

MCS-51 单片机应用

实验指导书

东南大学仪器科学与工程学院

2007 年

前 言

“XYZ22 型综合实验仪”适用于测控类及弱电类专业大学本科生以下课程的实验和实践教学：《MCS-51 单片机原理、接口及应用》、《检测（测控）技术及系统设计》、《智能仪器原理与设计技术》等；也适用于测控类及弱电类专业硕士研究生以下课程的实验和实践教学：《智能仪器设计》、《智能测控系统设计》等；也可以被相关专业的大专生选用。

本实验装置采用模块化设计，将有相互联系的专业课程的实验组合起来，包含这些专业课程要求的若干单个实验，最后形成“从传感器→前置调理、接口电路→A/D 转换电路→计算机（单片机）→后置接口电路→测控对象”接近实际应用系统的软硬件设计调试综合实验。训练和提高学生在检测（测控）技术及智能技术应用系统方面的设计调试能力。同时，帮助学生将各专业课程内容综合起来，融会贯通，形成系统的概念，迅速迈过从理论到实际的门槛。

大多数实验的实验内容都设置了基本要求和高级要求，能够满足不同层次学生的需要。

MCS-51 系列单片微型计算机现在被广泛应用的主要有四个产品：8051（80C51）、8031（80C31）、8751（87C51）和 8951（89C51）。它们之间的主要差别在于：

- 8051 片内有 4K（8 位）ROM 的程序存储器；
- 8751 片内有 4K（8 位）EPROM 的程序存储器；
- 8951 片内有 4K（8 位）EEPROM 的程序存储器；
- 8031 片内无程序存储器，必须外接程序存储器。

8051 单片机有三个八位准双向口 P1、P2、P3 口，一个双向总线 P0 口。P2 口和 P0 口一般作为系统扩展的地址线和数据线。P1 口作为 I/O 口。P3 口为多功能口，既可作为 I/O 口，又可作为特殊的第二功能口。

8051 单片机有二个十六位的可编程定时器/计数器 T0 和 T1，有 4 种操作方式。可用于定时、外部计数及串行口波特率发生器。

8051 单片机有一全双工串行口，也有 4 种工作方式。

8051 单片机有五个中断源。中断入口地址为：

- 0003H 外中断 0（/INT0）中断
- 000BH 定时器/计数器 0（T0）中断
- 0013H 外中断 1（/INT1）中断
- 001BH 定时器/计数器 1（T1）中断
- 0023H 串行口（SIO）接收、发送中断

8051 单片机有 128 个字节的内部 RAM 存储器。另有 21 个特殊功能寄存器，供 CPU 使用。

8051 单片机有 111 条指令，有 7 种寻址方式。

8051 单片机允许的晶振频率为 1.2M~12MHZ，通常使用 6MHZ 和 12MHZ。

8051 单片机可扩展外部数据存储器 and 外部程序存储器，扩展输入和输出接口，外接键盘、显示器、打印机及其它外部设备，广泛应用在中、小型测控系统和智能仪器、仪表中。

祝学云

目 录

第一章 实验设备介绍 1

- 一. 实验设备的联接..... 1
- 二. 伟福 (WAVE) G6W 型单片机仿真开发器介绍..... 1
- 三. XYZ22 型综合实验仪介绍..... 2

第二章 程序设计实验 11

- 实验一 数据传送实验..... 11
- 实验二 多字节十进制加法实验..... 12

第三章 I/O 口 (8255) 扩展及外部中断实验 14

- 实验三 8255 输入输出实验..... 14
- 实验四 外部中断实验..... 16

第四章 LED 显示器及行列式键盘扩展实验 18

- 实验五 动态 LED 显示实验..... 18
- 实验六 键盘显示器扩展实验..... 22

第五章 计数器定时器实验 26

- 实验七 计数器实验..... 26
- 实验八 定时器 (时钟) 实验..... 29

第六章 串行通信实验 33

- 实验九 串行通信实验..... 33

第七章 D/A 及 A/D 实验 39

- 实验十 D/A 转换实验 (DAC0832) 39
- 实验十一 A/D 转换实验 (ICL7135) 41

第一章 实验设备介绍

一. 实验设备的联接

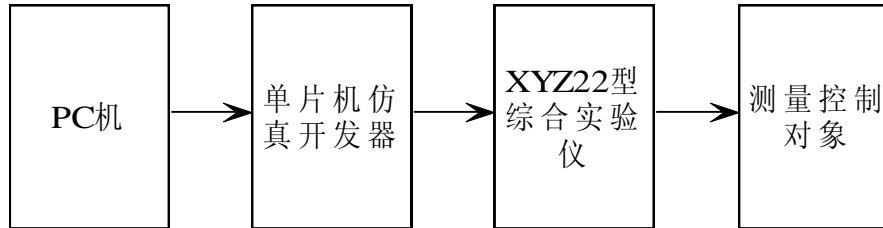


图 1 设备联接

二. 伟福 (WAVE) G6W 型单片机仿真开发器介绍

G6W 型单片机仿真开发器含有 WINDOWS、DOS 版本双平台；
可选择不同配置的仿真头（仿真不同的 CPU）；
仿真头上有一个晶振跳线器（选用内、外晶振），出厂设置选用内晶振；
打开电源时，先接通仿真器电源，再接通外部用户板（实验仪）的电源。防止 CMOS 型的 CPU 芯片过流，造成芯片温度过高。

1. 安装及基本参数设置

1) 安装

进入 C:\ 盘符下，建立子目录 C:\51,将编译器安装到 C:\51 目录下即可。

2) 基本参数设置

选择伟福汇编器；

选择 G6W 及 POD-51 型仿真头、8031 CPU；

选择程序空间在仿真器上、数据空间在用户板上；

选择仿真器与 PC 机连接的串行端口。

2. 编辑、编译、执行及修改文件

1) 编辑及修改

新建一个文件或打开一个文件，对此文件进行编辑及修改，最后将其保存。
汇编文件的文件名后缀为 .ASM。

3) 编译

对当前窗口文件进行编译、校验，如果程序文件有错，信息窗口将提示出错信息，以使用户修改。

4) 执行与退出

包括全速、跟踪、单步、执行到光标处等几种程序执行方式。

- 在窗口中检查 RAM 及特殊功能寄存器的内容
全速运行程序后，要先“暂停”再检查存储器内容。
CPU 窗口包含源程序的反汇编代码、特殊功能寄存器状态等。
数据窗口包含内部 RAM 和外部 RAM 的状态等。

三. XYZ22 型综合实验仪介绍

一) 面板功能块分布图

电 源 +2.5V,+5V,+12V,0V,-12V,-5V WR1 WR2	8031扩展8255 (1) 电路 8031: /INT0、/INT1、T0、 T1、/WR、/RD 8255 (1) : /CS、PA□、 PB□、PC□ 74138: /Y3~/Y7 74393: 1/4ALE、1/8ALE (Qin、Q0Q1Q2Q3)	六位LED显示器	
		8255 (2) : /CS (扩展显示器键盘)	
测量电桥 输入 RX 输出 Vout11 Vout12	测量放大器 输入 Vin4, Vin42 输出 Vout4	RS-422 电路 IN (+/-) OUT (+/-)	控制输出驱动电路 输入 CONIn 输出 CONout (+/-)
电压衰减器 输入 Vin2 输出 Vout2	ICL7135 电路 输入/输出: UNDER、 OVER、CLKIN、 BUSY、Vin5、 D5~D1、 B8~B1	电容/频率电路 输入 Cxin (+/-) 输出 Fout	工作 9 暂停 6 设定 1 定义 1 定义 2 E D
DAC0832 电路 输入/输出: /CS、 /WR、RFB、VREF、 Vout3、FG、D7~D0。	钮子开关电路 K1~K8	LED 发光二极管电路 L1~L8	单脉冲电路 (按钮K)

图 2 面板功能块分布图

二) 功能块分类

1. 电源部分

外接+5V, +12V, -12V 电源。

实验板上有以下电源插孔: +2.5V, +5V, +12V, -5V, -12V 及 0V (GND)。

实验板上每一功能块电源独立控制, 由短路块连通。

2. 测量电桥

电源+2.5V。

R_x 为热敏电阻 Pt100(调试时接电阻箱)。

输出 V_{out11}、V_{out12}。

WR3 为调零电位器。

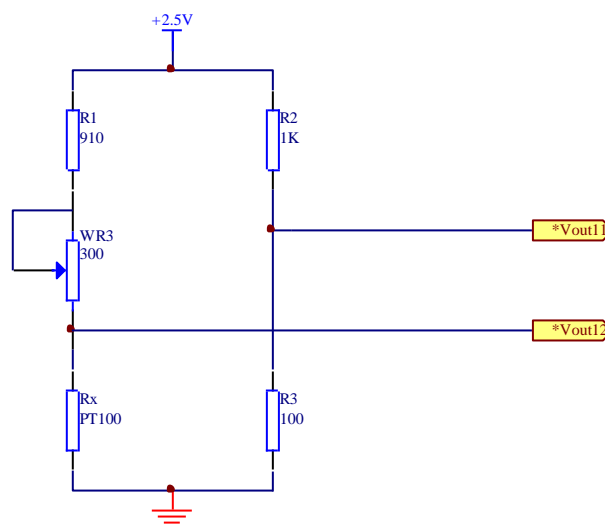


图 3 测量电桥

3. 测量放大器

电源+12V, -12V。

输入 Vin41、Vin42, 输出 Vout4。

WR4 为增益调整电位器, WR5 为调零电位器。

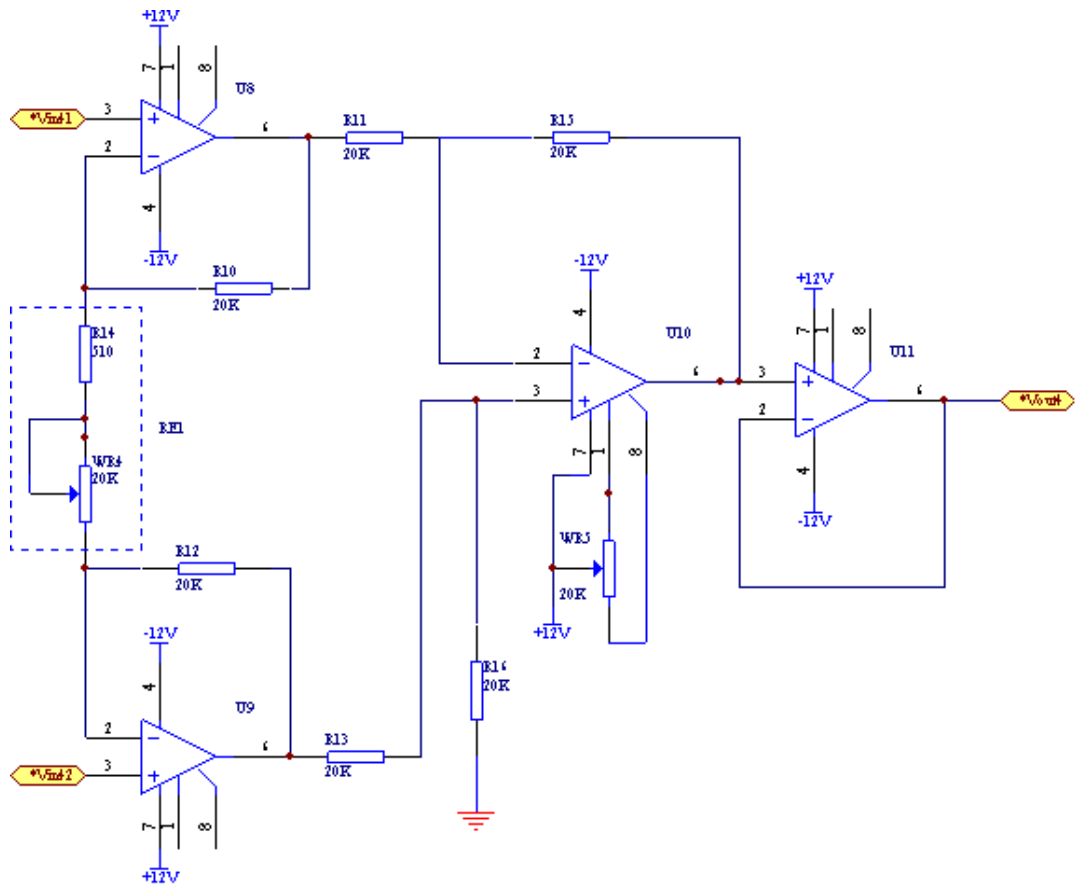


图 4 测量放大器

4. 电压衰减器

电源+12V, -12V。

输入 Vin2, 输出 Vout2。

衰减倍数: 1/6 (J01 通)
 1/3 (J02 通)
 1/2 (J03 通)

