



## INTELLIGENT LCD MODULE SPECIFICATIONS



Hardware Version: h1v4  
Firmware Version: f1v0

Datasheet Release: 2025-05-06

### **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. EXAMPLE USES .....	6
2.2. MAIN FEATURES .....	6
2.3. MODULE CLASSIFICATION INFORMATION .....	7
2.4. MODULE TERMINOLOGY .....	7
2.5. ACCESSORIES AND CABLES .....	8
2.5.1. CFA-FBSCAB.....	8
2.5.2. USB Cables.....	8
2.5.3. Accessory Cables.....	9
<b>3. MECHANICAL CHARACTERISTICS .....</b>	<b>10</b>
3.1. PHYSICAL CHARACTERISTICS .....	10
3.2. OPTICAL CHARACTERISTICS.....	10
3.3. LED INFORMATION .....	10
3.3.1. TFT BACKLIGHT LEDs .....	11
3.3.2. STATUS LEDs.....	11
3.3.3. KEYPAD LEDs .....	11
<b>4. ELECTRICAL SPECIFICATIONS .....</b>	<b>12</b>
4.1. ABSOLUTE MAXIMUM RATINGS.....	12
4.2. OPERATING CONDITIONS .....	12
4.3. GPIO CURRENT LIMITS .....	12
<b>5. CONNECTION INFORMATION .....</b>	<b>13</b>
5.1. LOCATION OF CONNECTORS .....	13
5.2. H1 CONNECTOR DETAILS .....	13
5.2.1. H1 Connector Pinout.....	14
5.2.2. H1 Pins 5 and 6 - ADC.....	14
5.3. H2 CONNECTOR.....	14
5.4. H3 CONNECTOR.....	15
5.5. MICROSD CARD SLOT.....	15
5.6. JUMPER CONFIGURATIONS.....	15
<b>6. ATX POWER SUPPLY AND CONTROL CONNECTIONS.....</b>	<b>16</b>
<b>7. FIRMWARE.....</b>	<b>18</b>
7.1. HOW TO IDENTIFY FIRMWARE REVISION NUMBER.....	18
7.2. FIRMWARE PROTECTIONS .....	18
7.2.1. WATCHDOG.....	18
7.2.2. FIRMWARE PROGRAM STORAGE.....	18
7.3. MODULE BOOTLOADER .....	18
7.3.1. ENTERING THE BOOTLOADER .....	18
7.3.2. BOOTLOADER HARDWARE TESTING.....	18
7.4. FUTURE FIRMWARE UPDATES .....	19
7.4.1. Firmware Update (From microSD or On-board Flash).....	19
7.4.2. Firmware Update (From Host) .....	19
7.5. MAIN FIRMWARE SPECIAL MODES .....	19

7.5.1.	Check Module Version .....	19
7.5.2.	Module Settings Reset .....	19
7.5.3.	USB Mass Storage Access Mode .....	19
7.5.4.	USB Debugging Output .....	20
<b>8.</b>	<b>COMMUNICATION INTERFACES.....</b>	<b>21</b>
8.1.	USB INTERFACE.....	21
8.2.	UART SERIAL INTERFACE (LOGIC LEVEL, INVERTED) .....	21
8.3.	UART SERIAL INTERFACE (FULL-SWING RS232) .....	21
8.4.	I2C TARGET INTERFACE .....	22
8.5.	SPI PERIPHERAL INTERFACE.....	22
<b>9.</b>	<b>NON-VOLATILE DATA STORAGE .....</b>	<b>23</b>
9.1.	ON-BOARD FLASH .....	23
9.2.	MICROSD CARD.....	23
<b>10.</b>	<b>COMMAND PACKET COMMUNICATIONS .....</b>	<b>24</b>
10.1.	HANDSHAKING / FLOW CONTROL .....	24
10.2.	PACKET TYPES.....	24
10.3.	DEBUGGING COMMUNICATIONS .....	24
<b>11.</b>	<b>GRAPHICS .....</b>	<b>25</b>
11.1.	GRAPHICAL OBJECTS .....	25
11.2.	GRAPHICAL OBJECT TYPES.....	25
11.3.	GRAPHICAL OBJECT STYLING.....	26
11.4.	DISPLAY COMPOSITION.....	27
11.5.	DISPLAY AND OBJECT COORDINATES .....	27
11.6.	GRAPHICS MEMORY .....	27
11.7.	TOUCHSCREEN FEATURES .....	27
<b>12.</b>	<b>COMMAND CODES.....</b>	<b>28</b>
<b>13.</b>	<b>BACKWARDS COMPATIBILITY .....</b>	<b>30</b>
13.1.	FUNCTIONAL DIFFERENCES .....	30
13.2.	FUNCTIONAL DIFFERENCES .....	34
13.3.	LEGACY CHARACTER GENERATOR ROM (CGROM) .....	35
<b>14.</b>	<b>MODULE RELIABILITY AND LONGEVITY .....</b>	<b>36</b>
14.1.	MODULE LONGEVITY (EOL / REPLACEMENT POLICY) .....	36
<b>15.</b>	<b>CARE AND HANDLING PRECAUTIONS.....</b>	<b>37</b>
15.1.	ESD (ELECTROSTATIC DISCHARGE) .....	37
15.2.	DESIGN AND MOUNTING.....	37
15.3.	AVOID SHOCK, IMPACT, TORQUE, OR TENSION.....	37
15.4.	IF LCD PANEL BREAKS .....	37
15.5.	CLEANING .....	37
15.6.	OPERATION .....	37
15.7.	STORAGE AND RECYCLING.....	37
<b>16.</b>	<b>MECHANICAL DRAWINGS .....</b>	<b>38</b>
<b>APPENDIX A: DEMONSTRATION SOFTWARE AND SAMPLE CODE .....</b>		<b>41</b>
A.	CRYSTALFONTZ CFTEST .....	41
B.	LINUX CLI EXAMPLES .....	41



