Application Note

Crystal Clear and Visibly Superior LCD Modules

Interfacing to a Hantronix 128x64 Chip-On-Glass LCD

Introduction:
This note describes a simple interface between a Hantronix HDG12864F-1 graphic LCD and an 8-bit micro-controller. The techniques described here are also useful for any other controller type and interfacing technique. The micro-controller is connected via its I/O lines and all signals to the LCD are controlled directly by software. This is not the only possible interfacing scheme, but it is the most popular and one that best illustrates the use of the module.

The Hantronix HDG12864F-1 is a chip-on-glass graphics LCD. All control electronics, contrast control and bias circuits and temperature compensation circuits are built into the module. Each of these circuits is described separately below.

Module Description:
Display mapping: Individual pixels can be controlled by writing a byte to a specific address. Each address is mapped to a corresponding set of 8 pixels on the display. Please refer to Figure 1. Note that the order of the columns is reversed. The first byte of data entered after setting the address registers to 0 will appear in the upper right of the display.

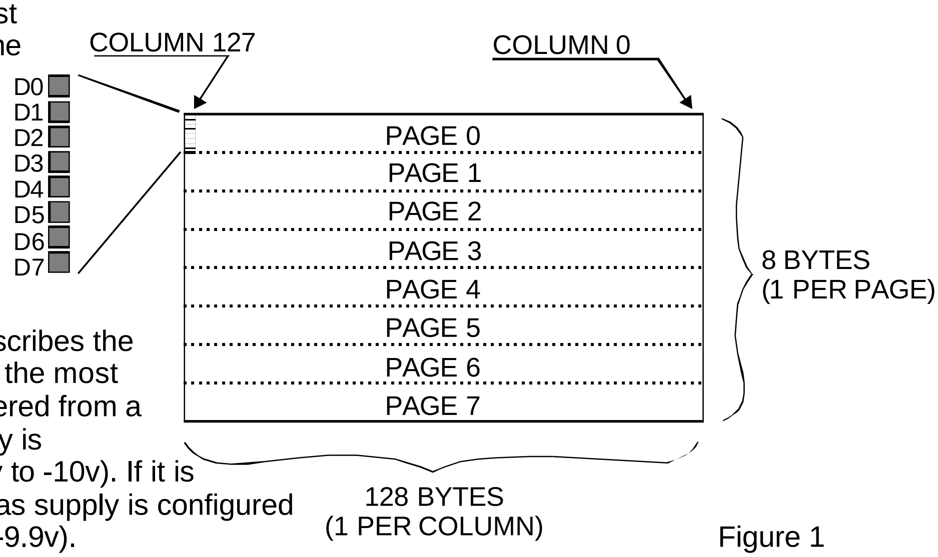


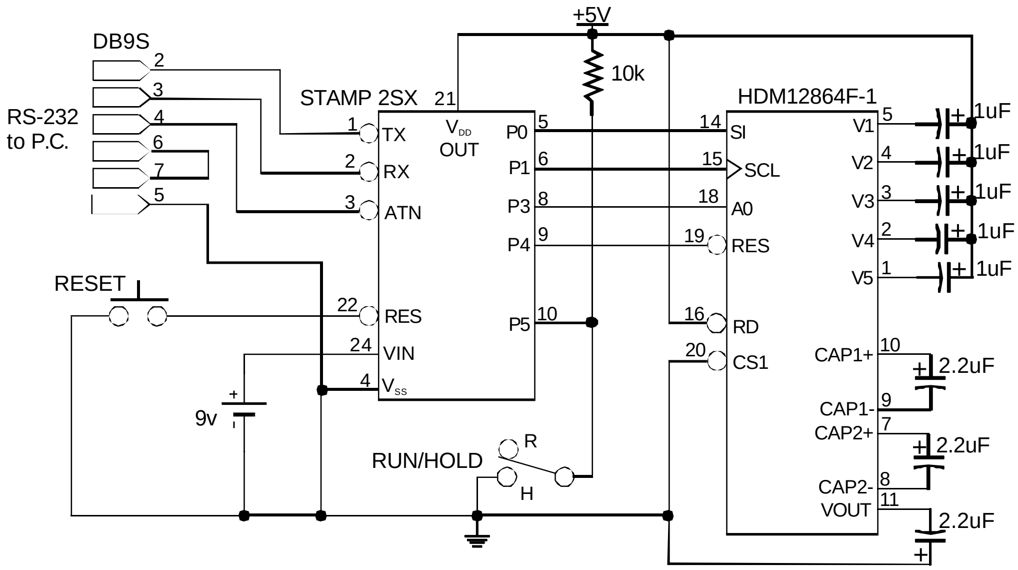
Figure 1

Bias Power Supply: The display requires approximately negative 10 volts to power the LCD. This can be supplied from an external source but the display can also generate it on board. This note describes the self-generated technique as this is the most common use. If the display is powered from a 5v source the internal power supply is configured as a voltage tripler (+5v to -10v). If it is powered from a 3.3v source the bias supply is configured as a voltage quadrupler (+3.3V to -9.9v).

Contrast Control: The contrast can be controlled from the microprocessor by sending a command to the module. The specification for the controller chip on this display refers to this function as “Electronic Volume Control”. This term is misleading and is probably the result of a translation error. The contrast is set by sending an 81h command followed immediately by the value for the contrast desired. There are 64 levels of contrast that can be set giving very fine control of this function. The initial value chosen for this value, which is set during initialization, should be determined by experiment as the desired contrast level is somewhat subjective.

