分析仪功能，可以采样 4 路外部波形，28 路内部波形，让学生直观地看到电路的工作时序，增加感性认识。外部 4 路逻辑波形由实验仪上的 LA0..LA3 接入，信号来源由学生选择，任意接到实验仪上所要观察的数字信号，内部 28 路逻辑波形分别为 8 路数据信号，16 路地址信号，还有 RD、WR、PSEN、ALE 4 路控制信号。当用户程序运行时，逻辑分析就采样信号，当程序停下来时，仿真板将采样到的数据传到系统机上并显示出来。

下图是单片机在读写静态存储器（SRAM）时，采样到的波形。由于要写数据地址为 08234H，所以要将 RAM CS 接到 CS0 上，将 LA2 接到 RAM CS 上，以便观察，采样速率为 10M/S。当程序将数据 56H 写到 SRAM 地址 08234H 时，地址高 8 位从 P2 口（80C196 为 P4 口）输出，地址最高 3 位译码使 CS0 为低（见 LA2 信号），选中 SRAM 芯片，地址低 8 位由 P0 口（80C196 为 P3 口）输出，经过 ALE 锁存，做为地址低 8 位，然后从 P0 口（80C196 为 P3 口）输出被写数据 56H，在数据有效时，WR 信号变低，将数据写入 SRAM。完成一次写操作。



当程序从 SRAM 读数据时，时序与写数据相似，地址高 8 位从 P2 口（80C196 为 P4 口）输出，地址最高 3 位译码使 CS0 为低，选中 SRAM 芯片，地址低 8 位由 P0 口（80C196 为 P3 口）输出，经过 ALE 锁存，做为地址低 8 位，然后 P0 口（80C196 为 P3 口）浮空，等待数据，RD 信号变低，将数据从 SRAM 中读到 P0 口（80C196 为 P3 口）。CPU 将此数据送到寄存器，完成一次读操作。

下图是单片机在通信时采样的波形，采样速率为 1M/S。以 2400 波特率发送字符 ‘A’，将 LA1 接到仿真板的 TXD 端，在程序中检测到 TI 后，延时一段时间再停止，以便采样到信号的停止位。

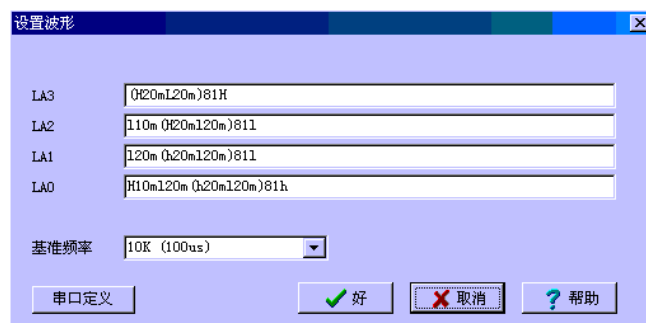


从采样到的波形可以看出，在 2400 波特率时，单个脉冲的宽度为 416uS。字符 ‘A’ 的二进制为 ‘0100 0001’，从上图可以看到串口通信时，产生信号的全过程，先发一个低脉冲做为串行信号的起始位，然后是数据的最低位，从低位到高位逐个发送数据位，最后再发一个高脉冲做为串行信号的停止位。完成一个串行通信的发送过程。

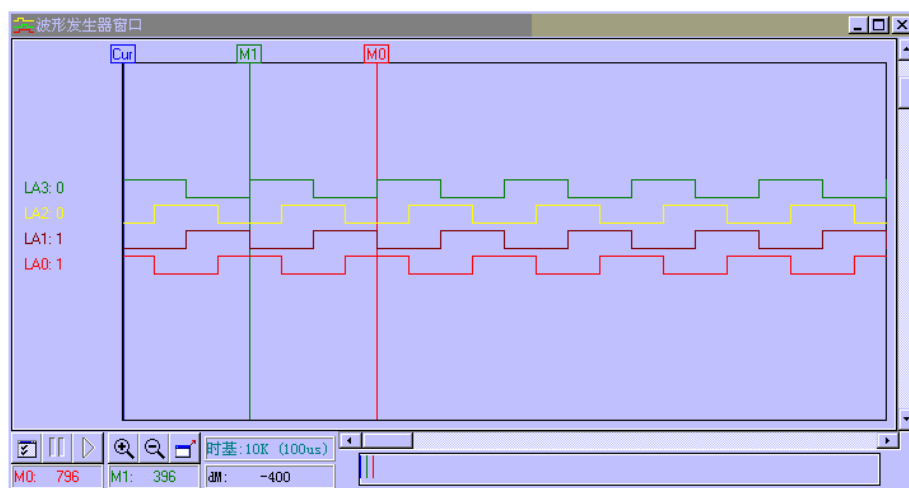
## 二、波形发生器

Lab8000 实验仪带有一个 4 路的波形发生器，可用来产生可编程的数字信号。数字信号通过实验仪上的 LA0..LA3 输出。可参见《伟福仿真器使用说明书》上波形发生器的使用方法。

