

用普通 I/O 口驱动 LCD 显示

文件编码: HA0092s

介绍:

在一些特定环境, 为了节省成本, 控制 I/O 口需求较少, 但芯片本身的 I/O 口又较多的情况下, 客户往往用普通 I/O 口驱动 LCD 显示, 而且在实际应用中很多学习者也需求这方面的知识, 所以下面给出一个范例, 以供参考。

应用说明:

1. 当用 I/O 口做 LCD 显示时, 用 I/O 口只能做到 3 种电平信号, 所以只能做 1/2Bias。
2. 在本试验中用 HT46R22(24PIN)做控制, 电路如图 1 所示, 其中 PA0~PA3、PB3~PB5 做 SEG0~SEG5, PB3、PB6、PB7 做 COM0~COM2。

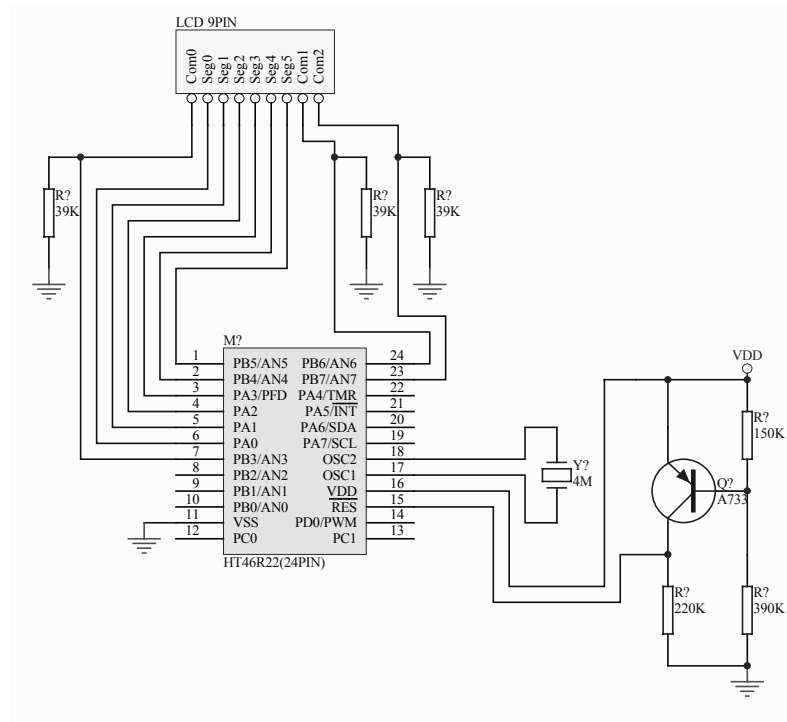


图 1

3. I/O 口 3 种电平的设置方法: 对 COM 口, 设置 IC 内部设置上拉电阻, 并外挂下拉电阻钳位, 例如, 输出高=>输入=>输出低, 这样就得到了 V_{DD} 、 $1/2V_{DD}$ 、 $0V_{DD}$ 三种电压。
4. 显示原理如图 2 所示, 在 LCD 点亮时, 对应的液晶段 COM 和 SEG 加 $\pm V_{DD}$, LCD 段就亮; 加 $0V_{DD}$ 或 $1/2V_{DD}$ 时 LCD 灭。显示时对液晶段先加 $+V_{DD}$, 再加 $-V_{DD}$ 的原理可以延长 LCD 的使用寿命。如果 LCD 玻璃是 3V, 则 $V_{DD}=3V$; 如果 LCD 玻璃是 5V 的, 则 $V_{DD}=5V$ 。
5. 范例程序是以分钟计时的方式, 从 “00” ~ “59” 循环显示。