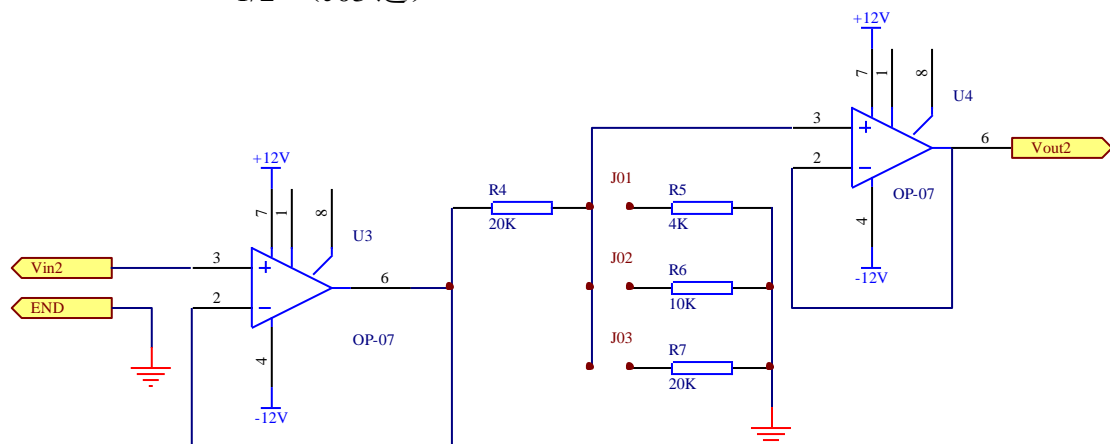


图 5 电压衰减器

5. 四位半 A/D 转换电路(ICL7135)

电源+2.5V, 5V, -5V。

输入 V_{in5} 。使用单端输入 INH1, INH0 (负端输入) 已接地。输入范围为 0-1.999V, 对应的输出数据为 00000-19999。

转换状态信号 BUSY (A/D 转换时为高电平, 转换结束时为低电平)。可供查询或中断申请用。

时钟 CLKIN 接 1/8 ALE, 单片机晶振频率为 12MHZ, 则 1/8 ALE 为 250KHZ, 转换频率约为 6 次/秒。转换控制端 R/H 已接高电平, 表示 ICL7135 将连续自动转换。

当输入信号超过 1.999V 时, 过量程标志 OVER 输出高电平。

当输入信号小于量程的 9% (0.1800V) 时, 欠量程标志 UNDER 输出高电平。但在单端输入时, 输入信号在 0V-0.1800V 时并不影响 A/D 转换器的正常工作。

位状态输出 D5、D4、D3、D2、D1, 分别表示现时输出的数据为万、千、百、十、个位。

BCD 码数据输出为 B8、B4、B2、B1。

WR6 调整 ICL7135 第 2 脚 (标准电压端 VREF) 电压为 1.000V。

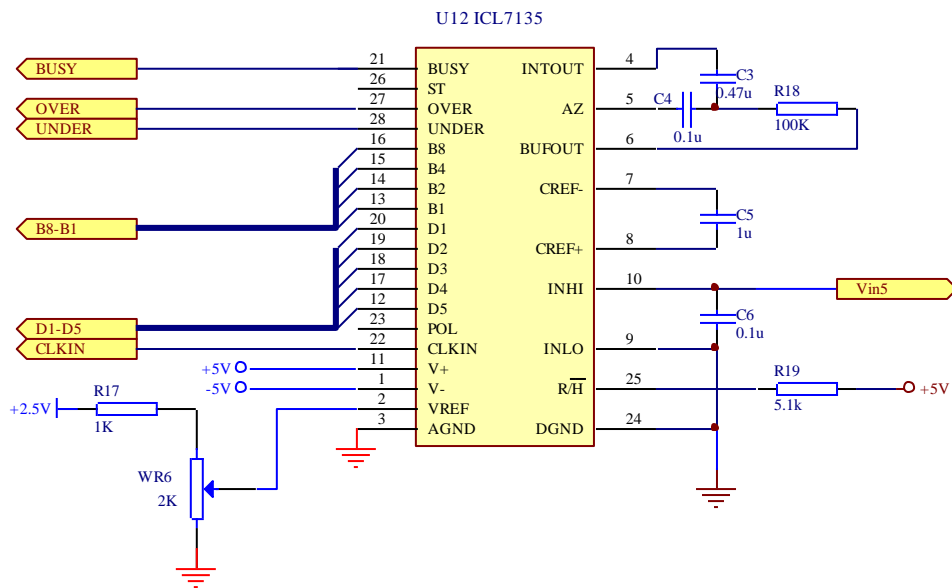


图 6 A/D 转换电路(ICL7135)

6. D/A 转换电路 (DAC0832)

电源+5V, +12V, -12V。

输入 D7~D0(00H-FFH), 输出 V_{out3} (0-5V)。

写信号/WR, 片选信号/CS。

标准电压端 VREF, 反馈电阻端 RFB, 反馈端 FG。

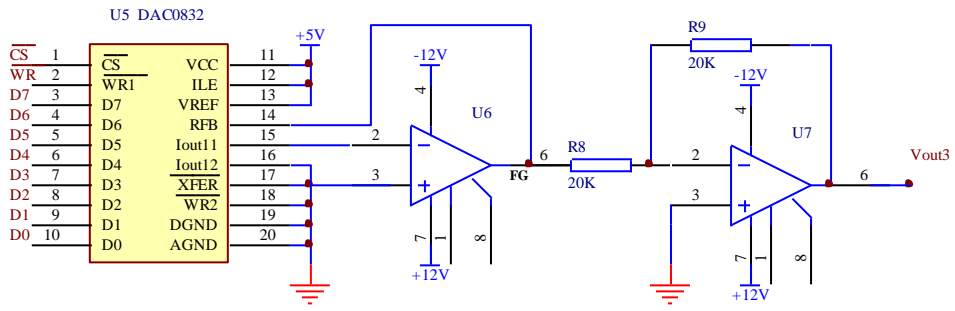


图7 D/A 转换电路 (DAC0832)

7. 钮子开关电路

电源+5V。输出为 K1~K8。

钮子开关拨向上方输出为低“0”，拨向下方输出为高“1”。

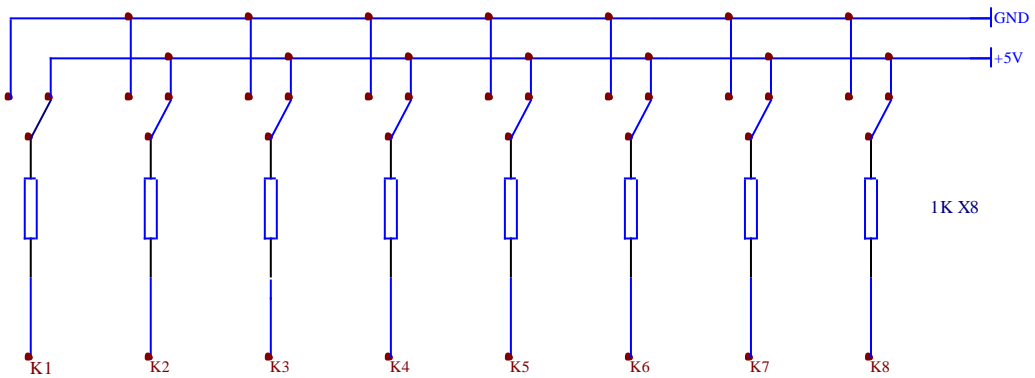


图8 钮子开关电路

8. LED 发光二极管电路

电源+5V。输入为 L1~L8。

输入为高时发光二极管亮，输入为低时发光二极管灭（7406 为反向驱动器）。

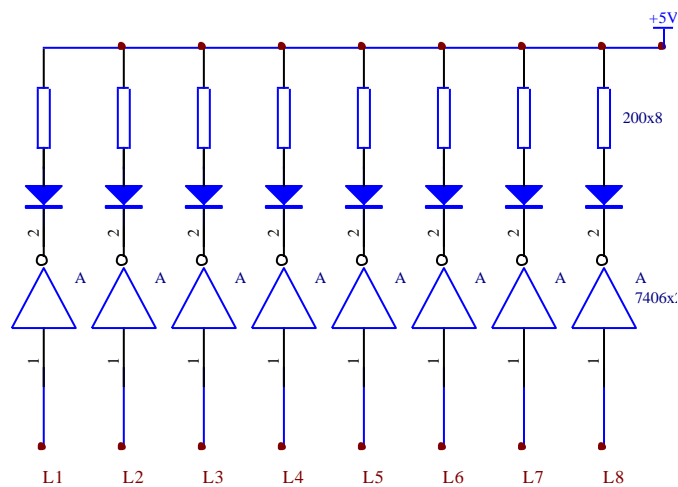


图9 LED 发光二极管电路

9. 单脉冲输出电路
电源+5V。K 为按钮。

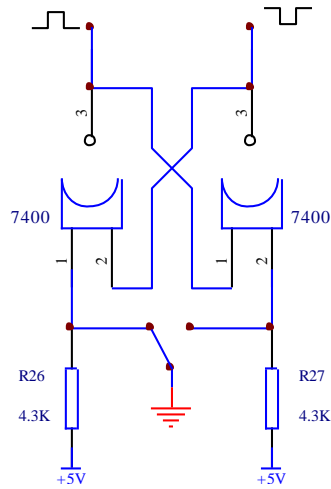


图 10 单脉冲输出电路

10. 8031 扩展 8255 (1)

