

8-BIT SINGLE CHIP MICROCOMPUTER

GMS84512

APPLICATION NOTE

- On Screen Display(OSD)
- Pulse Width Modulation(PWM)

HYUNDAI MicroElectronics

8-BIT SINGLE CHIP MICROCOMPUTER

OSD

GMS84512 APPLICATION NOTE

OSD (On Screen Display)



1. Overview

This is a program about multi-line OSD. GMS84512 is device displaying 3 lines at once for having 3 display memory.
Also, it is enable to extract 12 lines simultaneously by using OSD interrupt.

2. Hardware Introduction

2-1.MCU Operation

- ◆ OSD Interrupt
 - It is occurred when the display of each line is finished
 - It determines the position of next output line (n+3) and converted the value of display memory to output font.

- ◆ V Sync Interrupt
 - When a screen is outputted, V sync is occurred, so OSD interrupt is happened
 - It determines whether OSD is enable, also determines the vertical position and the horizontal position.
 - V sync is inputted to pin number 2 (Pin(VD)) and the polarity of signal is determined by chroma IC.

- ◆ External Interrupt
 - OSD is ON/OFF by external signal.
(It need not use for actual application field, it is written only to describe software.)

- ◆ R,G,B,Y Output
 - It determines the polarity of signal as specification of chroma IC and it is output pin of OSD data actually.

- ◆ Generator for OSD
 - If the clock of fast frequency is provided for OSD, horizontal dot sizes of font become small.
 - It is reasonable to provide clock more than 5.5Mhz to display 22 characters at one line.

APPLICATION

2-2.Circuit Diagram

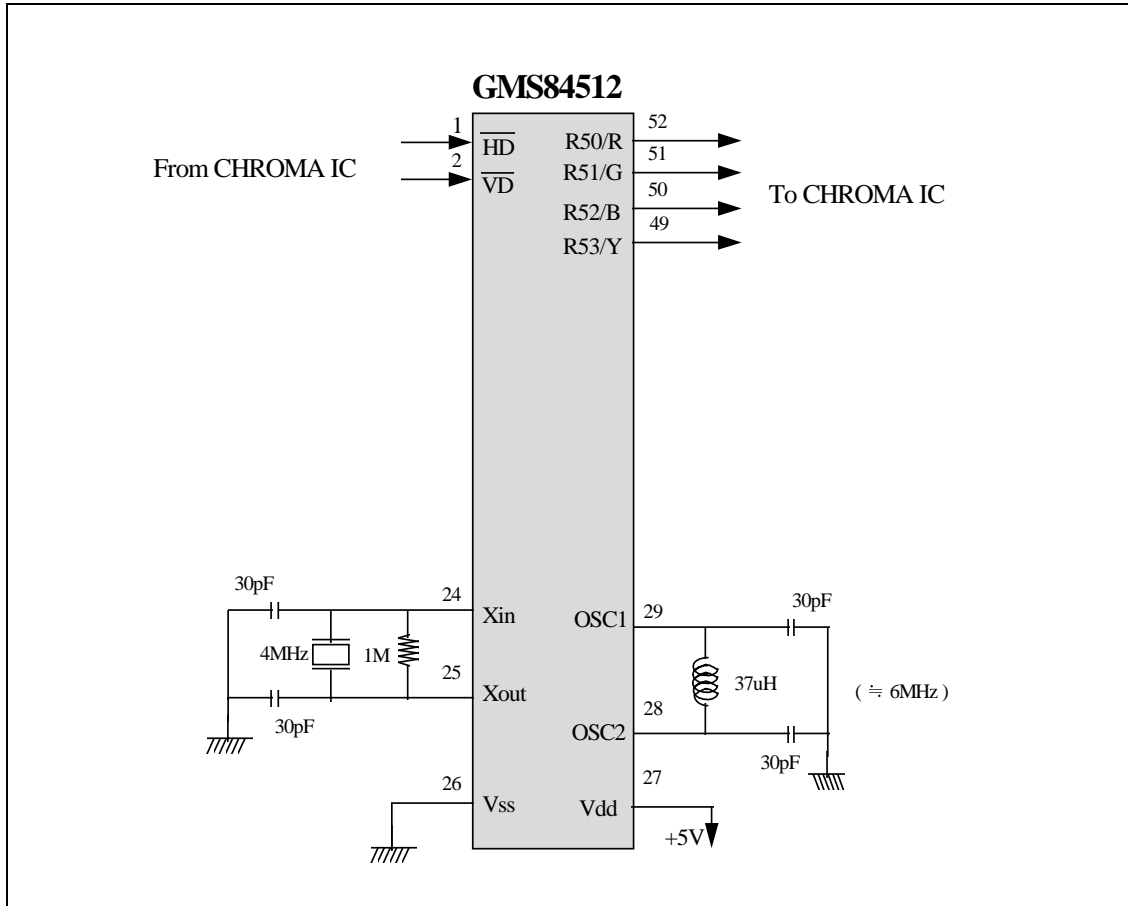


Fig 1. On Screen Display

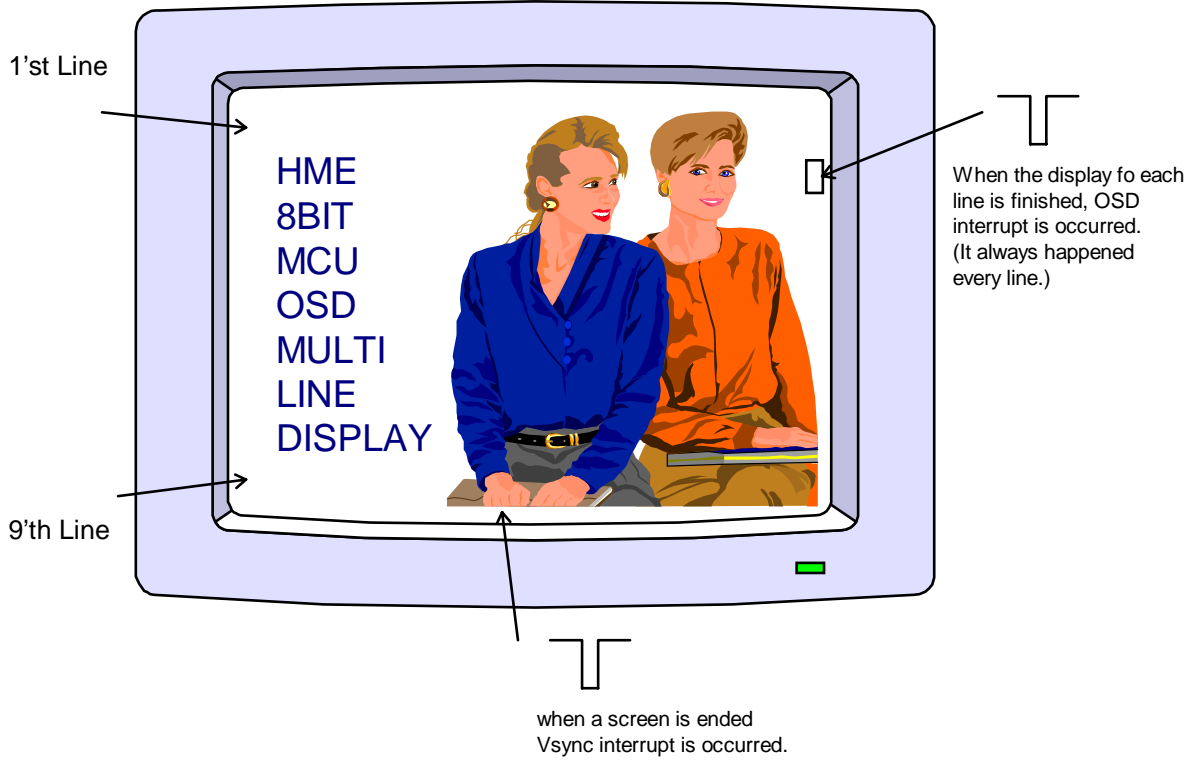
2-3. Pin Function

PIN NO.	NAME	I/O	DESCRIPTION
1	HD	Input	Horizontal synchronous signal
2	VD	Input	Vertical synchronous signal
15	R31/INT2	Input	External interrupt
28	OSC2	Output	Generator for OSD
29	OSC1	Input	Generator for OSD
49	R53/Y	Output	Output of OSD (BLINK)
50	R52/B	Output	Output of OSD (BLUE)
51	R51/G	Output	Output of OSD (GREEN)
52	R50/R	Output	Output of OSD (RED)

Table 1. PIN FUNCTIONS

2-4. H/W Operation

The following figure is a example of OSD screen using for sample program.



Example of displaying 9 lines.

- ◆ With having OSD interrupt in order to build multi-line OSD the contents of display have to be shown alternately.

APPLICATION

3. Software Description

3-1. Basic Operation

- ◆ OSD interrupt is occurred when the last character is output each line. It is necessary to change the position definition of next displaying line and the contents of display memory in this interrupt routine. Notice that processing data about next line and other line have to be done after current line is output. Brief description is following.

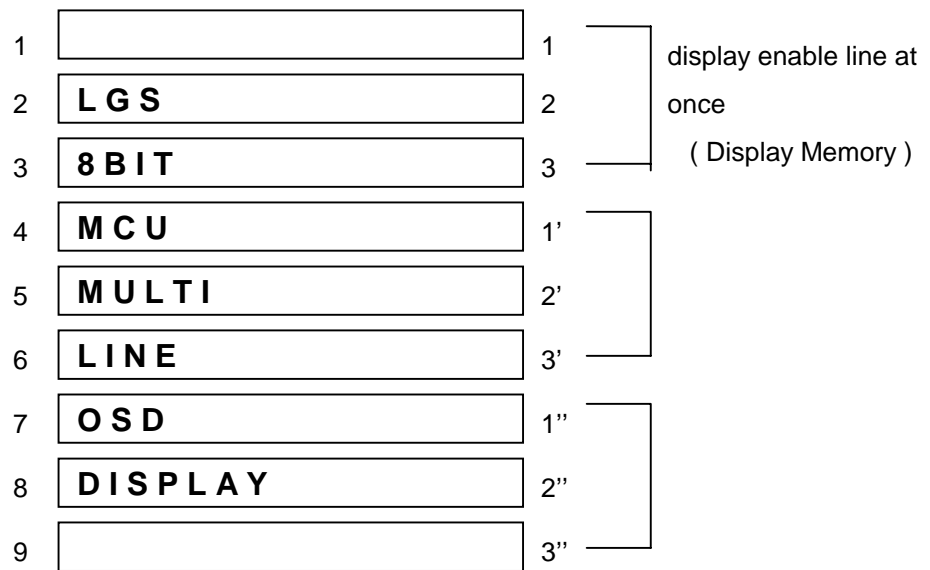


Fig 3 OSD Memory 이용 (Multi Line)

If 3'rd line is displayed currently position definition of 6'th line and the contents of display memory (LINE) have to be changed.

Don't process the next line data because real position to be located is incorrect for changing position definition of line and the contents of display memory.

The following figure shows OSD operation for passing time.

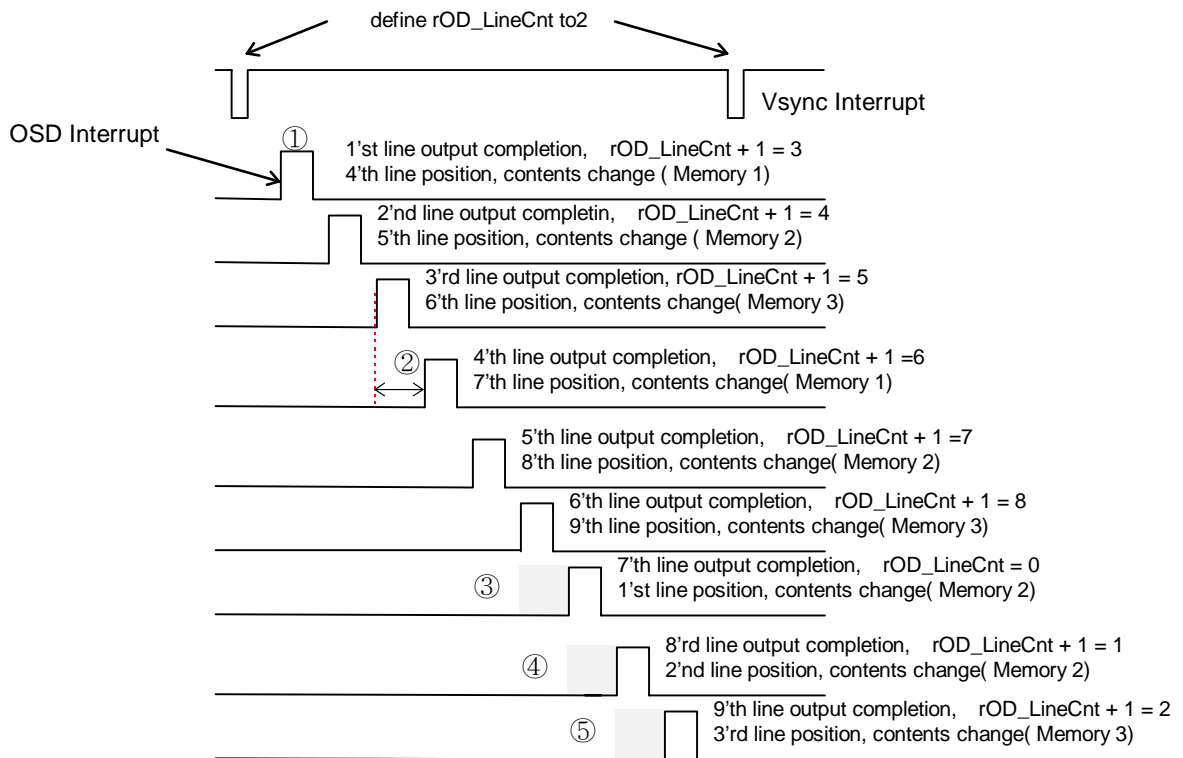


Fig 4. Time state diagram of Vsync Interrupt and OSD Interrupt

To begin with, if Vsync Interrupt is occurred define the position of 1'st, 2'nd, 3'rd line and store 2 into rOD_LineCnt register and then exit interrupt routine. If OSD interrupt is occurred when the last character of 1'st line is output, increase the value of rOD_LineCnt register by 1 and then store the position and the contents of 4'th line into the position register and display memory (Overwrite current memory 1 outputted)

So, after MICOM reset is released, the first OSD is full of blank at 1'st, 2'nd, 3'rd line because of being clear OSD memory in the reset routine and then it is displayed from 4'th line cyclic.(4,5,6,...9,1,2,3...)