C.	SAMPLE CODE FOR RPM CALCULATION INFORMATION.....	41
D.	SAMPLE CODE FOR TEMPERATURE SENSOR REPORT.....	42

## 1. General Information

Datasheet Revision
<p>Hardware Version: h1v4  Firmware Version: f1v0  Datasheet Release: 2025-05-06</p> <p>For information about firmware and hardware revisions, see the <a href="#">Part Change Notifications (PCN)</a></p> <p>For reference, previous datasheets may be downloaded by clicking the “Show Previous Versions of Datasheet” link under the “Datasheets” tab of the product web page.</p>
Product Change Notifications
<p>Check for or subscribe to <a href="#">Part Change Notices</a> for this display module on our website.</p>
Variations
<p>Slight variations between lots are normal (e.g., contrast, color, or intensity).</p>
Volatility
<p>This display module has volatile and non-volatile memory.</p>
Disclaimer
<p>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.</p> <p>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.</p> <p>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.</p> <p>Company and product names mentioned in this publication are trademarks or registered trademarks of their respective owners.</p> <p>Copyright © 2025 by Crystalfontz America, Inc., 12412 East Saltese Avenue, Spokane Valley, WA 99216 U.S.A.</p>

## 2. Introduction

The CrystalFontz CFA039A0-N-VDxT family of intelligent graphic display modules packs a lot of features into a compact and powerful module. These modules provide an easy to use, highly configurable display and user interface with various input/output capabilities with minimal host system supervision.

This document includes primarily hardware specifications. For information on the in-built command set, refer to the [CFA039A0 Command Reference Document](#), downloadable from the Files and Downloads tab on the product page: <https://www.crystallfontz.com/product/cfa039a0nvdct#docs>.

### 2.1. Example Uses

- Machine control: the touchscreen and keypad allow for simple and clear machine operation
- Monitoring: display real-time data such as temperature, pressure, and speed from a host machine
- Alarms: alert operators to abnormal conditions or failures through bright visual alarms
- Configuration: use visual buttons and controls to configure host parameters
- Thermal control: [CFA-FBSCAB modules](#) can be added to automatically monitor temperatures and control system cooling
- General purpose I/O: use the GPIOs to control external circuitry, or read digital or analog inputs
- Visual appeal: add animations and bright colorful images and interfaces to your product

