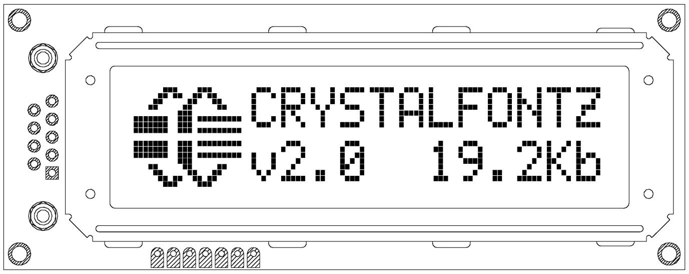
Crystalfontz America, Incorporated

SERIAL LCD MODULE SPECIFICATIONS



Crystalfontz Model Number	**CFA-632**
Hardware Version	**v2.2, June 2006**
Firmware Version	**v2.0, September 2003**
Data Sheet Version	**v2.1, October 2009**
Product Pages	CFA632 Serial & USB_List

Crystalfontz America, Incorporated
12412 East Saltese Avenue
Spokane Valley, WA 99216-0357

Phone: (888) 206-9720
Fax: (509) 892-1203
Email: techinfo@crystalfontz.com
URL: www.crystalfontz.com

REVISION HISTORY

HARDWARE	
2009/06/01	Current hardware version: **v2.2** New version number indicates RoHS compliant.
2003/09/01	Start Public Version Tracking. Hardware version: v2.1

FIRMWARE	
2003/09/01	Start Public Version Tracking. Current firmware version: **v2.0**

DATA SHEET	
2009/10/12	Current Data Sheet version: **v2.1** Changes since last released version: ● Wherever listed, changed hardware version from “v2.1” to “v2.2”. ● Added “RoHS compliant” to Features. See FEATURES (Pg. 5). ● Wherever needed, updated hypertext.
2005/10/15	Start Public Version Tracking. Data Sheet version: v2.0 Changes since last released version: Added Revision History (this page.) Updated Data Sheet formatting.

CONTENTS

CONTENTS, CONTINUED

LIST OF FIGURES

FEATURES

- ❑ 16x2 LCD has a large display area in a compact 108 mm x 43 mm (4.25" x 1.69") package.
- ❑ Large, easy-to-read characters: 7.3 mm (0.287") character height.
- ❑ 19200/9600/4800/2400 baud RS-232 interface.
- ❑ CFA632 is available in three popular choices.
  - ● CFA632-NFA-KS
    - ■ Reflective, no backlight.
    - ■ FSTN positive mode LCD (displays dark characters on light background—needs ambient light to be visible).
    - ■ Sunlight readable.
  - ● CFA632-YFB-KS
    - ■ Yellow-green array LED backlit with FSTN positive mode LCD (displays dark characters on light background).
    - ■ Sunlight readable.
  - ● CFA632-YMC-KS
    - ■ Yellow-green array LED backlit with STN blue negative mode LCD (displays illuminated yellow-green characters on dark background).
- ❑ Modules are thin:
  - ● CFA632-NFA (reflective) is 15.9 mm (0.626") including DB9 connector.
  - ● CFA632-YFB-KS and CFA632-YMC-KS (LED backlit) are 19.9 mm (0.783") including DB9 connector.
- ❑ Software controlled contrast.
- ❑ The CGROM is an enhanced version with many useful characters: numeric superscripts, icons, mathematical symbols, some fractions, a great variety of arrows, many currency symbols, and more.
- ❑ LCD characters are contiguous horizontally to allow the host software to display “gapless” bar graphs in horizontal direction.
- ❑ Software controlled terminal style automatic scrolling and line wrapping.
- ❑ Unique scrolling marquee feature continuously scrolls a message across the display without host intervention.
- ❑ Low speed "SPI" ($F_{clk\ max}$ = 18 KHz) and "Inverted TTL" RS-232 for embedded applications.
- ❑ Buffered communication.
- ❑ Single board construction—no connectors to fail.
- ❑ Optional 5.25" half-height drive-bay CFA632 Serial / USB LCD Mounting Bracket available.
- ❑ Nonvolatile memory capability (EEPROM) to customize the “power-on” display settings. All features of the display are controlled: characters displayed, user-definable characters, backlight setting, contrast setting, cursor position, cursor style, “wrap” setting, “scroll” setting, and even the scrolling marquee.
- ❑ Extended voltage (9-15 volts) and 5 volts power inputs standard on all modules for the controller. Backlight is 5 volts only.
- ❑ DB9 connector provided for simple RS-232 hookup.
- ❑ Low power: non-backlight operation will self-power from the DTR and RTS lines of most serial ports.
- ❑ RoHS compliant.

SYSTEM BLOCK DIAGRAM

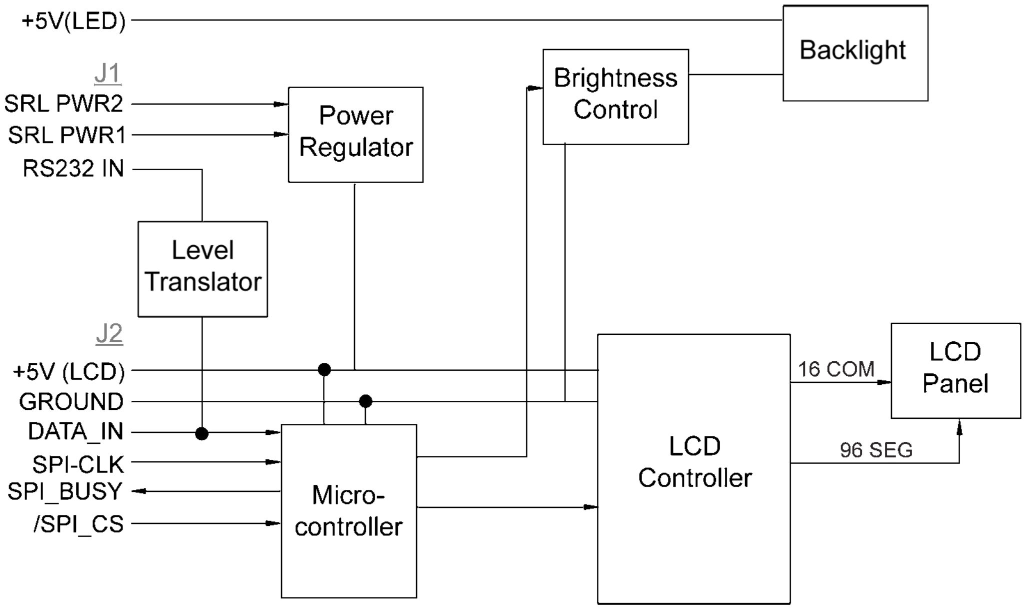


Figure 1. CFA632 System Block Diagram

MECHANICAL CHARACTERISTICS

ITEM	SIZE (mm)
Overall Size	108 (W) x 43 (H)
Viewing Area	82.3 (W) x 22.5 (H)
Active Area	74.82 (W) x 15.8 (H)
Character Size	3.9 (W) x 6.38 (H)
Dot Size	0.72 (W) x 0.86 (H)
Dot Pitch	0.78 (W) x 0.92 (H)
Depth*	CFA632-NFA-KS: 15.9 CFA632-YFB-KS: 19.9 CFA632-YMC-KS: 19.9

*Includes DB9 connector

GENERAL SPECIFICATIONS

All CFA632 models have these specifications:

CRITERIA	SPECIFICATION
Backlight PWM Frequency	300 Hz nominal
Driving Method	1 / 32 Duty, 1 / 6.7 Bias
Operating Temperature Range	0°C minimum to +50°C maximum
Storage Temperature Range	-10°C minimum to +60°C maximum
Viewing Direction	6 o'clock
Weight	60 grams (typical)