After all, the OSD data stored at interrupt ① become output at interrupt② such as this, results (OSD data) become output sequentially through interrupt service routine.

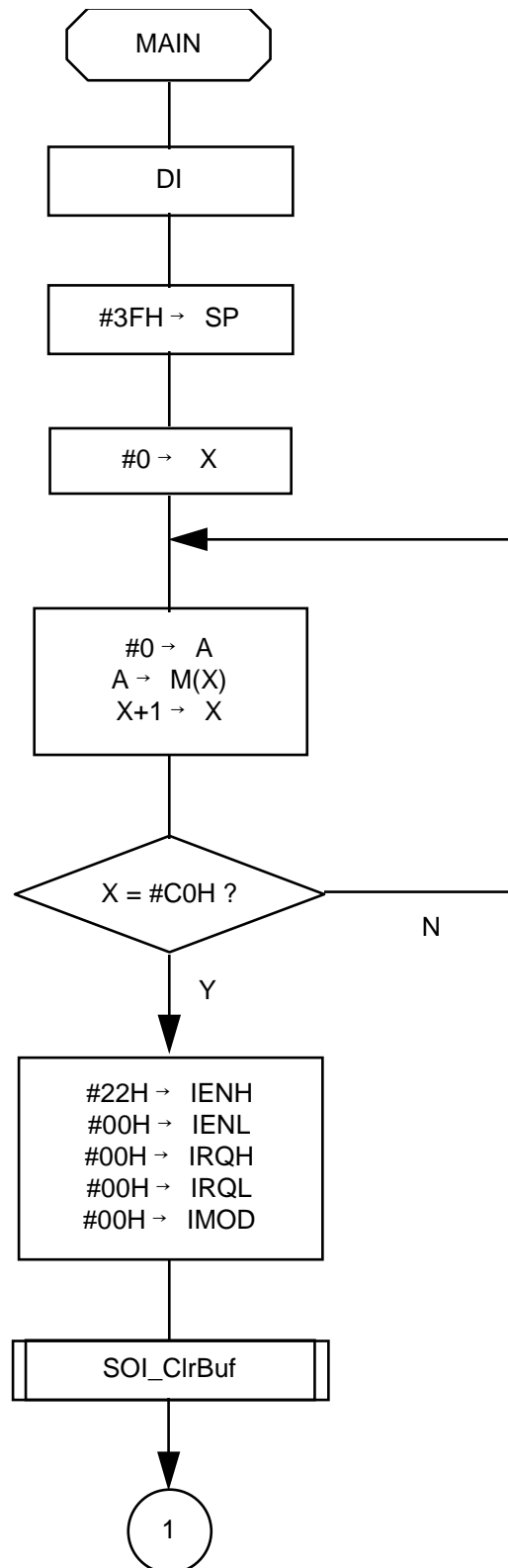
APPLICATION

3-2.RAM Description

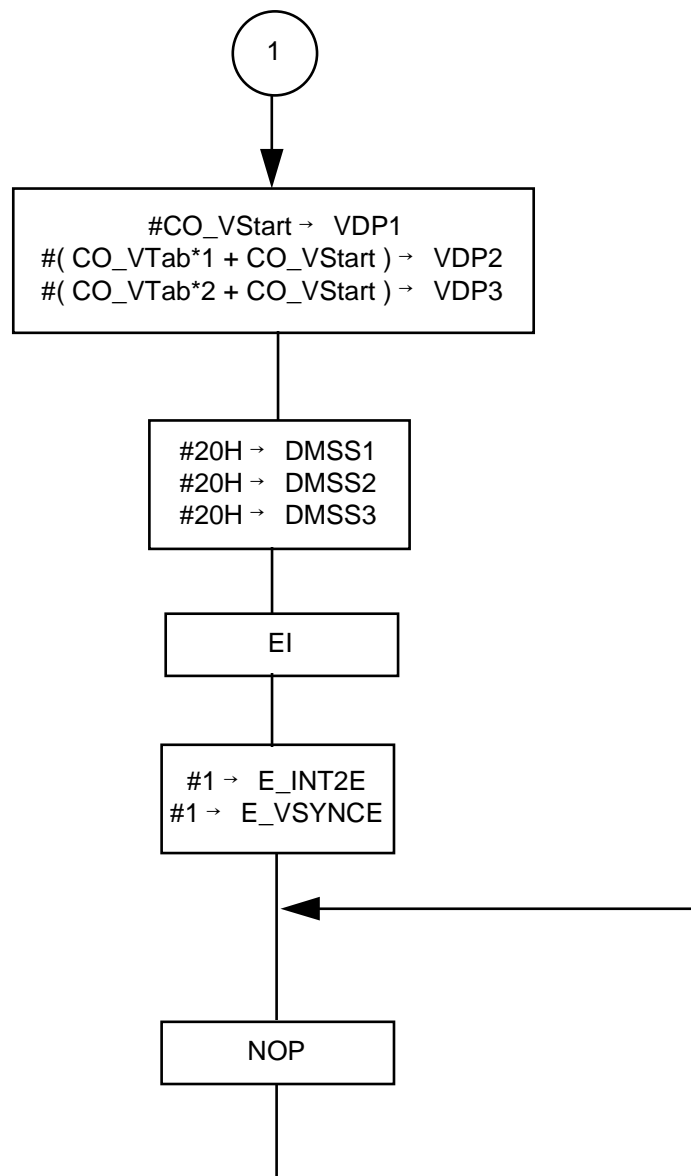
NAME	DESCRIPTION
rOD_LineCnt	Line counter for displaying next OSD
rOD_Color	OSD Color
rO_Temp1	DATA Working
rO_Temp2	DATA Working
rOD_CharCnt	The number of characters pushing Display Memory