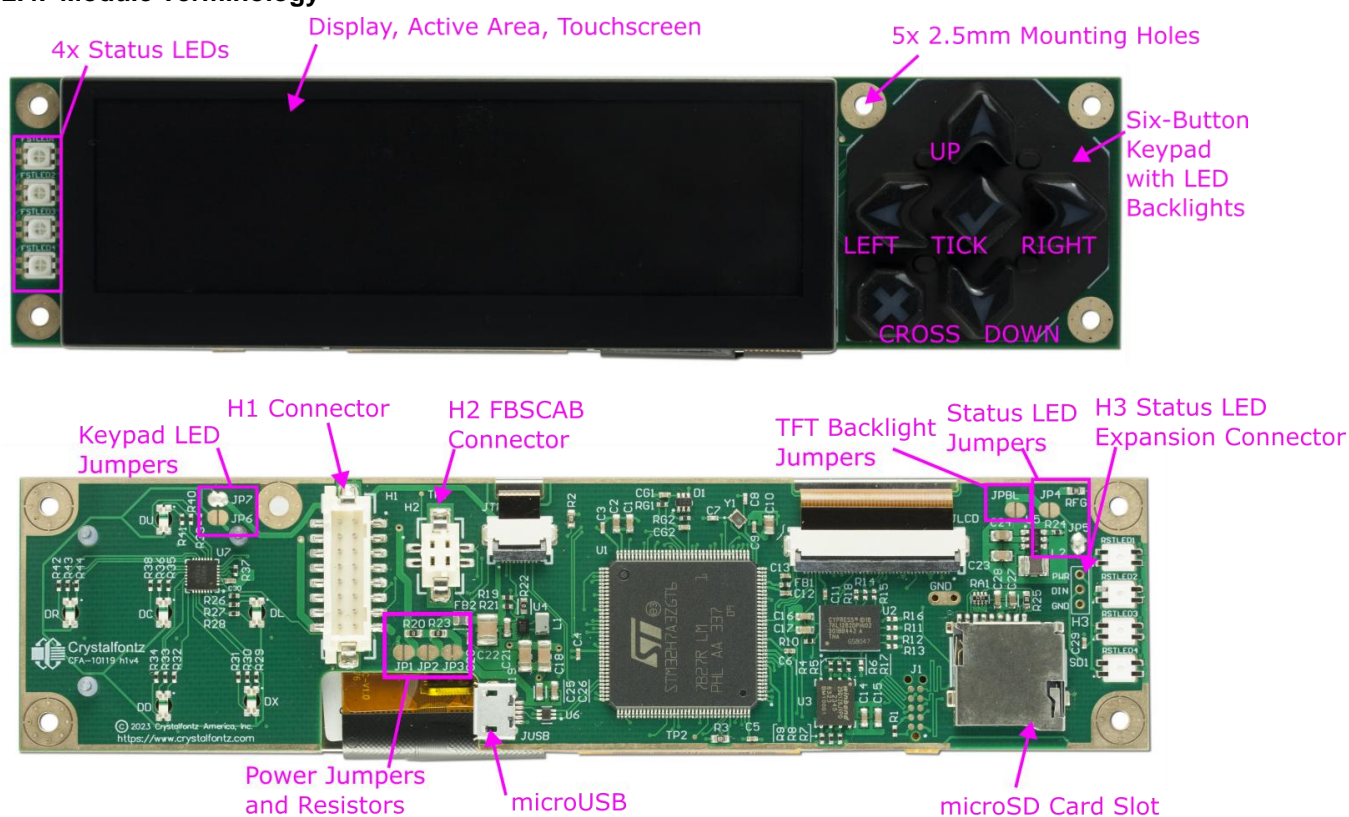
### 2.2. Main Features

- Bright backlit, 480x128 pixel, full-color TFT display with multi-point touch sensing (optional)
- 6-key keypad with individual RGB backlit lighting
- Four bright RGB general purpose indication LEDs
- Single wide-voltage power supply (+2.5v to +5.5v)
- Robust, packet-based communication protocol with 16-bit CRC for error-free communications
- Four host communication interfaces:
  - USB 2.0 Micro-B (virtual serial interface)
  - Serial logic-level ([RS232 with optional interface board](#))
  - I2C target (100Khz and 400Khz), with data-ready output
  - SPI peripheral, with data-ready output
- Two non-volatile data storage options for module settings, and user-data:
  - 16M-bytes on-board NOR flash
  - microSD card slot (microSDHC with FAT16/FAT32 filesystem)
- Configurable 13 pin I/O interface for custom host-controlled monitoring/control applications
- Slim and compact form factor
- Wide operating temperature range of -20°C to +70°C
- Packet based command set supports such features as:
  - Text display using true-type font (TTF) files from on-board flash or microSD
  - Display and state reporting of customizable touch-based controls
  - Display of image and video files from on-board flash or microSD
  - Configuration of module lighting (display backlight, RGB color of individual keys and indication LEDs)
  - GPIO state monitoring, ADC, and output control
  - Manual and automatic fan control (when using [CFA-FBSCAB](#) module)
  - Host ATX power/reset control
  - Storage data access functions
- Support for one or more [CFA-FBSCAB](#) modules, providing temperature monitoring and fan power control abilities (including automatic fan control)
- Host ATX power control (power and reset control via keypad and/or packet commands)
- Hardware watchdog can reset host on software/hardware failure
- Partial backwards compatibility with CrystalFontz CFA635, CFA735, and CFA835 modules
- Field upgradable firmware using a host or microSD card
- Wide range of configuration, testing, and other supporting software and source-code examples