Temperature compensation: This is a necessary function as the module is designed to operate over an extended temperature range of -20°C to +70°C. This function is also built into the module as a set of values which are preprogrammed at the time of manufacture.

Schematic Diagram:



Schematic:
This schematic is a simple design based on the Stamp 2SX from Parallax, Inc. This module is a complete micro-controller with a built in Basic interpreter. The program is downloaded to the module from a PC which the module then stores in flash memory. From this point on the circuit is self-contained and remembers its program even after power is removed.

Data and commands are sent to the module via a synchronous serial interface. This type of interface makes efficient use of I/O lines from the microprocessor. No status flag is needed as the display processes commands and data faster than they can be sent to the display via the serial interface. Reading from the display is not possible in the serial mode so the hardware and software interface can be made quite simple.

The serial interface operates in a very straightforward manner. The sequence of event is as follows. The A0, CS and WR lines are set to their proper states. The most significant bit of the data is placed on the SI line and the SCL (clock) line is pulsed low and then high. The data is clocked into the display on the rising edge of the clock. This is repeated 7 more times to finish sending the byte. See Figure 2.

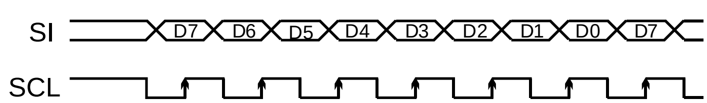


Figure 2

Software:
In this example the display is reset and then initialized. Initialization is necessary to set up the internal modes of the Epson SED1565 controller chip and must be done before any data is sent to the display. The following table outlines the various commands and their functions in the order they are normally executed. Commands that are necessary but are the same as the defaults are not listed and need not be executed. All code bytes are listed in hexadecimal.

COMMAND	DESCRIPTION	CODE	COMMENTS
RESET	SOFTWARE RESET	E2	SAFETY, NOT ABSOLUTELY NEEDED
V_5 VOLTAGE	SETS THE RESISTOR RATIO	24	FOR +5V ONLY
POWER MODE	SETS POWER MODE TO TRIPLE	2F	FOR +5V ONLY
VOLUME	SETS THE CONTRAST LEVEL	81,24	FOR +5V ONLY AVERAGE CONTRAST

After initialization the LCD is ready to accept display data. When first powered-up, the display RAM contains random data. This can be cleared by filling it with 0's or a full screen of data (image). The demo program listed here clears the display after initializing it. It then fills the display with an image consisting of text and icons.

