

HT48 & HT46 MCU I/O 口的应用-流水灯

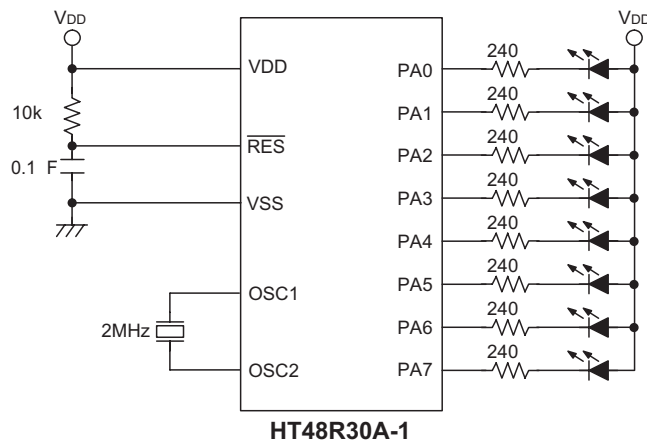
文件编码: HA0009s

介绍:

本例以 HT48R30A-1 为例, 模拟流水灯的功能, 用 LED 显示输出。PA 口每一位输出接一个 LED, 串联一个 240Ω 电阻到电源, 即可完成初步电路。通过这个例子, 用户可以了解到 HT48XX 单片机输入输出口的基本用法。

电路设计:

PA0~PA7 用作输出, 每一位连接一个电阻来控制一个 LED。通过左移 (RLC)、右移 (RRC) 指令来控制点亮的 LED 左右移动。电路图如下:



程序:

```
#include      ht48R30A-1.inc
;-----
data  .section  'data'      ;定义数据寄存器
count1 db  ?                ;延时计数器 0
count2 db  ?                ;延时计数器 1
lamp   db  ?                ;亮灯状态记录器
;-----
code  .section  at 0 'code';程序段
      org      00h
      jmp      start        ;主程序开始
start:
      mov     a,0           ; (1)
      mov     pac,a         ; PA 口设为输出
      cpl    acc           ; ACC 取反 (为 0FFH)
      mov     pa,a         ; 设定 PA 口初始状态
      mov     a,1           ; (2)
```

```

mov    lamp, a           ;亮灯状态初值送到亮灯状态记录器
llamp:                          ;亮灯左移的循环
mov    a, lamp           ;累加器的值送到亮灯状态记录器
cpl    acc                ;累加器值取反
mov    pa, a             ;点亮与亮灯状态记录器相对应的灯
call   delay ; (3)      ;调用延时子程序
clr    c                  ;清除进位标志
rlc    lamp ; (4)       ;亮灯状态记录器值左移一位
sz     lamp               ;亮灯状态记录器是否为 0
jmp    llamp ; (5)       ;如果不是则继续循环
rrc    lamp ; (6)       ;如果是, 亮灯状态记录器右移一位
rlamp:                          ;亮灯右移的循环
mov    a, lamp           ;将亮灯状态记录器值送到累加器
cpl    acc                ;累加器取反
mov    pa, a             ;点亮与亮灯状态记录器对应的灯
call   delay             ;调用延时子程序
clr    c                  ;清除进位标志
rrc    lamp ; (7)       ;亮灯状态记录器值右移一位
sz     lamp               ;判断亮灯状态记录器的值是否为 0
jmp    rlamp              ;如果不是则继续循环
rlc    lamp               ;将亮灯状态记录器的值右移
jmp    llamp ; (8)       ;跳到左移循环
delay proc                       ;延时子程序
mov    a, 2fh
mov    count1, a          ;定义延时计数器 1 的初值
mov    count2, a          ;定义延时计数器 2 的初值
d1:
sdz    count2             ;延时计数器 2 递减, 判断是否为 0
jmp    d1                  ;如果不是, 跳到 D1 循环
sdz    count1             ;延时计数器 1 递减, 判断是否为 0
jmp    d1                  ;如果不是, 跳到 D1 循环
ret
delay endp                    ;延时子程序结束
end                             ;主程序结束

```

程序说明:

这段小程序能够使八个 LED 依次点亮, 先从左向右, 然后从右向左。程序开始时先设定输入输出口的状态(1), 在这里设定所有的口为输出, 将 PA 口的控制寄存器 PAC 设为 0。然后设定输出的初始状态, PA 口的第一位设为 0, 点亮第一个 LED(2)。为了能观察到 LED 的闪亮, (3) 设定了延时子程序。通过亮灯状态记录器和累加器, 点亮的 LED 由左移指令控制左移(4)。为了判断左移时所有的 LED 是否都依次点亮过, 可以测试亮灯状态记录器是否为 0(5), 如果是则恢复亮灯状态记录器左移时的终值(6), 程序向下执行, 控制 LED 右移。这次用右移指令 RRC 来控制(7), 重复类似的过程。最后同样判断亮灯状态记录器是否为 0, 如果是, 则表明右移时所有的 LED 都已经点亮过, 程序跳到开始处, 重新开始循环(8)。