电源+5V。

8031 引出端 /INT0、/INT1、T0、T1、/WR、/RD。

8255 (1) 片选信号为 /CS，PA 口为 PA7~PA0，PB 口为 PB7~PB0，PC 口为 PC7~PC0。

译码电路 74138 输出 /Y3 ~ /Y7，地址范围为 6000H~FFFFH。

双分频电路 74393 一路输出为 1/4 ALE、1/8 ALE (输入已接 ALE)；另一路输入为 Qin，输出为 Q0~Q3 (2 分频~16 分频)。

原理图见下页 (图 11)。

11. 利用 8255 (2) 扩展 LED 显示器及行列式键盘

电源+5V。

8255 (2) 片选信号为 /CS。

LED 显示器为共阴极，MC1413 为反向驱动器，三极管 9013 为正向驱动。

原理图见下页 (图 12)。

12. RS-422 串行口

电源+5V。输入为 IN (+/-)，输出为 OUT(+/-)。

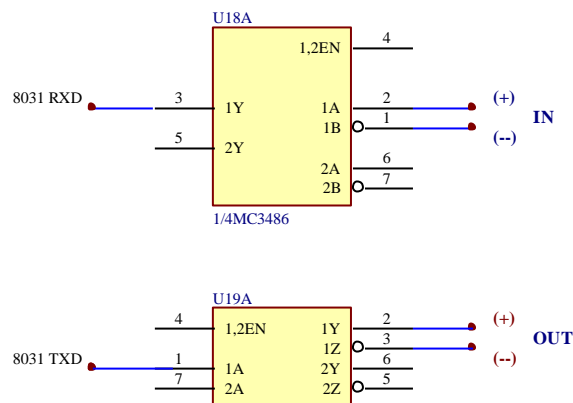


图 13 RS-422 串行口

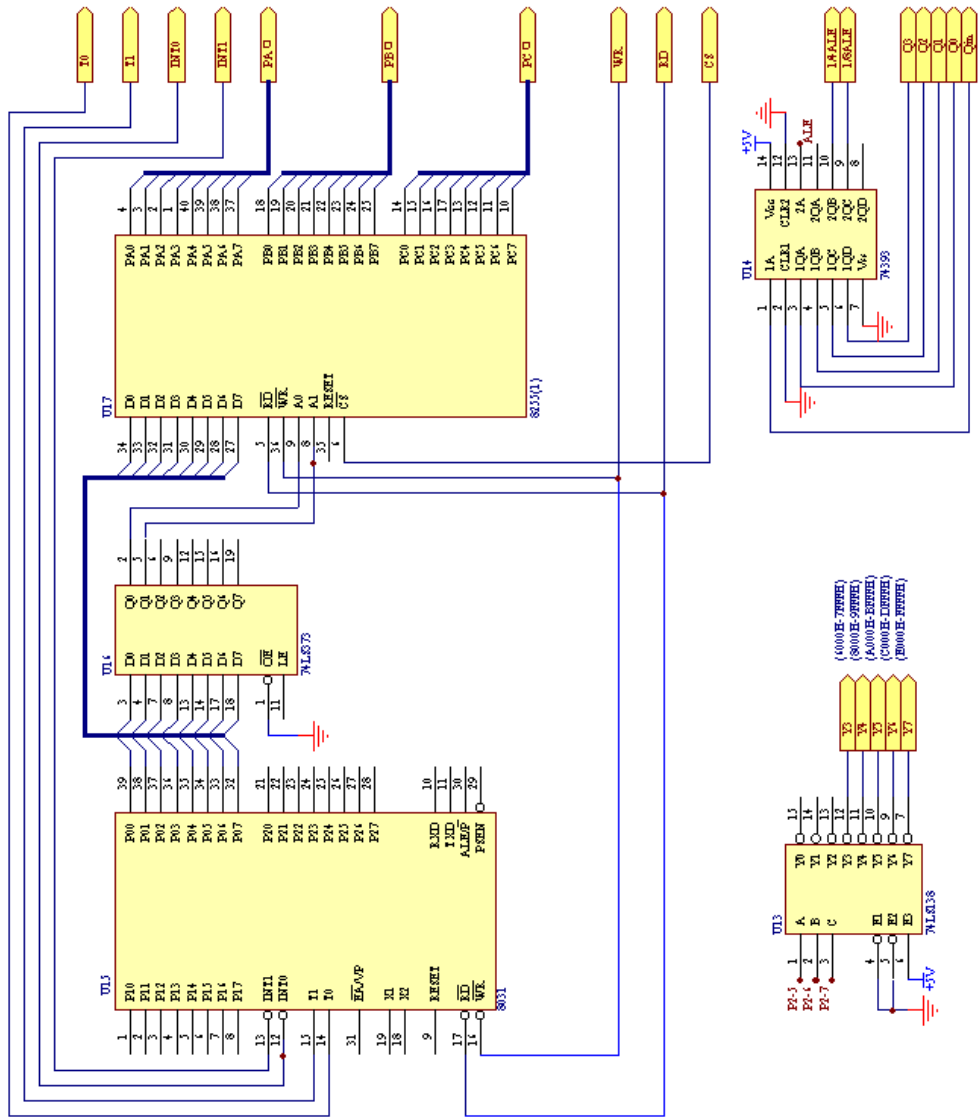


图 11 8031 扩展 8255 (1) 原理图

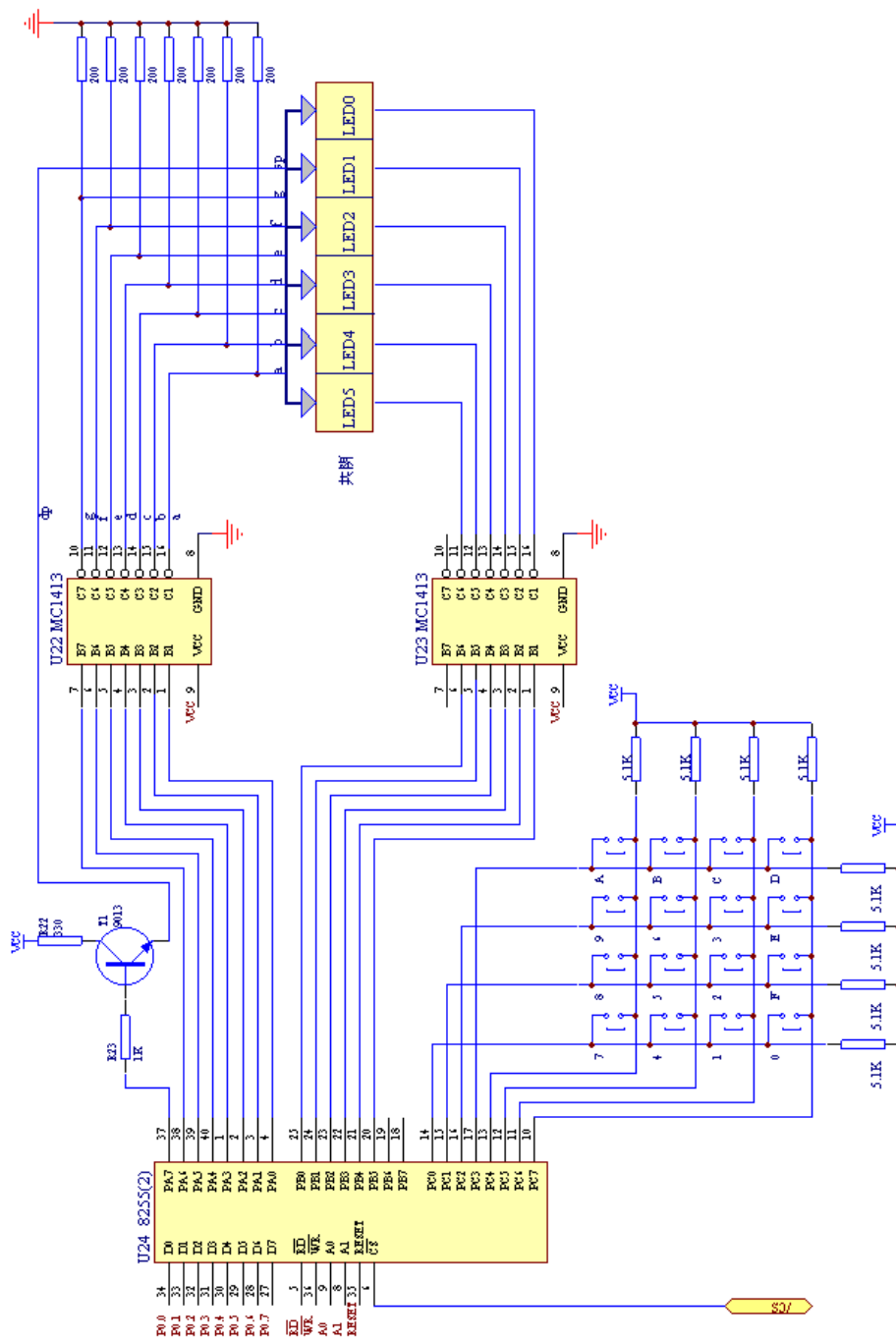


图 12 8255 (2) 扩展 LED 显示器及行列式键盘原理图

13. 固态继电器（负载）控制电路

电源+5V。

输入为 CONIn，输出为 CONout(+/-)。

控制信号 CONIn 为高时，负载接通电源。

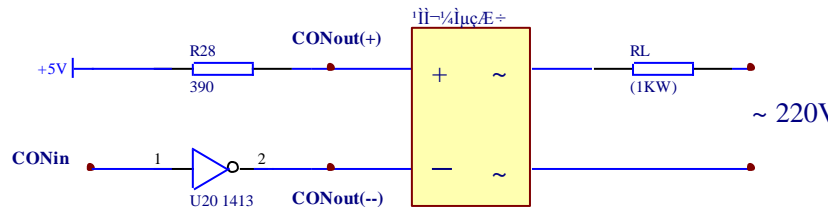


图 14 固态继电器控制电路

14. 555（电容/脉冲）电路

电源+5V。

Cxin (+/-) 为电容，输出为 Fout。

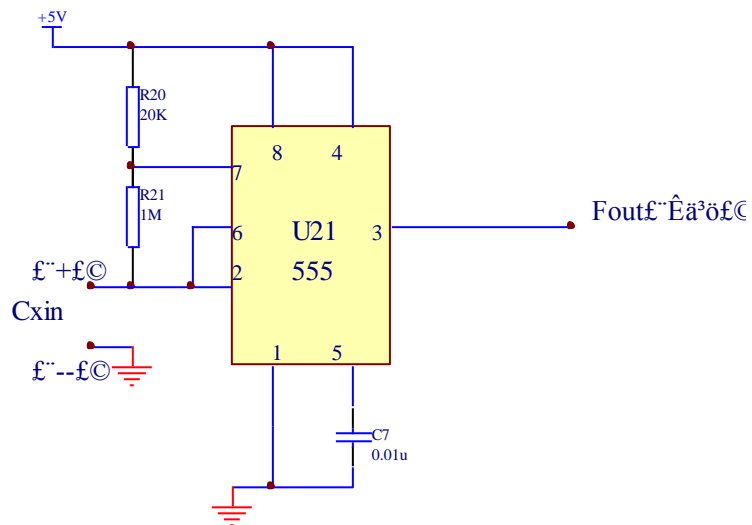


图 15 555 脉冲电路

第二章 程序设计实验

实验一 数据传送实验

一. 目的：熟悉单片机仿真开发器的使用；

掌握 8031 内部 RAM 的数据操作，了解存储器的特点和应用。

二. 内容：内部 RAM 40H~4FH 单元置数 A0H~AFH，然后将 40H~AFH 单元中内容送到内部 RAM 50H~5FH 单元中。

三. 编辑、编译、运行程序，检查结果。

四. 参考程序：

```
                ORG 0000H
RESET:          AJMP MAIN
                ORG 003FH
MAIN:           MOV R0, #40H
                MOV R2, #10H
                MOV A, #0A0H
A1:             MOV @R0, A
                INC R0
                INC A
                DJNZ R2, A1
                MOV R0, #40H
                MOV R1, #50H
                MOV R2, #10H
A2:             MOV A, @R0
                MOV @R1, A
                INC R0
                INC R1
                DJNZ R2, A2
A3:             SJMP A3
```