Figure 3

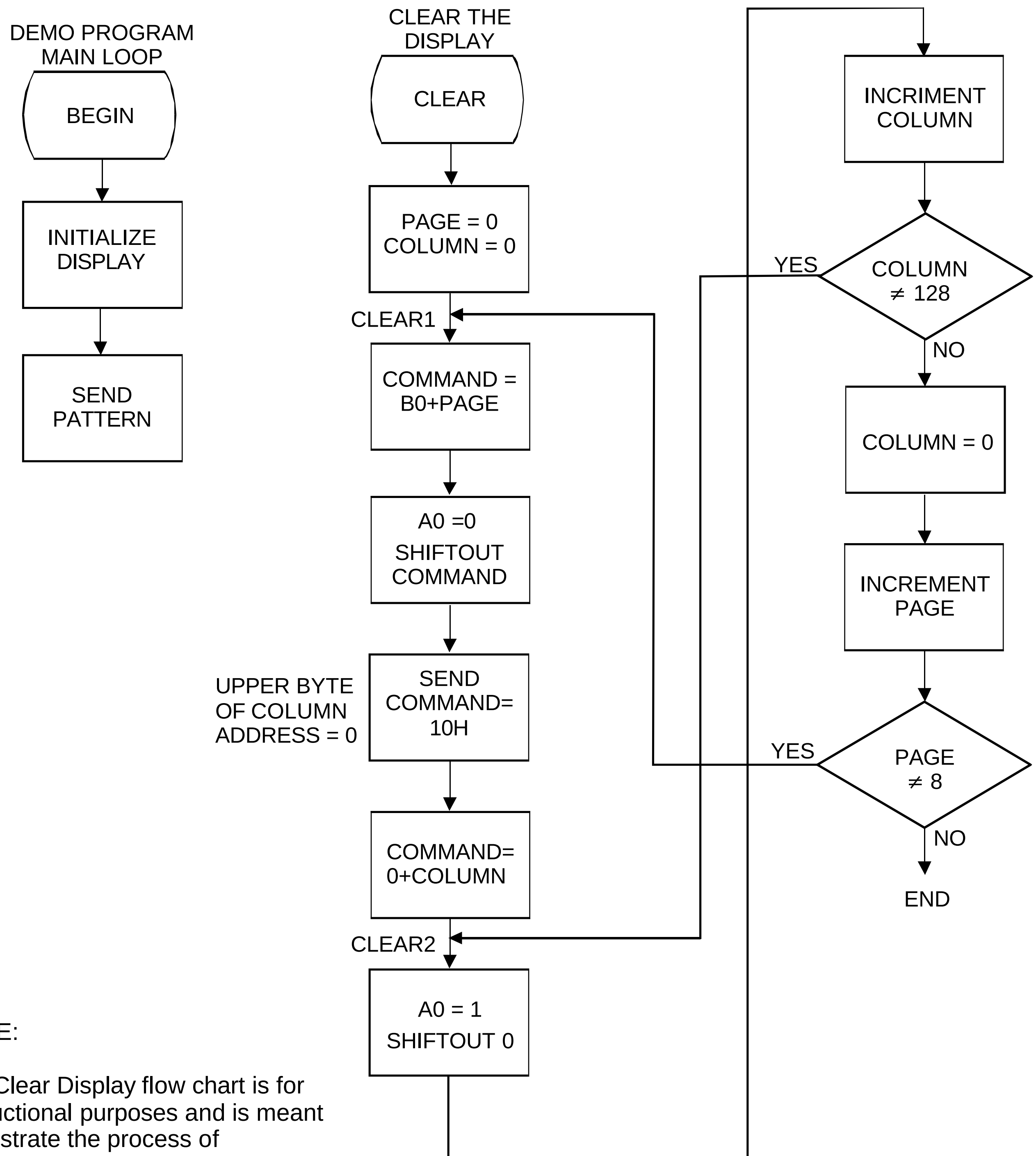
It should be noted that the display RAM is mapped to the display so that the first character in RAM at column 0, page 0 will appear in the upper right corner of the display. When a byte is written to the display the column address is incremented by one which moves the virtual cursor to the left one column (pixel). When the first page of pixels (128 bytes) has been written, a new page address must be written to the display. The column address is then loaded with 0 and the next page is filled. This continues until all 1024 bytes have been written. Figure 3 is the pattern displayed in this example.

