

240320d1_fl series sample program

```

;*****
; * MCU : 8051 *
; * CONTROLLER : SED1335 *
; * REMARK (SEL1 = 1, SEL2 = 0), TIMING AS 6800 SERIES *
; * LANDSCAPE MODE, 320x240 *
;*****

```

```

RES EQU 80H
CS EQU 81H
E EQU 85H
R_W EQU 86H
AO EQU 87H

```

```
ORG 00H
```

```

CHIP_SELECT:
CLR RES
NOP
NOP
SETB RES
NOP
NOP

```

```

SYSTEM_SET:
MOV A,#040H ; SYSTEM SET
CALL TTR1 ; COMMAND INPUT

MOV A,#010H ; P1 (DR=0,T/L=0,IV=0,W/S=0,M2=0,M1=0,M0=0)
CALL TTR2
MOV A,#087H ; P2 (WF = 1, FX = 7)
CALL TTR2
MOV A,#07H ; P3 (FY = 7)
CALL TTR2
MOV A,#39 ;P4 (CR=40)
;mov a,#01dh ; p4 (cr=30-1)
;mov a,#32 ; p4 (cr=32)
;MOV A,#0FH ; P4 (CR = 15)
CALL TTR2
MOV A,#02FH ;P5 (TC/R=40, 4MHz)
;MOV A,#04AH ; P5 (TCR = 74, FOLLOW G121C SPEC.)
CALL TTR2
MOV A,#0EFH ;P6 (L/F = 239)
;mov a,#09fh ; p6 (1/f =159)
;mov a,#040h ; p6 (1/f = 64)
;MOV A,#07FH ; P6 (L/F = 127)
CALL TTR2
MOV A,#10H ; P7 (APL = 16)
CALL TTR2
MOV A,#00H ; P8 (APH = 0)
CALL TTR2

```

```

SCROLL:
MOV A,#044H ; SCROLL
CALL TTR1 ; COMMAND INPUT

MOV A,#00H ; P1 (SAD 1L)
CALL TTR2
MOV A,#00H ; P2 (SAD 1H)
CALL TTR2
MOV A,#07FH ; P3 (SL1)
CALL TTR2
MOV A,#00H ; P4 (SAD 2L)
CALL TTR2
MOV A,#010H ; P5 (SAD 2H)
CALL TTR2
MOV A,#040H ; P6 (SL2)
CALL TTR2
MOV A,#00H ; P7 (SAD 3L)
CALL TTR2
MOV A,#04H ; P8 (SAD 3H)
CALL TTR2
MOV A,#00H ; P9 (SAD 4L)
CALL TTR2
MOV A,#030H ; P10 (SAD 4H)
CALL TTR2

```

```

HDOT_SCR:
MOV A,#05AH ; HDOT SCR

```

```

                240320d1_fl series sample program
CALL    TTR1          ; COMMAND INPUT

MOV     A,#00H       ; P1 = 0
CALL    TTR2

OVLAY:
MOV     A,#05BH      ; OVLAY
CALL    TTR1          ; COMMAND INPUT
MOV     A,#00H       ; P1 (OV=0, DM2=0, DM1=0, MX1=0, MX0=0)
CALL    TTR2

CALL    DISP_OFF
CALL    DELAY

CALL    SLEEP         ; DISPLAY AND LCD CONTROLLER SLEEP
CALL    DELAY

CALL    CLR_L1        ; CLEAR THE 1ST LAYER RAM MEMORY
CALL    CLR_L2        ; CLEAR THE 2ND LAYER RAM MEMORY

CALL    CG_RAM        ; ALL CHARACTER SET 1
CALL    CSRW
CALL    CSR_DIR
CALL    CSR_FORM
CALL    DISP_ON
CALL    C_SET1
CALL    DELAY

CALL    CG_RAM        ; ALL ROW ON SCREEN
CALL    CSRW
CALL    CSR_DIR
CALL    DISP_ON
CALL    ROW
CALL    DELAY

CALL    CG_RAM        ; ALL COLUMN ON SCREEN
CALL    CSRW
CALL    CSR_DIR
CALL    DISP_ON
CALL    COLUMN
CALL    DELAY

CALL    CG_RAM        ; ALL SMALL CHECKER ON SCREEN
CALL    CSRW
CALL    CSR_DIR
CALL    DISP_ON
CALL    S_CHECKER
CALL    DELAY

CALL    CG_RAM        ; ALL BIG CHECKER ON SCREEN
CALL    CSRW
CALL    CSR_DIR
CALL    DISP_ON
CALL    B_CHECKER
CALL    DELAY

;CALL    CG_RAM        ; FOR MODULE ASSEMBLY ALIGNMENT
;CALL    CSRW
;CALL    CSR_DIR
;CALL    DISP_ON
;CALL    ALIGNMENT
;CALL    DELAY