程序清单:

```
*****
;FILE NAME:      IO_LCD
;MCU:            HT46R22
;MASK OPTION:   WAKE-UP: PA6,PA7
;                PULL-HIGH: PA,PB,PC
;                IIC: DISABLE
;                PFD: DISABLE
;                PWM: DISABLE
;                WDT: ENABLE
;                CLRWDT: ONE
;                WDT CLOCK SOURCE: T1
;                WDT TIME OUT SELECT: WDT CLOCK SOURCE/32768
;                LVR: DISABLE
;                OSC: CRYSTAL
;                SYSVOLT: 3.0V
;                SYSFRAG: 4000KHZ
;AUTHOR:        RADOME
;HISTORY:       2005.08.22
*****
include Ht46r22.inc
include Micro.inc
;*****
IO_data .section 'data'
;*****
acc_bk      db      ?
status_bk   db      ?
pa_bk       db      ?
pb_bk       db      ?
pbc_bk      db      ?

pulse_count db      ?
display_temp db      ?
display_temp0 db      ?
display_temp1 db      ?
display_temp2 db      ?
number0     db      ?
number1     db      ?
number2     db      ?

msecond     db      ?
second      db      ?
minute      db      ?
```

```

;-----
segment0      equ    pa_bk.0
segment1      equ    pa_bk.1
segment2      equ    pa_bk.2
segment3      equ    pa_bk.3
segment4      equ    pb_bk.4
segment5      equ    pb_bk.5

com0          equ    pb_bk.3
com0_ctrl     equ    pbc_bk.3
com1          equ    pb_bk.6
com1_ctrl     equ    pbc_bk.6
com2          equ    pb_bk.7
com2_ctrl     equ    pbc_bk.7
;*****
IO_code .section 'code'
;*****
        org    0000h
        jmp    init

        org    0004h                ;External Interrupt
        reti

        org    0008h                ;Timer Interrupt
        jmp    timer_int

        org    0020h

;*****
;Initializers
;*****
init:
        clr    intc0
        clr    intc1
        clr    tmrc

        clr    msecond
        clr    second
        clr    minute
        clr    pulse_count
        clr    display_temp
        clr    display_temp0
        clr    display_temp1
        clr    display_temp2
    
```

```

    clr    number0
    clr    number1
    clr    number2

    mov    a,11011111b
    mov    pa,a
    mov    pa_bk,a
    mov    a,11010000b
    mov    pac,a
    set    pb
    set    pb_bk
    mov    a,00000001b
    mov    pbc,a
    mov    pbc_bk,a
    mov    a,00000001b
    mov    pc,a
    clr    pcc
    clr    pd
    clr    pdc

    mov    a,00000101b
    mov    intc0,a
    mov    a,94
    mov    tmr,a
    mov    a,10010110b
    mov    tmrc,a
;*****
;Main
;*****
main_loop:
    clr    wdt
    mov    a,minute
    add    a,bcd_table
    mov    tblp,a
    tabrdl number2
    mov    a,number2
    and    a,0f0h
    swap  acc
    mov    number0,a
    mov    a,number2
    and    a,0fh
    mov    number1,a

    mov    a,number0
    
```

```

        mov     display_temp0,a
        mov     a,number1
        mov     display_temp1,a
        mov     a,number2
        mov     display_temp2,a
        jmp     main_loop
;*****
;Interrupt
;*****
timer_int:                                ;2592us
        push

;-----
;COM Operation Mode
;-----
com_pulse:
        inc     pulse_count
        mov     a,pulse_count
        sub     a,7
        snz     c
        jmp     $+3
        mov     a,1
        mov     pulse_count,a
        mov     a,pulse_count
        sdz     acc
        jmp     $+2
        jmp     com1_out_high
        sdz     acc
        jmp     $+2
        jmp     com1_out_low
        sdz     acc
        jmp     $+2
        jmp     com2_out_high
        sdz     acc
        jmp     $+2
        jmp     com2_out_low
        sdz     acc
        jmp     com0_out_low

com0_out_high:
        clr     com0_ctrl
