

电工电子实验中心 实验报告

课程名称： 微机原理与接口技术实验

实验名称： 8254 定时/计数器、8259 中断控制器、
8255 并口控制器、A/D 与 D/A 转换实验、
点阵 LED 显示设计实验

姓 名： 陈伟 学号： 161710223

评定成绩： 审阅教师： 黄晓晴

实验时间： 2019 年 12 月 23 日

目录

一、8254 定时/计数器	2
1.1 实验目的要求	2
1.2 实验任务	2
1.3 实验代码	2
1.4 探究内容（选做）	4
1.5 实验的运行数据及分析	5
1.6 实验讨论及心得体会	5
二、8259 中断控制器	6
2.1 实验目的要求	6
2.2 实验任务	6
2.3 实验代码	6
2.4 探究内容（选做）	10
2.5 实验的运行数据及分析	10
2.6 实验讨论及心得体会	11
三、8255 并口控制器	12
3.1 实验目的要求	12
3.2 实验任务	12
3.3 实验代码	12
3.4 探究内容（选做）	16
3.5 实验的运行数据及分析	16
3.6 实验讨论及心得体会	17
四、A/D 与 D/A 转换实验	18
4.1 实验目的要求	18
4.2 实验任务	18
4.3 实验代码	18
4.4 探究内容（选做）	21
4.5 实验的运行数据及分析	21
4.6 实验讨论及心得体会	22
五、点阵 LED 显示设计实验	23
5.1 实验目的要求	23
5.2 实验任务	23
5.3 实验代码	23
5.4 探究内容（选做）	24
5.5 实验的运行数据及分析	25
5.6 实验讨论及心得体会	27

一、8254 定时/计数器

1.1 实验目的要求

- 1) 掌握 8254 定时/计数器的各种工作方式及编程方法。

1.2 实验任务

按照图 3-2-2 的要求连线，分别对 8254 芯片的 3 个定时/计数器编程，并选择合适的工作方式和初值，以达到如下的效果：

- 1) 定时/计数器 0，计数脉冲频率为 18.432KHz，OUT0 分两路输出，一路外接 2 位 LED，使其以亮 0.5 秒灭 0.5 秒循环闪亮，另一路作为计数器 1 的计数脉冲 CLK1。
- 2) 定时/计数器 1，OUT1 的输出外接 2 位 LED，使其以亮 3 秒灭 1 秒循环闪亮。
- 3) 定时/计数器 2 的计数脉冲来自单次脉冲单元，按压开关产生的脉冲作为计数器 2 的计数脉冲。OUT2 外接 2 位 LED，当按压开关到 17 次时 LED 长亮，并将按压开关剩余次数将在屏幕上显示。

1.3 实验代码

```
IOY0    EQU    0E000H
TIMER0EQU    IOY0+00H*4    ;8254 计数器 0 端口地址
TIMER1EQU    IOY0+01H*4    ;8254 计数器 1 端口地址
TIMER2EQU    IOY0+02H*4    ;8254 计数器 2 端口地址
TCTL     EQU    IOY0+03H*4    ;8254 控制寄存器端口地址

STACK1    SEGMENT STACK
            DW 256 DUP(?)
STACK1    ENDS

DATA    SEGMENT
            MES0    DB 'Pressed: $'
            MES1    DB 'Press any key to exit!', 0DH, 0AH, '$'
            NUM      DB ?
DATA    ENDS
```

```
CODE SEGMENT
    ASSUME  CS:CODE, DS:DATA, SS:STACK1
START: MOV  AX, DATA
      MOV  DS, AX

      MOV  DX, OFFSET MES1
      MOV  AH, 9
      INT 21H

      MOV  DX, TCTL      ;控制字，选用计数器 0，写 16 位值,方式 3,二进制计数
      MOV  AL, 00110110B
      OUT  DX, AL

      MOV  DX, TIMER0      ;写入 4800H
      MOV  AL, 00H
      OUT  DX, AL
      MOV  AL, 48H
      OUT  DX, AL

      MOV  DX, TCTL      ;控制字，选用计数器 1，写低字节,方式 2,二进制计数
      MOV  AL, 01010100B
      OUT  DX, AL

      MOV  DX, TIMER1      ;写入 04H
      MOV  AL, 04H
      OUT  DX, AL

      MOV  DX, TCTL      ;控制字，选用计数器 2，写低字节,方式 0,二进制计数
      MOV  AL, 10010000B
      OUT  DX, AL

      MOV  DX, TIMER2
      MOV  AL, 0FH
      OUT  DX, AL

L1:   MOV  DX, TIMER2      ;读入计数器 2 值保存
      IN  AL, DX
      MOV  NUM, AL
      CALL DISP

      MOV  AL, NUM
      CMP  AL, 0
```

```

JZ QUIT                ;计数至 0 时退出

MOV  DL, 0FFH          ;判主键盘有无键按下
MOV  AH, 6
INT 21H
JZ  L1                 ;无键按下跳转

QUIT: MOV  AX, 4C00H    ;结束程序退出
      INT 21H

DISP  PROC             ;显示子程序
      MOV  DX, OFFSET MES0      ;显示 MES0
      MOV  AH, 9
      INT 21H

      MOV  AL, NUM
      CMP  AL, 9              ;判断是否<=9
      JLE L2                 ;若是则为'0'-'9',ASCII 码加 30H
      ADD  AL, 7              ;否则为' A'-'F',ASCII 码加 37H

L2:   ADD  AL, 30H            ;在显示器上显示按压开关的次数
      MOV  DL, AL
      MOV  AH, 2
      INT 21H
      MOV  DL, 0DH
      INT 21H

      RET

DISP  ENDP

CODE  ENDS
      END          START