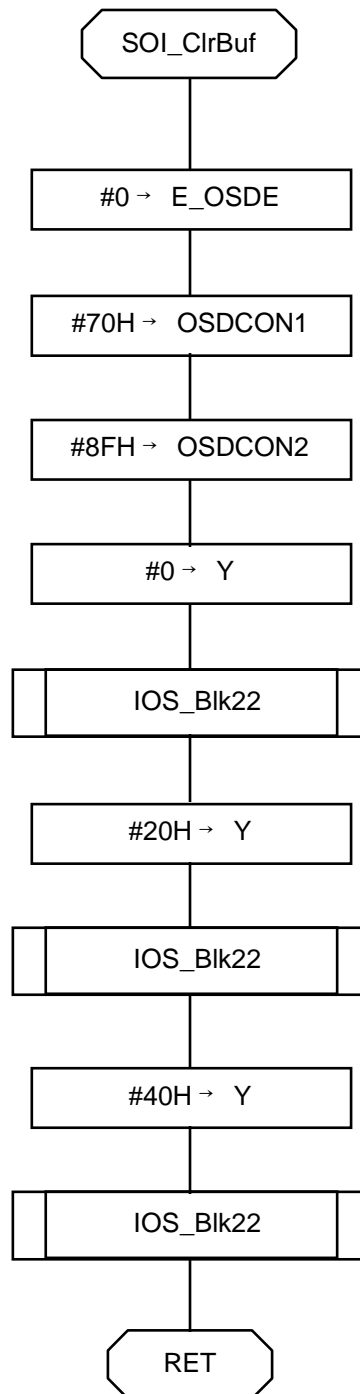
Table 2. RAM DESCRIPTION

3-3.Flow chart

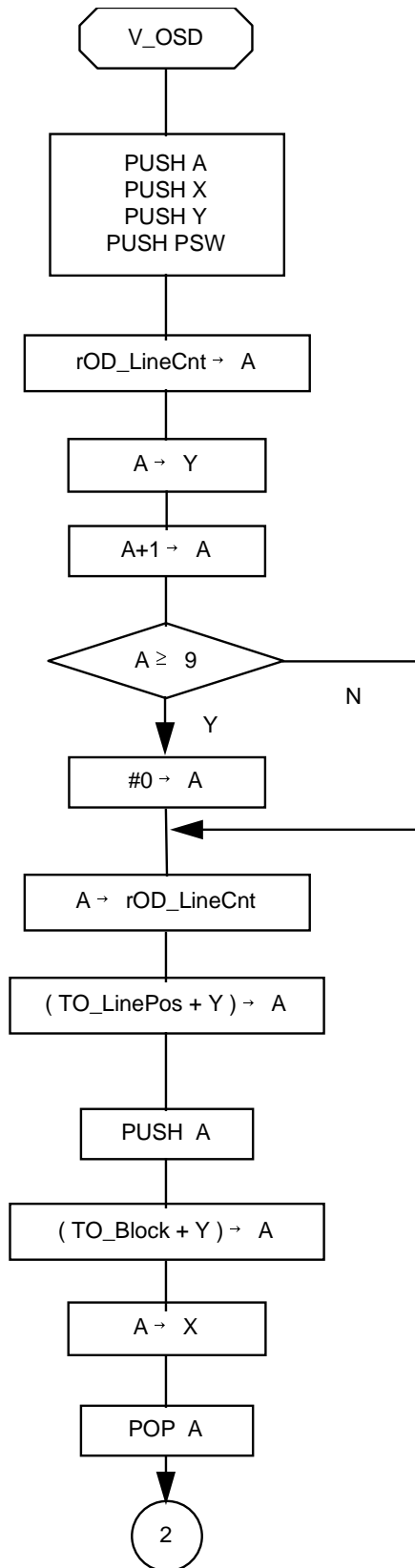


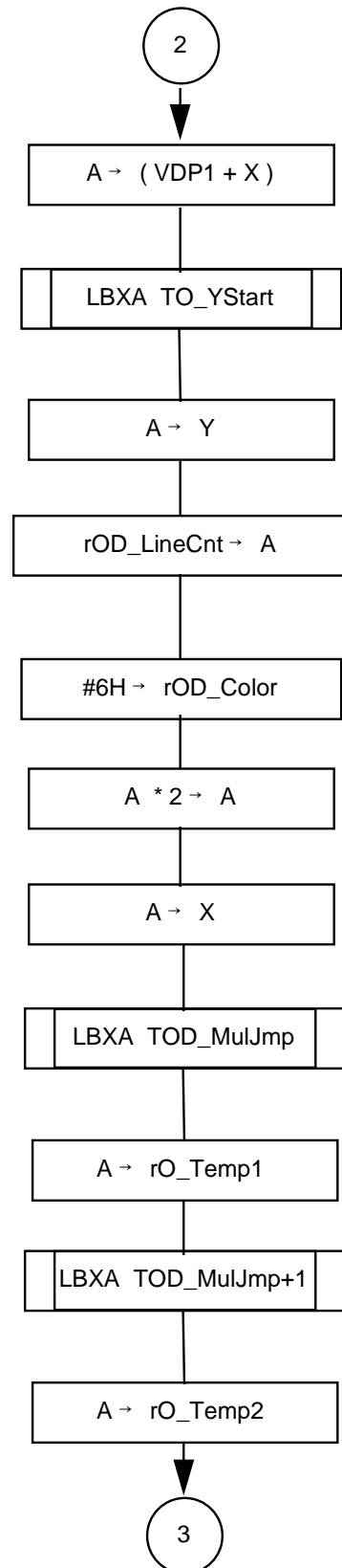
APPLICATION

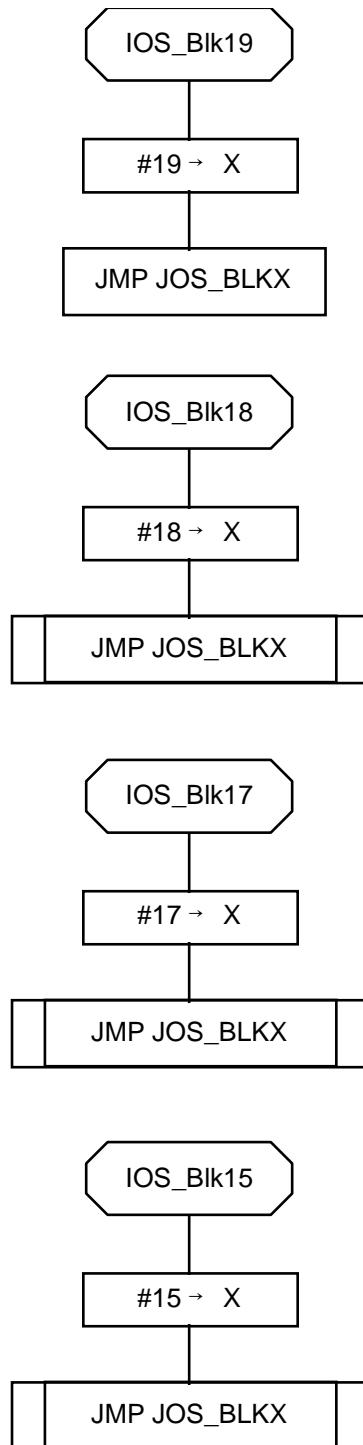




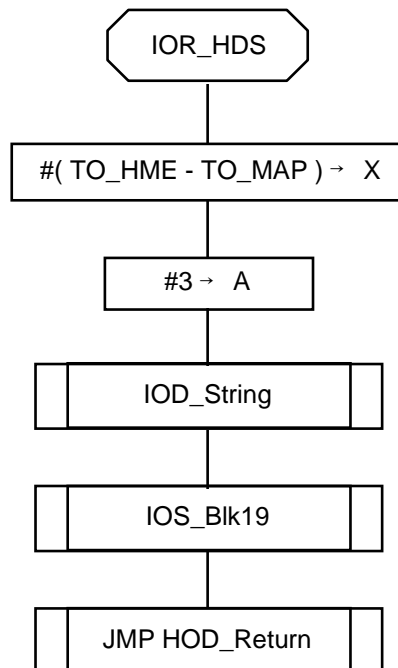
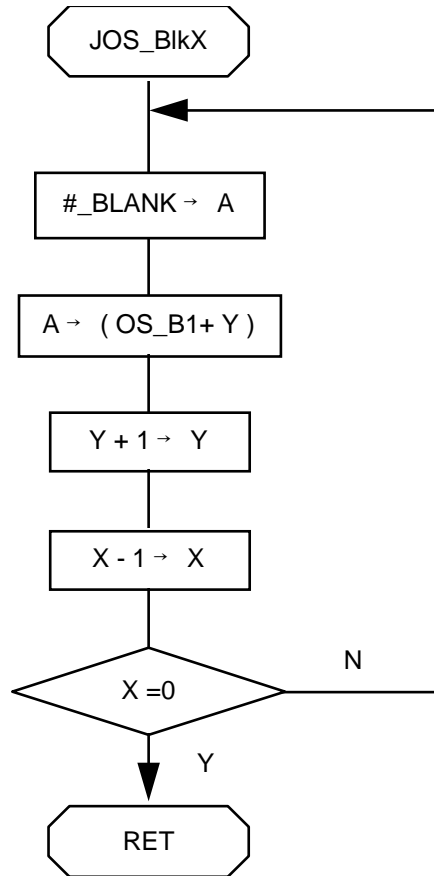
APPLICATION

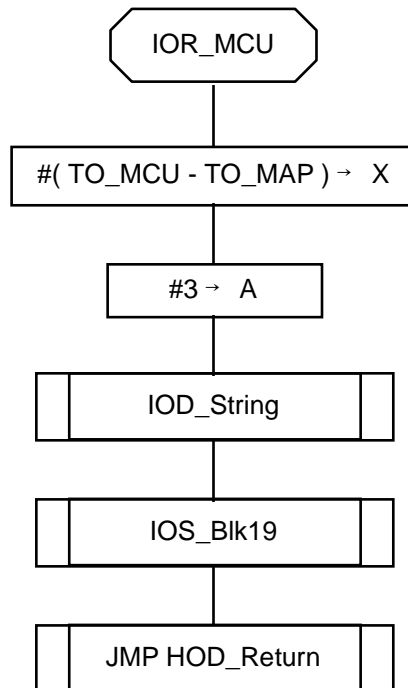
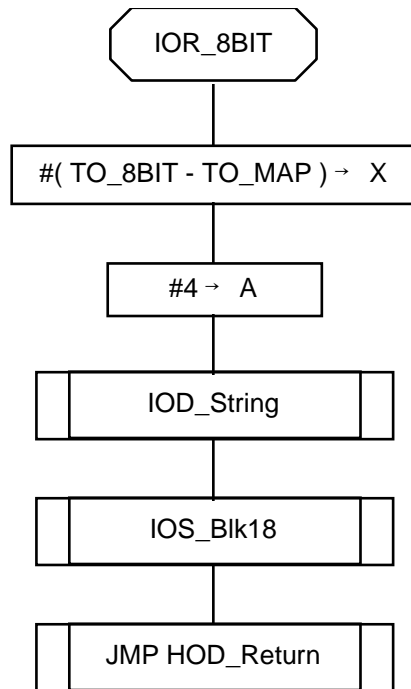




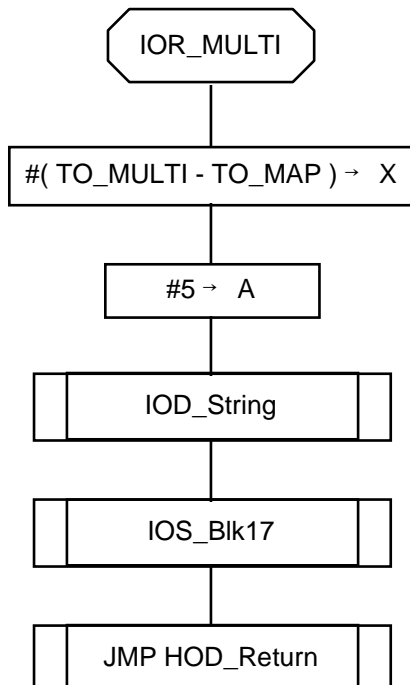
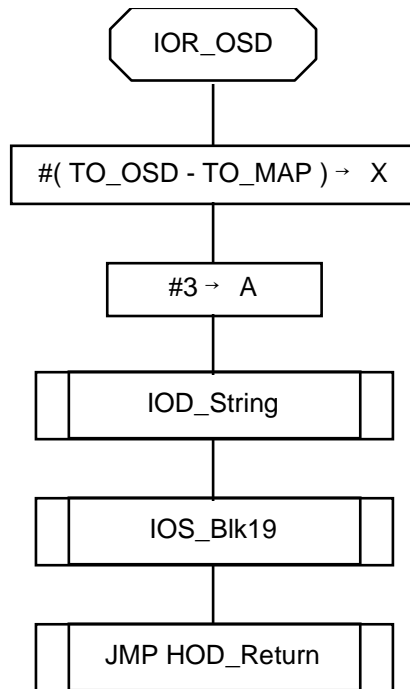


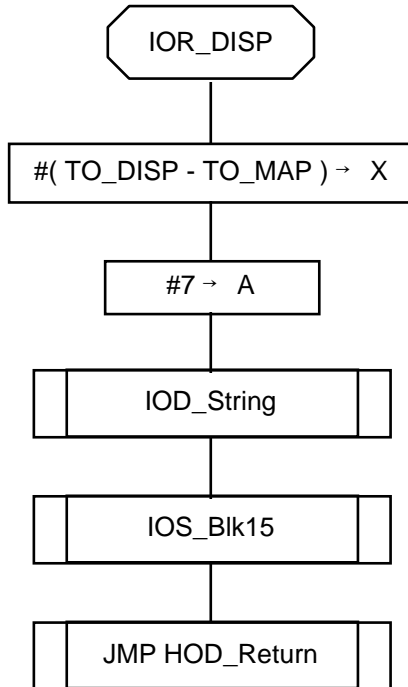
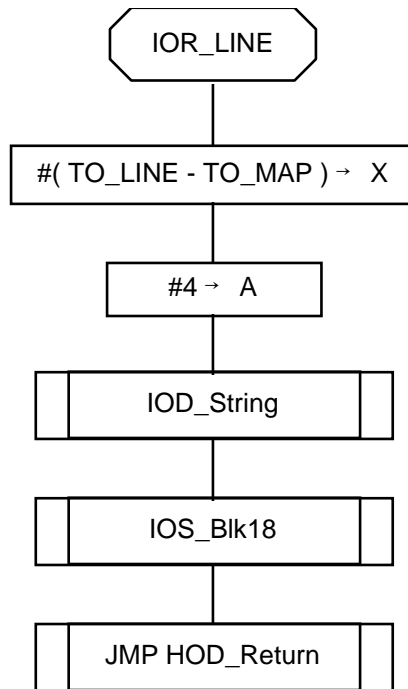
APPLICATION