It is possible to address a specific point on the display and change only a portion of it without disturbing the remainder of the display. This is done by simply setting the page and column addresses and writing the data.

The display can be updated at 1.6uS per byte maximum. At this rate the display could be completely filled with data in less than 2mS. A more realistic figure is around 10mS. This implies that a video image with a frame rate of about 70Hz could be displayed. While this is true the resultant image would be of very poor quality. This is because the optical response time is around 1 second at 25°C. Changing data any faster than about once every second will result in a fuzzy or ghostly looking image.

This display has several features that add to its versatility. The displayed image can be changed from positive to negative with a single command. It can also be put into two different power saving modes, also under software command. In standby mode the power consumption is reduced to about 0.2uW and about 0.05uW in the sleep mode. The contents of the display RAM is retained in both of these modes. In normal operation the display consumes about 500uW of power.

Program Flow Chart:



NOTE:

The Clear Display flow chart is for instructional purposes and is meant to illustrate the process of transferring a screen full of data to the display.

Software Sample:

The following sample program is written in Control Basic and will run as shown on a Basic Stamp 2SX module from Parallax, Inc. when connected to the LCD as shown in the schematic on page 2 of this application note. It is a simple program to illustrate the basic principles used to display data on this LCD module.

```
' HANTRONIX, INC.
' DEMO PROGRAM: HDG12864-1
' For Stamp 2SX from Parallax, Inc.
' 7-22-99 jmb

'I/O pins
       si con 0
       sc con 1
       a0 con 3
       reset con 4

' DATA TABLE

' this is the pattern to be displayed

'page0
Message data $ff,$01,$01,$01,$81,$c1,$61,$61,$61,$61,$e1,$01,$01,$01,$81,$c1
       data $61,$61,$61,$c1,$81,$01,$01,$01,$01,$01,$01,$01,$e1,$01,$01,$01
       data $01,$01,$01,$01,$01,$41,$41,$41,$81,$01,$01,$41,$41,$41,$81,$01
       data $01,$d1,$41,$01,$81,$41,$41,$81,$f1,$01,$81,$41,$41,$c1,$01,$81
       data $41,$41,$41,$01,$01,$81,$41,$41,$81,$c1,$01,$a1,$91,$91,$11,$e1
       data $01,$01,$01,$01,$01,$01,$01,$f1,$21,$41,$81,$01,$01,$91,$91,$a1
       data $c1,$01,$01,$01,$41,$81,$01,$81,$41,$01,$01,$01,$61,$91,$91,$91
       data $61,$01,$61,$91,$11,$11,$21,$01,$01,$f1,$21,$01,$01,$01,$01,$ff
'page1
       data $ff,$00,$00,$00,$ff,$01,$00,$00,$00,$00,$ff,$00,$00,$00,$83,$01
       data $00,$00,$00,$01,$ff,$00,$00,$00,$00,$00,$00,$00,$ff,$00,$00,$00
       data $00,$00,$00,$00,$02,$05,$05,$05,$04,$00,$02,$04,$04,$04,$03,$00
       data $04,$07,$04,$00,$07,$00,$00,$00,$07,$00,$80,$81,$81,$07,$00,$07
       data $85,$85,$85,$02,$00,$00,$80,$80,$80,$07,$00,$03,$e4,$24,$04,$03
       data $40,$20,$20,$20,$c0,$00,$01,$07,$01,$01,$81,$80,$03,$84,$04,$c4
       data $23,$20,$20,$c0,$04,$02,$01,$02,$04,$80,$80,$80,$03,$04,$a4,$84
