



## CFA634 Family of INTELLIGENT LCD MODULES SPECIFICATIONS



**Datasheet Release 2023-11-14**

Hardware Version: v4.0  
Firmware Version: v4.0

### **Crystalfontz America, Inc.**

12412 East Saltese Avenue  
Spokane Valley, WA 99216-0357  
Phone: 888-206-9720  
Fax: 509-892-1203  
Email: [support@crystalfontz.com](mailto:support@crystalfontz.com)  
URL: [www.crystalfontz.com](http://www.crystalfontz.com)



## **Table of Contents**

<b>1. General Information .....</b>	<b>5</b>
<b>2. Introduction .....</b>	<b>6</b>
2.1. Main Features.....	6
2.2. Module Classification Information .....	6
2.3. Color Variants.....	7
<b>3. Mechanical Characteristics .....</b>	<b>8</b>
<b>4. Electrical Specifications .....</b>	<b>9</b>
4.1. System Block Diagram .....	9
4.2. Absolute Maximum Ratings .....	10
4.3. LCD Duty and Bias .....	10
4.4. DC Characteristics.....	10
4.5. Typical Current Consumption .....	11
<b>5. Optical Characteristics .....</b>	<b>12</b>
5.1. Optical Characteristics Test Conditions and Definitions .....	12
5.2. LED Backlight Characteristics.....	12
<b>6. Interface Connection Information .....</b>	<b>13</b>
6.1. Interface Introduction.....	13
6.1.1. Module Interface Default Settings .....	13
6.2. Communications Interface Descriptions.....	14
6.2.1. USB Communications .....	14
6.2.2. RS-232 Full-Swing Serial Communications .....	14
6.2.3. Logic-Level Serial Communications .....	14
6.2.4. SPI Interface .....	14
6.2.5. I2C Interface .....	15
6.3. USB Connector Details.....	16
6.4. DB9 Connector and Power Details .....	16
6.5. J2 Connector and Power Details .....	17
6.6. I2C Connector and Power Details.....	18
<b>7. Interface Configuration Menu.....</b>	<b>19</b>
<b>8. Alternate Method to Power the Backlight for Non-USB Interfaces .....</b>	<b>20</b>
8.1. To Power the Backlight Using J2 Expansion Port Pins .....	20
<b>9. Host Communications Introduction.....</b>	<b>21</b>
<b>10. Host Communications (Legacy Serial-Stream Mode).....</b>	<b>22</b>
10.1. Control Codes and Their Functions .....	22
10.2. Description of Control Functions.....	23
01 - Cursor Home (\001, 0x01, Control+A) .....	23
02 - Hide Display (\002, 0x02, Control+B).....	23



03 - Show Display (\003, 0x03, Control+C).....	23
04 - Hide Cursor (\004, 0x04, Control+D) .....	23
05 - Show Underline Cursor (\005, 0x05, Control+E).....	23
06 - Show Blinking Block Cursor with Underscore (\006, 0x06, Control+F).....	23
08 - Backspace (\008, 0x08, Control+H).....	23
09 - Module Configuration (\009, 0x09, Control+I) .....	23
10 - Line Feed (\010, 0x0A, Control+J or Control+Enter) .....	24
11 - Delete In Place (\011, 0x0B, Control+K) .....	24
12 - Form Feed (\012, 0x0C, Control+L).....	25
13 - Carriage Return (\013, 0x0D, Control+M) .....	25
14 - Backlight Control (\014, 0x0E, Control+N) .....	25
15 - Contrast Control (\015, 0x0F, Control+O).....	25
17 - Set Cursor Position (\017, 0x11, Control+Q) .....	25
18 - Horizontal Bar Graph (\018, 0x18, Control+R) .....	25
19 - Scroll ON (\019, 0x13, Control+S) .....	26
20 - Scroll OFF (\020, Control+T) .....	26
21 - Marquee (\021, Control+U).....	26
22 - Marquee Enable (\022, 0x17, Control+V) .....	27
23 - Wrap ON (\023, 0x17, Control+W).....	27
24 - Wrap OFF (\024, 0x18, Control+X).....	28
25 - Set Custom Character Bitmap (\025, 0x19, Control+Y) .....	28
26 - Reboot (\026, 0x1A, Control+Z).....	29
27 - Escape Sequence Prefix (\027, 0x1B, Escape) .....	29
28 - Large Block Number (\028, Control Backslash) .....	30
30 - Send Data Directly to the LCD Controller (\030, 0x1E, Control+Equal).....	30
31 - Show Information Screen (\031, 0x1F, Control+Minus) .....	30
<b>11. Host Communications (Packet Based Mode).....</b>	<b>31</b>
11.1. Packet Structure .....	31
11.2. About Handshaking .....	32
11.3. Report Codes .....	32
11.4. Command Codes.....	34
0 (0x00): Ping Command.....	34
1 (0x01): Get Hardware & Firmware Version .....	35
2 (0x02): Write User Flash Area .....	35
3 (0x03): Read User Flash Area .....	35
4 (0x04): Store Current State as Boot State .....	35
5 (0x05): Reboot Module .....	36
6 (0x06): Clear LCD Screen .....	37
9 (0x09): Set LCD Special Character Data .....	37
10 (0x0A): Read 8 Bytes of LCD Memory.....	37
11 (0x0B): Set LCD Cursor Position .....	38
12 (0x0C): Set LCD Cursor Style.....	38
13 (0x0D): Set LCD Contrast.....	38
14 (0x0E): Display Backlight Brightness .....	39
16 (0x10): Set Up Fan Reporting (FBSCAB Required) .....	39
17 (0x11): Set Fan Power (FBSCAB Required) .....	39
18 (0x12): Read DOW Device Information (FBSCAB Required).....	40
19 (0x13): Set Up Temperature Reporting (FBSCAB Required) .....	40



20 (0x14): Arbitrary DOW Transaction (FBSCAB Required) .....	41
25 (0x19): Set Fan Power Fail-Safe (FBSCAB Required) .....	42
26 (0x1A): Set Fan Tachometer Glitch Filter (FBSCAB Required) .....	43
27 (0x1B): Query Fan Power & Fail-Safe Mask (FBSCAB Required) .....	43
30 (0x1E): Read Reporting & Status .....	44
31 (0x1F): Send Data to LCD .....	45
33 (0x21): Set Communications Interface Options .....	45
<b>12. Character Generator ROM (CGROM) .....</b>	<b>48</b>
<b>13. LCD Module Reliability and Longevity .....</b>	<b>49</b>
13.1. Module Longevity (EOL / Replacement Policy) .....	49
<b>14. Care and Handling Precautions .....</b>	<b>50</b>
14.1. ESD (Electrostatic Discharge) .....	50
14.2. Design and Mounting .....	50
14.3. Avoid Shock, Impact, Torque, or Tension .....	50
14.4. If LCD Panel Breaks .....	50
14.5. Cleaning .....	50
14.6. Operation .....	50
14.7. Storage and Recycling .....	51
<b>15. Mechanical Drawing .....</b>	<b>52</b>
<b>16. Appendix A: Demonstration Software and Sample Code .....</b>	<b>53</b>
16.1. Demonstration Software .....	53
16.1.1. cfTest .....	53
16.1.2. CrystalControl2 (CC2) .....	53
16.1.3. Linux Examples .....	53



## 1. General Information

### Datasheet Revision History

Datasheet Version: 2023-11-14  
Hardware Version: v4.0  
PCB10061 Version: 1v0  
Firmware Version: v4.0

This datasheet is for the CFA634 Series of LCD display modules.

For reference, previous datasheets may be downloaded by clicking the “Show Previous Versions of Datasheets” in the “Datasheets” section on the product’s web page.

### Product Change Notifications

You can check for or subscribe to [Part Change Notices](#) for this display module on our website.

To see the most recent PCN for the CFA634 family at the time of this datasheet release, see [PCN #11226](#).

### Variations

Slight variations between lots are normal (e.g., contrast, color, or intensity).

### Volatility

This display module has volatile and non-volatile memory.

### Disclaimer

Certain applications using CrystalFontz America, Inc. products may involve potential risks of death, personal injury, or severe property or environmental damage (“Critical Applications”). CRYSTALFONTZ AMERICA, INC. PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. Inclusion of CrystalFontz America, Inc. products in such applications is understood to be fully at the risk of the customer. In order to minimize risks associated with customer applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazard. Please contact us if you have any questions concerning potential risk applications.

CrystalFontz America, Inc. assumes no liability for applications assistance, customer product design, software performance, or infringements of patents or services described herein. Nor does CrystalFontz America, Inc. warrant or represent that any license, either express or implied, is granted under any patent right, copyright, or other intellectual property right of CrystalFontz America, Inc. covering or relating to any combination, machine, or process in which our products or services might be or are used.

All specifications in datasheets on our website are, to the best of our knowledge, accurate but not guaranteed. Corrections to specifications are made as any inaccuracies are discovered.

Company and product names mentioned in this publication are trademarks or registered trademarks of their respective owners.

Copyright ©2023 by CrystalFontz America, Inc., 12412 East Saltese Avenue, Spokane Valley, WA 99216 U.S.A.



## 2. Introduction

### 2.1. Main Features

- Large, easy-to-read, 20-character x 4-lines LCD in an overall compact size
- Attractive stainless-steel bezel
- Option to install in a standard half-height 5-1/4" drive bay mounting bracket
- Four color variants
- Six interface options with multiple simultaneously available, see [Interface Connection Information](#)
- Wide viewing angle with a 6 o'clock viewing direction
- Temperature range for operation is -20°C to + 70°C
- Adjustable contrast. The default contrast value is acceptable for most applications
- RockWorks RW1067 or compatible LCD controller
- Choose between legacy stream communication or robust packet-based protocol with 16-bit CRC that ensures error-free communications
- Extended power supply voltage range (2.5v to 5.5v for logic and backlight supplies)
- Non-volatile memory capability to customize the power-on display settings

### 2.2. Module Classification Information

CFA634 modules can be ordered with any of the options below. The 24-part numbers represent the different combinations of interfaces and colors.

CFA   634   -   \*   \*   \*   -   K   \*   \*  
 ①   ②   ③   ④   ⑤   ⑥   ⑦   ⑧





①	Brand	CrystalFontz America, Inc.
②	Model Identifier	634
③	Backlight Type & Color	T – LED, white Y – LED, yellow-green
④	Fluid Type, Image (positive or negative), and LCD Glass Color	D – FFSTN, negative, black F – FSTN, positive, neutral M – STN, negative, blue
⑤	Polarizer Film Type, Temperature Range, and Viewing Angle (o'clock)	I – Transmissive, Wide Temperature Range, 6:00 H – Transflective, Wide Temperature Range, 6:00
⑥	Special Code	K – Manufacturer's code
⑦	Interface	C – I <sup>2</sup> C L – LL (Logic Level Serial) N – LLi (Logic Level Serial, Inverted) P – SPI S – "Full Swing" RS232 Serial U – USB
⑧	Customize (optional)	1 or more characters

If a drive bay bracket is selected to be added, the prefix "DBBK" is added to the part number.



## 2.3. Color Variants

- CFA634-TFH-Kx: White edge-lit LED backlight with positive STN transfective mode. Displays dark (near-black) characters on light (near-white) background. The display can be read in normal office lighting, in dark areas, and in bright sunlight.
- CFA634-TMI-Kx: White edge-lit LED backlight with negative STN transmissive mode LCD. Displays light (near-white) characters on a blue background. The display can be read in normal office lighting and in dark areas. May be difficult to read in direct sunlight.
- CFA634-YDI-Kx: Yellow-green array LED backlight with negative FFSTN transmissive mode LCD. Displays yellow-green characters on a dark (near-black) background. The display can be read in normal office lighting and in dark areas. May be difficult to read in direct sunlight.
- CFA634-YFH-Kx: Yellow-green array LED backlight with positive FSTN transfective mode LCD. Displays dark (near-black) characters on yellow-green background. The display can be read in normal office lighting, in dark areas, and in bright sunlight.

