夏普 LQ104V1DG52 液晶屏驱动程序

液晶控制器：CX135，显示 256 色，联系电话：0755-81261729

89608258 QQ:191981784

```
FX      EQU     7FH             ;text x dot size FX+1
FY      EQU     7EH             ;text y dot size FY+1
CR      EQU     7DH             ;character return APL-1
TC_P    EQU     7CH             ;fram purse     12MHz>=[(TC_P)*9+1]*L_F*70Hz
L_F     EQU     7BH             ;y duty L_F+1
APL     EQU     7AH             ;number of text per line
APH         EQU         79H
LCDIR   EQU     101H        ;指令寄存器地址
LCDDR   EQU     0000H       ;数据寄存器地址
RAM0    EQU     20H
RAM1    EQU     21H
RAM2    EQU     22H
COUNT1  EQU     23H
START:  MOV     FX,#07H         ;can change     7
        MOV     FY,#01H         ;graphic 00
        MOV     CR,#79          ;640/8 - 1
        MOV     TC_P,#44        ; don't care        ;44
        MOV     L_F,#59         ; 480/8 - 1     ;59
        MOV     APL,#080H       ;设置列数 80H=10000000
        MOV     APH,#02H    ;02H=10,组成 1010000000=640 行
        CALL    RESET_P11
        CALL    DELAY2
        CALL    SET_P11
        CALL    DELAY2
        NOP
;*********************************************************************************
SYSTEM_SET:
        MOV     A,#40H
        CALL    COM_W
        MOV     A,#30H
        CALL    DATA_W
        MOV     A,FX            ;FX
        CALL    DATA_W
        MOV     A,FY            ;FY
        CALL    DATA_W
        MOV     A,CR            ;CR
        CALL    DATA_W
        MOV     A,TC_P          ;TC/R
```

```
        CALL    DATA_W
        MOV     A,L_F                   ;L/F
        CALL    DATA_W
        MOV     A,APL                   ;APL
        CALL    DATA_W
        MOV     A,APH                   ;APH
        CALL    DATA_W
SYSTEM_SET2:            ; COMMAND 80H
        MOV     A,#80H
        CALL    COM_W
        MOV     A,#0FH          ; TFT mode
        CALL    DATA_W
SCROLL:
        MOV     A,#44H
        CALL    COM_W
        MOV     A,#00H          ;SAD1L
        CALL    DATA_W
        MOV     A,#00H          ;SAD1H
        CALL    DATA_W
        MOV     A,#0EFH         ;           EF
        CALL    DATA_W
CSR_FORM:
        MOV     A,#5DH
        CALL    COM_W
        MOV     A,#00H          ;can change, must <= fx
        CALL    DATA_W
        MOV     A,#80H          ;graphic 80H
        CALL    DATA_W
HDOT_SCR:
        MOV     A,#5AH
        CALL    COM_W
        MOV     A,#00H                  ;NO SCROLL
        CALL    DATA_W
OVERLAY:
        MOV     A,#5BH
        CALL    COM_W
        MOV     A,#0CH                  ;layer1 only
        CALL    DATA_W
CSR_DIR:
        MOV     A,#4CH
        CALL    COM_W
        CALL    CL1
DISP_ON:
        MOV     A,#59H
```

```
            CALL       COM_W
            MOV         A,#55H
            CALL       DATA_W
  waitl:   LCALL    BOX
            LCALL    DELAY
            CALL       SHOW_WS1
            CALL     DELAY
            JMP waitl
SHOW_WS1:
                  CALL   CL1         ;清屏
                    CALL       POS1
                    MOV         A,#42H
                    CALL       COM_W
                 MOV    R5,#240                    ;240
LL1:       MOV R6,#160                  ;192 ;160
L1L:              MOV    R4,#08H                      ;08
LLL1:             MOV      A,#0C0H                  ; 0E0=11100000
                   CALL       DATA_W
                   DJNZ    R4,LLL1
                    DJNZ  R6,L1L
                     DJNZ       R5,LL1
                    LCALL    DELAY
                     CALL       POS1
                      MOV         A,#42H
                      CALL       COM_W
                     MOV    R5,#240
LL2:       MOV      R6,#160
L2L:              MOV    R4,#08H
LLL2:               MOV  A,#38H                      ; 1C=11100
                   CALL       DATA_W
                  DJNZ      R4,LLL2
                  DJNZ      R6,L2L
                    DJNZ  R5,LL2
                    LCALL    DELAY
                    CALL       POS1
                    MOV         A,#42H
                    CALL       COM_W
                  MOV    R5,#0F0H
LL3:       MOV      R6,#160
L3L:              MOV    R4,#08H
LLL3:                MOV      A,#07H                        ; 03=11
                     CALL       DATA_W
                     DJNZ      R4,LLL3
                     DJNZ      R6,L3L
```

```
                DJNZ    R5,LL3
                LCALL   DELAY
                CALL    POS1
                MOV     A,#42H
                CALL    COM_W
               MOV   R5,#0F0H
LL4:      MOV   R6,#160
L4L:          MOV   R4,#08H
