

## I. 单片机原理基本实验指导

# 实验一 单片机演示实验

### 一 实验目的：

1. 了解单片机的用途, 增加感性认识, 激发学习微机原理这门课的兴趣
2. 对单片机不同型号和种类有一定的了解, 抓住内在规律学好微机原理这门课

### 二 实验内容：

1. 观看单片机控制演示实验
  - a) 单片机控制红绿灯
  - b) 单片机控制软驱步进电机
  - c) 单片机演奏音乐
  - d) 单片机调速步进马达
2. 通过 CAI 软件了解单片机不同型号和种类

### 三 实验报告：

1. 到图书馆或网站查找不同种类的单片机（至少写出与 CAI 课件不同的五种型号）
2. 你认为单片机用途大吗？对该课程有兴趣吗？打算怎样学习这门课？

## 实验二 分支与循环程序

### 一 实验目的：

1. 熟悉单片机仿真软件的基本操作
2. 掌握分支与循环程序的调试方法

### 二 实验设备：

微机一台，CYGNAL IDE 软件

### 三 实验内容：

1. 打开计算机进入 Windows 系统，运行 CYGNAL IDE 程序，进入单片机仿真开发界面。选择 File 菜单中的 New 命令，输入程序后 Save，文件扩展名必须为 ASM。
2. 使用 Project 菜单中的 Build/Make Project (F7)，进行汇编/编译，直至无语法错误后，使用 Debug 菜单中的 Connect (Alt+C) 连接调试系统。使用 Debug 菜单中的 Download Object File (Alt+D) 下载目标文件，即将程序装到单片机芯片上。
3. 用 View 菜单中的 Debug Windows (调试窗口) 监视 MAX 单元变化，该窗口可观察各寄存器、端口、RAM 等状态。
4. 打开 RAM 观察窗口，从 30H 单元开始输入数据，第一个为比较数据的个数，其后为数据，然后 Debug 菜单 Refresh 进行写入操作。
5. Step (F11) 单步运行、Run to Cursor (F8) 运行到光标所在行、Breakpoints (F9) 设置断点、Step Over (F10) 步跃运行、Go (F5) 全速运行。
6. 用 Step 单步运行方式运行程序，要求观察到每个分支。同时监视 MAX 单元变化是否正确。
7. 在 NEXT 语句上设置断点（用 Debug 菜单中的 Breakpoints），用断点方式重新运行程序，观察 MAX 单元变化。
8. 清除断点，用全速方式运行程序，检验结果是否正确。以上菜单命令均可在工具条找到快捷按钮直接使用。
9. 换一组数据，全速运行，检验结果。
10. 修改程序，使之能求最小值。全速运行并检验结果。
11. 查看工具栏各按钮功能，并在实验报告中注明左起第八个按钮以后的按钮功能。
12. 实验完，使用 Tools 菜单中的 Erase code space 擦除闪存中的程序代码。

### 实验预习参考程序：

### 预习要求：

1. 黑体字**不得更改**，输入时不得有误
2. 补充空白语句
3. 除黑体字外程序加注释

```

$INCLUDE          (C8051F020. INC)

      ORG      0000H          ;
MAX   EQU      50H          ;
      LJMP    START        ;
START: MOV     WDTCN, #0DEH
      MOV     WDTCN, #0ADH
      MOV     XBR0, #04H
      MOV     XBR1, #1EH
      MOV     XBR2, #40H
      MOV     R0, #30H      ;
      MOV     A, @R0       ;
      DEC     A            ;
      MOV     R2, A        ;
      INC     R0           ;
      MOV     MAX, _____ ;
LOOP:  _____        ;
      MOV     A, _____ ;
      CJNE   A, MAX, TEST  ;
      SJMP   NEXT         ;
TEST:  _____        ;
      MOV     MAX, @R0     ;
NEXT:  DJNZ  R2, LOOP     ;
HERE:  JMP   HERE        ;
      END                ;

```

#### 四 实验报告：

1. 写出实验过程，所选的数据，观察的结果，并画出流程图
2. 为什么要有设断点运行程序这个命令
3. 试写实验中出现的**问题及原因**

## 实验三 中断与 CTC 功能

### 一 实验目的：

1. 了解单片机的中断功能。
2. 了解 CTC 的计数与定时工作方式。

### 二 实验设备：

1. 单片机开发系统一套。
2. 单片机实验主板一块。
3. 直流稳压电源一台。

### 三 实验电路：

见实验主板电原理图。

### 四 实验内容：

- ★ 中断实验，将仿真连线器插入实验板主板，参看主实验板电路图，EX1 按键通过 DIP 开关接单片机 INTO 作为外中断源，编写一段程序，使 EX1 按下时，LED1 点亮。程序中需要延时可用定时器 0，调试你的程序。

### 实验预习参考程序：

#### 预习要求：

1. 黑体字不得更改，输入时不要有误
2. 阅读实验内容及 CPU 主控主板电路图，根据实验内容找出与其相关的电路图对应部分
3. 补充空白语句
4. 除黑体字外程序加注释

```
$INCLUDE      (C8051F020. INC)
      ORG      0000H      ;
      LJMP    START      ;
      ORG      0003H      ;
      LJMP    INTO       ;
      ORG      0030H      ;
START: MOV     WDTCN, #0DEH
      MOV     WDTCN, #0ADH
      MOV     XBRO, #04H
      MOV     XBR1, #1EH
      MOV     XBR2, #40H
      MOV     OSCXCN, #066H
      MOV     R2, #00H
LD:    MOV     R3, #00H
      DJNZ    R3, $
```

```

    DJNZ    R2, LD
    MOV     A, OSCXCN
    JNB     ACC. 7, LD
    MOV     OSCICN, #1CH
    MOV     SP, #60H      ;
    SETB    P1. 0        ;
    SETB    _____ ;
    SETB    _____ ;
TEST:  NOP                ;
    NOP                ;
    LJMP    TEST         ;
INT0:  CLR     EX0        ;
    CLR     EA           ;
    CLR     P1. 0        ;
    LCALL   DELAY        ;
    SETB    P1. 0        ;
    SETB    EA           ;
    SETB    EX0         ;
    RETI                ;
DELAY: MOV     TMOD, _____ ;
    MOV     R7, #200     ;
LOOP1: MOV     TH0, #0E1H ;
    MOV     TL0, #0E0H  ;
    SETB    TR0         ;
    JNB     TF0, $       ;
    CLR     TR0         ;
    _____ ;
    DJNZ    R7, LOOP1    ;
    RET                ;
END

```

- ★ 外部时钟定时晶体振荡器为 6MHz，编写一段程序，将 CTC0 或 CTC1 设为定时方式，定时长 100ms，另设一软件计数器（常用来扩大定时范围）初值为 5，每当定时器溢出时重装时间常数，并将软件计数器减 1，减到 0 时使 LED1 翻转一次，并重新将计数器设为 5，如此反复循环。运行程序，观察 LED 的闪烁频率。然后改变定时器时间常数和软件计数器初值，重新运行程序，观察 LED 闪烁频率的变化。

### 实验预习参考程序：

#### 预习要求：

1. 黑体字**不得更改**，输入时不要有误
2. 补充空白语句
3. 除黑体字外程序加注释

```

$INCLUDE      (C8051F020. INC)
      ORG      0000H      ;
      LJMP     START      ;
      ORG      000BH      ;
      LJMP     CTC0       ;
      ORG      0030H      ;
START:  MOV     WDTCN, #0DEH
      MOV     WDTCN, #0ADH
      MOV     XBR0, #04H
      MOV     XBR1, #1EH
      MOV     XBR2, #40H
      MOV     OSCXCN, #066H
      MOV     R2, #00H
LD:     MOV     R3, #00H
      DJNZ    R3, $
      DJNZ    R2, LD
      MOV     A, OSCXCN
      JNB     ACC. 7, LD
      MOV     OSCICN, #1CH
      MOV     SP, #60H      ;
      MOV     IE, #82H      ;
      SETB    P1. 0        ;
      MOV     TMOD, #01H    ;
      MOV     R7, #05H      ;
      MOV     TH0, _____ ;
      MOV     TL0, _____ ;
      SETB    TR0          ;
LOOP:   NOP                ;
      NOP                ;
      LJMP    LOOP         ;
CTC0:   CLR     TF0        ;
      CLR     TR0          ;
      DJNZ    R7, RETURN   ;
      CPL     P1. 0        ;
      MOV     R7, #05H      ;
RETURN: MOV     TMOD, #01H  ;
      MOV     TH0, #3CH    ;
      MOV     TL0, #0B0H   ;
      _____          ;
      RETI                ;
      END                 ;

```

计数功能：将按键 EXC 通过 DIP 开关接入 T0，CTC0 设为计数方式。编一段程序使每当按下 5 次按键时，LED1 亮，再按 5 次后 LED2，再按 5 次按键

时 LED3 亮，再按 5 次按键后三个灯均灭，如此反复循环。运行程序时最好使用 **Multiple Step** 运行方式，调出定时器观察窗口，且当光标在 LD: 所在行时再按计数键并观察计数器的值。

### 实验预习参考程序：

#### 预习要求：

1. 黑体字**不得更改**，输入时不要有误
2. 补充空白语句
3. 除黑体字外程序加注释

```
$INCLUDE      (C8051F020. INC)
                ORG      0000H      ;
                LJMP     START      ;
                ORG      000BH      ;
                LJMP     TIMER      ;
                ORG      0030H      ;
START: MOV      WDTCN, #0DEH
MOV      WDTCN, #0ADH
MOV      XBRO, #00000100B
MOV      XBR1, #00011110B
MOV      XBR2, #01000000B
MOV      OSCXCN, #01100110B
MOV      R2, #00H
LD: MOV      R3, #00H
DJNZ     R3, $
DJNZ     R2, LD
MOV      A, OSCXCN
JNB     ACC. 7, LD
MOV      OSICN, #00011100B
MOV      SP, #60H      ;
MOV      IE, #82H      ;
MOV      TMOD, _____ ;
MOV      R3, #04H      ;
MOV      R2, #1FH      ;
MOV      TH0, #0FFH    ;
MOV      TL0, _____ ;
SETB     TR0          ;
LOOP: NOP      ;
NOP      ;
LJMP     LOOP        ;
TIMER: CLR     TF0      ;
CLR     ET0          ;
CLR     EA           ;
```

```

MOV     A, R2           ;
RL      A               ;
MOV     R2, A           ;
JB      ACC. 3, NEXT1  ;
MOV     A, #0FFH        ;
NEXT1:  MOV     P1, A    ;
        DJNZ   R3, TURN1 ;
        MOV     R3, #04H ;
        MOV     R2, #01FH ;
TURN1:  MOV     TMOD, #05H ;
        MOV     TH0, #0FFH ;
        MOV     TLO, _____ ;
        _____ TRO ;
        _____ EA ;
        _____ ETO ;
        RETI ;
        END ;