下图是用波形产生脉冲信号序列，LA3 接到步进电机的 A 相，LA2 接步进电机的 B 相，LA1 接步进电机的 C 相，LA0 接步进电机的 D 相。经驱动后控制步进电机的运转，采用双四拍控制方式，循环控制。波形定义说明如下：



LA3 的定义(h20m 120m)81 h, 表示产生 (高 20ms 低 20ms) 的脉冲 81 个, 然后保持高。直到循环。LA2 的定义 110m(h20m 120m)81 l, 表示产生 低电平 10ms 再产生 (高 20ms 低 20ms) 脉冲 81 个, 然后保持低。直到循环。LA1、LA0 的定义依此类推。由此定义产生波形如下:



将波形发生器产生的信号接到步机的 ABCD 相上, 驱动后控制电机的运转。

### 三、程序跟踪器

本实验还提供了强大的程序跟踪器功能, 让用户在程序全速运行时, 捕捉程序运行轨迹, 了解程序指令的执行流程。右图是在执行“软件实验五 程序跳转表”跟踪器捕捉到程序运行轨迹。

```

• void Func0() {}
• void Func1() {}
• void Func2() {}
• void Func3() {}

void FuncEnter(unsigned char FuncID)
{
    switch (FuncID) {
        case 0: Func0(); break;
        case 1: Func1(); break;
        case 2: Func2(); break;
        case 3: Func3(); break;
        default: break;
    }
}

void main()
{
    FuncEnter(0);
    FuncEnter(1);
    FuncEnter(2);
    FuncEnter(3);
    while(1);
}

```

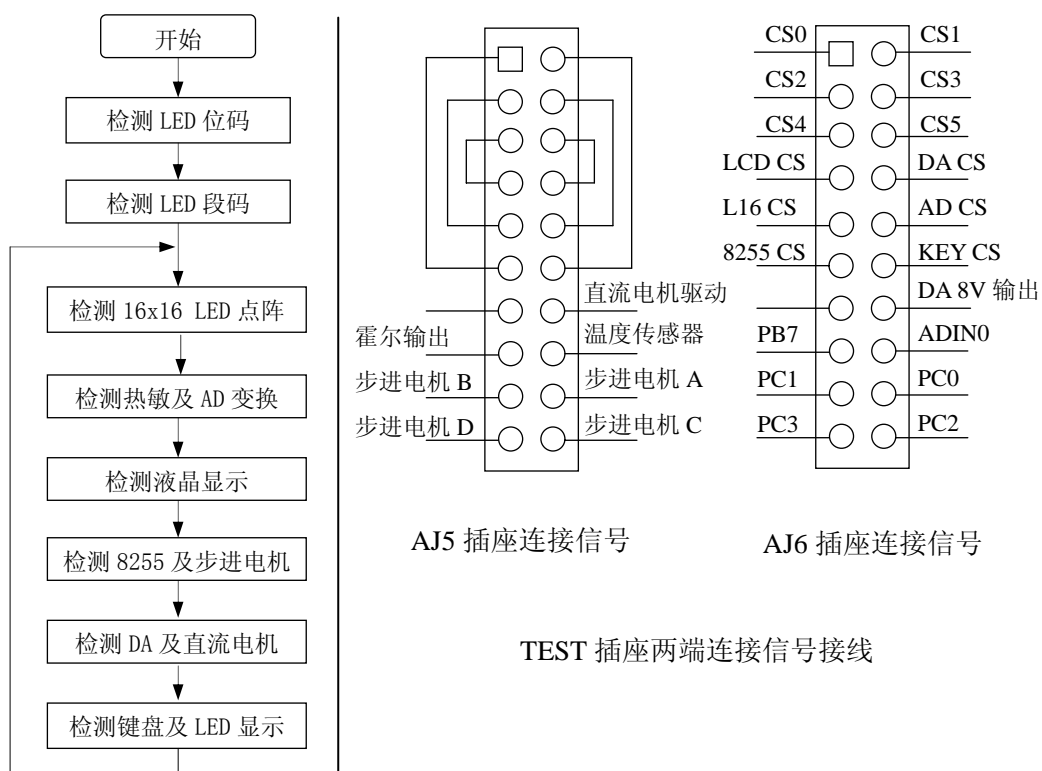
帧号	反汇编	源程序	文件名
0	001EH E4 CLR A	FuncEnter(0);	S5.C
1	001FH FF MOV R7, A		
2	0020H 120003 LCALL 0003H		
3	0003H EF MOV A, R7	switch (FuncID) {	S5.C
4	0004H 14 DEC A		
5	0005H 600D JZ 0014H		
6	0007H 14 DEC A		
7	0008H 600D JZ 0017H		
8	000AH 14 DEC A		
9	000BH 600D JZ 001AH		
10	000DH 2403 ADD A, #03H		
11	000FH 700C JNZ 001DH		
12	0011H 02003D LJMP 003DH	Func0();	S5.C
13	003DH 22 RET	void Func0() {}	S5.C
14	0023H 0F INC R7	FuncEnter(1);	S5.C
15	0024H 120003 LCALL 0003H		
16	0003H EF MOV A, R7	switch (FuncID) {	S5.C
17	0004H 14 DEC A		
18	0005H 600D JZ 0014H		
19	0014H 02003E LJMP 003EH	Func1();	S5.C
20	003EH 22 RET	void Func1() {}	S5.C
21	0027H 0F INC R7	FuncEnter(2);	S5.C
22	0028H 120003 LCALL 0003H		
23	0003H EF MOV A, R7	switch (FuncID) {	S5.C
24	0004H 14 DEC A		
25	0005H 600D JZ 0014H		
26	0007H 14 DEC A		
27	0008H 600D JZ 0017H		
28	0017H 02003F LJMP 003FH	Func2();	S5.C
29	003FH 22 RET	void Func2() {}	S5.C
30	002BH 0F INC R7	FuncEnter(3);	S5.C
31	002CH 120003 LCALL 0003H		
32	0003H EF MOV A, R7	switch (FuncID) {	S5.C
33	0004H 14 DEC A		
34	0005H 600D JZ 0014H		
35	0007H 14 DEC A		
36	0008H 600D JZ 0017H		
37	000AH 14 DEC A		
38	000BH 600D JZ 001AH		
39	001AH 120040 LCALL 0040H	Func3();	S5.C
40	0040H 22 RET	void Func3() {}	S5.C
41	001DH 22 RET	}	S5.C