LLL4:          MOV  A,#00H                    ; 1C=11100
               CALL    DATA_W
               DJNZ  R4,LLL4
               DJNZ  R6,L4L
               DJNZ  R5,LL4
               LCALL   DELAY
               CALL    POS1
               MOV     A,#42H
               CALL    COM_W
               MOV   R5,#0F0H
LL5:       MOV  R6,#160
L5L:           MOV   R4,#08H
LLL5:           MOV    A,#0FFH                    ; 03=11
                CALL    DATA_W
                DJNZ    R4,LLL5
                DJNZ    R6,L5L
                DJNZ    R5,LL5
               RET
;*********************************************************************************
COM_W:
        MOV DPTR,   #0100H          ;写指令
        MOVX    @DPTR, A
        NOP
        RET
PR1:PUSH    DPH
    PUSH    DPL
    MOV     DPTR,#LCDDR
    MOV     A,RAM0
    MOVX    @DPTR,A
    POP DPL
    POP DPH
    RET
;*********************************************************************************
DATA_W:
         MOV    DPTR,   #0000H    ;写数据
         MOVX   @DPTR, A
```

```
        NOP
        RET
DATA_R:
    MOV     DPTR, #0100H                ;读数据
    MOVX    A,   @DPTR
    NOP
    RET
FLAG_R:
    MOV     DPTR, #0000H            ;判忙
    MOVX    A,   @DPTR
    NOP
    RET
SET_P11:
    MOV     90H, #0FFH
    RET
RESET_P11:
    MOV     90H, #0FDH
    RET
;*********************************************************************************
;CLEAR LAYER 1
CL1:      CALL      POS1
          MOV       A,#42H     ;写指令代码
          CALL      COM_W      ;写指令
          MOV       R0,#240
L1:       MOV       R1,#160
l2:       MOV       R2,#08
L3:       MOV       A,#00H
          CALL      DATA_W
          DJNZ      R2,L3
          DJNZ      R1,L2
           DJNZ     R0,L1
          RET
DOTS:           CALL  CL1
          CALL      POS1
            LCALL     WRITEBMP
            LCALL     DELAY
            RET
WRITEBMP:
    MOV     RAM0,#46H
    LCALL   PUTIR
    MOV     RAM0,#0C8H
    LCALL   PR1
    MOV     RAM0,#00H
    LCALL PR1
```

```
        MOV     RAM0,#42H
        LCALL   PUTIR
        MOV       FX,#00H               ;can change          ;0F=15
        MOV    A,FX
        MOV       FY,#00H
        MOV    A,FY
        ;MOV    DPTR,#TAB
BMP1:   MOV     R1,#120
BMP2:   MOV     A,#00H
        MOVC    A,@A+DPTR
        MOV     RAM0,A
        LCALL PR1
        INC DPTR
        DJNZ    R1,BMP2
        RET
PUTIR:                  ;写指令代码
        PUSH    DPH     ;将直接地址的内容压入堆栈
        PUSH    DPL
        MOV     DPTR,#LCDIR
        MOV     A,RAM0
        MOVX @DPTR,A
        POP DPL
        POP DPH
        RET
BOX:    CALL    CL1
        CALL    POS1
        MOV     A,#42H
        CALL    COM_W
        MOV     A,#4CH              ;TEST 2
        CALL    COM_W
        MOV     A,#46H
        CALL    COM_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#42H
        CALL    COM_W
        MOV     R3,#80
BOX1:   MOV     A,#0FFH
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        MOV     A,#0FFH
```

```
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
         MOV    A,#0FFH
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        DJNZ    R3,BOX1
        MOV     R0,#2
BOXX:   MOV     R2,#0EfH        ;238
BOX3:   MOV     A,#0FFH
        CALL    DATA_W
        MOV     R4,#159
BOX2:   MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        DJNZ    R4,BOX2
         MOV    A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        DJNZ    R2,BOX3
        DJNZ    R0,BOXX
        MOV     A,#0FFH
        CALL    DATA_W
        MOV     R4,#159
BOXA:   MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
```

```
        CALL    DATA_W
        DJNZ    R4,BOXA
         MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        MOV     R5,#160
BOX4:   MOV     A,#0FFH
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        MOV     A,#0FFH
        CALL    DATA_W
        DJNZ    R5,BOX4
        RET
;*************************************************************************
POS1:
        MOV     A,#46H
        CALL    COM_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        MOV     A,#00H
        CALL    DATA_W
        RET
;*************************************************************************
DELAY:
    MOV     R7,#80
K:      MOV     R6,#100
K0:     MOV     R5,#250
K1:     DJNZ    R5,K1
        DJNZ    R6,K0
        DJNZ    R7,K
        RET
DELAY2:
         MOV     R5,#250
K12:     DJNZ    R5,K12
         RET
;*************************************************************************
```

END