

单片机原理及应用 实验指导书

前 言

本实验指导书共介绍了 12 个应用实验，每个实验都进行了原理说明，阐述了实验过程。所有实验都在伟福单片机实验开发系统 LAB2000 上进行。实验指导书最后给出了实验报告的基本要求和书写格式，希望在每次完成实验之后按照要求认真书写实验报告，这样不仅可以帮助回忆实验过程，也为今后实际工作打下较为坚实的基础。

目 录

第一章 单片机仿真系统简介

- 1.1 仿真实验系统的特点与基本组成
- 1.2 系统的安装和启动

第二章 8051 系列单片机实验

2.1 实验一 单片机仿真实验系统基本操作

- 2.1.1 实验目的与要求
- 2.1.2 实验内容
 - 1 存储器块清零程序
 - 2 上机操作及调试步骤
 - 3 二进制数转换成二 - 十进制数 (BCD 码) 程序
 - 4 二进制数转换成 ASCII 码程序
 - 5 数据排序程序

复习思考题

2.2 实验二 8051 定点数运算程序设计

- 2.2.1 实验目的与要求
- 2.2.2 实验内容
 - 1 二进制定点数加减运算程序
 - 2 二进制定点数乘法运算程序

复习思考题

2.3 实验三 8051 单片机 P1 口应用实验

- 2.3.1 实验目的与要求
- 2.3.2 8051 并行 I/O 口简介
- 2.3.2 实验内容
 - 1 采用 P1 口作输出
 - 2 采用 P1 口作输入和输出
 - 3 采用 P1 口控制继电器
 - 4 采用 P1 口控制音频电路发声

复习思考题

2.4 实验四 外部中断实验

- 2.4.1 实验目的与要求
- 2.4.2 8051 单片机中断系统简介
- 2.4.3 实验内容
 - 1 采用外部中断 INT0 触发 P1.0 的输出状态反转
 - 2 采用外部中断模拟交通信号灯控制

复习思考题

2.5 实验五 定时/计数器实验

2.5.1 实验目的与要求

2.5.2 8051单片机定时/计数器简介

4.5.3 实验内容

1 用T0中断方式定时,每秒钟使P1.0输出状态发生一次反转

2 用T0计数外部脉冲,通过P1口显示计数值

复习思考题

2.6 实验六 8255输入输出实验

2.6.1 实验目的与要求

2.6.2 8255可编程芯片简介

4.6.3 实验内容

利用8255方式0,实现PA口作输出,PB口作输入

复习思考题

2.7 实验七 D/A转换实验

2.7.1 实验目的与要求

2.7.2 D/A转换芯片0832简介

2.7.3 实验内容

利用0832产生不同波形输出

复习思考题

2.8 实验八 A/D转换实验

2.8.1 实验目的与要求

2.8.2 ADC0809 简介

2.8.3 实验内容

利用 0809 实现 A/D 转换并从 LED 上显示转换结果

复习思考题

2.9 实验九 键盘显示实验

2.9.1 实验目的与要求

2.9.2 数码管显示及键盘扫描工作原理

2.9.3 实验内容

1 数码管动态显示

2 键盘扫描显示

复习思考题

2.10 实验十 电子时钟实验

2.10.1 实验目的与要求

2.10.2 实验内容

利用定时器中断实现电子时钟

复习思考题

2.11 实验十一 串行口通讯实验

2.16.1 实验目的与要求

2.16.2 8051 单片机串行口简介

2.16.3 实验内容

通过串行口实现双机通信

复习思考题

2.12 实验十二 模拟空调温度控制实验

2.12.1 实验目的与要求

2.12.2 实验原理说明

2.12.3 实验内容

利用电位器和 A/D 转换器实现模拟空调温度控制

复习思考题

附录 实验报告基本要求与书写格式

第一章 伟福单片机仿真实验系统简介

1.1 仿真实验系统的特点与基本组成

伟福单片机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。本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。实验仪上有丰富的实验电路和灵活的组成方法。这些电路既可以和 8031 系列、也可以和 80C196 系列 CPU 及 8088 / 8086CPU 组合完成各种实验。本实验仪将高档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移植到了过来，让学生在实验时不仅能知道软件的执行过程，也能直观地看到程序运行时，电路上的信号状态，工作时序，非常详细地了解电路的工作情况。仿真实验系统具有三种使用方法：

- (1) 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 PC 机，仅用实验仪的板上仿真器进行仿真和实验。
- (3) 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。

配 MCS51/196 仿真板，可进行 8051 或 80C196 的实验，配 8088 / 8086 仿真板，可以进行 8088 / 8086 实验。实验系统自带键盘和显示器，自带系统监控程序。如果没有 PC 机也同样进行各种学习和实验。配备 PC 机集成调试软件，在有 PC 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。

PC 机软件具有全集成化仿真环境，中、英文两种界面，软件仿真与硬件仿真两种模式，软件仿真可以在无仿真仪的情况下进行。实验仪可以方便灵活地构成各种实验方案，在有 PC 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，板上也提供了 DIP40 / 28 / 24 / 20 / 16 / 14 插孔和 CPU 的地址数据总线引出插孔，供学生自己扩展其它实验，培养实际动手能力，加强对实验电路的理解。

伟福实验系统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线就可进行，以减轻学生的工作量，同时也提供了一些需较多连线的扩展性实验以进一步锻炼学生的动手能力。此外，它还为学生们提供了强大的软硬件调试手段。

伟福仿真实验系统上各功能模块的排列如图 1.1 所示。各个模块的引脚都被引出到相应的插孔，扩展插座为 40 芯通用插座，可兼容 28 芯、24 芯、16 芯和 14 芯插座。利用这个插座，可对双列直插式的各种微机芯片进行实验。总线扩展为三排插孔，用于引出各种总线信号，包括 8 根数据总线，16 根地址总线，以及 ALE，RD，WR 等控制总线，另外与 CPU 相关的一些控制信号和 I/O 信号例如 P1 口、RXD、TXD 等信号在仿真板上引出。

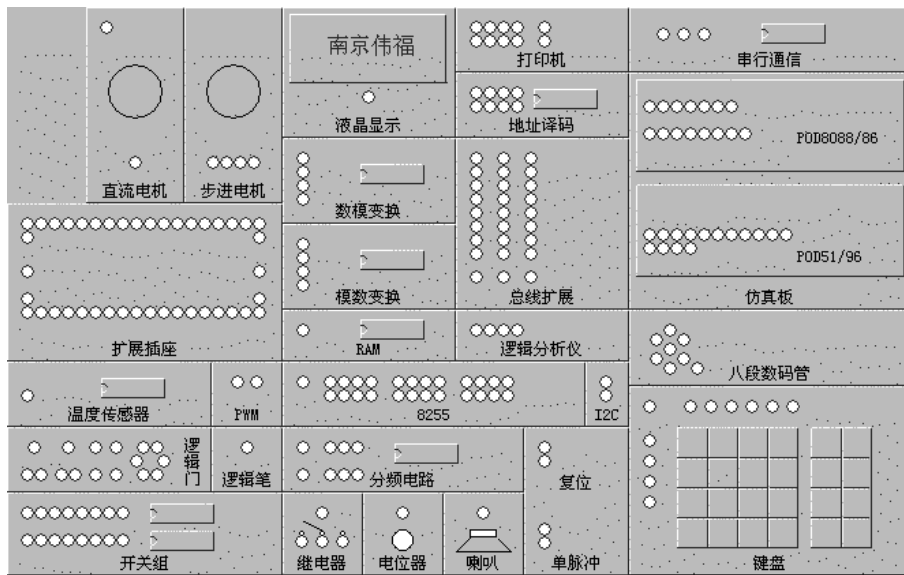


图 1.1 伟福仿真试验系统的功能模块排列

图中的圆点●表示仿真实验系统上的连线插孔，每个插孔旁边都有相应的文字标识，进行不同实验的时候，需要根据要求用专门的连线将它们连接起来。注意：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。实验线路连接完成后，应仔细检查无误后再接通电源。

在进行不同实验的时候可能需要采用不同译码地址，地址插孔的地址分配如表 1-1 所示。

表 1-1 地址译码插孔的地址分配表

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

1.2 系统的安装和启动

1. 在 PC 机上安装伟福仿真开发系统集成调试软件。
2. 进行 8051 单片机实验时，应插上 POD51 仿真板，并插上 8051 或 8052 CPU。
3. 将配套的串行通讯电缆的一端与实验仪上的“仿真器串口”9 芯 D 形插座相连，另一端与 PC 机的串行口相连。

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。仿真开发器初始化成功后,LED 会显示 8051,表示仿真系统正常。
6. 打开 PC 计算机电源,执行伟福(WAVE)集成调试软件。集成调试环境界面如图 1.2 所示。

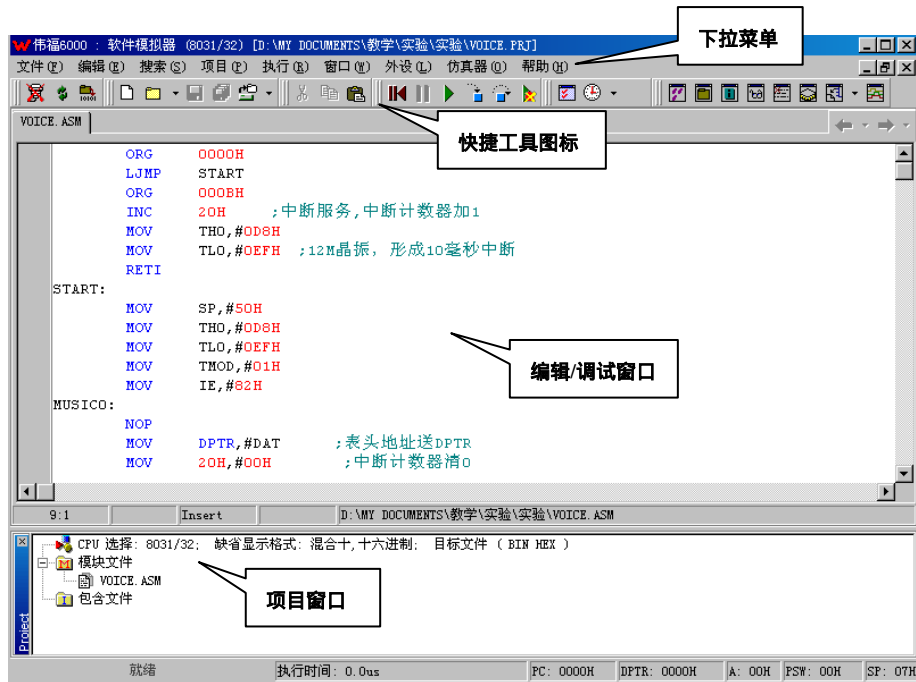


图 1.2 伟福(WAVE)集成调试环境

7. 通过下拉菜单“仿真器”对仿真实验系统进行设置,如图 1.3 所示。
 选择仿真器:伟福 Lab2000P 实验系统。
 选择仿真头:MCS51 实验。
 选择 CPU:(8031 / 32)。

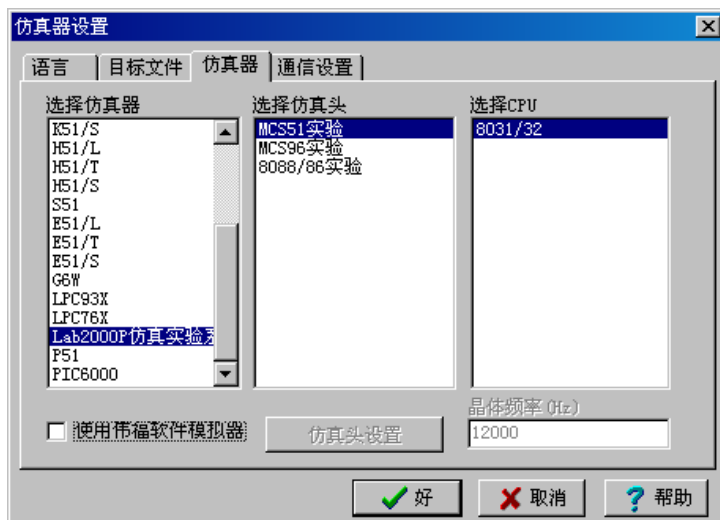


图 1.3 设置仿真实验系统

通过仿真器设置中的“语言”栏可以设置用户所采用的编程语言，如图 1.4 所示。采用汇编语言编程时，应选中“伟福汇编器”复选框，采用 C51 高级语言编程时，应选中“Keil C”复选框。

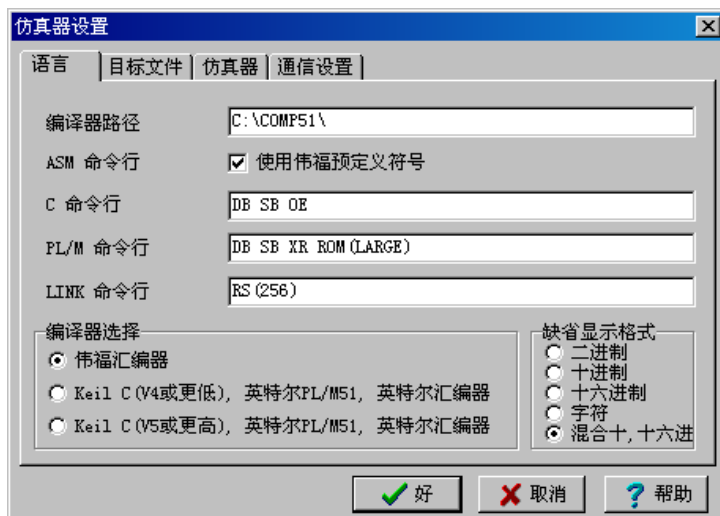


图 1.4 设置用户编程语言

通过仿真器设置中的“目标文件”栏可以设置最终生成的用户目标文件，一般可以按图 1.5 进行设置。

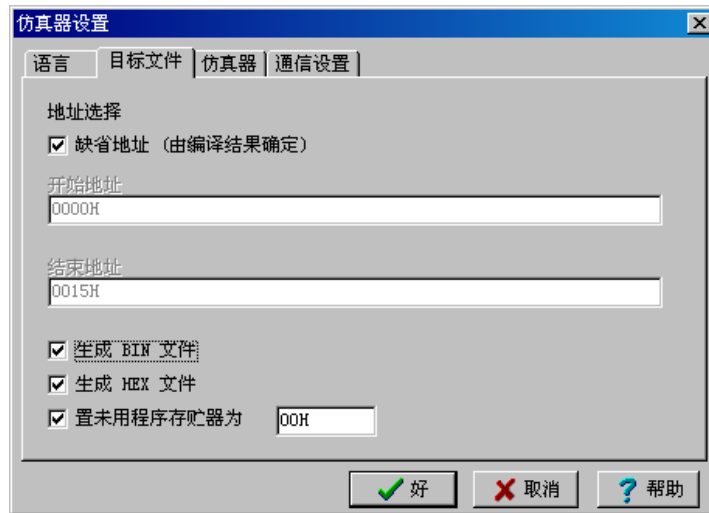


图 1.5 设置目标文件

通过仿真器设置中的“通信设置”栏可以设置集成调试环境与 PC 机的通信端口和波特率，如图 1.6 所示。

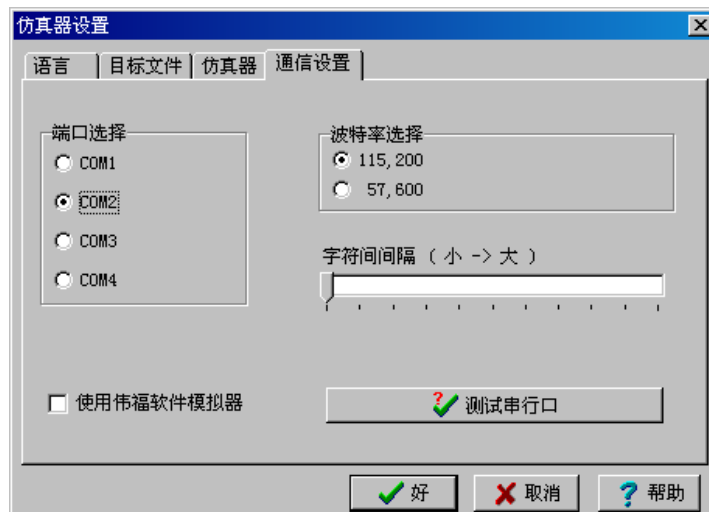


图 1.6 设置通信方式

第二章 8051 系列单片机实验

2.1 实验一 单片机仿真实验系统基本操作

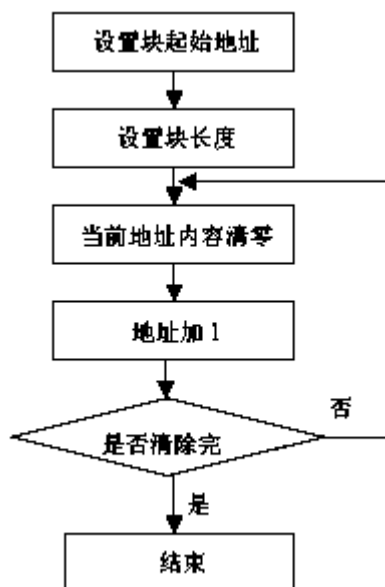
2.1.1 实验目的与要求

通过本实验，了解熟悉伟福单片机开发实验系统软、硬件的基本操作，同时学习简单的 8051 汇编语言程序的编写和调试方法。熟悉伟福单片机开发实验系统软、硬件，掌握伟福集成调试环境的使用方法。


2.1.2 实验内容

1. 存储器块清零程序

指定存储器中某块的起始地址和长度，编写程序将指定存储器中内容清零。程序参考流程如下：



2. 上机操作及调试步骤

(1) 启动伟福集成调试软件后，单击快捷工具栏中的“新建文件”图标 (或按 Alt+F+N 键)，在弹出的编辑窗口中键入上述源程序文件，如图 2.1 和图 2.2 所示。