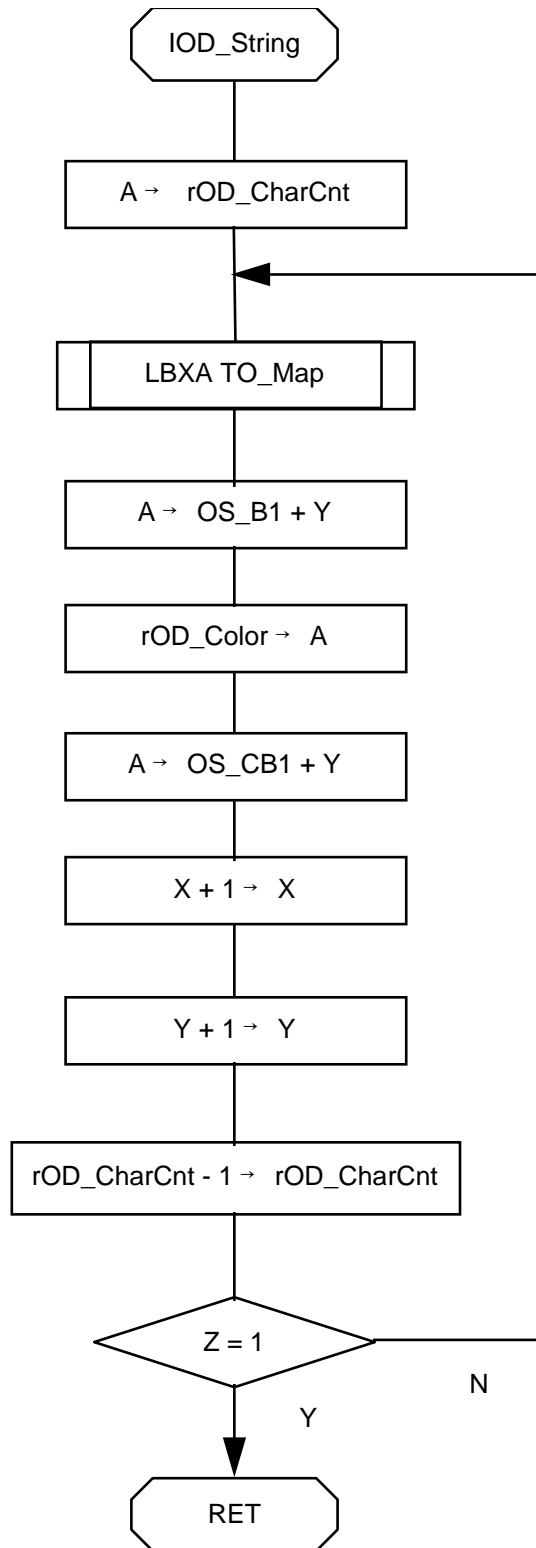


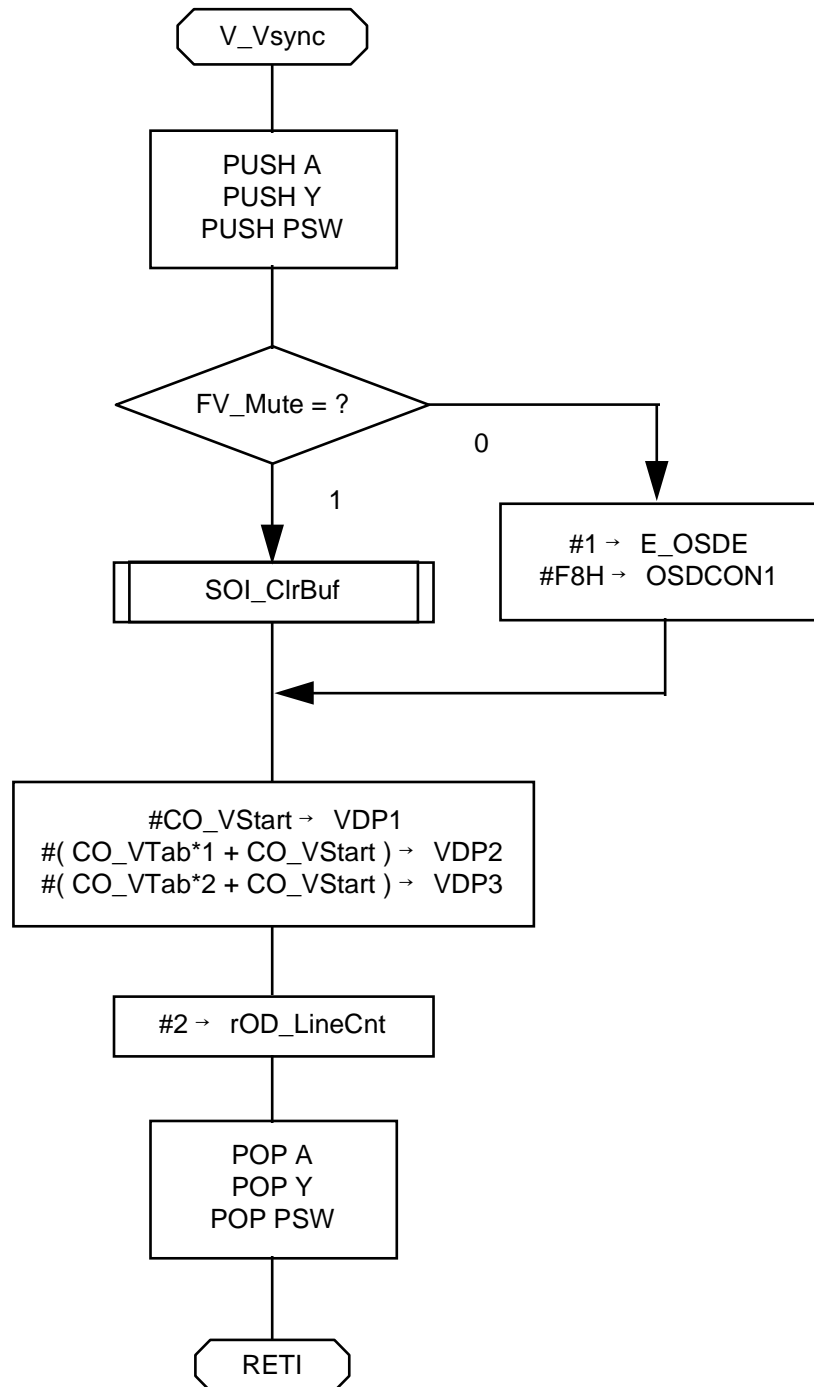
APPLICATION



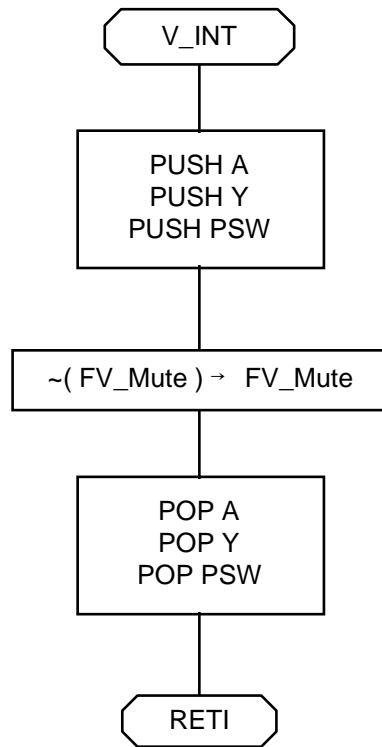


APPLICATION





APPLICATION



3-4. Source List

```

1      ;-----
2      lboxa MACRO          ; Load From Table by X-indexing
3          xyx              ; Change X <-> Y
4          lda    !1+y
5          xyx
6          ENDM
7      org 00h
8      rO_Temp1    ds    1      ; Working Area for OSD Interrupt
9      rO_Temp2    ds    1      ;
10     rOD_Mode     ds    1      ; OSD display Mode
11     rOD_Color    ds    1      ; The color of character to be displayed
12     rOD_CharCnt  ds    1      ; The number of character to be displayed
13     rOD_LineCnt  ds    1      ; The line counter to be displayed
14     rO_Flags     ds    1      ; The flags byte about OSD
15     FO_Enable    equ    0,rO_Flags ; Enable OSD display
16     FV_Mute      equ    1,rO_Flags ; Screen Mute
17     P_P4         equ    0C8H
18     P_P5         equ    0C9H    ; osd port
19     ;-----
20     ; * Interrupt Control Register
21     ;-----
22     IMOD         equ    0E6H    ; Interrupt Mode register
23     IENL         equ    0E8H    ; Interrupt Enable Register Low
24     IRQL         equ    0E9H    ; Interrupt Request Register Low
25     IENH         equ    0EAH    ; Interrupt Enable Register Low
26     E_VSYNCE     equ    1,IENH
27     E_INT2E      equ    5,IENH
28     E_OSDE       equ    7,IENH
29     IRQH         equ    0EBH    ; Interrupt Request Register Low
30     E_VSYNCR     equ    1,IRQH
31     E_INT2R      equ    5,IRQH
32     E_OSDR       equ    7,IRQH
33     ;-----

```

APPLICATION

```
34      ; * OSD Control Register
35      ;-----
36      HDP1      equ  0F0H      ; Block 1 horizontal position register
37      HDP2      equ  0F1H      ; Block 2      "
38      HDP3      equ  0F2H      ; Block 3      "
39      VDP1      equ  0F3H      ; Block 1 vertical position Register
40      VDP2      equ  0F4H      ; Block 2      "
41      VDP3      equ  0F5H      ; Block 3      "
42      DMSS1     equ  0F6H      ; Block 1 Display Mode
43      DMSS2     equ  0F7H      ; Block 2      "
44      DMSS3     equ  0F8H      ; Block 3      "
45      OSDCON1   equ  0F9H      ; OSD Output & Background Control
46      OSDCON2   equ  0FAH      ; OSD I/O Polarity & Oscillation Control
47      ;-----
48      ; * OSD Buffer Area
49      ;-----
50      OS_B1     equ  200H      ; OSD Buffer Block 1 Start Address
51      OS_B2     equ  220H      ;          2
52      OS_B3     equ  240H      ;          3
53      OS_CB1    equ  280H      ; OSD Color Buffer Block 1 Start Address
54      OS_CB2    equ  2A0H      ;          2
55      OS_CB3    equ  2C0H      ;          3
56      ;-----
57      ; * OSD character definition
58      ;-----
59      _0      =  0
60      _1      =  1
61      _2      =  2
62      _3      =  3
63      _4      =  4
64      _5      =  5
65      _6      =  6
66      _7      =  7
67      _8      =  8
```

```
68     _9  = 9
69     _A  = 0AH
70     _B  = 0BH
71     _C  = 0CH
72     _D  = 0DH
73     _E  = 0EH
74     _F  = 0FH
75     _G  = 10H
76     _H  = 11H
77     _I  = 12H
78     _J  = 13H
79     _K  = 14H
80     _L  = 15H
81     _M  = 16H
82     _N  = 17H
83     _O  = 18H
84     _P  = 19H
85     _Q  = 1AH
86     _R  = 1BH
87     _S  = 1CH
88     _T  = 1DH
89     _U  = 1EH
90     _V  = 1FH
91     _W  = 20H
92     _X  = 21H
93     _Y  = 22H
94     _Z  = 23H
95     _BLANK = 5FH
96     ;-----
97     ; * Constants about OSD
98     ;-----
99     ; * display Position
100    ; -----
101    CO_VStart equ 15
```

APPLICATION

```
102     CO_VTab          equ    11
103     ;-----
104     ; * Video RAM start Point of the Block
105     ;-----
106 0000 00     TO_YStart:  db    00H    ; Block-1 Start Address
107 0001 20          db    20H    ; Block-2 Start Address
108 0002 40          db    40H    ; Block-3 Start Address
109     ;-----
110     ; * Block Position for Multi-line
111     ;-----
112     TO_LinePos:                ; Line Display V-Position
113 0003 1A      db    (CO_VTab * 1) + CO_VStart ; 1'nd      "
114 0004 25      db    (CO_VTab * 2) + CO_VStart ; 2'nd      "
115 0005 30      db    (CO_VTab * 3) + CO_VStart ; 3'rd      "
116 0006 3B      db    (CO_VTab * 4) + CO_VStart ; 4'th      "
117 0007 46      db    (CO_VTab * 5) + CO_VStart ; 5'th      "
118 0008 51      db    (CO_VTab * 6) + CO_VStart ; 6'th      "
119 0009 5C      db    (CO_VTab * 7) + CO_VStart ; 7'th      "
120 000A 67      db    (CO_VTab * 8) + CO_VStart ; 8'th      "
121 000B 0F      db    CO_VStart
122     ;-----
123     ; * Block position for Channel OSD
124     ;-----
125     TO_ChPos:
126 000C 25      db    (CO_VTab * 2) + CO_VStart ; 2'nd Line Display V-Position
127 000D 30      db    (CO_VTab * 3) + CO_VStart ; 3'rd Line Display V-Position
128 000E 0F      db    CO_VStart                ; 1'st Line Display V-Position
129     ;-----
130     ; * Used block to multi line
131     ;-----
132     TO_Block:
133 000F 01      db    1      ; OSD block utilizing to make 1'st line
134 0010 02      db    2      ; OSD block utilizing to make 2'nd line
135 0011 00      db    0      ; OSD block utilizing to make 3'rd line
```

```

136 0012 01      db      1      ; OSD block utilizing to make 4'th line
137 0013 02      db      2      ; OSD block utilizing to make 5'th line
138 0014 00      db      0      ; OSD block utilizing to make 6'th line
139 0015 01      db      1      ; OSD block utilizing to make 7'th line
140 0016 02      db      2      ; OSD block utilizing to make 8'th line
141 0017 00      db      0      ; OSD block utilizing to make 9'th line
142      TOD_MuJmp:
143 0018 BBD0      dw      IOD_ClrLine
144 001A F8D0      dw      IOR_HME
145 001C 05D1      dw      IOR_8BIT
146 001E 12D1      dw      IOR_MCU
147 0020 1FD1      dw      IOR_OSD
148 0022 2CD1      dw      IOR_MULTI
149 0024 39D1      dw      IOR_LINE
150 0026 46D1      dw      IOR_DISP
151 0028 BBD0      dw      IOD_ClrLine
152      ;=====
153      TO_Map:
154 002A 15      TO_HDS db      _H,_M,_E
      002B 10
      002C 1C
155 002D 08      TO_8BIT: db      _8,_B,_I,_T
      002E 0B
      002F 12
      0030 1D
156 0031 16      TO_MCU: db      _M,_C,_U
      0032 0C
      0033 1E
157 0034 18      TO_OSD: db      _O,_S,_D
      0035 1C
      0036 0D
158 0037 16      TO_MULTI: db      _M,_U,_L,_T,_I
      0038 1E
      0039 15

```

APPLICATION

```
003A 1D
003B 12
159 003C 15    TO_LINE: db    _L,_I,_N,_E
003D 12
003E 17
003F 0E
160 0040 0D    TO_DISP: db    _D,_I,_S,_P,_L,_A,_Y
0041 12
0042 1C
0043 19
0044 15
0045 0A
0046 22
161          ; * Program start Address
162          ; -----
163          org 0d000h
164          MAIN:
165  D000 60          di
166  D001 1E3F        idx    #3Fh    ; Stack Pointer
167  D003 8E          txsp
168          ; * Port 4 & 5is output port
169  D004 E400C8      ldm    P_P4 ,#00000000B
170  D007 E400C9      ldm    P_P5 ,#00000000B
171          ; * RAM Clear (00.. BFh)
172          ; -----
173  D00A 1E00        idx    #0
174          JM_RAMClr:
175  D00C C400        lda    #0
176  D00E FB          sta    {x}+
177  D00F 5EC0        cmpx  #0C0h
178  D011 70F9        bne    JM_RAMClr
179          ; * Interrupt Control initialize
180          ; -----
181  D013 E422EA      ldm    IENH,#00100010B ; INT2,V-Sync
```

```

182 D016 E400E8      ldm    IENL,#00000000B
183 D019 E400EB      ldm    IRQH,#00000000B ; Clear All Interrupt Request
184 D01C E400E9      ldm    IRQL,#00000000B
185 D01F E400E6      ldm    IMOD,#00000000B
186 D022 3B3FD0      call   SOI_ClrBuf
187      ; * OSD initializatin & Buffer Clear
188      ; -----
189 D025 E40FF3      ldm    VDP1,#CO_VSTART ; OSD Block-1 Position
190 D028 E425F4      ldm    VDP2,#(CO_VTAB*2 + CO_VSTART) ; OSD Block-2 Position
191 D02B E467F5      ldm    VDP3,#(CO_VTAB*8 + CO_VSTART) ; OSD Block-3 Position
192 D02E E420F6      ldm    DMSS1,#00100000B ; the value = 0
193 D031 E420F7      ldm    DMSS2,#00100000B ; the value = 0
194 D034 E420F8      ldm    DMSS3,#00100000B ; the value = 0
195 D037 E0          ei
196 D038 A1EA        set1   E_INT2E
197 D03A 21EA        set1   E_VSYNCE
198      JMR_Loop:
199 D03C FF          nop
200 D03D 2FFD        bra    JMR_Loop ; Loop again
201      ;-----
202      SOI_ClrBuf:
203 D03F F1EA        clr1   E_OSDE ; disable OSD interrupt
204 D041 E48FFA      ldm    OSDCON2,#10001111B
205 D044 3E00        ldy    #00 ; 1'st address of OSD block
206 D046 3BC4D0      call   IOS_Blk22
207 D049 3E20        ldy    #20h ; 2'nd address of OSD block
208 D04B 3BC4D0      call   IOS_Blk22
209 D04E 3E40        ldy    #40h ; 3'rd address of OSD block
210 D050 3BC4D0      call   IOS_Blk22
211 D053 6F          ret
212      V_INT:
213 D054 0E          push   a
214 D055 4E          push   y
215 D056 6E          push   psw

