Application Note

HANTRONIX

Crystal Clear and Visibly Superior LCD Modules

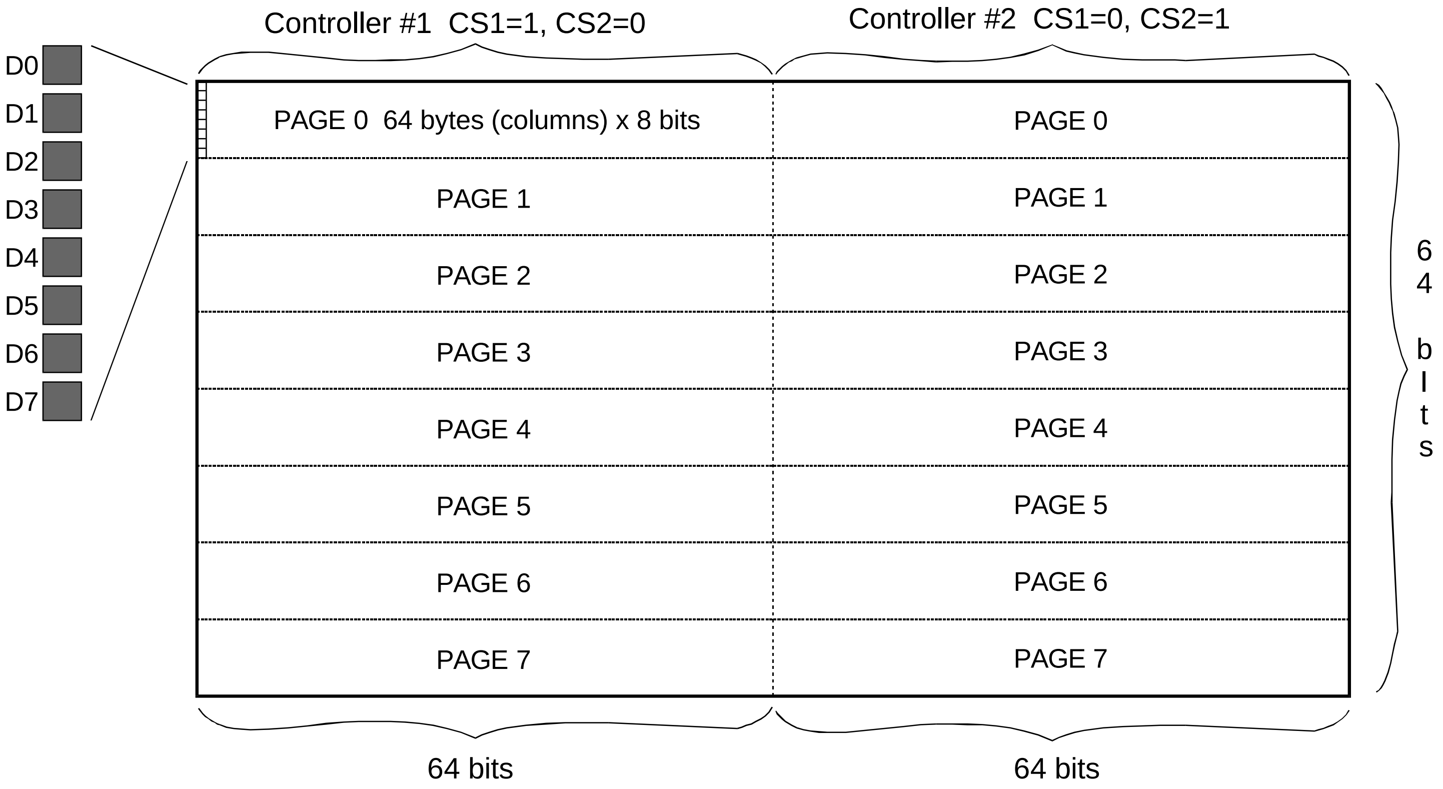
Interfacing a Hantronix 128x64 Graphic Module to an 8-bit Microcontroller

Introduction:

Due to its thin profile, light weight, low power consumption and easy handling, liquid crystal graphic display modules are used in a wide variety of applications. This note details a simple interface technique between a Hantronix HDM64GS12 and a micro-controller. The HDM64GS12 has a built-in Hitachi HD61202, or Samsung KS107, controller which performs all of the refreshing and data storage tasks of the LCD display. This note applies to any display using these controllers. The driving micro-controller is the popular 87C751.

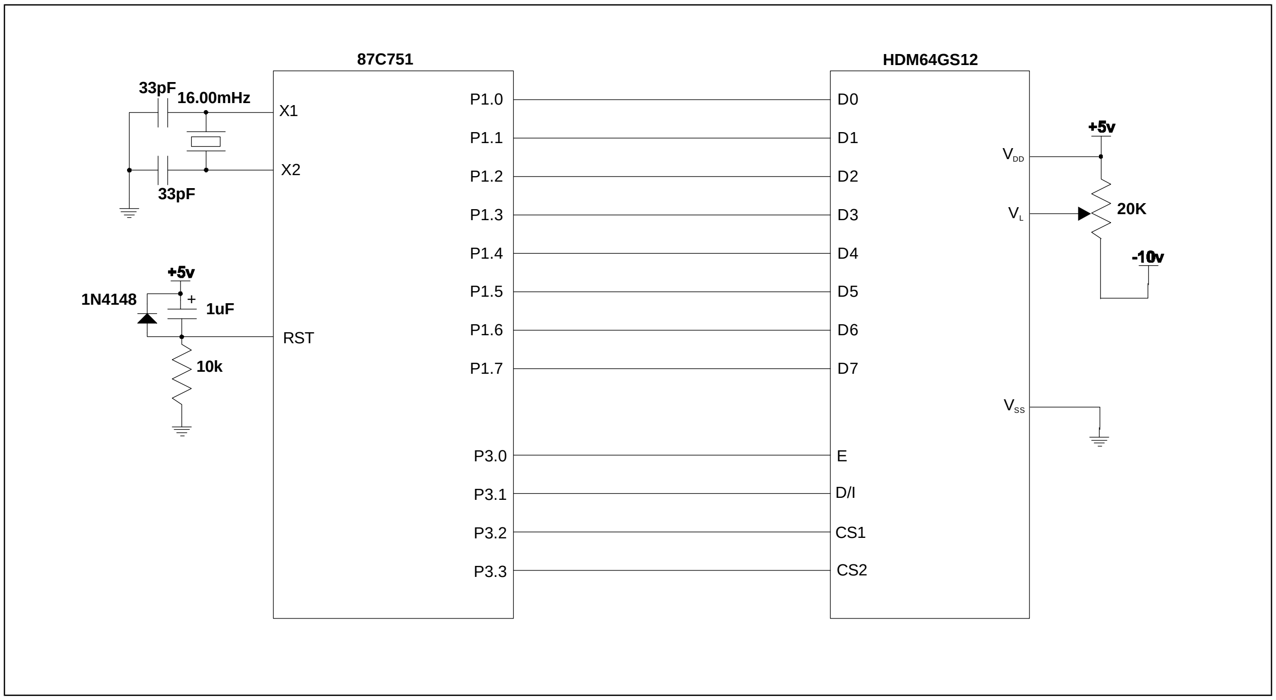
The display is split logically in half. It contains two controllers with controller #1 (Chip select 1) controlling the left half of the display and controller #2 (Chip select 2) controlling the right half. Each controller must be addressed independently.

The page addresses, 0-7, specify one of the 8 horizontal pages which are 8 bits (1 byte) high. A drawing of the display and how it is mapped to the refresh memory is shown below.



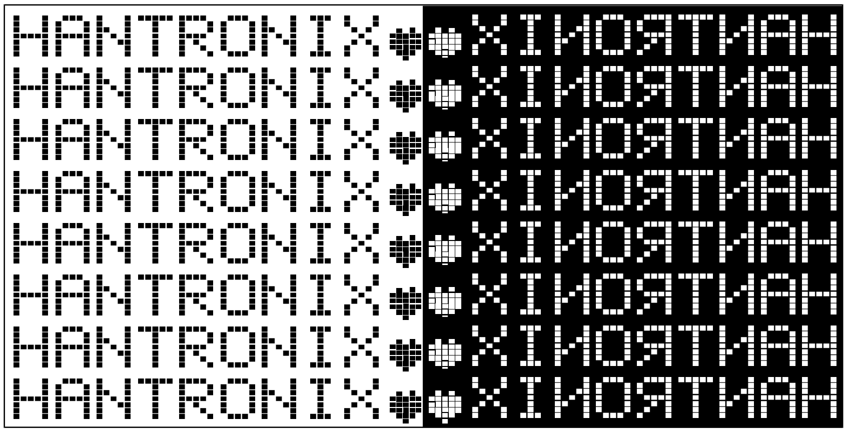
The schematic on page two is a simple circuit to illustrate one possible interface scheme. This is the circuit that the code example will work with directly.

Schematic Diagram:

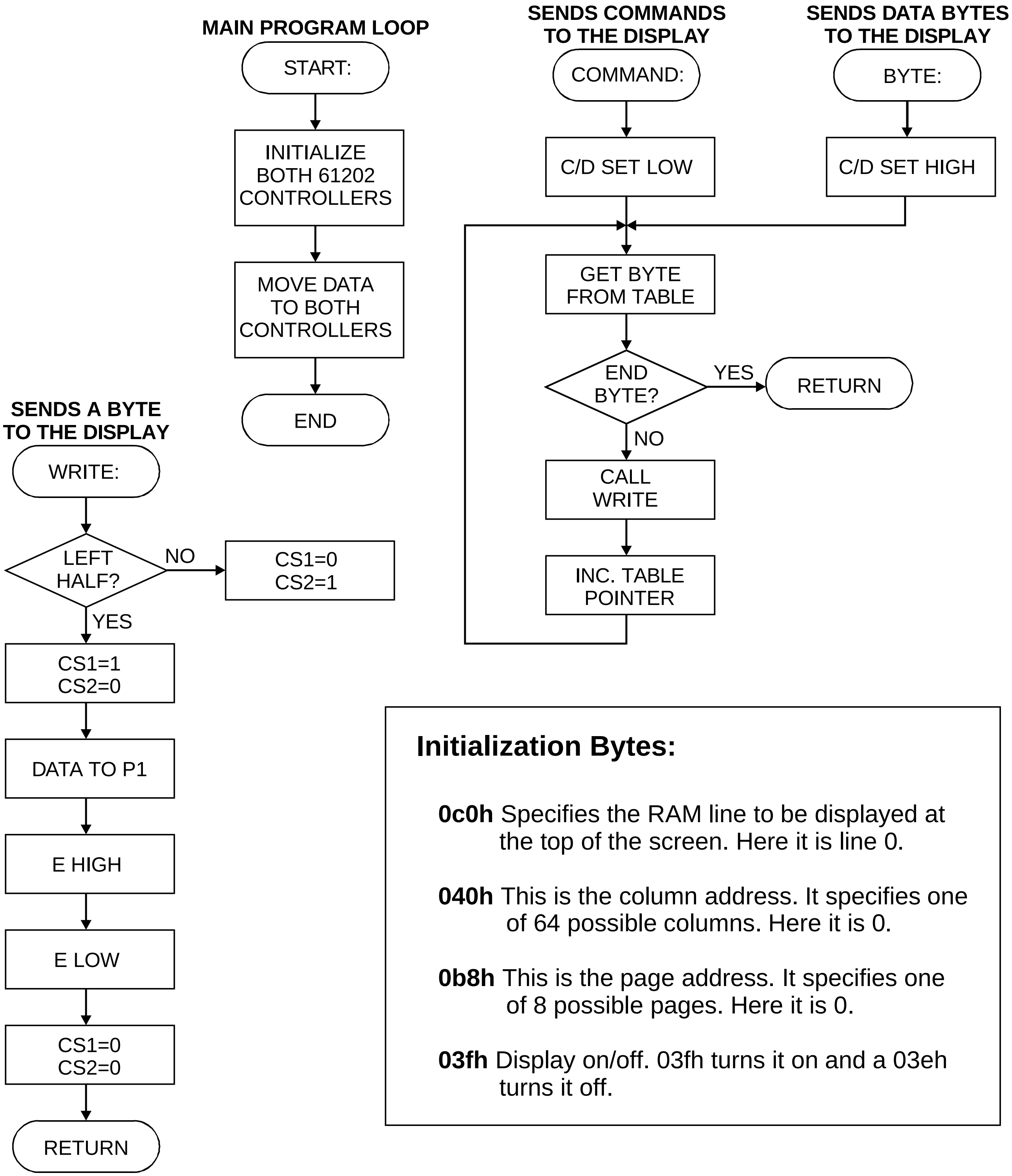


The following software is in 8051 assembly language and will run as-is on the hardware shown above. The busy status flag is not tested in this software. It is usually not necessary to do so when the display module is connected to the processor via I/O lines. When the module is connected to the processor's data bus and mapped into it's memory area the status should be tested to guarantee reliable service.

Displayed Pattern:



Software Flowchart:



Initialization Bytes:

0c0h Specifies the RAM line to be displayed at the top of the screen. Here it is line 0.

040h This is the column address. It specifies one of 64 possible columns. Here it is 0.

0b8h This is the page address. It specifies one of 8 possible pages. Here it is 0.