```

1.4 探究内容（选做）

- 1) 尝试改变计数器的工作方式或初值，观察输出的变化；
- 2) 利用实验箱上的其他单元，构造其他的实验方案。例如，输出可控频率的方波至电子发声单元，使其发出预设的声音甚至音乐。

1.5 实验的运行数据及分析

对于探究内容一改变工作方式或初值，观察输出结果。先写下本实验的补全代码部分的工作思路。

(1) 以 18.432KHz 的频率，实现亮 0.5 秒，灭 0.5 秒，显然这是个方波信号发生器，所以采用方式 3 计数。初值= $T \times f_{clk} = 1 \times 18432 = 4800H$

(2) 以 OUT1 为 CLK(1Hz)，实现亮 3 秒，灭 1 秒，这是个速率信号发生器，所以采用方式 2 计数。初值= $T \times f_{clk} = 4 \times 1 = 4 = 04H$

(3) 按 17 次 LED 常亮，那么采用方式 0 计数，初值为 16，那么第 17 个脉冲到来的时候，OUT2 输出 1，LED 常亮。

课堂增加考核：实验如果使计数器 0 实现小灯亮 1 秒，灭 1 秒，只需要使初值为 $2 \times 18432 = 9000H$ ，此处**注意：**以二进制计数的话，计数最大值只能为 25536(初值为 0)，超过的话可以通过两个计数器级联的方式计数。

实验结果与预期结果相同，结果正确

1.6 实验讨论及心得体会

通过本实验的操作，我对 8254 定时/计数器的工作特点与编程方式掌握的比较熟练了。对于 8254 关键是要掌握 6 种不同的工作方式以及初值的计算。实验过程没有太大的困难。

二、8259 中断控制器

2.1 实验目的要求

1. 学习中断控制器 8259 的工作原理。
2. 掌握可编程控制器 8259 的应用编程方法。

2.2 实验任务

用单次脉冲单元的 KK1+和 KK2+产生单次脉冲，模拟两个中断源，在 KK1+触发的中断服务程序中显示字符“0”，在 KK2+触发的中断服务程序中显示字符“1”。