```

APPLICATION

```
216 D057 4B0620      not1  FV_Mute
217 D05A 6D          pop   psw
218 D05B 4D          pop   y
219 D05C 0D          pop   a
220 D05D 7F          reti
221      V_VSync:
222 D05E 0E          push  a
223 D05F 4E          push  y
224 D060 6E          push  psw
225      IF      [FV_Mute]
226 D061 330605 @      bbc   FV_Mute,@_2
227 D064 3B3FD0      call  SOI_ClrBuf
228      ELSE
229 D067 2F07 @      bra   @_3
230      @ @_2:
231 D069 F1EB        clr1  E_OSDR
232 D06B E1EA        set1  E_OSDE
233 D06D E4F8F9      ldm   OSDCON1,#11111000B ; OSD On
234      ENDIF
235      @ @_3:
236 D070 E40FF3      ldm   VDP1,#CO_VSTART ; V1_START POSITON
237 D073 E41AF4      ldm   VDP2,#(CO_VTAB*1 + CO_VSTART) ; V2_START POSITON
238 D076 E425F5      ldm   VDP3,#(CO_VTAB*2 + CO_VSTART) ; V3_START POSITON
239 D079 E40205      ldm   ROD_LINECNT,#2
240 D07C 6D          pop   psw
241 D07D 4D          pop   y
242 D07E 0D          pop   a
243 D07F 7F          reti
244      V_OSD:
245 D080 E0          ei
246 D081 0E          push  a ; Push Acc.
247 D082 2E          push  x ; Push X Reg.
248 D083 4E          push  y ; Push Y Reg.
249 D084 6E          push  psw ; Push Program Status Word
```

```

250 D085 C505     lda     rOD_LineCnt    ; Dispalyng Line
251 D087 9F      tay
252 D088 88      inc     a             ; Next displaying Line
253 D089 4409    cmp     #9            ; Max. 9 Line
254             IF     c == 1
255 D08B 5002    @         bcc     @__5
256 D08D C400    lda     #0           ; Display from 1'st Line
257             ENDIF
258             @ @__5:
259 D08F E505    sta     rOD_LineCnt
260 D091 D50300  lda     !TO_LinePos+y
261             ; *****
262             ; OSD position translation
263             ; *****
264             HO_SetPos:
265 D094 0E      push    a             ; a = the position to display OSD data at a screen
266 D095 D50F00  lda     !TO_Block+y   ; a = determine OSD buffer
267 D098 E8      tax
268             ; x = determine OSD buffer
269 D099 0D      pop     a
270 D09A E6F3    sta     VDP1+x       ; SET V-POSITION DATA
271             lbra TO_YStart
272 D09C FE      @         xyx
273 D09D D50000  @         lda     !TO_YStart+y
274 D0A0 FE      @         xyx
275 D0A1 9F      tay
276             ; y = display OSD Buffer ADDRESS
277 D0A2 CC05    idx     rOD_LineCnt
278 D0A4 E40603  ldm     ROD_COLOR,#06H ; Major Color = Cyan
279             HOD_MulJmp:
280 D0A7 08      asl     a
281 D0A8 E8      tax
282             lbra TOD_MulJmp
283 D0A9 FE      @         xyx
284 D0AA D51800  @         lda     !TOD_MulJmp+y
285 D0AD FE      @         xyx

```

APPLICATION

```
284 D0AE E500      sta    rO_Temp1
285          lbxa  TOD_MulJmp+1
286 D0B0 FE        @      xyx
287 D0B1 D51900   @      lda    !TOD_MulJmp+1+y
288 D0B4 FE        @      xyx
289 D0B5 E501      sta    rO_Temp2
290 D0B7 CC05      ldx    rOD_LineCnt
291 D0B9 3F00      jmp    [rO_Temp1]
292          IOD_ClrLine:
293 D0BB 3BC4D0    call   IOS_Blk22
294          HOD_Return:
295 D0BE 6D        pop    psw      ; Restore Register's
296 D0BF 4D        pop    y
297 D0C0 2D        pop    x
298 D0C1 0D        pop    a
299 D0C2 E0        ei      ; Enable Interrupt
300 D0C3 7F        V_Return: reti
301 D0C4 1E16      IOS_Blk22: ldx  #22
302 D0C6 2F10      bra    JOS_BlkX
303 D0C8 1E13      IOS_Blk19: ldx #19
304 D0CA 2F0C      bra    JOS_BlkX
305 D0CC 1E12      IOS_Blk18: ldx #18
306 D0CE 2F08      bra    JOS_BlkX
307 D0D0 1E11      IOS_Blk17: ldx #17
308 D0D2 2F04      bra    JOS_BlkX
309 D0D4 1E0F      IOS_Blk15: ldx #15
310 D0D6 2F00      bra    JOS_BlkX
311          JOS_BlkX:
312 D0D8 C45F      lda    #_BLANK
313 D0DA F50002    sta    !OS_B1+y
314 D0DD 9E        inc    y
315 D0DE AF        dec    x
316 D0DF 70F7      bne    JOS_BlkX
317 D0E1 6F        ret
```

```
318     IOD_String:
319 D0E2 E504     sta     rOD_CharCnt
320     HOD_StrLp:
321     lboxa TO_Map
322 D0E4 FE       @       xyx
323 D0E5 D52A00  @       lda     !TO_Map+y
324 D0E8 FE       @       xyx
325 D0E9 F50002     sta     !OS_B1+y
326 D0EC C503     lda     rOD_Color
327 D0EE F58002     sta     !OS_CB1+y
328 D0F1 8F       inc     x
329 D0F2 9E       inc     y
330 D0F3 A904     dec     rOD_CharCnt
331 D0F5 70ED     bne     HOD_StrLp
332 D0F7 6F       ret
333     IOR_HME:
334 D0F8 1E00     idx     #(TO_HDS - TO_MAP)
335 D0FA C403     lda     #3
336 D0FC 3BE2D0     call    IOD_String
337 D0FF 3BC8D0     call    IOS_Bl19
338 D102 1BBED0     jmp     HOD_Return
339     IOR_8BIT:
340 D105 1E03     idx     #(TO_8BIT - TO_MAP)
341 D107 C404     lda     #4
342 D109 3BE2D0     call    IOD_String
343 D10C 3BCCD0     call    IOS_Bl18
344 D10F 1BBED0     jmp     HOD_Return
345     IOR_MCU:
346 D112 1E07     idx     #(TO_MCU - TO_MAP)
347 D114 C403     lda     #3
348 D116 3BE2D0     call    IOD_String
349 D119 3BC8D0     call    IOS_Bl19
350 D11C 1BBED0     jmp     HOD_Return
351     IOR_OSD:
```

APPLICATION

```
352 D11F 1E0A      ldx    #(TO_OSD - TO_MAP)
353 D121 C403      lda     #3
354 D123 3BE2D0    call   IOD_String
355 D126 3BC8D0    call   IOS_Blk19
356 D129 1BBED0    jmp     HOD_Return
357             IOR_MULTI:
358 D12C 1E0D      ldx    #(TO_MULTI - TO_MAP)
359 D12E C405      lda     #5
360 D130 3BE2D0    call   IOD_String
361 D133 3BD0D0    call   IOS_Blk17
362 D136 1BBED0    jmp     HOD_Return
363             IOR_LINE:
364 D139 1E12      ldx    #(TO_LINE - TO_MAP)
365 D13B C404      lda     #4
366 D13D 3BE2D0    call   IOD_String
367 D140 3BCCD0    call   IOS_Blk18
368 D143 1BBED0    jmp     HOD_Return
369             IOR_DISP:
370 D146 1E16      ldx    #(TO_DISP - TO_MAP)
371 D148 C407      lda     #7
372 D14A 3BE2D0    call   IOD_String
373 D14D 3BD4D0    call   IOS_Blk15
374 D150 1BBED0    jmp     HOD_Return
375             ;-----
376             ; * Interrupt Vector Table
377             ;-----
378             org 0ffe2h
379 FFE2 C3D0      dw     V_Return    ; Serial I/O Interrupt    ffe2
380 FFE4 C3D0      dw     V_Return    ; Basic Interval Timer    e4
381 FFE6 C3D0      dw     V_Return    ; WatchDog Timer          e6
382 FFE8 C3D0      dw     V_Return    ; External Interrupt 3    e8
383 FFEA C3D0      dw     V_Return    ; Timer 3                  ea
384 FFEC C3D0      dw     V_Return    ; Timer 1 (250ms Interrupt) ec
385 FFEE 5ED0      dw     V_VSync     ; V-Sync Interrupt        ee
```

```
386 FFF0 C3D0    dw    V_Return    ; 1 mS Interrupt    f0
387 FFF2 C3D0    dw    V_Return    ; Timer 2          f2
388 FFF4 C3D0    dw    V_Return    ; Timer 0          f4
389 FFF6 54D0    dw    V_INT       ; External Interrupt 2 f6
390 FFF8 C3D0    dw    V_Return    ; External Interrupt 1 f8
391 FFFA 80D0    dw    V_OSD       ; Multi-line OSD    fa
392 FFFC C3D0    dw    V_Return    ; Undefined         fc
393 FFFE 00D0    dw    MAIN        ; Reset (NMI)      fe
394             end
```

-- 0 Error(s) --

--- Total Machine Code : 440 Bytes --

8-BIT SINGLE CHIP MICROCOMPUTER

PWM

GMS84512 APPLICATION NOTE

PWM (Pulse Width Modulation)



1. Overview

This program is about PWM, it can control the- analog value of the external device. GMS84512 is built in 1 channel 14bit PWM and 8 channel 7bit PWM.

2. Hardware Introduction

2-1.MCU Operation

- ◆ 14Bit PWM (PWM8)
 - It has mixing with R32 port, it is selected by setting bit1 of PWMCR1 register.
 - The duty of basic pulse is determined by upper 8bit data having 256 steps. And the position of pulse added basic pulse is determined by lower 6bit data. That is 64 pulse positions are determined where they are located by lower 6bit data.
 - Also polarity is determined by bit2 of PWMCR2.
 - After data are stored at PWM data register, define I/O direction mode register to output and then PWM data is to be output.

- ◆ 7Bit PWM (PWM0 ~ 7)
 - They are mixing with R45~R40, R37, R36. Each PWM is selected by the control bit of PWMCR1, PWMCR2 register.
 - The duty of 128 steps is selected by 7bit data.
 - Also, the polarity of PWM is determined by bit3 of PWMCR2 register. (Common 8ch)
 - After data are stored at PWM data register, define I/O direction mode register to output and then PWM data is to be output.
 - Notes that port construction is NMOS open drain so have to connect pull-up resistance.

APPLICATION

2-2. Circuit Diagram

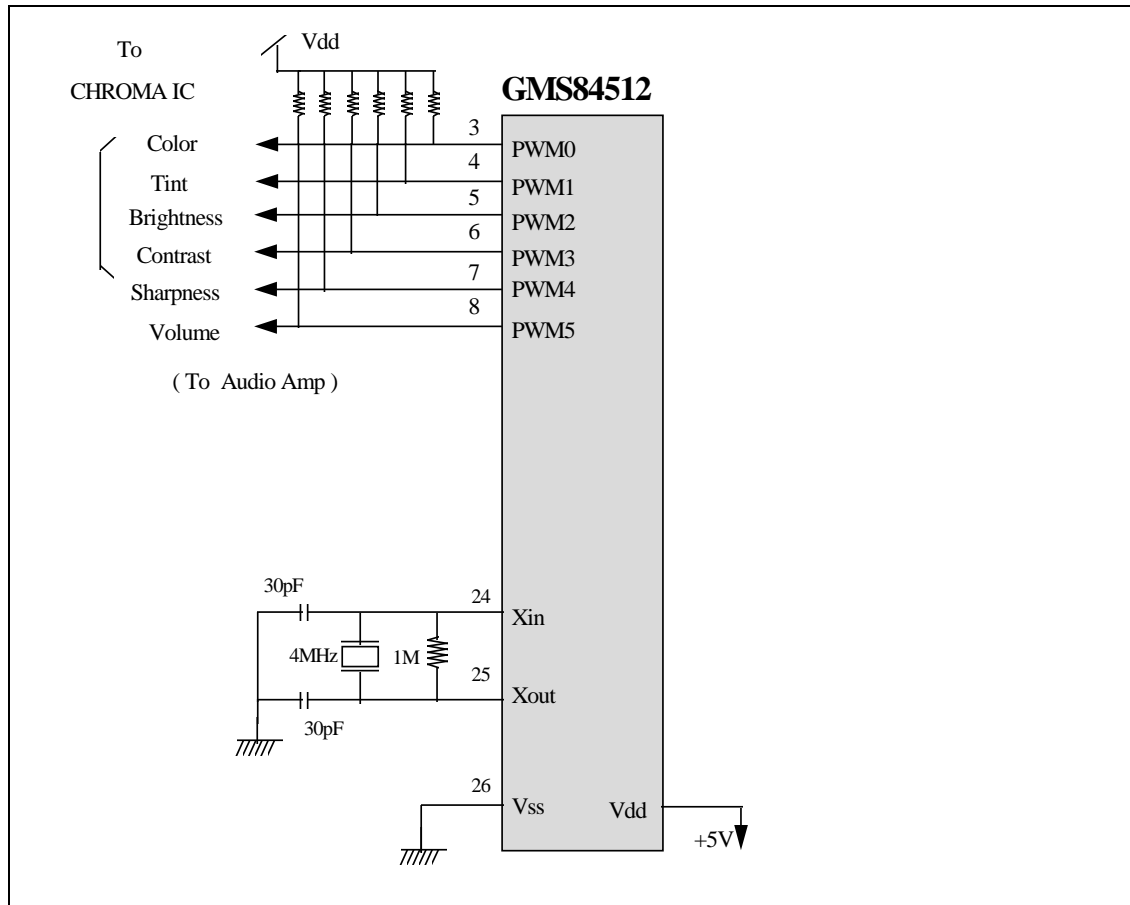


Fig.1. PWM Output

2-3. Pin Function

PIN NO.	NAME	I/O	DESCRIPTION
3	R45/PWM0	Output	7Bit PWM output (Color)
4	R44/PWM1	Output	7Bit PWM output (Tint)
5	R43/PWM2	Output	7Bit PWM output (Brightness)
6	R42/PWM3	Output	7Bit PWM output (Contrast)
7	R41/PWM4	Output	7Bit PWM output (Sharpness)
8	R40/PWM5	Output	7Bit PWM output (Volume)
9	R37/PWM6	Input/Output	Normal Data I/O Port
10	R36/PWM7	Input/Output	Normal Data I/O Port
14	R32/PWM8	Input/Output	Normal Data I/O Port

Table 1. PIN FUNCTIONS

2-4. H/W Operation

Following figure is the example of 14-bit PWM in active high.