        set     com0
        set     com1_ctrl
        set     com2_ctrl
        jmp     segment_pulse
    
```

```

com0_out_low:
    clr    com0_ctrl
    clr    com0
    set    com1_ctrl
    set    com2_ctrl
    jmp    segment_pulse
com1_out_high:
    set    com0_ctrl
    clr    com1_ctrl
    set    com1
    set    com2_ctrl
    jmp    segment_pulse
com1_out_low:
    set    com0_ctrl
    clr    com1_ctrl
    clr    com1
    set    com2_ctrl
    jmp    segment_pulse
com2_out_high:
    set    com0_ctrl
    set    com1_ctrl
    clr    com2_ctrl
    set    com2
    jmp    segment_pulse
com2_out_low:
    set    com0_ctrl
    set    com1_ctrl
    clr    com2_ctrl
    clr    com2

;-----
;SEGMENT Operation Mode
;-----
segment_pulse:
;-----
segment0_out:
    mov    a,display_temp0
    add    a,number_table
    mov    tblp,a
    tabrdl display_temp
    rl    display_temp
    rl    display_temp
    rl    display_temp
    mov    a,display_temp
    and    a,00000111b
    
```

```

inc    acc
sdz    acc
jmp    $+2
jmp    segment0_000
sdz    acc
jmp    $+2
jmp    segment0_001
sdz    acc
jmp    $+2
jmp    segment0_010
sdz    acc
jmp    $+2
jmp    segment0_011
sdz    acc
jmp    $+2
jmp    segment0_100
sdz    acc
jmp    $+2
jmp    segment0_101
sdz    acc
jmp    segment0_111
jmp    segment0_110

segment0_000:
    mov    a,com_000_table
    jmp    segment0_next
segment0_001:
    mov    a,com_001_table
    jmp    segment0_next
segment0_010:
    mov    a,com_010_table
    jmp    segment0_next
segment0_011:
    mov    a,com_011_table
    jmp    segment0_next
segment0_100:
    mov    a,com_100_table
    jmp    segment0_next
segment0_101:
    mov    a,com_101_table
    jmp    segment0_next
segment0_110:
    mov    a,com_110_table
    jmp    segment0_next
    
```

```

segment0_111:
    mov    a,com_111_table

segment0_next:
    add    a,pulse_count
    mov    tblp,a
    tabrdl acc
    clr    segment0
    sz     acc
    set    segment0
;-----
segment1_out:
    rl     display_temp
    rl     display_temp
    rl     display_temp
    mov    a,display_temp
    and    a,00000111b
    inc    acc
    sdz    acc
    jmp    $+2
    jmp    segment1_000
    sdz    acc
    jmp    $+2
    jmp    segment1_001
    sdz    acc
    jmp    $+2
    jmp    segment1_010
    sdz    acc
    jmp    $+2
    jmp    segment1_011
    sdz    acc
    jmp    $+2
    jmp    segment1_100
    sdz    acc
    jmp    $+2
    jmp    segment1_101
    sdz    acc
    jmp    segment1_111
    jmp    segment1_110

segment1_000:
    mov    a,com_000_table
    jmp    segment1_next
segment1_001:
    
```

```

        mov     a,com_001_table
        jmp     segment1_next
segment1_010:
        mov     a,com_010_table
        jmp     segment1_next
segment1_011:
        mov     a,com_011_table
        jmp     segment1_next
segment1_100:
        mov     a,com_100_table
        jmp     segment1_next
segment1_101:
        mov     a,com_101_table
        jmp     segment1_next
segment1_110:
        mov     a,com_110_table
        jmp     segment1_next
segment1_111:
        mov     a,com_111_table

segment1_next:
        add     a,pulse_count
        mov     tblp,a
        tabrdl  acc
        clr     segment1
        sz     acc
        set     segment1
;-----
segment2_out:
        clr     acc
        sz     display_temp.7
        set     acc.0
        inc     acc
        sdz    acc
        jmp     $+2
        jmp     segment2_000
        sdz    acc
        jmp     $+2
        jmp     segment2_001
        sdz    acc
        jmp     $+2