Part Number	Fluid	LCD Glass Color	Image	Polarizer Film	LED Backlight
 CFA634-TFH-Kx	STN	neutral	positive	transfective	white
 CFA634-TMI-Kx	STN	blue	negative	transmissive	white
 CFA634-YDI-Kx	FFSTN	black	negative	transmissive	yellow-green
 CFA634-YFH-Kx	FSTN	neutral	positive	transfective	yellow-green



### 3. Mechanical Characteristics

Item	Specification (mm)	Specification (inch, reference)
Overall Width and Height	130.0 (W) x 63.0 (H)	5.118 (W) x 2.480 (H)
Module Depth (including DB9 connector)	19.90 (D)	0.783 (D)
Active Area	93.54 (W) x 34.0 (H)	3.683 (W) x 1.339 (H)
Viewing Area	102.0 (W) x 41.5 (H)	4.016 (W) x 1.634 (H)
Character Pitch	4.68 (W) x 8.90 (H)	0.184 (W) x 0.350 (H)
Character Size	3.84 (W) x 7.30 (H)	0.151 (W) x 0.287 (H)
Pixel Size	0.72 (W) x 0.86 (H)	0.028 (W) x 0.034 (H)
Pixel Pitch	0.78 (W) x 0.92 (H)	0.031 (W) x 0.036 (H)
Weight (typical)	110 grams	3.9 ounces



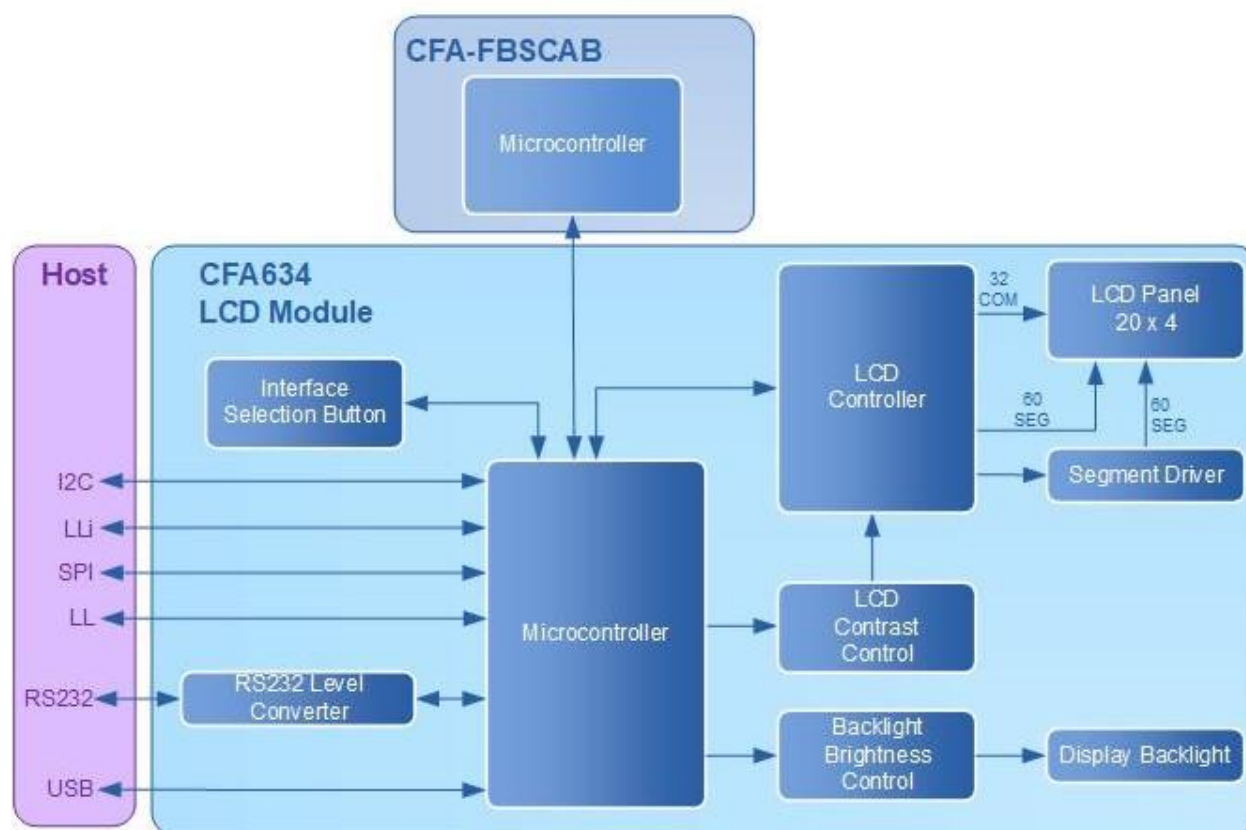


## 4. Electrical Specifications

### 4.1. System Block Diagram

The CFA634 offers six interface choices. More than one interface can be used simultaneously.

Use any of the six interfaces by changing the interface selection in a command or by using the push button, and using the appropriate jumper configurations. See [Interface Connection Information](#) and [Module Configuration \(\009, 0x09, Control+I\)](#).





#### 4.2. Absolute Maximum Ratings

Absolute Maximum Ratings	Symbol	Minimum	Maximum
Operating Temperature	$T_{OP}$	-20°C	+70°C
Storage Temperature	$T_{ST}$	-30°C	+80°C
Humidity Range (Non-condensing)	RH	10%	90%
Supply Voltage for Logic	$V_{DD}$	0v	+5.6v
Supply Voltage for Backlight	$V_{LED}$	0v	+16.0v
Input and Output Pins for RS232 Serial (CFA634)			
RS232 Input Pin	$V_{RX}$	-25v	+25v
RS232 Output Pin	$V_{TX}$	-13v	+13v
These are stress ratings only. Extended exposure to the absolute maximum ratings listed above may affect device reliability or cause permanent damage. Functional operation of the module at conditions beyond those listed under DC Characteristics is not implied. Changes in temperature can result in changes in contrast.  Powering from pins 4 and 7 on DB9, maximum voltage rating is +15v. ( <a href="#">see pin descriptions</a> )			

#### 4.3. LCD Duty and Bias

Driving Method	Specification
Duty	1/32
Bias	1/6.7

#### 4.4. DC Characteristics

DC Characteristics	Symbol	Minimum	Maximum
Supply Voltage for Logic	$V_{DD}$	+2.5v	+5.5v
Supply Voltage for Backlight	$V_{LED}$	+2.5v	+5.5v
Logic Input High Voltage	$V_{IH}$	+1.85v	-
Logic Input Low Voltage	$V_{IL}$	-	+1.36v
Logic Output High Voltage	$V_{OH}$	+2.4v	+3.3v
Logic Output Low Voltage	$V_{OL}$	+0.4v	+1.3v
RS232 Serial (supplied through DB9 connector using on-board RS232 level shifter)		±9v	±15v



#### 4.5. Typical Current Consumption

Current consumption is the same for all interfaces; although it does differ by color variant.

##### CFA634-TFH-Kx

Items Enabled		Typical Current Consumption	
Logic	LCD Backlight	V <sub>DD</sub> = +4.5v	V <sub>DD</sub> = +5.0v
X	0%	16 mA	15 mA
X	100%	130 mA	118 mA

##### CFA634-TMI-Kx

Items Enabled		Typical Current Consumption	
Logic	LCD Backlight	V <sub>DD</sub> = +4.5v	V <sub>DD</sub> = +5.0v
X	0%	15 mA	14 mA
X	100%	125 mA	112 mA

##### CFA634-YDI-Kx

Items Enabled		Typical Current Consumption	
Logic	LCD Backlight	V <sub>DD</sub> = +4.5v	V <sub>DD</sub> = +5.0v
X	0%	15 mA	14 mA
X	100%	132 mA	119 mA

##### CFA634-YFH-Kx

Items Enabled		Typical Current Consumption	
Logic	LCD Backlight	V <sub>DD</sub> = +4.5v	V <sub>DD</sub> = +5.0v
X	0%	26 mA	24 mA
X	100%	145 mA	132 mA



## 5. Optical Characteristics

CFA634-TFH-Kx, CFA634-YDI-Kx, and CFA634-YFH-Kx					
Item		Symbol	Condition	Typical	Unit
Viewing Angle	Horizontal	$\theta_{x+}$	CR $\geq$ 2 Viewing Angle = 25°	60	Degree
		$\theta_{x-}$		30	
	Vertical	$\theta_{y+}$		40	
		$\theta_{y-}$		40	
Contrast Ratio		CR	—	4	-
Response Time		T rise	Ta=25°C	150	ms
		T fall		120	ms
Viewing Direction: 6 o'clock					

CFA634-TMI-Kx					
Item		Symbol	Condition	Typical	Unit
Viewing Angle	Horizontal	$\theta_{x+}$	CR $\geq$ 2 Viewing Angle = 25°	40	Degree
		$\theta_{x-}$		35	
	Vertical	$\theta_{y+}$		30	
		$\theta_{y-}$		30	
Contrast Ratio		CR	—	4	-
Response Time		T rise	Ta=25°C	150	ms
		T fall		120	ms
Viewing Direction: 6 o'clock					

### 5.1. Optical Characteristics Test Conditions and Definitions

Note that slight color variations from module-to-module and batch-to-batch are normal.

- Viewing Angle
  - Vertical (V) $\theta$ : 0°
  - Horizontal (H) $\phi$ : 0°
- Frame Frequency: 78 Hz
- Driving Waveform: 1/16 Duty, 1/13 Bias
- Ambient Temperature (Ta): 25°C

### 5.2. LED Backlight Characteristics

Test Condition: Supply Voltage =  $V_{DD}$ . Negative modules are measured with all pixels on. Positive modules are measured with all pixels off.

Item	Symbol	Typical
CFA634-TFH-Kx	$L_v$	200 cd/m <sup>2</sup>
CFA634-TMI-Kx		400 cd/m <sup>2</sup>
CFA634-YDI-Kx		70 cd/m <sup>2</sup>
CFA634-YFH-Kx		65 cd/m <sup>2</sup>
Backlight PWM Frequency	-	1 MHz (nominal)



## 6. Interface Connection Information

### 6.1. Interface Introduction

The CFA634 can be ordered with any of the six interfaces enabled. The six interface options available on the CFA634 module are:

- USB v2.0 Full Speed (KU)
- Full-swing RS232 (KS)
- Logic-level Serial (KL)
- Logic Level Serial, Inverted (KN)
- I2C Peripheral (KC)
- SPI Peripheral (KP)

Each interface can be enabled, disabled, and configured individually via [the interface-button on the back of the CFA634 module](#), or by using commands sent to the CFA634 from the host. Interfaces may be used simultaneously if required.

One USB connector, one DB9 connector, and two pin-headers are provided on the CFA634 for host communications and power supply connection.

Connector Name	Interfaces Available	Supply Power Available
J_USB	USB v2.0 full-speed	USB 5.0v standard
J_DB9	Full-swing RS232 Logic-level Serial	Yes (pins 4 and 7)
J_I2C	I2C Peripheral	Yes (pin 4)
J2	SPI Peripheral Full-swing RS232 Logic-level Serial	Yes (pins 2 and 3)

#### 6.1.1. Module Interface Default Settings

The CrystalFontz CFA634 default interface settings are as follows:

Interface Name	Interface Setting	Interface Mode
USB	<b>Enabled</b>	Legacy Serial-Stream
Logic-Level Serial	Disabled	N/A
SPI	Disabled	N/A
I2C	Disabled	N/A
RS232 Serial	<b>Enabled</b>	Legacy Serial-Stream

The default settings are the same for all CFA634 modules. Issuing a reset command will set a module to these interface settings no matter which module was ordered.

Interface settings and mode can be adjusted using the button on the back of the module (Interface Configuration Menu) or by using command [Interface Selection](#) or command [Set Communications Interface Options](#). The default settings can be reinstated by using command [Reset to Defaults](#) or command [Reboot and Reset](#).



## 6.2. Communications Interface Descriptions

### 6.2.1. USB Communications

The USB 2.0 interface requires only one connection to the host for both data communications and power supply. The USB interface may be used in legacy-stream mode for simplicity and backwards compatibility, or packet-interface mode for higher reliability and enhanced features. The module has a low profile 2mm latching polarized connector for USB connection. Example cables from CrystalFontz include the [WR-USB-Y03](#) and the [WR-USB-Y61](#)

The module enumerates as a virtual COM port to the host system. It is supported natively by Microsoft Windows 10 and 11, most Linux distributions, and OS-X. A [CrystalFontz USB driver pack](#) is available for Windows XP through to Windows 8.1, and Windows Server.

### 6.2.2. RS-232 Full-Swing Serial Communications

Industry standard RS-232 full-swing serial is available on the DB9 connector and J2 header. Serial baud rates available are 115200, 19200 (default), 9600, 4800, 2400, using 8-N-1 format (8 data bits, no parity bit, 1 stop bit). The RS-232 interface may be used in legacy-stream mode for simplicity and backwards compatibility, or packet-interface mode for higher reliability and enhanced features.

See the [DB9 Connector and Power Details](#) and [J2 Connector and Power Details](#) sections for more connection information.

### 6.2.3. Logic-Level Serial Communications

Logic-level serial and inverted logic-level serial are available on the DB9 and J2 headers. Serial baud rates available are 115200, 19200 (default), 9600, 4800, 2400, using 8-N-1 format (8 data bits, no parity bit, 1 stop bit).