五. 分析程序。

六. 运行程序后（暂停），检查并记录相关单元内容。

七. 按以上步骤，验证教材中程序设计的例题。

实验二 多字节十进制加法实验

- 一. 目的：掌握 MCS-51 汇编语言程序设计方法。
- 二. 内容：多字节的十进制加法，加数存储单元首地址由 R0 指出，被加数和结果的存储单元首地址由 R1 指出，字节数由 R2 指出。
- 三. 编辑、编译、运行程序，检查结果。
- 四. 参考程序：设加数存储单元为：31H、30H，被加数存储单元为：21H、20H，结果存储单元为：22H、21H、20H。

```
                ORG 0000H
RESET:  AJMP MAIN
                ORG 0100H
MAIN:    MOV SP, #60H
         MOV R0, #31H
         MOV @R0, #22H
         DEC R0
         MOV @R0, #33H
         MOV R1, #21H
         MOV @R1, #44H
         DEC R1
         MOV @R1, #55H
         MOV R2, #02H
         ACALL DACE
HERE:   SJMP HERE
DACE:   CLR C
DAL:    MOV A, @R0
         ADDC A, @R1
         DA A
         MOV @R1, A
         INC R0
         INC R1
         DJNZ R2, DAL
         CLR A
         MOV ACC. 0, C
         MOV @R1, A
         RET
```

五. 分析程序。

六. 运行程序后（暂停），检查并记录相关单元内容。

修改加数和被加数（十进制数），重复以上过程。

七. 填写下列单元地址及内容：

加数单元地址 () ()

加数单元地址 () ()

结果单元地址 () () ()

加数数值 _____

加数数值 _____

结果数值 _____

八. 按以上步骤，验证作业中程序设计的习题。

第三章 I/O 口（8255）扩展及外部中断实验

实验三 8255 输入输出实验

一. 目的:

掌握 8031 单片机并行口扩展方法；熟悉 8255 芯片性能，掌握其编程方法。

二. 内容: 8255（1）从 PA 口输入，从 PB 口输出。

三. 硬件连接框图（图 16）:

相关电路原理见第一章图 8，图 9 和图 11。

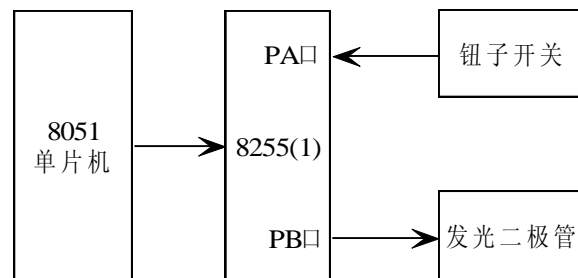


图 16

四. 实验步骤:

- 1、硬件连接： 钮子开关 K1~K8 连接 8255（1）的 PA0~PA7；
发光二极管 L1~L8 连接 8255（1）的 PB0~PB7；
8255（1）片选信号 /CS 连接 /Y3 ~/Y7（任选其一）；
将相关功能块的电源接通。
- 2、软件设计： 8255（1）初始化，数据从 PA 口输入，取反后从 PB 口输出。

3、执行程序，观察并记录实验结果

五. 参考程序框图：（图 17）

六. 查询资料，了解其它常用的并行 I/O 口芯片。

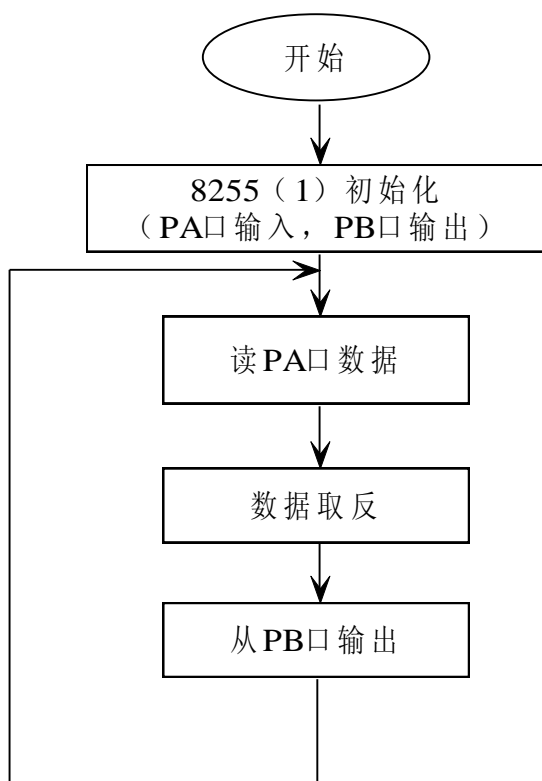


图 17 参考程序框图

实验四 外部中断实验

一. 目的:

了解单片机中断原理、中断过程、中断方式的选择及编程方法。

二. 内容:

利用单脉冲输出电路的脉冲,通过 $\overline{INT0}$ 向 CPU 申请外部中断, CPU 响应中断后,执行中断服务子程序。

中断服务子程序内容为:从 8255 (1) PA 口输入数据,取反后从 PB 口输出。

三. 硬件连接框图 (图 18):

相关电路原理图见第一章图 8,图 9,图 10 和图 11。

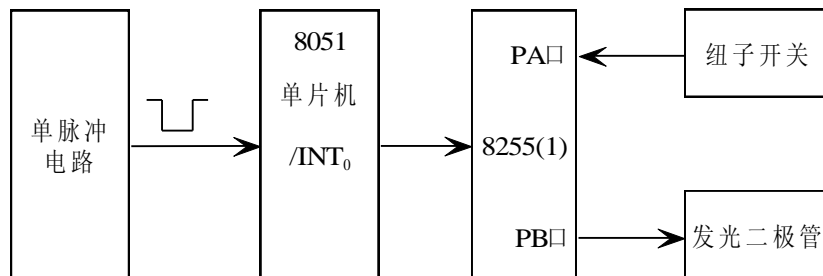


图 18

四. 实验步骤:

1、 硬件连接:

钮子开关 K1~K8 连接 PA0~PA7; 发光二极管 L1~L8 连接 PB0~PB7; 8255 (1) 片选信号 \overline{CS} 连接 $\overline{Y3} \sim \overline{Y7}$ (任选其一);

单脉冲输出电路负脉冲输出端连接 $\overline{INT0}$;

将相关功能块的电源接通。

2、 软件设计：

主程序中等待外部中断；

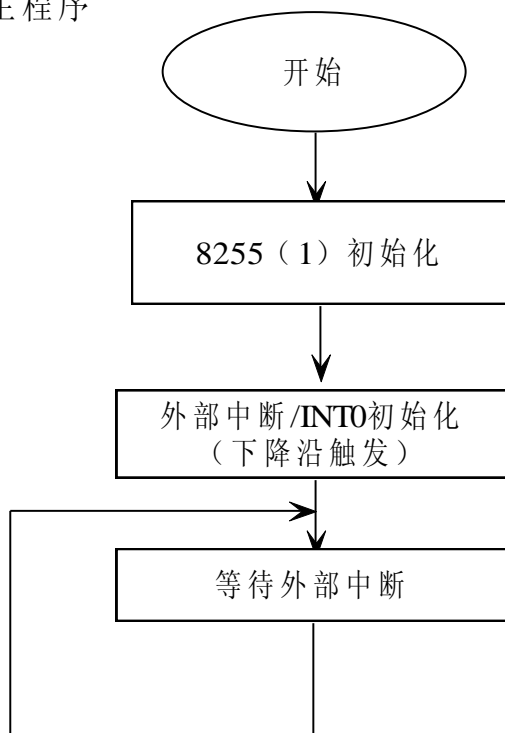
中断服务子程序中数据从 PA 口输入，取反后从 PB 口输出。

3、 执行程序，观察并记录实验结果。

五、 参考程序框图（图 19）：

六、 8051 单片机 5 个中断源的中断矢量入口地址？哪些中断申请标志由 CPU 自动清除？哪些中断申请标志必须由软件清除？

主程序



中断服务程序

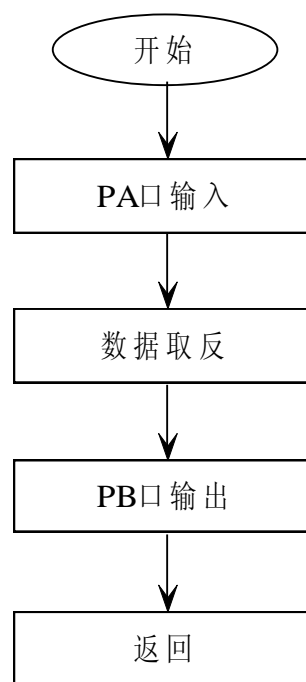


图 19 参考程序框图

第四章 LED 显示器及行列式键盘扩展实验

实验五 动态 LED 显示实验

一. 目的:

1. 了解动态 LED 显示器电路的结构。
2. 通过实验掌握动态 LED 显示器显示程序设计方法。

二. 内容:

设计程序在 LED 显示器上显示字符“1 2 3 4 5 6”。

三. 硬件连接框图 (图 20):

LED 显示器电路原理图见第一章图 12。

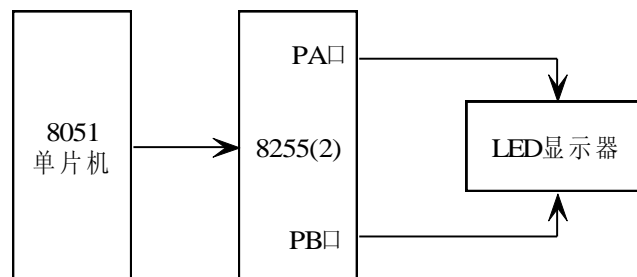


图 20

四. 实验步骤:

1、 硬件连接:

8255 (2) 片选信号 /CS 连接 /Y3 ~ /Y7 (任选其一);

将相关功能块的电源接通。

2、 软件设计:

1) 8255 (2) PA 口 PA7 控制小数点, PA6~PA0 控制字形 (经反向驱动输出); 8255 (2) PB 口 PB5~PB0 控制字位 (经反向驱动输出)。

LED 显示器为共阴极。

2) 2ms 软件延时子程序:

```
DELAY2MS:    MOV R3, #04H
              DELAY0:    MOV R4, #0FFH
              DELAY1:    DJNZ R4, DELAY1
                              DJNZ R3, DELAY0
                              RET
```

3) 字形码 (0~F):

```
DB  40H, 79H, 24H, 30H
DB  19H, 12H, 02H, 78H
DB  00H, 10H, 08H, 03H
DB  46H, 21H, 0EH, 41H
```

五. 参考程序框图: (见下页图 21)

六. 参考程序清单: (见下页附录一)

