PRODUCT NAME

TM58XX

TITLE

the method of TM58XX drives the LCD with HT1621

APPLICATION NOTE

1. explanation
2. program
3. LCD Layout
4. application circuit diagram

1. Initial is deficient to HT1621：

Mainly is to the HT1621 frequency of use, the audio frequency choice, whether turns on the buzzer and so on to carry on the choice

First lets enable 片选 the position (CS\=0), ID is 4 is the form order is 100.

Then delivers the corresponding order character, after delivers sets the CS\ foot 1, namely not to HT1621 operation when CSP is 1.

。

2. Clear screen：

The HT1621 interior has 32*4 the position the RAM unit, also may say outputs the buffer, its is clear 0, namely LCD did not demonstrate.

3. Output：

Demo board use LCD is five demonstrations positions, this procedure realizes latter two fixedly to output dE, first three output 000-->111...-->FFF rear drive condition engine off

When export data,first set Sets at slice selection position to be effective,then send Id as 5(101),namely order format is writing operation. Following,send 欲显段地址 to 1621,give the gonna data to1621 output.Finally,close the 片选位,namely set the CS foot to 1.

Other explanation：

(1) To 1621 writes the operation question：

When operate to 1621,first sets at 片选 the position to be effective（CSP＝0），and when not operate to 1621,sets at 片选位 invalid（CSP＝1）。

