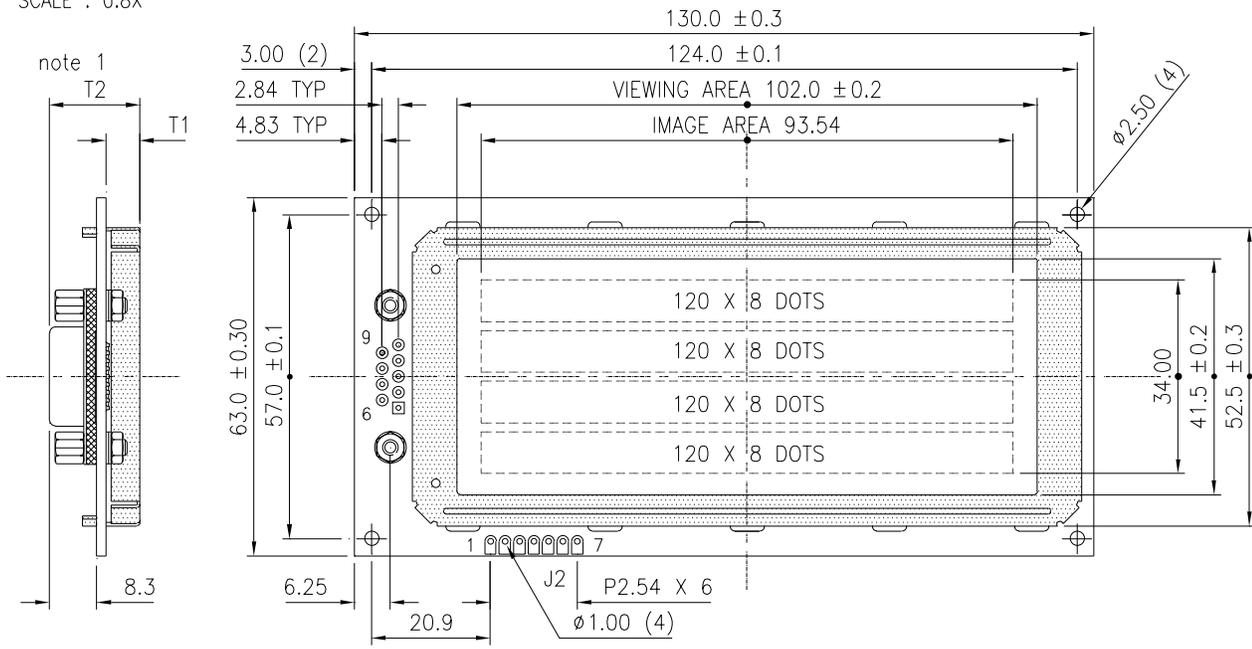
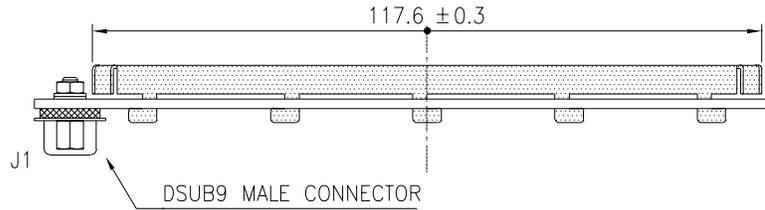


SCALE : 0.8X



NOTE 1

- T1 = 6.0 (NON-B/L)
= 10.0 (LED-B/L)
- T2 = 15.9 (NON-B/L)
= 19.9 (LED-B/L)



J1 INTERFACE

PIN NO.	FUNCTION	SIGNAL NAME FROM PC SERIAL PORT	PIN NO.	FUNCTION	SIGNAL NAME FROM PC SERIAL PORT
1	NC	DCD	6	NC	DSR
2	NC	RxD	7	SRL PWR2	RTS
3	RS232 DATA IN	TxD	8	NC	CTS
4	SRL PWR1	DTR	9	NC	RI
5	Vss	SG			