2.3 实验代码

需要修改的地方已用红色标注

```
INTR_IVADD    EQU    003CH ;INTR 对应的中断矢量地址
INTR_OCW1     EQU    021H  ;INTR 对应 PC 机内部 8259 的 OCW1 地址
INTR_OCW2     EQU    020H  ;INTR 对应 PC 机内部 8259 的 OCW2 地址
INTR_IM       EQU    07FH  ;INTR 对应的中断屏蔽字
IOY0          EQU    0E000H ;片选 IOY0 对应的端口始地址

MY8259_ICW1   EQU    IOY0+00H ;实验系统中 8259 的 ICW1 端口地址
MY8259_ICW2   EQU    IOY0+04H ;实验系统中 8259 的 ICW2 端口地址
MY8259_ICW3   EQU    IOY0+04H ;实验系统中 8259 的 ICW3 端口地址
MY8259_ICW4   EQU    IOY0+04H ;实验系统中 8259 的 ICW4 端口地址
MY8259_OCW1   EQU    IOY0+04H ;实验系统中 8259 的 OCW1 端口地址
MY8259_OCW2   EQU    IOY0+00H ;实验系统中 8259 的 OCW2 端口地址
MY8259_OCW3   EQU    IOY0+00H ;实验系统中 8259 的 OCW3 端口地址

STACK1        SEGMENT STACK
                DW 256 DUP(?)
STACK1        ENDS

DATA          SEGMENT
    MES        DB    'Press any key to exit!',0AH,0DH,0AH,0DH,'$'
    CS_BAK     DW ? ;保存 INTR 原中断处理程序入口段地址的变量
```

```

        IP_BAK DW ?           ;保存 INTR 原中断处理程序入口偏移地址的变量
        IM_BAKDB ?           ;保存 INTR 原中断屏蔽字的变量
DATA     ENDS

CODE     SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:STACK1

START:
        MOV     AX, DATA
        MOV     DS, AX
        MOV     DX, OFFSET MES      ;显示退出提示
        MOV     AH, 09H
        INT     21H

        CLI

        MOV     AX, 0000H           ;替换 INTR 的中断矢量
        MOV     ES, AX
        MOV     DI, INTR_IVADD      ;保存 INTR 原中断处理程序入口偏移地址
        MOV     AX, ES:[DI]
        MOV     IP_BAK, AX
        MOV     AX, OFFSET MYISR     ;设置当前中断处理程序入口偏移地址
        MOV     ES:[DI], AX

        ADD     DI, 2               ;保存 INTR 原中断处理程序入口段地址
        MOV     AX, ES:[DI]
        MOV     CS_BAK, AX
        MOV     AX, SEG MYISR        ;设置当前中断处理程序入口段地址
        MOV     ES:[DI], AX

        MOV     DX, INTR_OCW1       ;设置中断屏蔽寄存器，打开 INTR 的屏蔽位
        IN      AL, DX               ;保存 INTR 原中断屏蔽字
        MOV     IM_BAK, AL

        AND     AL, INTR_IM          ;允许 PC 机内部 8259 的 IR7 中断
        OUT     DX, AL

        MOV     DX, MY8259_ICW1      ;初始化实验系统中 8259 的 ICW1
        MOV     AL, 13H               ;边沿触发、单片 8259 、需要 ICW4
        OUT     DX, AL

        MOV     DX, MY8259_ICW2      ;初始化实验系统中 8259 的 ICW2
        MOV     AL, 08H
        OUT     DX, AL

```



```

MOV DX, MY8259_ICW4    ;初始化实验系统中 8259 的 ICW4
MOV AL, 01H            ;非自动结束 EOI
OUT DX, AL

MOV DX, MY8259_OCW3    ;向 8259 的 OCW3 发送读取 IRR 命令
MOV AL, 0AH
OUT DX, AL

MOV DX, MY8259_OCW1    ;初始化实验系统中 8259 的 OCW1
MOV AL, 0FCH           ;打开 IR0 和 IR1 的屏蔽位
OUT DX, AL
STI

WAIT1:
MOV AH, 1              ;判断是否有按键按下
INT 16H                ;无按键则跳回继续等待，有则退出
JZ WAIT1

QUIT:
CLI
MOV AX, 0000H          ;恢复 INTR 原中断矢量
MOV ES, AX
MOV DI, INTR_IVADD     ;恢复 INTR 原中断处理程序入口偏移地址
MOV AX, IP_BAK
MOV ES:[DI], AX

ADD DI, 2
MOV AX, CS_BAK         ;恢复 INTR 原中断处理程序入口段地址
MOV ES:[DI], AX
MOV DX, INTR_OCW1
MOV AL, IM_BAK         ;恢复 INTR 原中断屏蔽寄存器的屏蔽字
OUT DX, AL
STI
MOV AX, 4C00H          ;返回到 DOS
INT 21H

MYISR PROC NEAR        ;中断处理程序 MYISR
PUSH AX
QUERY:                 ;向 8259 的 OCW3 发送读取 IRR 命令
MOV DX, MY8259_OCW3
IN AL, DX              ;读出 IRR 寄存器值

```

```
    AND AL, 03H
    CMP AL, 01H
    JE  IR0ISR      ;若为 IR0 请求，跳到 IR0 处理程序
    JNE IR1ISR      ;若为 IR1 请求，跳到 IR1 处理程序
    JMP OVER

IR0ISR:              ;IR0 处理，显示字符串 STR0
    MOV AL, 30H
    MOV AH, 0EH
    INT 10H
    MOV AL, 20H
    INT 10H
    JMP OVER

IR1ISR:              ;IR1 处理，显示字符串 STR1
    MOV AL, 31H
    MOV AH, 0EH
    INT 10H
    MOV AL, 20H
    INT 10H
    JMP OVER

OVER:
    MOV DX, INTR_OCW2    ;向 PC 机内部 8259 发送中断结束命令
    MOV AL, 20H
    OUT DX, AL
    MOV AL, 20H
    OUT 20H, AL
    POP AX

    IRET

MYISR ENDP

CODE      ENDS
END START
```

2.4 探究内容（选做）

无探究内容

2.5 实验的运行数据及分析

（1）实验原理

```
INTR_IVADD    EQU    003CH    ;INTR 对应的中断矢量地址
INTR_OCW1     EQU    021H     ;INTR 对应 PC 机内部 8259 的 OCW1 地址
INTR_OCW2     EQU    020H     ;INTR 对应 PC 机内部 8259 的 OCW2 地址
INTR_IM       EQU    07FH     ;INTR 对应的中断屏蔽字
IOY0          EQU    0E000H    ;片选 IOY0 对应的端口始地址
```