```

```
JMP    CHIP_SELECT
```

```
.*****
```

```

C_SET1:
MOV     A,#042H      ; MWTRITE
CALL    TTR1          ; COMMAND INPUT
MOV     R7,#40        ;ROW CHARACTER
;mov    r7,#20      ;row_char
;MOV    R7,#16      ;ROW_CHAR
c_set21:MOV    R5,#30
;mov    r5,#32
MOV     A,#041H      ; start display characters from "A"

```

240320d1_fl series sample program

```

c_set11:CALL    TTR2
          inc    a
          djnz   r5,c_set11
          DJNZ   R7,C_SET2L      ; 8 row of character
          RET

;*****
DISP_ON:
          MOV    A,#059H          ; DISPLAY ON
          CALL   TTR1            ; COMMAND INPUT

          MOV    A,#07H          ; (FP5=0, FP4=0, FP3=0, FP2=0, FP1=0, FP0=1, FC1=1, FC0=1)
          CALL   TTR2
          RET

;*****
DISP_OFF:
          MOV    A,#058H          ; DISPLAY OFF
          CALL   TTR1            ; COMMAND INPUT

          MOV    A,#056H          ; P1 (FP5=0, FP4=1, FP3=0, FP2=1, FP1=0, FP0=0, FC1=1, FC0=0)
          CALL   TTR2
          RET

;*****
SLEEP:
          MOV    A,#53H
          CALL   TTR1

          CALL   DELAY

          MOV    A,#040H          ; SYSTEM SET (TURN ON THE DISPLAY)
          CALL   TTR1            ; COMMAND INPUT

          MOV    A,#030H          ; P1 (DR=0,T/L=0,IV=1,W/S=0,M2=0,M1=0,M0=0)
          CALL   TTR2
          RET

;*****
COLUMN:
          MOV    A,#042H          ; SHOW ALL COLUMN IN SCREEN
          CALL   TTR1            ; MWTRITE
          ;      ; COMMAND INPUT
          mov    r6,#16
          ;MOV    R6,#8          ; NO. OF ROW_CHAR
COL1:    ;MOV    R7,#32          ; NO. OF COL_CHAR
          mov    r7,#64
COL2:    MOV    A,#082H
          CALL   TTR2
          DJNZ   R7,COL2
          DJNZ   R6,COL1
          RET

;*****
ROW:
          MOV    A,#042H          ; SHOW ALL ROW IN SCREEN
          CALL   TTR1            ; MWTRITE
          ;MOV    R6,#8          ; COMMAND INPUT
          ;      ; NO. OF ROW_CHAR
ROW1:    ;MOV    R7,#32          ; NO. OF COL_CHAR
          mov    r7,#64
ROW2:    MOV    A,#083H
          CALL   TTR2
          DJNZ   R7,ROW2
          DJNZ   R6,ROW1
          RET

;*****
S_CHECKER:
          MOV    A,#042H          ; SHOW ALL SMALL CHECKER IN SCREEN
          CALL   TTR1            ; MWTRITE
          ;MOV    R6,#8          ; COMMAND INPUT
          ;      ; NO. OF ROW_CHAR
S_C1:    ;MOV    R7,#32          ; NO. OF COL_CHAR
          mov    r7,#64
S_C2:    MOV    A,#081H
          CALL   TTR2
          DJNZ   R7,S_C2
          DJNZ   R6,S_C1
          RET

;*****
B_CHECKER:
          MOV    A,#042H          ; SHOW ALL BIG CHECKER IN SCREEN
          CALL   TTR1            ; MWTRITE
          ;MOV    R7,#16          ; COMMAND INPUT

```

240320d1_f1 series sample program

```

    mov     r7,#20
BCHKS:  CALL  CHECKER_1
        CALL  CHECKER_2
        DJNZ  R7,BCHKS
        RET
;*****
ALIGNMENT:
        MOV   A,#042H           ; MWTRITE
        CALL  TTR1              ; COMMAND INPUT
        MOV   R7,#18h
        ;MOV  R7,#16
AL_1:   MOV   A,#84H
        CALL  TTR2
        DJNZ  R7,AL_1

        MOV   R6,#12h
        ;MOV  R6,#14           ; NO. OF ROW_CHAR
AL_2:   MOV   R7,#1eh
        ;MOV  R7,#16           ; NO. OF COL_CHAR
AL_3:   MOV   A,#020H
        CALL  TTR2
        DJNZ  R7,AL_3
        DJNZ  R6,AL_2
        MOV   R7,#1eh
        ;MOV  R7,#16
AL_4:   MOV   A,#85H
        CALL  TTR2
        DJNZ  R7,AL_4

        RET
;*****

CSRW:   MOV   A,#046H           ; CURSOR ADDRESS REGISTER
        CALL  TTR1              ; COMMAND INPUT

        MOV   A,#00H           ; P1 (CSRL = 0)
        CALL  TTR2
        MOV   A,#00H           ; P2 (CSRH = 0)
        CALL  TTR2
        RET
;*****

CSR_DIR: MOV   A,#04CH           ; CURSOR INCREMENT DIRECTION
        CALL  TTR1
        RET
;*****

CSR_FORM: MOV   A,#05DH           ; CURSOR SIZE AND DISPLAY MODE
        CALL  TTR1              ; COMMAND INPUT

        MOV   A,#07H           ; P1 (CRX = 7)
        CALL  TTR2
        MOV   A,#087H          ; P2 (CRY = 7, CM = 1)
        CALL  TTR2
        RET
;*****

TTR1:   ; COMMAD TRANSFER SUB-ROUTINE
        CLR   R_W
        SETB  A0
        CLR   CS
        NOP
        SETB  E
        MOV   P1,A
        NOP
        NOP
        CLR   E
        CLR   A0
        SETB  CS
        RET

TTR2:   ; DATA TRANSFER SUB-ROUTINE
        CLR   R_W
        CLR   A0
        CLR   CS
        NOP