The logic-level serial interface may be used in legacy-stream mode for simplicity and backwards compatibility, or packet-interface mode for higher reliability and enhanced features.

See the [DB9 Connector and Power Details](#) and [J2 Connector and Power Details](#) sections for more connection information.

### 6.2.4. SPI Interface

The SPI interface is available only on the J2 header. The SPI interface may be used up to 8Mhz, and the following options are available:

- Phase: first or second edge (CPHA)
- Polarity: high or low (CPOL)
- Bit order: MSB or LSB first

The SPI interface may be used in legacy-stream mode for simplicity and backwards compatibility, or packet-interface mode for higher reliability and enhanced features.

SPI is useful with embedded microprocessors that lack a hardware UART. On the host microprocessor, any three general-purpose output (GPIO) ports and a "send byte" routine can be used to control the display module. If the host microprocessor has a hardware SPI port, that can be used, provided that the port's speed can satisfy the timing constraints. A software state-machine and a timer interrupt could also be used to reduce the microprocessor's load while satisfying the timing requirements.

Further, using SPI, additional displays can be controlled with a single added output port (SPI SS) per display. The other lines (MOSI, MISO, and SCK) are common.



### 6.2.5. I<sup>2</sup>C Interface

The I<sup>2</sup>C interface is available only on J\_I2C, the I<sup>2</sup>C header/connector.

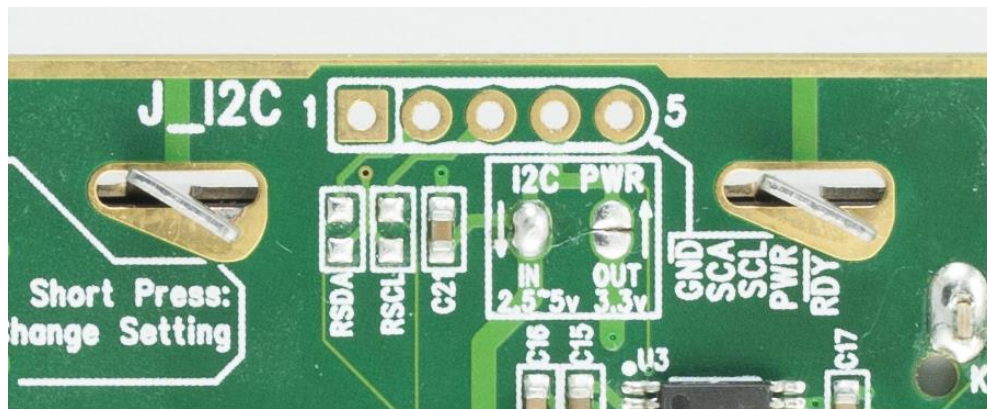
The I<sup>2</sup>C interface option allows the module to operate in peripheral mode supporting the 100-kbps original and 400-kbps fast mode communication speeds.

The I<sup>2</sup>C interface may be used in legacy-stream mode for simplicity and backwards compatibility, or packet-interface mode for higher reliability and enhanced features.

Benefits to using I<sup>2</sup>C are: only two data/bus lines are required, no strict clocking to confine the environment, and a simple controller/peripheral relationship with multi-device support utilizing a software configurable address.

The CFA634 also provides a I<sup>2</sup>C Data Ready flag/interrupt output pin (I<sup>2</sup>C header, pin 5), that goes logic-low (active low) when data is available to be read from the module.

The CFA634 has internal SCL/SDA pull-ups of 40k $\Omega$  to 80k $\Omega$  that may not be sufficient in all circumstances. Land patterns for 0603 pull ups are available for adding external pull-ups as appropriate. They are labeled RSDA and RSCL.



The display module ships with a default I<sup>2</sup>C address of 42<sub>10</sub>. The address is configurable using the Interface Configuration Menu, or [packet command 33 \(Set Communications Interface Options\)](#).

The I<sup>2</sup>C protocol specifies each module with two addresses. The first 7 bits of both addresses are the same, in this case 00101010<sub>2</sub>. The address is then left bit-shifted with a read bit (1) or write bit (0). If a controller device wishes to write to the module, it would transmit 01010100<sub>2</sub> which is 42<sub>10</sub> left bit-shifted with a 0 (84<sub>10</sub>). If instead a controller device wishes to read from the display module, it would transmit 01010101<sub>2</sub> which is 42 left bit-shifted with a 1 (85<sub>10</sub>).





### 6.3. USB Connector Details

The USB interface requires only one connection to the host for both data communications and 5V power supply. The module has a low profile 2mm latching polarized connector for USB connection.

We offer four USB cables to connect between the display module and the host:

1. [WR-USB-Y03](#) (~1.94 meters) connects between the 2mm USB connector on the display module and a USB-A socket on the host.
2. [WR-USB-Y11](#) (~0.77 meters) connects between the 2mm USB connector on the display module and single pin connectors typically found on motherboards.
3. [WR-USB-Y33](#) (~0.69 meters) connects between the 2mm USB connector on the display module and 0.1" USB pins on the host's motherboard.
4. [WR-USB-Y61](#) (~1.8 meters) connects between the 2mm USB connector on the display module and a USB-C socket on the host.

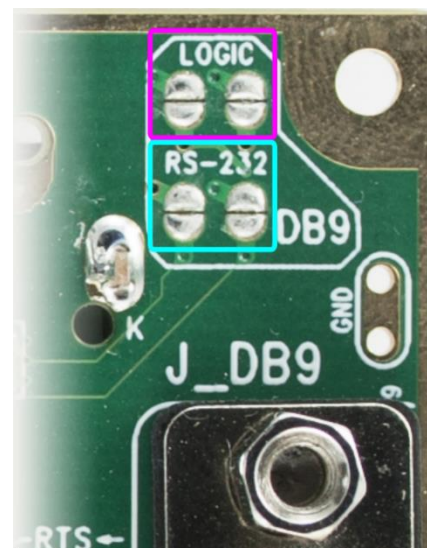
### 6.4. DB9 Connector and Power Details

RS-232 and Logic Level serial interfaces are available on the DB9 connector. Only one of these interfaces is available at a time on the connector.

Select which interface is available on the DB9 connector by solder-bridging two of the four DB9 jumper pads on the rear of the CFA634 module.

When the top two jumpers outlined in pink are closed and the bottom two jumpers outlined in blue are open, Logic Level Serial is selected.

When the top two jumpers (pink) are open and the bottom two jumpers (blue) are closed, RS-232 is selected.



Below is a view looking into the male DB9 connector (labeled J\_DB9):



Pin	CFA634 Pin Description	PC Pin Name
1	Not Connected	DCD (Data Carrier Detect)
2	Serial Data Out	R <sub>x</sub> (Receive Data)
3	Serial Data In	T <sub>x</sub> (Transmit Data)
4	Power A (2.5-volts to 5.5-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 (2.5-volts to 5.5-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 can power the display module, but will not be able to power the backlight through the DTR and RTS lines. Use software to drive these lines high (most software already will). To power the backlight in RS-232 mode, refer to [section 8. Alternate Method to Power the Backlight for non-USB Interfaces](#), below.

For non-backlight operation when the module is connected to a PC's 9-pin serial port, only a "straight through" RS-232 9-pin female-to-female DB9 cable is required. Connect one of the cable's connectors to the DB9 male connector on the display module. Connect the cable's other connector to the DB9 male connector on the host. DB9 cables are available on our website: [WR-232-Y04](#) ~3m, [WR-232-Y10](#) ~0.3m.

To connect the display module with RS-232 interface to a PC expansion slot, use a [WR-232-Y09](#) ribbon cable (~0.965 meters).

To connect a display module with RS-232 to a motherboard with a 10-pin male connector with normal or alternate pinout, use a [WR-232-Y23](#) (~0.655 meters) cable.

The [WR-232-Y18](#) DB9 RS-232 plus DC power connector piggybacks a power supply so that the cable outputs 5v to the host.

## 6.5. J2 Connector and Power Details

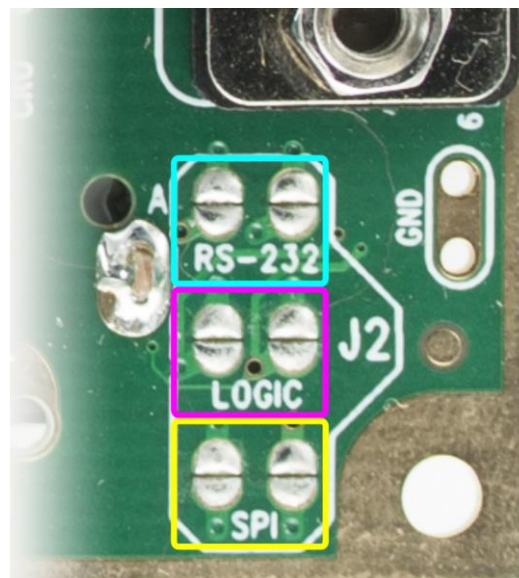
RS-232, Logic Level Serial, and SPI interfaces are available on the J2 header/connector. Only one of these interfaces is available at a time on the connector.

Select an interface by solder-bridging the two corresponding J2 jumper pads on the rear of the CFA634 module.

Close the top two (outlined in blue) for RS-232.

Close the center two (outlined in pink) for LL Serial.

Close the bottom two (outlined in yellow) for SPI.



J2 Pin	Pin Label	Use in RS-232/LL Serial Mode	Use in SPI Mode
1	GROUND	Power Ground	Power Ground
2	LCD(2.5-5.5V)	Display/Logic Power Supply	Display/Logic Power Supply
3	LED(2.5V-5.5V)	Backlight Power Supply	Backlight Power Supply
4	MOSI/RX	RS232/Serial Data In	SPI Controller-In/Peripheral-Out Data
5	SS	N/A	SPI Chip-Select Input
6	SCK	N/A	SPI Clock Input
7	MISO/TX/BUSY	RS232/Serial Data Out	SPI Controller-Out/Peripheral-In Data



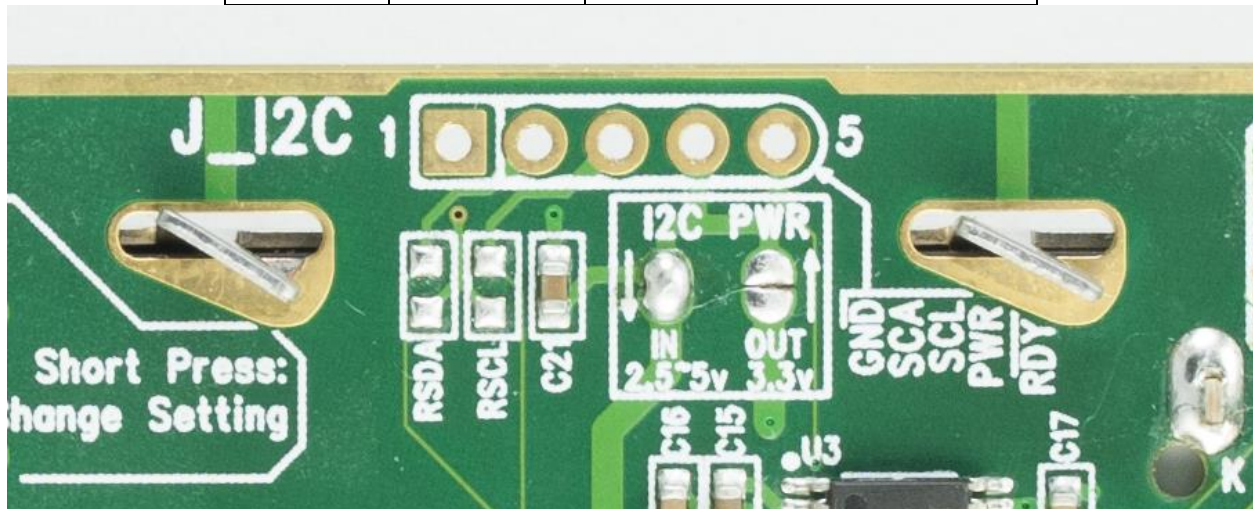
## 6.6. I2C Connector and Power Details

The I<sup>2</sup>C interface is available on the I2C header/connector.

The CFA634 has internal SCL/SDA pull-ups of 40k $\Omega$  to 80k $\Omega$  that may not be sufficient in all circumstances. Land patterns for 0603 pull ups are available for adding external pull-ups as appropriate. They are labeled RSDA and RSCL.

The I2C PWR IN and OUT jumpers either connects pin 4 to the board's 3.3v (IN closed, OUT open), or allows the I2C power to be separately supplied (IN open, OUT closed). By default, modules are shipped with IN closed and OUT open. Only one of these jumpers should be closed at a time.