七. 高级要求: 显示小数点。在左边第 3 位上增加小数点的显示, 该位字形码如何确定? 修改程序, 观察结果。

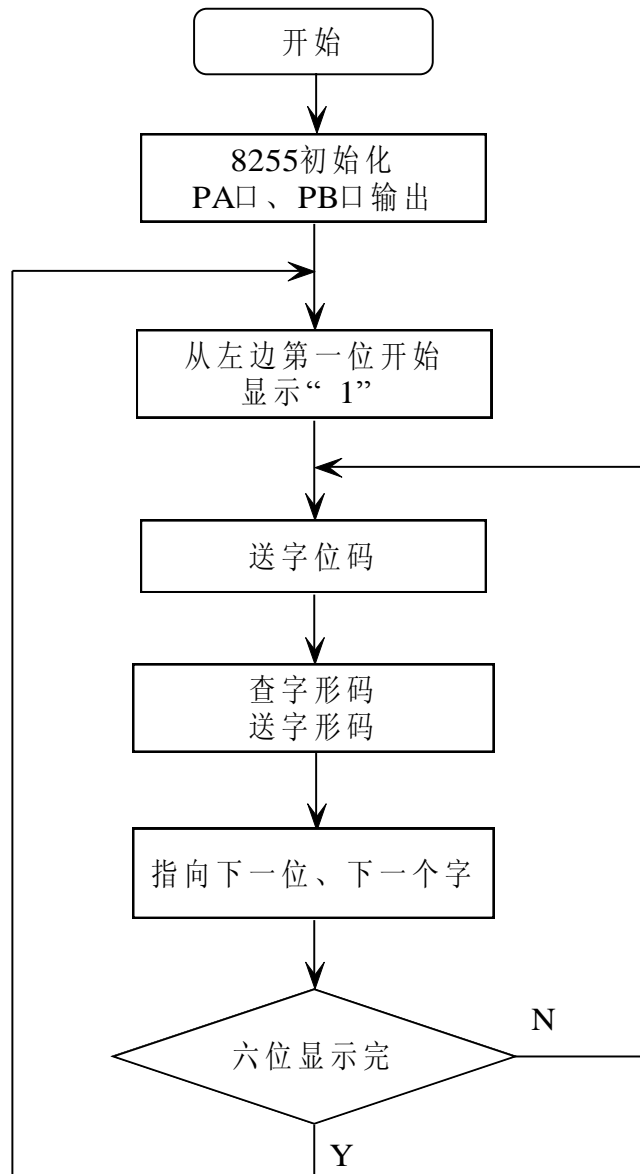


图 21 程序框图

附录一（参考程序清单）：

8255(2) /CS 接 /Y3(6000H)。

```

org 0000h
ajmp main
org 0100h
main:  mov dptr, #6003h    ;8255(2) 初始化
      mov a, #80h       ;PA 口 PB 口输出
      movx @dptr, a     ;
  
```

```

loop2:  mov r7, #06h           ;LED 显示六位
        mov r6, #01h         ;从左边第一位开始显示
        mov r5, #01h         ;从“1”开始显示
loop1:  mov a, r6              ;PB 口送位码
        mov dptr, #6001h     ;
        movx @dptr, a        ;
        mov a, r5            ;查字形码
        mov dptr, #dsp       ;
        movc a, @a+dptr      ;
        mov dptr, #6000h     ;送字形码
        movx @dptr, a        ;
        acall delay2ms       ;延时 2MS
        mov a, r6            ;指向下一位
        rl a                  ;
        mov r6, a            ;
        inc r5                ;显示下一个数
        djnz r7, loop1       ;六位显示完?
        ajmp loop2           ;六位显示完, 重新开始
delay2ms: mov r3, #04h       ;2MS 延时子程序
delay0:  mov r4, #0ffh       ;
delay1:  djnz r4, delay1     ;
        djnz r3, delay0     ;
        ret                  ;
dsp:     db 40h, 79h, 24h, 30h ;字形表
        db 19h, 12h, 02h, 78h
        db 00h, 10h, 08h, 03h
        db 46h, 21h, 06h, 0eh

```


实验六 键盘显示器扩展实验

一. 目的:

1. 了解键盘显示器扩展电路的结构。
2. 熟悉行列式键盘输入程序设计方法。

二. 内容:

从键盘上输入 0~9 及 A~F, 在 LED 显示器上显示。

三. 硬件连接框图 (图 22):

键盘显示器原理图见第一章图 12。

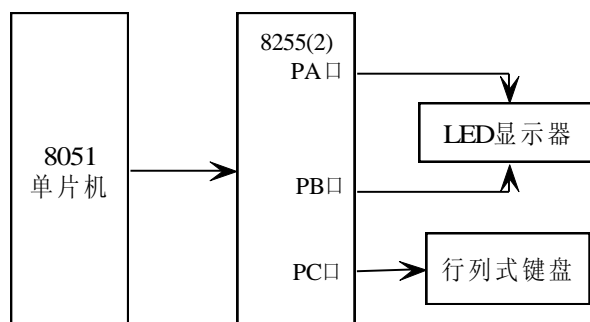


图 22

四. 实验步骤:

1、 硬件连接:

8255 (2) 片选信号 /CS 连接 /Y3 ~ /Y7 (任选其一);
将相关功能块的电源接通。

2、 软件设计:

1) 8255 (2) PA 口 PA7 控制小数点, 8255 (2) PA 口 PA6~PA0 控制字形 (经反向驱动输出); 8255 (2) PB 口 PB5~PB0 控制字位 (经反向驱动输出)。LED 显示器为共阴极。

2) 8255 (2) PC 口 PC7~PC4 为行列式键盘的行线, PC3~PC0 为行列式键盘的列线。行线、列线均由上拉电阻接+5V。

3) 行线输出全零时读列线输入; 列线输出全零时读行线输入。

4) 键值表 (0~9 及 A~F 键按下时, PC7~PC4 、 PC3~PC0 输入合并后的数值)

```
DB 7EH, 0BEH, 0BDH, 0BBH
DB 0DEH, 0DDH, 0DBH, 0EEH
DB 0EDH, 0EBH, 0E7H, 0D7H
DB 0B7H, 77H, 7BH, 7DH
```

五. 参考程序清单：（见附录二）

六. 高级要求：查询资料，举出其它形式的键盘、显示器设计方法。分析优缺点。

附录二（参考程序清单）：

8255(2) /CS 接 /Y3(6000H)。

```
ORG 0000H
AJMP START
ORG 0100H
START: MOV A, #06H      ;六位显示缓冲区（75H-7AH）清零
      MOV R0, #75H      ;
CLE:   MOV @R0, #00H;
      INC R0;
      DEC A;
      JNZ CLE;
KEY:   ACALL LED        ;调显示子程序
      MOV DPTR, #6003H ;扫描键盘。8255（2）初始化
      MOV A, #88H      ;PA口PB口输出（LED显示器）
      MOVX @DPTR, A    ;PC口高四位输入低四位输出（行列式键盘）
      MOV DPTR, #6002H ;PC口低四位输出零
      MOV A, #00H      ;
      MOVX @DPTR, A    ;
      MOVX A, @DPTR    ;PC口高四位输入
      ANL A, #0F0H     ;
      MOV 74H, A       ;键值高四位送74H
      CJNE A, #0F0H, READ ;键值高四位非全“1”有键按下
      AJMP KEY         ;键值高四位全“1”无键按下
READ:  MOV DPTR, #6003H ;有键按下。8255（2）初始化（行列输入输出
      MOV A, #81H      ; 交换），PA口PB口输出（LED显示器），
      MOVX @DPTR, A    ;PC口高四位输出低四位输入（行列式键盘）
      MOV DPTR, #6002H ;PC口高四位输出零
      MOV A, #00H      ;
      MOVX @DPTR, A    ;
      MOVX A, @DPTR    ;PC口低四位输入
      ANL A, #0FH      ;
      ADD A, 74H       ;键值低四位送74H
      MOV 74H, A       ;
      ACALL DELAY12MS ;软件延时
KEY1:  ACALL LED
      MOV DPTR, #6003H ;再扫描一次
```

```

MOV A, #88H          ;
MOVX @DPTR, A       ;
MOV DPTR, #6002H    ;
MOV A, #00H         ;
MOVX @DPTR, A       ;
MOVX A, @DPTR       ;
ANL A, #0F0H        ;
MOV 73H, A          ;
MOV DPTR, #6003H    ;
MOV A, #81H         ;
MOVX @DPTR, A       ;
MOV DPTR, #6002H    ;
MOV A, #00H         ;
MOVX @DPTR, A       ;
MOVX A, @DPTR       ;
ANL A, #0FH         ;
ADD A, 73H          ;
CLR C                ;两次值相同去再扫描一次（按键未松）
SUBB A, 74H         ;
JZ KEY1             ;
MOV R1, #00H        ; 按键已松。R1 作计数器
MOV R5, #10H        ; 16 个键
SEARCH: MOV DPTR, #KTABLE ;键值表首址
MOV A, R1           ;
MOVC A, @A+DPTR    ;查键表值
CLR C              ;
SUBB A, 74H        ;比较读入键值与键值表数据
JZ FIND            ;相同则键值有效，去处理显示缓冲区
INC R1             ;不相同比较下一个键值表数据
DJNZ R5, SEARCH   ;
AJMP KEY           ;16 个键值表数据比较完
FIND:  MOV A, R1    ;键值有效准备显示
MOV R5, A          ;R1 数值与键盘标称值（0-F）相同
MOV R7, #05H      ;
MOV R0, #79H      ;显示缓冲区（7AH-75H）数据向前移位
MOV R1, #7AH      ;
ROL:  MOV A, @R0   ;
MOV @R1, A        ;
DEC R0            ;
DEC R1            ;
DJNZ R7, ROL     ;
MOV 75H, R5      ;新读键盘标称值（0-F）送显示缓冲区 75H
AJMP KEY         ;

```

```

LED:    MOV DPTR, #6003h ;显示子程序
        MOV A, #80h      ;8255 (2) 初始化
        MOVX @DPTR, A    ;
SHOW:   MOV R0, #75h     ;显示缓冲区首地址
        MOV R2, #01h     ;位码
        MOV A, R2        ;
LOOP:   MOV DPTR, #6001H ;位码送 PB 口
        MOVX @DPTR, A    ;
        MOV DPTR, #6000H ;PA 口送全黑字形码
        MOV A, #7FH      ;
        MOVX @DPTR, A    ;
        MOV DPTR, #LTABLE ;字形表首地址
        MOV A, @R0        ;取显示缓冲区值
        MOVC A, @A+DPTR  ;取字形码
        MOV DPTR, #6000H ;字形码送 PA 口
        MOVX @DPTR, A    ;
        ACALL DELAY1MS   ;软件延时
        INC R0            ;指向下一显示缓冲区地址
        MOV A, R2        ;
        JB ACC. 5, LOOP1  ;第六位显示完?
        RL A              ;未完显示下一位
        MOV R2, A        ;
        AJMP LOOP        ;
LOOP1:  RET              ;第六位显示完返回

DELAY12MS: MOV R3, #0FFH ;
DLAY0:    MOV R4, #0FFH
DLAY1:    DJNZ R4, DLAY1
          DJNZ R3, DLAY0
          RET
DELAY1MS: MOV R3, #02H ;
DELAY0:    MOV R4, #0FFH
DELAY1:    DJNZ R4, DELAY1
          DJNZ R3, DELAY0
          RET
LTABLE:   DB 40H, 79H, 24H, 30H, 19H ;字形表
          DB 12H, 02H, 78H, 00H, 10H
          DB 08H, 03H, 46H, 21H, 06H
          DB 0EH
KTABLE:   DB 7EH, 0BEH, 0BDH, 0BBH ;键值表
          DB 0DEH, 0DDH, 0DBH, 0EEH
          DB 0EDH, 0EBH, 0E7H, 0D7H
          DB 0B7H, 77H, 7BH, 7DH
          END