03fh Display on/off. 03fh turns it on and a 03eh turns it off.

Software Source Code:

```
$MOD751

; *****************************************************
; *                                                   *
; *         HD61202 Application Note V1.0             *
; *                                                   *
; *****************************************************

; The processor clock speed is 16MHz.
; Cycle time is .750mS.
; HD61202 demo software to display
; the Hantronix logo on a 128 x 64 LCD.

        org     00h
        ljmp    start

        org     100h

; Initialize the 64gs12

Start:
        mov     p3,#00
        mov     r0,#00h         ;set 64gs12 left
        mov     dptr,#msgi1     ;initialization bytes
        lcall   command
        mov     r0,#01h         ;set 64gs12 right
        mov     dptr,#msgi1     ;initialization bytes
        lcall   command

; Display pattern

        mov     r4,#0b8h        ;page command
        mov     r5,#08h;        ;page count
Loop1:
        mov     r0,#00h         ;set 64gs12 left
        mov     dptr,#msg1l
        lcall   byte
        clr     p3.1            ;set command
        inc     r4              ;bump page add
        mov     a,r4
        mov     r1,a
        lcall   write
        djnz    r5,loop1        ;repeat 8 times
        mov     r4,#0b8h        ;page add. Command
        mov     r5,#8h          ;page count
Loop2:
        mov     r0,#01h         ;set 64gs12 right
        mov     dptr,#msg1r
        lcall   byte
        clr     p3.1            ;set command
        inc     r4              ;bump page add
        mov     a,r4
        mov     r1,a
        lcall   write
        djnz    r5,loop2        ;repeat 8 times
        sjmp    $               ;end
```

```
; ****************************************************
; SUBROUTINES

; COMMAND sends the byte pointed to by
; the DPTR to the graphics module
; as a series of commands.

Command:
        clr     p3.1            ;set command
Command2:
        clr     a
        movc    a,@a+dptr       ;get byte
        cjne    a,#099h,command1        ;done?
        Ret
Command1:
        mov     r1,a
        lcall   write           ;send it
        inc     dptr
        sjmp    command2

; BYTE sends the byte pointed to by
; the DPTR to the graphics module
; as a series of data bytes.

Byte:
        setb    p3.1            ;set data
        sjmp    command2

; WRITE sends the byte in R1 to the
; display.

Write:
        mov     a,r0            ;CS the display
        jnz     write1          ;right half
        setb    p3.2            ;left half
Write2:
        mov     p1,r1           ;get data
        setb    p3.0            ;strobe it
        Nop
        clr     p3.0
        clr     p3.2            ;de-select module
        clr     p3.3
        Ret
Write1:
        setb    p3.3
        sjmp    write2
```

```
;************************************************
; TABLES AND DATA

; Initialization bytes
Msgi1:
        db      0c0h,40h,0b8h,3fh,99h

; "Hantronix", left half

Msg1l:
        db      0,0feh,10h,10h,10h,0feh,0            ;H
        db      0fch,12h,12h,12h,0fch,0              ;A
        db      0feh,08h,10h,20h,0feh,0              ;N
        db      02h,02h,0feh,02h,02h,0               ;T
        db      0feh,12h,32h,52h,8ch,0               ;R
        db      7ch,82h,82h,82h,7ch,0                ;O
        db      0feh,08h,10h,20h,0feh,0              ;N
        db      0,0,82h,0feh,82h,0                   ;I
        db      0,0c6h,28h,10h,28h,0c6h,0            ;X
        db      0,38h,7ch,0f8h,7ch,38h,0             ;heart
        db      0,99h

; "Hantronix", right half (reverse video)

Msg1r:
        db      0ffh,0c7h,83h,07h,83h,0c7h,0ffh      ;heart
        db      0ffh,39h,0d7h,0efh,0d7h,39h,0ffh     ;X
        db      0ffh,0ffh,7dh,01h,7dh,0ffh           ;I
        db      01h,0dfh,0efh,0f7h,01h,0ffh          ;N
        db      83h,7dh,7dh,7dh,83h,0ffh             ;O
        db      073h,0adh,0cdh,0edh,01h,0ffh         ;R
        db      0fdh,0fdh,01h,0fdh,0fdh,0ffh         ;T
        db      01h,0dfh,0efh,0f7h,01h,0ffh          ;N
        db      03h,0edh,0edh,0edh,03h,0ffh          ;A
        db      0ffh,01h,0efh,0efh,0efh,01h,0ffh     ;H
        db      0ffh,99h

        end
```