J2 INTERFACE

* to be used only if SPI is selected

PIN NO.	1	2	3	4	5*	6*	7*
FUNCTION	Vss	Vcc	LED+	DATA-IN	SPI CS	SPI CLK	SPI BUSY

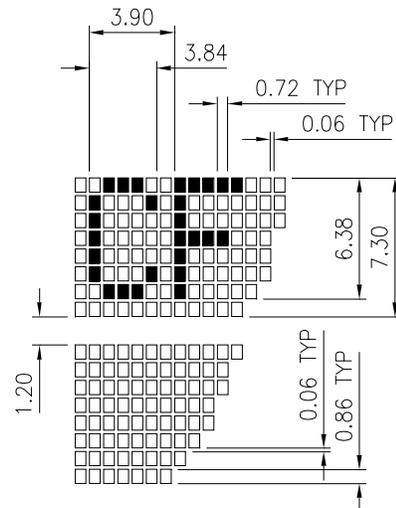
**ASCII input
Direct Plug-in to COM port**

Item	Symbol	Condition	Min.		Typ.	Max.		Unit
			N	W		N	W	
Operating Temperature	Top		0	-20		50	70	°C
Storage Temperature	Tstg		-10	-30		60	80	°C
Supply for Logic (if J2 is used)	Vcc-Vss	Ta= 25°C	4.5		5.0		5.5	V
Supply Current Drain	Icc	Ta= 25°C			6.0			mA
Backlight Current Drain	I _{BL}	Ta= 25°C			525			mA
SPI CLK Frequency	f _{CLK}	Ta= 25°C					18	kHz

RS232 Baud Rates (via DIP switch settings)	1200	2400	4800	9600

FEATURE HIGHLIGHTS :

- Built-in Software Control for Contrast and Backlighting
- Smooth Horizontal Scrolling
- Gapless Bar Graphing
- Big Character Formation



Crystalfontz Technology

*** PRELIMINARY ***

20x4 and 16x2 intelligent serial interface display command set

This document corresponds with beta code v0.7.

Please contact Brent Crosby at (509) 291-3514, (888) 206-9720, or brent@crystalfontz.com with any bugs.

For "binary" data the notation \xxx is used, where xxx is the decimal representation of the number. \000 to \255 cover all possible values. This same notation is supported by the free Crystalfontz WinTest utility software.

Some keyboards will not be able to create all the codes. We recommend you use Crystalfontz WinTest utility to send the examples to the display. You may highlight each line of the examples, copy them, and then past them into the WinTest edit box with a right-click, "Paste". Be sure not to highlight the invisible carriage return at the end of the line.

The display will power itself (without the backlight) from the DTR and RTS control lines on most PC parallel ports. Supply +5 volts and ground to use the backlight.

The cable is a 9-pin female to 9-pin female, "straight through". Radio Shack 26-125 works great with the "AT" compatible 9-pin serial port available on most PCs.

The baud is set with the dip switches:

Switch 1	Switch 2	Baud
Up (on)	Up (on)	= 9600
Down (off)	Up (on)	= 4800
Up (on)	Down (off)	= 2400
Down (off)	Down (off)	= 1200

Display Control Codes

ASCII	Keystrokes	Function
\000	Control @	Null
\001	Control A	Cursor Home
\002	Control B	Hide Display
\003	Control C	Restore Display
\004	Control D	Hide Cursor
\005	Control E	Show Underline Cursor
\006	Control F	Show Block Cursor
\007	Control G	Show Inverting Block Cursor
\008	Control H	Backspace (destructive)
\009	Control I	<i>not used</i>
\010	Control J	Line Feed
\011	Control K	Delete in place
\012	Control L	Form Feed (Clear Display)
\013	Control M	Carriage Return
\014	Control N	Backlight Control
\015	Control O	Contrast Control
\016	Control P	<i>not used</i>
\017	Control Q	Set Cursor Position (Column and Row)
\018	Control R	Horizontal Bar Graph
\019	Control S	Scroll ON
\020	Control T	Scroll OFF

\021	Control U	Set Scrolling Marquee Characters
\022	Control V	Enable Scrolling Marquee
\023	Control W	Wrap ON
\024	Control X	Wrap OFF
\025	Control Y	Set Custom Character Bitmap
\026	Control Z	Reboot
\027	Escape	Escape Sequence Prefix
\028	Control Backslash	3x4 Large Block Number
\029	Control Quote	4x4 Large Block Number
\030	Control Equal	Send Data Directly to the LCD Controller
\031	Control Minus	Show Information Screen

Explanation of Control Functions

Cursor Home (\001 , Control A)

Moves cursor to the top left character position. No data is changed. Identical to Control Q,0,0.

Hide Display (\002 , Control B)

Display is blanked, no data is changed.

Restore Display (\003 , Control C)

Restores blanked display; nothing else is changed.

Hide Cursor (\004 , Control D)

Cursor is not shown; nothing else is changed.

Show Underline Cursor (\005 ; Control E)

Shows a non-blinking underline cursor at the printing location.

Show Block Cursor (\006 ; Control F)

Shows a blinking block cursor at the printing location.

Show Inverting Block Cursor (\007 ; Control G)

Shows a blinking block cursor at the printing location. This cursor inverts the character rather than replacing the character with a block. This cursor style is the default cursor at power-up.

Backspace (\008 ; Control H)

Moves the cursor back one space and erases the character in that space. Will wrap from the left-most column to the right-most column of the line above. Will wrap from the left-most column of the first row to the right-most column of the last row.

Line Feed (\010 ; Control J {also Control Enter on some keyboards})

Moves the cursor down one row. If SCROLL is on and the cursor is at the bottom row, the display will scroll up one row and the bottom row will be cleared. If SCROLL is off, and the cursor is at the bottom row, it will wrap up to the same character position on the top row.

Delete in place (\011 ; Control K)

Deletes the character at the current cursor position. Cursor is not moved.

Form Feed (\012 ; Control L)

Clears the display and returns cursor to Home position (upper left). All data is erased.

Carriage Return (\013 ; Control M)

Moves cursor to the left-most column of the current row.

Backlight Control (\014 ; Control N)

Send "Control-N" , followed by a byte from 0-100 for the backlight brightness. 0=OFF; 100=ON , intermediate values will vary the brightness. There are a total of 25 possible brightness levels.

Examples:

```
\014\000  
\014\050  
\014\100
```

Contrast Control (\015 ; Control O)

Send "Control O" , followed by a byte from 0-100 for the contrast setting of the displayed characters. 0 = very light, 100 = very dark, 50 is typical. There are a total of 25 possible contrast levels.

Examples:

```
\015\040  
\015\050  
\015\060
```

Set Cursor Position (Column and Row) (\017 ; Control Q)

Send "Control Q" followed by one byte for the column (0-19 for a 20x4 display, or 0-15 for a 16x2 display), and a second byte for the row (0-3 for a 4x20 or 0-1 for a 2x16). The upper-left position is 0,0. The lower-right position is 15,1 for a 16x2, and 19,3 for a 20x4. Here is an example for moving the cursor to column 11 of the second line:

```
\017\010\001
```

Horizontal Bar Graph (\018 ; Control R)

Send "Control R" followed by the following bytes:

```
graph_index  
style  
start_column  
end_column  
length  
row
```

graph_index determines which special characters are used:

graph_index	custom characters used
\000	0,1
\001	2,3
\002	4,5
\003	6,7

style is the bit pattern to use in drawing the graph:

```
\255 (11111111b) is a thick bar  
\000 (00000000b) will not be visible (all pixels are off)  
\085 (01010101b) is a striped bar  
\060 (00111100b) is a medium width bar, centered  
\015 (00001111b) is a medium width bar, low in the row  
\240 (11110000b) is a medium width bar, high in the row
```

any value is valid between \000 and \255, the msb is at the top of the row, the lsb is at the bottom of the row.

start_column and **end_column** are the character X coordinates of the graph area. Each must be between \000 and \019 for the 20x4 or between \000 and \015 for the 16x2. **start_column** must be less than or equal to **end_column**.

length is the length in pixels of the graph. Positive values will graph from the left edge of **start_column**, negative values will graph from the right edge of **end_column**. There are six pixels per character, so the maximum value of length for a 20x4 display is $20 * 6 = 120$. For a 16x2, the maximum value is $16 * 6 = 96$.

row is the character Y coordinate. \000-\003 is valid for the 20x4, \000-\001 is valid for the 16x2.

Examples:

```
\018\000\255\000\014\010\001
\018\000\015\000\014\236\001
```

Notes:

The entire graph area is completely re-written by each graph command, so there is no need to clear the area between successive updates of the same graph. If a length of \000 is written the entire graph area is cleared to spaces. Negative values can be calculated as $256 - \text{value}$. For instance if you want a graph to extend 20 pixels towards the left, from the rightmost column of the graph area, send 236 ($256 - 20 = 236$). No additional graph "setup" command is needed.

The graphs use some of the custom characters, and so may goof up the display if there are user-defined custom characters or large numbers shown.

Scroll ON (\019 ; Control S)

Turns Scroll feature ON. Then a **Line Feed** (Control J) command from the bottom row will scroll the display up by one row, independent of **Wrap**. If **Wrap** is also on, a wrap occurring on the bottom row will cause the display to scroll up one row. Scroll is on at power up.

Scroll OFF (\020 ; Control T)

Turns Scroll feature OFF. Then a **Line Feed** (Control J) command from the bottom row will move the cursor to the top row of the same column, independent of **Wrap**. If **Wrap** is also on, a wrap occurring on the bottom row will also wrap vertically to the top row. Scroll is on at power up.

Set Scrolling Marquee Characters (\021 ; Control U)

Send "Control U" followed by the following bytes:

index
character

index determines which of the 20 scrolling marquee characters gets set, \000 to \019 are valid

character is the value that this position in the scrolling marquee will be set to

Examples:

This will set the first 12 characters to "Crystallfontz", put another message on the display, and enable the scrolling marquee. The second line of this example ends in a space:

```
\004\022\255\012\017\000\001Scrolling Marquee
\021\000c\021\001r\021\002y\021\003s\021\004t\021
\005a\021\006l\021\007f\021\008o\021\009n\021\010t
```

```
\021\011z\021\012 \021\013 \021\014 \021\015 \021
\016 \021\017 \021\018 \021\019 \022\001
```

Notes:

There is a section of memory that holds 20 hidden characters. These 20 characters, along with the contents of one line of the display can be rotated pixel by pixel across the display in a circular fashion. Use this command multiple times to set the 20 "hidden" characters. Use the normal display functions to set the other characters in the line you want to rotate, then enable rotation with **Enable Scrolling Marquee** command. The hidden characters are set to blanks at power-up.

You will probably want to disable the scrolling marquee movement while you are setting the scrolling marquee characters or modifying characters on the rotating line. You will probably also want to move the cursor off the line or hide it.

Erratum:

If the first character of the line above the rotating line on a 20x4 display has any pixels on its bottom row set, they will appear as "ghost" pixels in the upper right of the scrolling line. This will only happen if the cursor or a custom character with its bottom row of pixels set is in the leftmost position of the line directly above the scrolling marquee line. These pixels are an artifact of the display controller memory architecture.

Enable Scrolling Marquee (\022 ; Control V)

Send "Control V" followed by the following bytes:

line
speed

line determines which line will scroll with the scrolling marquee, or if the scrolling marquee is disabled.

line function

```
\000 enable scrolling marquee on line 1
\001 enable scrolling marquee on line 2
\002 enable scrolling marquee on line 3 (20x4 only)
\003 enable scrolling marquee on line 4 (20x4 only)
\255 disable scrolling marquee
```

speed Determines the speed of the message scrolling. The line always shifts every 250ms, or at 4Hz.

speed sets the number of pixels that the line is scrolled by, and therefore the amount of time it takes the one character--or the whole message to repeat.

speed	shift rate	pixels per shift	character shift rate	20x4	16x4	20x4	16x2
				visible screen shift time	visible screen shift time	message repeat time	message repeat time
\001	4Hz	1	0.67 c/sec	30 sec	24 sec	60 sec	54 sec
\002	4Hz	2	1.33 c/sec	15 sec	12 sec	30 sec	27 sec
\003	4Hz	3	2.00 c/sec	10 sec	8 sec	20 sec	18 sec
\004	4Hz	4	2.67 c/sec	7.5 sec	6 sec	15 sec	13.5 sec
\005	4Hz	5	3.33 c/sec	6 sec	4.8 sec	12 sec	10.8 sec
\006	4Hz	6	4.00 c/sec	5 sec	4 sec	10 sec	9 sec

See the example in **Set Scrolling Marquee Characters** above.

Wrap ON (\023 ; Control W)

Turns **Wrap** feature ON. Then, a printable character received when the cursor is at the right-most column will cause the cursor to move down one row, to the left-most column. If the cursor is already at the right-most column of the bottom row, it will wrap to the top row if **Scroll** is OFF, or the display will scroll up one row if **Scroll** is ON.

Wrap OFF (\024 ; Control X)

Turns **Wrap** feature OFF. Then, a printable character received when the cursor is at the right-most column will cause the cursor to disappear (as it will be off the right edge of the screen) and any subsequent characters will be ignored until some other command moves the cursor back onto the display. This function is independent of **Scroll**.

Set Custom Character Bitmap (\025 ; Control Y)

The custom characters are mapped at \128 through \135 corresponding to **character 0** to **character 7**. Send "Control Y" followed by the following bytes:

character
data0
data1
data2
data3
data4
data5
data6
data7

character determines which of the eight custom characters is modified. 0-7 is valid.

data0-data7 are the bitmap information for this character. Any value is valid between 0 and 63, the msb is at the left of the character cell of the row, the lsb is at the right of the character cell. **data0** is at the top of the cell, **data7** is at the bottom of the cell.

Notes: The large digits use all of the custom characters, so if you modify the custom characters when large digits are displayed, the display will probably become corrupted. The bar graphs also use some of the custom characters.

Example:

```
\001\128\129\130\131\017\000\001\132\133\134\135  
\025\000\000\000\001\003\000\031\031\031  
\025\001\028\054\032\001\003\051\051\051  
\025\002\014\027\049\032\032\047\032\047  
\025\003\000\000\032\048\000\062\000\062  
\025\004\031\031\031\000\003\001\000\000  
\025\005\051\051\051\003\001\032\054\028  
\025\006\047\032\047\032\032\049\027\014  
\025\007\062\000\062\000\048\032\000\000
```

Reboot (\026 ; Control Z)

Send one "Control Z" followed by another "Control Z" to reboot the display.

Escape Sequences

There are 4 **Escape** sequences currently supported . These correspond to the escape sequences that are

sent for the four arrows keys in HyperTerminal with an ANSI terminal selected (and also WinTest). These sequences move the cursor only, and do not wrap.

ESC [A (equivalent to \027\091\065)
UP arrow

ESC [B (equivalent to \027\091\066)
DOWN arrow

ESC [C (equivalent to \027\091\067)
RIGHT arrow

ESC [D (equivalent to \027\091\068)
LEFT arrow

3x4 Large Block Number (\028 ; Control Backslash)

This command is only valid on the 20x4. On the 16x2 it is parsed and then discarded. Send "Control Backslash" followed by the following bytes:

**column
number**

column is the starting column of the number. \000-\017 are valid.

number is the number to display. \048 to \057 ('0' to '9') are valid.

Notes:

The large numbers use all the custom characters. There will be some corruption if they are used at the same time as the graphs or user defined custom characters. There is no large number initialization command needed.

Examples:

```
\004\012\028\0010\028\0051\028\0092\028\0133\028\0174  
\004\012\028\0005\028\0046\028\0087\028\0128\028\0169
```

4x4 Large Block Number (\029 ; Control Quote)

This command is only valid on the 20x4. On the 16x2 it is parsed and then discarded. Send "Control Quote" followed by the following bytes:

**column
number**

column is the starting column of the number. \000-\016 are valid.

number is the number to display. \048 to \057 ('0' to '9') are valid.

Notes:

The large numbers use all the custom characters. There will be some corruption if they are used at the same time as the graphs or user defined custom characters. There is no large number initialization command needed.

Examples:

```
\004\012\029\0000\029\0051\029\0102\029\0153  
\004\012\029\0006\029\0057\029\0108\029\0159
```

Send Data Directly to the LCD Controller (\030 ; Control Equal)

Send "Control Equal" followed by the following bytes:

**location
data**

location is the destination register on the LCD Controller:

location	register
\000	Control Register, (RS=0, RE=0)
\001	Data Memory, (RS=1, RE=x)
\002	Control Register, (RS=0, RE=1)

data is the data to write to the controller

Notes:

This command executes a low level write directly to the controller. Use this command at your own risk. Control Z, Control Z will reboot the display and recover from most mistakes.

Example:

```
\030\002\031\030\002\130
```

Show Information Screen (\031 ; Control Minus)

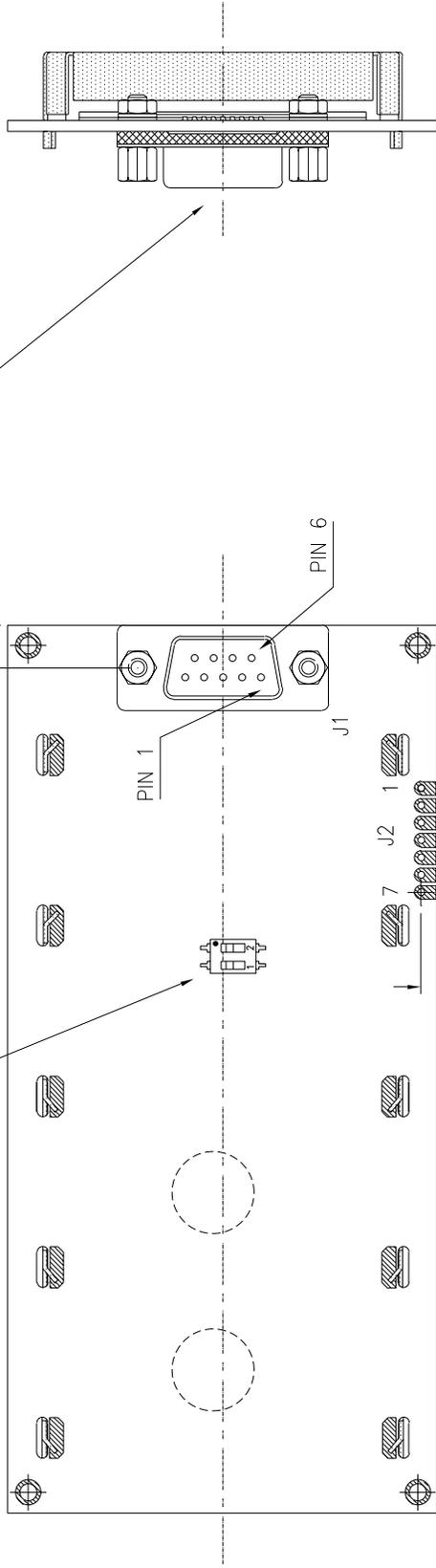
This command will show the baud rate, version and model number.

BACK VIEW

DSUB9 MALE CONNECTOR

6.25

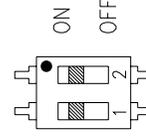
DIP SWITCH



J2 PIN-OUT DEFINITION	
PIN NO.	FUNCTION
1	VSS
2	REGULATED +5V SUPPLY
3	RS232 DATA / SPI DATA
4	SPI CS
5	SPI CLK
6	LED +VE SUPPLY, 5V
7	LED -VE SUPPLY

J1 PIN-OUT DEFINITION		SIGNAL NAME FROM PC SERIAL PORT
PIN NO.	FUNCTION	
1	NC	DCD
2	NC	RxD
3	RS232 DATA IN	TxD
4	SRL PWR1	DTR
5	VSS	SG
6	NC	DSR
7	SRL PWR2	RTS
8	NC	CTS
9	NC	RI

BAUD RATE	DIP SWITCH SETTINGS	
	PIN 1	PIN 2
1200	OFF	OFF
2400	ON	OFF
4800	OFF	ON
9600	ON	ON



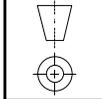
THIS DRAWING IS APPLICABLE TO THE FOLLOWING SUB-VARIANTS :-

- SKD204-634SS (NON-BACKLIT, GRAY STN DISPLAY)
- SKD204-634SG-LG (LED-BACKLIT, YEL-GREEN STN DISPLAY)

DISKETTE CY9927

DRAWN BY : ALEXIS	DWG NBR 9M634P01	REV A
APPROVED :	DATE 19 FEB 1999	SHEET 2 OF 2

SCALE 1 = 1	GEN. TOL. X.X = ±0.20
PROJECT 634	X.XX = ±0.10
	ANG = ±0.5°



TITLE MECHANICAL PRODUCT SPECIFICATIONS