(2) Attention address, data or order length question。

```
;----------------------------
;Title:          Write HT1621
;Function:
;Note:         The voltage applied to Vlcd pin must be lower than Vdd.
;Clock:         4m
;Config word: xt_osc & _wdt_off & _cpt_on & _type_general & _lv_don't use
;Data:         2004_5_9


;        list   p=tm58p20
;----------------------------
indf                equ     00h
tmr0                equ     01h
pc                  equ     02h
status              equ     03h
fsr                 equ     04h
porta               equ     05h
portb               equ     06h
csp                 equ     4
wrp                 equ     5
dtp                 equ     6
portc               equ     07h
wakeup              equ     20h
irqm                equ     21h
irqf                equ     22h
;----------------------------
;status  bits define
c                   equ     00h
dc                  equ     01h
z                   equ     02h
pd                  equ     03h
to                  equ     04h
pa0                 equ     05h
pa1                 equ     06h
pa2                 equ     07h
;----------------------------
;option bits define
ps0                 equ     00h
ps1                 equ     01h
ps2                 equ     02h
psa                 equ     03h
rte                 equ     04h
rts                 equ     05h
;----------------------------
cnt                 equ     08h
```

```
id              equ     09h
temp            equ     0ah
cmd             equ     0bh
temp1           equ     0ch
addr            equ     0dh
dat             equ     0eh
temper          equ     0fh


        org     7ffh
        lgoto   main
        org     00h
        nop
;--------------------------
lcd_tableh
      addam     pc,1
      retla     b'00000101'   ;0
      retla     b'00000000'   ;1
      retla     b'00000110'   ;2
      retla     b'00000010'   ;3
      retla     b'00000011'   ;4
      retla     b'00000011'   ;5
      retla     b'00000111'   ;6
      retla     b'00000000'   ;7
      retla     b'00000111'   ;8
      retla     b'00000011'   ;9
      retla     b'00000111'   ;a
      retla     b'00000111'   ;b
      retla     b'00000110'   ;c
      retla     b'00000110'   ;d
      retla     b'00000111'   ;e
      retla     b'00000111'   ;f
      retla     b'00000111'   ;f

lcd_tablel
      addam     pc,1
      retla     b'00001111'   ;0
      retla     b'00000110'   ;1
      retla     b'00001011'   ;2
      retla     b'00001111'   ;3
      retla     b'00000110'   ;4
      retla     b'00001101'   ;5
      retla     b'00001101'   ;6
      retla     b'00000111'   ;7
      retla     b'00001111'   ;8
      retla     b'00001111'   ;9
      retla     b'00000111'   ;a
      retla     b'00001100'   ;b
      retla     b'00001000'   ;c
```

```
        retla   b'00001110'   ;d
        retla   b'00001001'   ;e
        retla   b'00000001'   ;f
        retla   b'00000001'   ;f

;---------------------------
dispwriteid_:
        movla   3
        movam   cnt
        bcm     status,c
        rlm     id,m
        rlm     id,m
        rlm     id,m
        rlm     id,m
        rlm     id,m
        movm    id,a
        movam   temp
        lcall   dispwritebits_
        ret
;---------------------------
dispwritecmd_:
        movla   8
        movam   cnt
        movm    cmd,a
        movam   temp
        lcall   dispwritebits_
        ret
;---------------------------
dispwritebits_:
        rlm     temp,m
        btmsc   status,c
        lgoto   highs
low:
        bcm     portb,dtp       ;write data '0'
        bcm     portb,wrp
        nop
        nop
        nop
        nop
        nop
        nop
        bsm     portb,wrp
        nop
        nop
        nop
        nop
        nop
        nop
        nop
```

```
        lgoto   loop1
highs:
        bsm     portb,dtp       ; write data '1'
        bcm     portb,wrp
        nop
        nop
        nop
        nop
        nop
        nop
        bsm     portb,wrp
        nop
        nop
        nop
        nop
        nop
        nop
loop1:
        decmsz  cnt,m
        lgoto   dispwritebits_
        ret
;--------------------------
dispwritebit_:
        movla   1
        movam   cnt
        movla   80h
        movam    temp
        lcall   dispwritebits_
        ret
;--------------------------
writeto46addr_:
        movla   7
        movam   cnt
        bcm     status,c
        rlm     addr,m
        movm    addr,a
        movam   temp
        lcall   dispwritebits_
        ret

dispwriteaddr_:
        movla   6
        movam   cnt
        bcm     status,c
        rlm     addr,m
        rlm     addr,m
        movm    addr,a
        movam   temp
        lcall   dispwritebits_
```

```
        ret
;--------------------------
dispwritedata_:
        movla   4
        movam   cnt
        movm    dat,a
        movam    temp
        lcall   dispwritebits1_
        ret
;--------------------------
;input: temp,cnt
;func:only for write dat low nibble
;-------------
dispwritebits1_:
loop00:
        rrm     temp,m
        btmsc   status,c
        lgoto   high11
low00:
        bcm     portb,dtp
        bcm     portb,wrp
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        bsm     portb,wrp
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        lgoto   loop10
high11:
        bsm     portb,dtp
        bcm     portb,wrp
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        bsm     portb,wrp
        nop
```

```
        nop
        nop
        nop
        nop
        nop
        nop
        nop
loop10:
        decmsz    cnt,m
        lgoto     loop00
        ret
;--------------------------
initdisp_:
;turn on lcd display    id:100
        bcm       portb,csp       ;使能 ht1621
        movla     4
        movam     id
        lcall     dispwriteid_

        movla     3
        movam     cmd             ;turn on lcd bias gererator
        lcall     dispwritecmd_
        lcall     dispwritebit_

        movla     b'00011000'     ;rc 256k
        movam     cmd
        lcall     dispwritecmd_
        lcall     dispwritebit_

        movla     b'00101001'     ;select bias ,duty
        movam     cmd              ;1/4duty,1/3bias
        lcall     dispwritecmd_
        lcall     dispwritebit_

        movla     b'11100011'     ;normal  mode
        movam     cmd
        lcall     dispwritecmd_
        lcall     dispwritebit_

        movla     1               ;turn sys osc on
        movam     cmd
        lcall     dispwritecmd_
        lcall     dispwritebit_

        movla     b'01100000'     ;set  buzzer 2khz
        movam     cmd
        lcall     dispwritecmd_
        lcall     dispwritebit_
```

```
        movla   b'00001000'   ;close buzzer
        movam   cmd
        lcall   dispwritecmd_
        lcall   dispwritebit_

        movla   b'00000101'   ;close watch dog
        movam   cmd
        lcall   dispwritecmd_
        lcall   dispwritebit_

        bsm     portb,csp     ; when not operate to HT1621,CSP is 1
        nop
        ret
;--------------------------
clearall_:
        movla   d'24'         ; clear the 24 units after address 08H
        movam   temp1         ; because the units before 08H is not use in the program,
                              ;so we can pay no attention.
        bcm     portb,csp
        movla   d'5'
        movam   id
        lcall   dispwriteid_  ;3 digit ID
        movla   d'8'
        movam   addr
        lcall   dispwriteaddr_; 6 digit address
bbb:
        clrm    dat
        lcall   dispwritedata_ ; 8 digit data
        decmsz  temp1,1
        lgoto   bbb
        bsm     portb,csp
        nop
        nop
        ret
;--------------------------
;seg10 seg9     display the first digit data
;seg8  seg13    display the second digit data
;seg17 seg21    display the third digit data
;seg25 seg29    display the forth digit data
;seg31 seg30    display the fifth digit data
; the data of the first three digit display from000 ,111,and finally to FFF.
; dE the behind two data is displayed fixedly.
output
seg10:
        movla   d'5'          ;D'5'  is 101,namely is writting order
        movam   id
        bcm     portb,csp     ;片选位 is efficiency
        lcall   dispwriteid_
        movla   d'10'
```