I2C Pin	Pin Label	Pin Description
1	GROUND	Power Ground
2	SDA	I2C Data
3	SCL	I2C Clock
4	POWER	Supply Power (2.5v to 5.5v) – Provides or receives power depending on I2C PWR jumper
5	RDY	I2C Data Ready





## 7. Interface Configuration Menu

The various communication interfaces present on the CFA634 may be configured by using the small tactile button labeled CONFIGURE on the rear of the module.

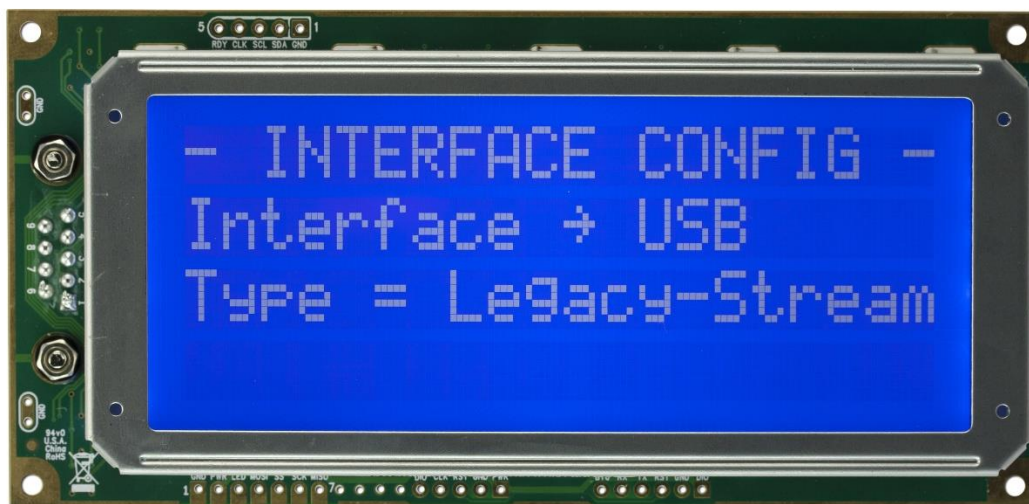
On entering the Interface Configuration Menu the module's saved settings are loaded. Once configuration has been completed by the user, the module will save the new settings, and reboot itself.



**NOTE:** While the configuration menu is shown, the CFA634 will not accept, reply, or send any commands/data on any of the interfaces. Normal CFA634 operation is halted.

### To display the menu:

Press the CONFIGURE button on the rear of the CFA634 module.



A right-pointing arrow indicates which item is currently selected.

### To change the selected value:

Short-press the CONFIGURE button.

### To select a different item:

Long-press the CONFIGURE button.

### To exit the menu:

Wait 5 seconds. After 5 seconds without pressing the CONFIGURE button, a save settings countdown will be displayed.

To save settings, wait for the 5 second count-down to complete.

To cancel saving settings, press the CONFIGURE button.



## 8. Alternate Method to Power the Backlight for Non-USB Interfaces

The module's backlight brightness is controlled by Pulse Width Modulation (PWM). The PWM may cause noise on supplies that have high output impedance. A filter capacitor from +5V(LCD) to GROUND close to the module can mitigate the noise. Alternately, operate the backlight at full brightness, which disables the PWM.


To use the backlight in any of the following configurations requires a regulated +5V supply capable of sourcing 400mA, hereby referred to as the supply. The supply can connect to any +5V(LED) terminal. This may be the same supply that is used for +5V(LCD).

**IMPORTANT:** Connecting a supply greater than 5.25V to VDD or 5V(LED) will damage the display module.

### 8.1. To Power the Backlight Using J2 Expansion Port Pins

To power the backlight, connect the supply to pin 3 labeled LED on the J2 connector. The supply's ground should be connected to pin 1 of J2 labeled GROUND, shown below.

By default, JPBL is closed, tying the backlight power and controller power together. To provide separate power, open JPBL.



Pin	Pin Name	Function
7	MISO	Controller In, Peripheral Out SPI Busy (for status output)
6	SCK	SPI Clock (Input)
5	SS	SPI Chip Select (Active Low Input)
4	MOSI	Controller Out, Peripheral In Serial Data In (input)
3	LED	LED Backlight Power (2.5-5.5v)
2	LCD	Controller and LCD Power (2.5-5.5v)
1	GROUND	Ground (Backlight and Controller)



## 9. Host Communications Introduction

The CFA634 firmware v4.0 and onwards supports two different host communication systems.

All communications interfaces may be configured to operate in either mode at any time independently via the on-display interface configuration menu accessed by pressing the button on the rear of the module, or by using Packet-mode Command 33 (Set Communications Interface Options).

For example, the USB interface could be set to use Packet-based system, while the I2C interface could be set to use the Legacy serial-stream system. Both interfaces can be used simultaneously if desired.

### Legacy serial stream system:

Advantages:

- very simple, fast
- very low host system requirements

Disadvantages:

- no error checking or handshaking
- limited command set
- one-way communication only

### Crystalfontz packet-based system:

Advantages:

- error-checked robust communication
- expanded command set
- two-way communications
- FBSCAB module use

Disadvantages:

- requires simple packet processing and CRC calculation/checking by host system

See the [Interface Default Settings](#) section for default interface host communication mode settings.





## 10. Host Communications (Legacy Serial-Stream Mode)

### 10.1. Control Codes and Their Functions

The module accepts plain ASCII characters and displays them on the screen at the current cursor position. For example, if “Hello World” is sent, the display shows “Hello World”. The module also supports a set of control codes 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 the decimal representation (base 10) of a number. Use the codes \000 to \255 to include all possible values for a character. This notation is supported by our [cfTest](#) software. We recommend trying this utility by using the examples listed under [Description of Control Functions](#).

ASCII	HEX	Keystrokes	Control Code Functions
\000	0x00	-	Not Accessible
<a href="#">\001</a>	0x01	Control+A	Cursor Home
<a href="#">\002</a>	0x02	Control+B	Hide Display
<a href="#">\003</a>	0x03	Control+C	Show Display
<a href="#">\004</a>	0x04	Control+D	Hide Cursor
<a href="#">\005</a>	0x05	Control+E	Show Underline Cursor
<a href="#">\006</a>	0x06	Control+F	Show Blinking Block Cursor with Underscore
\007	0x07	-	Not Accessible
<a href="#">\008</a>	0x08	Control+H	Backspace (Destructive)
<a href="#">\009</a>	0x09	Control+I	Module Configuration
<a href="#">\010</a>	0x0A	Control+J or Control +Enter	Line Feed
<a href="#">\011</a>	0x0B	Control+K	Delete In Place
<a href="#">\012</a>	0x0C	Control+L	Form Feed (Clear Display)
<a href="#">\013</a>	0x0D	Control+M	Carriage Return
<a href="#">\014</a>	0x0E	Control+N	Backlight Control
<a href="#">\015</a>	0x0F	Control+O	Contrast Control
\016	0x10	-	Not Accessible
<a href="#">\017</a>	0x11	Control+Q	Set Cursor Position (Column and Row)
<a href="#">\018</a>	0x12	Control+R	Horizontal Bar Graph
<a href="#">\019</a>	0x13	Control+S	Scroll ON
<a href="#">\020</a>	0x14	Control+T	Scroll OFF
<a href="#">\021</a>	0x15	Control+U	Marquee Settings
<a href="#">\022</a>	0x16	Control+V	Marquee Enable
<a href="#">\023</a>	0x17	Control+W	Wrap ON
<a href="#">\024</a>	0x18	Control+X	Wrap OFF
<a href="#">\025</a>	0x19	Control+Y	Set Custom Character Bitmap
<a href="#">\026</a>	0x1A	Control+Z	Reboot
<a href="#">\027</a>	0x1B	Escape	Escape Sequence Prefix



ASCII	HEX	Keystrokes	Control Code Functions
<a href="#">\028</a>	0x1C	Control+\	Large Block Number
<a href="#">\029</a>	0x1D	-	Not Accessible
<a href="#">\030</a>	0x1E	Control+Equal	Send Data Directly to LCD Controller
<a href="#">\031</a>	0x1F	Control+Minus	Show Information Screen
<a href="#">\128-\135</a>	-	-	Custom Characters 0 through 7

## 10.2. Description of Control Functions

### 01 - Cursor Home (\001, 0x01, Control+A)

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

### 02 - Hide Display (\002, 0x02, Control+B)

Display is blanked; no data is changed.

### 03 - Show Display (\003, 0x03, Control+C)

Restores blanked display; nothing else is changed.

### 04 - Hide Cursor (\004, 0x04, Control+D)

Cursor is not shown; nothing else is changed.

### 05 - Show Underline Cursor (\005, 0x05, Control+E)

Sets cursor to nonblinking underscore.

### 06 - Show Blinking Block Cursor with Underscore (\006, 0x06, Control+F)

Sets cursor to blinking block underscore.

### 08 - Backspace (\008, 0x08, Control+H)

Moves the cursor back one space and erases the character in that space. Wraps from the left-most column to the right-most column of the line above. Wraps from the left-most column of the top row to the right-most column of the bottom row. Ignores wrap.

### 09 - Module Configuration (\009, 0x09, Control+I)

Stores the current state of the module in non-volatile flash memory, recalling the flash memory contents to the display, and controlling the boot behavior. This command remains in place for backwards compatibility with previous version CFA634 modules. Many of the features of the current CFA634 cannot be controlled using this command. See the packet-based commands for control of the full feature set.

#### Power-On/Reboot Display Duration (\009\000)

\009\000\000 - Displays permanently

\009\000\001 - Displays until input

\009\000\002 - Displays for ~5 seconds then goes blank

#### Power-On/Reboot Screen Selection (\009\001)

\009\001\000 – CrystalFontz default display

\009\001\001 – User saved display



### **Enable Interface (\009\002)**

Enables the specified interface in legacy serial-stream mode. Multiple interfaces may be enabled at a time.

\009\002\000 – USB  
\009\002\001 – RS232  
\009\002\002 – SPI  
\009\002\003 – I2C  
\009\002\004 – Logic Level Serial (non-inverted)  
\009\002\005 – Logic Level Serial (inverted)

### **Set RS-232 Interface Baud Rate (\009\003)**

\009\003\000 – 2400 bps  
\009\003\001 – 4800 bps  
\009\003\002 – 9600 bps  
\009\003\003 – 19200 bps (default)  
\009\003\004 – 115200 bps

### **Set I2C Address (\009\004)**

\009\004\000 to \0127 – Set I<sup>2</sup>C interface address (0 to 127)

### **Save Settings (\009\005)**

\009\005 – Save user settings to non-volatile flash memory

### **Recall Settings (\009\006)**

\009\006 – Recall user settings (as saved by \009\005)

### **Reset to Defaults (\009\007)**

\009\007 – Reset the module to [module interface default settings as described in section 6](#), including all interface settings.

### **Interface Selection (\009\008)**

\009\008\000 – Set all interfaces to legacy-stream mode  
\009\008\001 – Set all interfaces to packet mode

On receiving this command, the module loads its saved settings, changes all interfaces to enabled and in legacy-stream or packet mode (per 3<sup>rd</sup> byte sent, as detailed above), saves its settings, then reboots. This interface selection command is intended to be used to rescue the module from a state where it cannot be communicated with or re-configured using the interface configuration button on the rear of the module.

## **10 - Line Feed (\010, 0x0A, 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 scrolls up one row and the bottom row is cleared.

## **11 - Delete In Place (\011, 0x0B, Control+K)**

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





## 12 - Form Feed (\012, 0x0C, Control+L)

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

## 13 - Carriage Return (\013, 0x0D, Control+M)

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

## 14 - Backlight Control (\014, 0x0E, Control+N)

Send "Control+N", followed by a byte from 0 to 100 for the backlight brightness.

Examples:

```
\014\000 - backlight off
\014\050 - backlight 50% brightness
\014\100 - backlight full brightness
```

## 15 - Contrast Control (\015, 0x0F, Control+O)

Send "Control+O", followed by a byte from 0 to 100 for the contrast setting.

0 = very light    40 = typical    100 = very dark

Examples:

```
\015\020 - very light
\015\050 - close to ideal
\015\080 - very dark
```

## 17 - Set Cursor Position (\017, 0x11, Control+Q)

Send "Control+Q" followed by one byte for the column (0 to 19), and a second byte for the row (0 to 3). The upper-left position is 0,0. The lower-right position is 19,3.

For example, to move the cursor to column 11 of the second line:

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

## 18 - Horizontal Bar Graph (\018, 0x18, Control+R)

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

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

**graph\_index** determines the custom characters that are used. **IMPORTANT:** Graphs use some of the custom characters and may corrupt the display contents if user-defined custom characters or large numbers are shown.