        jmp     segment2_010
        sdz    acc
        jmp     $+2
    
```

```

        jmp     segment2_011
        sdz    acc
        jmp    $+2
        jmp    segment2_100
        sdz    acc
        jmp    $+2
        jmp    segment2_101
        sdz    acc
        jmp    segment2_111
        jmp    segment2_110

segment2_000:
        mov    a,com_000_table
        jmp    segment2_next
segment2_001:
        mov    a,com_001_table
        jmp    segment2_next
segment2_010:
        mov    a,com_010_table
        jmp    segment2_next
segment2_011:
        mov    a,com_011_table
        jmp    segment2_next
segment2_100:
        mov    a,com_100_table
        jmp    segment2_next
segment2_101:
        mov    a,com_101_table
        jmp    segment2_next
segment2_110:
        mov    a,com_110_table
        jmp    segment2_next
segment2_111:
        mov    a,com_111_table

segment2_next:
        add    a,pulse_count
        mov    tblp,a
        tabrdl acc
        clr    segment2
        sz     acc
        set    segment2
;-----
segment3_out:
    
```

```

mov    a,display_temp1
add    a,number_table
mov    tblp,a
tabrdl display_temp
rl     display_temp
rl     display_temp
rl     display_temp
mov    a,display_temp
and    a,00000111b
inc    acc
sdz    acc
jmp    $+2
jmp    segment3_000
sdz    acc
jmp    $+2
jmp    segment3_001
sdz    acc
jmp    $+2
jmp    segment3_010
sdz    acc
jmp    $+2
jmp    segment3_011
sdz    acc
jmp    $+2
jmp    segment3_100
sdz    acc
jmp    $+2
jmp    segment3_101
sdz    acc
jmp    segment3_111
jmp    segment3_110

segment3_000:
    mov    a,com_000_table
    jmp    segment3_next
segment3_001:
    mov    a,com_001_table
    jmp    segment3_next
segment3_010:
    mov    a,com_010_table
    jmp    segment3_next
segment3_011:
    mov    a,com_011_table
    jmp    segment3_next
    
```

```

segment3_100:
    mov    a,com_100_table
    jmp    segment3_next
segment3_101:
    mov    a,com_101_table
    jmp    segment3_next
segment3_110:
    mov    a,com_110_table
    jmp    segment3_next
segment3_111:
    mov    a,com_111_table
    
```

```

segment3_next:
    add    a,pulse_count
    mov    tblp,a
    tabrdl acc
    clr    segment3
    sz     acc
    set    segment3
    
```

```

;-----
segment4_out:
    rl     display_temp
    rl     display_temp
    rl     display_temp
    mov    a,display_temp
    and    a,00000111b
    inc    acc
    sdz    acc
    jmp    $+2
    jmp    segment4_000
    sdz    acc
    jmp    $+2
    jmp    segment4_001
    sdz    acc
    jmp    $+2
    jmp    segment4_010
    sdz    acc
    jmp    $+2
    jmp    segment4_011
    sdz    acc
    jmp    $+2
    jmp    segment4_100
    sdz    acc
    jmp    $+2
    
```

```

        jmp     segment4_101
        sdz     acc
        jmp     segment4_111
        jmp     segment4_110

segment4_000:
        mov     a,com_000_table
        jmp     segment4_next
segment4_001:
        mov     a,com_001_table
        jmp     segment4_next
segment4_010:
        mov     a,com_010_table
        jmp     segment4_next
segment4_011:
        mov     a,com_011_table
        jmp     segment4_next
segment4_100:
        mov     a,com_100_table
        jmp     segment4_next
segment4_101:
        mov     a,com_101_table
        jmp     segment4_next
segment4_110:
        mov     a,com_110_table
        jmp     segment4_next
segment4_111:
        mov     a,com_111_table

segment4_next:
        add     a,pulse_count
        mov     tblp,a
        tabrdl  acc
        clr     segment4
        sz     acc
        set     segment4
;-----