```

### 参考程序二:

```

$INCLUDE      (C8051F020. INC)
ORG          0000H ;
LJMP        START ;
ORG          0003H ;
LJMP        INTO ;
ORG          000BH ;
LJMP        CTC0 ;
ORG          0013H ;
LJMP        INT1 ;
ORG          001BH ;
LJMP        CTC1 ;
ORG          0030H ;
START:  MOV     WDTCN, #0DEH ;
        MOV     WDTCN, #0ADH ;
        MOV     XBRO, #00000100B ;
        MOV     XBR1, #00011110B ;
        MOV     XBR2, #01000000B ;
        MOV     POMDOUT, #0FFH ;
        MOV     P1MDOUT, #0FFH ;
        MOV     P2MDOUT, #0FFH ;
        MOV     P3MDOUT, #0FFH ;
        MOV     CKCON, #00001000B ;
        MOV     OSCXCN, #01100110B ;
        MOV     R2, #00 ;
LD:     MOV     R3, #00 ;
        DJNZ   R3, $ ;
        DJNZ   R2, LD ;

```



```

MOV     A, OSCXCN
JNB     ACC. 7, LD
MOV     OSCICN, #00011100B
MOV     SP, 60H      ;
SETB    EA          ;
SETB    EX0         ;
SETB    ET0         ;
SETB    EX1         ;
SETB    ET1         ;
SETB    TR0         ;
SETB    TR1         ;
MOV     TMOD, #15H  ;
MOV     TH1, #00H   ;
MOV     TL1, #00H   ;
MOV     TH0, #0FFH  ;
MOV     TL0, #0FDH  ;
LOOP:   MOV     A, P1      ;
        JNB     ACC. 3, S1  ;
        JNB     ACC. 4, S2  ;
        JNB     ACC. 5, S3  ;
        AJMP    LOOP       ;
S1:     CLR     P1. 2      ;
        LJMP    LOOP       ;
S2:     SETB    P1. 2      ;
        LJMP    LOOP       ;
S3:     CPL     P1. 2      ;
        LJMP    LOOP       ;
CTC1:   CLR     TF1        ;
        CPL     P1. 0      ;
        RETI                    ;
CTC0:   CLR     TF0        ;
        MOV     TH0, #0FFH  ;
        MOV     TL0, #0FDH  ;
        CPL     P1. 1      ;
        RETI                    ;
INT0:   CPL     P1. 1      ;
        RETI                    ;
INT1:   CPL     P1. 2      ;
        RETI                    ;
        END                    ;

```

## 五 实验报告：

1. 画出流程图
2. 写出实验过程，所选的数据，观察的现象
3. 总结调试中出现的问题及原因

## 实验四 串行口应用

## 一 实验目的：

1. 掌握利用串行口扩展外部接口的方法
2. 掌握利用串行口实现双机通信的方法

## 二 实验设备：

1. 单片机开发系统一套
2. 单片机实验主板一块
3. 直流稳压电源一台。

## 三 实验电路：

见实验主板电原理图

## 四 实验内容：

- ★ 利用串行口扩展 8 位并行输出口：硬件连接如实验主板电路，编写一段程序，将以 NUMB 为首址的 10 个非压缩 BCD 码依次输出到 LED 数码管显示，每个显示 1 秒（可用软件延时实现）。然后调试编制的程序

## 实验预习参考程序：

### 预习要求：

1. 黑体字不得更改，输入时不要有误
2. 补充空白语句
3. 除黑体字外程序加注释

```
$INCLUDE      (C8051F020. INC)
NUM EQU 20H ;
ORG 0000H ;
LJMP START ;
ORG 0030H ;
START: MOV   WDTCN, #0DEH
      MOV   WDTCN, #0ADH
      MOV   XBRO, #00000100B
      MOV   XBR1, #00011110B
      MOV   XBR2, #01000000B
      MOV   POMDOUT, #0FFH
      MOV   P1MDOUT, #0FFH
      MOV   P2MDOUT, #0FFH
      MOV   P3MDOUT, #0FFH
      MOV   CKCON, #00001000B
      MOV   OSCCN, #01100110B
      MOV   R2, #00
LD:   MOV   R3, #00
```

```

        DJNZ    R3, $
        DJNZ    R2, LD
        MOV     A, OSCXCN
        JNB     ACC. 7, LD
        MOV     OSCICN, #00011100B
LOOP:   MOV     R0, #NUM          ;
        MOV     R2, #10         ;
NEXT:   LCALL  DISP            ;
        LCALL  DELAY           ;
        INC    R0              ;
        DJNZ   R2, NEXT        ;
        LJMP  LOOP             ;
DISP:   MOV     DPTR, #CODE1    ;
        MOV     A, @R0         ;
        MOVC   A, @A+DPTR     ;
        MOV     SBUF0, A       ;
        JNB    TI0, $          ;
        CLR    TI0            ;
        RET
DELAY:  MOV     R3, #_____   ;
LD1:    MOV     R4, #100       ;
LD2:    MOV     R5, #200       ;
        DJNZ   R5, $           ;
        DJNZ   R4, LD2         ;
        DJNZ   R3, LD1         ;
        RET
CODE:   DB      03H, 9FH, 25H, 0DH, 99H, 49H, 41H, 1FH, 01H, 09H, 11H, 0C1H, 63H, 85H, 61H, 71H
        ; 0 1 2 3 4 5 6 7 8 9 A b C d E F
        END