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 between \000 and \255 is valid. The Most Significant Bit (MSB) is at the top of the row, the Least Significant Bit (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 \015. `start_column` must be less than or equal to `end_column`.

`length` is the length in pixels of the graph. Positive values graph from the left edge of `start_column`, negative values graph from the right edge of `end_column`. There are six pixels per character, so the maximum value of `length` is  $16 \times 6 = \backslash096$ .

`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 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  $256 - \text{value}$ . For instance, for a graph to extend 20 pixels towards the left from the right most column of the graph area, send 236 ( $256 - 20 = 236$ ). No additional graph "setup" command is needed.

### 19 - Scroll ON (\019, 0x13, Control+S)

Turns scroll feature on. A subsequent Line Feed (Control+J) command from the bottom row scrolls the display up by one row, independent of Wrap. If Wrap is also on (Control+W), a wrap occurring on the bottom row causes the display to scroll up one row. Scroll is on by default.

### 20 - Scroll OFF (\020, Control+T)

Turns scroll feature off. With Scroll off, a Line Feed (Control+J) command from the bottom row moves 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 also wraps vertically to the top row. Scroll is on by default at power-up.

### 21 - Marquee (\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.



Example:

```
\004\022\255\001\005\012\017\000\001Scrolling Marquee\032
```

```
\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\032\021\013\032\021\014\032\021\015\032\021\016\032\021\017\032\021\018\032\021\019\032
```

```
\022\001\001\016
```

\004	Hide cursor
\022\255\001\005	Disable marquee
\012	Clear display and set cursor to home
\017\000\001	Set cursor position to column 0, row 1
Scrolling Marquee\032	Send text "Scrolling Marquee" and a space
\021\000C	Set Marquee space 0 to "C"
\021\001r	Set Marquee space 1 to "r"
	.
	.
	.
\021\019\032	Set Marquee space 19 to a space
\022\001\001\016	Enable the Marquee on row 1 (second line down) shifting every 960ms

## 22 - Marquee Enable (\022, 0x17, Control+V)

Send "Control V" followed by the following bytes:

```
line
scroll_step_size
update_speed
```

`line` determines which line scrolls with the scrolling marquee (values 0 to 3), or if the scrolling marquee is disabled (value of 255).

`scroll_step_size` and `update_speed` determine the scroll speed for the marquee, according to the formula:  $\text{shift\_period\_ms} = (7 - \text{scroll\_step\_size}) * \text{update\_speed} * 10$

**Note:** The CFA634 does not support individual pixel text scrolling. Characters are shifted across the display one character at a time.

## 23 - Wrap ON (\023, 0x17, Control+W)

Turns wrap feature on. When wrap is on, a printable character received when the cursor is at the right-most column causes 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 wraps to the top row if Scroll is off, or the display scrolls up one row if Scroll is on.



## 24 - Wrap OFF (\024, 0x18, Control+X)

Turns wrap feature off. When wrap is off, a printable character received when the cursor is at the right-most column causes the cursor to disappear as it is off the right edge of the screen. Any subsequent characters are ignored until some other command moves the cursor back onto the display. This function is independent of Scroll.

## 25 - Set Custom Character Bitmap (\025, 0x19, 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 are 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. For example:

```
\b00111111 = \063 = a fully dark row (all 6 pixels in the 1 or on state)
\b00101010 = \042 = every other pixel on (first pixel on)
\b00010101 = \021 = every other pixel on (first pixel off)
\b00000000 = \000 = all pixels off
```

**IMPORTANT:** The large digits use all of the custom characters, so if the custom characters are modified when large digits are displayed, ***the display module is likely to become corrupted.*** The bar graphs also use some of the custom characters.



Example:

```
\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
\012\001\128\129\130\131\017\000\001\132\133\134\135
```

## 26 - Reboot (\026, 0x1A, Control+Z)

This command reboots the module to its default power-on or user-saved state. This can be useful if a number of commands have been sent in testing, or mistakenly which have put the module into an unknown state.

Send one “Control+Z” followed by another “Control+Z” to reboot the display module. If the display module’s state is unknown, 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 has just been received by the module when an application crashes, the module interprets the next 9 bytes as the parameters to the command, then wait for more commands. If the status of the module is unknown, send 9 blanks (\032) followed by two “Control+Z”s (\026).

If the host PC/device is connected by USB, the virtual COM port must be reconnected after sending this command.

### Reboot and reset to factory-defaults:

Send one “Control+Z” followed by a “Control+P” (\026\016) to reboot the module and reset its settings to defaults. After the reboot and reset the module will be configured as when first supplied by CrystalFontz. All user configured settings including interface settings, saved boot screen, etc. will be reset.

## 27 - Escape Sequence Prefix (\027, 0x1B, Escape)

Four escape sequences are supported. These correspond to the escape sequences that are sent for the four arrow keys in HyperTerminal with an ANSI terminal selected. These sequences only move the cursor, they 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



## 28 - Large Block Number (\028, Control Backslash)

Send “Control+Backslash” followed by the following bytes:

```
style
column
number
```

`style` determines if a large number (3x4 or 4x4) is displayed.

style	Function
\000	3x4 Large Number
\001	4x4 Large Number

`column` is the starting column of the number.

```
\000-\017 are valid for a style of \000 (3x4)
\000-\016 are valid for a style of \001 (4x4)
```

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

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

Example:

```
\004\012\028\000\0010\028\000\0051\028\000\0092\028\000\0133\028\000\0174
\004\012\028\000\0005\028\000\0046\028\000\0087\028\000\0128\028\000\0169
\004\012\028\001\0000\028\001\0051\028\001\0102\028\001\0153
\004\012\028\001\0006\028\001\0057\028\001\0108\028\001\0159
```

## 30 - Send Data Directly to the LCD Controller (\030, 0x1E, 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)

`data` is the data to write to the controller

**IMPORTANT:** This command executes a low-level write directly to the controller. “Control+Z” followed by another “Control+Z” reboots the display and recovers from most mistakes. **Use this command at your own risk – this command can disable or damage the display module.**

## 31 - Show Information Screen (\031, 0x1F, Control+Minus)

This function clears the screen and displays the default CrystalFontz CFA634 power-on / reboot screen.



## 11. Host Communications (Packet Based Mode)

### 11.1. Packet Structure

The packet format allows for very reliable communications between the CFA634 and the host without the traditional problems that occur in a stream-based serial communication such as having to send data in inefficient ASCII format, to “escape” certain “control characters”, or losing sync if a character is corrupted, missing, or inserted.

Reconciling packets is recommended over using delays when communicating with the module. To reconcile packets, ensure the acknowledgement packet is received before sending any additional packets.

All packets have the following structure:

`<type><data_length><data><CRC>`

`<type>` is one byte, and identifies the type and function of the packet:

```
TTcc cccc
|||| ||| |--Command, response, error or report code 0-63
||-----Type:
    00 = normal command from host to CFA634
    01 = normal response from CFA634 to host
    10 = normal report from CFA634 to host (not in
        direct response to a command from the host)
    11 = error response from CFA634 to host (a packet
        with valid structure but illegal content
        was received by the CFA634)
```

`<data_length>` specifies the number of bytes to follow in the data field.

The valid range of `<data_length>` is 0 to 22.

`<data>` is the payload of the packet. Each type of packet has a specified `<data_length>` and format for `<data>` as well as algorithms for decoding data, detailed below.

`<CRC>` is a standard 16-bit CRC of all the bytes in the packet except the CRC itself. The CRC is sent LSB first. At the port, the CRC immediately follows the last used element of `data[]`. [See Appendix A: Demonstration Software and Sample Code](#) for details.

The following C definition may be useful for understanding the packet structure.

```
typedef struct
{
    unsigned char command;
    unsigned char data_length;
    unsigned char data[MAX_DATA_LENGTH];
    unsigned short CRC;
} COMMAND_PACKET;
```

CrystalFontz supplies a demonstration and test program, [cfTest](#), that can be used to experiment with and test the CFA634's operation in both interface modes.





## 11.2. About Handshaking

The CFA634's packets make implementing hardware or software handshaking unnecessary.

The host should wait for a corresponding acknowledge packet from the CFA634 before sending the next command packet. The CFA634 responds to all packets within 250 mS. The host software should retry sending the packet if the CFA634 fails to respond within 250 mS, and report an error if a packet is not acknowledged after several retries. This situation indicates a hardware problem – for example, a disconnected cable.

Some operating systems add delays between the data arriving at the physical port from the CFA634 and when it is available to the user program. In this case, the host program may need to increase its timeout window to account for the additional overhead of the operating system.

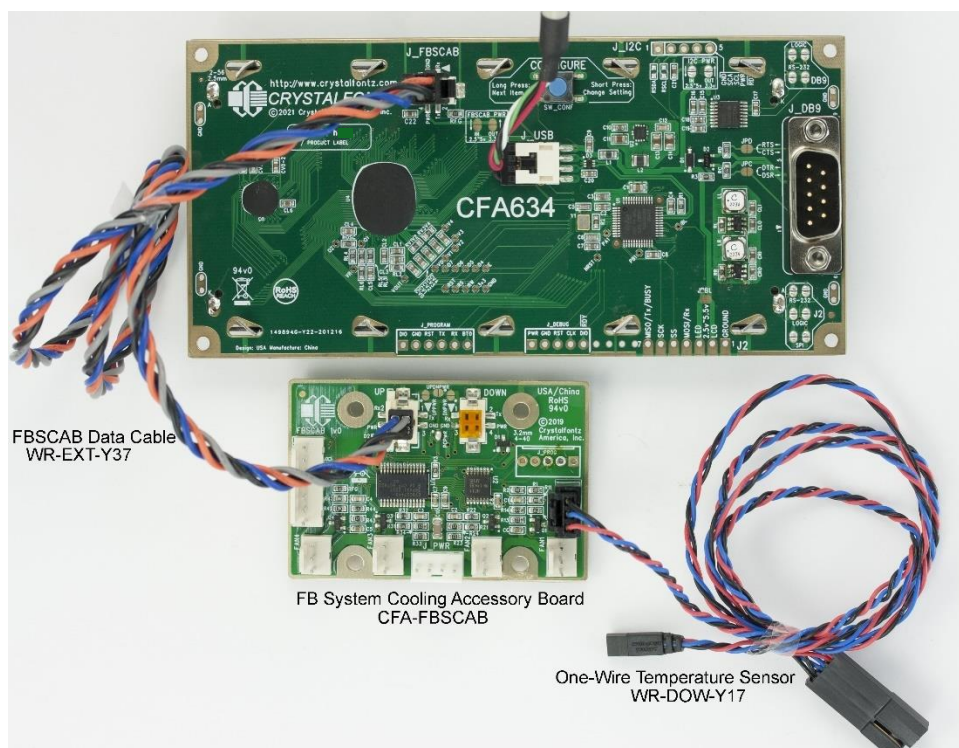
The CFA634 can be configured to send several types of report packets along with regular acknowledge packets. The host should be able to buffer several incoming packets and must process and remove packets from its input buffer faster than the packets can arrive given the baud rate and the reporting configuration of the CFA634.

The report packets are sent asynchronously with respect to the command packets received from the host. The host should not assume that the first packet received after it sends a command is the acknowledge packet for that command. The host should inspect the `type` field of incoming packets and process them accordingly.

## 11.3. Report Codes

The CFA634 can be configured to report two items: Fan Speed Report and Temperature Sensor Report. The CFA634 sends reports automatically when the data becomes available. Reports are not sent in response to a particular packet received from the host.

These reports both require the use of a [CFA-FBSCAB](#) which is an add-on module.







## 0x81: Fan Speed Report (FBSCAB Required)

If any of the fans connected to the CFA634 + [FBSCAB](#) is configured to report its speed information to the host, the CFA634 sends Fan Speed Reports for each selected fan every 1/2 second, see command [16 \(0x10\): Set Up Fan Reporting](#).

Report packet format:

```
type = 0x81
data_length = 4
data[0] = index of the fan being reported:
    0 = FAN 1
    1 = FAN 2
    2 = FAN 3
    3 = FAN 4
data[1] = number_of_fan_tach_cycles
data[2] = MSB of Fan_Timer_Ticks
data[3] = LSB of Fan_Timer_Ticks
```

If the FBSCAB module is disconnected, data[1], data[2], and data[3] are returned as 0.