```

第五章 计数器定时器实验

实验七 计数器实验

一. 目的:

掌握计数器的使用方法, 学会计数器工作方式的设置。

二. 内容:

利用计数器对单脉冲电路产生的脉冲进行计数, 并在 LED 显示器上显示。

三. 硬件连接框图 (图 23):

相关电路原理图见第一章图 10 及图 12。

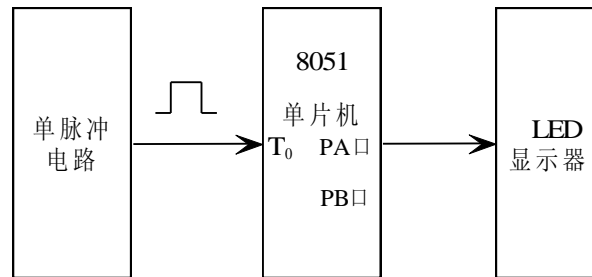


图 23

四. 实验步骤:

1、 硬件连接:

将单脉冲电路的正脉冲输出端连接 8031 计数器 0 的输入端 T0, 8255 (2) 片选信号 /CS 连接 /Y3 ~ /Y7 (任选其一);

将相关功能块的电源接通。

2、 软件设计:

1) 定时器/计数器 T0 设置成计数方式, 工作方式为方式 1, 计数初值 0000H, 软件启动。

2) 对单脉冲电路的正脉冲进行计数, 并将计数值在 LED 显示器上显示。

3) 8255 (2) PA 口 PA7 控制小数点, PA6~PA0 控制字形 (经反向驱动)。

8255 (2) PB 口 PB5~PB0 控制字位 (经反向驱动)。LED 显示器为共阴极。

五. 参考程序清单 (见下页附录三):

六. 高级要求: 要求将计数器中的数值以十进制数的形式显示。修改程序,

观察结果。

附录三（参考程序清单）：

8255(2) /CS 接 /Y3(6000H)。

```
ORG 0000H
AJMP START
ORG 0100H
START: MOV TMOD, #05H;    T0 初始化。计数方式 1，软件启动
      MOV TH0, #00H;      初值
      MOV TL0, #00H;
      CLR EA;             关中断
      CLR ET0;
      SETB TR0;          启动计数器

      MOV DPTR, #6003H;   8255 初始化（扩展 LED 显示器）
      MOV A, #80H
      MOVX @DPTR, A

      MOV R7, #06H;       显示缓冲区清 0
      MOV R1, #70H;
CLE:   MOV @R1, #00H;
      INC R1
      DJNZ R7, CLE

READT0: ACALL LED;
      MOV A, TH0;         读计数器
      MOV 41H, A;
      MOV A, TL0;
      MOV 40H, A;
      MOV A, 41H;
      CJNE A, TH0, READT0; 处理进位

      MOV 73H, A;         高位拆字
      ANL A, #0FH
      MOV 72H, A
      MOV A, 73H
      ANL A, #0F0H
      SWAP A
      MOV 73H, A

      MOV A, 40H;         低位拆字
      MOV 71H, A
      ANL A, #0FH
```

```

MOV 70H, A
MOV A, 71H
ANL A, #0F0H
SWAP A
MOV 71H, A
AJMP READT0

LED:  MOV R7, #06H;      显示子程序
      MOV R0, #75H;     显示缓冲区 75H-70H
      MOV R2, #01h;     位码
      MOV A, R2
LOOP: MOV DPTR, #6001H;  B 口位选
      MOVX @DPTR, A

      MOV DPTR, #LTABLE; 查字形码
      MOV A, @R0
      MOVC A, @A+DPTR
      MOV DPTR, #6000H;  A 口字选
      MOVX @DPTR, A
      ACALL DELAY2MS

      DEC R0;
      MOV A, R2
      RL A
      MOV R2, A
      DJNZ R7, LOOP;
      RET

DELAY2MS: MOV R3, #04H
DELAY0:  MOV R4, #0FFH
DELAY1:  DJNZ R4, DELAY1
         DJNZ R3, DELAY0
         RET

LTABLE: DB 40H, 79H, 24H, 30H
        DB 19H, 12H, 02H, 78H
        DB 00H, 10H, 08H, 03H
        DB 46H, 21H, 06H, 0EH

```

实验八 定时器（时钟）实验

一、目的：

掌握定时器的使用方法，学会定时器工作方式的设置。

二、内容：

在 LED 显示器上显示时、分、秒。

利用定时器产生 50MS 定时中断。

到达相应时间后，对时、分、秒进行修改。

三、硬件连接框图（图 24）：

LED 显示器电路原理图见第一章图 12。

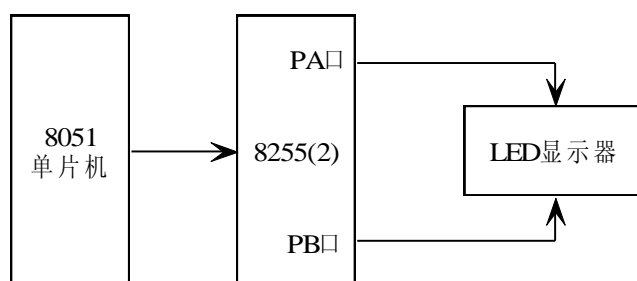


图 24

四、实验步骤：

1、硬件连接：

将 8255（2）片选信号 /CS 连接 /Y3 ~ /Y7（任选其一）；

将相关功能块的电源接通。

2、软件设计：

1) 定时器/计数器 T1 设置成定时方式，工作方式为方式 1，软件启动，允许中断（晶振频率为 12M）。

2) 利用定时器 T1 产生 50MS 定时中断，20 次中断后秒加 1，60 秒后分加 1，60 分后时加 1，24 时后恢复 0 时。

3) 将时、分、秒送入 LED 显示器显示。

4) 8255（2）PA 口 PA7 控制小数点，PA6~PA0 控制字形（经反向驱动），PB 口 PB5~PB0 控制字位（经反向驱动）。LED 显示器为共阴极。

五、参考程序清单（见下页附录四）：

六、高级要求：

1、利用键盘设置时钟初值，并利用键盘控制时钟的暂停和启动。修改程序，观察结果。

2、查询资料，了解专用时钟芯片及其与 8031 单片机的连接方式。

附录四（参考程序清单）：

8255(2) /CS 接 /Y3(6000H)。

```

                ORG 0000H
                AJMP MAIN          ;
                ORG 001BH          ; T1 的中断入口地址
                AJMP SERVE        ; 转向中断服务程序

                ORG 0100H
MAIN:           MOV  TMOD, #10H    ; 设 T1 工作于定时模式 1
                MOV  TH1, #3CH
                MOV  TL1, #0B0H    ; 赋计数初值（12MHZ 时 50MS）

                MOV  DPTR, #6003H  ; 8255(2) 初始化
                MOV  A, #80H
                MOVX @DPTR, A

                MOV  20H, #14H     ; 装入中断次数（20 次）
                CLR  A
                MOV  30H, A        ; 时单元清 0（时分秒初值）
                MOV  31H, A        ; 分单元清 0
                MOV  32H, A        ; 秒单元清 0

                SETB ET1           ; 允许 T1 中断
                SETB EA           ; 允许 CPU 中断
                SETB TR1          ; 启动定时器 T1
LOOP:           ACALL LED         ; 调用显示子程序（等待中断）
                AJMP  LOOP

LED:            MOV  A, 30H        ; 显示子程序
                ANL  30H, #0F0H    ; 将时单元拆为 40 与 41 单元
                MOV  40H, 30H
                MOV  30H, A
                ANL  30H, #0FH
                MOV  41H, 30H
                MOV  A, 31H        ; 将分单元拆为 42 与 43 单元
                ANL  31H, #0F0H
                MOV  42H, 31H
                MOV  31H, A
                ANL  31H, #0FH
                MOV  43H, 31H
                MOV  A, 32H        ; 将秒单元拆为 44 与 45 单元
```

```

ANL  32H, #0F0H
MOV  44H, 32H
MOV  32H, A
ANL  32H, #0FH
MOV  45H, 32H

MOV  R0, #40H      ; R0 指向显示缓冲区首位
MOV  R1, #01H     ; 送位码到 R1
MOV  R5, #06H     ; R5 作计数器, 显示 6 位

LOOP1:  MOV  DPTR, #6001H ; 8255 的 PB 口送位选码
        MOV  A, R1
        MOVX @DPTR, A
        MOV  DPTR, #SEG
        MOV  A, @R0
        MOVC A, @A+DPTR

        MOV  DPTR, #6000H ; 8255 的 PA 口送段选码
        MOVX @DPTR, A
        ACALL DELAY1MS ; 调用延时子程序

        INC  R0      ; 指向下一个缓冲区
        MOV  A, R1
        RL  A      ; 指向下一个显示位
        MOV  R1, A
        DJNZ R5, LOOP1
        RET

DELAY1MS: MOV  R7, #02H ; 延时 1 毫秒
          DELAY0: MOV  R6, #0FFH
          DELAY1: DJNZ R6, DELAY1
                  DJNZ R7, DELAY0
                  RET

SEG:      DB  40H, 79H, 24H, 30H, 19H
          DB  12H, 02H, 78H, 00H, 10H ; 0-9 字形表

SERVE:   PUSH  PSW ; 中断服务子程序
          PUSH  ACC ; 保护现场

          MOV  TH1, #3CH
          MOV  TL1, #0B0H ; 重新赋计数初值
          DJNZ 20H, RETUNT ; 1 秒未到, 返回

```

```

MOV  20H, #14H      ; 重置中断次数
MOV  A, #01H
ADD  A, 32H        ; “秒位” 加 1
DA   A
MOV  32H, A        ; 转换为 BCD 码

CJNE A, #60H, RETUNT ; 未计满 60 秒, 返回
MOV  32H, #00H     ; 计满 60 秒, “秒位” 清 0
MOV  A, #01H
ADD  A, 31H        ; “分位” 加 1
DA   A
MOV  31H, A        ; 转换为 BCD 码

CJNE A, #60H, RETUNT ; 未计满 60 分, 返回
MOV  31H, #00H     ; 计满 60 分, “分位” 清 0
MOV  A, #01H
ADD  A, 30H        ; “时位” 加 1
DA   A
MOV  30H, A        ; 转换为 BCD 码

CJNE A, #24H, RETUNT ; 未计满 24 小时, 返回
MOV  30H, #00H     ; 计满 24 小时, “时位” 清 0

RETUNT: POP  ACC
        POP  PSW      ; 恢复现场
        RETI         ; 返回

```

第六章 串行通信实验

实验九 串行通信实验

一. 目的:

利用串行口实现单片机间的通信。

二. 内容:

将 A 单片机的内存单元 40H~4FH 的内容通过串行口传输到 B 单片机中, 存放在 B 单片机的内存单元 50H~5FH。

三. 硬件连接框图 (图 25):