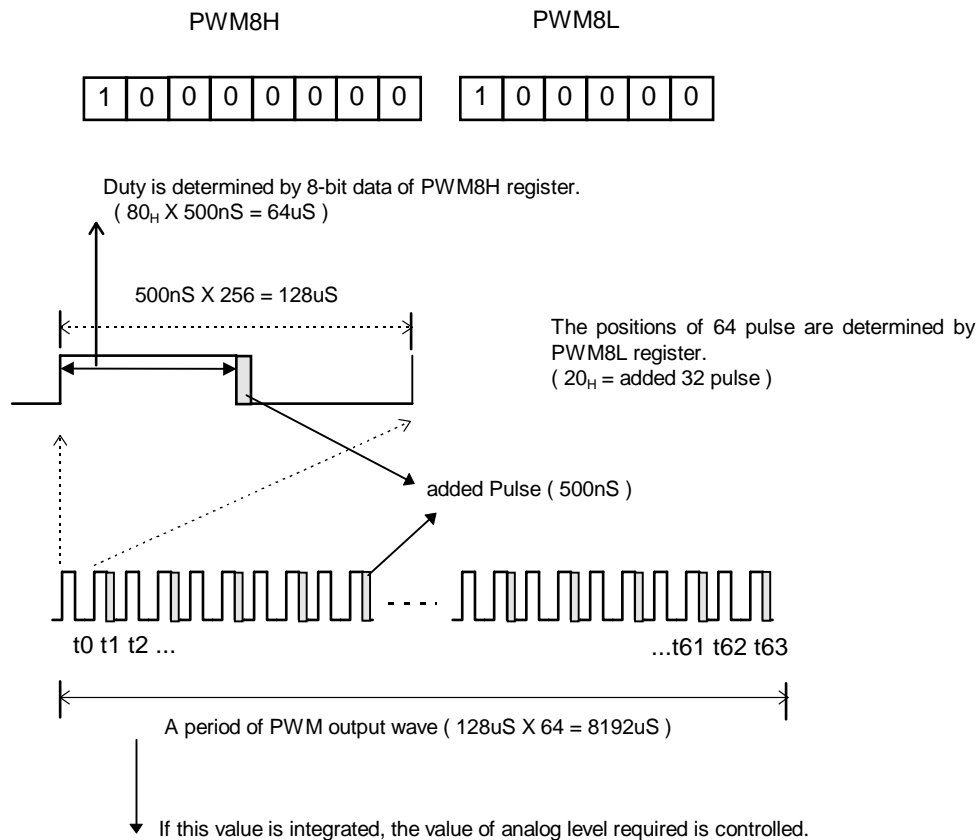
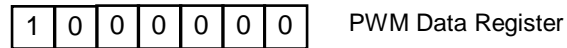


Fig 2. The example of output wave for 14Bit PWM

- ◆ As this example, the resolution of $0.5\mu s/8192\mu s$ is enable to output. Normally, the upper 8bit is fixed and the lower 6bit is varied to control precisely at small range.
- ◆ Both the registers of PWM8H and PWM8L are enable R/W. And when you will change the contents of PWM8H, be sure to write twice, that is, after the data of PWM8H is written you have to rewrite the same data. On the contrary, you will change the contents of PWM8L. you only write data to the PWM8L regardless of the contents of PWM8H register.

APPLICATION

The following show how the 7-bit PWM output is controlled in a active high.



Duty is determined by 7-bit data.
($40_H \times 8\mu S = 512\mu S$)

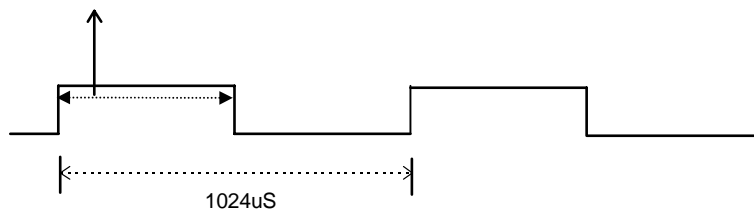


Fig 3. The example of output wave for 7-bit PWM

- ◆ Such as this example, the resolution of $8\mu s/1024\mu s$ is enable to output.
- ◆ The port connected 7-bit PWM is made of NMOS open drain, so you have to connect pull-up resistance. (The value of resistance is decided by input characteristic having external device)
- ◆ PWM data register is only enable to write, it(PWM0 to PWM7) operates independently about every bit.

3. Software Description

3-1. Basic Operation

TIMER0 Interrupt is made to occur every 2 second. For making period having 2 second, we selected 16bit PWM mode, supplied source clock to $32\mu s$, wrote F4H to TIMER1 and wrote 24H to TIMER0. In conclusion we make interrupt having 2 second period.

$$F424_H \times 32\mu S = 2,000,000\mu S = 2Sec$$

Also, we change the value of PWMCnt register with decreasing SecCnt register every 10 in this interrupt routine. Finally, we change PWMCnt register every 20 seconds and then product PWM data regard to the value of PWMCnt varied to Main Routine

- ◆ PWM output data is produced every 20 second.(100%-75%-50%-25%-100%) (Reference is defined to 100%)
To make PWM output we don't used original 128 steps but used the table having 100 steps made by us and then we wrote the value to PWM data register

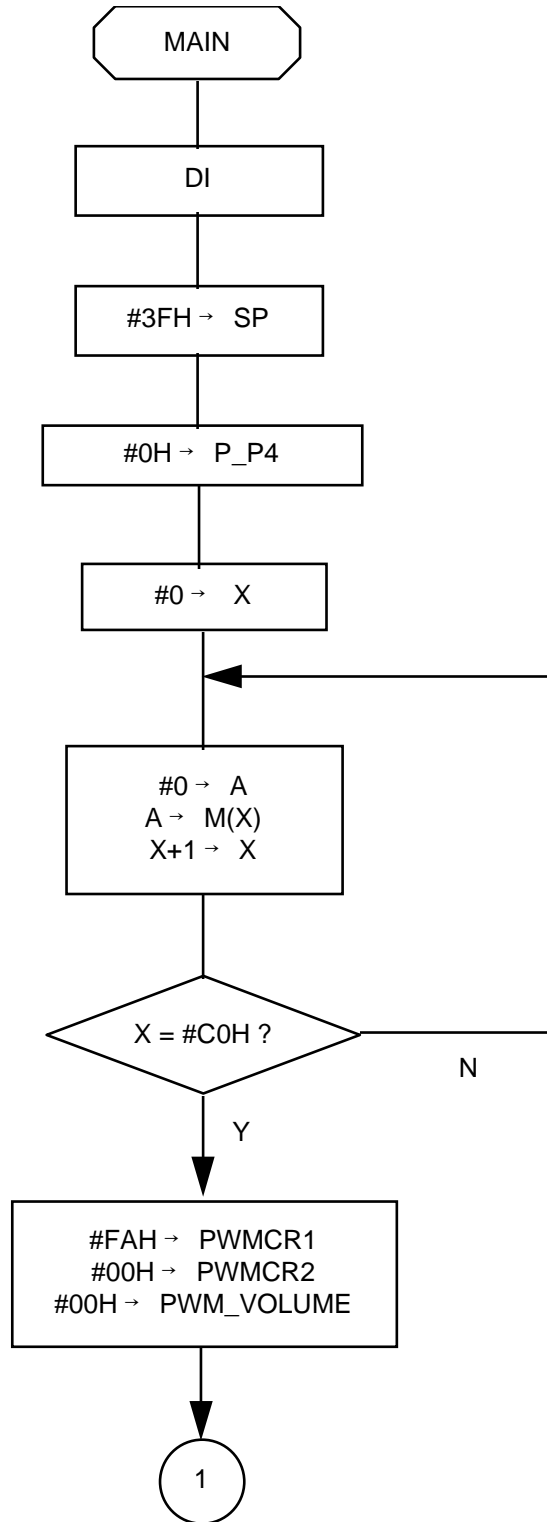
3-2. RAM Description

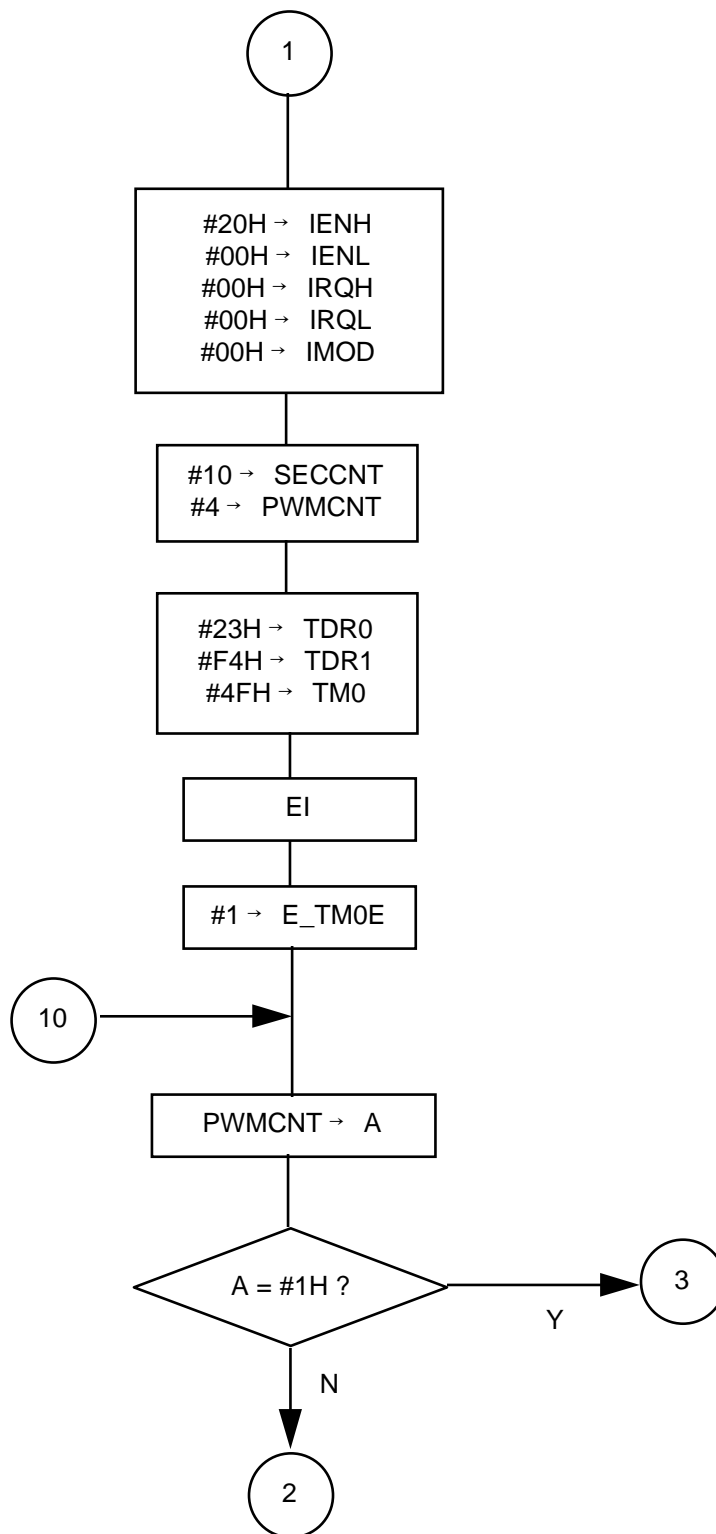
NAME	DESCRIPTION
Contrast	PWM DATA (Contrast)
Bright	PWM DATA (Brightness)
Color	PWM DATA (Color)
Tint	PWM DATA (Tint)
Sharp	PWM DATA (Sharpness)
SecCnt	Count a period having 20sec
PWMCnt	Change the value of PWM every 20sec
F_PWM	The flag for changing the value of PWM