```

240320d1_fl series sample program

```

SETB    E
MOV     P1,A
NOP
NOP
CLR     E
SETB    A0
SETB    CS
RET

;*****
CLR_L1:
        CALL    ADD1
        MOV     A, #42H
        CALL    TTR1
        MOV     R0, #20
L1:     MOV     R1, #10H
L2:     MOV     A, #20H
        CALL    TTR2
        DJNZ   R1, L2
        DJNZ   R0, L1
        RET

CLR_L2:
        CALL    ADD2
        MOV     A, #42H
        CALL    TTR1
        MOV     R0, #128
L3:     MOV     R1, #10H
L4:     MOV     A, #0H
        CALL    TTR2
        DJNZ   R1, L4
        DJNZ   R0, L3
        RET

;*****
ADD1:
        MOV     A, #46H
        CALL    TTR1
        MOV     A, #00H
        CALL    TTR2
        MOV     A, #00H
        CALL    TTR2
        RET

ADD2:
        MOV     A, #46H
        CALL    TTR1
        MOV     A, #00H
        CALL    TTR2
        MOV     A, #010H
        CALL    TTR2
        RET

;*****
DELAY:
        MOV     R5, #30
DL1:    MOV     R6, #250          ; TIME DELAY LOOP SUB-ROUTINE
DL2:    MOV     R7, #200
DL3:    DJNZ   R7, DL3
        DJNZ   R6, DL2
        DJNZ   R5, DL1
        RET

;*****
CHECKER_1:
        ;mov    r0, #15
        mov     r0, #8h
cs_1:   MOV     A, #80H
        CALL    TTR2
        MOV     A, #20H
        CALL    TTR2
        djnz   r0, cs_1
        RET

CHECKER_2:
        ;mov    r0, #15
        mov     r0, #8h
cs_2:   MOV     A, #20H
        CALL    TTR2
        MOV     A, #80H
        CALL    TTR2

```

```

    djnz    r0,cs_2
    RET
;*****
CG_RAM:                                     ; CG_RAM CODE GENERATION
    MOV     A,#5CH
    CALL    TTR1
    MOV     A,#00H
    CALL    TTR2
    MOV     A,#44H
    CALL    TTR2

    MOV     A,#4CH
    CALL    TTR1

    MOV     A,#46H
    CALL    TTR1
    MOV     A,#00H
    CALL    TTR2
    MOV     A,#48H
    CALL    TTR2

    MOV     A,#42H
    CALL    TTR1

    MOV     A,#0FFH ; 1ST CG_CODE 80H - FULL ON
    CALL    TTR2
    MOV     A,#0FFH
    CALL    TTR2
    MOV     A,#0FFH
    CALL    TTR2
    MOV     A,#0FFH
    CALL    TTR2
    MOV     A,#0FFH
    CALL    TTR2
    MOV     A,#0FFH
    CALL    TTR2
    MOV     A,#0FFH
    CALL    TTR2
    MOV     A,#0FFH
    CALL    TTR2
    MOV     A,#0FFH
    CALL    TTR2

    MOV     A,#0AAH ; 2ND CG_CODE 81H - CHECKER
    CALL    TTR2
    MOV     A,#55H
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#55H
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#55H
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#55H
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#55H
    CALL    TTR2

    MOV     A,#0AAH ; 3RD CG_CODE 82H - COLUMN
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2
    MOV     A,#0AAH
    CALL    TTR2

    MOV     A,#0FFH ; 4TH CG_CODE 83H - ROW
    CALL    TTR2
    MOV     A,#00H

```

240320d1_f1 series sample program

```
CALL    TTR2
MOV     A,#OFFH
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#OFFH
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#OFFH
CALL    TTR2
MOV     A,#00H
CALL    TTR2

MOV     A,#OFFH ; 4TH CG_CODE 84H - ALIGNMENT 1
CALL    TTR2
MOV     A,#OFFH
CALL    TTR2
MOV     A,#OFFH
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2

MOV     A,#00H ; 4TH CG_CODE 85 - ALIGNMENT 2
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#00H
CALL    TTR2
MOV     A,#OFFH
CALL    TTR2
MOV     A,#OFFH
CALL    TTR2
MOV     A,#OFFH
CALL    TTR2

RET
END
```