实验箱上采用的 INTR0 对应系统内部的 IR7（7 号中断），所以把中断矢量地址改为 003CH，中断屏蔽字改为 07FH，将 IR7 取消屏蔽。

实验大致原理为用单脉冲模拟一次中断，那么产生一个中断请求（INT 为 1），此时 INT 引发了系统内部的 7 号中断，由于修改了 7 号中断的程序，所以调用我们自己写的程序执行。我们的中断服务程序先读取实验箱上 8259 的 IRR（中断请求寄存器），然后判断是 0 号中断还是 1 号中断，分别显示 0 或 1。

读取 IRR

```
MYISR PROC NEAR                ;中断处理程序 MYISR
    PUSH AX
QUERY:                          ;向 8259 的 OCW3 发送读取 IRR 命令
    MOV DX, MY8259_OCW3
    IN  AL, DX                  ;读出 IRR 寄存器值
```

判断中断类型号

```
AND AL, 03H
CMP AL, 01H
JE  IR0ISR                     ;若为 IR0 请求，跳到 IR0 处理程序
JNE IR1ISR                     ;若为 IR1 请求，跳到 IR1 处理程序
JMP OVER
```

(2) 修改 1

如果选用实验箱上 INTR1，对应系统内部的 IR6，此时修改中断矢量地址和 INTR 对应的中断屏蔽字即可

INTR_IVADD	EQU	0038H	← INTR对应的中断矢量地址
INTR_OCW1	EQU	021H	← INTR对应PC机内部8259的OCW1地址
INTR_OCW2	EQU	020H	← INTR对应PC机内部8259的OCW2地址
INTR_IM	EQU	0BFH	← INTR对应的中断屏蔽字
IOY0	EQU	0E000H	← 片选IOY0对应的端口始地址

(3) 修改 2

如果实验箱上的 8259 中断选用 IR2 和 IR3，那么要修改对应的屏蔽位，修改为 0F3H 即可。底下读取 IRR 后的判别也做出相应修改即可。

```

MOV DX, MY8259_OCW1    ;初始化实验系统中8259的OCW1
MOV AL, 0F3H            ;打开IR2和IR3的屏蔽位
OUT DX, AL
STI

AND AL, 0CH
CMP AL, 04H
JE IR2ISR                ;若为IR2请求，跳到IR2处理程序
JNE IR3ISR               ;若为IR3请求，跳到IR3处理程序
JMP OVER

```

实验结果与预期结果相同，结果正确

2.6 实验讨论及心得体会

通过本实验的操作，我对 8259 中断控制器的工作特点与编程方式掌握的比较熟练了。重点掌握初始化命令字 ICW1、ICW2、ICW3、ICW4，控制命令字 OCW1、OCW2、OCW3。实验过程没有遇到太大的困难。

三、8255 并口控制器

3.1 实验目的要求

1. 掌握 8255 的工作方式及应用编程。
2. 学习键盘扫描的原理及电路接法。
3. 掌握利用 8255 实现按键扫描及数码管显示方法。

3.2 实验任务

用 8255 实现键盘扫描与显示功能, 当小键盘有键按下时, 键值在数码管最右位的位置上显示, 数码管上显示最新六位键值内容, 当主键盘有键按下时, 返回系统。

3.3 实验代码

```
IOY0 EQU 0E000H ;片选 IOY0 对应的端口始地址
PA55 EQU IOY0+00H*4 ;8255 的 A 口地址
PB55 EQU IOY0+01H*4 ;8255 的 B 口地址
PC55 EQU IOY0+02H*4 ;8255 的 C 口地址
PCTL EQU IOY0+03H*4 ;8255 的控制寄存器地址

DATA SEGMENT
    BUFF DB 6 DUP(10H)
    TABLE1 DB 11H,21H,41H,81H,12H,22H,42H,82H ;取反后的键盘扫描
    DB 14H,24H,44H,84H,18H,28H,48H,88H ;数码管的段码表
    DCTBL DB 3FH,06H,5BH,4fh,66h,6dh,7dh,07h,7fh,6fh
    DB 77h, 7ch,39h,5eh,79h,71h,00h

    MES DB 'Press any key on the small keyboard!',0DH,0AH
    DB 'Press key to display on the led!',0dh,0ah,'$'
    MESS DB 'Press main keyboard any key to exit!', 0dh, 0ah, 0dh, 0ah, '$'
    KEYC DB ?
    KEY DB ?
DATA ENDS

STAC SEGMENT PARA STACK
```

```
        DB 256 DUP(?)
STAC    ENDS
```

```
CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA, SS:STAC
```

```
START:
```

```
    MOV    AX, DATA
    MOV    DS, AX
    MOV DX, OFFSET MES
    MOV AH, 9
    INT 21H
```

```
    MOV DX, OFFSET MESS
    MOV AH, 9
    INT 21H
```

```
LOP1:
```

```
    CALL TESTKEY
    CALL DISP
    MOV DL, 0FFH
    MOV AH, 6
    INT 21H
    JZ LOP1
```

```
QUIT:
```

```
    MOV AX, 4C00H
    INT 21H
```

```
TESTKEY PROC
```

```
KEY0:
```

```
    MOV AL, 81H                ;8255 控制字 PC0-3 入,PC4-7 出
    MOV DX, PCTL
    OUT DX, AL
```

```
    MOV AL, 00                ;C 口输出 0
    MOV DX, PC55
    OUT DX, AL
```

```
    IN AL, DX                  ;读入行值，屏蔽列值后保存
    AND AL, 0FH
    MOV KEYC,AL
```