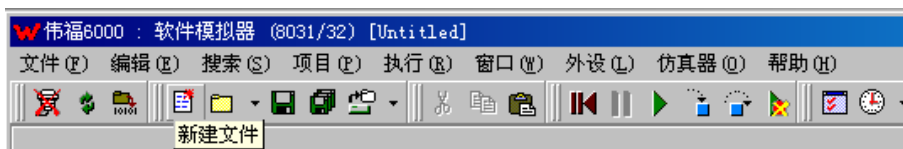


图 2.1 创建新文件

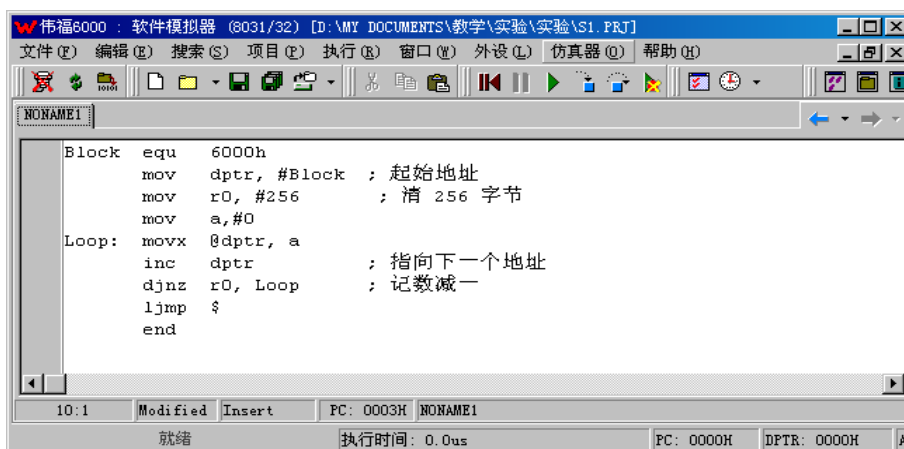



图 2.2 在编辑窗口中键入源程序文件

(2) 单击快捷工具栏中的“保存文件”图标  (或 F2 键)，在弹出的保存文件窗口中选定保存源程序的文件夹，并给定源程序文件名(汇编语言源程序文件的扩展名为“.ASM”)，单击“保存”按钮，如图 2.3 和 2.4 所示。

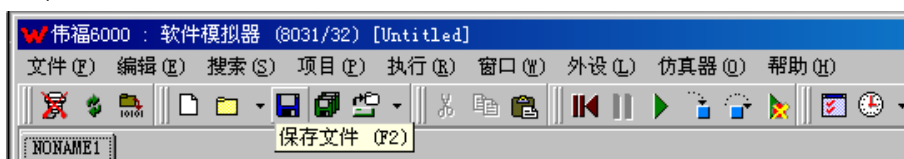


图 2.3 保存源程序文件

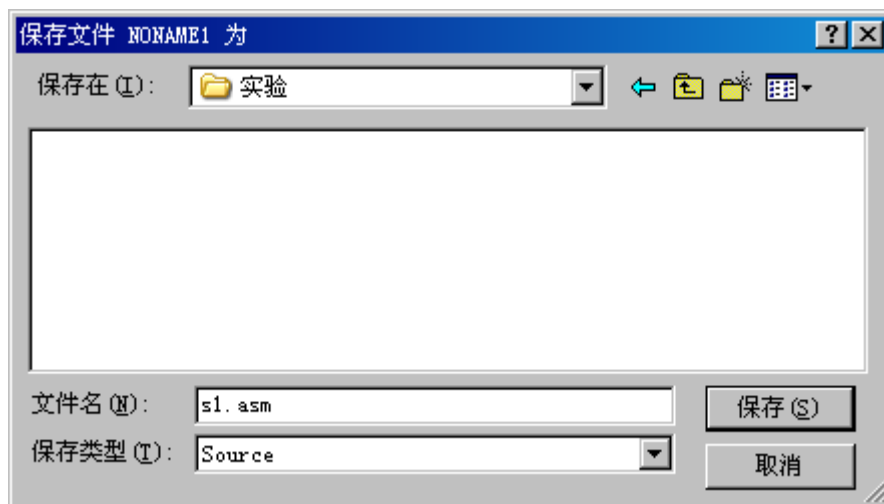


图 2.4 选定保存文件夹并给定文件名

(3) 单击下拉菜单“文件”中的“新建项目”，如图 2.5 所示，项目窗口如图 2.6

所示。

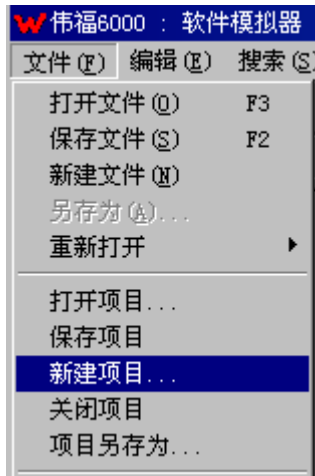


图2.5 新建项目

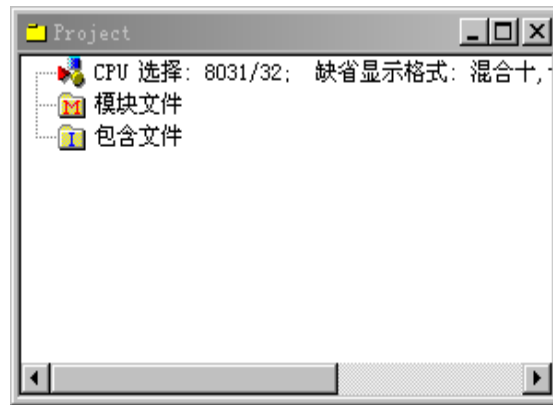


图2.6 项目窗口

新建项目会自动分如下三步：

加入模块文件。在加入模块文件的对话框中选择刚才保存的文件S1.ASM,按所示。打开键。如果是多模块项目，可以同时选择多个文件再打开，如图2.7所示。

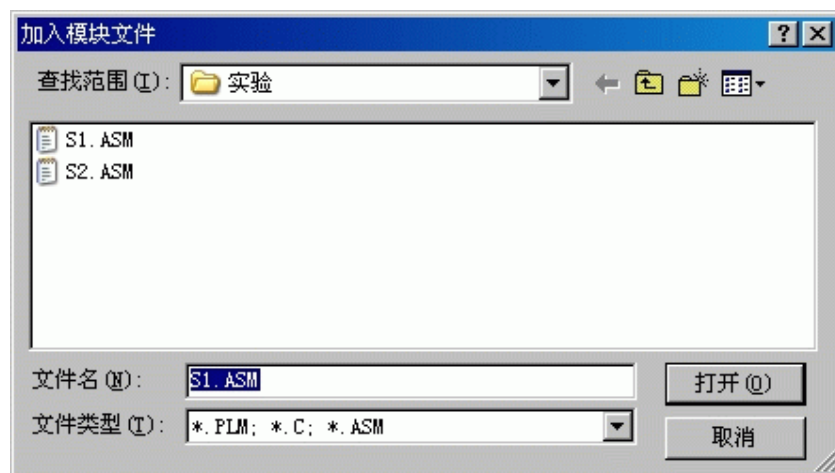


图 2.7 加入模块文件窗口

加入包含文件。在加入包含文件对话框中，选择所要加入的包含文件（可多选）。包含文件通常用于多模块或高级语言编程中，如果没有包含文件，按取消键，如图2.8所示。

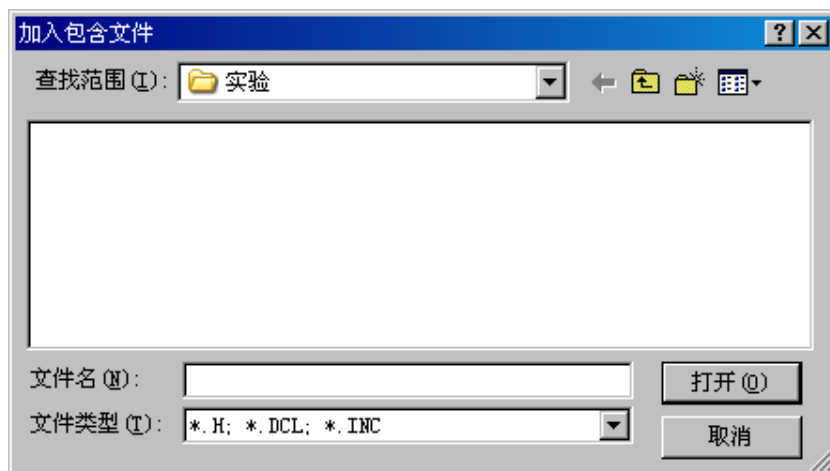


图 2.8 加入包含文件窗口

保存项目。在保存项目对话框中输入项目名称“S1”，无须加后缀。软件会自动将后缀设成“.PRJ”。按保存键将项目保存到与源程序相同的文件夹下，如图2.9所示。

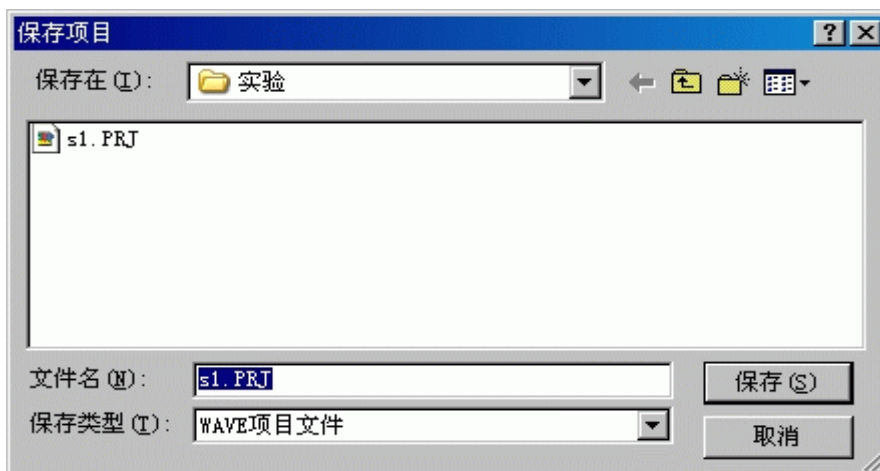


图 2.9 保存项目窗口

项目保存好后，如果项目是打开的，可以从项目窗口看到该项目中的“模块文件”。对于上例，可以看到项目中已经有了一个模块文件“S1.ASM”，如图2.10所示。

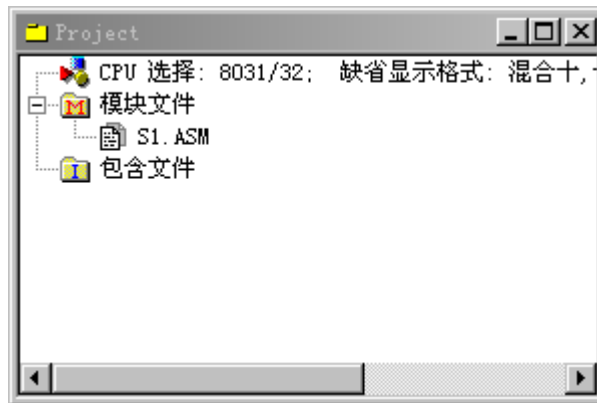



图 2.10 加入模块文件后的项目窗口

(4) 单击快捷工具栏中的“编译”图标  (或 F9 键), 对刚才保存的项目进行编译和连接, 如图 2.11 所示。在编译之前, 软件会自动将项目中的源程序文件存盘。如果发生编译连接错误, 则在“信息窗口”中报告如图 2.12 所示编译连接出错信息。此时可将鼠标指向“信息窗口”中的错误行并双击鼠标左键, 光标将自动跳转到编辑窗口中源程序的相应出错位置, 便于用户修改。修改源程序文件并保存之后, 重新进行第(3)步。如果编译连接通过, 将在“信息窗口”中报告如图 2.13 所示编译连接正确信息。此时就可以对源程序进行调试了。

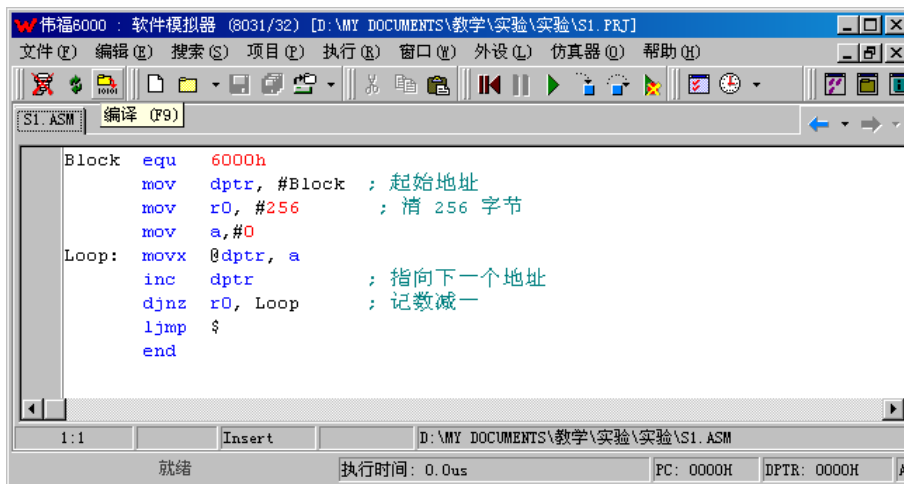


图 2.11 对已保存的项目进行编译连接

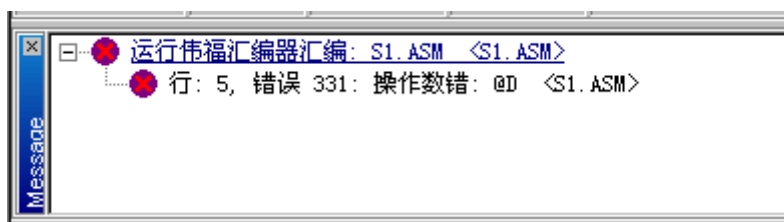


图 2.12 编译连接出错信息

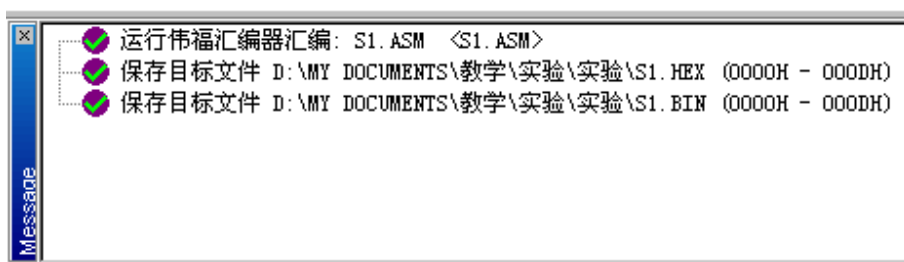



图 2.13 编译连接正确信息

(5) 单击快捷工具栏中的“跟踪”图标或按F7键启动单步跟踪调试程序，如图 2.14所示。单步跟踪就是一条指令一条指令地执行，若有子程序调用，也会跟踪到子程序中去。这种方式可以观察程序每一条指令的执行的结果，光标“=>”所指的就是下次将要执行的程序指令。源程序窗口最左边的“o”代表此行为有效程序，此行在编译过程中产生了可以执行的机器指令。

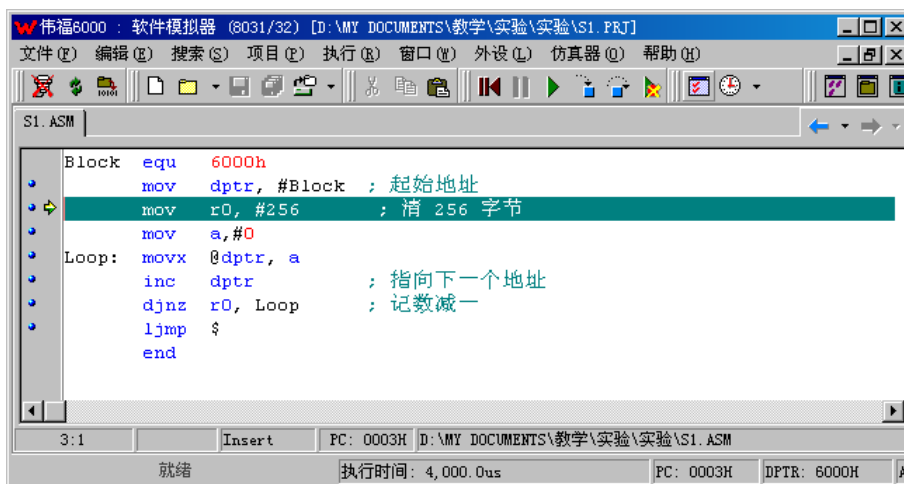






图 2.14 启动单步调试程序

跟踪调试执行速度较慢，也可以采用单步调试(快捷图标为)，此时若有子程序

调用,不会跟踪到子程序中去,而是将子程序一次执行完成。还可以单击快捷图标将

启动程序全速运行,单击快捷图标停止程序运行,单击快捷图标将复位CPU。调试时除了上述快捷图标之外,还可以采用“执行到光标处”的功能,先将鼠标指向程序想要暂停的地方,例如程序中最后一条指令“LJMP \$”,单击鼠标右键,弹出一个右键菜单,选择右键菜单中的“执行到光标处”,程序将全速执行到指定行暂停,如图2.15所示。