### 2.3. Module Classification Information

<b>CFA</b>	<b>Brand</b>	Crystalfontz America, Inc.
<b>039</b>	<b>Diagonal Dimension</b>	0.39" diagonal
<b>A0</b>	<b>Series and Version</b>	Series A version 0
<b>N</b>	<b>Family</b>	N = Intelligent
<b>V</b>	<b>Type</b>	V = Value Add Packet Based
<b>D</b>	<b>Aspect Ratio</b>	D = 15:4
<b>x</b>	<b>Touch Option</b>	C = Capacitive Touch N = Non-Touch
<b>T</b>	<b>Backlight Option</b>	T = Standard Brightness, < 500 nits

### 2.4. Module Terminology





## 2.5. Accessories and Cables

### 2.5.1.CFA-FBSCAB

Connecting an [CFA-FBSCAB module](#) (FBSCAB) to a CFA039A0-N-VDxT module to provides extra I/O functionality. The CFA-FBSCAB must be purchased separately. CFA-FBSCAB modules are added in a daisy-chain fashion. Up to 32 CFA-FBSCAB modules may be attached to a single CFA039A0-N-VDxT.

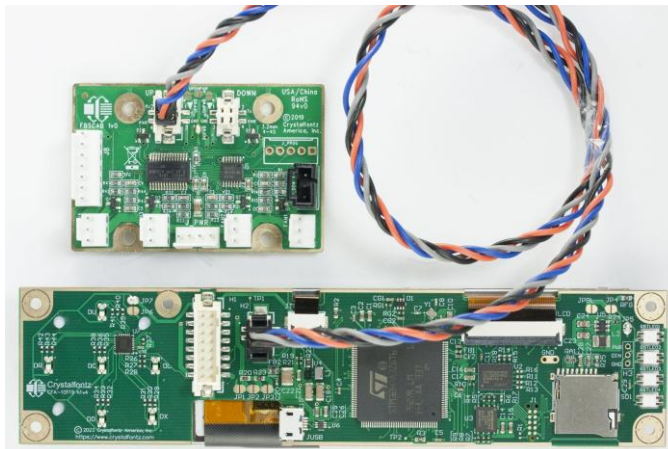


Figure 2. Optional CFA-FBSCAB Connected to the CFA039A0-N-VDxT with WR-EXT-Y37 Cable

The additional functionality offered by an CFA-FBSCAB includes:

- Manual or automatic PWM power control of up to four 12V fans (PC standard, 2- or 3-pin)
- RPM monitoring of up to four attached fans
- Connection of up to 16 Dallas-one-wire (DOW) temperature sensors ([WR-DOW-Y17](#))
- Five additional, host-controlled GPIOs
- See the [CFA-FBSCAB datasheet](#) for more.

When one or more CFA-FBSCAB modules are attached, use command 37 (0x25): CFA-FBSCAB to control or monitor them. See the [CFA039A0 Command Reference](#) for more information on this command.

### 2.5.2.USB Cables

Crystallfontz offers the following USB cables to simplify integrating the CFA039A0-N-VDxT into a system. Cable lengths are approximate. Additional cables may be available on [crystallfontz.com](http://crystallfontz.com).

Cable	Host Connection	Description
<a href="#">WR-USB-Y58</a>	USB C	6-foot, USB-C to micro-USB Cable
<a href="#">WR-USB-Y27</a>	USB A	6-foot, USB-A to micro-USB Cable
<a href="#">WR-USB-Y35</a>	USB A	10-foot, USB-A to micro-USB Cable
<a href="#">WR-USB-Y34</a>	0.1" pitch 4-pin	700mm, 0.1", 4-pin to micro-USB Cable Typically used with motherboards
<a href="#">WR-USB-Y52</a>	2mm pitch 4-pin	700mm, 2mm, 4-pin to micro-USB Cable Typically used with motherboards



### 2.5.3. Accessory Cables

Crystalfontz offers the following cables to simplify integrating the CFA039A0-N-VDxT into a system. Cable lengths are approximate. Additional cables may be available on [crystalfontz.com](http://crystalfontz.com).

