Programming the NewHaven display

General notes

1. The I2C controller does not support writing data from a master to this slave except for the 'Read Busy Flag and Address' command where b7 BF high bit indicates the device is busy so won't accept instructions.

Initializing the Display

Display contrast is sensitive to internal voltage Vo setting. This code assumes a 4.5V power supply, change Vo generator bits in the 'Follower control' command for other voltages.
To initialize the display send out the following code:

```
hi2csetup i2cmaster,$7C,i2cfast_8,i2cbyte
hi2cout ($00,$39,$14,$70,$5e,$6a,$0f,$06,$01,$02)
```

Initialization sequence detail

```
hi2csetup i2master, $7C,i2cfast_8,i2cbyte  'configure I2C display interface, set address & speed
                'This command sets the Picaxe as master to transmit data to slave I2C address
                $7C using fast speed mode (400kHz) , and byte rather than word packets.
                Note that the slave last address bit is 0 for read,$7c, and 1 for write $7d to the
                slave
hi2cout ($00,$39,$14,$70,$5e,$6a,$0f,$06,$01)  'initialize the display using the instructions
below…
        '$00 Control byte contains only two bits, b8 Co, & b7 RS. For all instructions except writing
                data to CG,DD & Icon RAM RS=0. So in the majority of cases the control byte will
                be $00, if all following bytes are data, or $80, if another control byte follows the
                data byte. If writing to ram the code becomes $40 if no further control bytes are to
                be sent or $c0 if additional control bytes follow. In this case the control byte is $00
                so all remaining bytes are instructions. An I2C end terminates the sequence.
                Here's what all the instruction bytes mean:
        $39 'Function Set' b111001 b4-1 – 8bit bus mode b3=0 - 1 line display mode b2=0 -5x8
                display rather than double height mode, b0=1 use extension instruction set
        $14 'Bias Selection / Internal OSC frequency adjust' b10100 b3=1 bias is ¼ b2-0 1000
                frame frequency is 183khz.
        $70 'Contrast set (low byte)' b1110000 b3-b0=contrast
        $5e 'Power / Icon control / Contrast set (high byte)' b01011110 b3=1 Icon display is on,
                b2=1 Booster circuit is on, b1-b0=10 top 2 of 6 bit contrast set. Contrast is set half
                way at 32.
        $6b 'Follower control' b01101010 b3-1 Follower on b2-0 Vo generator control
        $0c 'Display ON / OFF' b00001111 b2=1 high entire display on b1=1 Cursor is off. If you
                b0=0 cursor blink off. If you want cursor on and blinking issue $0f
        $06 'Entry Mode Set' b00000110 b1=1 Increment DDRAM address b0=0 display is not
                shifted
        $01 'Clear Display' – note this also brings the cursor to the left edge on the display's first
                line. Note that this command takes quite a bit of time as it writes $32 the space
                character to all display positions then returns the cursor home. On faster parts,
                8MHz and above add a delay (pause or pauseus) of about 1ms. Experiment, if too
                short the following commands won't execute properly, ie if text is written to the
                display immediately following this command the first couple of characters may not
                appear or may be corrupt.
```

Splash Screen

I like to include splash screens right after initializing the display to indicate the name of the device, its software version and the owner etc. You don't need to re-issue the hi2csetup command if you're writing to the same device.

```
hi2cout ($80,$0c,$40," CleverLoad V1.0 ")'turn cursor off & write first line
hi2cout ($80,$c0,$40,"SAGACITIC SOLN'S")'set cursor to start of 2nd line & write.
```

Special Characters

Up to eight special characters can be loaded into CGRAM. These special characters when created automatically occupy the first 16 positions in the character generator table, so to call them in a command issue their addresses $00 to $0f as shown below

I typically place this data in a subroutine called 'Special Characters'

In order to write the CGRAM first set the character generator RAM address pointer to the first row of the character to be written. when writing all eight characters, start with the address pointer command $40, essentially address zero. In order to set this address the Instruction table IS bit, b0 in the 'Function Set' instruction, must be set to 0, it's normally 1. To do this issue the command:

```
  hi2cout ($00,$38,$40)  'The $38 sets instruction table bit low, the mode needed to be able to
```

write the CGRAM. The $40 sets the CGRAM address counter to the first row of the first character.

Each character is defined by issuing eight bytes one byte for each row in the 5x8 character. Each byte sent automatically increments the CGRAM address counter. In each byte, bits b0-b4 define the off – on characteristics of the five bits in each character row. The five bit code is shown in Hex, so that $00 would be a blank row, $1F would be a row where all elements are on and b10011, hex $13, would create a row that had the first (left most) and two (rightmost) elements in the row on, and the others off.

In the following example 5 bargraph characters and three battery icon characters are created. The empty battery symbol is in location $00, followed by 5 bargraph symbols. Each bargraph symbol consists of one to five vertical lines with the fifth line being wider. Then two more battery symbols, half then full battery are created. After creating the characters they are sent to line1 of the display.

```
WriteCGRAM:    'writes CGRAM bargraph & battery characters
  hi2cout ($00,$38,$40)  'set instruction table lo to write CGRAM & set address
  hi2cout
($40,$0e,$1f,$11,$11,$11,$11,$1f,$00,$00,$10,$10,$10,$10,$10,$10,$00,$00,$18,$18,$18,$18,$
18,$18,$00)
  hi2cout
($40,$00,$1c,$1c,$1c,$1c,$1c,$1c,$00,$00,$1e,$1e,$1e,$1e,$1e,$1e,$00,$01,$1f,$1f,$1f,$1f,$1f
,$1f,$01)
  hi2cout ($40,$0e,$1f,$11,$11,$1f,$1f,$1f,$00,$0e,$1f,$1f,$1f,$1f,$1f,$1f,$00)
  hi2cout ($00,$39)          'set instruction table back to hi
  hi2cout ($80,$c0,$40,"CGRAM Write Done")
  pause 1600 ' allow some time for the write complete command to be read
  char=0
return

Use this section to display the special characters
  hi2cout ($00,$0c,$01,$02)                    'turn off cursor clear display go home
  pause 1 'remember clearing the display takes time
  hi2cout ($40,$00,$01,$02,$03,$04,$05,$06,$07,$08,$09,$0a,$0b,$0c,$0d,$0e,$0f)'send some
characters
  pause 2000
  hi2cout ($00,$01,$02)
  hi2cout($40,"  0  20    50dBm")
  hi2cout ($80,$c0,$40,$04,$04,$04,$04,$04,$04,$04,$01,"     22",$06)
  goto main
```

Common Commands You'll use a lot

Set the display's address…7c

```
hi2setup i2cmaster,7c,I2cfast_8,i2cbyte
```

Initialize the display
hi2cout ($00,$39,$14,$70,$5e,$6b,$0c,$06,$01)

Issue multiple commands
hi2cout ($00,cmd.cmd,cmd,cmd…)
Only an I2C end command terminates this sequence.

Issue several commands
Multiple instruction wirtes…hi2cout ($80,cmd,$80,cmd,$80,cmd
If mixing multiple instructon wirtes and data writes use the follwing:
hi2cout ($80,cmd,$80,cmd,$40,"A few words ")
Where $40 is the last control byte (sets the display wirte location in this case to char 1 line 1) before a string of data to be written.

Clear the display
$00
This command wirtes blank characters to all 32 locations on the display so it takes some time. The comand also sets the cursor to $00, the first character on line 1.
Follow this command with a pause 1 command to incorporate a small delay (pretty much any speed or use a pauseus for tailored smaller delays).

Set the location of the next character to be written and write a string
hi2cout ($40,"Text to be write")
will add to text on display after last entry written
Notes:
Both lines of the display can be cleared, but not just one line.
It may be easier although not code efficent to rewrite a complete line, either line 1 or line 2, rather than clearing the complete display and then re-writing it.

Return Cursor and DDRAM pointer to home location (first character of line 1)
$02