The following C function decodes the fan speed from a Fan Speed Report packet into RPM:

```
int OnReceivedFanReport(COMMAND_PACKET *packet, char *output)
{
    int return_value = 0;
    int number_of_fan_tach_cycles = packet->data[1];

    if (number_of_fan_tach_cycles < 3)
        sprintf(output, " STOP");
    else if (number_of_fan_tach_cycles < 4)
        sprintf(output, " SLOW");
    else if (0xFF == number_of_fan_tach_cycles)
        sprintf(output, " ----");
    else
    {
        //Specific to each fan, most commonly 2 PPR
        int pulses_per_revolution = 2;
        int fan_timer_ticks = (*(unsigned short *)(&(packet->data[2])));

        return_value = ((27692308L/pulses_per_revolution)*
            (unsigned long)(number_of_fan_tach_cycles-3))/
            (fan_timer_ticks);
        sprintf(output, "%5d", return_value);
    }
    return(return_value);
}
```



## 0x82: Temperature Sensor Report (FBSCAB Required)

If any of the temperature sensors connected to the CFA634 + [FBSCAB](#) are configured to report to the host, the CFA634 + [FBSCAB](#) sends Temperature Sensor Reports for each selected sensor every second. See the command [19 \(0x13\): Set Up Temperature Reporting](#).

Report packet format:

```
type = 0x82
data_length = 4
data[0] = index of the temperature sensor being reported:
    0 = temperature sensor 1
    1 = temperature sensor 2
    . . .
    15 = temperature sensor 16
data[1] = MSB of Temperature_Sensor_Counts
data[2] = LSB of Temperature_Sensor_Counts
data[3] = DOW_crc_status
```

If a DOW sensor error occurs, data[3] is returned with a value of 0.

The following C function decodes the Temperature Sensor Report packet into °C and °F:

```
void OnReceivedTempReport(COMMAND_PACKET *packet, char *output)
{
    //First check the DOW CRC return code from the CFA634
    if(packet->data[3]==0)
        strcpy(output, "BAD CRC");
    else
    {
        double degc = (*(short*)&(packet->data[1]))/16.0;
        double degf = (degc*9.0)/5.0+32.0;
        sprintf(output, "%9.4f°C = %9.4f°F", degc, degf);
    }
}
```

## 11.4. Command Codes

Below is a list of valid packet commands for the CFA634. Each command packet is answered by either a response packet or an error packet. The low 6 bits of the type field of the response or error packet is the same as the low 6 bits of the `type` field of the command packet being acknowledged.

### 0 (0x00): Ping Command

The CFA634 returns the Ping Command to the host.

Request packet format:

```
type: 0x00 = 010
data_length = 0 to 16
data[] = up to 16 bytes of arbitrary data
```

Successful return packet format:

```
type: 0x40 | 0x00 = 0x40 = 6410
data_length = (identical to received packet)
data[] = (identical to received packet)
```



## 1 (0x01): Get Hardware & Firmware Version

Returns the hardware and firmware version information to the host.

Request packet format:

```
type: 0x01 = 110  
data_length = 0
```

Successful return packet format:

```
type: 0x40 | 0x01 = 0x41 = 6510  
data_length = 16  
data[] = "CFA634:h4.0,v4.0"
```

## 2 (0x02): Write User Flash Area

Writes to the 16 bytes of nonvolatile memory reserved for arbitrary use by the host. This memory can be used to store a serial number, IP address, gateway address, netmask, or any other data required. All 16 bytes must be supplied.

Request packet format:

```
type: 0x02 = 210  
data_length = 16  
data[] = 16 bytes of arbitrary user data to be stored in the CFA634's  
non-volatile memory
```

Successful return packet format:

```
type: 0x40 | 0x02 = 0x42 = 6610  
data_length = 0
```

## 3 (0x03): Read User Flash Area

Reads the User Flash Area and returns the data to the host.

Request packet format:

```
type: 0x03 = 310  
data_length = 0
```

Successful return packet format:

```
type: 0x40 | 0x03 = 0x43 = 6710  
data_length = 16  
data[] = 16 bytes of user data from the CFA634's non-volatile memory
```

## 4 (0x04): Store Current State as Boot State

Stores the current module state (interface, display configuration, FBSCAB settings, etc.) to non-volatile memory. After running this command, the module starts in the stored state on power-up or after reboot. This command is the equivalent of setting the Module Configuration "Show user boot screen" and "Show boot screen until user input" options.

Request packet format:

```
type: 0x04 = 410  
data_length = 0
```



Successful return packet format:

```
type: 0x40 | 0x04 = 0x44 = 6810  
data_length = 0
```

If the current state and the boot state do not match after saving, the module returns an error instead of an ACK. In this unlikely error case, the boot state will be undefined.

## 5 (0x05): Reboot Module

Reboots the module to its default power-on or user-saved state. This can be useful if a number of commands have been sent which have put the module into an unknown state. It may also be used to reset the module to default settings.

### CFA634 Module Reboot

Rebooting the CFA634 may be used to test the boot configuration or to re-enumerate connected devices on the 1-Wire bus (i.e., [WR-DOW-Y17](#) temperature sensors). The CFA634 returns the acknowledge packet immediately; then reboots itself.

At boot-up, there is up to a 500-millisecond delay between power-on and turning on the fans. By default, all fans are set to “on” at 100%. When fans are not used, set power to 0% (command [17 \(0x11\): Set Fan Power](#)), and save this setting as the default boot state (command [4 \(0x04\): Store Current State as Boot State](#)).

Request packet format:

```
type: 0x05 = 510  
data_length = 3  
data[0] = 8  
data[1] = 18  
data[2] = 99
```

### Reboot and Reset CFA634 to Factory Defaults

After the reboot and reset the module is configured to the [module interface default settings as described in section 6.2](#). This may be different from when first supplied by Crystalfontz. All user configured settings including interface settings, saved boot screen, etc. will be reset.

If an FBSCAB is connected, this command also reset it to default settings.

Request packet format:

```
type: 0x05 = 510  
data_length = 3  
data[0] = 10  
data[1] = 8  
data[2] = 98
```

In the above cases, successful return packet format:

```
type: 0x40 | 0x05 = 0x45 = 6910  
data_length = 0
```



## 6 (0x06): Clear LCD Screen

Sets the contents of the LCD DDRAM to ' '=0x20=32 and moves the cursor to the left-most column of the top line. The DDRAM contents is stored by the command [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type: 0x06 = 610  
data_length = 0
```

Successful return packet format:

```
type: 0x40 | 0x06 = 0x46 = 7010  
data_length = 0
```

## 9 (0x09): Set LCD Special Character Data

Sets the font definition for one of the special characters (CGRAM).

Set LCD Special Character Data is stored by the command [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type: 0x09 = 910  
data_length = 9  
data[0] = index of special character to modify (0-7 valid)  
data[1-8] = bitmap of the new font for this character  
    data[1] is at the top of the cell.  
    data[8] is at the bottom of the cell.  
    any value is valid between 0 and 63.  
    the msb is at the left of the character cell  
    lsb is at the right of the character cell.
```

Successful return packet format:

```
type: 0x40 | 0x09 = 0x49 = 7310  
data_length = 0
```

## 10 (0x0A): Read 8 Bytes of LCD Memory

This command returns the contents of the LCD's DDRAM or CGRAM for debugging purposes.

**Note:** Firmware version prior to v1.9 did not return the address code.

Request packet format:

```
type: 0x0A = 1010  
data_length = 1  
data[0] = LCD address code of desired data  
Valid LCD address codes:  
    0x40 (64) to 0x7F (127) for CGRAM  
    0x80 (128) to 0x93 (147) for DDRAM, line 0  
    0xA0 (160) to 0xB3 (179) for DDRAM, line 1  
    0xC0 (192) to 0xD3 (211) for DDRAM, line 2  
    0xE0 (224) to 0xF3 (243) for DDRAM, line 3  
    (returns an if address is outside of these values)
```

Successful return packet format:

```
type: 0x40 | 0x0A = 0x4A = 7410
```



```
data_length = 9
data[0] requested address code.
data[1-8] requested data read from the LCD controller's memory.
```

### 11 (0x0B): Set LCD Cursor Position

This command places the cursor at the desired location on the CFA634's LCD screen. For the cursor to be visible, command [12 \(0x0C\): Set LCD Cursor Style](#) may also need to be set.

Set LCD Cursor Position is stored by command [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type: 0x0B = 1110
data_length = 2
data[0] = column (0-19 valid)
data[1] = row (0-3 valid)
```

Successful return packet format:

```
type: 0x40 | 0x0B = 0x4B = 7510
data_length = 0
```

### 12 (0x0C): Set LCD Cursor Style

This command selects among four hardware generated cursor options.

Set LCD Cursor Style is stored by command [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type: 0x0C = 1210
data_length = 1
data[0] = cursor style (0-4 valid)
  0 = no cursor.
  1 = blinking block cursor.
  2 = static underscore cursor
  3 = blinking underscore cursor
```

Successful return packet format:

```
type: 0x40 | 0x0C = 0x4C = 7610
data_length = 0
```

### 13 (0x0D): Set LCD Contrast

Sets contrast or vertical viewing angle of the display. Initiated by the host, responded to by the CFA634. Set LCD Contrast is stored by command [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type: 0x0D = 1310
data_length = 1
data[0] = contrast setting (0-254 valid)
  60 = light
  120 = about right
  150 = dark
  151-254 = very dark (may be useful at cold temperatures)
```



Successful return packet format:

```
type = 0x40 | 0x0D = 0x4D = 7710
data_length = 0
```

#### 14 (0x0E): Display Backlight Brightness

Sets the brightness of the display backlight as a percentage of fully bright. Set LCD Backlight is stored by command [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type: 0x0E = 1410
data_length = 1
data[0] = LCD backlight power setting (0-100 valid)
  0 = off
  1-100 = variable brightness
```

The return packet for both of the above options will be:

```
type: 0x40 | 0x0E = 0x4E = 7810
data_length: 0
```

#### 16 (0x10): Set Up Fan Reporting (FBSCAB Required)

Configures the CFA634 + [FBSCAB](#) to report the fan speed information to the host every 500 mS.

Reporting a fan will override the fan power setting to 100% for up to 1/8 of a second every 1/2 second. See Fan Connections in the [FBSCAB Datasheet](#) for a detailed description.

Request packet format:

```
type = 0x10 = 1610
data_length = 1
data[0] = bitmask indicating which fans are enabled to report (0-15 valid)
  ---- 8421 Enable Reporting of this Fan's Tach Input
  |||| |--- Fan 1: 1 = enable, 0 = disable
  |||| |--- Fan 2: 1 = enable, 0 = disable
  |||| |--- Fan 3: 1 = enable, 0 = disable
  |||| |---- Fan 4: 1 = enable, 0 = disable
```

Successful return packet format:

```
type = 0x40 | 0x10 = 0x50 = 8010
data_length = 0
```

If data[0] is not 0, the CFA634 + FBSCAB sends 0x81: Fan Speed Report packets for each enabled fan every 500 mS, see [0x81: Fan Speed Report](#). Each of the report packets is staggered by 1/8 of a second.

#### 17 (0x11): Set Fan Power (FBSCAB Required)

When power is applied to the CFA634 + FBSCAB + WR-DOW-Y17 temperature sensors, the CFA634 detects any devices (WR-DOW- Y17) connected to the Dallas Semiconductor 1-Wire (DOW) bus and stores the devices' information. This command allows the host to read the device information.