右键菜单中还有其他一些功能,用户可以自己通过实践和阅读伟福仿真器的说明书来逐步掌握。下面介绍一下设置断点功能。

将鼠标移到源程序窗口的左边灰色区,光标变成“手指圈”,单击左键设置断点,也可以用右键菜单的“设置/取消断点”功能或用Ctrl+F8组合键设置断点。如果断点有效图标为“红圆绿勾”,无效断点的图标为“红圆黄叉”。

执行到光标处(C)	F4
设置PC	
添加观察项	Ctrl+F7
编辑观察项...	Ctrl+F5
转到指定行(G)...	Ctrl+G
转到指定地址/标号	Ctrl+A
转到当前 PC 所在行(P)	Ctrl+P

图1.15 部分右键菜单

断点设置好后,就可以用全速执行的功能,当程序执行到断点时,会暂停下来,这你可以观察程序中各变量的值,及各端口的状态,判断程序是否正确。

在程序调试过程中单击CPU快捷图标,可以观察CPU的运行状态,此时源程序窗口将出现反汇编(Disassembly)结果,以及工作寄存器和CPU寄存器窗口,如图2.16所示。可以通过这些窗口观察程序运行时CPU内部寄存器的变化情况。请读者在对上述程序进行调试的过程中,学会观察这些窗口中不同寄存器内容的状态,从而判断程序运行是否正常。

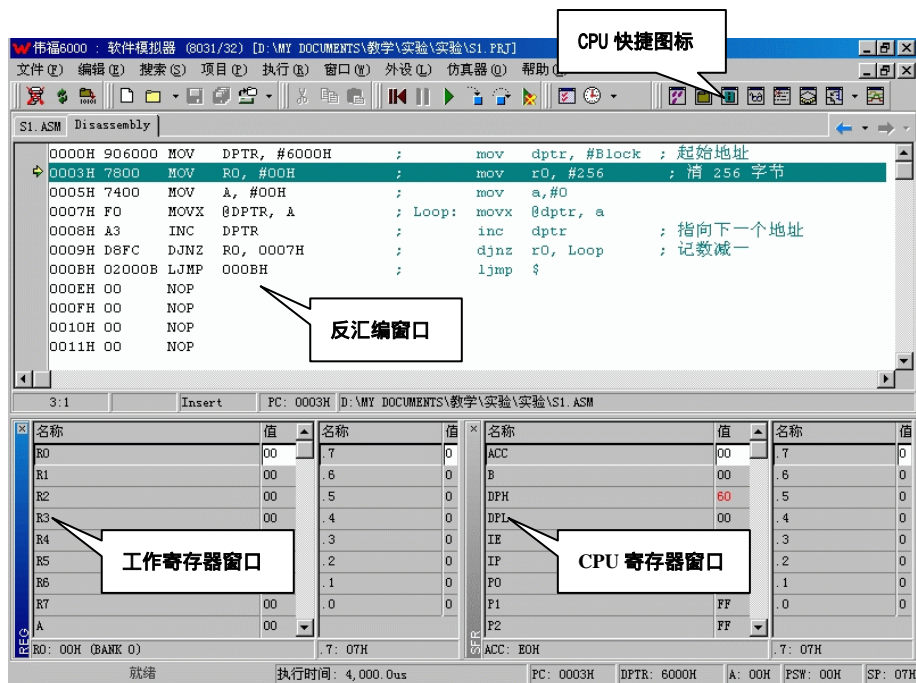



图 2.16 CPU 运行状态观察窗口

伟福集成调试软件中还提供了一个 DATA 快捷图标 ，单击该图标右边的小箭头，出现如图 2.17 所示存储器空间选择菜单。通过该菜单可以分别观察 8051 单片机内部数据存储器空间(DATA)、程序存储器空间(CODE)、外部数据存储器空间(XDATA)、外部数据存储器分页空间(PDATA)以及位寻址空间(BIT)。图 2.18 所示为本实验程序运行后的 XDATA 空间数据变化的情况。

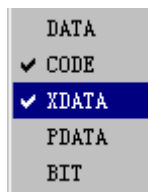


图 2.17 数据空间菜单

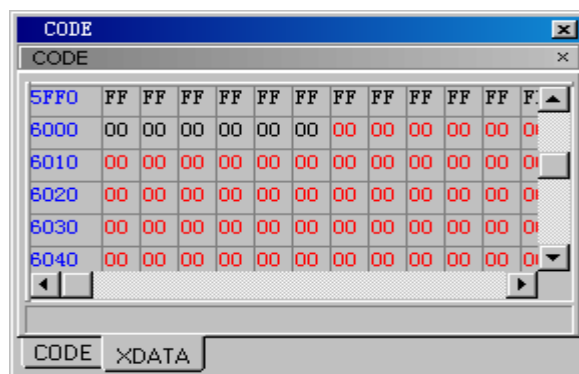


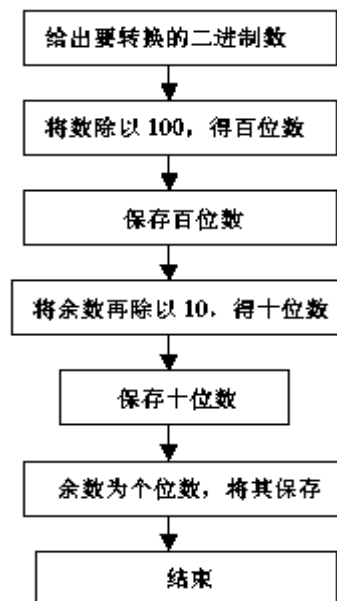
图 2.18 外部数据存储器 XDATA 空间窗口

至此，已经介绍了如何应用伟福集成软件开发环境调试 8051 单片机汇编语言程序的基本方法，读者可以通过实际应用来加深理解，例如若要将从 6000H 开始的 256 个字

节外部数据存储器块的内容置成 11H 值，请修改以上程序完成此操作，注意在程序调试过程中数据的变化情况。

3. 二进制数转换成二 - 十进制数 (BCD 码) 程序

编写程序将累加器 A 中一个给定的二进制数，转换成三个二 - 十进制数(BCD 码)，并存入 Result 开始的三个单元。程序参考流程如下：



通过本实验学习简单的数值转换算法，了解数值的各种表达方法。同时进一步熟悉伟福单片机实验系统以及集成开发软件环境的使用方法。按照前面介绍的仿真调试方法，启动伟福集成开发环境，用“执行到光标处”功能，启动程序运行到最后一条指令处，然后单击图 2.17 存储器空间菜单中的“DATA”选项，打开 8051 内部数据存储器空间，如图 2.19 所示，可以看到已经将 A 中原来的内容“123H”拆为了三个 BCD 码“1”、“2”和“3”，并分别存入了从“Result”(20H)开始的三个单元中。

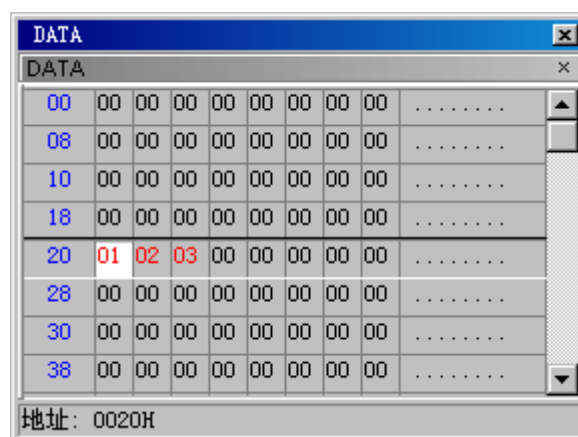
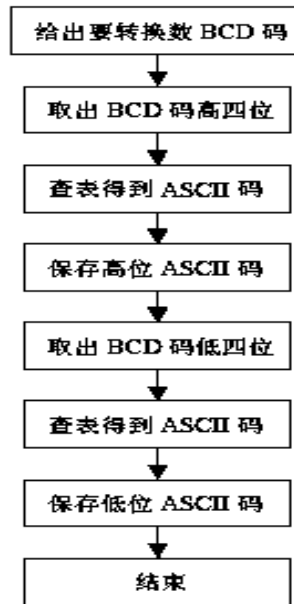


图 2.19 内部数据存储器 DATA 空间窗口

4. 二进制数转换成 ASCII 码程序

此实验主要让学生了解数值的二进制表示和 ASCII 码表示之间的区别，利用查表功能快速地进行数值转换。

编写程序，将累加器 A 中的内容拆为二个 ASCII 码，并存入 Result 开始的二个单元。程序参考流程如下：



按照前面介绍的调试方法，启动伟福集成开发环境，用“执行到光标处”功能，启动程序运行到最后一条指令处，然后单击图 2.17 存储器空间菜单中的“DATA”选项，打开 8051 内部数据存储单元空间，如图 2.20 所示，可以看到已经将 A 中原来的二进制数“01000011”拆为了 2 个 ASCII 码“34”和“33”，并分别存入了从“Result”（20H）开始的 2 个单元中。

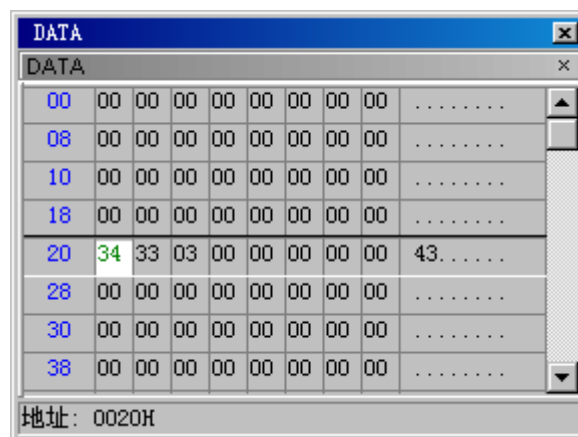
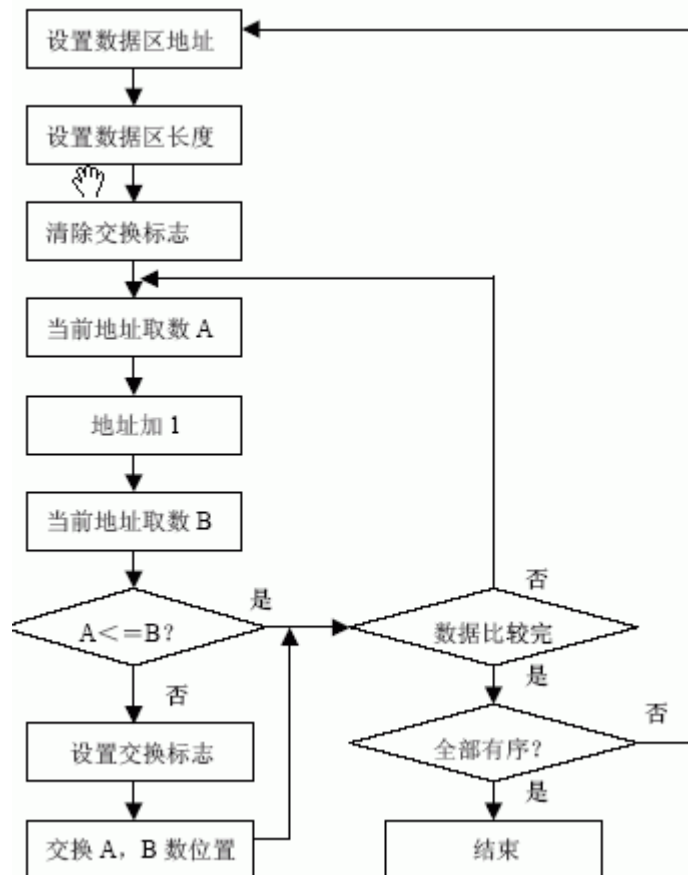


图 2.20 从 DATA 窗口观察程序运行结果

5. 数据排序程序

给出一组随机数，编写程序将此组数据排序，使之成为有序数列。有序的数列更有利于查找。程序参考流程如下：



本程序用的是“冒泡排序”算法，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，直到全部数据有序。通过本实验可以了解数据有序和无序的概念以及数据排序的简单算法。

启动伟福集成开发环境，单击图 2.17 存储器空间菜单中的“DATA”选项，打开 8051 内部数据存储器空间，将鼠标指向其中从 50H 开始的单元，单击右键，将其中内容修改为一组无序的数据，如图 2.21 所示，再用“执行到光标处”功能，启动程序运行到最后一条指令处，可以看到此时数据已经按从小到大的顺序排列了，如图 2.22 所示。



图2.21 未排序的数据

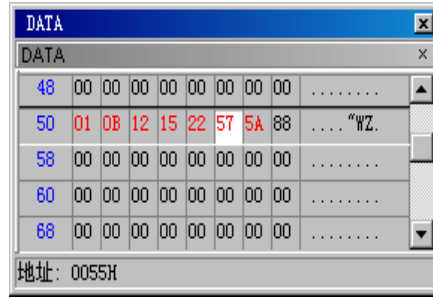


图2.22 已排序的数据

复习思考题

1. 回忆并写出实验上机操作过程。
2. 画出详细的实验程序框图。

2.2 实验二 8051定点数运算程序设计

2.2.1 实验目的与要求

通过本实验，学习编写 8051 单片机定点数运算汇编语言程序，同时进一步熟悉用伟福单片机开发实验系统进行软件调试的方法。熟练掌握伟福集成调试环境的使用方法，编写并调试通过 8051 单片机定点数加减运算程序、乘法运算程序以及除法运算程序，对于子程序调用要学会使用入口和出口参数，同时要学会通过 CPU 窗口和工作寄存器窗口观察程序运行过程以及最后结果。

2.2.2 实验内容

1 二进制定点数加减运算程序。

(R2R3)和(R6R7)中的内容为两个用原码表示的二进制数，最高位为符号位，编写程序将 $(R2R3) \pm (R6R7)$ 的结果送 R4R5。程序中 DADD 为原码加法子程序入口，DSUB 为原码减法子程序入口，出口时 $CY=1$ 发生溢出， $CY=0$ 为正常。程序参考流程如图 2.23 所示。

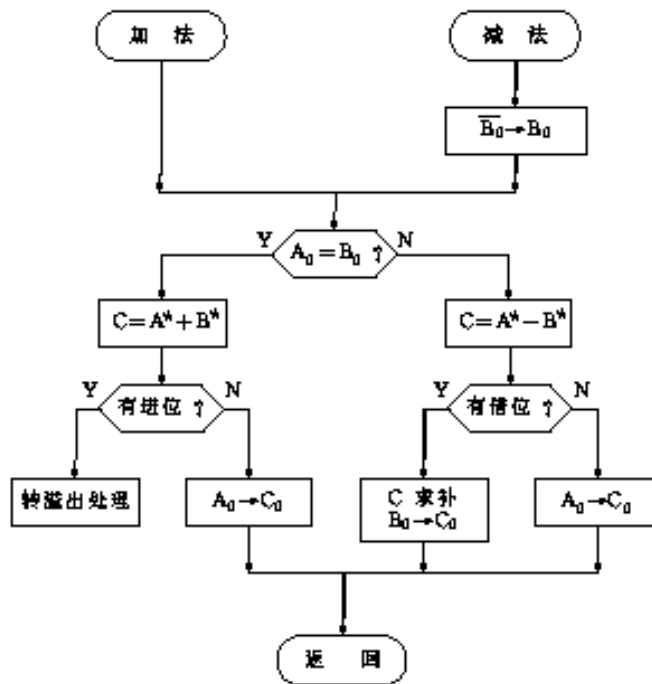


图 2.23 程序参考流程

2 二进制定点数乘法运算程序

编写程序将(R2R3)和(R6R7)两个双字节无符号数相乘，结果送 R4R5R6R7，程序参考流程如图2.24所示。

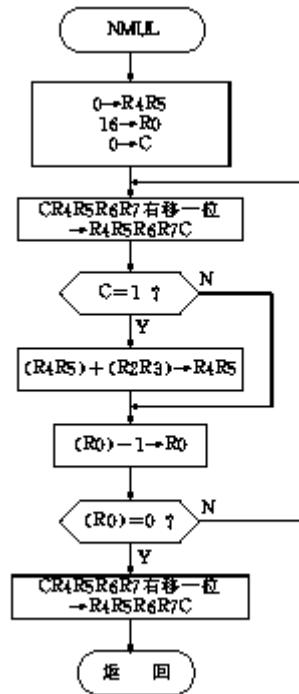


图2.24 程序参考流程

复习思考题

1. 写出实验上机操作过程中如何观察指令的执行状态，如何观察程序的运行结果。
2. 画出详细的实验程序框图。
3. 自己编写一段定点数除法运算程序并通过实验验证其正确性。

2.3 实验三 8051单片机P1口应用实验

2.3.1 实验目的与要求

学习8051单片机P1口作为双向I/O口的使用方法，采用循环指令编写延时子程序。了解用弱电控制强电的基本原理，采用单片机P1口实现继电器开关控制。要求：

采用8051单片机的P1口做输出口，接4只发光二极管，编写程序，使发光二极管循环点亮。

采用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，并将此状态在发光二极管上显示出来。

采用P1口输出电平控制继电器的吸合和断开，从而实现对外部装置的开关量控制。

采用P1口控制音频电路发声。

2.3.2 8051并行I/O口简介

8051 单片机有四个并行 I/O 口，称为 P0、P1、P2、P3，每个口都有 8 根引脚，它们都是双向通道，每一条 I/O 引脚都能独立地用作输入或输出，作输出时数据可以锁存，作输入时数据可以缓冲。P0~P3 这四个 I/O 口的功能不完全相同，它们的负载能力也不相同，P1、P2、P3 都能驱动四个 LS TTL 门电路，并且不需外加电阻就能直接驱动 MOS 电路。P0 口在驱动 TTL 电路时能带动 8 个 LS 型 TTL 门，但驱动 MOS 电路时若作为地址/数据总线，可直接驱动；而作为 I/O 口时，则需外接上拉电阻才能驱动 MOS 电路。图 2.25 所示为 P0~P3 各口中一位的逻辑图。