串行口电路原理图见第一章图 13。

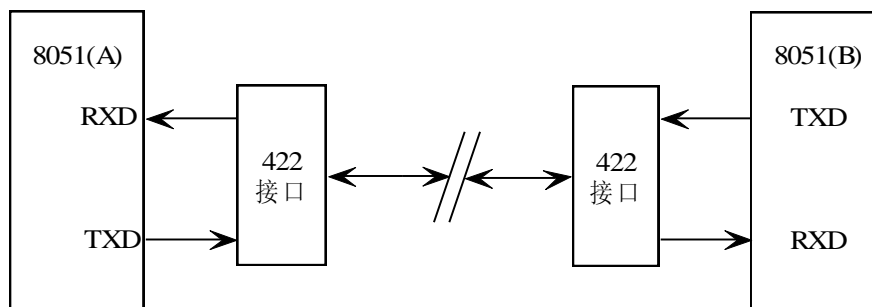


图 25

四. 实验步骤:

1、 硬件连接:

- 1) A 单片机实验仪 RS-422 接口输入端 IN(+,-)连接 B 单片机实验仪 RS-422 接口输出端 OUT (+, -);
- 2) B 单片机实验仪 RS-422 接口输入端 IN(+,-)连接 A 单片机实验仪 RS-422 接口输出端 OUT (+, -)。

3) 连接相关功能块电源。

2、 软件设计：

1) A、B 单片机串行口工作方式方式为方式 1，定时器 T1 做波特率发生器，波特率为 1200HZ（晶振频率为 12M）。

2) A 机在程序中给 40H~4FH 单元赋值。

3) B 机接收的数据存放在 50H~5FH 单元中。

3、 执行程序后，暂停。从窗口中检查 B 机 50H~5FH 单元中的内容。

五. 参考程序框图（见下页图 26）：

六. 参考程序清单（见下页附录五）：

七. 高级要求：

要求先将 A 单片机的内存单元 40H~4FH 的内容通过串行口传输到 B 单片机中，存放在 B 单片机的内存单元 50H~5FH。

然后再将 B 单片机的内存单元 40H~4FH 的内容通过串行口传输到 A 单片机中，存放在 A 单片机的内存单元 50H~5FH。

A、B 单片机之间利用联络字进行协调。

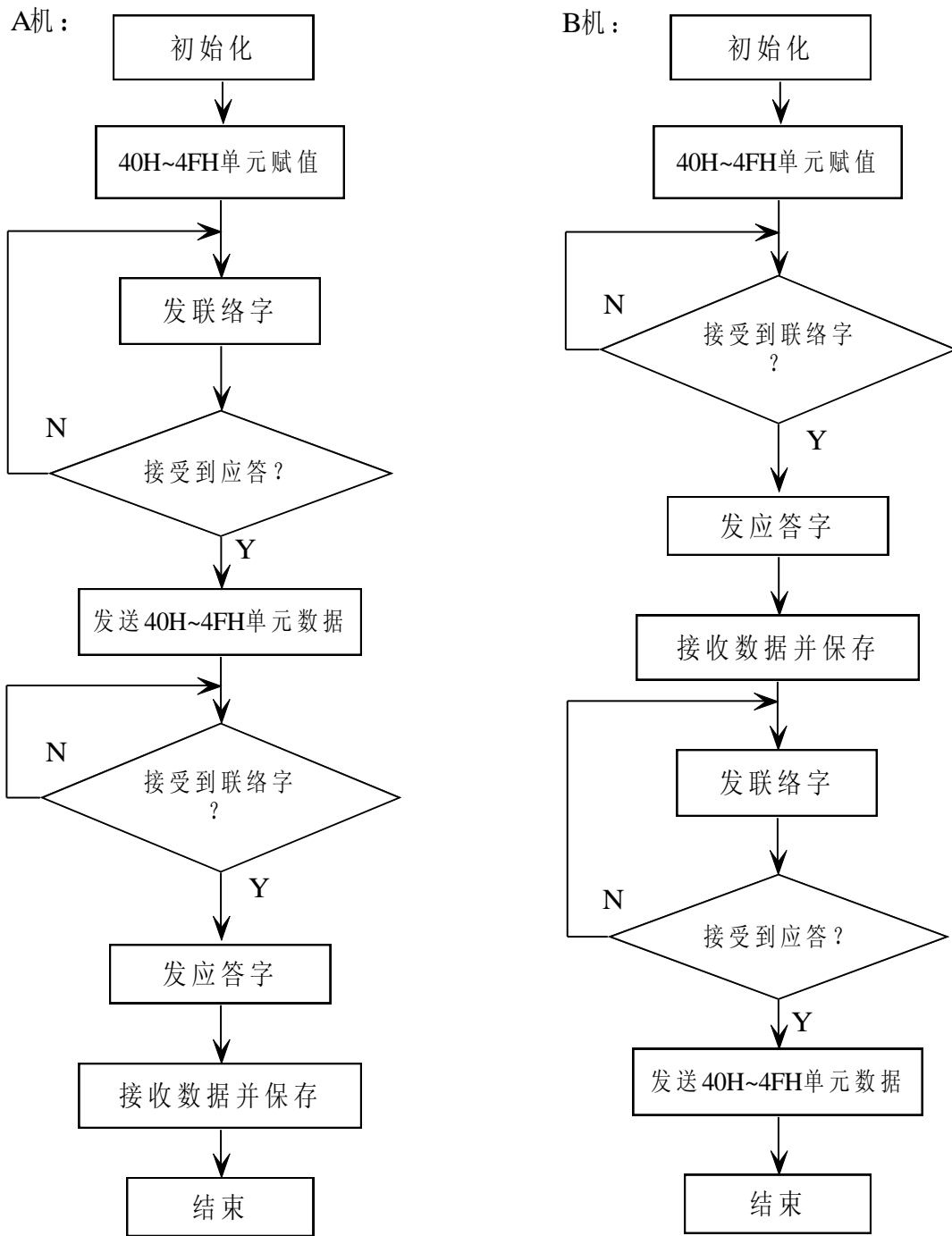


图 26 串行口实验程序流程图

附录五（参考程序清单）：

发送：

```

                ORG    0000H
                AJMP   MAIN
                ORG    0100H
MAIN:           MOV    R0, #40H           ;40H-4FH 赋值
                MOV    R7, #10H
                MOV    A, #00H
LOOP:           MOV    @R0, A
                INC    A
                INC    R0
                DJNZ   R7, LOOP
FMT_T_S:        MOV    TMOD, #20H        ;波特率设置
                MOV    TH1, #0F3H
                MOV    TL1, #0F3H
                SETB   TR1
                MOV    SCON, #50H        ;串行口初始化
                MOV    PCON, #80H        ;置 SMOD=1
FMT_RAM:        MOV    R1, #40H          ;数据首址送 R1
                MOV    R6, #00H          ;清累加和寄存器
                MOV    R5, #10H          ;送字节数至 R5
TX_ACK:         MOV    A, #06H           ;发呼叫信号“06H”
                MOV    SBUF, A
WAIT1:          JBC    TI, RX_YES        ;等待发送完
                SJMP   WAIT1
RX_YES:         JBC    RI, NEXT1         ;接收乙机回答
                SJMP   RX_YES
NEXT1:          MOV    A, SBUF            ;判断乙机是否同意接收
                CJNE   A, #00H, TX_ACK
TX_BYTES:       MOV    A, R5             ;向乙机发送要传送的字节个数
                MOV    SBUF, A
WAIT2:          JBC    TI, TX_NEWS
                SJMP   WAIT2
TX_NEWS:        MOV    A, @R1            ;发送
                MOV    SBUF, A
                ADD    A, R6
                MOV    R6, A
                INC    R1                 ;指向下一个数据
WAIT3:          JBC    TI, NEXT2
                SJMP   WAIT3
    
```

```

NEXT2:    DJNZ  R5, TX_NEWS    ;判发送结束
TX_SUM:   MOV   A, R6          ;数据发送完, 发累加和给乙机
          MOV   SBUF, A
WAIT4:    JBC   TI, RX_OFH
          SJMP  WAIT4
RX_OFH:   JBC   RI, IF_OFH    ;等待乙机回答
          SJMP  RX_OFH
IF_OFH:   MOV   A, SBUF
          CJNE  A, #0FH, FMT_T_S ;判断传送正确否
          SJMP  $

```

接收:

```

          ORG   0000H
          AJMP  MAIN
          ORG   0100H
MAIN:     MOV   TMOD, #20H     ;T1 初始化
          MOV   TH1, #0F3H
          MOV   TL1, #0F3H
          SETB  TR1
          MOV   SCON, #50H    ;串行口初始化
          MOV   PCON, #80H    ;SMOD=1
FMT_RAM:  MOV   R1, #50H      ;数据首址送 R1
          MOV   R6, #00H      ;清累加和寄存器
RX_ACK:   JBC   RI, IF_06H    ;接收呼叫信号
          SJMP  RX_ACK
IF_06H:   MOV   A, SBUF       ;判断呼叫信号是否有误
          CJNE  A, #06H, TX_15H
TX_00H:   MOV   A, #00H       ;向甲机回送同意接收信号
          MOV   SBUF, A
WAIT1:    JBC   TI, RX_BYTES  ;等待应答信号发送完
          SJMP  WAIT1
TX_15H:   MOV   A, #15H       ;向甲机报告呼叫信号不正确
          MOV   SBUF, A
WAIT2:    JBC   TI, HAVE1
          SJMP  WAIT2
HAVE1:    LJMP  RX_ACK        ;返回到接受呼叫信号状态
RX_BYTES: JBC   RI, HAVE2     ;接收数据块长度
          SJMP  RX_BYTES
HAVE2:    MOV   A, SBUF       ;给长度寄存器赋值
          MOV   R5, A
RX_NEWS:  JBC   RI, HAVE3     ;接收数据
          SJMP  RX_NEWS
HAVE3:    MOV   A, SBUF       ;将接受到的数据存入 RAM

```



```

MOV    @R1, A
INC    R1
ADD    A, R6           ;形成累加和
MOV    R6, A
DJNZ   R5, RX_NEWS    ;判断数据是否接收完毕
RX_SUM: JBC    RI, HAVE4      ;接收校验和
        SJMP   RX_SUM
HAVE4:  MOV    A, SBUF      ;判断传送是否正确
        SUBB   A, R6
        CJNE  A, #00H, TX_E
TX_RIGHT: MOV   A, #0FH      ;向甲机报告传送正确
        MOV   SBUF, A
WAIT3:  JBC    TI, GOOD
        SJMP  WAIT3
TX_E:   MOV    A, #0F0H     ;向甲机报告传送有误
        MOV   SBUF, A
WAIT4:  JBC    TI, AGAIN
        SJMP  WAIT4
AGAIN:  LJMP   FMT_RAM      ;返回重新接收数据状态
GOOD:   SJMP   $

```

第七章 D/A 及 A/D 实验

实验十 D/A 转换实验 (DAC0832)

一. 目的

掌握 DAC0832 与 8031 的接口方法。掌握 D/A 转换程序设计方法。

二. 内容

将数字量 (00H~FFH) 从 8031 传输到 8255(1), 再从 8255(1)传输到 DAC0832。测量 DAC0832 输出电压。

三. 硬件连接框图 (图 27):