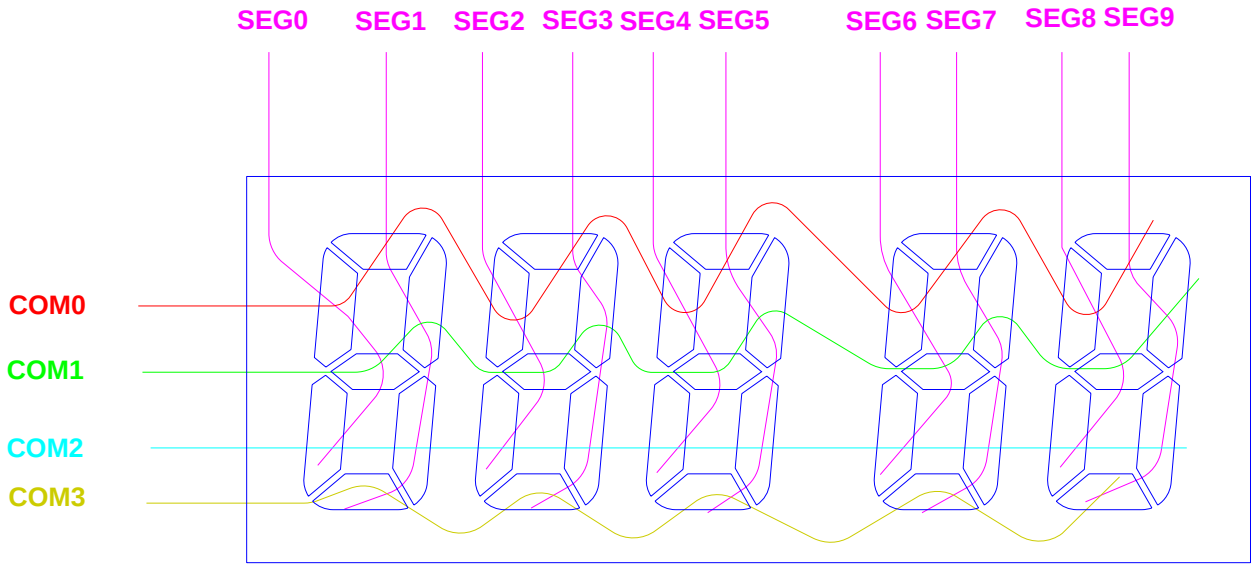
```
        movam    addr           ; send the address of 欲显段 to HT1621
        lcall    dispwriteaddr_
        movm     temp1,a
        lcall    lcd_tableh     ;将欲显数据高半位送 HT1621
        movam    dat            ; because the length of the HT1621 RAM is 4 digits,
        lcall    dispwritedata_ ; so send as 2 times.
        bsm      portb,csp
        nop
        nop
seg9:
        movla    d'5'
        movam    id
        bcm      portb,csp
        lcall    dispwriteid_
        movla    d'9'
        movam    addr
        lcall    dispwriteaddr_
        movm     temp1,a
        lcall    lcd_tablel     ;将欲显数据低半位送 HT1621
        movam    dat
        lcall    dispwritedata_
        bsm      portb,csp
        nop
        nop
seg8:
        movla    d'5'           ; The following shows spaces approach Ibid.
        movam    id
        bcm      portb,csp
        lcall    dispwriteid_
        movla    d'8'
        movam    addr
        lcall    dispwriteaddr_
        movm     temp1,a
        lcall    lcd_tableh
        movam    dat
        lcall    dispwritedata_
        bsm      portb,csp
        nop
        nop
seg13:
        movla    d'5'
        movam    id
        bcm      portb,csp
        lcall    dispwriteid_
        movla    d'13'
        movam    addr
        lcall    dispwriteaddr_
        movm     temp1,a
```

```
        lcall       lcd_tablel
        movam       dat
        lcall       dispwritedata_
        bsm         portb,csp
        nop
        nop
seg17:
        movla       d'5'
        movam       id
        bcm         portb,csp
        lcall       dispwriteid_
        movla       d'17'
        movam       addr
        lcall       dispwriteaddr_
        movm        temp1,a
        lcall       lcd_tableh
        movam       dat
        lcall       dispwritedata_
        bsm         portb,csp
        nop
        nop
seg21:
        movla       '5'
        movam       d
        bcm         ortb,csp
        lcall       dispwriteid_
        movla       '21'
        movam       ddr
        lcall       dispwriteaddr_
        movm        emp1,a
        lcall       cd_tablel
        movam       at
        lcall       dispwritedata_
        bsm         ortb,csp
        nop
        nop
seg25:
        movla       '5'
        movam       d
        bcm         ortb,csp
        lcall       dispwriteid_
        movla       '25'
        movam       ddr
        lcall       dispwriteaddr_
        movla       dh
        lcall       cd_tableh
        movam       at
        lcall       dispwritedata_
```

