

## 164 驱动范例程序

```
*****
;164 驱动程序
*****
CLK_164      EQU      PORTA, 0      ;164 时钟口
DAT_164      EQU      PORTA, 1      ;164 数据口
CLK_IO       EQU      TRISA, 0      ;时钟口方向
DAT_IO       EQU      TRISA, 1      ;数据口方向

KCOUNT      EQU      ?            ;扫键个数(示例程序扫描 8 个按键)
KDATA        EQU      ?            ;读取到的按键信息
DTEMP        EQU      ?            ;扫按键时传送给 164 的数据

TEMP         EQU      ?            ;临时缓存
COUNT       EQU      ?

*****
;164 移位写数据子程序
*****
MOVE_164:
    LD        TEMP, A
    LDIA     08H
    LD        COUNT, A
MOVE_LOOP:
    CLRB     CLK_164
    CLRB     DAT_164
    RLCR     TEMP
```

```

SZB          STATUS, C
SETB         DAT_164
SETB         CLK_164
SZDECR      COUNT
JP           MOVE_LOOP
CLRB         CLK_164
CLRB         DAT_164
RET          00H

```

```

;*****

```

```

;164 读键值示例子程序

```

```

;164 芯片 D0-D7 口上的按键分别对应 KDATA 寄存器 b0-b7 位,

```

```

;若口线按键按下, KDATA 相应位置 1, 否则置 0

```

```

;*****

```

```

KSCAN_164:

```

```

    ;必须先关闭所有显示 COM 口

```

```

    CLR          KCOUNT

```

```

    CLR          KDATA

```

```

    LDIA         B' 11111110'          ;从 164 芯片 D0 引脚按键开始扫

```

```

描

```

```

    LD          DTEMP, A

```

```

KSCAN_LOOP:

```

```

    CLRB         DAT_IO                ;164 数据口作输出口传输数据

```

```

    CALL         DELAY50US             ;改变口线方向必要的延时

```

```

    LD          A, DTEMP

```

```

    CALL         MOVE_164

```

```

    SETB         DAT_IO                ;164 数据口作输入口检测按键

```

```

    CALL         DELAY50US             ;改变口线方向必要的延时

```

```

CLR B      STATUS, C
SNZ B      DAT_164
SET B      STATUS, C
RRC R      KDATA          ;保存按键数据

INCR       KCOUNT
LDIA       .8
SUBA       KCOUNT
SZB        STATUS, Z
JP         KSCAN_BACK

SET B      STATUS, C
RLCR       DTEMP
JP         KSCAN_LOOP

KSCAN_BACK:
CLR B      DAT_IO          ;扫描完成后将数据口作输出口
RET

;*****
;8M 下 50us 非精准延时子程序
;*****

DELAY50US:
LDIA       .32
LD         TEMP, A
SZDECR     TEMP
JP         $-1
RET

;*****

```