Table 2. RAM DESCRIPTION

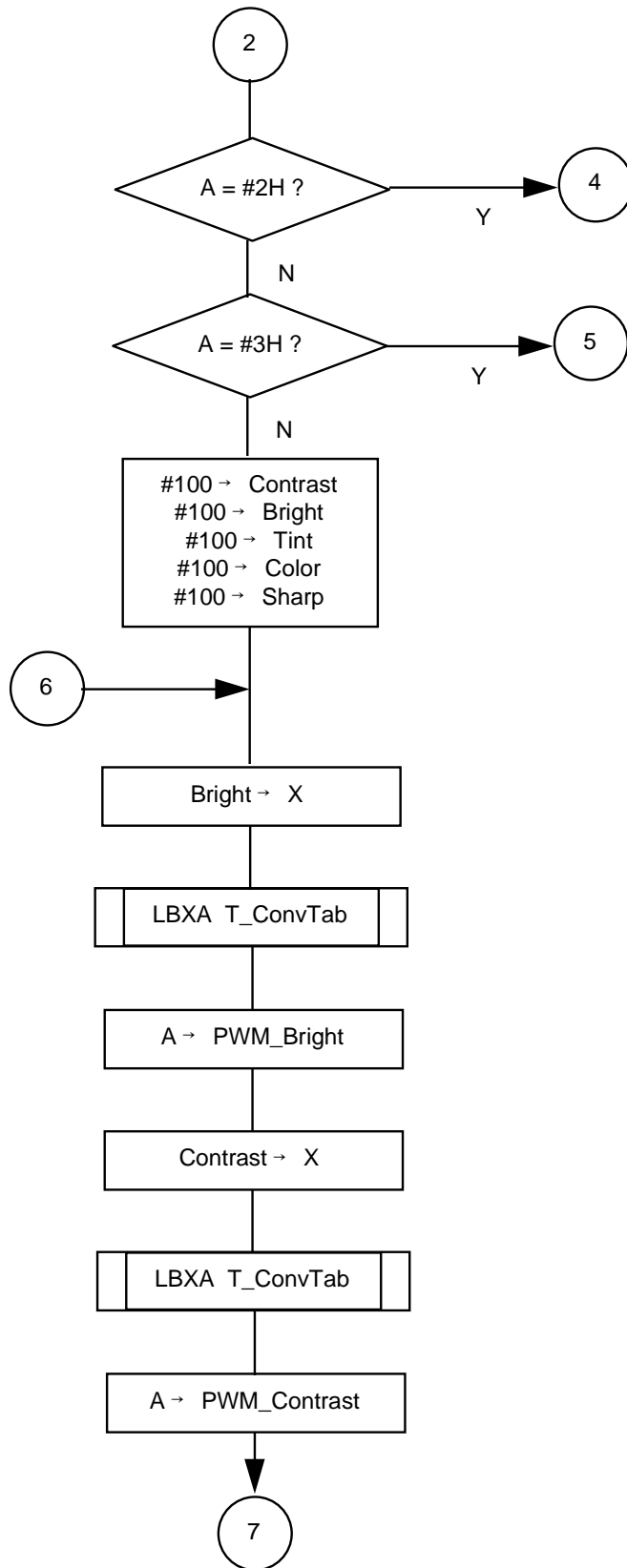
APPLICATION

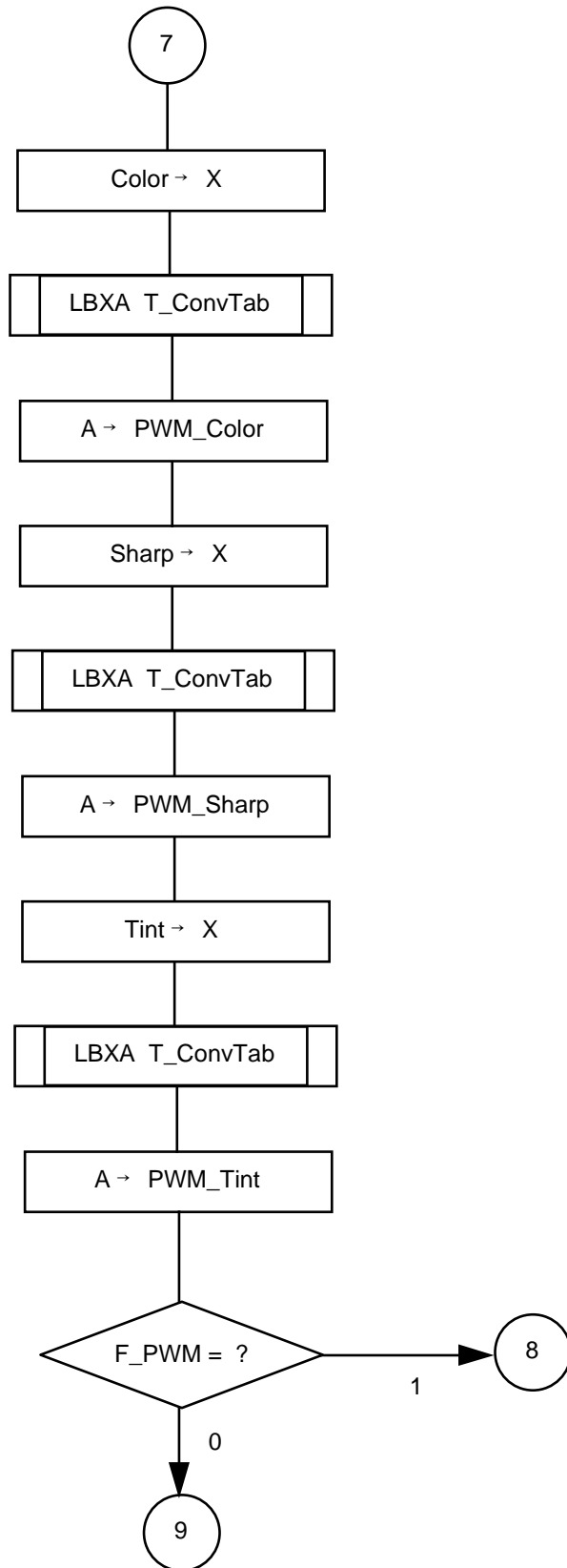
3-3.Flow chart



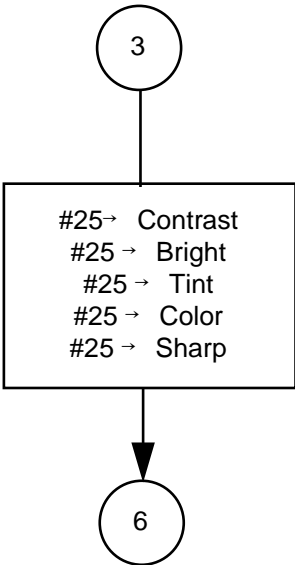
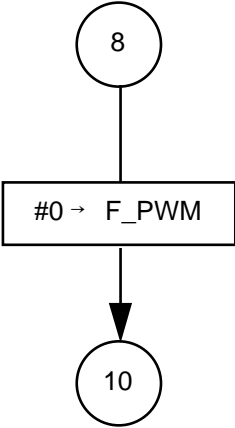
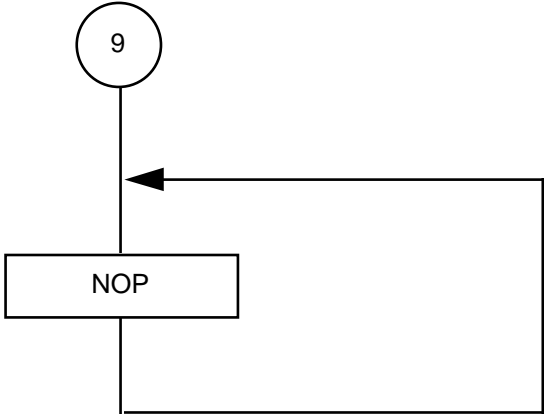


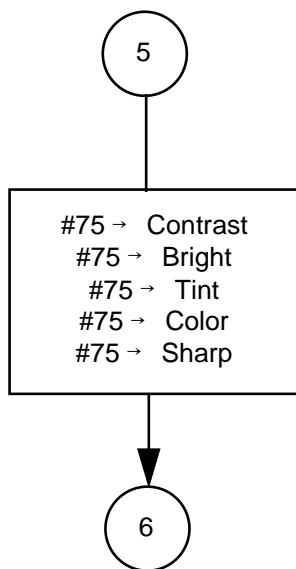
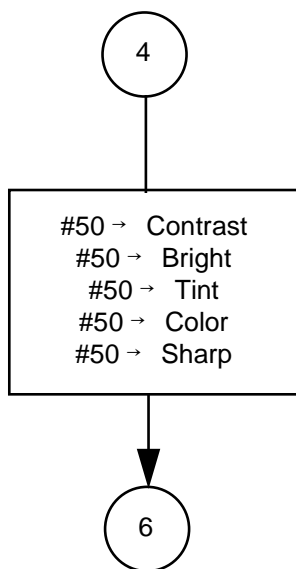
APPLICATION



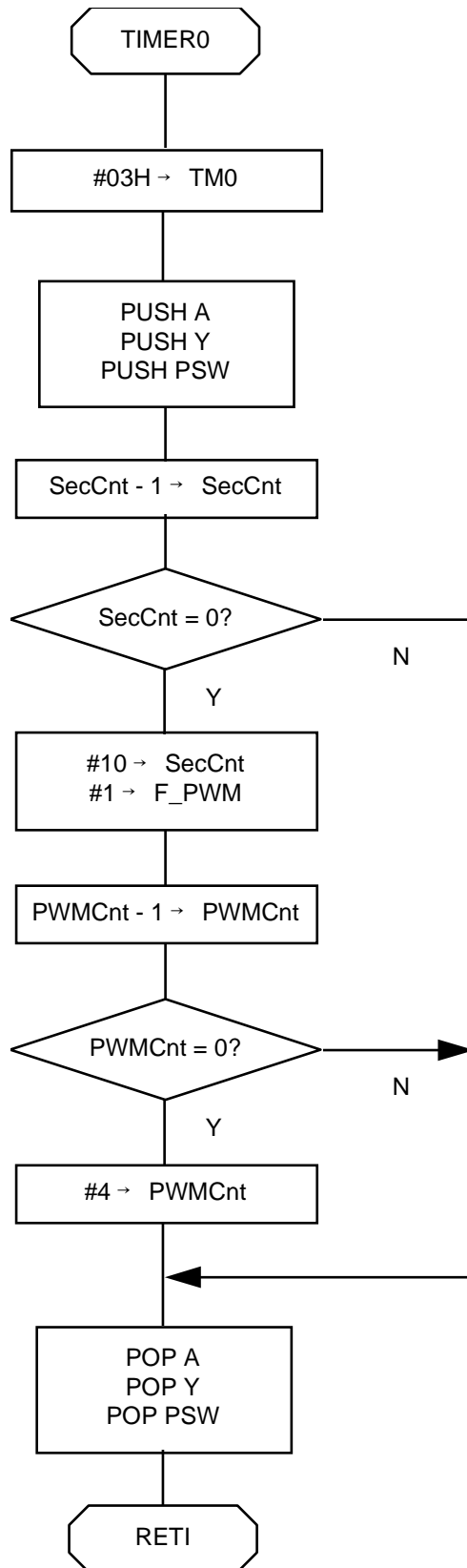


APPLICATION





APPLICATION



3-4. Source List

```

1      ;-----
2      lboxa  MACRO          ; Load From Table by X-indexing
3          xyx              ; Change X <-> Y
4          lda    !\1+y
5          xyx
6      ENDM
7      org 00h
8      ;-----
9      Contrast      ds  1
10     Bright        ds  1
11     Color          ds  1
12     Tint           ds  1
13     Sharp          ds  1
14     SecCnt         ds  1
15     PWMCnt         ds  1
16     F_PWM          equ 0, 08H
17     ;
18     P_P4           equ 0C8H
19     TM0            equ 0D0H    ; Timer 0, 1 Control
20     TDR0           equ 0D2H    ; Timer 0 counter
21     TDR1           equ 0D3H    ; Timer 1 counter
22     ;-----
23     ; * PWM Control Register
24     ;-----
25     PWM0           equ 0DAH    ; PWM 0 Data
26     PWM1           equ 0DBH    ; PWM 1 Data
27     PWM2           equ 0DCH    ; PWM 2 Data
28     PWM3           equ 0DDH    ; PWM 3 Data
29     PWM4           equ 0DEH    ; PWM 4 Data
30     PWM5           equ 0DFH    ; PWM 5 Data
31     PWM6           equ 0E0H    ; PWM 6 Data