KEY1:

```
MOV AL, 88H           ;8255 控制字 PC0-3 出,PC4-7 入
MOV DX, PCTL
OUT DX, AL
```

```
MOV AL, 00           ;C 口输出 0
MOV DX, PC55
OUT DX, AL
IN AL, DX
AND AL, 0F0H         ;读入列值,屏蔽行值后合并取反
OR AL, KEYC
NOT AL
CMP AL, 0             ;无键按下退出子程序
JZ KEYEND
```

```
MOV SI, OFFSET TABLE1 ;查找按键的值
MOV CX, 16
MOV DL, 00H
```

KEY2:

```
CMP AL,[SI]
JZ KEY3
INC SI
INC DL
DEC CX
JZ KEYEND
JMP KEY2
```

KEY3:

```
MOV KEY, DL
MOV SI, OFFSET BUFF+1
MOV DI, OFFSET BUFF
MOV CX, 5
```

KEY4: ;显示缓冲区内容向前移一位

```
MOV AL, [SI]
MOV [DI], AL
INC SI
INC DI
LOOP KEY4
```

```
MOV AL, KEY           ;当前键值存入 BUF[5]单元
MOV [DI], AL
```

```

MOV AL, 88H           ;8255 控制字,PC0-3 出 ,PC4-7 入
MOV DX, PCTL
OUT DX, AL

```

KEY5:

```

MOV AL, 00           ;判断按键是否释放
MOV DX, PC55
OUT DX, AL
IN AL, DX
AND AL, 0F0H
CMP AL, 0F0H
JNZ KEY5

```

KEYEND:

```

RET
TESTKEY ENDP

```

DISP PROC

```

PUSH DS
PUSH AX
MOV CL, 1
MOV SI, OFFSET BUFF

```

DIS2:

```

MOV AL, [SI]         ;输出段码
LEA BX, DCTBL
XLAT
MOV DX, PB55
OUT DX, AL
MOV DX, PA55         ;输出位码
MOV AL, CL
NOT AL
OUT DX, AL
CALL DELAY
INC SI               ;段码地址+1
ROL CL, 1           ;位码左移 1 位
CMP CL, 40H         ;位码是最后位码?
JNZ DIS2            ;不是转 DIS2
POP AX              ;是返回
POP DS
RET

```

DISP ENDP

DELAY PROC NEAR


```

PUSH CX
PUSH BX
MOV BX, 80H

```

DEL1:

```

MOV CX, 0FFFFH
LOOP $
DEC BX
JNZ DEL1
POP BX
POP CX
RET

```

DELAY ENDP

```

CODE ENDS
END START

```

3.4 探究内容（选做）

- 1) 尝试改变参考程序中预设的键值，按下小键盘后在数码管上显示其他内容。
- 2) 尝试改变实验方案中的输出方式。例如，利用实验箱上的点阵 LED 显示单元用 8255 的 A 口和 B 口控制其显示更为丰富的内容。

3.5 实验的运行数据及分析

实验结果与预期结果相同，结果正确

```

DIS2:
MOV AL, [SI]           ;输出段码
LEA BX, DCTBL
XLAT
MOV DX, PB55
OUT DX, AL
MOV DX, PA55           ;输出位码
NOT AL
OUT DX, AL
CALL DELAY
INC SI                 ;段码地址+1
ROL CL, 1              ;位码左移1位
CMP CL, 40H           ;位码是最后位码?
JNZ DIS2               ;不是转DIS2
POP AX                 ;是返回
POP DS
RET
DISP ENDP