segment5_out:
        clr     acc
        sz     display_temp.7
        set     acc.0
        inc     acc
        sdz     acc
        jmp     $+2
    
```

```

        jmp     segment5_000
        sdz     acc
        jmp     $+2
        jmp     segment5_001
        sdz     acc
        jmp     $+2
        jmp     segment5_010
        sdz     acc
        jmp     $+2
        jmp     segment5_011
        sdz     acc
        jmp     $+2
        jmp     segment5_100
        sdz     acc
        jmp     $+2
        jmp     segment5_101
        sdz     acc
        jmp     segment5_111
        jmp     segment5_110

segment5_000:
        mov     a,com_000_table
        jmp     segment5_next
segment5_001:
        mov     a,com_001_table
        jmp     segment5_next
segment5_010:
        mov     a,com_010_table
        jmp     segment5_next
segment5_011:
        mov     a,com_011_table
        jmp     segment5_next
segment5_100:
        mov     a,com_100_table
        jmp     segment5_next
segment5_101:
        mov     a,com_101_table
        jmp     segment5_next
segment5_110:
        mov     a,com_110_table
        jmp     segment5_next
segment5_111:
        mov     a,com_111_table
    
```

```

segment5_next:
    add    a,pulse_count
    mov    tblp,a
    tabrdl acc
    clr    segment5
    sz     acc
    set    segment5

;-----
;Output COM&SEGMENT
;-----
output_com_segment:
    mov    a,pb_bk
    mov    pb,a
    mov    a,pbc_bk
    mov    pbc,a
    mov    a,pa_bk
    mov    pa,a

;-----
;Time
;-----
    inc    msecond
    mov    a,msecond
    sub    a,192
    snz    c
    jmp    timer_end
    clr    msecond

    inc    second
    mov    a,second
    sub    a,120
    snz    c
    jmp    timer_end
    clr    second

    inc    minute
    mov    a,minute
    sub    a,60
    snz    c
    jmp    timer_end
    clr    minute

timer_end:
    pop
    reti
    
```

```

;*****
;Table
;*****
        org     0700h
number_table:
        ;      gbefadc0
        dw     11111010b      ;"0" 0
        dw     00010010b      ;"1" 1
        dw     10101110b      ;"2" 2
        dw     10011110b      ;"3" 3
        dw     01010110b      ;"4" 4
        dw     11011100b      ;"5" 5
        dw     11111100b      ;"6" 6
        dw     00011010b      ;"7" 7
        dw     11111110b      ;"8" 8
        dw     11011110b      ;"9" 9
        dw     00000100b      ;"- " a
        dw     00000000b      ;" " b
;-----
com_000_table:
        dw     0
        dw     1,0,1,0,1,0
com_001_table:
        dw     0
        dw     1,0,1,0,0,1
com_010_table:
        dw     0
        dw     0,1,1,0,1,0
com_011_table:
        dw     0
        dw     0,1,1,0,0,1
com_100_table:
        dw     0
        dw     1,0,0,1,1,0
com_101_table:
        dw     0
        dw     1,0,0,1,0,1
com_110_table:
        dw     0
        dw     0,1,0,1,1,0
com_111_table:
        dw     0
        dw     0,1,0,1,0,1
;-----
    
```

```
bcd_table:
    dw    00h, 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h
    dw    10h, 11h, 12h, 13h, 14h, 15h, 16h, 17h, 18h, 19h
    dw    20h, 21h, 22h, 23h, 24h, 25h, 26h, 27h, 28h, 29h
    dw    30h, 31h, 32h, 33h, 34h, 35h, 36h, 37h, 38h, 39h
    dw    40h, 41h, 42h, 43h, 44h, 45h, 46h, 47h, 48h, 49h
    dw    50h, 51h, 52h, 53h, 54h, 55h, 56h, 57h, 58h, 59h
    dw    60h, 61h, 62h, 63h, 64h, 65h, 66h, 67h, 68h, 69h
    dw    70h, 71h, 72h, 73h, 74h, 75h, 76h, 77h, 78h, 79h
    dw    80h, 81h, 82h, 83h, 84h, 85h, 86h, 87h, 88h, 89h
    dw    90h, 91h, 92h, 93h, 94h, 95h, 96h, 97h, 98h, 99h

    end

;*****
```