Set Fan Power is stored by command [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type = 0x11 = 1710
data_length = 4
data[0] = power level for FAN 1 (0-100 valid)
data[1] = power level for FAN 2 (0-100 valid)
data[2] = power level for FAN 3 (0-100 valid)
data[3] = power level for FAN 4 (0-100 valid)
```

Successful return packet format:

```
type = 0x40 | 0x11 = 0x51 = 8110
data_length = 0
```

### 18 (0x12): Read DOW Device Information (FBSCAB Required)

When an FBSCAB module is connected to the CFA634, it detects any devices ([WR-DOW-Y17](#) temperature sensors) connected to the Dallas Semiconductor 1-Wire (DOW) bus and stores the device's information. This command will allow the host to read the device information.

The first byte returned is the Family Code of the Dallas 1-Wire device.

Request packet format:

```
type: 0x12 = 1810
data_length = 1
data[0] = device index (0-15 valid)
```

Successful return packet format:

```
type: 0x40 | 0x12 = 0x52 = 8210
data_length = 9
data[0] = device index (0-15 valid)
data[1-8] = ROM ID of the device
```

If data[1] is 0x22 ([DS1822](#) Econo 1-Wire Digital Thermometer temperature sensor) or 0x28 ([DS18B20](#) High Precision 1-Wire Digital Thermometer temperature sensor used on our [WR-DOW-Y17](#)), then that device can be set up to automatically convert and report the temperature every second. See the command [19 \(0x13\): Set Up Temperature Reporting](#).

### 19 (0x13): Set Up Temperature Reporting (FBSCAB Required)

This command configures the CFA634 to report the temperature information to the host every second.

Any sensor enabled must have been detected as a 0x22 (DS1822 temperature sensor) or 0x28 (DS18B20 temperature sensor) during DOW enumeration. This can be verified by using the command [18 \(0x12\): Read DOW Device Information](#).

Request packet format:

```
type: 0x13 = 1910
data_length = 4
data[0-3] = 32-bit bitmask indicating which temperature sensors fans are
```



```

enabled to report (0-255 valid in each location)
data[0]:
08 07 06 05 04 03 02 01 Enable Reporting of sensor with index of:
| | | | | | | |-- 0: 1 = enable, 0 = disable
| | | | | | |----- 1: 1 = enable, 0 = disable
| | | | | |----- 2: 1 = enable, 0 = disable
| | | | |----- 3: 1 = enable, 0 = disable
| | | |----- 4: 1 = enable, 0 = disable
| | |----- 5: 1 = enable, 0 = disable
| |----- 6: 1 = enable, 0 = disable
|----- 7: 1 = enable, 0 = disable

data[1]:
16 15 14 13 12 11 10 09 Enable Reporting of sensor with index of:
| | | | | | | |-- 8: 1 = enable, 0 = disable
| | | | | | |----- 9: 1 = enable, 0 = disable
| | | | | |----- 10: 1 = enable, 0 = disable
| | | | |----- 11: 1 = enable, 0 = disable
| | | |----- 12: 1 = enable, 0 = disable
| | |----- 13: 1 = enable, 0 = disable
| |----- 14: 1 = enable, 0 = disable
|----- 15: 1 = enable, 0 = disable

data[2]= must be 0
data[3]= must be 0

```

Successful return packet format:

```

type: 0x40 | 0x13 = 0x53 = 8310
data_length = 0

```

## 20 (0x14): Arbitrary DOW Transaction (FBSCAB Required)

The CFA634 + FBSCAB can function as an CFA-RS232 to Dallas 1-Wire bridge. This command specifies arbitrary transactions on the 1-Wire bus. 1-Wire commands follow this basic layout:

```

<bus reset>          //Required
<address_phase>      //Must be "Match ROM" or "Skip ROM"
<write_phase>        //optional, but at least one of write_phase or
read_phase must be sent
<read_phase>         //optional, but at least one of write_phase or
read_phase must be sent

```

Request packet format:

```

type: 0x14 = 2010
data_length = 2 to 16
data[0] = device_index (0-32 valid)
data[1] = number_of_bytes_to_read (0-14 valid)
data[2-15] = data_to_be_written[data_length-2]

```

If `device_index` is 32, then no address phase will be executed. If `device_index` is in the range of 0 to 31, and a 1-Wire device was detected for that `device_index` at power on, then the write cycle will be prefixed with a "Match ROM" command and the address information for that device.

If `data_length` is two, then no specific write phase will be executed (although address information may be written independently of `data_length` depending on the value of `device_index`).



If `data_length` is greater than two, then `data_length-2` bytes of `data_to_be_written` will be written to the 1-Wire bus immediately after the address phase.

If `number_of_bytes_to_read` is zero, then no read phase will be executed. If `number_of_bytes_to_read` is not zero, then `number_of_bytes_to_read` will be read from the bus and loaded into the response packet.

Successful return packet format:

```
type: 0x40 | 0x14 = 0x54 = 8410
data_length = 2 to 16
data[0] = device index (0-31 valid)
data[data_length-2] = data read from the 1-Wire bus
data[data_length-1] = 1-Wire CRC
```

## 25 (0x19): Set Fan Power Fail-Safe (FBSCAB Required)

The combination of the CFA634 + FBSCAB + WR-FAN-X01 cable can be used as part of an active cooling system. The fans can be slowed down to reduce noise when a system is idle or when the ambient temperature is low. The fans speed up when the system is under heavy load or the ambient temperature is high.

Varying fan speeds under host software control gives the ultimate flexibility in system design but would typically have a fatal flaw: a host software or hardware failure could cause the cooling system to fail. If the fans were set at a slow speed when the host software failed, system components may be damaged due to inadequate cooling.

The fan power fail-safe command allows host control of the fans without compromising safety. When the fan control software activates, it should set the fans that are under its control to fail-safe mode with an appropriate timeout value. If for any reason the host fails to update the power of the fans before the timeout expires, the fans previously set to fail-safe mode will be forced to 100% power.

Bitmask options:

```
#define FAN_1      0x01
#define FAN_2      0x02
#define FAN_3      0x04
#define FAN_4      0x08
```

Request packet format:

```
type = 0x19 = 2510
data_length = 2
data[0] = bit mask of fans set to fail-safe (1-15 valid)
data[1] = timeout value in 1/8 second ticks:
  1 = 1/8 second
  2 = 1/4 second
  255 = 31 7/8 seconds
```

Successful return packet format:

```
type = 0x40 | 0x19 = 0x59 = 8910
data_length = 0
```



## 26 (0x1A): Set Fan Tachometer Glitch Filter (FBSCAB Required)

The combination of the CFA634 + [FBSCAB](#) + [WR-FAN-X01](#) cable controls fan speed using PWM. PWM turns the power to a fan on and off quickly to change the average power delivered to the fan. The CFA634 uses approximately 18 Hz for the PWM repetition rate. The fan's tachometer output is only valid if power is applied to the fan. Most fans produce a valid tachometer output very quickly after the fan has been turned back on but some fans take time after being turned on before their tachometer output is valid.

This command allows a variable-length delay to be set for after the fan has been turned on before the CFA634 recognizes transitions on the tachometer line. The delay is specified in counts, each count being nominally 552.5  $\mu$ S long (1/100 of one period of the 18 Hz PWM repetition rate).

In practice, most fans will not need the delay to be changed from the default length of 1 count. If a fan's tachometer output is not stable when its PWM setting is other than 100%, simply increase the delay until the reading is stable.

Typically:

- (1) start at a delay count of 50 or 100
- (2) reduce it until the problem reappears
- (3) slightly increase the delay count to give it some margin.

Setting the glitch delay to higher values makes the RPM monitoring slightly more intrusive at low power settings. Also, the higher values increase the lowest speed that a fan with RPM reporting enabled will seek at 0% power setting.

The Fan Glitch Delay is one of the items stored by [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type = 0x1A = 2610
data_length = 4
data[0] = delay count of fan 1 (0 to 100 valid)
data[1] = delay count of fan 2 (0 to 100 valid)
data[2] = delay count of fan 3 (0 to 100 valid)
data[3] = delay count of fan 4 (0 to 100 valid)
```

Successful return packet format:

```
type = 0x40 | 0x1A = 0x5A = 9010
data_length = 0
```

## 27 (0x1B): Query Fan Power & Fail-Safe Mask (FBSCAB Required)

The combination of the CFA634 + FBSCAB + WR-FAN-X01 cable is required to use this command. This command can be used to verify the current fan power and verify which fans are set to fail-safe mode.

Bitmask options:

```
#define FAN_1          0x01
#define FAN_2          0x02
#define FAN_3          0x04
#define FAN_4          0x08
```



Request packet format:

```
type = 0x1B = 2710  
data_length = 0
```

Successful return packet format:

```
type = 0x40 | 0x1B = 0x5B = 9110  
data_length = 5  
data[0] = fan 1 power  
data[1] = fan 2 power  
data[2] = fan 3 power  
data[3] = fan 4 power  
data[4] = bitmask of fans with fail-safe set
```

### 30 (0x1E): Read Reporting & Status

This command can be used to verify the current items configured to report to the host, as well as some other miscellaneous status information. The combination of CFA634 + [FBSCAB](#) + [WR-DOW-Y17](#) temperature sensors is required to report the temperature information. The combination of the CFA634 + [FBSCAB](#) + [WR-FAN-X01](#) cable is required to control fans.

Request packet format:

```
type = 0x1E = 3010  
data_length = 0
```

Successful return packet format:

```
type = 0x40 | 0x1E = 0x5E = 9410  
data_length = 15  
data[ 0] = fan 1-4 reporting status (as set by command 16)  
data[ 1] = temperatures 1-8 reporting status (as set by command 19)  
data[ 2] = temperatures 9-15 reporting status (as set by command 19)  
data[ 3] = temperatures 16-23 reporting status (as set by command 19)  
data[ 4] = temperatures 24-32 reporting status (as set by command 19)  
data[ 5] = always 0  
data[ 6] = always 0  
data[ 7] = always 0  
data[ 8] = always 0  
data[ 9] = fan RPM glitch delay[0] (as set by command 26)  
data[10] = fan RPM glitch delay[1] (as set by command 26)  
data[11] = fan RPM glitch delay[2] (as set by command 26)  
data[12] = fan RPM glitch delay[3] (as set by command 26)  
data[13] = contrast setting (as set by command 13)  
data[14] = backlight setting (as set by command 14)
```

**NOTE:** Previous and future firmware versions may return fewer or additional bytes.



### 31 (0x1F): Send Data to LCD

Place data at any position on the LCD.

Send Data to LCD is stored by command [4 \(0x04\): Store Current State as Boot State](#).

Request packet format:

```
type: 0x1F = 3110
data_length = 3 to 22
data[0]: col = x = 0 to 19
data[1]: row = y = 0 to 3
data[2-21]: characters/text to place on the LCD
```

Successful return packet format:

```
type: 0x40 | 0x1F = 0x5F = 9510
data_length = 0
```

### 33 (0x21): Set Communications Interface Options

Five interface options are available on the CFA634 module. USB 2.0, Logic-level serial, Full-swing RS232 serial, I<sup>2</sup>C (peripheral mode) and SPI (peripheral mode). All interfaces may be used at any time (when enabled with this command) including being used simultaneously. All interfaces may operate in either Legacy serial-stream or CrystalFontz packet-based mode.

Settings changed by this command must be saved by using the [Save Settings](#) or [Store Current State](#) commands to persist after power-up/restart.

Option flags (applies to all interfaces packet commands):

```
bit0 = enable interface
NOTE: USB interface cannot be fully disabled

bit1 = enable the CrystalFontz packet-based interface interpreter
NOTE: can only be set if bit6 is not set
NOTE: CFA634 will accept/send packets on this interface. Interface must
be enabled for interpreter on an interface to be enabled. normal reply
packets are only sent to the originating interface. the following
options are only available if the interpreter is enabled

bit2 = send FBSCAB report packets on this interface
NOTE: only applies if bit1 is set

bit3 = send errors from commands received on this interface
NOTE: only applies if bit1 is set

bit4 = send errors from commands received on any interface
NOTE: only applies if bit1 is set

bit5 = RESERVED

bit6 = enable the Legacy serial-stream interface interpreter
NOTE: can only be set if bit1 is not set
```



## Logic-level Serial Interface

Request packet format:

```
type: 0x21 = 3310
data_length = 4
data[0]: interface = 0 (logic-level serial)
data[1]: interface option flags (see above)
data[2]: baud rate:
    0 = 19200 (default)
    1 = 115200
    2 = 9600
    3 = 4800
    4 = 2400
data[3] : inverted level (0=no 1=yes)
```

## USB Interface

Request packet format:

```
type: 0x21 = 3310
data_length = 2
data[0]: interface = 1 (USB)
data[1]: interface option flags (see above)
```

## SPI Interface

Request packet format:

```
type: 0x21 = 3310
data_length = 4
data[0]: interface = 2 (SPI)
data[1]: interface option flags (see above)
data[2]: SPI mode:
    bit0: SPI phase (0=first edge, 1=second edge)
    bit1: SPI polarity (0=low, 1=high)
    bit2: SPI bit order (0 = msb first, 1 = lsb first)
data[3] = 0 (reserved, set to 0)
```

## I<sup>2</sup>C Interface

Request packet format:

```
type: 0x21 = 3310
data_length = 4
data[0]: interface = 3 (I2C)
data[1]: interface option flags (see above)
data[2]: I2C address = 0 to 127 (default 0x42)
data[3]: I2C bus speed (0=100kHz, 1=400kHz) (default 100kHz)
```

Successful return packet format for all interface configuration packets:

```
type: 0x40 | 0x21 = 0x61 = 9710
data_length = 0
```





## Full-swing RS232 serial Interface

Request packet format:

```
type: 0x21 = 3310
data_length = 4
data[0]: interface = 4 (full swing rs232)
data[1]: interface option flags (see above)
data[2]: baud rate:
    0 = 19200 (default)
    1 = 115200
    2 = 9600
    3 = 4800
    4 = 2400
data[3] : inverted level (0=no 1=yes)
```