```

APPLICATION

```
32     PWM7           equ    0E1H       ; PWM 7 Data
33     PWM8H          equ    0E2H       ; PWM 8 Data (High)
34     PWM8L          equ    0E3H       ; PWM 8 Data (Low)
35     PWMCR1         equ    0E4H       ; PWM Control Register 1
36     PWMCR2         equ    0E5H       ; PWM Control Register 2
37     ;
38     PWM_Contrast   equ    PWM3       ; Contrast
39     PWM_Bright     equ    PWM2       ; Brightness
40     PWM_Color      equ    PWM0       ; Color
41     PWM_Tint       equ    PWM1       ; Tint
42     PWM_Sharp      equ    PWM4       ; Sharpness
43     PWM_Volume     equ    PWM5       ; Treble
44     ;-----
45     ; * Interrupt Control Register
46     ;-----
47     IMOD           equ    0E6H       ; Interrupt Mode register
48     IENL           equ    0E8H       ; Interrupt Enable Register Low
49     IRQL           equ    0E9H       ; Interrupt Request Register Low
50     IENH           equ    0EAH       ; Interrupt Enable Register Low
51     E_TM0E        equ    4,IENH
52     IRQH           equ    0EBH       ; Interrupt Request Register Low
53
54     ;-----
55     org 0d000h
56     ;
57     D000 60      Main:  di
58     ;
59     D001 1E3F    idx    #3Fh        ; Stack Pointer
60     D003 8E      txsp
61     ;
62     ; * Port 4 is only output.
63     D004 E400C8  ldm    P_P4 ,#0000000B
64     ;
65     ; * RAM Clear (00.. BFh)
```

```
66      ; -----
67  D007 1E00      ldx    #0
68  D009 C400      JM_RAMClr: lda  #0
69  D00B FB        sta    {x}+
70  D00C 5EC0      cmpx   #0C0h
71  D00E 70F9      bne    JM_RAMClr
72      ;
73      ; * PWM Control Register
74      ; -----
75  D010 E4FCE4      ldm    PWMCR1,#11111100B ; PWM 0,1,2,3,4,5,8 & oscillation Start
76  D013 E400E5      ldm    PWMCR2,#00000000B ; 8Bit
77  D016 E400DF      ldm    PWM_VOLUME,#0 ; pwm data = 0
78      ;
79      ; * Interrupt Control Initialize
80      ; -----
81  D019 E410EA      ldm    IENH,#00010000B ; Timer0(TM0)
82  D01C E400E8      ldm    IENL,#00000000B
83  D01F E400EB      ldm    IRQH,#00000000B ; Clear All Interrupt Request
84  D022 E400E9      ldm    IRQL,#00000000B
85  D025 E400E6      ldm    IMOD,#00000000B ; Interrpt Accept Mode = refer to H/W
86      ;
87  D028 E40A05      ldm    SECCNT,#10 ; 10sec counter
88  D02B E40406      ldm    PWMCNT,#4 ;
89      ; * Timer Initialize
90      ; -----
91  D02E E423D2      ldm    TDR0,#23H ; 32uS * (F423h+1) = 2sec
92  D031 E4F4D3      ldm    TDR1,#0F4H ;
93      ;
94  D034 E44FD0      ldm    TM0,#01001111B ; TM0 = 32uS, 16 Bit Mode
95  D037 E0          ei
96  D038 81EA        set1   E_TM0E
97      ;
98      ; -----
99  PWMOUT:
```

APPLICATION

```
100 D03A C506     lda     PWMCnt
101           SWITCH a
102 D03C 2F46     @       bra     @__1
103           CASE     4
104           @ @__3:
105           [Contrast] = 100     ; Contrast <---100%
106 D03E E46400 @       ldm     CONTRAST,#100
107           [Bright] =    100     ; Brightness <---100%
108 D041 E46401 @       ldm     BRIGHT,#100
109           [Tint]  =    100     ; Tint <---100%
110 D044 E46403 @       ldm     TINT,#100
111           [Color] =    100     ; Color <---100%
112 D047 E46402 @       ldm     COLOR,#100
113           [Sharp] =    100     ; Sharpness <---100%
114 D04A E46404 @       ldm     SHARP,#100
115           BREAK
116 D04D 2F45     @       bra     @__2
117           ;
118           CASE     3
119           @ @__4:
120           [Contrast] = 75       ; Contrast <--- 75%
121 D04F E44B00 @       ldm     CONTRAST,#75
122           [Bright] =    75       ; Brightness <--- 75%
123 D052 E44B01 @       ldm     BRIGHT,#75
124           [Tint]  =    75       ; Tint <--- 75%
125 D055 E44B03 @       ldm     TINT,#75
126           [Color] =    75       ; Color <--- 75%
127 D058 E44B02 @       ldm     COLOR,#75
128           [Sharp] =    75       ; Sharpness <--- 75%
129 D05B E44B04 @       ldm     SHARP,#75
130           BREAK
131 D05E 2F34     @       bra     @__2
132           ;
133           CASE     2
```

```

134     @ __5:
135     [Contrast] = 50 ; Contrast <--- 50%
136 D060 E43200 @     Idm    CONTRAST,#50
137     [Bright] = 50 ; Brightness <--- 50%
138 D063 E43201 @     Idm    BRIGHT,#50
139     [Tint] = 50 ; Tint <--- 50%
140 D066 E43203 @     Idm    TINT,#50
141     [Color] = 50 ; Color <--- 50%
142 D069 E43202 @     Idm    COLOR,#50
143     [Sharp] = 50 ; Sharpness <--- 50%
144 D06C E43204 @     Idm    SHARP,#50
145     BREAK
146 D06F 2F23 @     bra    __2
147     ;
148     CASE 1
149     @ __6:
150     [Contrast] = 25 ; Contrast <--- 25%
151 D071 E41900 @     Idm    CONTRAST,#25
152     [Bright] = 25 ; Brightness <--- 25%
153 D074 E41901 @     Idm    BRIGHT,#25
154     [Tint] = 25 ; Tint <--- 25%
155 D077 E41903 @     Idm    TINT,#25
156     [Color] = 25 ; Color <--- 25%
157 D07A E41902 @     Idm    COLOR,#25
158     [Sharp] = 25 ; Sharpness <--- 25%
159 D07D E41904 @     Idm    SHARP,#25
160     BREAK
161 D080 2F12 @     bra    __2
162     ;
163     ENDS
164 D082 2F10 @     bra    __2
165     @ __1:
166 D084 4401 @     cmp    #1
167 D086 F0E9 @     beq    __6

```

APPLICATION

```
168 D088 4402 @      cmp    #2
169 D08A F0D4 @      beq    @__5
170 D08C 4403 @      cmp    #3
171 D08E F0BF @      beq    @__4
172 D090 4404 @      cmp    #4
173 D092 F0AA @      beq    @__3
174     @ @__2:
175     ; * Bright output
176 D094 CC01     idx    Bright
177     lbra T_ConvTab     ; Data Load at Table
178 D096 FE      @      xyx
179 D097 D5EAD0 @      lda    !T_ConvTab+y
180 D09A FE      @      xyx
181 D09B E5DC     sta    PWM_Bright
182     ; * Contrast output
183 D09D CC00     idx    Contrast
184     lbra T_ConvTab
185 D09F FE      @      xyx
186 D0A0 D5EAD0 @      lda    !T_ConvTab+y
187 D0A3 FE      @      xyx
188 D0A4 E5DD     sta    PWM_Contrast
189     ; * Color output
190 D0A6 CC02     idx    Color
191     lbra T_ConvTab
192 D0A8 FE      @      xyx
193 D0A9 D5EAD0 @      lda    !T_ConvTab+y
194 D0AC FE      @      xyx
195 D0AD E5DA     sta    PWM_Color
196     ; * Sharpness output
197 D0AF CC04     idx    Sharp
198     lbra T_ConvTab
199 D0B1 FE      @      xyx
200 D0B2 D5EAD0 @      lda    !T_ConvTab+y
201 D0B5 FE      @      xyx
```

```
202 D0B6 E5DE      sta    PWM_Sharp
203      ; * Tint output
204 D0B8 CC03      ldx    Tint
205      lbrx  T_ConvTab
206 D0BA FE        @      xyx
207 D0BB D5EAD0    @      lda    !T_ConvTab+y
208 D0BE FE        @      xyx
209 D0BF E5DB      sta    PWM_Tint
210      ;
211 D0C1 030804    bbs    F_PWM, OUT
212      ;
213      Loop:
214 D0C4 FF        nop
215 D0C5 1BC4D0    jmp    Loop
216      ;
217      OUT:
218 D0C8 1108      clr1   F_PWM
219 D0CA 1B3AD0    jmp    PWMOUT
220      TIMER0:
221 D0CD E403D0    ldm    TM0,#03H
222 D0D0 0E        push   a
223 D0D1 4E        push   y
224 D0D2 6E        push   psw
225 D0D3 A905      dec    SecCnt
226 D0D5 700C      bne    IntRet
227 D0D7 E40A05    ldm    SECCNT,#10
228 D0DA 0108      set1   F_PWM
229 D0DC A906      dec    PWMCnt
230 D0DE 7003      bne    IntRet
231 D0E0 E40406    ldm    PWMCNT,#4
232      IntRet:
233 D0E3 0D        pop    a
234 D0E4 4D        pop    y
235 D0E5 6D        pop    psw
```

APPLICATION

```
236 D0E6 E44FD0      ldm    TM0,#4FH
237
238      Return:
239 D0E9 7F          reti
240      ;-----
241      ; * PWM data based analog value
242      ;-----
243 D0EA 00          T_ConvTab: db  00      ; ANALOG DATA = 0
244 D0EB 01          db    01, 02, 03, 04, 06  ; 01<->05
245 D0EC 02
246 D0ED 03
247 D0EE 04
248 D0EF 06
249 D0F0 07          db    07, 08, 10, 11, 12  ; 06<->10
250 D0F1 08
251 D0F2 0A
252 D0F3 0B
253 D0F4 0C
254 D0F5 0E          db    14, 15, 16, 17, 18  ; 11<->15
255 D0F6 0F
256 D0F7 10
257 D0F8 11
258 D0F9 12
259 D0FA 14          db    20, 21, 22, 24, 25  ; 16<->20
260 D0FB 15
261 D0FC 16
262 D0FD 18
263 D0FE 19
264 D0FF 1A          db    26, 27, 28, 30, 31  ; 21<->25
265 D100 1B
266 D101 1C
267 D102 1E
268 D103 1F
269 D104 20          db    32, 33, 34, 36, 38  ; 26<->30
```

D105 21
D106 22
D107 24
D108 26
250 D109 27 db 39, 40, 41, 42, 44 ; 31<->35
D10A 28
D10B 29
D10C 2A
D10D 2C
251 D10E 2D db 45, 46, 48, 49, 50 ; 36<->40
D10F 2E
D110 30
D111 31
D112 32
252 D113 33 db 51, 52, 54, 55, 56 ; 41<->45
D114 34
D115 36
D116 37
D117 38
253 D118 3A db 58, 59, 60, 62, 63 ; 46<->50
D119 3B
D11A 3C
D11B 3E
D11C 3F
254 ;
255 D11D 40 db 64, 65, 66, 68, 69 ; 51<->55
D11E 41
D11F 42
D120 44
D121 45
256 D122 46 db 70, 72, 73, 74, 75 ; 56<->60
D123 48
D124 49
D125 4A

APPLICATION

	D126 4B			
257	D127 4C	db	76, 78, 79, 80, 82	; 61<->65
	D128 4E			
	D129 4F			
	D12A 50			
	D12B 52			
258	D12C 53	db	83, 84, 85, 86, 88	; 66<->70
	D12D 54			
	D12E 55			
	D12F 56			
	D130 58			
259	D131 59	db	89, 90, 92, 93, 94	; 71<->75
	D132 5A			
	D133 5C			
	D134 5D			
	D135 5E			
260	D136 60	db	96, 98, 99, 100, 101	; 76<->80
	D137 62			
	D138 63			
	D139 64			
	D13A 65			
261	D13B 66	db	102, 103, 104, 106, 108	; 81<->85
	D13C 67			
	D13D 68			
	D13E 6A			
	D13F 6C			
262	D140 6D	db	109, 110, 111, 112, 113	; 86<->90
	D141 6E			
	D142 6F			
	D143 70			
	D144 71			
263	D145 72	db	114, 115, 118, 119, 121	; 90<->95
	D146 73			
	D147 76			

```
D148 77
D149 79
264 D14A 7B      db      123,124,125,126,127  ; 96<->100
D14B 7C
D14C 7D
D14D 7E
D14E 7F
265      ;
266      org 0ffe2h
267      ;
268 FFE2 E9D0    dw      Return  ; Serial I/O Interrupt ffe2
269 FFE4 E9D0    dw      Return  ; Basic Interval Timer   e4
270 FFE6 E9D0    dw      Return  ; WatchDog Timer       e6
271 FFE8 E9D0    dw      Return  ; External Interrupt 3   e8
272 FFEA E9D0    dw      Return  ; Timer 3                ea
273 FFEC E9D0    dw      Return  ; Timer 1 (250ms Interrupt)  ec
274 FFEE E9D0    dw      Return  ; V-Sync Interrupt         ee
275 FFF0 E9D0    dw      Return  ; 1 mS Interrupt          f0
276 FFF2 E9D0    dw      Return  ; Timer 2                f2
277 FFF4 CDD0    dw      TIMER0   ; Timer 0                f4
278 FFF6 E9D0    dw      Return  ; External Interrupt 2 (Remocon)f6
279 FFF8 E9D0    dw      Return  ; External Interrupt 1   f8
280 FFFA E9D0    dw      Return  ; Multi-line OSD         fa
281 FFFC E9D0    dw      Return  ;      Undefined         fc
282 FFFE 00D0    dw      MAIN    ; Reset (NMI)           fe
283      end
284      ;
```

-- 0 Error(s) --

--- Total Machine Code : 365 Bytes --