```

输出位码的时候要用 NOT 指令，因为共阴极数码管只接 0 时才亮，而根据题

意, 此时应该保持其余 5 个不亮(1), 当前亮(0), 所以需要加一个 NOT 指令, 将位码取反。

```
TESTKEY PROC
KEY0:
    MOV AL, 81H           ;8255控制字PC0-3入,PC4-7出
    MOV DX, PCTL
    OUT DX, AL

    MOV AL, 00           ;C口输出0
    MOV DX, PC55
    OUT DX, AL
    IN AL, DX           ;读入行值,屏蔽列值后保存
    AND AL, 0FH
    MOV KEYC,AL

KEY1:
    MOV AL, 88H           ;8255控制字PC0-3出,PC4-7入
    MOV DX, PCTL
    OUT DX, AL

    MOV AL, 00           ;C口输出0
    MOV DX, PC55
    OUT DX, AL
    IN AL, DX           ;读入列值,屏蔽行值后合并取反
    AND AL, 0F0H
```

这个实验的重点是在这里, 读取键值, 看懂原理不是很难。

这个实验还有个问题, 关于键盘消抖的问题, 键盘读到一个数据后, 延长一定的时间再读一次键盘, 就能很好的解决这个问题。重复以下这个代码

```
KEY1:
    MOV AL, 88H           ;8255 控制字 PC0-3 出,PC4-7 入
    MOV DX, PCTL
    OUT DX, AL

    MOV AL, 00           ;C 口输出 0
    MOV DX, PC55
    OUT DX, AL
    IN AL, DX
    AND AL, 0F0H         ;读入列值,屏蔽行值后合并取反
    OR AL, KEYC
    NOT AL
    CMP AL, 0           ;无键按下退出子程序
    JZ KEYEND
```

3.6 实验讨论及心得体会

通过本实验的操作, 我对 8255 并口控制器的工作特点与编程方式掌握的比较熟练了。实验中没有遇到什么困难, 代码都能看懂了。

四、A/D 与 D/A 转换实验

4.1 实验目的要求

1. 学习掌握模/数, /数模信号转换基本原理。
2. 掌握 ADC0809, 0832 芯片的使用方法。

4.2 实验任务

从键盘输入两位十六进制值，经 0832 转换成数字量后，送 0809 的通道 0，再经 A/D 转换后在屏幕上显示此数字量。

4.3 实验代码

```
CRLF MACRO                                ;宏定义了回车
    MOV DL, 0DH
    MOV AH, 02H
    INT 21H
    MOV DL, 0AH
    INT 21H
    ENDM

IOY0 EQU 0E000H        ;片选 IOY0 对应的端口始地址
IOY1 EQU 0E040H        ;片选 IOY1 对应的端口始地址
ADCS EQU IOY0          ;AD0809 的端口地址
DACS EQU IOY1          ;DAC0832 的端口地址

STAC SEGMENT PARA STACK
    DW 256 DUP(?)
STAC ENDS

DATA SEGMENT
    MES0 DB 'PRESS 1 TO INPUT DATA!', 0DH, 0AH
           DB 'PRESS 2 TO QUIT!', 0DH, 0AH, 0DH, 0AH, '$'
    MES1 DB '*****PLEASE INPUT DATA OF HEX!****', 0DH, 0AH, '$'
    MES2 DB '0832 OUTPUT DATA  =  $'
    MES3 DB '0809 INPUT DATA =  $'
    BUF DB 2 DUP(?)
DATA ENDS
```

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STAC

START:

MOV AX, DATA
MOV DS, AX

LEA DX, MES0
MOV AH, 9
INT 21H

LOP1:

MOV DL, 0FFH ;检测键盘输入
MOV AH, 6
INT 21H
JZ LOP1

CMP AL, '1'
JZ DA
CMP AL, '2'
JZ EXIT0
JMP START

EXIT0: JMP EXIT

DA: LEA DX, MES1 ;显示 MES1
MOV AH, 9
INT 21H

LEA DX, MES2 ;显示 MES2
MOV AH, 9
INT 21H

MOV AH, 1 ;十六进制值存入 BUF 和 BUF[1]
INT 21H
MOV BUF, AL
INT 21H
MOV BUF[1], AL
MOV AH, 2
CRLF

;十六进制转换十进制

DA0:

MOV AL, BUF
SUB AL, 30H

```
CMP AL, 9
JBE A0
SUB AL, 7

A0:                                ;十六进制转换十进制
MOV BL, AL
MOV AL, BUF[1]
SUB AL, 30H
CMP AL, 9
JBE B0
SUB AL, 7

B0:
MOV CL, 4
ROL BL, CL
XOR AL, BL
MOV DX, DACS                      ;启动 0832
OUT DX, AL

AD:                                ;启动 0809 INO
MOV DX, ADCS
OUT DX, AL
CALL DELAY
LEA DX, MES3                      ;显示 MES3
MOV AH, 9
INT 21H

MOV DX, ADCS                      ;读入 0809 INO 值
IN AL, DX
MOV BL, AL
AND AL, 0F0H                      ;显示高位

MOV CL, 4
ROL AL, CL
CALL CRT1
MOV AL, BL                        ;显示低位
AND AL, 0FH
CALL CRT1
CRLF
INT 21H
JMP START

EXIT:
MOV AX, 4C00H
```

```

INT 21H

CRT1 PROC                ;在屏幕上显示一位 16 进制字符
    ADD AL, 30H
    CMP AL, 39H
    JBE D0
    ADD AL, 7

D0: MOV DL, AL
    MOV AH, 2
    INT 21H
    RET
CRT1 ENDP

DELAY PROC NEAR
    PUSH CX
    MOV CX, 0FFFFH
    LOOP $
    POP CX
    RET
DELAY ENDP

CODE ENDS
    END START

```

4.4 探究内容（选做）

1) 假设需要周期性的定时对某模拟信号持续采样，进行 A 转换后输出，利用实验箱上的现有资源设计实验方案。

课程设计再完成

4.5 实验的运行数据及分析

本实验难度不是很大，先输入两位十六进制数（为 ASCII 码），再转化为二进制，此时为数字量，通过 DAC0832 将数字量转化为模拟量，out 接 ADC0809 的 IN0，再将转换的数字量转化为 ASCII 码显示在屏幕上。实验结果与预期结果符合，（有一定的误差，误差在范围内），实验正确。也通过将 OUT 接一个小灯，当输入的值不同时，灯的亮度也不一样，值越大，亮度越大。

```

B0:
    MOV CL, 4
    ROL BL, CL
    XOR AL, BL
    MOV DX, DACS      ;启动0832
    OUT DX, AL

AD:                                ;启动0809 IN0
    MOV DX, ADCS
    OUT DX, AL
    CALL DELAY
    LEA DX, MES3      ;显示MES3
    MOV AH, 9
    INT 21H

    MOV DX, ADCS      ;读入0809 IN0值
    IN AL, DX

```

4.6 实验讨论及心得体会

通过本实验的操作，我对 DAC0832 和 ADC0809 工作原理掌握的比较熟练了。实验中，也发现了转换存在精度损失，不过也在范围以内，实验难度不大，没有出现调试的困难，实验顺利。

五、点阵 LED 显示设计实验

5.1 实验目的要求

1. 了解 LED 点阵的基本结构。
2. 学习 LED 点阵扫描显示程序的设计方法。

5.2 实验任务

在 TD-PIT++实验系统上的 LED 点阵单元采用了 4 片 $\Phi 1.9$ 的共阴极 LED 点阵组成 16×16 的点阵。利用取字模软件得到汉字字符数组，设计程序，在点阵上滚动显示“南京航空航天大学”。

5.3 实验代码

```
;Led-HZ.asm,32 位 LED 点阵汉字显示实验
INCLUDE NUAA.inc
.386P
IOY0      EQU    0E000H          ;片选 IOY0 对应的端口始地址
STACK1 SEGMENT STACK
          DW 256 DUP(?)
STACK1 ENDS
DATA     SEGMENT  USE16
ADDR     DW      ?
DATA     ENDS
CODE SEGMENT  USE16
          ASSUME CS:CODE,DS:DATA
START: MOV  AX,DATA
          MOV  DS,AX
A2:      MOV  ADDR,OFFSET HZDOT    ;取汉字数组始地址
          MOV  SI,ADDR
A1:      MOV  CX,20H                ;控制 1 屏显示时间
LOOP2: CALL DISPHZ
          SUB  SI,32
          LOOP LOOP2
KEY:     MOV  AH,1                  ;判断是否有按键按下?
          INT  16H
          JNZ  QUIT
          ADD  SI,2
          MOV  AX,SI
```



```

        SUB  AX,ADDR
        CMP  AX,288                ;比较文字是否显示完毕
        JNB  A2
        JMP  A1
QUIT:   MOV  EAX,0                  ;灭灯
        MOV  DX,IOY0
        OUT  DX,EAX
        MOV  AX,4C00H              ;结束程序退出
        INT  21H
DISPHZ PROC NEAR                  ;显示 1 屏汉字子程序
        PUSH CX
        MOV  CX,16
        MOV  BX,0FFFEH
LOOP1:  MOV  AL,BYTE PTR[SI]
        MOV  AH,BYTE PTR[SI+1]
        ROL  EAX,16
        MOV  AX,BX
        ADD  SI,2
        ROL  BX,1
        NOT  EAX
        MOV  DX,IOY0
        OUT  DX,EAX
        CALL DALLY
        LOOP LOOP1
        POP  CX
        RET
DISPHZ ENDP
DALLY  PROC NEAR                  ;软件延时子程序
        MOV  AX,0FFFFH
D1:     DEC  AX
        JNZ  D1
        RET
DALLY  ENDP
CODE ENDS
        END START