P0 为三态双向口，它可作为输入输出端口使用，也可作为系统扩展时的低 8 位地址/8 位数据总线使用。P0 口作为双向 I/O 端口使用时，通常要在外部加一个上拉电阻来提高驱动能力。P0 口作为低 8 位地址/8 位数据总线使用时，在 ALE 信号的控制下分时输出低 8 位地址和 8 位数据信号。

P1 口为准双向口，每一位都可以分别定义为输入或输出使用。P1 口作为输入口使用时，有两种工作方式，即所谓“读端口”和“读引脚”。读端口时实际上并不从外部读入数据，而只把端口锁存器中的内容读入到内部总线，经过某种运算和变换后，再写回到端口锁存器。读引脚时才真正地把外部的输入信号读入到内部总线。逻辑图中各有两个输入缓冲器，CPU 根据不同的指令分别发出“读端口”或“读引脚”信号。在读引脚时，为了保证输入正确的外部输入电平信号，首先要向端口锁存器写入一个“1”，使驱动场效应管截止，引脚信号直接加到三态缓冲器，实现正确的读入。否则，端口锁存器中原来状态有可能为“0”，加到输出驱动场效应管栅极的信号为“1”，该场效应管导通，对地呈现低阻抗。这时即使引脚上输入的是“1”信号，也会因端口的低阻抗而使信号变化，使得外加的“1”信号读入时不一定是“1”。正是由于 P1 口在进行输入操作之前需要有这样一个附加准备动作，故称之为“准双向口”。P1 作为输出口时，如果要输出“1”，只要将“1”写入 P1 口锁存器，使输出驱动场效应管截止，输出引脚由内部上拉电阻拉成高电平，输出为“1”。要输出“0”时，将“0”写入 P1 口锁存器，使输出

驱动场效应管导通，输出引脚被接到地，输出为“0”。

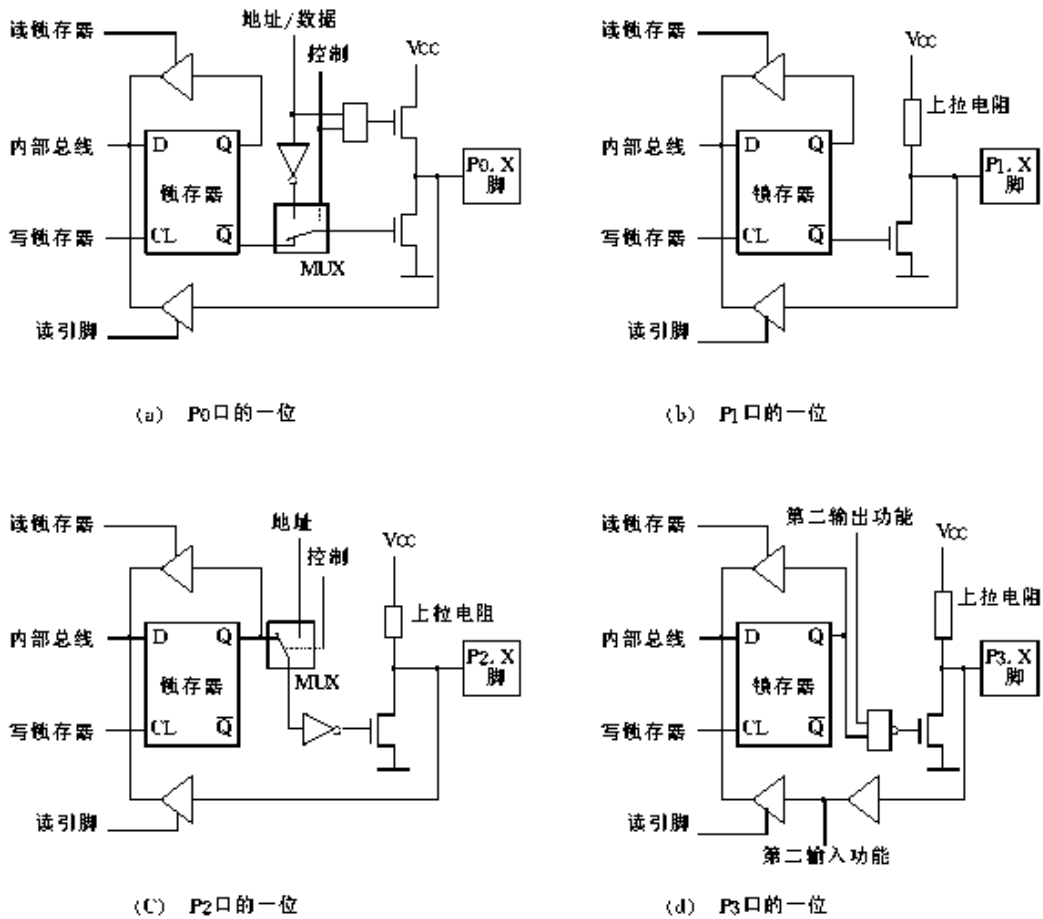


图2.25 8051单片机并行I/O口一位的逻辑图

表 2-1 P3 口的第二功能定义

端口引脚	第二功能
P3.0	RXD (串行输入口)
P3.1	TXD (串行输出口)
P3.2	$\overline{\text{INT0}}$ (外部中断 0 输入)
P3.3	$\overline{\text{INT1}}$ (外部中断 1 入)
P3.4	T0 (定时器 0 外部输入)
P3.5	T1 (定时器 1 外部输入)
P3.6	$\overline{\text{WR}}$ (外部 RAM 写选通)
P3.7	$\overline{\text{RD}}$ (外部 RAM 读选通)

P2 口也是一个准双向口，它有两种使用功能：作为普通 I/O 端口或作为系统扩展时的高 8 位地址总线。P2 口作 I/O 端口使用时，其用法与 P1 口相同。P2 口作外部地址总

线使用时，输出高 8 位地址总线。需要特别指出的是，只要进行了外部系统扩展，由于对片外地址的操作是连续不断的，此时 P0 口和 P2 口就不能再用作 I/O 端口了。

P3 口为多功能口，除了用作通用 I/O 口之外，它的每一位都有各自的第二功能，如表 2-1 所示。P3 口作通用 I/O 口时其使用方法与 P1 口相同，P3 口的第二功能可以单独使用，即不用第二功能的引脚仍可以作通用 I/O 口线使用。

2.3.3 实验内容

1 采用P1口作输出

P1口是准双向口，它作为输出口使用时具有锁存功能。实验所需要LED电平显示电路如图2.26所示。

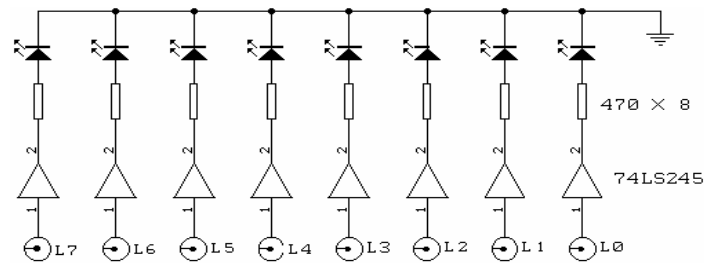


图2.26 电平显示电路

在实验系统上如表2-2所示连线。编写并执行程序，使发光二极管LED循环显示。程序参考流程如下：

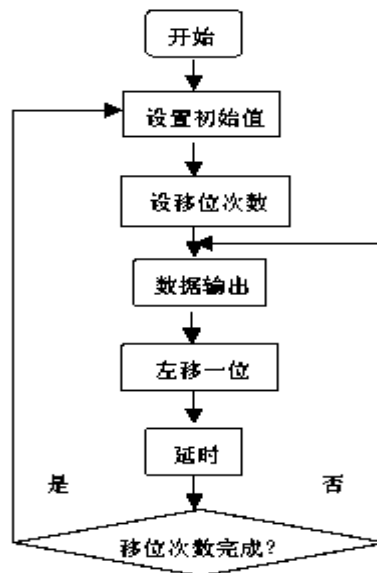


表 2-2 连线说明

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

关于延时子程序的时间计算，查指令表可知 MOV 和 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，所以上面延时子程序该的执行时间为：

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

2 采用P1口作输入和输出

由 P1 口的准双向口结构可知，当作为输入口时，必须先对它置“1”。若不先对它置“1”，读入的数据是不正确的。实验所需要 LED 电平显示电路和逻辑电平开关电路如图 2.26 和图 2.27 所示。

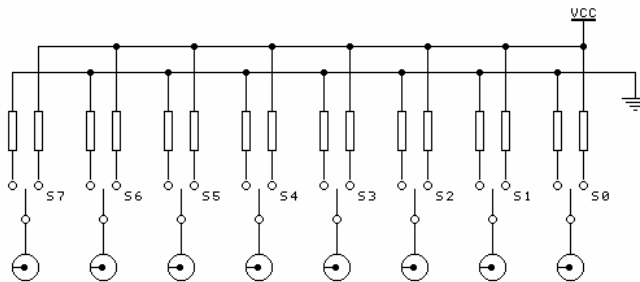


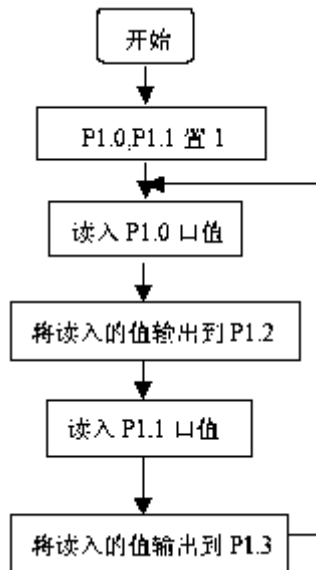
图 2.27 逻辑电平开关电路

在实验系统上如表 2-3 所示连线。编写并执行程序，使发光二极管 LED 随拨动开关的位置而点亮或熄灭。

表 2-3 连线说明

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

程序参考流程如下：



3 采用P1口控制继电器

自动化控制设备中，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和电气电路提供良好的电隔离，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，实验电路如图2.28所示。

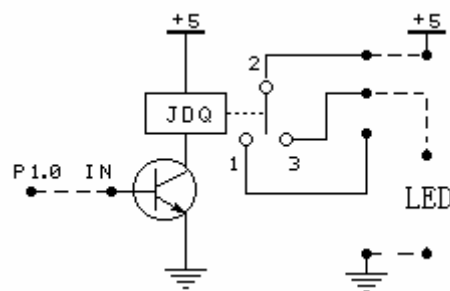


图2.28 继电器控制电路

当控制端P1.0为高电平时，继电器工作常开触点吸合，连接该触点的LED被点亮。当P1.0控制端为低电平时，继电器不工作，常开触点断开，连接该触点的LED熄灭。

在实验系统上如表2-4所示连线。编写并采用单步方式执行程序，当执行P1.0被置1时，LED点亮，执行到P1.0被置0时，LED熄灭。如果全速运行程序，则对应的LED将随继电器的通、断而亮灭。

表2-4 连线说明

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

程序参考流程如下：



4 采用P1口控制音频电路发声

从P1.0输出不同频率的方波信号，经放大滤波后驱动扬声器发声。声音的频率由延时程序控制。实验电路如图2.29所示。

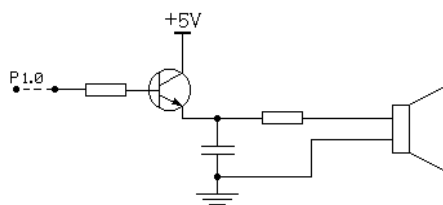


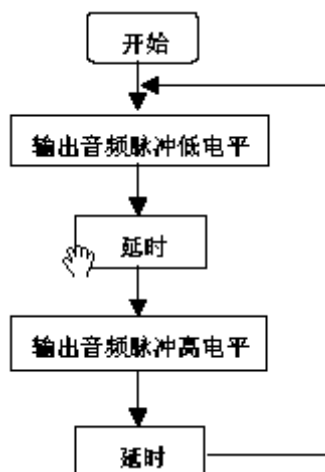
图2.29 音频输出电路

在实验系统上如表2-5所示连线，编写并执行程序使扬声器将发出一定频率的声音。

表2-5 连线说明

连线	连接孔 1	连接孔 2
1	P1.0	扬声器脉冲输入

程序参考流程如下：



复习思考题

1. 复习 8051 单片机并行 I/O 端口特性，P0 ~ P3 口各有什么特点？
2. 为什么说 P1 口是“准双向口”？在实验中用 P1 口输入时如果不先向 P1 口写入“1”而直接读取引脚，会出现什么现象？
3. 如果希望让扬声器发出不同频率，不同长短的声音，应如何修改实验程序？

2.4 实验四 外部中断实验

2.4.1 实验目的与要求

通过本实验加深对8051单片机中断系统的理解，学习外部中断的基本使用方法及相关编程。要求：

用单次脉冲申请中断，在中断处理程序中使P1.0的输出状态发生反转，并通过发光二极管观察P1.0的电平。

模拟交通信号灯控制。

2.4.2 8051单片机中断系统简介

8051 单片机有 5 个中断源，有两个中断优先级，高优先级的中断源可以中断低优先

级的服务程序，反之不行。当两个同样级别的中断申请同时到来时，则按一个固定的查寻次序来处理中断响应。8051 单片机的 5 个中断源及其优先级次序如表 2-6 所示。

表 2-6 8051 单片机的中断源

中断源	入口地址	优先级顺序	说 明
外部中断 0	0003H	最高	来自 P3.2 引脚(INT0)的外部中断请求
定时/计数器 0	000BH		定时/计数器 T0 溢出中断请求
外部中断 1	0013H		来自 P3.3 引脚(INT1)的外部中断请求
定时/计数器 T1	001BH		定时/计数器 T1 溢出中断请求
串行口	0023H	最低	串行口完成一帧数据的发送或接收中断

8051 单片机没有专门的开中断和关中断指令，中断的开放和关闭是由特殊功能寄存器 IE 来实现两级控制的。所谓两级控制是指在寄存器 IE 中有一个总允许位 EA，当 EA=0 时，就关闭了所有的中断申请，CPU 不响应任何中断申请。而当 EA=1 时，对各中断源的申请是否开放，还要看各中断源的中断允许位的状态。

中断允许寄存器 IE 的地址为 A8H，其中各位都可以位寻址，位地址为 A8H ~ AFH。总允许位 EA 和各中断源允许位在 IE 寄存器中的分布如下：

D7	D6	D5	D4	D3	D2	D1	D0
EA			ES	ET1	EX1	ET0	EX0

其中各控制位的含义如下：

EA: 中断总允许位。EA=0 时，CPU 关闭所有的中断申请，只有 EA=1 时，才能允许各个中断源的中断申请，但还要取决于各中断源中断允许控制位的状态。

ES: 串行口中断允许位。ES=1，串行口开中断，ES=0，串行口关中断。

ET1: 定时/计数器 T1 的溢出中断允许位。ET1=1 允许 T1 溢出中断，ET1=0 则不允许 T1 溢出中断。

EX1: 外部中断 1(INT1)的中断允许位。ET1=1 允许外部中断 1 申请中断，EX1=0 则不允许中断。

ET0: 定时/计数器 T0 的溢出中断允许位。ET0=1 允许中断，ET0=0 不允许中断。

EX0: 外部中断 0(INT0)的中断允许位。EX0=1 允许中断，EX0=0 不允许中断。

8051 单片机在复位时，IE 各位的状态都为“0”，所以 CPU 是处于关中断的状态。对于串行口来说，其中断请求在被响应之后，CPU 不能自动清除中断标志，在这些情况下要注意用指令来实现中断的开放或关闭，以便进行各种中断处理。

8051 单片机的中断优先级控制比较简单，每个中断源的优先级由特殊功能寄存器 IP 来管理。

IP 寄存器的地址为 B8H，格式如下：

D7	D6	D5	D4	D3	D2	D1	D0
			PS	PT1	PX1	PT0	PX0

其中各位的含义如下：

PS: 串行口中断优先级控制位。

PT1: 定时器/计数器 T1 中断优先级控制位。

PX1: 外部中断 INT1 中断优先级控制位。

PT0: 定时器/计数器 T0 中断优先级控制位。

PX0: 外部中断 INT0 中断优先级控制位。

IP 寄存器中若某一个控制位置“1”，则相应的中断源就规定为高优先级中断，反之，若某一个控制位置“0”，则相应的中断源就规定为低优先级中断。IP 寄存器的地址为 B8H，其中各控制位也是可以位寻址的，位地址为 B8H-BCH。

外部中断触发方式通过特殊功能寄存器 TCON (地址为 88H) 控制，TCON 中各控制位分布如下：

D7	D6	D5	D4	D3	D2	D1	D0
TF1		TF0		IE1	IT1	IE0	IT0

其中各控制位的含义如下：

IT0: 选择外中断 INT0 的中断触发方式。IT0=0 为电平触发方式，低电平有效。IT0=1 为负边沿触发方式，INT0 脚上的负跳变有效。IT0 的状态可以用指令来置“1”或清“0”。

IE0: 外中断 INT0 的中断申请标志。当检测到 INT0 上存在有效中断申请时，由硬件使 IE0 置位。当 CPU 转向中断服务程序时，由硬件清“0” IE0 中断申请标志。

IT1: 选择外中断 INT1 的触发方式(功能与 IT0 类似)。

IE1: 外部中断 INT1 的中断申请标志(功能与 IE0 类似)。

TF0: 定时/计数器 T0 溢出中断申请标志。当 T0 溢出时，由内部硬件将 TF0 置“1”，当 CPU 转向中断服务程序时，由硬件将 TF0 清“0”，从而清除 T0 的中断申请标志。