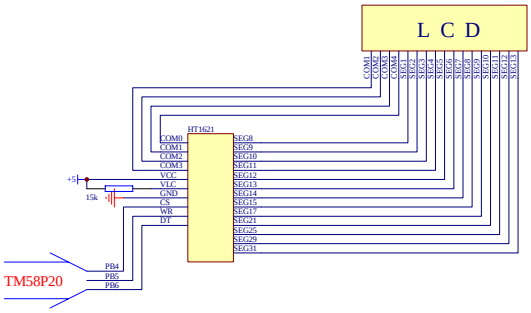
```
        bsm     ortb,csp
        nop
        nop
seg29:
        movla   '5'
        movam   d
        bcm     ortb,csp
        lcall   dispwriteid_
        movla   '29'
        movam   ddr
        lcall   dispwriteaddr_
        movla   dh
        lcall   cd_tablel
        movam   at
        lcall   dispwritedata_
        bsm     ortb,csp
        nop
        nop
seg31:
        movla   '5'
        movam   d
        bcm     ortb,csp
        lcall   dispwriteid_
        movla   '31'
        movam   ddr
        lcall   dispwriteaddr_
        movla   eh
        lcall   cd_tableh
        movam   at
        lcall   dispwritedata_
        bsm     ortb,csp
        nop
        nop
seg30:
        movla   '5'
        movam   d
        bcm     ortb,csp
        lcall   dispwriteid_
        movla   '30'
        movam   ddr
        lcall   dispwriteaddr_
        movla   eh
        lcall   cd_tablel
        movam   at
        lcall   dispwritedata_
        bsm     ortb,csp
        bcm     ortb,wrp
        bcm     ortb,dtp
```

```
        ret
;--------------------------
main:
        clrm        orta            ;I/O □ initialization
        movla       0h
        iodir       rta

        clrm        ortb
        movla       0h
        iodir       ortb

        clrm        ortc
        movla       0h
        iodir       ortc

        clrm        mr0
        movla       7h
        select
start:
        lcall       initdisp_       ; initialize1621
        lcall       clearall_       ; clear screen

        clrm        temp1
        lp0
        lcall       output          ; after export 000,111,and to FFFF, dynamic engine off
        incm        temp1,  1
        movla       10h
        subam       temp1,  0
        btmss       status, c
        goto        lp0
aaa:
        lgoto       aaa
;--------------------------
        end
;--------------------------
```

LCD Layout


SEG0
SEG1
SEG2
SEG3
SEG4
SEG5
SEG6
SEG7
SEG8
SEG9
COM0
COM1
COM2
COM3

application circuit diagram:



tenx	Title:		
	Size: B	Number:	Revision:
	Date:		Sheet of
	File:		Drawn by: