

HT47R20A-1 单片机 I/O 口的使用介绍

文件编码：HA0033s

介绍

HT47R20A-1 有一个 8 位双向输入/输出口 PA、一个 4 位的输入口 PB，分别对应 RAM 中的 [12H] 和 [14H]。PA 的高四位引脚是带有上拉电阻的 NMOS 输入/输出，PA 的低四位引脚可由掩膜选择设定为 NMOS 输入/输出或 CMOS 输出，并且 PA 的低位引脚可由掩膜选择设定为有或无上拉电阻，PA 的每个引脚都可以具有唤醒功能（由掩膜选择设定）。PB 只能用于输入，且每个引脚都可由掩膜选择设定有或无上拉电阻。就 PA、PB 口作为输入而言，不具有锁存功能，就 PA 口作为输出而言，所有数据被锁存住，而且不受任何影响，直到输出锁存被写入新的值为止。

PA 口某个引脚在当成输入口使用时必须先向对应位写“1”，以关闭 NMOS；也就是先执行“SET [m].i”指令关闭 NMOS，然后执行“MOV A,[m]”指令来读取寄存器的数据。并且这两条指令必须紧接在一起，否则会因为“Read—Modify—Write”的过程而改变引脚的值。

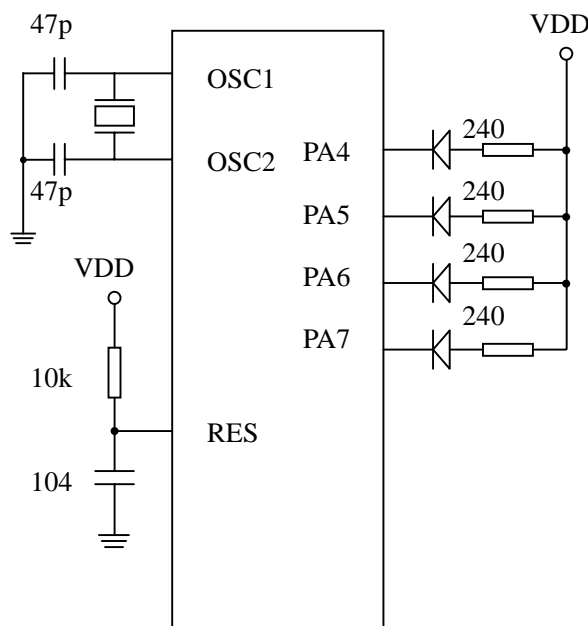
有些指令在对端口进行输出操作时，会先从端口读入数据，经修改后把结果输出到端口，即我们所说的“Read—Modify—Write”的过程。举例来说：SET [m].i，CLR [m].i，CPL [m]和 CPLA [m]这些指令会先将整个端口状态读入 CPU 中，接着执行所定义的运算，最后再将执行的结果写入锁存或是累加器中。所以在对端口执行此类指令时，可能会改变内部锁存器的值，需要加以注意。

I/O 口的使用

下面以流水灯来说明 HT47R20A-1 的 I/O 口使用。

硬件部分：

PA4~PA7 引脚分别接发光二极管 LED，如图：



这里需要注意的是：PA 口的高位引脚是带有上拉电阻的 NMOS 输入/输出，作为输出使用时，其驱动能力较低；因此接二极管时应将二极管负极接到 PA 口的高位，二极管正极接一上拉电阻后接到 VDD。

软件部分：

下面的程序利用定时/计数器中断来控制 LED 循环发光。定时/计数器时钟来源为系统时钟，计数初值为 00H。

程序清单：

```
include ht47r20a-1.inc
    data .section 'data'
        count db ?
    code .section at 0 'code'
        org    00h
        jmp    start
        org    04h
        reti
        org    08h
        reti
        org    0ch
        reti
        org    10h
        jmp    timer_int          ; 定时/计数器中断入口地址
; -----
start:
    clr    intc0
    clr    intc1
    clr    adcr.1                ; 定时/计数器允许
    mov    a,08h                 ; 设置定时/计数器时钟来源为系统时钟
    mov    tmrc,a
    mov    a,0eeh
    mov    count,a              ; count 用来控制 LED 循环发光
    mov    a,00h                ; 设置定时/计数器初值
    mov    tmrb1,a
    mov    tmrbh,a
    mov    tmral,a
    mov    tmrah,a
    set    tmrc.4                ; 打开定时/计数器
    set    intc1.0              ; 定时/计数器中断允许
    set    intc0.0              ; 总中断允许
    jmp    $
; -----
timer_int:                       ; 定时/计数器中断服务子程序
```

```

rr    count
mov   a, count    ; 点亮下一个引脚的 LED
mov   pa, a
reti

```

I/O 口的 “ Read—Modify—Write ” 现象说明

以下程序说明 “ Read—Modify—Write ” 的过程。我们将 PA.0 和 PA.7 用导线接在一起。

程序清单：

```

include ht47r20a-1.inc
code .section at 0 'code'
    org 00h
    jmp start
    org 04h
    reti
    org 08h
    reti
    org 0ch
    reti
    org 10h
    reti
; -----
start:                                ; PA.7 (锁存器值) 引脚值  PA.0 (锁存器值) 引脚值
    clr pa                            ; (0)0                (0)0
    set pa.7 ;(1) ;                    (1)0                (0)0
    set pa.0 ;(2) ;                    (0)0                (1)0
    ; set pa.7 ;(3)
    sz pa.7
    jmp read_1
read_0:
    jmp $
read_1:
    jmp $

```

程序说明：

执行以上程序，我们会发现 PA.0 和 PA.7 引脚的值始终为 0，但实际上 PA.0 和 PA.7 内部锁存器的值是变化的，参见程序注释。

如果用单步调试，我们就能观察到 “ Read—Modify—Write ” 的过程。当程序执行完(2)时，将连接 PA.0 和 PA.7 的导线断开，则执行完下一条语句后 PA.0 引脚的值会变为 1。

加上(3)语句，如果在执行完(2)时断开连接，则最后结果为 PA.0=1、PA.7=1；如果在执行完(3)时断开连接，则最后结果为 PA.0=0、Pa.7=1。

在使用过程中应注意以上区别。