TF1: 定时器 1 溢出中断申请标志(功能与 TF0 相同)。

可见定时/计数器溢出中断和外部中断的申请标志，在 CPU 响应中断之后能够自动撤除。

8051 单片机在接收到中断源发来的中断申请以后，先把这些申请锁定在各自的中断标志位中，然后在下一个机器周期按表 2-5 规定的内部优先顺序和中断优先级分别来查询这些标志，并在一个机器周期之内完成检测和优先排队。响应中断的条件有三个：

(1) 必须没有同级或更高级别的中断正在得到响应，如果有的话，则必须等 CPU 为它们服务完毕，返回主程序并执行一条指令之后才能响应新的中断申请。

(2) 必须要等当前正在执行的指令执行完毕以后，CPU 才能响应新的中断申请。

(3) 若正在执行的指令是 RETI (中断返回) 或是任何访问 IE 寄存器或 IP 寄存器的指令，则必须要在执行完该指令以及紧随其后的另外一条指令之后才可以响应新的中断申请。

CPU 响应中断后将自动跳转到中断源的入口地址开始执行程序，每个中断源都有其固定的入口地址，它们的处理过程也有所区别。一般情况下，中断处理包括两个部分：一是保护现场，二是为中断服务。

所谓保护现场就是将需要在中断服务程序中使用而又不希望破坏其中原来内容的工作寄存器压入堆栈中保护起来，等中断服务完成后再从堆栈中弹出以恢复原来的内容。通常需要保护的寄存器有 PSW、A 以及其他工作寄存器。

在编写中断服务程序时，要注意以下几点：

(1)各中断源的入口地址之间仅相隔 8 个单元，如果中断服务程序的长度超过 8 个地址单元时，应在中断入口地址处安排一条转移指令，转到其他有足够空余存储器单元的地址空间。

(2)若在执行当前中断服务程序时需要禁止更高级中断源，则要用软件指令关闭中断，在中断返回之前再开放中断。

(3)在保护和恢复现场时，为了不使现场信息受到破坏或造成混乱，保护现场之前应关中断，若需要允许高级中断，则应在保护现场之后再开中断。同样在恢复现场之前也应先关中断，恢复现场之后再开中断。

(4)及时清除那些不能被硬件自动清“0”的中断请求标志，以免产生错误的中断。

4.4.3 实验内容

1 采用外部中断INT0触发P1.0的输出状态反转

在实验系统上如图2.30和表2-7所示连线。P1.0接LED灯，INT0（P3.2）端接单脉冲发生器。编写并执行程序，每按一次单脉冲按钮，注意观察LED的状态。如果中断服务程序长度小于8个字节，可以直接从入口地址开始编写，如果中断服务程序较长，则应在中断入口放一条长转移指令，跳转到真正的服务程序地址。

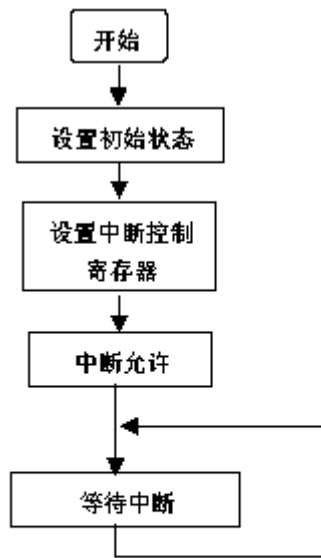


图 2.30

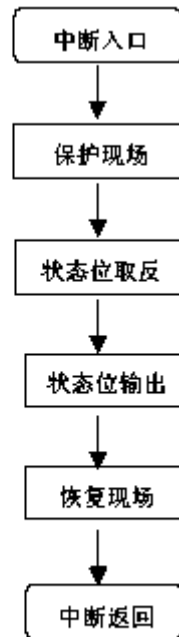
表 2-7

连线	连接孔 1	连接孔 2
1	P1.0	L0
2	单脉冲输出	INT0

程序参考流程如下：



主程序框图



外部中断子程序框图

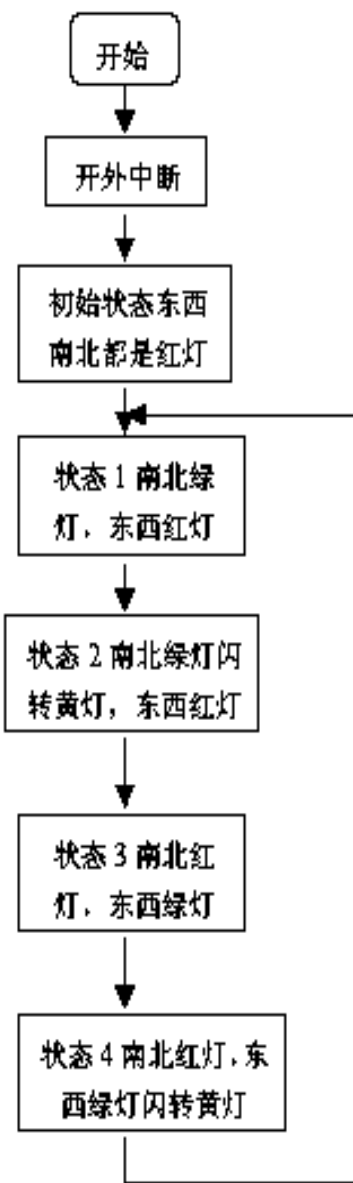
2 采用外部中断模拟交通信号灯控制

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，采用单脉冲申请外部中断，表示有急救车通过。在实验系统上如表2-8所示连线，P1.0~P1.5接LED灯，INT0 (P3.2)端接单脉冲发生器。编写并执行程序，每按一次单脉冲按钮，注意观察LED的状态。

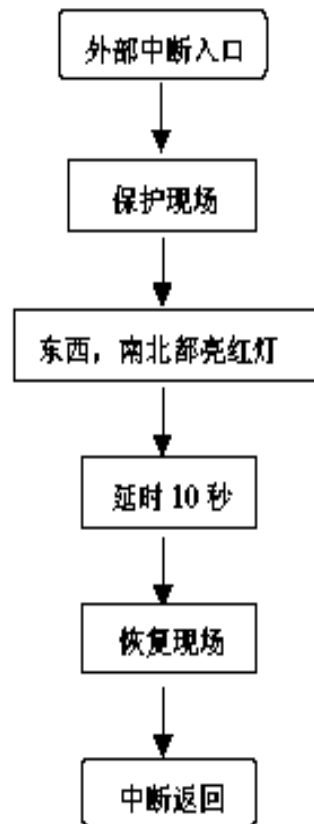
表 2-8

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)

程序参考流程如下：



主程序框图



外部中断子程序框图

复习思考题

- 1 若在实验中采用INT1，应如何修改实验程序？
- 2 为什么在中断服务程序中需要保护现场？通过实验后有何发现
- 3 8051单片机在响应中断之后是否能够自动取消所有中断请求标志？对于低电平触发的外部中断应如何处理？

2.5 实验五 定时/计数器实验

2.5.1 实验目的与要求

通过本实验学习8031内部定时器的使用和编程方法,进一步掌握中断处理程序的编程方法。要求

8051内部定时器T0用中断方式计时,实现每一秒钟使P1.0的输出状态发生一次反转,从而形成方波输出。

8051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,并将计数值按二进制数通过P1口驱动LED灯显示出来。

2.5.2 8051单片机定时/计数器简介

8051 单片机内部有两个 16 位可编程定时/计数器,记为 T0 和 T1。它们的工作方式可以通过指令对相应特殊功能寄存器编程来设定,或作定时器用,或作外部事件计数器用。

定时/计数器在硬件上由双字节加法计数器 TH 和 TL 组成,作定时器使用时,计数脉冲由单片机内部振荡器提供,计数频率为 $f_{osc}/12$,每个机器周期加 1。作计数器使用时,计数脉冲由 P3 口的 P3.4(或 P3.5)即 T0(或 T1)引脚输入,外部脉冲的下降沿触发计数,计数器在每个机器周期的 S5P2 期间采样外部脉冲,若一个周期的采样值为 1,下一个周期的采样值为 0,则计数器加 1,故识别一个从 0 到 1 的跳变需要 2 个机器周期,所以对外部计数脉冲的最高计数频率为 $f_{osc}/24$,同时还要求外部脉冲的高低电平保持时间均要大于一个机器周期。

定时/计数器的工作方式由特殊功能寄存器 TMOD 编程决定,定时/计数器的启动运行由特殊功能寄存器 TCON 编程控制。不论用作定时器还是用作计数器,每当产生溢出时,都会向 CPU 发出中断申请。

方式控制寄存器 TMOD 的地址为 89H,控制字格式如下:

D7	D6	D5	D4	D3	D2	D1	D0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

低 4 位为 T0 的控制字,高 4 位为 T1 的控制字,其中各位的具体含义介绍如下。

GATE 为门控位。它对定时/计数器的启动起辅助控制作用。GATE=1 时,定时/计数器的计数受外部引脚 P3.2(INT0)或 P3.3(INT1)输入电平的控制,此时只有当 P3 口的 P3.2(或 P3.3)引脚即 INT0 (或 INT1)上的电平为 1 才能启动计数;GATE=0 时,定时器/计数器的运行不受外部引脚输入电平的控制。

C/T 为方式选择位。C/T = 0 为定时器方式,采用单片机内部振荡脉冲的 12 分频信号作为计数脉冲,若采用 12MHz 的晶振,则计数频率为 1MHz,从计数值便可计算出定时时间。C/T=1 为计数器方式,采用外部引脚(T0 为 P3.4, T1 为 P3.5)的输入脉冲作为计数脉冲,当 T0(或 T1)上的输入信号发生从高到低的负跳变时,计数器加 1。最高计数频率为单片机晶振频率的 1/24。

M1、M0 二位的状态确定定时/计数器的工作方式,详见表 2-9。

表 2-9 定时/计数器的方式选择

M1	M0	工作方式
0	0	方式 0，为 13 位定时/计数器。
0	1	方式 1，为 16 位定时/计数器。
1	0	方式 2，为自动重装常数的 8 位定时/计数器。
1	1	方式 3，仅适用于 T0，分成 2 个 8 位定时/计数器。

定时/计数器控制寄存器 TCON 的地址为 88H，格式如下：

D7	D6	D5	D4	D3	D2	D1	D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 为定时/计数器 T1 的溢出标志位。当 T1 被允许计数以后，T1 从初值开始加 1 计数，计数器的最高位产生溢出时置“1”TF1，并向 CPU 申请中断，当 CPU 响应中断时，由硬件清“0”TF1。TF1 也可由软件查询清“0”。

TR1 为定时/计数器的运行控制位，由软件置位和复位。当方式控制寄存器 TMOD 中的 GATE 位为 0，且 TR1 为 1 时允许 T1 计数，TR1 为 0 时禁止 T1 计数。当 GATE 为 1 时，仅当 TR1 为 1 且 INT1 (P3.2)输入为高电平时才允许 T1 计数，当 TR1 为 0 或 INT1 输入为低电平时都禁止 T1 计数。

TR0 为定时器 T0 的运行控制位，其功能与 TR1 类似。

TF0 为定时器 T0 的溢出标志位，其功能与 TF1 类似。

8051 单片机的定时/计数器在进行定时或计数之前要进行初始化编程，通常包括如下几个步骤：

- (1) 确定工作方式，即给方式控制寄存器 TMOD 写入控制字。
- (2) 计算定时器/计数器初值，并将初值写入寄存器 TL 和 TH。
- (3) 根据需要对中断控制寄存器 IE 置初值，决定是否开放定时器中断。
- (4) 使 TCON 中的 TR0 或 TR1 置“1”，启动定时器/计数器。

在初始化过程中，要设置定时或计数的初始值，则计算初值 X 的公式如下：

计数方式： $X = M - \text{要求的计数值}$
 式中 M 取决于不同工作方式，可以是 2^{13} 、 2^{16} 或 2^8 。

定时方式： $X = (M - (\text{要求的定时值} / t))$
 式中 $t = 12 / f_{osc}$ ，称为定时周期。

4.5.3 实验内容

1 用 T0 中断方式定时，每秒钟使 P1.0 输出状态发生一次反转

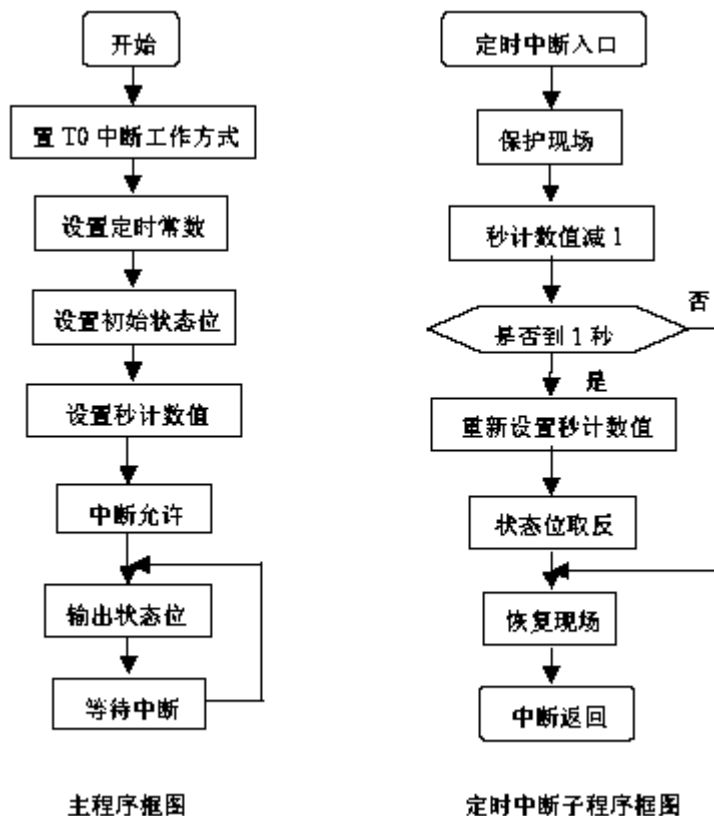
将 T0 作定时器定时 1 秒，由于实验系统 $f_{osc} = 6\text{MHz}$ ，定时周期 $t = 2\mu\text{s}$ 。为了实现定时 1 秒，通过中断服务程序进行定时扩展，即设置 T0 定时 100us 产生一次中断，同时设置一个软件计数器来计数中断次数，当达到中断 1000 次时即完成了定时 1 秒钟。

在实验系统上如表 2-10 所示连线。P1.0 接 LED 灯，编写并执行程序，注意观察 LED 的状态的变化。

表 2-10

连线	连接孔 1	连接孔 2
1	P1.0	L0

程序参考流程如下：



2 用T0计数外部脉冲,通过P1口显示计数值

采用T0作计数器，外部计数脉冲由 P3.4引入，8051单片机在每个机器周期采样一次输入信号，因此至少需要两个机器周期才能检测到一次脉冲跳变。这就要求被采样的外部脉冲高低电平保持时间均要大于一个机器周期，以保证电平在变化之前即被采样，并且外部计数脉冲的最高计数频率为 $f_{osc}/24$ 。

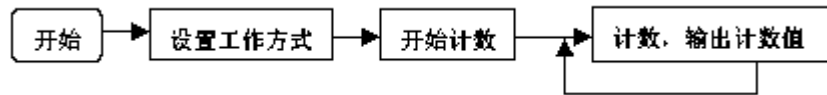
在实验系统上如表2-11所示连线。

表 2-11

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	T0

P1.0 ~ P1.3接LED灯, T0 (P3.4) 端接单脉冲发生器。编写并执行程序, 每按一次单脉冲按钮, T0将计数一个脉冲, 注意观察LED的状态。

程序参考流程如下:



复习思考题

- 1 若在实验中采用定时器T1, 应如何修改实验程序?
- 2 实验中是如何实现定时时间扩展的? 能否实现更长时间的扩展?
- 3 利用8051定时器控制寄存器TMOD中的GATE位, 设计一个用T1实现测量脉冲宽度的实验。

2.6 实验六 8255输入输出实验

2.6.1 实验目的与要求

通过实验进一步学习了解 8255 可编程芯片的结构及编程方法，要求编程实现 8255 的 PA、PB 和 PC 口的输入输出。

2.6.2 8255可编程芯片简介

8255 具有 3 个 8 位的并行 I/O 口，分别称为 PA、PB、PC，其中 PA 具有一个 8 位数据输出锁存/缓冲器和一个 8 位数据输入锁存器，可编程为 8 位输入输出或双向寄存器。PB、PC 与 PA 类似，不同的是 PB 不能用作双向寄存器，而 PC 又分高 4 位(PC7 ~ PC4)和低 4 位(PC3 ~ PC0)，PC 除了用作输入输出之外，还可用作 PA、PB 口选通工作方式下的状态控制信号。

8255 内部读写控制逻辑用于管理所有的数据、控制字或状态字的传递，它接受来自 CPU 的地址及控制信号来控制各个端口的工作状态。其控制信号有：

复位信号 RESET：高电平有效。复位时控制寄存器被清 0，所有端口都设置为输入方式。

片选信号/CS：低电平有效。允许 8255 与 CPU 交换信息。

读信号/RD：低电平有效。允许 CPU 从 8255 端口读取数据或外设状态信息。

写信号/WR：低电平有效。允许 CPU 将数据、控制字写入 8255 中。

端口选择信号 A1、A0：它们与 /RD、/WR 及 /CS 信号配合来选择 I/O 端口及内部控制寄存器，并控制信息的传送方向，如表 2-12 所示。

表 2-12 8255 的端口选择及其功能

A1	A0	/RD	/WR	/CS	功 能 说 明
0	0	0	1	0	A 口 数据总线
0	1	0	1	0	B 口 数据总线
1	0	0	1	0	C 口 数据总线
0	0	1	0	0	数据总线 A 口
0	1	1	0	0	数据总线 B 口
1	0	1	0	0	数据总线 C 口
1	1	1	0	0	数据总线 控制寄存器
x	x	x	x	1	数据总线为三态
1	1	0	1	0	非法状态
x	x	1	1	0	数据总线为三态

8255 有三种工作方式：方式 0、方式 1 和方式 2。

方式 0 为基本输入输出方式，PA、PB 和 PC 口都可以设定为输入或输出，作为输出口时，输出的数据被锁存，作为输入口时，输入数据不锁存。

方式 1 为选通输入输出方式，PA、PB 和 PC 三个口分为两组：A 组包括 PA 口和 PC

口的高 4 位，PA 口可编程设定为输入或输出，PC 口高 4 位用作输入输出操作的控制和同步信号。B 组包括 PB 口和 PC 口的低 4 位，PB 口可编程设定为输入或输出，PC 口低 4 位用作输入输出操作的控制和同步信号。PA 和 PB 口的输入输出数据都被锁存。

方式 2 为双向总线方式，仅用于 PA 口，将 PA 为 8 位双向总线端口，PC 的 PC3 ~ PC7 用作输入输出的同步控制信号，此时 PB 口只能编程设定为方式 0 或方式 1。

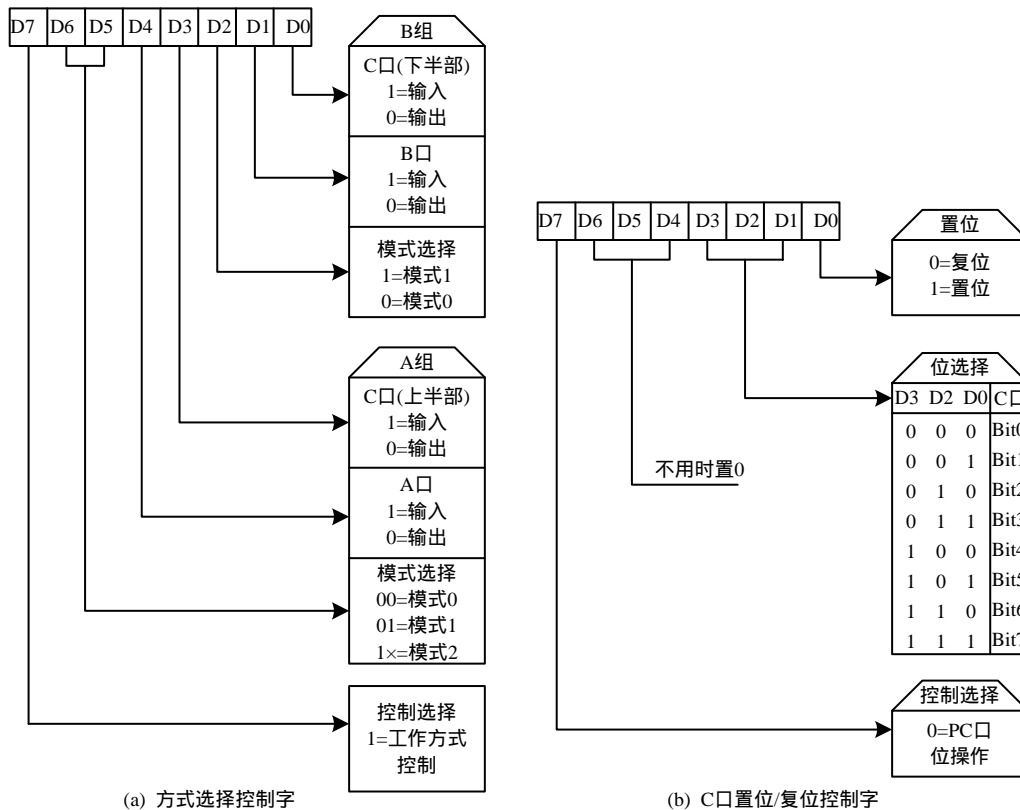


图 2.31 8255 的控制字格式

8255 有两种控制字，即控制 PA、PB、PC 口工作方式的方式控制字和控制 PC 口各位置“1”或清“0”的控制字。两种控制字写入的控制寄存器相同，只是用 D7 位来区分时哪一种控制字：D7=1 为工作方式控制字，D7=0 为 PC 口置“1”或清“0”的控制字。这两种控制字的格式如图 2.31 所示。

4.6.3 实验内容

利用8255方式0，实现PA口作输出，PB口作输入

在实验系统上如图2.32和表2-13所示连线，8255的片选端/CS接地址译码信号/CS0，则命令口地址为8003H，PA口地址为8000H，PB口地址为8001H，PC口地址为8002H。PA0 ~ PA7接LED0 ~ LED7(LED灯)，PB0 ~ PB7接K0 ~ K7(开关量)。数据线、读/写控制、地址线、复位信号在实验板上已经接好。编写并执行程序，拨动开关K0 ~ K7到不同位置，注意观

察LED灯的状态的变化。

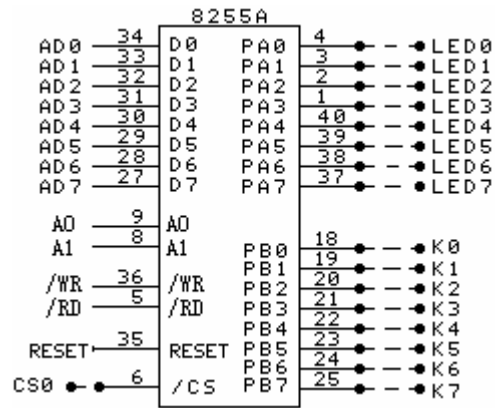
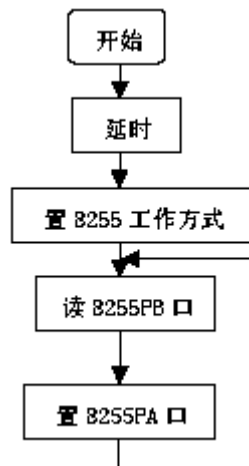


图2.32 8255实验连线

表 2-13

连线	连接孔 1	连接孔 2
1	CS0	8255-CS
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

程序参考流程如下：



复习思考题

- 1 若要求用8255的PB口作输出，PA口作输入，应如何修改实验程序？
- 2 利用8255的方式1可实现与单片机的查询或中断方式接口，设计一个通过查询8255的PC1(IBF B)和PC7(OBF A)实现从PB口输入，从PA口输出的实验程序。
- 3 如果需要采用8255的PC7输出连续方波，如何利用PC口的置位/复位控制命令来实现？

2.7 实验七 D/A转换实验

2.7.1 实验目的与要求

通过实验进一步学习 D/A 转换器的基本原理以及在单片机系统中扩展 D/A 转换器的方法，掌握 D/A 转换芯片 0832 的性能及编程。要求利用 0832 扩展电路，编制程序产生锯齿波、三角波、方波，用示波器查看波形。

2.7.2 D/A转换芯片0832简介

0832 是典型的带内部双缓冲数据缓冲器的 8 位 D/A 芯片，其逻辑结构如图 2.33 示。图中 \overline{LE} 是寄存命令，当 $\overline{LE}=1$ 时，寄存器输出随输入变化，当 $\overline{LE}=0$ 时，数据锁存在寄存器中，而不再随数据总线上的数据变化而变化。当 \overline{ILE} 端为高电平、 \overline{CS} 与 $\overline{WR1}$ 同时为低电平时，使得 $\overline{LE1}=1$ ；当 $\overline{WR1}$ 变为高电平时，输入寄存器便将输入数据锁存。当 \overline{XFER} 与 $\overline{WR2}$ 同时为低电平时，使得 $\overline{LE2}=1$ ，DAC 寄存器的输出随寄存器的输入变化， $\overline{WR2}$ 上升沿将输入寄存器的信息锁存在该寄存器中。 R_{FB} 为外部运算放大器提供的反馈电阻。 V_{REF} 端是由外电路为芯片提供一个 +10V 到 -10V 的基准电源。 I_{out1} 和 I_{out2} 是电流输出端，两者之和为常数。

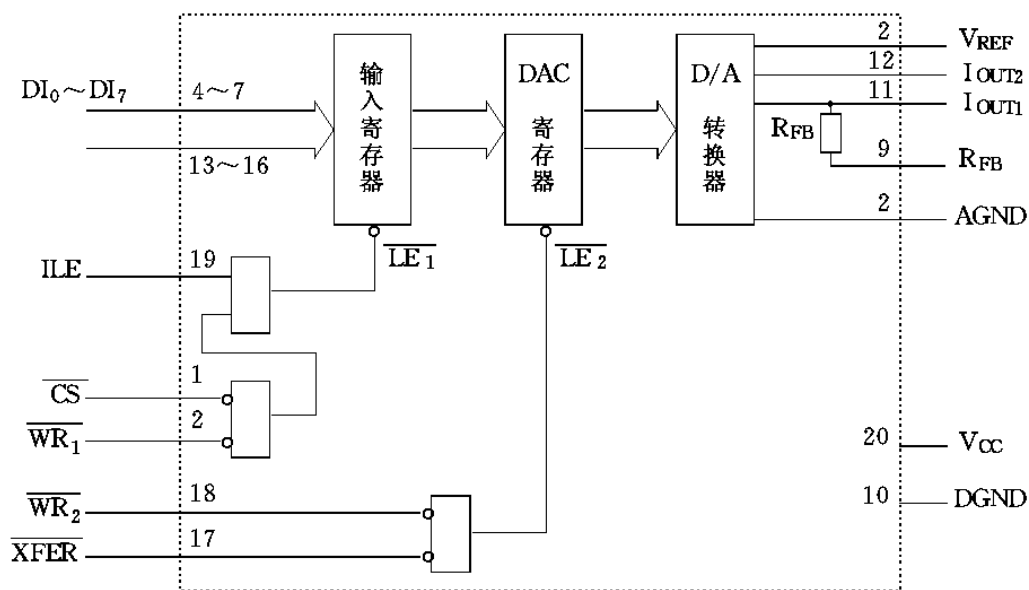


图 2.33 DAC0832 逻辑框图

0832 是电流输出型 D/A 转换器，实际应用中可以通过外接运算放大器将电流转换为电压。0832 的分辨率为 8 位，如其满度电压为 5V，则 1 个 LSB 对应的最小转换电压为： $V_m=5V/2^8=5V/256=19.5mV$ 。

2.7.3 实验内容

利用0832产生不同波形输出

实验系统中DAC0832的接口电路如图2.34所示，按表2-14所示接线，编写并执行正向阶梯波程序、三角波程序、方波程序，用电压表或示波器探头接-5V~+5V输出，观察输出电压或波形。

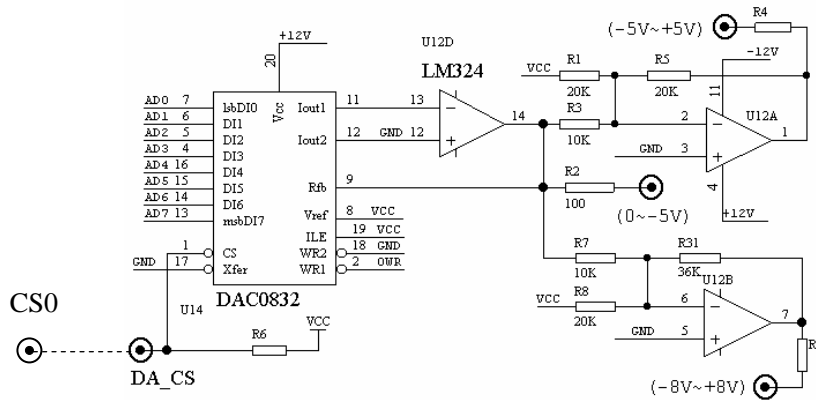
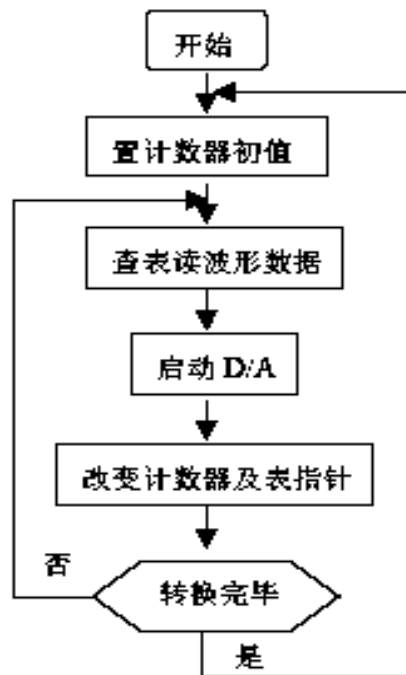


图2.34 DAC0832接口电路

表 2-14

连线	连接孔 1	连接孔 2
1	DA_CS	CS1
2	-5V~+5V	电压表或示波器

程序参考流程如下：



复习思考题

- 1 若希望输出一个梯形波，应如何修改实验程序？
- 2 设计一个用查表方式实现的正弦波实验程序。

2.8 实验八 A/D转换实验

2.8.1 实验目的与要求

通过实验进一步学习A/D转换器的基本原理以及在单片机系统中扩展A/D转换器的方法，掌握A/D转换芯片0809的性能及编程，了解单片机如何进行数据采集。

要求利用实验系统上的ADC0809，以及电位器提供模拟量输入，编制A/D转换程序，将模拟量转换成二进制数字量，用8255的PA口输出到LED发光二极管显示。

2.8.2 ADC0809简介

ADC0809是一种较为常用的8路模拟量输入、8位数字量输出的逐次比较式ADC芯片，图2.35为ADC0809的原理结构框图。芯片的主要部分是一个8位的逐次比较式A/D转换器。为了能够实现8路模拟信号的分时采集，在芯片内部设置了多路模拟开关及通道地址锁存和译码电路，因此能对多路模拟信号进行分时采集和转换。转换后的数据送入三态输出数据锁存器。ADC0809的最大不可调误差为 $\pm 1\text{LSB}$ ，典型时钟频率为640kHz，时钟信号应由外部提供。每一个通道的转换时间约为 $100\mu\text{s}$ 。

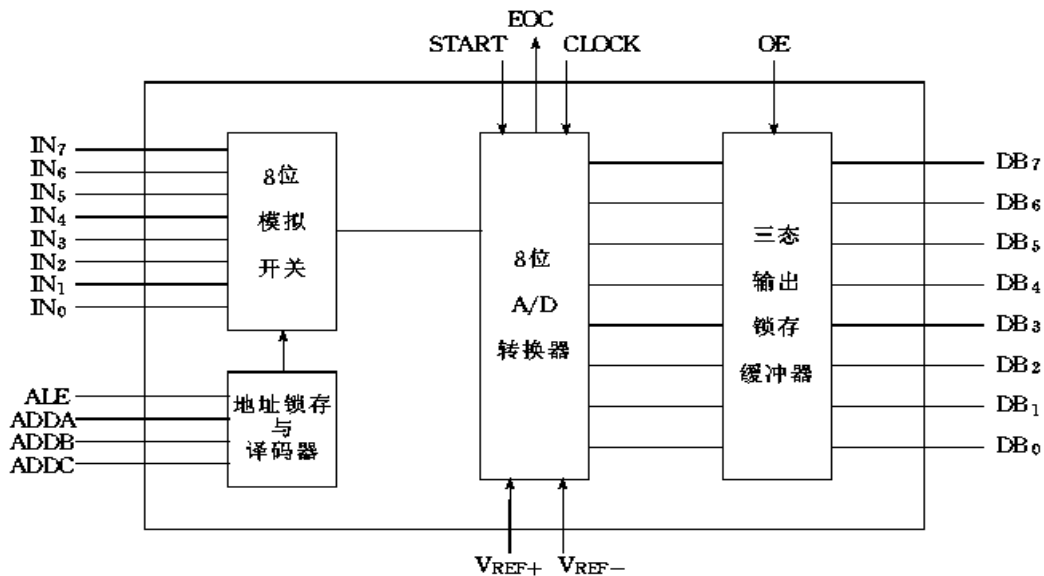


图 2.35 ADC0809 的原理结构框图

ADC0809 各引脚的功能如下：

IN₀ ~ IN₇： 8 路模拟量输入端。

D₀ ~ D₇： 数字量输出端。

START： 启动脉冲输入端，脉冲上升沿复位 0809，下降沿启动 A/D 转换。

ALE： 地址锁存信号，高电平有效时把三个地址信号送入地址锁存器，并经地址译码得到地址输出，用以选择相应的模拟输入通道。

EOC： 转换结束信号，转换开始时变低，转换结束时变高，变高时降转换结果打入

三态输出锁存器。如果将 EOC 和 START 相连，加上一个启动脉冲则连续进行转换。

OE: 输出允许信号输入端。

CLOCK: 时钟输入信号，最高允许值为 640KHz。

$V_{REF(+)}$: 正基准电压输入端。

$V_{REF(-)}$: 负基准电压输入端。通常将 $V_{REF(+)}$ 接+5V， $V_{REF(-)}$ 接地。

VCC: 电源电压，可从+5V ~ +15V。

由于 0809 芯片没有专门的片选信号输入端，因此在与单片机进行接口时必须通过进行时序分析，图 2.36 为 ADC0809 的工作时序。

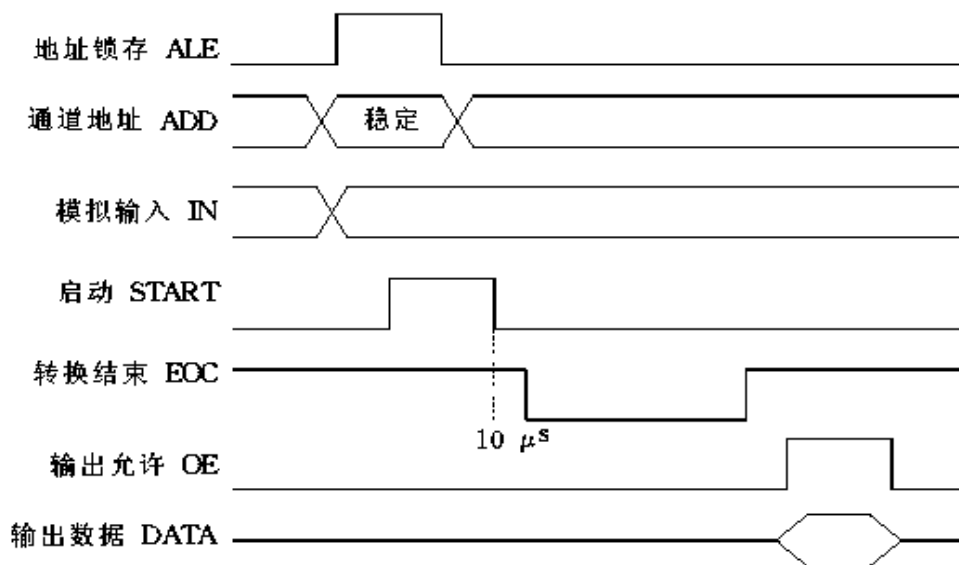


图 2.36 ADC0809 的工作时序

2.8.3 实验内容

利用0809实现A/D转换并从LED上显示转换结果

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。

实验电路如图 2.37 所示。ADC0809 每进行一次 A/D 转换需要 100us 的时间。采用延时方式读入 A/D 转换结果，也可以用查询或中断方式读入结果，将 0809 的 EOC 引脚与 CPU 的外部中断 INT0 (P3.2) 相连，采用查询方式时通过读取 8051 的 P3.2 引脚状态，来确定 A/D 转换是否结束。在中断方式下，A/D 转换结束后会自动产生 EOC 信号，8051 在开中断状态下，每当 0809 转换结束时通过中断服务程序自动读取 A/D 转换结果。