```

- ★ 双机通信：按实验主板电路编写一段程序，每当 K1 键按下一次时，发送一个字符“ODH”到串行口，而每收到一个“ODH”字符时，点亮 LED1 灯；每当 K2 键按下时，发送“OEH”字符到串行口，每收到一个“OEH”字符时，熄灭 LED1 数码管，然后两人一组，进行双方通讯实验。两人的主板中 RXD 与对方的 TXD 引线相连，二主板地线相连，通信方式可选串口模式 1，波特率取 12000bps

### 实验预习参考程序：

#### 预习要求：

1. 黑体字不得更改，输入时不要有误
2. 补充空白语句
3. 除黑体字外程序加注释

```

$INCLUDE      (C8051F020. INC)
        ORG    0000H          ;

```

```

        LJMP     START          ;
        ORG     0030H          ;
START:  MOV     WDTCN, #0DEH
        MOV     WDTCN, #0ADH
        MOV     XBRO, #00000100B
        MOV     XBR1, #00011110B
        MOV     XBR2, #01000000B
        MOV     POMDOUT, #0FFH
        MOV     P1MDOUT, #0FFH
        MOV     P2MDOUT, #0FFH
        MOV     P3MDOUT, #0FFH
        MOV     CKCON, #00001000B
        MOV     OSCXCN, #01100110B
        MOV     R2, #00
LD:     MOV     R3, #00
        DJNZ   R3, $
        DJNZ   R2, LD
        MOV     A, OSCXCN
        JNB   ACC. 7, LD
        MOV     OSCICN, #00011100B
        MOV     SCON0, #50H    ;
        MOV     TMOD, #20H    ;
        MOV     TH1, #68H    ;
        MOV     TL1, #68H    ;
        SETB   TR1          ;
        MOV     P1, #0FFH    ;
LOOP:   JB     P1. 3, NEXT1   ;
        JNB   P1. 3, $       ;
        MOV     SBUFO, #0DH   ;
        JNB   TI, $         ;
        CLR    TI           ;
NEXT1:  JB     P1. 4, NEXT2   ;
        JNB   P1. 4, $       ;
        MOV     SBUFO, #0EH   ;
        JNB   TI, $         ;
        CLR    TI           ;
NEXT2:  JNB   RI, LOOP       ;
        CLR   RI            ;
        MOV    A, SBUFO      ;
        CJNE  A, #0DH, LED   ;
        CLR   P1. 0         ;
        SJMP  LOOP          ;
LED:    CJNE  A, #0EH, LOOP   ;
        SETB  P1. 0         ;

```

```
SJMP LOOP ;  
END ;
```

## 五 实验报告：

1. 将实验内容 1 的显示方式 0~9 改为显示 0~F，并且相邻两个字符之间熄灭 1S
2. 设计其它双机通讯的方案及程序，关键部分加注释
3. 写出实验过程中发现的问题

# 实验五 时间控制程序

## 一 实验目的：

1. 了解带有端口操作的程序调试
2. 了解中断、串行口操作的程序调试
3. 了解在系统调试操作过程

## 二 实验设备：

1. 单片机开发系统一套
2. 单片机实验主板一块
3. 直流稳压电源一台。

## 三 实验电路：

见实验主板电原理图

## 四 实验内容：

1. 调试实验参考程序，进行汇编
2. 用修改寄存器的方法向 P1 写入适当的控制字，观察各 LED 灯及微电机动作是否正常
3. 设置 R3 的值（0—9），用通过单步运行 LCALL DISP 语句，观察数码管显示是否正确
4. 在 INTC 子程首句设置断点，复位后全速运行，观看中断产生是否正确（在断点处能停下）
5. 在每个 LOOP 设断点，思考设断点的位置，进入断点后使用单步运行程序，观察其实验现象
6. 清除断点，使用单步运行程序，比较断点与单步运行程序有什么不同？复位后全速运行程序，看定时程序是否正确进行

## 实验预习参考程序：

## 预习要求：

1. 黑体字**不得更改**，输入时不要有误
2. 补充空白语句
3. 除黑体字外程序加注释

```

$INCLUDE          (C8051F020. INC)
TIME1 EQU 2 ;
TIME2 EQU 4 ;
TIME3 EQU 6 ;
TIME4 EQU 10 ;
      ORG 0000H ;
LJMP  START ;
      ORG 000BH ;
      LJMP INTC ;
      ORG 0030H ;
START: MOV  WDTCN, #0DEH
      MOV  WDTCN, #0ADH
      MOV  XBRO, #04H
      MOV  XBR1, #1EH
      MOV  XBR2, #40H
      MOV  OSCXCN, #066H
      MOV  R2, #00H
LD:    MOV  R3, #00H
      DJNZ R3, $
      DJNZ R2, LD
      MOV  A, OSCXCN
      JNB  ACC. 7, LD
      MOV  OSCICN, #1CH
      MOV  SP, #60H ;
      MOV  TMOD, _____ ;
      MOV  DPTR, #15536 ;
      MOV  TH0, ____ ;
      MOV  TL0, ____ ;
      MOV  SCON0, #00H ;
      MOV  IE, #82H ;
      MOV  R7, #20 ;
LOOP0: MOV  R3, #01H ;
      LCALL DISP ;
      MOV  ____, #0FEH ;
      MOV  R2, #TIME1 ;
      ____ TR0 ;
LOOP1: CJNE R2, #0, LOOP1 ;
      INC  R3 ;
      LCALL DISP ;
      MOV  ____, #0FCH ;