电路原理图见第一章图 7。

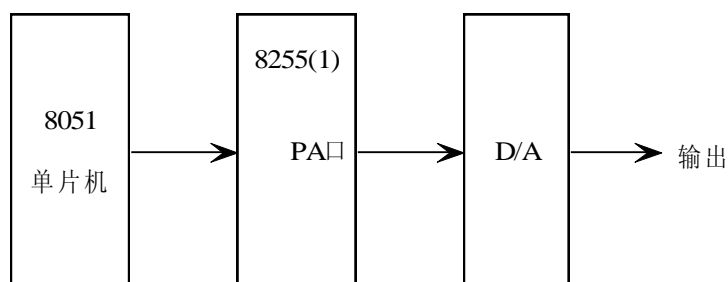


图 27

四. 实验步骤

1、 硬件连接:

1) DAC0832 电路的标准电压端 VREF 接+5V, 片选信号端 /CS 接 /Y3 ~ /Y7 (任选其一), 写控制端 /WR 接 8031 的 /WR, D7 ~ D0 接 8255(1)PA 口的 PA7~PA0, 反馈端 FG 接反馈电阻输入端 RFB;

2) 8255(1)片选信号 /CS 接 /Y3 ~ /Y7 (任选其一);

3) 连接相关功能块电源。

2、 软件设计:

1) 8255(1)初始化 (PA 口输出)。

2) 根据下列表格, 将相关数字量从 8031 传输到 8255(1)PA 口, 再从 8255(1)PA 口传输到 DAC0832。

3) 传输数据:

```
MOV A, #data
MOV DPTR, #addr8255
MOVX @DPTR, A
MOV DPTR, #addr0832
MOVX @DPTR, A
```

五. 执行程序。用万用表测量 DAC0832 电路的输出端 Vout3 的输出电压。

在程序中修改输出的相关数字量，重复测量。

六. 填写以下表格：（自行增加其它数字量）

数 字 量	输 出 电 压 (V)
00H	
3FH	
7FH	
80H	
BFH	
FFH	

七、参考程序框图（见图 28）：

八、高级要求：

程序中输出的数字量按一定的规律变化，在 DAC0832 电路的输出端 Vout3 输出某一波形。用示波器观察。

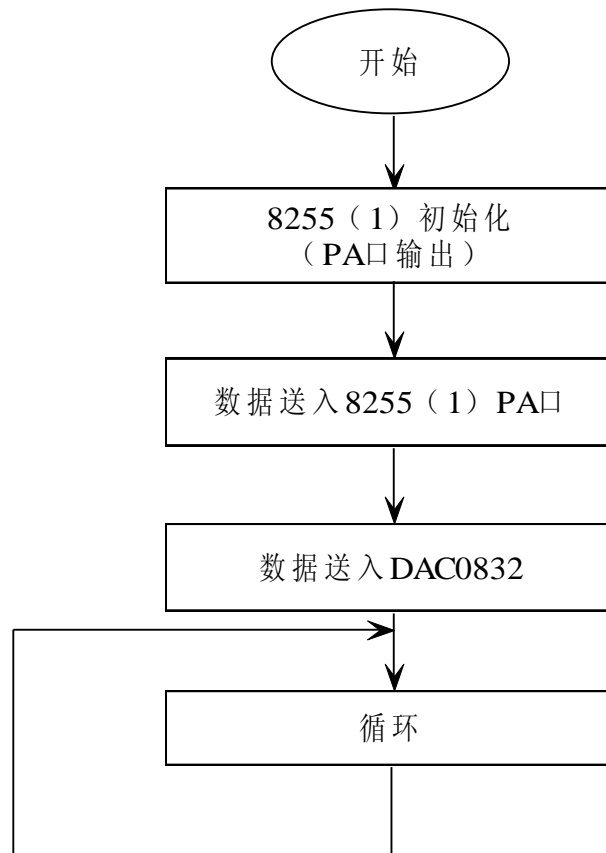


图 28 程序流程图

实验十一 A/D 转换实验 (ICL7135)

一. 目的:

1. 掌握四位半 A/D 芯片 ICL7135 和 MCS-51 单片机接口设计方法。
2. 了解 8031 读取 ICL7135 转换结果的程序设计方法。

二. 内容:

将直流电压从 ICL7135 的输入端输入, 转换成对应的数字量后, 送 LED 显示器显示。显示范围: 00000-19999。

三. 硬件连接框图 (图 29):

相关电路原理图见第一章图 6、图 11 和图 12。

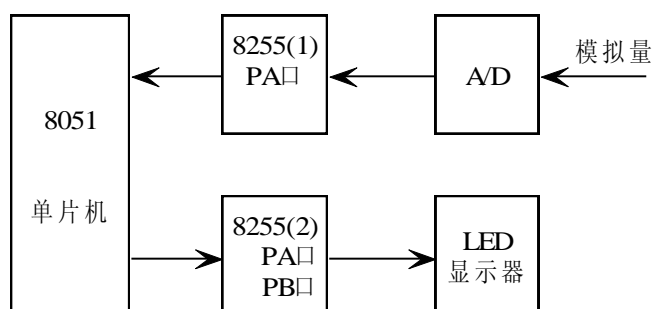


图 29

四. 实验步骤:

1、硬件连接:

- 1) 将+2.5V 电压接入电位器 WR2 的一端, 0V (GND) 接入另一端。中间抽头输出 0V~2.5V, 此电压接入 ICL7135 输入端 Vin5;
- 2) 8255 (1) PA 口、PB 口接 ICL7135 数据输出端 B8~B1 (BCD 码输出) 及位状态输出端 D5~D1 (万、千、百、十、个位输出状态);
- 3) ICL7135 的时钟输入端 CLKIN 接 1/8 ALE, 转换状态输出端 BUSY 接 8031 的外部中断输入端;
- 4) 8255 (1) 片选信号 /CS 接 /Y3~/Y7 (任选其一);
8255 (2) 片选信号 /CS 接 /Y3~/Y7 (任选其一);
- 5) 连接相关功能块电源。

2、软件设计:

- 1) 8255 (1) 初始化 (PA 口、PB 口输入, 连接 ICL7135)。8255 (2) 初始化 (PA 口、PB 口输出, 扩展 LED 显示器)。设置外部中断 (用 BUSY 申请中断), 低电平有效。

- 2) 主程序调用显示程序，并循环等待中断。
- 3) 中断服务子程序判别万、千、百、十、个位输出状态，分别读入万、千、百、十、个位数据并保存。修改显示缓冲器内容。
- 4) 8255 (2) PA 口 PA7 控制小数点，PA6~PA0 控制字形（经反向驱动），PB 口 PB5~PB0 控制字位（经反向驱动）。LED 显示器为共阴极。

五. 参考程序清单（见下页附录六）：

六. 高级要求：

- 1) 处理 ICL7135 的溢出标志位 OVER 及欠量程标志位 UNDER。修改程序，过量程、欠量程时在 LED 显示器上提示。
- 2) 查询资料，了解 ICL7135 与 8031 单片机二线式连接法（利用 A/D 转换的时间长短计算出 A/D 数值）。

附录六(参考程序清单)：

8255 (1) PA 口的 PA7~PA4 接 ICL7135 数据输出端 B8~B1 (BCD 码输出)，PA3~PA0 接 ICL7135 位状态输出端 D4~D1 (千、百、十、个位输出状态)；
 无千、百、十、个位输出时，即为万位输出；
 ICL7135 的时钟输入端 CLKIN 接 1/8 ALE，转换状态输出端 BUSY 接 8031 的外部中断输入端 /INT0；
 8255 (1) 片选信号 /CS 接 /Y7 (E000H)；
 8255 (2) 片选信号 /CS 接 /Y3 (6000H)。

```

ORG 0000H
AJMP START
ORG 0003H           ;INT0 中断矢量地址入口
LJMP INTER

ORG 0100H
START: MOV R0, #50H
      MOV R2, #06H
LOOP0: MOV @R0, #10H      ;给显示缓冲区 50H-55H 赋值
      INC R0              ;显示全黑
      DJNZ R2, LOOP0

      MOV DPTR, #6003H    ;对 8255(2) 进行初始化(LED 显示器)
      MOV A, #80H
      MOVX @DPTR, A
      MOV A, #92H        ;对 8255(1) 进行初始化(接 ICL7135)
      MOV DPTR, #0E003H
      MOVX @DPTR, A

```

```

MOV SP, #70H           ;设置堆栈区
SETB ITO               ;/INT0 跳变触发
SETB EA
SETB EX0               ;开中断

MAIN: ACALL LED        ;调显示子程序
NOP                    ;等待中断
NOP
AJMP MAIN

LED: CLR EA            ;关中断
MOV R0, #01H          ;位码
MOV R1, #50H          ;显示缓冲区首址

LOOP: MOV A, R0        ;送位码
MOV DPTR, #6001H
MOVX @DPTR, A

MOV A, @R1             ;送段码
MOV DPTR, #SEG
MOVC A, @A+DPTR
MOV DPTR, #6000H
MOVX @DPTR, A
ACALL DELAY

MOV A, #7FH           ;清黑
MOVX @DPTR, A

GOON: INC R1           ;指向下一位
MOV A, R0
RL A
MOV R0, A
CJNE R0, #40H, LOOP   ;六位显示完返回
SETB EA               ;开中断
RET

DELAY: MOV R7, #02H
DELAY0: MOV R6, #0FFH
DELAY1: DJNZ R6, DELAY1
        DJNZ R7, DELAY0
RET

```

```

        ORG 1000H
INTER:  PUSH PSW                ;中断子程序
        PUSH DPH                ;现场保护
        PUSH DPL
        PUSH ACC
        CLR EA                    ;关中断

D5:    MOV DPTR, #0E000H
        MOVX A, @DPTR            ; 等待万位输出
        MOV R3, A
        ANL A, #0FH
        JNZ D5
        MOV A, R3
        ANL A, #0F0H            ;万位 BCD 码放入 50H
        SWAP A
        MOV 50H, A

D4:    MOV DPTR, #0E000H
        MOVX A, @DPTR
        MOV R3, A
        ANL A, #08H            ;等待千位输出
        JZ D4
        MOV A, R3
        ANL A, #0F0H
        SWAP A
        MOV 51H, A            ;千位 BCD 码放入 51H

D3:    MOV DPTR, #0E000H        ;等待百位输出
        MOVX A, @DPTR
        MOV R3, A
        ANL A, #04H
        JZ D3
        MOV A, R3
        ANL A, #0F0H
        SWAP A
        MOV 52H, A            ;百位 BCD 码放入 52H

D2:    MOV DPTR, #0E000H        ;等待十位输出
        MOVX A, @DPTR
        MOV R3, A
        ANL A, #02H
        JZ D2
        MOV A, R3
        ANL A, #0F0H

```

```

        SWAP A
        MOV 53H, A                ;十位 BCD 码放入 53H

D1:    MOV DPTR, #0E000H        ;等待个位输出
        MOVX A, @DPTR
        MOV R3, A
        ANL A, #01H
        JZ D1
        MOV A, R3
        ANL A, #0F0H
        SWAP A
        MOV 54H, A                ;个位 BCD 码放入 54H

RETURN: POP ACC                  ;恢复现场
        POP DPL
        POP DPH
        POP PSW
        SETB EA                    ;开中断
        RETI

SEG:   DB 40H, 79H, 24H, 30H, 19H, 12H, 02H, 78H
        DB 00H, 10H, 08H, 03H, 46H, 21H, 06H, 0EH
        DB 7FH                    ; 7FH 为全黑字形码

```