按表 2-15 连线，编写并执行程序，旋动实验系统上的电位器，注意观察 LED 状态的变化。

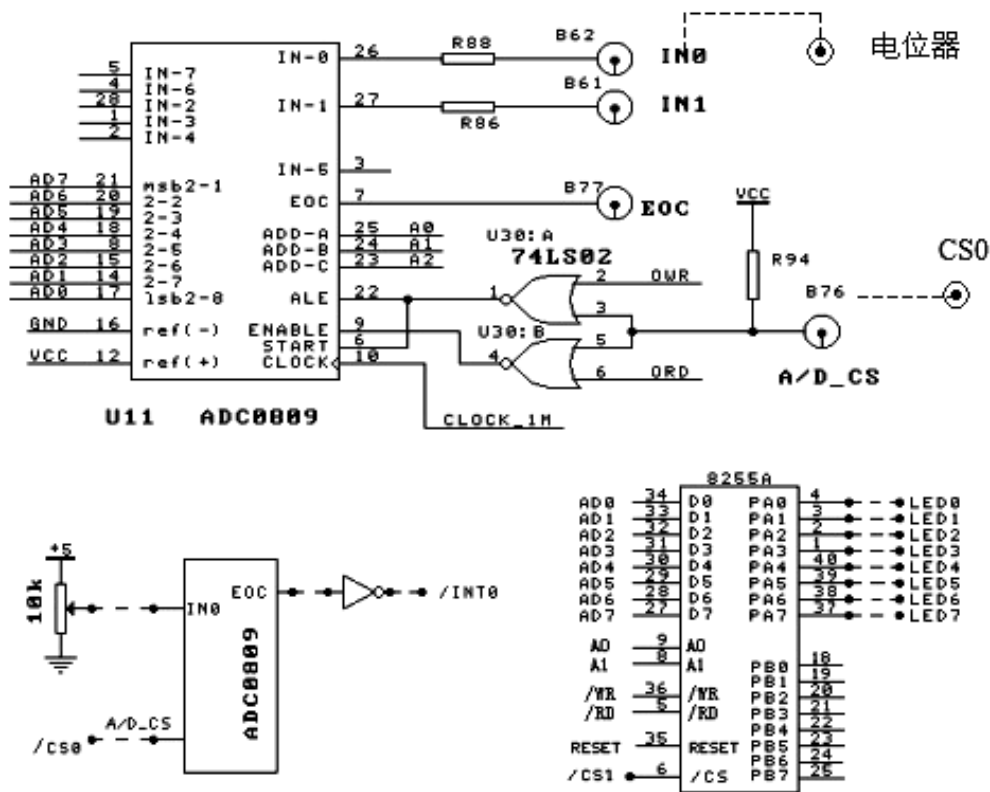
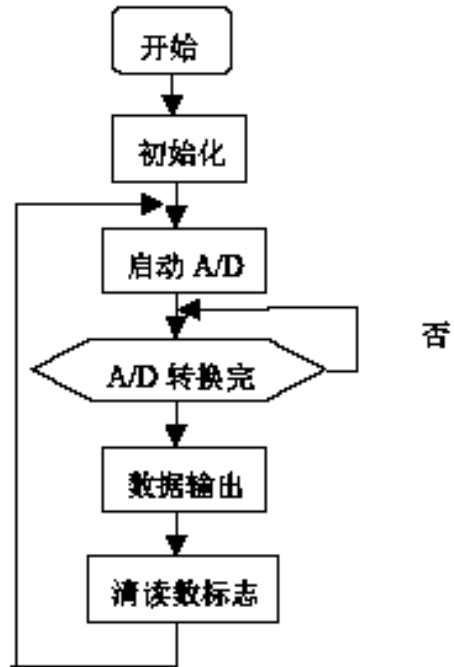


图2.37 ADC0809接口电路

表 2-15

连线	连接孔 1	连接孔 2
1	IN0	电位器输出
2	AD_CS	CS2
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

程序参考流程如下：



复习思考题

- 1 认真复习实验，希望采用查询方式读取A/D结果，应如何修改实验程序？
- 2 设计一个用中断方式实现读取A/D转换结果并显示的实验程序。

2.9 实验九 键盘显示实验

2.9.1 实验目的与要求

通过实验学习键盘扫描和8段LED数码显示器的工作原理，掌握键盘和显示器的接口方法。要求：利用实验系统提供的显示电路，实现动态显示数据。利用实验系统提供的键盘显示电路，实现键盘扫描和数码显示，把按键输入的键码在六位数码管上显示出来。

2.9.2 数码管显示及键盘扫描工作原理

数码显示器是由7段条形的LED组成，在右下角带有一个园形的LED作小数点用，这样一共有8段，恰好适用于8位的并行系统。点亮适当的字段，就可显示出不同的数字。数码显示器有共阴极和共阳极两种，共阴极显示器的公共阴极接地，当各段阳极上的电平为“1”时，该段点亮，电平为“0”时，段就熄灭；共阳极显示器的公共阳极接+5V电源，当各段阴极上的电平为“0”时，该段就点亮，电平为“1”时，段就熄灭。

实验系统提供了6位8段LED数码管显示电路，8位段码、6位位码分别由两片74LS374输出。位码经MC1413或ULN2003倒相驱动后，选择相应显示位。8位段码输出地址为0X004H，位码输出地址为0X002H。此处“X”由地址译码输入端KEY/LED CS决定，如将KEY/LED CS接到CS0上，则段码地址为08004H，位码地址为08002H。8段数码管的字型代码如下：

a	显示字形	g	f	e	d	c	b	a	段码
-----	0	0	1	1	1	1	1	1	3fh
f b	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
g	4	1	1	0	0	1	1	0	66h
e c	5	1	1	0	1	1	0	1	6dh
-----	6	1	1	1	1	1	0	1	7dh
d	7	0	0	0	0	1	1	1	07h
.h	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

实验系统提供了一个6×4的小键盘，向行扫描地址(0X002H)逐行输出低电平，然后从列地址(0X001H)读回。如果有键按下，则相应列的值应为低，如果无键按下，由于上拉

的作用,列的值为高。这样就可以通过输出的行值和读取的列值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。扫描地址和列地址中的“X”由地址译码输入端KEY/LED CS 决定,例如将KEY/LED CS信号接CS0上,则行扫描地址为08002H,列地址为08001H。行扫描码还可以分时用作LED数码管的位选通信号。

2.9.3 实验内容

1 8段数码管动态显示

实验原理电路如图2.38所示,按表2-16连线,编写并执行程序,注意观察8段LED数码管显示字形的变化。

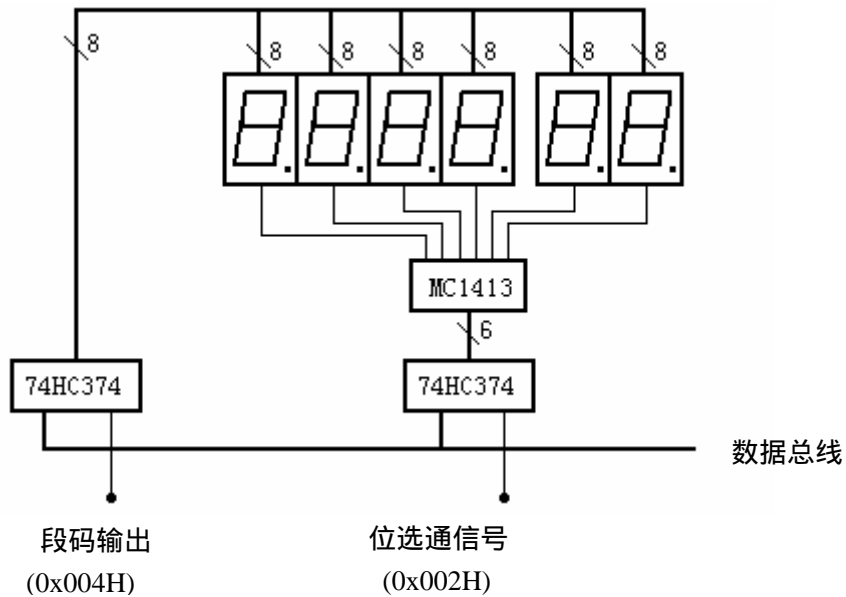
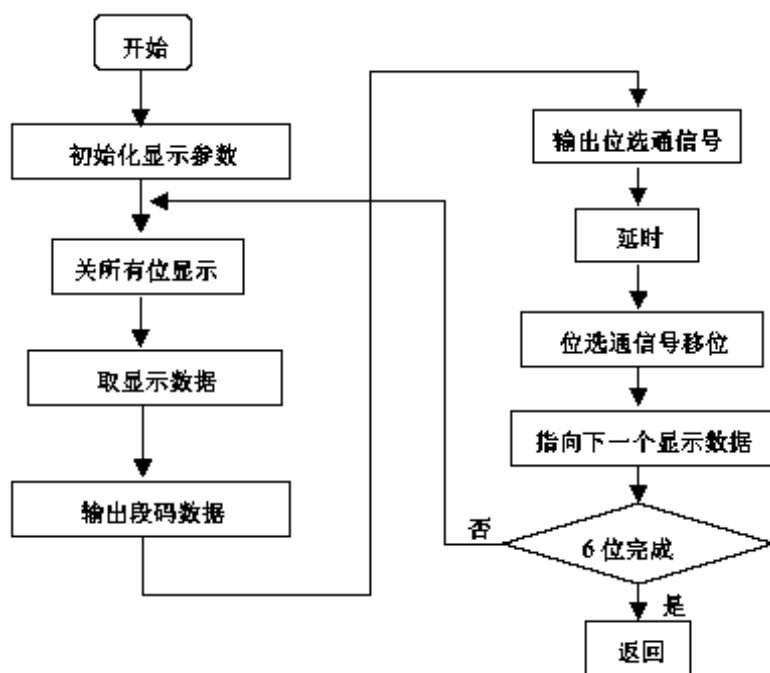


图2.38 数码管显示原理电路

表 2-16

连线	连接孔 1	连接孔 2
1	KEY/LED_CS	CS2

程序参考流程如下：



2 键盘扫描显示

本实验实际上是前一个实验的扩展，原理电路如图2.39所示，按表2-17连线。

实验程序可分成三个模块：

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。

显示模块：将显示单元的内容在显示器上动态显示。

主程序：调用键输入模块和显示模块。

编写并执行程序，按下不同按键，注意观察8段LED数码管显示数据的变化。

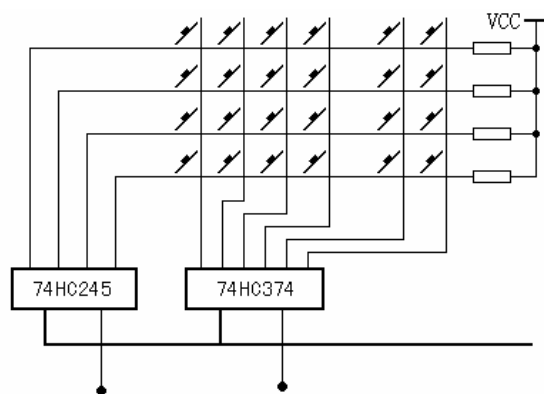
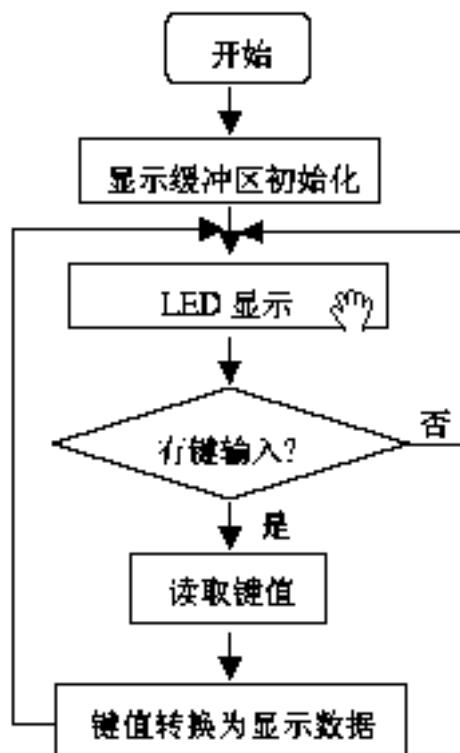
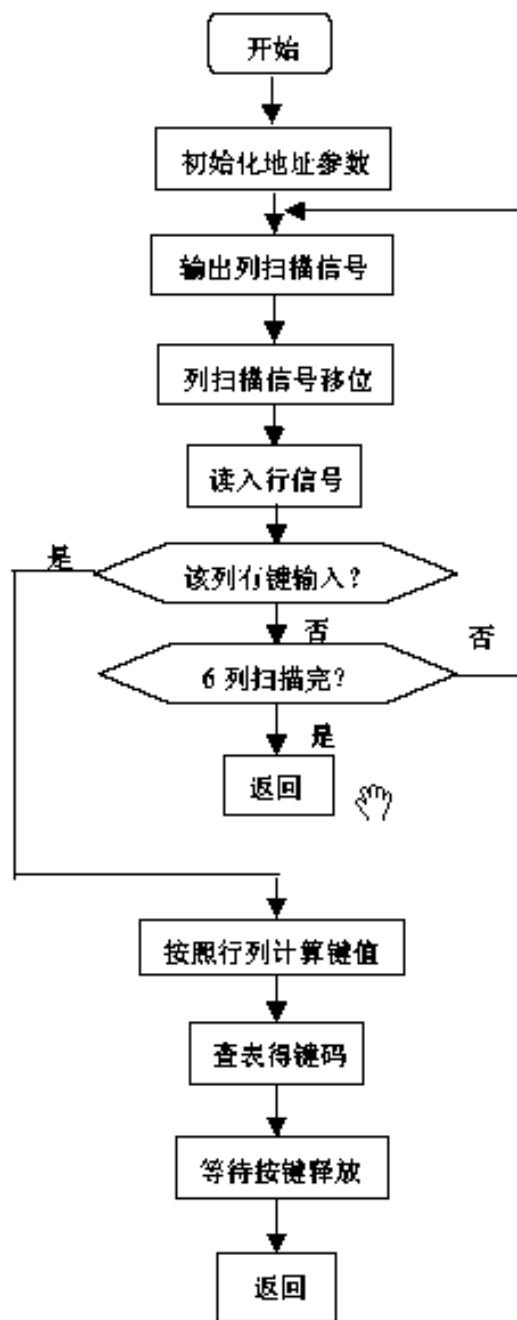


图2.39 扫描式键盘原理电路

程序参考流程如下：



主程序框图



读键输入子程序框图

复习思考题

- 1 认真复习实验过程，弄清8段数码管动态显示原理。
- 2 改写数码管动态显示实验程序实现显示“HELLO”字符。
- 3 改写键盘扫描显示实验程序，实现按键显示“123456”字符。

2.10 实验十 电子时钟实验

2.10.1 实验目的与要求

通过实验加深理解 8051 定时器和中断系统工作原理，进一步掌握定时器任意定时时间的编程方法，以及数码显示电路的驱动编程方法。要求利用 8051 定时器和实验系统提供的数码显示电路，设计一个电子时钟。显示格式为：XX XX XX 由左向右分别为：时、分、秒