```

```

MOV      R2, #TIME2      ;
LOOP2:  CJNE   R2, #0, LOOP2 ;
        INC    R3         ;
        LCALL  DISP       ;
        MOV    _____, #____H ;
        MOV    R2, #TIME3 ;
LOOP3:  CJNE   R2, #0, LOOP3 ;
        INC    R3         ;
        LCALL  DISP       ;
        MOV    _____, #078H ;
        MOV    R2, #TIME4 ;
LOOP4:  CJNE   R2, #0, LOOP4 ;
        SJMP   LOOP0      ;
DISP:   MOV    A, R3       ;
        ADD    A, #8       ;
        MOVC   A, @A+PC    ;
        MOV    SBUF0, A    ;
        JNB    TI, $       ;
        CLR    TI         ;
        RET                    ;
CODE1:  DB     0C0H, 0F9H, 0A4H, 0B0H, 99H ;
        DB     92H, 82H, 0F8H, 80H, 90H ;
INTC:   CLR    TR0        ;
        MOV    DPTR, #15536 ;
        MOV    TH0, DPH   ;
        MOV    TL0, DPL   ;
        SETB   _____ ;
        DJNZ   R7, INTC1 ;
        MOV    R7, #20    ;
        DEC    R2         ;
INTC1:  RETI              ;
        END                ;

```

## 五 实验报告：

1. 画出程序的流程图
2. 写出实验中断点与单步两种方法在调试程序中的不同作用
3. 总结调试中出现的问题及体会

## 实验六 外部数据存储器的扩展

### 一 实验目的：

1. 掌握外部存储器的扩展方法
2. 对外部存储器进行读写操作

## 二 实验设备：

1. 单片机开发系统一套
2. 单片机实验主板一块
3. 直流稳压电源一台。

## 三 实验电路：

见实验主板电原理图

## 四 实验内容：

- ★ 在外部数据存储器中写入一些数据，读出数据后和源数据进行比较，并利用LED显示（绿为正确、红为错误）。

## 预习要求：

1. 黑体字不得更改，输入时不要有误
2. 补充空白语句
3. 除黑体字外程序加注释

## 实验预习参考程序：

```
$INCLUDE      (C8051F020. INC)
      ORG      0000H      ;
NUM   EQU    23H        ;
      LJMP    START      ;
      ORG    0030H      ;
START: MOV    WDTCN, #0DEH
      MOV    WDTCN, #0ADH
      MOV    XBRO, #00000100B
      MOV    XBR1, #00011110B
      MOV    XBR2, #01000000B
      MOV    OSCXCN, #01100110B
      MOV    R2, #00H
LD:   MOV    R3, #00H
      DJNZ   R3, $
      DJNZ   R2, LD
      MOV    A, OSCXCN
      JNB    ACC. 7, LD
      MOV    OSICN, #00011100B
      MOV    DPTR, #8000H      ;
      MOV    A, #NUM          ;
      MOVX   @DPTR, A        ;
      NOP                   ;
      NOP                   ;
      MOVX   A, @DPTR        ;
```



```

        CJNE    A, #NUM, ERROR    ;
        CLR     _____        ;
        LJMP    OVER              ;
ERROR:   CLR     _____        ;
OVER:   AJMP    OVER              ;
        END                       ;

```

## 五 实验报告：

1. 画出设计的程序流程图
2. 写出输入十个数, 然后读出并判断的程序, 当有错误时, 停止程序, 并将错误数据的地址存入 ACC 以便检查
3. 在编写的程序中, 输入数据结束后设置断点, 执行到断点后, 更改 XRAM 中的值后再运行, 以便检查错误处理程序
4. 总结调试中出现的问题及体会

# 实验七 模数/数模转换接口实验

## 一 实验目的

1. 掌握 ADC0809 与 51 系列单片机接口电路的设计方法和 A/D 转换程序设计方法
2. 掌握 DAC0832 与 51 系列单片机接口电路的设计方法和 D/A 转换程序设计方法
3. 了解信号频率与采样速率的关系, 进一步理解信号采样定理的物理意义

## 二 实验设备

1. 单片机开发系统一套
2. 模数/数模转换实验板一块
3. 低频信号发生器一台
4. 示波器一台
5. 多路输出直流稳压电源一台
6. 万用表一块

## 三 实验原理

1. ADC0809 芯片是 8 路模拟输入, 8 位数字输出的 A/D 转换芯片, 通过设置 3 位选通地址, 可以选通 8 路中任何一路作为模拟输入 (此实验, 选择 0 和 5 通道)。当模拟输入是一个固定电压信号时, 输出一个固定的数字量; 当模拟输入是一个连续变化的电压信号时, 输出数字量在 00—FFH 范围内相应变化
2. DAC0832 是 8 位输入, 两级数字锁存, 电流输出的 D/A 芯片, 利用后接运放实现电流—电压变化。当输入锁存器一个数字时, 输出一个相应的模拟电压值

## 四 实验电路：

见实验 A/D、D/A 转换板电原理图

## 五 实验内容

1. 确定 A/D 各通道及 D/A 的地址
2. 编写程序，编译链接
3. A/D 调试：将 A/D 转换板上模拟输入端 IN0 通道接入直流电压（0V 或 5V），转换结果存入累加器 A，观察转换结果
4. D/A 调试：将 D/A 转换输出端接万用表，通过累加器 A 输入不同的数字量，观察转换结果
5. 将 A/D 转换模拟输入端 IN5 通道接低频信号发生器，D/A 转换输出接示波器，编制程序，采集 40 个数据，存入某一存储空间，然后反复将数据经 D/A 转换输出（或 A/D 输入一点即由 D/A 输出一），采用中断或查询方式均可，比较输入，输出两路模拟信号
6. 改变输入模拟信号的频率，重复执行上述程序，比较 A/D 输入，D/A 输出两路模拟信号，注意其变化
7. 自己可以设计涉及 A/D, D/A 应用（或联合应用）的其他实验方案

### 实验预习参考程序：

#### 预习要求：

1. 黑体字**不得更改**，输入时不要有误
2. 补充空白语句
3. 除黑体字外程序加注释

#### A/D 转换

```
$INCLUDE          (C8051F020. INC)
    ORG           0000H          ;
    AJMP          START          ;
    ORG           0003H          ;
    AJMP          INTO           ;
    ORG           002BH
    AJMP          CTC2
    ORG           0030H          ;
START: MOV        WDTCN, #0DEH
    MOV        WDTCN, #0ADH
    MOV        XBRO, #04H
    MOV        XBR1, #06H
    MOV        XBR2, #42H
    MOV        EMIOCF, #0FH
    MOV        CKCON, #20H
    MOV        OSCICN, #17H
    MOV        POMDOUT, #0FFH
    MOV        P1MDOUT, #0FFH
    MOV        P2MDOUT, #0FFH
    MOV        P3MDOUT, #0FFH
    SETB      EXEN2
```

```

MOV      RCAP2L, #0FFH
MOV      RCAP2H, #0FFH
SETB    TR2
SETB    ET2
SETB    EA          ;
SETB    EX0        ;
MOV      R0, #20H   ;
MOV      R1, #20H   ;
MOV      A, #_____ ;
LOOP:    MOV      DPTR, #_____ ;
MOVX    @DPTR, A    ;
AJMP    $           ;
INT0:    MOVX    A, @DPTR ;
MOV      @R0, A     ;
INC     R0          ;
MOV      A, #_____ ;
MOV      DPTR, #_____ ;
MOVX    @DPTR, A    ;
RETI    ;
CTC2:    CLR     TF2
CPL     P0.5
RETI    ;
END      ;

D/A 转换
$INCLUDE (C8051F020. INC)
ORG      0000H      ;
LJMP    START      ;
ORG      0030H      ;
START:   MOV      WDTCN, #0DEH
MOV      WDTCN, #0ADH
MOV      XBR0, #00000100B
MOV      XBR1, #00001110B
MOV      XBR2, #01000010B
MOV      EMIOCF, #00001111B
MOV      CKCON, #00001000B
MOV      OSCICN, #00010111B
MOV      POMDOUT, #0FFH
MOV      P1MDOUT, #0FFH
MOV      P2MDOUT, #0FFH
MOV      P3MDOUT, #0FFH
MOV      DPTR, _____ ;
MOV      R1, #00H      ;
NEXT:    MOV      A, R1      ;
MOVX    @DPTR, A      ;

```

```
ADD    A, #02H    ;
MOV    R1, A      ;
NOP                    ;
NOP                    ;
LJMP   NEXT      ;
END                    ;
```

## 六 实验报告

1. 记录实验内容 1~6 中，各步骤的实验条件、实验现象，并对所观察到的现象进行解释，说明
2. 列出所设计的源程序清单（每条程序加注释）
3. 设计性实验内容 7，应说明其设计思想，只涉及 A/D 或 D/A 的方案要求完成 A/D、D/A 两项实验

# 实验八 键盘显示接口实验

## 一 实验目的

1. 掌握 8255 与 51 系列单片机接口电路的设计方法。
2. 掌握利用 8255 扩展键盘，显示接口的电路设计。
3. 掌握动态显示程序的设计方法。
4. 掌握键盘扫描，键识别及键功能散转程序设计的基本思路。

## 二 实验设备

1. 单片机开发系统一套。
2. 键盘显示实验板一块。
3. 直流稳压电源一台。

## 三 实验原理

利用通用 I/O 并行扩展芯片 8255 管理键盘和显示器，8255 的 PC 口作为扫描输出，PB 口作为键盘输入，PA 口作为显示段码数据，可以管理 64 个按键和 8 位数据显示。

## 四 实验电路

见键盘显示板原理图

## 五 实验内容

1. 实验板提供 4 位 LED 显示器和 8 个键
2. 确定 8255 各口的地址
3. 编写程序;编译链接
4. 键盘调试：设计键盘扫描程序，按下某键，用累加器观察键值
5. 显示调试：通过指令设置段码数据和位码数据，观察显示结果
6. 设置键：K1-1 为个位加一键、K1-2 为个位减一键、K1-3 为左移键、K1-4 为

- 右移键、K2-1 为复位键、K2-2、K-3、K-4 键功能自己设定
7. 自己可以设计涉及键盘、显示应用（或联合应用）的其他实验方

### 实验预习参考程序：

#### 预习要求：

1. 黑体字**不得更改**，输入时不要有误
2. 补充空白语句
3. 除黑体字外程序加注释
4. 画出程序的流程图

```

$INCLUDE      (C8051F020. INC)
PPA    EQU    0CFF0H    ;
PPB    EQU    0CFF1H    ;
PPC    EQU    0CFF2H    ;
PPD    EQU    0CFF3H    ;
        ORG    0000H    ;
        LJMP   START    ;
        ORG    0030H    ;
START:  MOV    WDTCN, #0DEH
        MOV    WDTCN, #0ADH
        MOV    EMIOCF, #0FH
        MOV    XBRO, #04H
        MOV    XBR1, #0EH
        MOV    XBR2, #40H
        MOV    POMDOUT, #0FFH
        MOV    P1MDOUT, #0FFH
        MOV    P2MDOUT, #0FFH
        MOV    P3MDOUT, #0FFH
        MOV     DPTR, #PPD    ;
        MOV     A, #82H    ;
        MOVX    @DPTR, A    ;
        MOV     20H, #01H    ;
        MOV     21H, #02H    ;
        MOV     22H, #03H    ;
        MOV     23H, #04H    ;
        MOV     R7, #00H    ;被按下的按键代码，0 为无按键按下
                                ;1 为 K1-1, 2 为 K2-1, 3 为 K1-2, 4 为 K2-2
                                ;5 为 K1-3, 6 为 K2-3, 7 为 K1-4, 8 为 K2-4
LOOP:   LCALL   KEY    ;
NEXT:   DEC     R7    ;
        MOV     A, R7    ;
        ADD     A, R7    ;
        ADD     A, R7    ;

```

```

        MOV     DPTR, #KEYJMP    ;
        JMP     @A+DPTR          ;
HERE:   SJMP    HERE            ;

KEY:    LCALL   DIS              ;
        MOV     A, #0F0H         ;
        MOV     DPTR, #PPC       ;
        MOVX    @DPTR, A         ;
        MOV     DPTR, #PPB       ;
        MOVX    A, @DPTR         ;
        CPL     A                 ;
        ANL     A, #00000011B    ;
        JZ      KEY              ;
        LCALL   DIS              ;
        MOV     A, #0F0H         ;
        MOV     DPTR, #PPC       ;
        MOVX    @DPTR, A         ;
        MOV     DPTR, #PPB       ;
        MOVX    A, @DPTR         ;
        ANL     A, #00000011B    ;
        JZ      KEY              ;

```

#### 键检测子程

```

KEYC:   LCALL   DIS              ;
KEY1:   MOV     A, #0F7H         ;
        MOV     DPTR, #PPC       ;
        MOVX    @DPTR, A         ;
        MOV     DPTR, #PPB       ;
        MOVX    A, @DPTR         ;
        JNB     ACC. 0, SET1      ;
        JNB     ACC. 1, SET2      ;
KEY2:   MOV     A, #0FBH         ;
        MOV     DPTR, #PPC       ;
        MOVX    @DPTR, A         ;
        MOV     DPTR, #PPB       ;
        MOVX    A, @DPTR         ;
        JNB     ACC. 0, SET3      ;
        JNB     ACC. 1, SET4      ;
KEY3:   MOV     A, #0FDH         ;
        MOV     DPTR, #PPC       ;
        MOVX    @DPTR, A3        ;
        MOV     DPTR, #PPB       ;
        MOVX    A, @DPTR         ;

```

```

        JNB     ACC. 0, SET5      ;
        JNB     ACC. 1, SET6      ;
KEY4:   MOV     A, #0FEH          ;
        MOV     DPTR, #PPC        ;
        MOVX    @DPTR, A          ;
        MOV     DPTR, #PPB        ;
        MOVX    A, @DPTR         ;
        JNB     ACC. 0, SET7      ;
        JNB     ACC. 1, SET8      ;
        LJMP    KEY              ;
SET1:   MOV     R7, #1           ;
        RET                                ;
SET2:   MOV     R7, #2           ;
        RET                                ;
SET3:   MOV     R7, #3           ;
        RET                                ;
SET4:   MOV     R7, #4           ;
        RET                                ;
SET5:   MOV     R7, #5           ;
        RET                                ;
SET6:   MOV     R7, #6           ;
        RET                                ;
SET7:   MOV     R7, #7           ;
        RET                                ;
SET8:   MOV     R7, #8           ;
        RET                                ;

```

;键盘散转入口

```

KEYJMP: LJMP    K1_1            ;
        LJMP    K2_1            ;
        LJMP    K1_2            ;
        LJMP    K2_2            ;
        LJMP    K1_3            ;
        LJMP    K2_3            ;
        LJMP    K1_4            ;
        LJMP    K2_4            ;
K1_1:   MOV     R2, 20H          ;
        CJNE    R2, #09H, NEXT1  ;
        MOV     20H, #00H        ;
        LCALL   DIS              ;
        LJMP    LOOP            ;
NEXT1:  INC     20H              ;
        LCALL   DIS              ;
        LJMP    LOOP            ;

```

```

K1_2:  MOV    R2, 20H      ;
        CJNE  R2, #00H, NEXT2 ;
        MOV   20H, #09H   ;
        LCALL DIS        ;
        LJMP  LOOP        ;
NEXT2:  DEC   20H         ;
        LCALL DIS        ;
        LJMP  LOOP        ;
K1_3:  MOV   A, 22H      ;
        MOV   23H, A     ;
        MOV   A, 21H     ;
        MOV   22H, A     ;
        MOV   A, 20H     ;
        MOV   21H, A     ;
        MOV   20H, #00H  ;
        LCALL DIS        ;
        LJMP  LOOP        ;
K1_4:  MOV   A, 21H      ;
        MOV   20H, A     ;
        MOV   A, 22H     ;
        MOV   21H, A     ;
        MOV   A, 23H     ;
        MOV   22H, A     ;
        MOV   23H, #00H  ;
        LCALL DIS        ;
        LJMP  LOOP        ;
K2_1:  MOV   20H, #00H   ;
        MOV   21H, #00H   ;
        MOV   22H, #00H   ;
        MOV   23H, #00H   ;
        LCALL DIS        ;
        LJMP  LOOP        ;
K2_2:
        LCALL DIS        ;
        LJMP  LOOP        ;
K2_3:

```



```

        LCALL  DIS          ;
        LJMP   LOOP        ;
K2_4:

```

```

        LCALL  DIS          ;
        LJMP   LOOP        ;

```

;显示子程

```

DIS:    MOV     R0, #20H    ;
        MOV     R2, #0FEH  ;
LDIS:   MOV     DPTR, #NUM  ;
        MOV     A, @R0     ;
        MOVC   A, @A+DPTR  ;
        MOV     DPTR, #PPA ;
        MOVX   @DPTR, A    ;
        MOV     DPTR, #PPC ;
        MOV     A, R2     ;
        MOVX   @DPTR, A    ;
        LCALL  DELAY      ;
        INC    R0         ;
        RL    A           ;
        MOV     R2, A     ;
        JB     ACC. 4, LDIS ;
        RET

```

;延时子程

```

DELAY:  MOV     R3, #0AH    ;
LOOPD:  MOV     R4, #00H    ;
        DJNZ   R4, $       ;
        DJNZ   R3, LOOPD   ;
        RET

```

;数字代码表

```

NUM:    DB      7EH, 0CH, 0B6H, 9EH, 0CCH, 0DAH, 0FAH, 0EH, 0FEH, 0DEH
        ;0, 1, 2, 3, 4, 5, 6, 7, 8, 9
        END

```

## 六 实验报告

1. 记录实验内容 1~5 中，各步骤的实验条件、实验现象，并对所观察到的现象进行解释、说明

2. 列出所设计的源程序清单
3. 完成设计性实验内容 7，应说明其设计思想，只涉及键盘或显示的方案，要求完成键盘、显示两项实验

# 实验九 条件传输方式

## 一 实验目的

1. 掌握条件传输方式的基本原理和传输过程
2. 掌握条件传输方式中各联络信号的作用和基本功能
3. 了解 8255 工作方式 1 的工作过程

## 二 实验设备：

1. MCS-51 单片机实验主板一块
2. 条件传输方式实验板一块
3. 直流稳压电源一台
4. MCS-51 单片机开发系统一套

## 三 实验原理：

1. 数据的条件传输方式只有在传输条件满足传输双方约定时才进行数据传输，以保证数据传输的可靠性，适用于外设的响应速度低于 CPU 处理速度时的应用情况
2. 8255 方式 1 是条件传输方式，其传输条件（双方的联络信号）由 PC 口实现，一般在应用时联络信号自动产生，不易观察。实验板将应答一方设计为手动响应，PA 口为条件输出，指示灯 DBF 显示 PA 口锁存器的状态。手动操作 ACK 作为接受到 PA 口的输出数据的应答，PB 口为条件输入，指示灯 LBF 显示 PB 口锁存器的状态，手动操作 STB 作为向 PB 口输入新数据的选通信号

## 四 实验电路：

见实验条件传输板电原理图

## 五 实验内容：

1. 明确 8255 各口的地址
2. 编写程序；编译链接
3. 完成 8255 方式 1 的设置，PA 口条件输出，PB 口条件输入
4. 条件输入：利用 DIP 开关设置输入数据，在累加器 A 中观察输入结果
5. 条件输出：通过累加器 A 写入数据，观察发光二极管的亮灭情况
6. 自己可以设计涉及条件输入、条件输出的方案

## 实验预习参考程序：

### 预习要求：

4. 黑体字不得更改，输入时不要有误
5. 补充空白语句
6. 除黑体字外程序加注释

```
$INCLUDE          (C8051F020. INC)
```

```

PPA    EQU    0BFF0H    ;
PPB    EQU    0BFF1H    ;
PPC    EQU    0BFF2H    ;
PPK    EQU    0BFF3H    ;
      ORG    0000H    ;
      LJMP   START    ;
      ORG    0003H    ;
      LJMP   INTO     ;
      ORG    0013H    ;
      LJMP   INT1     ;
      ORG    0030H    ;
START:  MOV    WDTCN, #0DEH
      MOV    WDTCN, #0ADH
      MOV    XBRO, #00000100B
      MOV    XBR1, #00010110B
      MOV    XBR2, #01000010B
      MOV    EMIOCF, #00001111B
      MOV    POMDOUT, #0FFH
      MOV    P1MDOUT, #0FFH
      MOV    P2MDOUT, #0FFH
      MOV    P3MDOUT, #0FFH
      SETB   EA        ;
      SETB   EX0       ;
      SETB   EX1       ;
      MOV    A, #0A6H   ;
      MOV    DPTR, #PPK ;
      MOVX   @DPTR, A   ;
      MOV    A, #05H   ;
      MOVX   @DPTR, A   ;
      MOV    A, #0DH   ;
      MOVX   @DPTR, A   ;
      MOV    IE, #85H  ;
      LJMP   $         ;

```

;INT0 中断子程

```

INT0:  CLR    EA        ;
      MOV    DPTR, #PPB ;
      MOVX   A, @DPTR   ;
      MOV    R2, A      ;
      SETB   EA        ;
      RETI                ;

```

;INT1 中断子程

```
INT1:  MOV    DPTR, #PPA    ;
        MOV    A, R2        ;
        MOVX   @DPTR, A    ;
        SETB   EA          ;
        RETI                ;
        END                ;
```

## 六 实验报告：

1. 记录实验内容 1~5 中，各步骤的实验条件、实验现象，并对所观察到的现象进行解释，说明
2. 列出所设计的源程序清单（每条程序加注释）
3. 完成设计性实验内容 6，应说明其设计思想，只涉及条件输入、条件输出的方案，要求完成条件输入、条件输出两项实验