These CFA632 specifications vary by model:

PART NUMBER	FLUID	LCD GLASS COLOR	IMAGE	POLARIZER FILM	BACKLIGHT
CFA632-NFA-KS	FSTN	gray	positive	reflective	none
CFA632-YFB-KS	FSTN	gray	positive	transflective	yellow-green array LED
CFA632-YMC-KS	STN	blue	negative	transmissive	yellow-green array LED

ELECTRICAL SPECIFICATIONS

Required Voltage:

Logic (+5 volts):
Supplied through J2: 4.75 volts minimum, 5.0 volts nominal, 5.25 volts maximum
Supplied through DB9 connector using the on-board regulator: 9 volts minimum, 15 volts maximum

Backlight: (+5 volts): 4.75 volts minimum, 5.0 volts nominal, 5.25 volts maximum

Current Consumption:

Overall module current (LCD + backlight at 100%): 380 mA
Overall module current (LCD + backlight at 0%): 9 mA

ESD (Electro-Static Discharge) Specifications:

The circuitry is industry standard CMOS logic and susceptible to ESD damage. Please use industry standard anti-static precautions as you would for any other PCB such as expansion cards or motherboards. For more information, read CARE AND HANDLING PRECAUTIONS (Pg. 27).

RELIABILITY

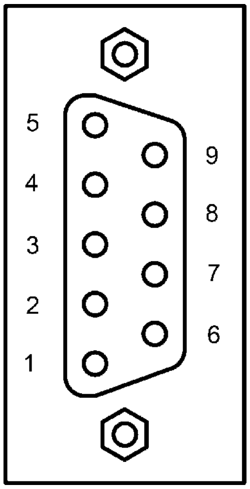
ITEM	SPECIFICATION
LCD Module (excluding the Backlight)	50,000 to 100,000 hours (typical)
Yellow-green LED Backlight	50,000 to 100,000 hours (typical)

RS-232 POWER AND SIGNAL CONNECTIONS TO A PERSONAL COMPUTER

FOR NON-BACKLIGHT OPERATION (THROUGH 9-PIN SERIAL PORT)

For non-backlight operation when the module is connected to a PC’s 9-pin serial port, all you need is a “straight through” RS-232 9-pin female DB9 to 9-pin female DB9 cable. Three cable lengths are available from Crystalfontz: part number WR232Y01 (6-foot), WR232Y04 (10-foot), and WR232Y10 (13-inch). A 6-foot serial RS-232 cable is also available through Radio Shack catalog number 26-152.

This is a view looking into the male DB9 connector (J1) on the CFA632:

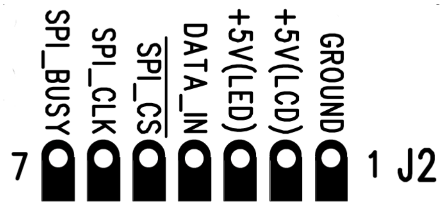


PIN	CRYSTALFONTZ DISPLAY FUNCTION	CORRESPONDING PC PIN NAME
1	Not Connected	DCD (Data Carrier Detect)
2	Not Connected	Rx (Receive Data)
3	Data In	Tx (Transmit Data)
4	Power A (9 volts to 15 volts)	DTR (Data Terminal Ready)
5	Ground (VSS)	Signal Ground
6	Connected to Power A if JPD is closed	DSR (Data Set Ready)
7	Power B (9 volts to 15 volts)	RTS (Request To Send)
8	Connected to Power B if JPC is closed	CTS (Clear To Send)
9	Not Connected	RI (Ring Indicator)

Most RS-232 ports will be able to power the display through their DTR and RTS lines. Have the software drive these lines high—most software already will.

TO POWER THE BACKLIGHT

To power the backlight, you must connect a regulated 5 volt supply capable of sourcing 400 mA to the LED+ terminal of the LCD's solder connector (J2). The supply's ground should connect to the VSS terminal of the LCD's solder connector (J2):



PIN	PIN NAME	CRYSTALFONTZ DISPLAY FUNCTION
1	GROUND	Ground (backlight and controller)
2	+5V(LCD)	Controller and LCD power (+5 volts only)
3	+5V(LED)	LED Backlight power (+5 volts only)
4	DATA_IN	SPI or RS-232 data in (input)
5	$\overline{\text{SPI\_CS}}$	SPI Chip Select (active low input)
6	SPI_CLK	SPI Clock (input)
7	SPI_BUSY	SPI Busy (output)

The easiest place in a PC to get +5 volts is from a disk drive power connector. The red wire is typically 5 volts and the black wires are typically ground. Crystalfontz sells a cable (part number WR232Y02) that will connect from a spare power connector on your PC to the display's LED backlight.

NOTE

If you make your own cable, be sure to measure the output voltage before you connect the display. The colors are usually correct but some computers may have unconventional colors.

NOTE

Do not use the yellow wire of the disk drive power cable. This wire is typically the 12 volt supply and will ruin the display and/or backlight if it is connected instead of 5 volts.

RS-232 POWER AND SIGNAL CONNECTIONS IN EMBEDDED SYSTEMS

FOR NON-BACKLIGHT OPERATION

To connect the display in most embedded systems, you only need to use the LCD's solder connector (J2). This connector has standard 0.1 inch spacing. Generally, you would connect a regulated 5 volt supply for the controller to +5V(LCD) and +5V(LED), GROUND to GROUND, and RS-232 data to DATA_IN. DATA_IN will accept full +10 volt to –10 volt swing RS-232 signals and will also accept 0 volt to 5 volt "CMOS" or "TTL" levels (close JPE). If your RS-232 data is inverted, which is typical of a microcontroller "UART Tx" pin, you can close JPB.

If you would like, you can connect an unregulated 9 volt to 15 volt supply to Power A or Power B of the DB9 connector instead of connecting a regulated +5 volt supply to VCC. Power A and Power B will only power the LCD itself, not the backlight.

NOTE

Do not connect a supply greater than 5.25 volts to VDD or LED+. If you do, damage to your LCD module will result.

INTERFACE OPTIONS USING "INVERTED RS-232" OR "LOW-SPEED SPI"

"Inverted RS-232" or "Low-Speed SPI" may be selected by closing jumpers.

Inverted RS-232

Inverted RS-232 is useful when the display is used with an embedded microcontroller's built-in UART. These UARTs typically output a logic level (0-5 volts) inverted version of the RS-232 waveform, ready to be passed into an RS-232 driver. By setting the display to accept inverted RS-232 data, you can connect this logic level signal directly to the display and avoid having the RS-232 driver. The display will show a lowercase "`i`" before the baud rate on the information screen if the jumper is closed ("`i9600bd`").

Low-Speed SPI

To use the SPI interface, close JPA. Please refer to APPENDIX A: SPI TIMING DIAGRAMS (Pg. 29).

Low-Speed SPI is intended to be used with low-end embedded microprocessors that lack a hardware UART. The reason that it is "Low-Speed" is that the SPI is done in firmware on the display. On the host microprocessor, any three general-purpose output ports and a small "send byte" routine can be used to control the display. If the host microprocessor has a hardware SPI port, that can be used, provided that the port's speed can be set to satisfy the timing constraints. A software state-machine and a timer interrupt could also be used, this would reduce the microprocessor's load while still satisfying the timing requirements.

Another useful feature of SPI is that additional displays can be controlled with only one additional output port ($\overline{\text{SPI\_CS}}$) per display. The other lines (SPI_CLK and SPI_DATA) are common. The display will show "SPI" instead of the baud rate on the information screen.

The CFA632 has a 64-character input buffer. For the RS-232 interface, it is nearly impossible to overflow this buffer since the display can process commands more quickly than the 19200-baud RS-232 interface can deliver them, so normally no flow control is needed. The exceptions are the routines that access the EEPROM (the \009 series) and a very rare combination of commands that take a long time to execute, followed by a burst of characters that is larger than the input buffer.

If your processor can deliver data through the Low-Speed SPI interface at a rate faster than 1000 bytes/second, then the processor should make sure the SPI_BUSY line is low before sending a new command. The SPI_BUSY line will be set “high” by the display when there are 32 or more characters in the display's input buffer, and returned “low” when there are less than 32 characters.

TO POWER THE BACKLIGHT

To use the backlight, connect a regulated +5 volt supply capable of sourcing 400 mA to the +5V(LED) terminal. This may be the same supply that is used for +5V(LCD). The backlight brightness is controlled by PWM (Pulse Width Modulation). The PWM may cause noise on supplies that have high output impedance. If this noise causes problems in your application, you may want to connect a filter capacitor from +5V(LCD) to GROUND close to the display or operate the backlight at full brightness (which disables the PWM).

HOST COMMUNICATIONS

The CFA632 communicates with its host using an RS-232 interface. The baud rate is set with the DIP switches on the back of the display. Please refer to the legend silk-screened on the back of the module for the settings.

CFA632 DEMONSTRATION AND TEST PROGRAMS

The following programs are available for free download on our website:

❑ WinTest Serial LCD Demo Software
This website page has a link to download a demonstration and test program, WinTest.exe. The C source code is included. The program demonstrates software techniques for displaying banners, graphs, large digits, spectrum, and oscilloscope. Example commands may be copied from the Control Codes listed below and pasted into WinTest. (See Description of Control Functions (Pg. 13).

❑ 632/634 BasicStamp2 Sample
This website page has a link to download a simple program that demonstrates communications from a Parallax Basic Stamp. BS2 source code is available.

❑ 632/634 Linux-Compatible Command-Line Demonstration Program
This is a link to a Linux-compatible demonstration program. The C source code is included.

ABOUT HANDSHAKING

❑ The display can generally execute commands faster than the RS-232 serial interface can deliver them so no handshaking is necessary from the display’s point of view.

❑ The few exceptions involve accessing the EEPROM (For example, see Control the Boot Screen (\009, Control+I) (Pg. 13)).

❑ Some host hardware or software may require the display to assert its DSR and/or CTS lines before it will send data to the display. In that case, JPC and/or JPD may be closed. Closing JPC will connect DSR to DTR through a 1 K ohm resistor. Closing JPD will connect CTS to RTS through a 1 K ohm resistor.

CONTROL CODES AND FUNCTIONS

The CFA632 will accept plain ASCII characters and display them on the screen at the current cursor position. For example, if you send “Hello World”, the display shows “Hello World”. The display also supports a set of control characters that access other features of the display. Some of these commands require one or more parameters. In general, the parameters are “binary” characters, meaning that any value between 0 and 255 may be required.

The notation "\xxx" is used for “binary” data, where "xxx" is the decimal representation of the number. Use the codes \000 to \255 to include all possible values for a character. This notation is supported by the WinTest Serial LCD Demo Software. We recommend you try this utility by using the examples listed under Description of Control Functions (Pg. 13). Select a line of text by highlighting it, copy it, then paste it 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.

Below is a list of the control codes. Click on the links under the ASCII column to jump to an explanation of the code.

ASCII	KEYSTROKES	CONTROL CODE FUNCTION
\000	Control+@	Reserved
\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	Control the Boot Screen
\010	Control +J or Control +Enter	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	(Reserved)
\017	Control+Q	Set Cursor Position (Column and Row)
\018	Control+R	Horizontal Bar Graph
\019	Control+S	Scroll ON

ASCII	KEYSTROKES	CONTROL CODE FUNCTION
\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	(Reserved)
\029	Control+Quote	(Reserved)
\030	Control+Equal	Send Data Directly to LCD Controller
\031	Control+Minus	Show Information Screen
\128		Custom Character 0
\129		Custom Character 1
\130		Custom Character 2
\131		Custom Character 3
\132		Custom Character 4
\133		Custom Character 5
\134		Custom Character 6
\135		Custom Character 7

DESCRIPTION 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 nonblinking 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 top row to the right-most column of the bottom row.

Control the Boot Screen (\009, Control+I)

This command allows the current state of the display to be stored in the display's EEPROM, recalling of the EEPROM contents to the display, and controlling the boot behavior.

All features of the display are controlled: the characters displayed, the bitmaps of the user-definable characters, the backlight setting, the contrast setting, the cursor position, the cursor style, the wrap setting, the scroll setting, and even the scrolling marquee's data and state.

NOTE

Since writing and reading the EEPROM takes quite a bit time, it is possible to overflow the display's input buffer if data is continuously sent while the display is busy executing these commands. The display will still buffer data while these commands are being executed. However, the buffer is 64 bytes long and can be overflowed. You must take care to not overflow the input buffer while the EEPROM commands are executing.

Send "Control+I" followed by the command.

\009 \000 Set boot to “backward compatibility mode”

Execution time: 5.1 mS

If JPK is open: Crystalfontz boot screen is displayed. The boot screen will clear automatically after about 5 seconds. If a character is received during that 5 seconds, the boot screen will be cleared immediately, then that character will be processed.

If JPK is closed: No logo is shown.

\009 \001 Set boot to show Crystalfontz logo until a character arrives

Execution time: 5.1 mS

The Crystalfontz boot screen is displayed. When a character is received, the boot screen will be cleared immediately and the character will be processed.

\009 \002 Set boot to show User Boot Screen for 5 seconds

Execution time: 5.1 mS

The User Boot Screen is recalled from the EEPROM and shown on the display. The boot screen will clear automatically after about 5 seconds. If a character is received during that 5 seconds, the boot screen will be cleared immediately, then that character will be processed.

\009 \003 Set boot to show User Boot Screen until a character arrives

Execution time: 5.1 mS

The User Boot Screen is recalled from the EEPROM and shown on the display. When a character is received, the boot screen will be cleared immediately, then that character will be processed.

\009 \004 Set boot to show User Boot Screen and leave it displayed

Execution time: 5.1 mS

The User Boot Screen is recalled from the EEPROM and shown on the display. Incoming characters are processed normally without the screen being cleared. Settings for display blank, cursor style and position, WRAP, SCROLL are maintained.

\009 \005 Save current display state to EEPROM

Execution time: 800 mS

The entire display state is saved to the EEPROM.

\009 \006 Recall current display state from EEPROM

Execution time: 91 mS

The entire display state is recalled from the EEPROM

Line Feed (\010, Control+J or Control+Enter)

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.

NOTE

If SCROLL is set, the cursor is placed on the bottom line, several Line Feed characters are sent (forcing the display to scroll), and these Line Feeds characters are immediately followed by a burst of more than 64 additional characters—it is possible to overflow the display's input buffer. Don't do this. Contrary to design standards for internet server software from our favorite software monopoly, the display will not start executing the input buffer data as if it were code. The display will simply overwrite the oldest data in the input buffer with the most recently received data.

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.

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 50 is typical 100 = very dark

Examples:

```
\015\050
\015\060
\015\070
```

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

Send "Control+Q" followed by one byte for the column (0-15), and a second byte for the row (0-1). The upper-left position is 0,0. The lower-right position is 15,1.

For example, to move 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 custom 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.

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

Any value is valid between \000 and \255. The MSB (Most Significant Bit) is at the top of the row, the LSB (Least Significant Bit) 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 \015. ***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*** is 16 x 6 = \096.

row is the character Y coordinate. \000-\001 is valid.

Examples:

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

NOTE

The entire graph area is completely rewritten 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 - 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.

NOTE

The graphs use some of the custom characters and may corrupt the display contents 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 (Control+W), 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 (Control+W for Wrap ON, Control+X for Wrap OFF). If wrap is 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.

The examples below will set the first 12 characters to "Crystalfontz", put another message on the display, and enable the scrolling marquee. Please note that the second line ends in a space.

Examples:

```
\004\022\255\001\005\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\001\016
```

NOTE

A section of memory 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 and then enable rotation with Enable Scrolling Marquee command. The hidden characters are set to blanks at power-up, or loaded from the User Boot Screen.

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.

WinTest, available at WinTest Serial LCD Demo Software easily allows setting up the scrolling marquee and adjusting the parameters, so you can find a setting you like in a minimum amount of time.

NOTE

If the scrolling banner is saved as the default boot state, one of the characters will be overwritten with a blank. The character overwritten depends on the position of the banner when the Control the Boot Screen (\009, Control+I) (Pg. 13) command is sent. This error affects all firmware versions.

Enable Scrolling Marquee (\022, Control+V)

Send "Control+V" followed by the following bytes:

line
scroll_step_size
update_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
`\255`	disable scrolling marquee (valid values for *scroll_step_size* and *update_speed* must still be sent)

scroll_step_size controls the number of pixels that the message is shifted by at each update:

scroll_step_size	**RESULT**
`\001`	shift by one pixel, smooth but slow
`\002`	shift by two pixels
`\003`	shift by three pixels
`\004`	shift by four pixels
`\005`	shift by five pixels
`\006`	shift by six pixels (equal to shifting by one character, fast)

update_speed determines how often updates will occur. The units are 1/96 of a second (about 10 mS). The valid range is \005 (52 mS) to \100 (1.042 S).

NOTE

Since the liquid crystal fluid in the display takes some time to react, the minimum usable value is about 16 or 167 mS. The Enable Scrolling Marquee command supports a large range of speeds to accommodate future displays and user preference.

The following equations will allow you to determine the speed at which the message scrolls:

Update Frequency = ***update_speed*** / 96 Hz
Update Period = 96 / ***update_speed*** Seconds

New Character Frequency = (***scroll_step_size x update_speed***) / (96 x 6) Hz
New Character Period = (96 x 6) / (***scroll_step_size x update_speed***) Seconds

Message Repeat Period = (36 x 96 x 6) / (***scroll_step_size x update_speed***) Seconds

See the example in Set Scrolling Marquee Characters (\021, Control+U) (Pg. 17). WinTest, available at WinTest Serial LCD Demo Software, easily allows setting up the scrolling marquee and adjusting the parameters so you can find a setting you like in a minimum amount of time.

Wrap ON (\023, Control+W)

Turns wrap feature on. When wrap is on, 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. When wrap is off, 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. The custom characters are displayed by sending \128 to \135:

character	**CUSTOM CHARACTER MODIFIED**
\128	0
\129	1
\130	2
\131	3
\132	4
\133	5
\134	6
\135	7

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 and 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.

NOTE

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.

Examples:

```
    \012\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)

CFA632 firmware is very stable and robust, and it is unlikely that you will ever need a "reboot" command. If the firmware did crash, the command processor would most likely be inoperable and unable to detect the reboot command. However, you may have certain situations where it is nice to have a command that will return the display to a known state. For instance, if the baud rate on the host is set to an incorrect speed, the data is interpreted as meaningless garbage which the display firmware tries to interpret. Some data may set the contrast to an unusable value; some data may program the LCD controller to an indeterminate state. Or perhaps you always want the display to wake up in a given state when your program starts, without going through all the commands that affect the way the display interprets commands (such as the state of Scroll or Wrap, for instance).

Send one "Control+Z" followed by another "Control+Z" to reboot the display. If you are not sure of the display state, it may be necessary to send up to 9 characters to satisfy the parameters of some previous command. For example, if the Set Custom Character Bitmap command (See Set Custom Character Bitmap (\025, Control+Y) (Pg. 20)) has just been received by the display when an application crashes, the display will interpret the next 9 bytes as the parameters to the command, then wait for more commands. If you are not sure what the status of the display is, send 9 blanks (`\032`) followed by two “Control+Z”s (`\026).`

If the display in your system is powered by the serial port's RTS and DTR lines, the display can be rebooted by dropping those lines momentarily (for example, 500 mS) and then bringing them high again.

Escape Sequence Prefix (\027, Escape)

Four escape sequences are supported. These correspond to the escape sequences that are sent for the four arrows keys in HyperTerminal with an ANSI terminal selected (and also our WinTest). These sequences move the cursor only and do not wrap.

ESCAPE SEQUENCE	ARROW KEY
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

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.

NOTE

This command executes a low level write directly to the controller. Use this command at your own risk. "Control+Z" followed by another "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.

CHARACTER GENERATOR ROM (CGROM)

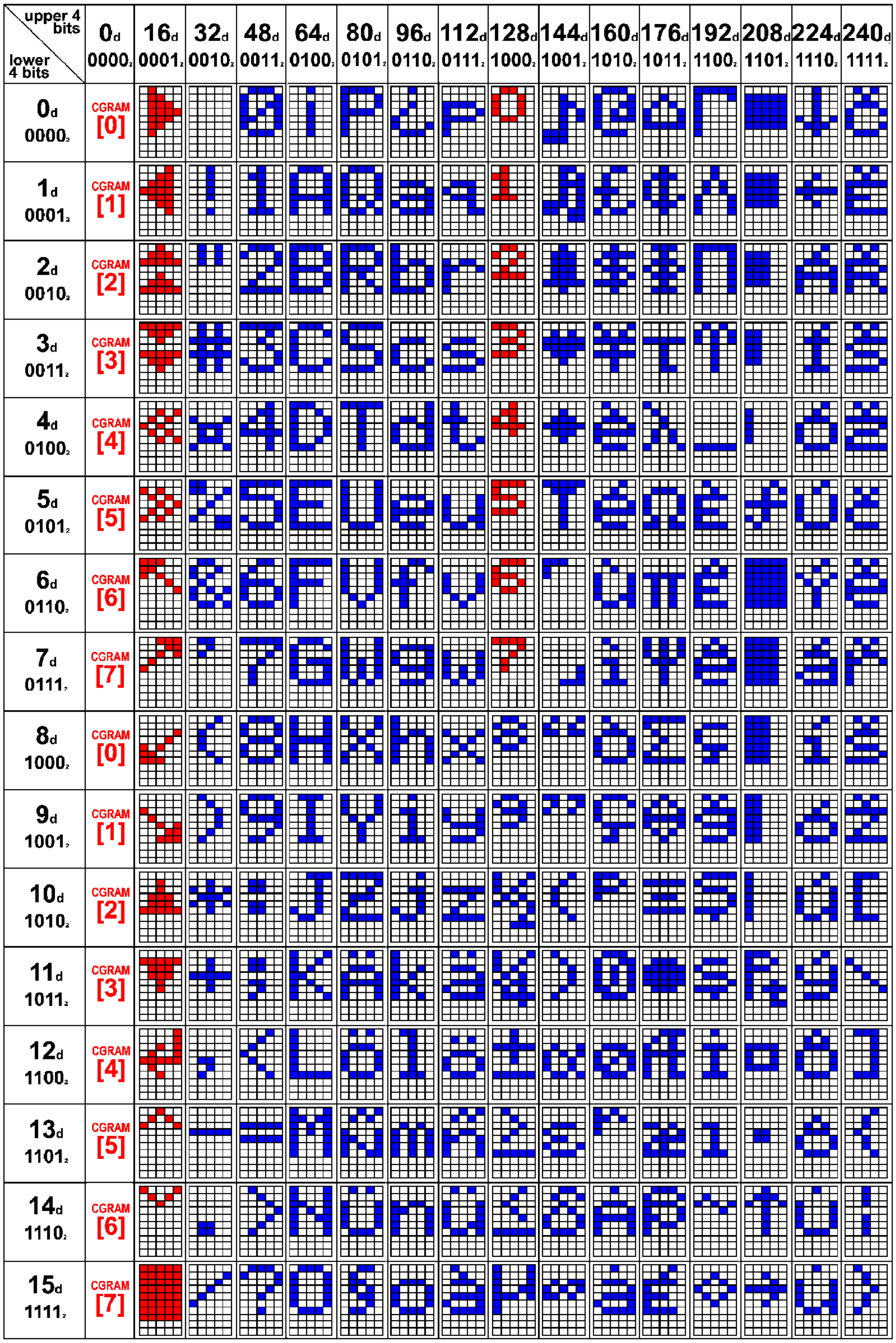


Figure 2. CFA632 Character Generator ROM (CGROM)

HOW TO USE CGROM

The CGROM defines which characters are shown by the display for a given code received through the serial port. The CFA632 firmware v2.0 has an enhanced CGROM that includes many useful special characters (including numeric superscripts, icons, mathematical symbols, some fractions, a great variety of arrows, many currency symbols. . .).

Most of the characters can be accessed by sending the appropriate ASCII code to the display. For instance, the letter “A” can be shown by sending an “`A`" (which is the same as a "`decimal 65`", a "`hex 0x41`", or a “`\065`”).

Some characters do not have an obvious match. For instance, the code to display a superscript “9” is "`decimal 137`", "`hex 0x89`", or “`\137`”. The relationship between the codes and the characters are shown on in the CFA632 Character Generator ROM (CGROM) (Pg. 23)

To find the code for a given character, add the two numbers that are shown in bold for its row and column. For example, the superscript “9” is in the column labeled "128d" and in the row labeled "9d". So you would add 128 + 9 to get 137. When you send a byte with the value of 137 to the display, then a superscript “9” will be shown at the current cursor position.

In the CFA632 v1.0, the CGROM table had several columns that did not have any characters assigned to the codes. The commands to control the display were mapped to some of those unused codes. These codes are shown in red in the table. Specifically, the first 32 codes (0 to 31) are reserved for the display's special functions (cursor positioning, contrast control, bar graphs. . .), and the codes from 128 to 135 are used to access the custom characters.

To access the characters shown in red, it is necessary to send some data directly to the LCD controller. Send Data Directly to the LCD Controller (\030, Control+Equal) (Pg. 22) allows any character in the CGROM to be displayed.

For example, if you want to display an arrow that points to the upper left, look at the table and note that it is in column "16d", and in row "6d", so its code is 22 (16 + 6 = 22). Use the following sequence to display the arrow that points to the upper left:

```
\030\001\022
```

This sequence of three bytes must be used to display any of the characters shown in red in the table. It can also be used to access any character in the table.

CFA632 MODULE OUTLINE DRAWING

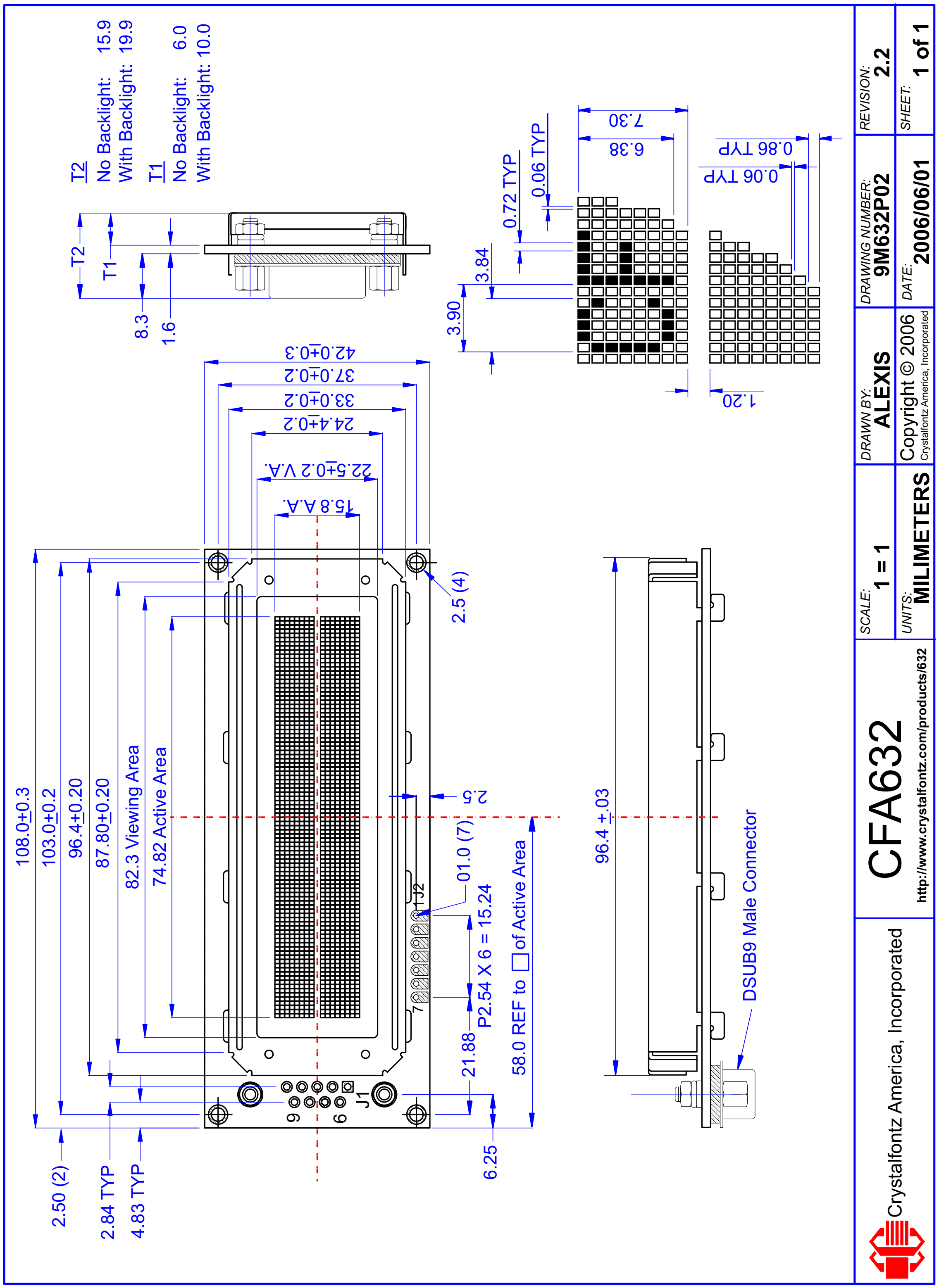
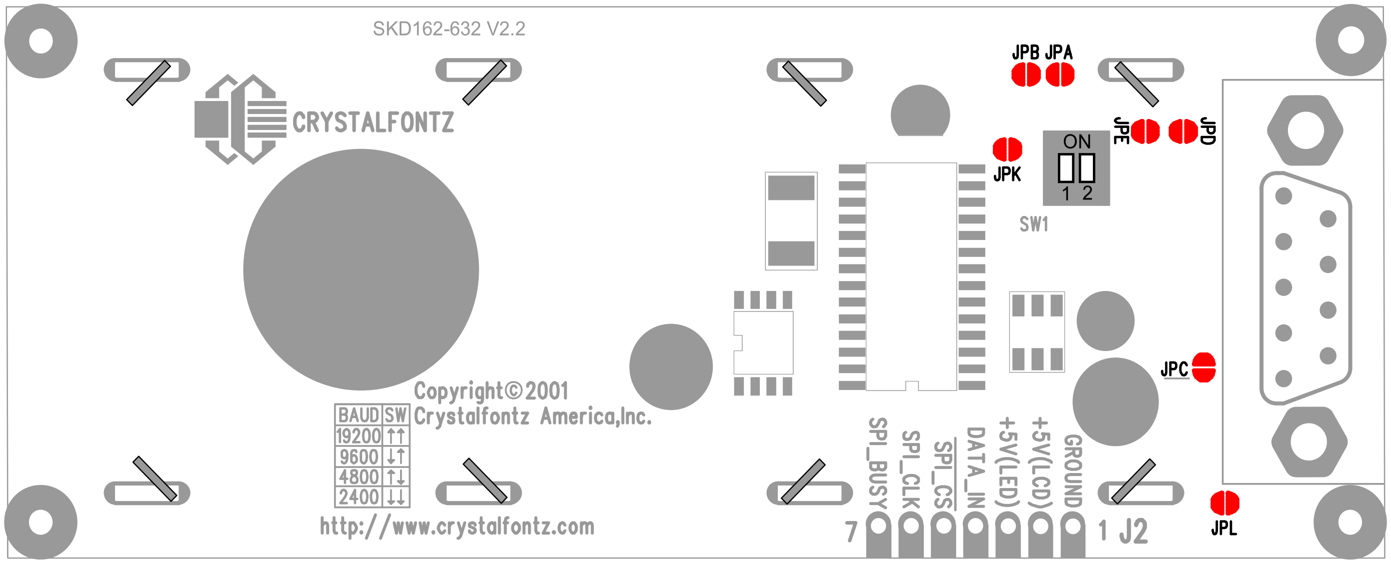


Figure 3. CFA632 Module Outline Drawing

JUMPER LOCATIONS AND FUNCTIONS

The CFA632 has seven jumpers. The jumpers are normally open. The jumpers may be closed by melting a ball of solder across their gap.You may re-open the jumpers by removing the solder. (Solder wick works well for this.)

CFA632 Hardware v2.2 Jumper Locations and Functions



JPA	**open**	RS-232 interface selected
	closed	SPI interface selected
JPB	open	RS-232 data is normal
	closed	RS-232 data is inverted
JPC	**open**	DSR is open
	closed	DTR is fed back on DSR through 1 K resistor
JPD	open	CTS is open
	closed	RTS is fed back on CTS through 1 K resistor
JPE	**open**	DATA_IN is RS-232 (-10 v to +10 v swing)
	closed	DATA_IN is 0 v to 5 v swing
JPK	open	Show splash screen
	closed	Disable splash screen
JPL	open	Frame Ground* is open
	closed	Frame Ground is connected to Signal/Power supply GROUND

*Frame Ground is a trace connecting the mounting holes, bezel, and shell of the DB9 (J2).

BAUD	SW
19200	↑↑
9600	↓↑
4800	↑↓
2400	↓↓

Crystalfontz America, Incorporated	CFA632 http://www.crystalfontz.com/products/632	SCALE: 1 = 1	DRAWN BY: ALEXIS	DRAWING NUMBER: 9M632P04	REVISION: 2.2
		UNITS: MILIMETERS	Copyright © 2006 Crystalfontz America, Incorporated	DATE: 2006/06/01	SHEET: 1 of 1

Figure 4. CFA632 Jumper Locations and Functions

CARE AND HANDLING PRECAUTIONS

For optimum operation of the CFA632 and to prolong its life, please follow the precautions described below.

ELECTROSTATIC DISCHARGE (ESD)

Please use industry standard antistatic precautions as you would for any other PCB such as expansion cards or motherboards. Ground your body, work surfaces, and equipment.

DESIGN AND MOUNTING

- If you are not using a Crystalfontz overlay, place a transparent plate (for example, acrylic, polycarbonate, or glass) in front of the CFA632, leaving a small gap between the plate and the display surface. We use GE HP-92 Lexan, which is readily available and works well.
- Do not disassemble or modify the CFA632. Solder only to the I/O terminals.
- Do not reverse polarity to the power supply connections. Reversing the polarity will immediately ruin the module.

AVOID SHOCK, IMPACT, TORQUE, AND TENSION

- Do not expose the CFA632 to strong mechanical shock, impact, torque, and tension.
- Do not drop, toss, bend, or twist the CFA632.
- Do not place weight or pressure on the CFA632.

IF LCD PANEL BREAKS

- If the LCD panel breaks, be careful to not get the liquid crystal fluid in your mouth or eyes.
- If the liquid crystal fluid touches your skin, clothes, or work surface, wash it off immediately using soap and plenty of water.
- Do not eat the LCD panel.

CLEANING

- To clean the front of the LCD, a standard household glass cleaner works well. Gently wipe with a nonabrasive soft cloth.
- CFA632 *without Crystalfontz overlay:* The exposed surface of the LCD “glass” is actually the front polarizer laminated to the glass. The polarizer is made out of a fairly soft plastic and is easily scratched or damaged. The polarizer will eventually become hazy if you do not take great care when cleaning it. Long contact with moisture (from condensation or cleaning) may permanently spot or stain the polarizer.

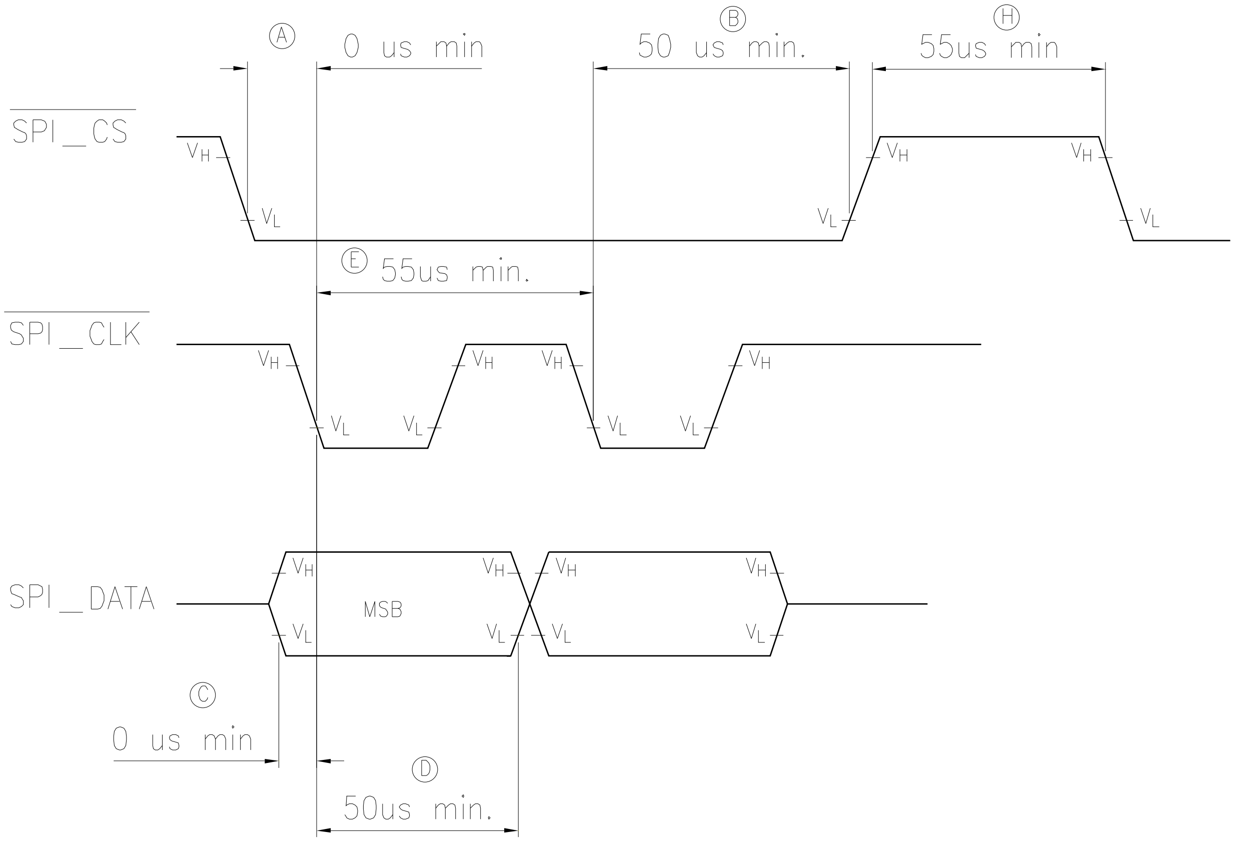
OPERATION

- 0°C minimum, 50°C maximum with minimal fluctuation. Operation outside of these limits may shorten life and/or harm display.
  - At lower temperatures of this range, response time is delayed.
  - At higher temperatures of this range, display becomes dark. (You may need to adjust the contrast.)
- Operate away from dust, moisture, and direct sunlight.

STORAGE

- Store in an ESD-approved container away from dust, moisture, and direct sunlight.
- -10°C minimum, 60°C maximum with minimal fluctuation. Rapid temperature changes can cause moisture to form, resulting in permanent damage.
- Do not allow weight to be placed on the CFA632s while they are in storage.

APPENDIX A: SPI TIMING DIAGRAMS



Appendix A Figure 1. CFA-632 SPI Timing Diagrams

A $\overline{\text{SPI\_CS}}$ must be low before $\overline{\text{SPI\_CLK}}$ falls

B $\overline{\text{SPI\_CS}}$ must stay low for the entire transfer

C SPI_DATA must be correct before $\overline{\text{SPI\_CLK}}$ falls

D SPI_DATA must be held for a minimum of 50 us after $\overline{\text{SPI\_CLK}}$ falls

E The maximum clock rate is 1/55 us or 18.182 kHz

F Data is transferred MSB first; two out of eight cycles are shown.

G CPOL = 1, CPHA = 1 for most processors

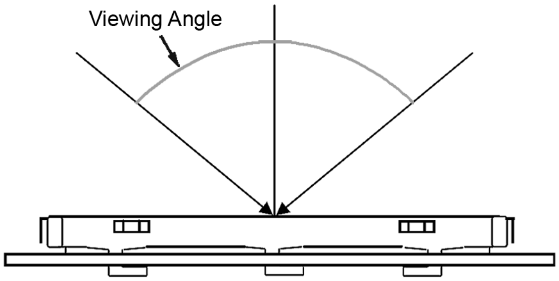
H Take $\overline{\text{SPI\_CS}}$ high for a minimum of 55 us to guarantee bit synchronization.

I Maximum throughput is 1/(9 x 55) = 2020 bytes / second

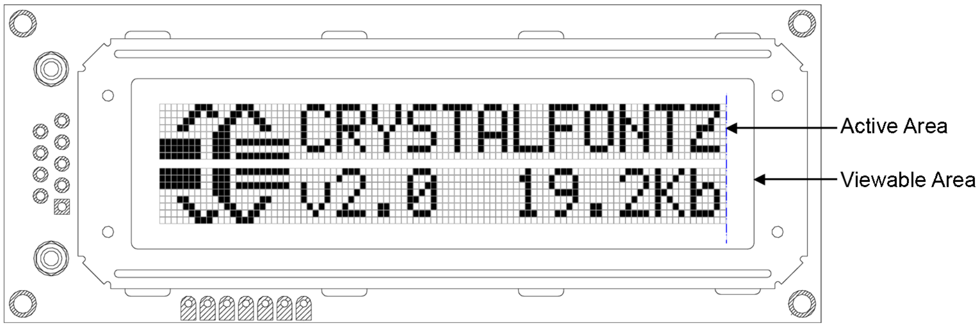
APPENDIX B: QUALITY ASSURANCE STANDARDS

INSPECTION CONDITIONS

- Environment
  - Temperature: 25±5°C
  - Humidity: 30~85% RH
- For visual inspection of active display area
  - Source lighting: two 20 Watt or one 40 Watt fluorescent light
  - Display adjusted for best contrast
  - Viewing distance: 30±5 cm (about 12 inches)
  - Viewable angle: inspect at 45° angle of vertical line right and left, top and bottom



DEFINITION OF ACTIVE AREA AND VIEWABLE AREA



ACCEPTANCE SAMPLING

DEFECT TYPE	AQL*
Major	≤0.65%
Minor	≤1.00%

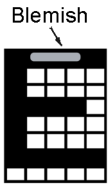
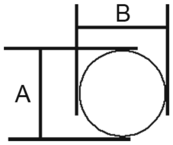
*Acceptable Quality Level: maximum allowable error rate or variation from standard

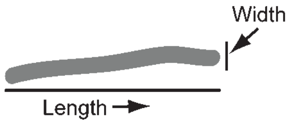
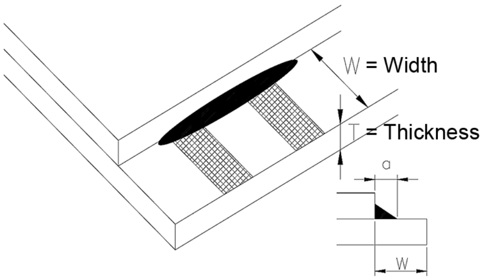
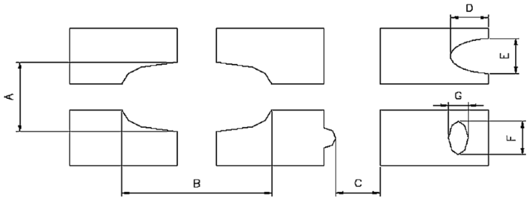
DEFECTS CLASSIFICATION

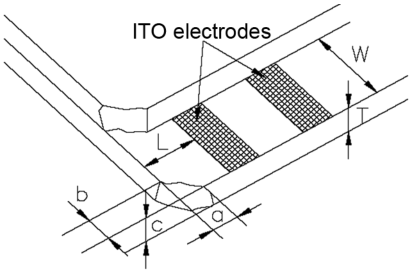
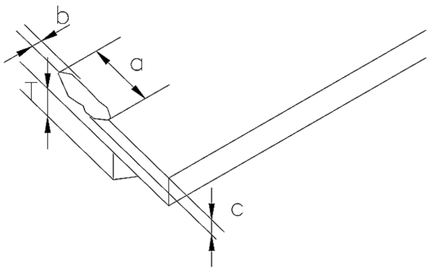
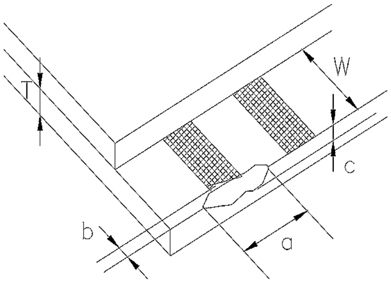
Defects are defined as:

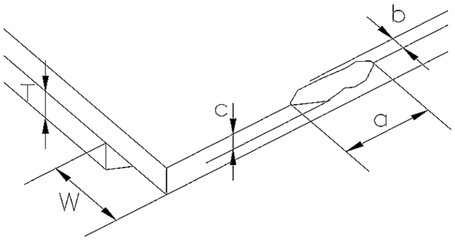
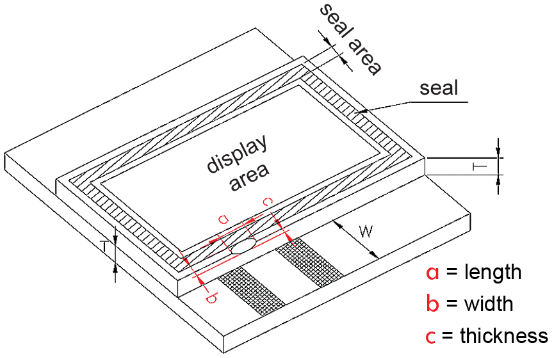
- Major Defect: results in failure or substantially reduces usability of unit for its intended purpose
- Minor Defect: deviates from standards but is not likely to reduce usability for its intended purpose

ACCEPTANCE STANDARDS

<table>
<tr><th>#</th><th>DEFECT TYPE</th><th colspan="3">CRITERIA</th><th>MAJOR / MINOR</th></tr>
<tr><td>1</td><td>Electrical defects</td><td colspan="3">1. No display, display malfunctions, or shorted segments.
2. Current consumption exceeds specifications.</td><td>Major</td></tr>
<tr><td>2</td><td>Viewing area defect</td><td colspan="3">Viewing area does not meet specifications. (See Inspection Conditions (Pg. 30).</td><td>Major</td></tr>
<tr><td>3</td><td>Contrast adjustment defect</td><td colspan="3">Contrast adjustment fails or malfunctions.</td><td>Major</td></tr>
<tr><td rowspan="3">4</td><td rowspan="3">Blemishes or foreign matter on display segments</td><td rowspan="3"> </td><td><i>Defect Size (mm)</i></td><td><i>Acceptable Qty</i></td><td rowspan="3">Minor</td></tr>
<tr><td>≤0.3</td><td>3</td></tr>
<tr><td colspan="2">≤2 defects within 10 mm of each other</td></tr>
<tr><td rowspan="5">5</td><td rowspan="5">Other blemishes or foreign matter outside of display segments</td><td rowspan="5">Defect size = (A + B)/2
 </td><td><i>Defect Size (mm)</i></td><td><i>Acceptable Qty</i></td><td rowspan="5">Minor</td></tr>
<tr><td>≤0.15</td><td>Ignore</td></tr>
<tr><td>0.15 to 0.20</td><td>3</td></tr>
<tr><td>0.20 to 0.25</td><td>2</td></tr>
<tr><td>0.25 to 0.30</td><td>1</td></tr>
</table>

#	DEFECT TYPE	CRITERIA			MAJOR / MINOR
6	Dark lines or scratches in display area 	*Defect Width (mm)*	*Defect Length (mm)*	*Acceptable Qty*	Minor
		≤0.03	≤3.0	3	
		0.03 to 0.05	≤2.0	2	
		0.05 to 0.08	≤2.0	1	
		0.08 to 0.10	≤3.0	0	
		≥0.10	>3.0	0	
7	Bubbles between polarizer film and glass		*Defect Size (mm)*	*Acceptable Qty*	Minor
			≤0.20	Ignore	
			0.20 to 0.40	3	
			0.40 to 0.60	2	
			≥0.60	0	
8	Glass rest defect			a≤1/4W	Minor
9	Display pattern defect				Minor
		Dot Size (mm)	*Acceptable Qty*		
		((A+B)/2)≤0.2	≤3 total defects ≤2 pinholes per digit		
		C>0			
		((D+E)/2)≤0.25			
		((F+G)/2)≤0.25			

<table>
<tr><th>#</th><th>DEFECT TYPE</th><th colspan="4">CRITERIA</th><th>MAJOR / MINOR</th></tr>
<tr><td rowspan="3">10</td><td rowspan="3">Chip in corner</td><td colspan="4">
</td><td rowspan="3">Minor</td></tr>
<tr><td><i>a</i></td><td><i>b</i></td><td><i>c</i></td><td><i>Acceptable Qty</i></td></tr>
<tr><td>&lt;4 mm</td><td>≤W</td><td>c≤T</td><td>3</td></tr>
<tr><td rowspan="4">11</td><td rowspan="4">Chip on “non-contact” edge of LCD</td><td colspan="4">
</td><td rowspan="4">Minor</td></tr>
<tr><td><i>a</i></td><td><i>b</i></td><td><i>c</i></td><td><i>Acceptable Qty</i></td></tr>
<tr><td>≤3 mm</td><td>≤1 mm</td><td>≤T</td><td>Ignore</td></tr>
<tr><td>≤4 mm</td><td>≤1.5 mm</td><td>≤T</td><td>3</td></tr>
<tr><td rowspan="4">12</td><td rowspan="4">Chip on “contact” edge of LCD, on the active side</td><td colspan="4">
</td><td rowspan="4">Minor</td></tr>
<tr><td><i>a</i></td><td><i>b</i></td><td><i>c</i></td><td><i>Acceptable Qty</i></td></tr>
<tr><td>≤2 mm</td><td>≤W/4</td><td>≤T</td><td>Ignore</td></tr>
<tr><td>≤3 mm</td><td>≤W/4</td><td>≤T</td><td>3</td></tr>
</table>

#	DEFECT TYPE	CRITERIA				MAJOR / MINOR
13	Chip on “contact” edge of LCD, on the inactive side					Minor
		a	*b*	*c*	*Acceptable Qty*	
		≤3 mm	≤1 mm	≤T	Ignore	
		≤4 mm	≤1.5 mm	≤T	3	
14	Chip in seal area	 seal area seal display area T T W a b c a = length b = width c = thickness				
		a	*b*	*c*	*Acceptable Qty*	
		<3 mm	≤1.5 mm	≤1/2 T	3	Minor
		Unacceptable if c>50% of glass thickness or if the seal area is damaged.				Major
15	Backlight defects	1. Light fails or flickers. (Major) 2. Color and luminance do not correspond to specifications. (Major) 3. Exceeds standards for display’s blemishes or foreign matter (see test 5, Pg. 31), and dark lines or scratches (see test 6, Pg. 32). (Minor)				See list
16	COB defects	1. Pinholes >0.2 mm. 2. Seal surface has pinholes through to the IC. 3. More than 3 locations of sealant beyond 2 mm of the sealed areas.				Minor
17	PCB defects	1. Oxidation or contamination on connectors.* 2. Wrong parts, missing parts, or parts not in specification.* 3. Jumpers set incorrectly. (Minor) 4. Solder (if any) on bezel, LED pad, zebra pad, or screw hole pad is not smooth. (Minor) *Minor if display functions correctly. Major if the display fails.				See list

#	DEFECT TYPE	CRITERIA	MAJOR / MINOR
18	Soldering defects	1. Unmelted solder paste. 2. Cold solder joints, missing solder connections, or oxidation.* 3. Solder bridges causing short circuits.* 4. Residue or solder balls. 5. Solder flux is black or brown. *Minor if display functions correctly. Major if the display fails.	Minor