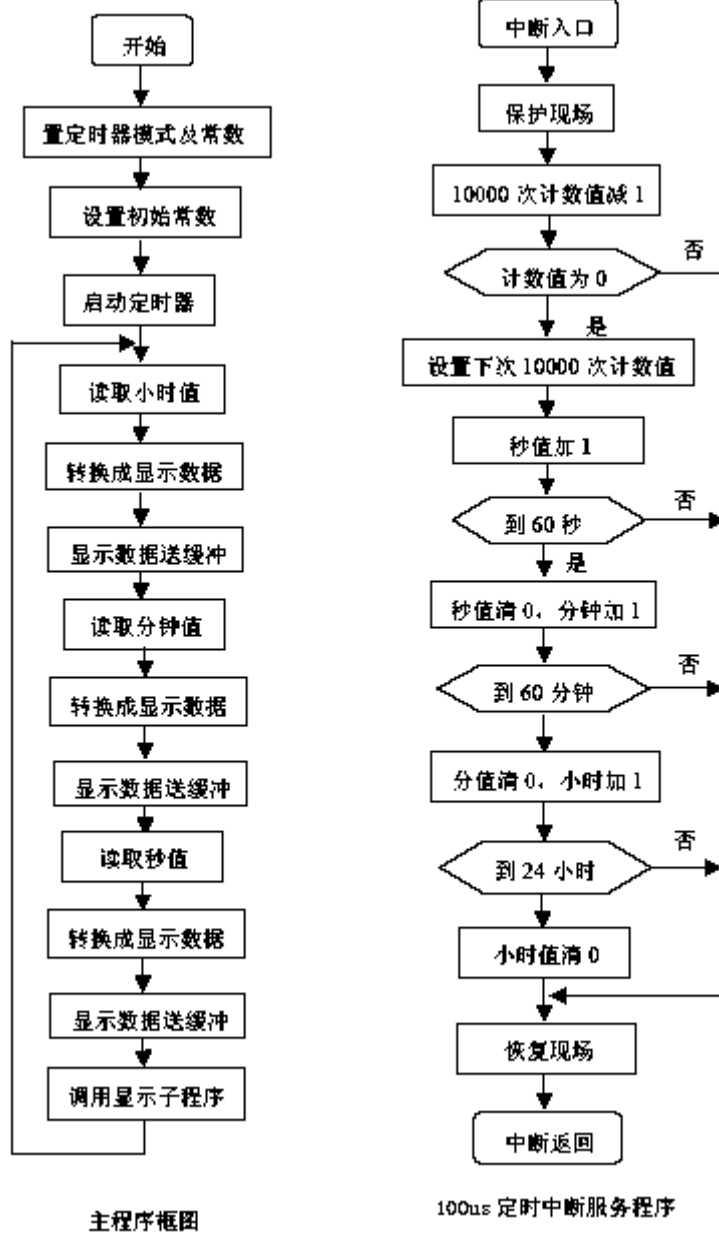
2.10.2 实验内容

设置8051内部定时器每100us中断一次，在中断服务程序中,对中断次数进行计数，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。本实验只需接上显示/键盘的选择信号，按表2-17所示连线，编写并执行程序，使显示器上按上述格式显示时、分、秒值。

表 2-17

连线	连接孔 1	连接孔 2
1	KEY/LED_CS	CS2

程序参考流程如下：



复习思考题

- 1 认真复习实验过程，进一步弄清定时器任意定时时间的工作原理和8段数码管动态显示原理。
- 2 改写实验程序实现用按键设定电子时钟的时、分、秒值。

2.11 实验十一 串行口通讯实验

2.11.1 实验目的与要求

通过实验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，学习串行口通讯中断方式的程序编写方法，掌握单片机串行口简易三线式通讯的实现方法。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。

2.11.2 8051单片机串行口简介

8051 单片机内部有一个可编程的全双工串行接口，它在物理上分为两个独立的发送缓冲器和接收缓冲器 SBUF，这两个缓冲器占用一个特殊功能寄存器地址 99H，究竟是发送缓冲器还是接收缓冲器工作是靠软件指令来决定的。对外有两条独立的收、发信号线 RXD(P3.0)和 TXD(P3.1)，因此可以同时接收和发送数据，实现全双工传送，使用串行口时可以用定时器 T1 或 T2 作为波特率发生器。

8051 串行口通过两个特殊功能寄存器 SCON 和 PCON 来进行控制，分别介绍如下。

串行口控制寄存器 SCON(地址为 98H):这个特殊功能寄存器包含有串行口的工作方式选择位，接收发送控制位及串行口的状态标志，格式如下：

D7	D6	D5	D4	D3	D2	D1	D0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 和 SM1 为串行口的工作方式选择位，详见表 2-18。

表 2-18

SM0	SM1	工作方式
0	0	方式 0，移位寄存器方式(用于 I/O 口扩展)。
0	1	方式 1，8 位 UART，波特率可变(T1 溢出率/n)。
1	0	方式 2，9 位 UART，波特率为 $f_{osc}/64$ 或 $f_{osc}/32$ 。
1	1	方式 3，9 位 UART，波特率可变(T1 溢出率/n)。。

表中，n 为 16 或 32，取决于特殊功能寄存器 PCON 中 SMOD 位的值，SMOD=1 时，n= 16；SMOD=0 时，n=32。UART 表示通用异步收发器。

串行口有四种工作方式，说明如下。

方式 0 为移位寄存器输入/输出方式。串行数据从 RXD 线输入或输出，而 TXD 线专用于输出时钟脉冲给外部移位寄存器。这种方式主要用于进行 I/O 口的扩展，输出时将片内发送缓冲器中的内容串行地移入外部的移位寄存器，输入时将外部移位寄存器中的内容移入片内接收缓冲器，波特率固定为 $f_{osc}/12$ 。

方式 1 为 8 位异步接收发送。一帧数据有 10 位，包括 1 位起始位(0)，8 位数据位和 1 位停止位(1)。串行口电路在发送时能自动插入起始位和停止位，在接收时，停止位进入 SCON 中的 RB8 位。方式 1 的传送波特率是可变的，由定时器 1 的溢出率决定。

方式 2 为 9 位异步接收发送。一帧数据包括有 11 位，除了 1 位起始位、8 位数据位、

1 位停止位之外，还可以插入第 9 位数据，字符格式如图 2.40 所示。

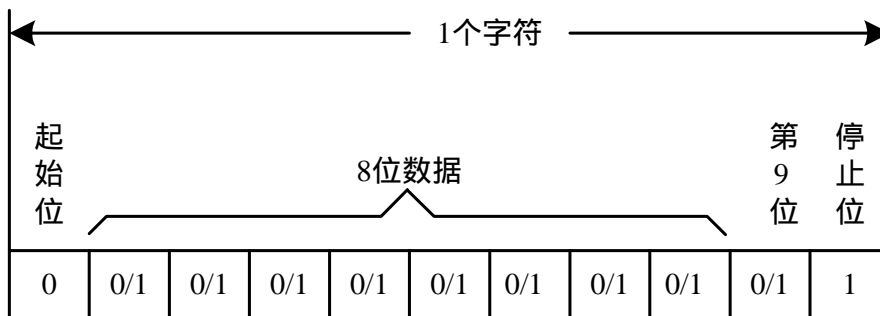


图 2.40 串行口方式 2 的 9 位 UART 数据格式

发送时，第 9 位数据的值可通过 SCON 中的 TB8 指定为“0”或“1”，用一些附加的指令可使这一位作奇偶校验位。接收时，第 9 位数据进入特殊功能寄存器 SCON 中的 RB8 位。方式 2 的波特率为 $f_{osc}/64$ 或 $f_{osc}/32$ 。

方式 3 也是 9 位异步接收发送，一帧数据有 11 位，工作方式与方式 2 相同，只是传送时的波特率受定时器 1 的控制，即波特率可变。

SCON 寄存器中另外各位的意义如下：

SM2 为允许在方式 2 和方式 3 时进行多机通讯的控制位。若允许多机通讯，则应使 $SM2=1$ ，然后根据收到的第 9 位数据值来决定从机是否接收主机的信号，当 $SM2=0$ 时禁止多机通讯。

REN 为允许串行接收位。由软件置位以允许接收。由软件清“0”来禁止接收。

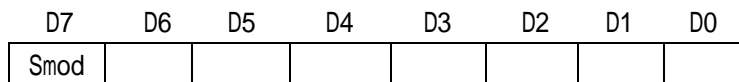
TB8 为方式 2 和方式 3 时发送的第 9 位数据。需要由软件置位或复位。

RB8 为方式 2 和方式 3 时接收到的第 9 位数据。在方式 1，若 $SM2=0$ ，则 RB8 是接收到的停止位；在方式 0，不使用 RB8。

TI 为发送中断标志。由硬件在方式 0 串行发送第 8 位结束时置“1”，或在其他方式串行发送停止位的开始时置“1”。该位必须由软件清“0”。

RI 为接收中断标志。由硬件在方式 0 接受到第 8 位结束时置“1”，或在其他方式串行接收到停止位的中间时置“1”。该位必须由软件清“0”。

特殊功能寄存器 PCON(地址为 87H):在 PCON 寄存器中,只有一位与串行口工作有关。其格式如下：



串行口工作于方式1，方式2和方式3时，数据传送的波特率与 2^{Smod} 成正比，也就是说，当 $Smod=1$ 时，将使串行口传送的波特率加倍。

2.16.3 实验内容

8051单片机的RXD、TXD接线柱在POD51仿真板上，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，如图2.41所示。

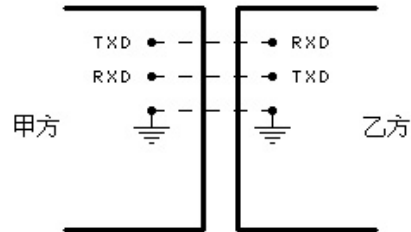


图2.41 双机通信连线示意图

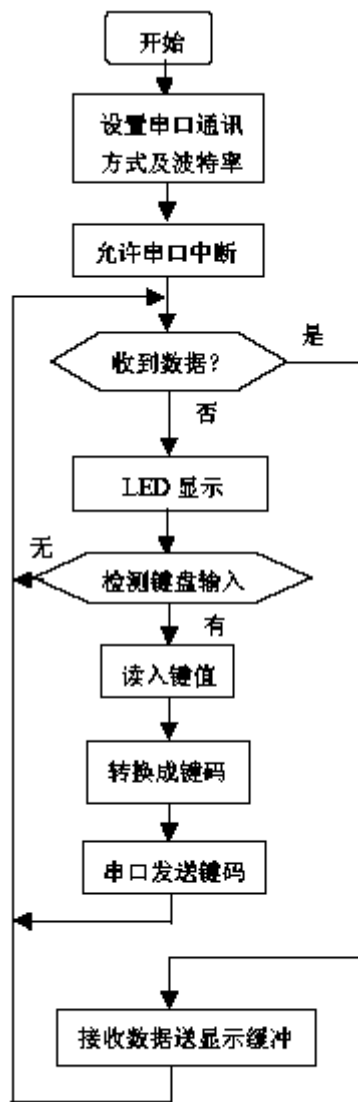
也可以将本机的TXD接到RXD上，这样按下的键，就会在本机LED上显示出来。若想与标准的RS232设备通信，就要做电平转换，输出时要将TTL电平换成RS232电平，输入时要将RS232电平换成TTL电平。可以将仿真板上的RXD、TXD信号接到实验板上的“用户串口接线”的相应RXD和TXD端，经过电平转换，通过“用户串口”接到外部的RS232设备。

按表2-19连线，编写并执行程序，按下一台实验系统的按键，注意观察另一台实验系统上数码管的显示值。

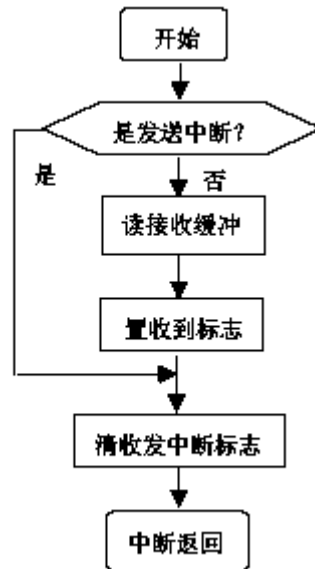
表 2-19

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

程序参考流程如下：



主程序



串口中断子程序

复习思考题

1. 复习实验过程，加深对串行工作原理的理解。
2. 修改实验程序实现下位机8051与上位机PC之间的串行通信。

2.12 实验十二 模拟空调温度控制实验

2.12.1 实验目的与要求

通过实验学习闭环控制的基本原理，进一步熟悉 A/D 转换器工作原理和编程方法，进一步了键盘扫描和 LED 动态显示原理和编程方法。利用实验系统提供的键盘、显示电路和 A/D 转换电路，完成类似空调恒温控制实验，采用实验仪系统上的电位器模仿温度变化，加热和致冷电机用发光二极管代替。要求用键盘设定恒温温度，当外界温度超过设定温度 ± 2 时，启动加热或致冷电机。

2.12.2 实验原理说明

这是一个综合硬件实验，其中各部分实验前面都已经单独做过，现联合起来形成一个控制系统。实验原理电路如图 2.42 所示，用电位器模拟温度采样，用 8255 的 PA0 模拟加热电机，PA1 模拟制冷电机，LED 显示电路和键盘电路在实验系统上已经接好，A/D 转换电路只要接上模拟量输入和地址选择信号即可。

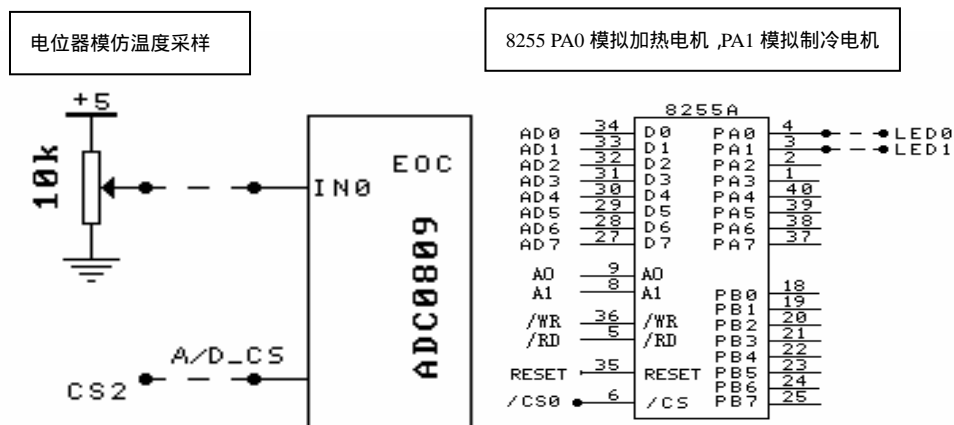


图 2.42 实验原理电路

2.17.3 实验内容

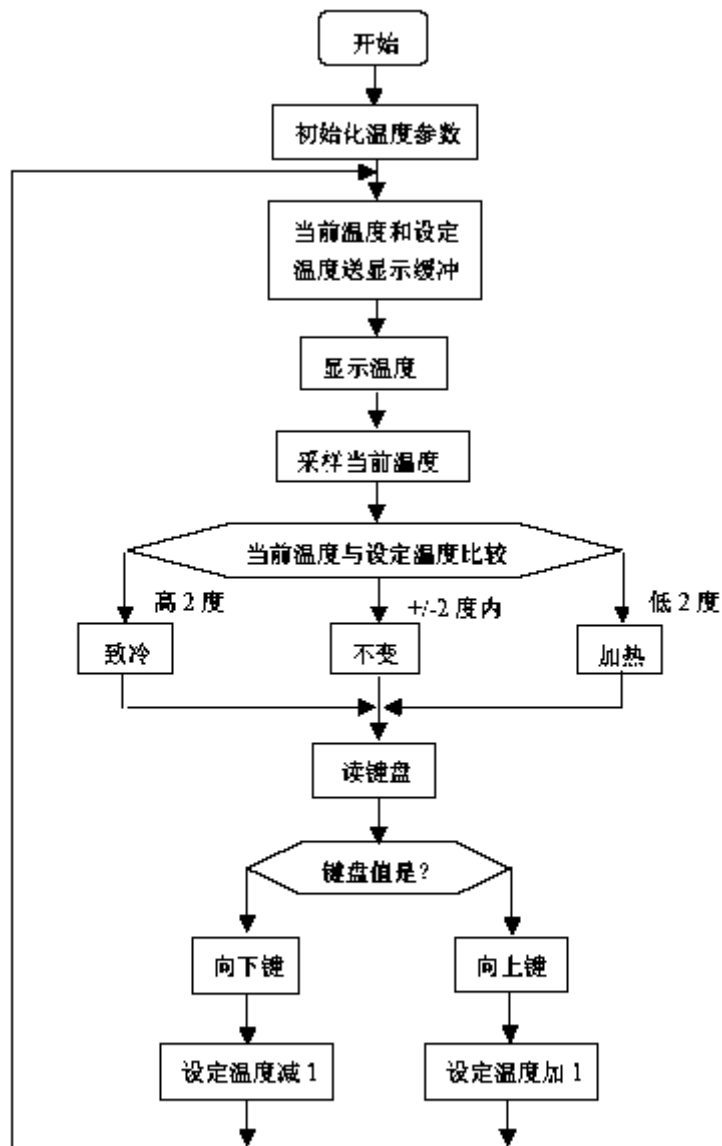
利用电位器和 A/D 转换器实现模拟空调温度控制

如表 2-20 所示连线，编写并执行程序，调节电位器，注意观察发光二极管状态的变化。

表 2-20

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

程序参考流程如下：



温度控制主程序框图

复习思考题

1. 复习实验过程，加深对空调温度控制工作原理的理解。
2. 如果不用8255而直接用8051的P1口控制加热和制冷电机，应如何修改实验程序？

附录 实验报告基本要求与书写格式

一、实验名称:

说明：本次实验的名称。

二、实验目的:

说明：本次实验的主要目的,参考每次的实验指导书。

三、实验环境:

说明：实验用到的硬件软件环境。

四、实验内容与步骤:

说明：实现实验目的而进行的实验内容，如果有步骤要求则简要列出步骤。

五、实验总结：

说明：1．画出硬件连接图和程序流程图。

2．参照本次实验的主程序，重新设计主程序并给出详尽注释。

3．本次实验得到了什么？收获是什么？有些什么别的想法？

六、建议与意见:

说明：对于此次实验内容或在实验过程中有任何问题或建议，以及对于改善实验效果有什么建议，均可提出。在书写实验报告的过程中，主要是帮助自己回顾和总结实验。重点放在第五部分，前四项可以十分简要地列写，第六项有则提出，无则不写。