       data $03,$00,$84,$84,$05,$e6,$04,$40,$24,$27,$24,$c0,$00,$00,$00,$ff
'page2
       data $ff,$00,$00,$00,$83,$87,$8c,$8c,$8c,$8c,$8f,$80,$80,$80,$83,$87
       data $8c,$8c,$8c,$87,$83,$80,$80,$8c,$8c,$8c,$8c,$8c,$8f,$80,$80,$80
       data $80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80
       data $80,$80,$80,$80,$80,$80,$80,$80,$80,$84,$8a,$8a,$8a,$89,$80,$84
       data $8a,$8a,$8a,$89,$80,$8f,$8a,$8a,$8a,$84,$80,$88,$8f,$88,$80,$80
       data $87,$89,$89,$88,$87,$80,$80,$80,$80,$8f,$80,$80,$81,$8f,$80,$87
       data $88,$88,$88,$87,$80,$80,$80,$80,$81,$82,$82,$8f,$80,$88,$8f,$88
       data $80,$8f,$80,$80,$81,$8f,$80,$84,$88,$88,$88,$87,$00,$00,$00,$ff
'page3
       data $ff,$00,$00,$00,$31,$31,$31,$31,$31,$31,$31,$31,$31,$31,$01,$01
       data $09,$11,$61,$81,$01,$81,$61,$11,$09,$01,$01,$01,$f9,$01,$01,$01
       data $f9,$01,$01,$01,$01,$81,$61,$11,$f9,$01,$01,$01,$c1,$21,$11,$09
       data $09,$09,$09,$11,$21,$c1,$01,$01,$01,$e1,$11,$09,$09,$09,$09,$09
       data $09,$f9,$01,$01,$09,$09,$09,$09,$f9,$09,$09,$09,$09,$01,$01,$f9
       data $01,$01,$01,$01,$81,$61,$11,$f9,$01,$01,$01,$01,$01,$81,$71,$09
       data $71,$81,$01,$01,$01,$01,$01,$f9,$01,$01,$01,$01,$01,$01,$01,$f9
```

```
	data $01,$01,$31,$31,$31,$31,$31,$31,$31,$31,$31,$31,$00,$00,$00,$ff
'page4
	data $ff,$00,$00,$00,$33,$33,$33,$33,$33,$33,$33,$33,$33,$33,$00,$40
	data $20,$10,$0c,$02,$01,$02,$0c,$10,$20,$40,$00,$00,$7f,$00,$00,$00
	data $7f,$20,$18,$04,$03,$00,$00,$00,$7f,$00,$00,$00,$0f,$10,$20,$40
	data $40,$40,$40,$20,$10,$0f,$00,$00,$00,$40,$21,$1a,$06,$02,$02,$02
	data $02,$7f,$00,$00,$00,$00,$00,$00,$7f,$00,$00,$00,$00,$00,$00,$7f
	data $20,$18,$04,$03,$00,$00,$00,$7f,$00,$00,$00,$60,$1c,$07,$04,$04
	data $04,$07,$1c,$60,$00,$00,$00,$7f,$01,$01,$01,$01,$01,$01,$01,$7f
	data $00,$00,$33,$33,$33,$33,$33,$33,$33,$33,$33,$33,$00,$00,$00,$ff
'page5
	data $ff,$00,$00,$00,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06
	data $06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$86,$86,$86
	data $86,$86,$86,$86,$86,$86,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06
	data $06,$06,$06,$06,$86,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06
	data $06,$06,$66,$e6,$c6,$86,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06
	data $06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06
	data $06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06,$06
	data $06,$06,$06,$06,$e6,$06,$e6,$c6,$86,$06,$06,$06,$00,$00,$00,$ff
'page6
	data $ff,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$08,$08,$08,$7f,$08
	data $08,$08,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$83,$00,$00
	data $00,$38,$44,$82,$01,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$c0
	data $f0,$f8,$fc,$ff,$ff,$ff,$fc,$f8,$f0,$c0,$00,$00,$00,$00,$00,$00
	data $00,$00,$00,$00,$03,$07,$1e,$fc,$f4,$e4,$c4,$f4,$f4,$64,$04,$04
	data $fc,$00,$00,$00,$00,$00,$00,$00,$04,$04,$04,$04,$7c,$fc,$ff,$fe
	data $fe,$fc,$7c,$7c,$00,$fc,$fc,$fc,$00,$00,$00,$00,$00,$00,$00,$00
	data $00,$00,$00,$00,$03,$00,$03,$ff,$ff,$fe,$00,$00,$00,$00,$00,$ff