## 12. Character Generator ROM (CGROM)

To find the code for a given character, add the bolded row and column values together. For example, the superscript "9" is in the column labeled "128<sub>d</sub>" and in the row labeled "9<sub>d</sub>". Add 128 to 9 to get 137. Sending a byte with the value of 137 to the display module, will display a superscript "9".

upper 4 bits lower 4 bits	0 <sub>d</sub> 0000.	16 <sub>d</sub> 0001.	32 <sub>d</sub> 0010.	48 <sub>d</sub> 0011.	64 <sub>d</sub> 0100.	80 <sub>d</sub> 0101.	96 <sub>d</sub> 0110.	112 <sub>d</sub> 0111.	128 <sub>d</sub> 1000.	144 <sub>d</sub> 1001.	160 <sub>d</sub> 1010.	176 <sub>d</sub> 1011.	192 <sub>d</sub> 1100.	208 <sub>d</sub> 1101.	224 <sub>d</sub> 1110.	240 <sub>d</sub> 1111.
0 <sub>d</sub> 0000.	CGRAM [0]	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
1 <sub>d</sub> 0001.	CGRAM [1]	F	!	@	A	B	C	D	E	F	G	H	I	J	K	L
2 <sub>d</sub> 0010.	CGRAM [2]	~	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3 <sub>d</sub> 0011.	CGRAM [3]	~	0	1	2	3	4	5	6	7	8	9	A	B	C	D
4 <sub>d</sub> 0100.	CGRAM [4]	~	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
5 <sub>d</sub> 0101.	CGRAM [5]	~	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`
6 <sub>d</sub> 0110.	CGRAM [6]	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n
7 <sub>d</sub> 0111.	CGRAM [7]	~	o	p	q	r	s	t	u	v	w	x	y	z	{	}
8 <sub>d</sub> 1000.	CGRAM [0]	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
9 <sub>d</sub> 1001.	CGRAM [1]	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
10 <sub>d</sub> 1010.	CGRAM [2]	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
11 <sub>d</sub> 1011.	CGRAM [3]	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
12 <sub>d</sub> 1100.	CGRAM [4]	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
13 <sub>d</sub> 1101.	CGRAM [5]	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
14 <sub>d</sub> 1110.	CGRAM [6]	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
15 <sub>d</sub> 1111.	CGRAM [7]	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~

Figure 11. Character Generator ROM (CGROM)



## 13. LCD Module Reliability and Longevity

We work to continuously improve our products, including backlights that are brighter and last longer. Slight color variations from module-to-module and batch-to-batch are normal.

Item	Specification	
Yellow-green LED Backlight	50,000 to 100,000 hours (typical)	
White LED Backlight	Power-On Hours	% of Initial Brightness
	<10,000	>70%
	<50,000	>50%
<i>For display modules with white LED backlights, adjust backlight brightness so the display is readable but not too bright. Dim or turn off the backlight during periods of inactivity to conserve the white LED backlight lifetime.</i>		
<i>Under operating and storage temperature specification limitations, humidity non-condensing RH up to 65%, and no exposure to direct sunlight. Value listed above are approximate and represent typical lifetime.</i>		

### 13.1. Module Longevity (EOL / Replacement Policy)

Crystalfontz is committed to making our LCD modules available for as long as possible. For each module we introduce, we intend to offer it indefinitely. We do not preplan a module's obsolescence. The majority of modules we have introduced are still available.

We recognize that discontinuing a module may cause problems for some customers. However, rapidly changing technologies, component availability, or low customer order levels may force us to discontinue ("End of Life", EOL) a module. For example, we must occasionally discontinue a module when a supplier discontinues a component or a manufacturing process becomes obsolete. When we discontinue a module, we will do our best to find an acceptable replacement module with the same fit, form, and function.

In most situations, there will not be a noticeable difference when comparing a "fit, form, and function" replacement module to the discontinued module it replaces. However, sometimes a change in component or process for the replacement module results in a slight variation, perhaps an improvement, over the previous design.

Although the replacement module is still within the stated Datasheet specifications and tolerances of the discontinued module, changes may require modification to circuits and/or firmware. Possible changes include:

- Backlight LEDs. Brightness may be affected (perhaps the new LEDs have better efficiency) or the current they draw may change (new LEDs may have a different VF).
- Controller. A new controller may require minor code changes.
- Component tolerances. Module components have manufacturing tolerances. In extreme cases, the tolerance stack can change the visual or operating characteristics.

Please understand that we avoid changing a module whenever possible; we only discontinue a module if we have no other option. We post Part Change Notices (PCNs) on the product's website page as soon as possible. If interested, subscribe to future [Part Change Notices](#).



## 14. Care and Handling Precautions

For optimum operation and lifetime of the CFA634, follow the precautions described below.

### 14.1. ESD (Electrostatic Discharge)

The circuitry is industry standard CMOS logic and susceptible to ESD damage. Use industry standard antistatic precautions for sensitive devices like expansion cards, motherboards, or integrated circuits. Ground personnel, work surfaces, and equipment.

### 14.2. Design and Mounting

- The exposed surface of the “glass” is a polarizer laminated on top of glass. To protect the soft plastic polarizer from damage, the module ships with a protective film over the polarizer. Peel the protective film slowly. Peeling the protective film abruptly may generate static electricity.
- When handling the module, avoid touching the polarizer. Finger oils are difficult to remove.
- To protect the polarizer from damage, place a transparent plate (e.g., acrylic, polycarbonate, or glass) in front of the module, leaving an air gap between the plate and the display surface.
- Do not disassemble or modify the module.
- Do not modify the tabs of the metal bezel or make connections to them.
- Do not reverse polarity to the power supply connections. This immediately ruins the module.

### 14.3. Avoid Shock, Impact, Torque, or Tension

- Do not expose the CFA634 to strong mechanical shock, impact, torque, or tension.
- Do not drop, toss, bend, or twist the CFA634.
- Do not place weight or pressure on the CFA634.

### 14.4. 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 warm soapy water.

### 14.5. Cleaning

- The polarizer (laminated to the glass), is soft plastic that can easily be scratched or damaged, so use extra care when you clean it.
- Do not clean the polarizer with liquids.
- Do not wipe the polarizer with any type of cloth or swab (for example, Q-tips).
- Use the removable protective film to remove smudges (for example, fingerprints), and any foreign matter. If you no longer have the protective film, use standard transparent office tape (for example, Scotch® brand “Crystal Clear Tape”).
- If the polarizer becomes dusty, carefully blow it off with clean, dry, oil-free compressed air.
- The polarizer will eventually become hazy if you do not use care when cleaning it.
- Contact with moisture may permanently spot or stain the polarizer.

### 14.6. Operation

- Protect the CFA634 from ESD and power supply transients.
- Observe the operating temperature limits: a minimum of -20°C to a maximum of +70°C with minimal fluctuation. Operation outside of these limits may shorten life or harm the module.
- At lower temperatures of this range, response time is delayed.
- At higher temperatures of this range, display becomes dark. (Adjust the contrast.)

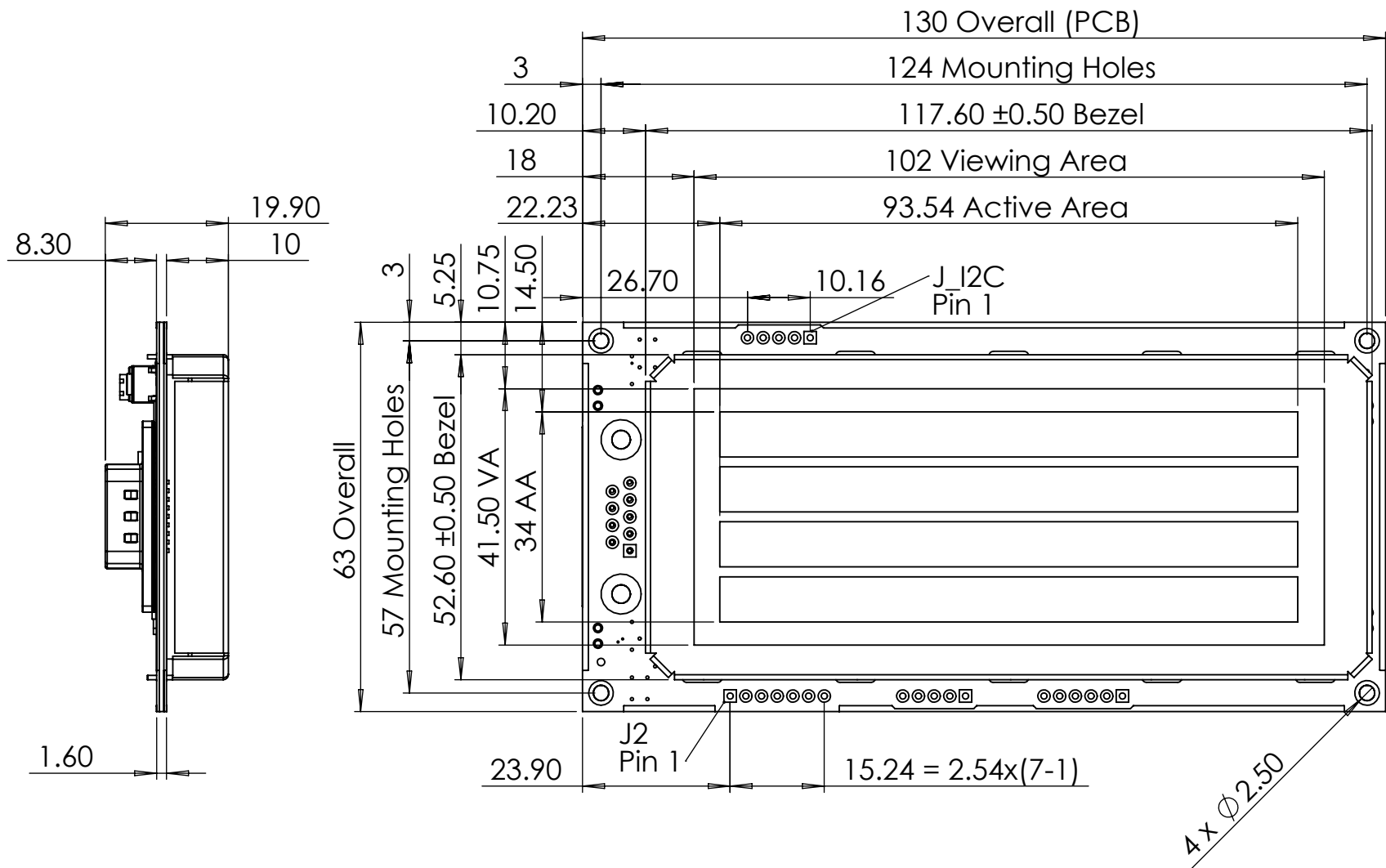


- Operate away from dust, moisture, and direct sunlight.
- Adjust backlight brightness so the display is readable, but not too bright.
- Dim or turn off the backlight during periods of inactivity to conserve the backlight lifetime.


#### 14.7. Storage and Recycling

- Store in an ESD-approved container away from dust, moisture, and direct sunlight.
- Observe the storage temperature limitations: -30°C minimum, +80°C maximum with minimal fluctuation. Rapid temperature changes cause moisture to form, resulting in permanent damage.
- Do not allow weight to be placed on the CFA634 while in storage.
- Recycle outdated CrystalFontz modules at an approved facility.

15. Mechanical Drawing



Units: millimeters  
Tolerance: ±0.3mm

 Crystalfontz © 2023 Crystalfontz America, Inc.	Part Number:	Date:	Filename:	HW Revision:
	CFA634 Family	2023-11-02	CFA634 Family Drawing.pdf	4v0
		Web:	www.crystalfontz.com/cfa634fhn	Sheet:
				1 of 2



## 16. Appendix A: Demonstration Software and Sample Code

### 16.1. Demonstration Software

Demonstration software is available for free download under the LCD Software tab on the website page. Or click on the links in the software descriptions below. No registration is required for download.

#### 16.1.1. cfTest

[cfTest](#) for Windows is testing and configuration software that works on all Crystallfontz Intelligent LCD modules. This software allows experimentation with the command set for all Crystallfontz Intelligent LCDs.

Streaming communication-based modules (CFA632, CFA634) and packet communication-based modules (CFA533, CFA631, CFA633, CFA634, CFA735, CFA835) are supported. Microsoft Windows®, Linux, and Mac (OSX) versions are available.

#### 16.1.2. CrystalControl2 (CC2)

For use with USB and Serial interfaces. [CrystalControl2](#) for Windows displays a great variety of varying information to a Crystallfontz Intelligent LCD Modules in a configurable way. We provide a [User Manual](#) and support through our [forum](#).

#### 16.1.3. Linux Examples

Linux examples are available on the [Crystallfontz Github](#).