```

5.4 探究内容（选做）

探究将屏幕上的字转 90 度，这个内容写在 5.5 部分吧。

5.5 实验的运行数据及分析

(1) 实验原理

```

DISPHZ PROC NEAR                                ;显示1屏汉字子程序
    PUSH CX
    MOV CX, 16
    MOV BX, 0FFFEH
LOOP1:  MOV AL, BYTE PTR[SI]
        MOV AH, BYTE PTR[SI+1]
        ROL EAX, 16
        MOV AX, BX
        ADD SI, 2
        ROL BX, 1
        NOT EAX
        MOV DX, IOY0
        OUT DX, EAX
        CALL DALLY
        LOOP LOOP1
    POP CX
    RET

```

实验中 AL, AH 这里存着的是字模码的某行对应的列值，将这个存放在 EAX 的高 16 位，EAX 的低 16 位是 BX(行值)的值，最后要将 EAX 取反输出。

这个字是从上往下输出的，例如 BX 为 0FFFE，那么这个第 0 位是接在 R0 上的，取反后 R1-R7 为 0，R0 为 1，第 1 行显示，接着依次显示第 2 行、第 3 行，不断刷新，通过视觉停留营造一种显示汉字的感觉。

```

KEY:  MOV AH, 1                                ;判断是否有按键按下?
      INT 16H
      JNZ QUIT
      ADD SI, 2
      MOV AX, SI
      SUB AX, ADDR
      CMP AX, 288                               ;比较文字是否显示完毕
      JNB A2
      JMP A1

```

这里有个+2，相当于对于这个汉字，显示时往上移了一行，最底下一行显示的是下一个汉字的第一行，那么一直+2 的过程，显示结果为将汉字从下往上移动。

(2) 修改显示时间

修改这个地方可以使显示的时间变长, 因为是通过 loop 指令反复的执行显示同样的数据, 所以 CX 值越大, 显示时间就越长。

```

START: MOV AX, DATA
      MOV DS, AX
A2:    MOV ADDR, OFFSET HZDOT      ;取汉数组始地址
      MOV SI, ADDR
A1:    MOV CX, 0FFFFH              ;控制1屏显示时间
LOOP2: CALL DISPHE
      SUB SI, 32

```

(3) 旋转字体

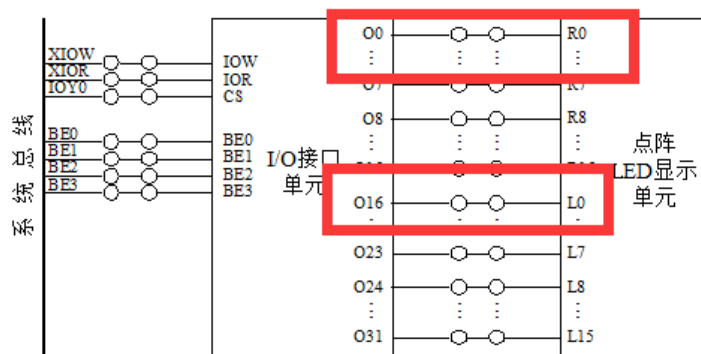
```

      MOV BX, 07FFFFH
LOOP1: MOV HL, BYTE PTR[SI]
      MOV AH, BYTE PTR[SI+1]
      ROL EAX, 16
      MOV AX, BX
      ROR EAX, 16
      ADD SI, 2
      ROL BX, 1

```

这是将汉字转 90 度的代码, 实验时发现有些小问题, 不过大体功能实现了, 主要原理是 BX 存放在 EAX 的高 16 位, AX 存放在 EAX 低 16 位, 那么 AX 控制行, BX 控制列, 此处不要用 NOT 因为对于共阳极 LED 来说, 其中每一行共用一个阳极 (行控制), 每一列共用一个阴极 (列控制), 此时正好控制了灯的亮灭。

实验时显示的汉字有左右颠倒, 没找到原因。走在路上的时候突然想到可能是数据从高到低存放还是从低到高没有把握好, 因为没有实验箱, 所以没法调试了, 修改的代码可能会有点小问题, 但是整体思路是对的。



5.6 实验讨论及心得体会

本次实验还是很有意思的，在点阵 LED 上显示“南京航空航天大学”，代码现成的，接线也很简单，但是看懂代码还是不太容易，最后在老师的讲解下还是看懂了。一开始不了解视觉停留，所以对于能在整个屏幕显示汉字不是很能理解，最后理解之后发现原理其实很简单，最终完成实验并且实现了汉字转 90 度的拓展实验。