'page7
	data $ff,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$81,$81,$81,$81,$81
	data $81,$81,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$83,$82,$82
	data $82,$82,$82,$82,$83,$82,$80,$80,$80,$80,$80,$80,$80,$80,$80,$81
	data $83,$83,$81,$84,$87,$84,$81,$83,$83,$81,$80,$80,$80,$80,$80,$80
	data $80,$80,$80,$80,$80,$80,$80,$8f,$89,$8b,$8b,$8b,$8b,$88,$88,$88
	data $8f,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80
	data $80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80
	data $80,$80,$80,$80,$be,$80,$be,$9f,$8f,$83,$80,$80,$80,$80,$80,$ff


command		var	byte	'current controller command
page		var	byte	'current controller page address
column var	byte	'current controller column address
tabdata		var	byte	'current table data
strAddr		var	word	'current table pointer

strAddr = Message		'set start of message

'Initialize Stamps I/O lines
begin
	dirl = %11011111	'P0-P4 & P6-P7 to output
	low sc
	low a0
	low reset		'reset the display
	high reset

' Initialize the display
	command = $e2'software reset
```

```
       gosub commsend
       command = $24'set resistor ratio
       gosub commsend
       command = $2f'set power mode
       gosub commsend
       command = $81'contrast set command
       gosub commsend
       command = $24'contrast setting
       gosub commsend


' Clear the display
clear
       page = 0
       column = 0
clear1
       command = $b0
       command = command+page
       gosub commsend      'set page address
       command = $10
       gosub commsend      'set column address
       command = 0
       command = command+column
       gosub commsend
       high   a0
clear2
       shiftout si,sc,msbfirst,[0]      'send a 0
       column = column+1   ' increment column
       if column <>128 then clear2
       column = 0
       page = page+1'incriment page
       if page <>8 then clear1
       command = $af
       gosub commsend      'turn on the LCD

' Start of the demo program. This program
' displays a bitmap pattern

demo
       page = 0
       column = 0
demo1
       command = $b0
       command = command+page
       gosub commsend      'set page address
       command = $10
       gosub commsend      'set column address
       command = 0
       command = command+column
       gosub commsend
       high a0
demo2
       read strAddr,tabdata
       shiftout si,sc,msbfirst,[tabdata]'write
data
       strAddr = strAddr+1 'increment pointer
       column = column+1          'increment column
       if column <>128 then demo2
       column = 0
       page = page+1'increment page
       if page <>8 then demo1

end

' SUBROUTINES

' commsend sends the byte command as a command
commsend
       low a0
       shiftout si,sc,msbfirst,[command]
       return
```

References:

Specification for the Epson SED1565 LCD controller chip. Epson America **www.eea.epson.com/library/grlib.htm**

Specification for the Hantronix HDM12864F-1 LCD graphics display. Hantronix, Inc. **www.hantronix.com/dwng.htm**

Basic Stamp Manual Version 1.9. Parallax, Inc. **www.parallaxinc.com**