## 第六章 系统自动检测功能

本实验系统带有自动检测功能，可以检测实验仪上各种电路的工作是否正常，这些电路包括 A/D 变换电路、D/A 变换电路、直流电机控制、步进电机控制、8255 控制、16x16LED 点阵显示电路、液晶显示电路、温度传感器、键盘/LED 显示电路。

启动自检功能的方法是：1) 关闭电源。2) 随机附带的 20 芯自检电缆的两端插到两个 TEST 插座 AJ5 和 AJ6 中。TEST 两端连接的信号如下图显示。3) 按住八段数码管上方的 TEST 按钮。4) 开启电源。5) 等到 LED 最右边无显示后，松开 TEST 键。系统就进入了自动检测状态。

在自检状态，程序流程如下：



系统自检步骤:

1. LED 一位一位地显示“8”字,如果有一位不显示,检测相应位的输出,及位驱动是否正确。
2. 然后再显示八段码的各段,如果所有位的一段不亮,可能段输出时有错,如果是只有一位的一段不亮可能是 LED 管坏。
3. 在热敏及 AD 变换检测时,会在 LED 的左边显示温度值,因为是负温度系数传感器,所以当温度升高,显示数值会减小。可以用手抓紧温度传感器,使其升温,观察显示是否有变化。如果温度显示不变,检测热敏是否有输出及 A/D 变换电路是否工作正常。
4. 检测液晶显示时,会在液晶上显示一条竖线,并自右往左扫描,在动态扫描时,显示的竖线会较淡,如果显示不对,可以按复位键,对液晶屏复位。或换液晶屏,检查其是否损坏。如果读不到液晶返回的信号,会在 LED 的第四位显示‘-’。
5. 检测 8255 及步进电机时,由 8255 输出步进电机的驱动脉冲。如果电机不转,检查 8255 的 PC0~PC3 是否有输出及驱动块是否损坏。
6. 检测 DA 及直流电机时,由 DA 输出直流电压,经驱动后,加在直流电机上,电机旋转后,霍尔元件会输出脉冲,经 8255 读入。如果 8255 读不到霍尔输出的脉冲,会在 LED 的第五位显示‘-’,此时应检查电机是否运转,霍尔元件是否损坏,8255 工作是否正常。
7. 检测键盘及 LED 显示,在键盘上按下某键,会在 LED 的最右边显示该键值。如果显示不对,观察键盘是否被锁住。
8. 16x16 LED 点阵滚动显示“南京伟福实业有限公司”字样及一条竖线。观察是否有缺点,缺行,缺列的现象。16x16 点阵由四块 8x8 点阵组成,如果有一块显示乱码或不显示,可能是这块点阵损坏或插错方向。

**伟福®**

**伟福 Lab6000 系列  
单片机仿真实验系统**

**使用说明书**

**南京伟福实业有限公司**



