

1637 驱动范例程序

```
*****  
*****  
;1637 驱动程序包  
*****  
CLK_1637      EQU          PORTA, 0    ;1637 时钟口  
DAT_1637     EQU          PORTA, 1    ;1637 数据口  
DAT_IO       EQU          TRISA, 1    ;1637 数据口方向  
  
R01          EQU          ?          ;显示 RAM  
R02          EQU          ?  
R03          EQU          ?  
R04          EQU          ?  
R05          EQU          ?  
R06          EQU          ?  
KDATA        EQU          ?          ;读取到的 1637 键值  
  
TEMP         EQU          ?          ;临时寄存器  
COUNT       EQU          ?  
DCOUNT       EQU          ?  
WR_DELY      EQU          ?  
*****  
;1637 扫键程序  
;1637 扫键结果存放到 KDATA 寄存器  
*****  
KSCAN_DATA:
```

```
CALL      START_1637
LDIA     B' 01000010'      ;送读键值指令
CALL     WR_1637
CALL     ACK_1637      ;应答信号

CALL     RD_1637      ;读取数据
CALL     ACK_1637
CALL     STOP_1637      ;结束条件
RET

;-----
RD_1637:
    SETB     DAT_IO      ;数据口为输入, 读取数据

    LDIA     .20
    CALL     DELAY

    CLR      KDATA
    CLRB     CLK_1637
    LDIA     08H
    LD      COUNT, A

RD_LOOP:
    CLRB     CLK_1637
    NOP
    NOP
    CLRB     STATUS, C
    SZB     DAT_1637
    SETB     STATUS, C

    RRCR     KDATA      ;读取键值到 KDATA
```

```

SETB     CLK_1637
SZDECR   COUNT
JP       RD_LOOP

CLR      DAT_IO      ;按键检测完后数据口置输出
RET

```

```

;*****

```

```

;1637 显示子程序, 地址自动增加模式

```

```

;*****

```

```

DISP_1637:

```

```

    CALL     START_1637    ;开始条件
    LDIA     B' 01000000'   ;送写控制命令, 地址自加模式

```

```

    CALL     WR_1637       ;写数据到 1637

```

```

    CALL     ACK_1637      ;ACK

```

```

    CALL     STOP_1637     ;结束条件

```

```

    CALL     START_1637    ;开始

```

```

    LDIA     B' 11000000'   ;地址命令设置, 自加 1 模式

```

```

    CALL     WR_1637       ;1637 写数据

```

```

    CALL     ACK_1637      ;应答信号 ACK

```

```

    CLR      DCOUNT

```

```

DISP_1637_LOOP:

```

```

    LDIA     07H

```

```
ANDA    DCOUNT
ADDIA   TAB_DISP_DATA$L
LD      TABLE_SPL, A
LDIA    TAB_DISP_DATA$H
LD      TABLE_SPH, A
SZB     STATUS, C
INCR    TABLE_SPH
TABLEA          ;获得数据地址

LD      FSR, A
LD      A, INDF    ;1-真-灯亮、0-假-灯灭
CALL    WR_1637
CALL    ACK_1637

INCR    DCOUNT    ;总共要显示 6 个 COM
LDIA    .6
SUBA    DCOUNT
SNZB    STATUS, C
JP      DISP_1637_LOOP
CLR     DCOUNT

CALL    STOP_1637    ;结束

CALL    START_1637
LDIA    B'10001001'    ;显示控制命令
CALL    WR_1637
CALL    ACK_1637    ;应答
CALL    STOP_1637    ;结束通信
RET
```

```
;-----  
TAB_DISP_DATA:  
    DW      R01  
    DW      R02  
    DW      R03  
    DW      R04  
    DW      R05  
    DW      R06  
    DW      TEMP  
    DW      TEMP  
  
;*****  
;数据输入的开始条件是 CLK 为高电平时, DIO 由高变低  
;结束条件是 CLK 为高时, DIO 由低电平变为高电平  
;*****  
START_1637:  
    SETB     CLK_1637  
    SETB     DAT_1637  
    LDIA     .2  
    CALL     DELAY  
    CLRB     DAT_1637  
    RET  
  
;-----  
STOP_1637:  
    CLRB     DAT_1637  
    LDIA     .4  
    CALL     DELAY  
    SETB     CLK_1637  
    LDIA     .4
```

```
CALL      DELAY
SETB     DAT_1637
RET
```

```
;
```

```
ACK_1637:
```

```
CLRB     CLK_1637
LDIA     .2
CALL     DELAY
CLRB     DAT_1637
LDIA     .10
CALL     DELAY
SETB     CLK_1637
LDIA     .2
CALL     DELAY
CLRB     CLK_1637
LDIA     .1
CALL     DELAY
SETB     DAT_1637
RET
```

```
;
```

```
WR_1637:
```

```
LD      TEMP, A
LDIA    08H
LD      COUNT, A
```

```
WR_LOOP:
```

```
LDIA    .3
CALL    DELAY
CLRB    CLK_1637
```

被传输

;DAT 数据在 CLK 低时变化, 在 CLK 高时

```
RRCR      TEMP
SNZB      STATUS, C
CLRB      DAT_1637
SZB       STATUS, C
SETB      DAT_1637
LDIA     .3
CALL     DELAY
SETB      CLK_1637      ;CLK 上升沿传输数据
SZDECR   COUNT
JP       WR_LOOP
RET

;-----
DELAY:      ;延时
LD       WR_DELY, A
SZDECR   WR_DELY
JP       $-1
RET

;*****
```