Cable	Use	Description
<a href="#">WR-EXT-Y15</a>	General Purpose	16", general purpose H1 extension cable
<a href="#">WR-EXT-Y19</a>	General Purpose	3.5", general purpose H1 extension cable
<a href="#">WR-EXT-Y37</a>	CFA-FBSCAB	18", connect with a CFA-FBSCAB
<a href="#">WR-PWR-Y24</a>	Power	660mm, 3.5" HDD power connector from host to power the module
<a href="#">WR-PWR-Y25</a>	Power, ATX	12", Standard 3.5" HDD power connector from host to power the module with motherboard reset functionality
<a href="#">WR-PWR-Y38</a>	Power, ATX	24", Standard 3.5" HDD power connector from host to power the module with motherboard reset functionality

### 3. Mechanical Characteristics

#### 3.1. Physical Characteristics

Item	Specification (mm)	Specification (inch, reference)
Module Width and Height	152.0 (W) x 37.0 (H)	5.98 (W) x 1.46 (H)
Module Depth with Keypad, with Connectors	20.5	0.82
Active Area (display)	95.04 (W) x 25.34 (H)	3.742 (W) x 0.9976 (H)
Pixel Pitch	0.066 (W) x 0.198 (H)	0.0026 (W) x 0.0078 (H)
Keystroke Travel (approximate)	2.4	0.094
Weight (typical)	49 grams	1.73 ounces
Weight (typical) (with capacitive touch)	72 grams	2.54 ounces

#### 3.2. Optical Characteristics

Item		Symbol	Condition	Min	Typ	Max	Unit
Response Time		T <sub>r</sub>	θ=0°, Φ=0	-	10	-	ms
		T <sub>f</sub>		-	15	-	ms
Contrast Ratio		(CR)	At Optimized Viewing Angle	-	500	-	-
White Chromaticity		W <sub>x</sub>	θ=0°, Φ=0	0.269	0.319	0.369	
		W <sub>y</sub>		0.273	0.323	0.373	
Greyscale Inversion Angle	Horizontal	θ <sub>L</sub>	CR≥10 Ta=25±2°C ILED=40mA	-	65	-	Degree
		θ <sub>R</sub>		-	65	-	
	Vertical	θ <sub>T</sub>		-	65	-	
		θ <sub>B</sub>		-	50	-	
Brightness		C	-	300	400	-	cd/m <sup>2</sup>
		N		-	500	-	
Viewing Direction		6 o'clock					

#### 3.3. LED Information

To conserve the LED lifetime and reduce power consumption dim or turn off LEDs during periods of inactivity. Command 14: LED Control includes LED control for the display backlight, keypad LEDs, and indicator LEDs. An automatic dimming function is available using command 14. See the [CFA039A0 Command Reference](#) for more information on this command.

### **3.3.1.TFT Backlight LEDs**

The display backlight is controlled by the module and no additional connections from the host are required. The backlight is connected to  $V_{CC}$  and no additional power is required.

### **3.3.2.Status LEDs**

The four RGB status LEDs located to the left of the display are controlled by the module. No additional connections from the host are required. By default, the status LEDs are powered by the on-board voltage regulator with JP5 closed. To power the status LEDs with the input power, open JP5 and close JP4.

The module also supports the addition of up to four external Innolux N-PI32TATPRGPB.ZR LEDs (or similar LEDs which use the same communications interface) using the H3 connector on the rear of the module. Consult the N-PI32TATPRGPB.ZR datasheet for the single wire communication protocol.

### **3.3.3.Keypad LEDs**

The keypad LEDs are controlled by the module and do not require additional connections from the host. By default, the keypad LEDs are powered by  $V_{CC}$  with JP7 closed. JP7 should be closed for operating voltages above 3.3V. For operating voltages below 3.3V, open JP7 and close JP6 to power the keypad LEDs via the 3.3V regulator output.

## 4. Electrical Specifications

### 4.1. Absolute Maximum Ratings

Absolute Maximum Ratings	Symbol	Minimum	Maximum	Unit
Operating Temperature	$T_{OP}$	-20	70	°C
Storage Temperature	$T_{ST}$	-30	80	°C
Humidity Range (Non-condensing)	RH	-	90% (1)	-
Supply Voltage	$V_{CC}-V_{SS}$	-0.3	6.5	V
Logic Voltage	$V_{DDIO}-V_{SS}$	-0.3	7.0	V

*Note: 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 beyond those listed under DC Characteristics is not implied. Changes in temperature can result in changes in contrast.*

*(1)Temp ≤ 60°C. When the temp is > 60°C, the max rating is < 90%.*

### 4.2. Operating Conditions

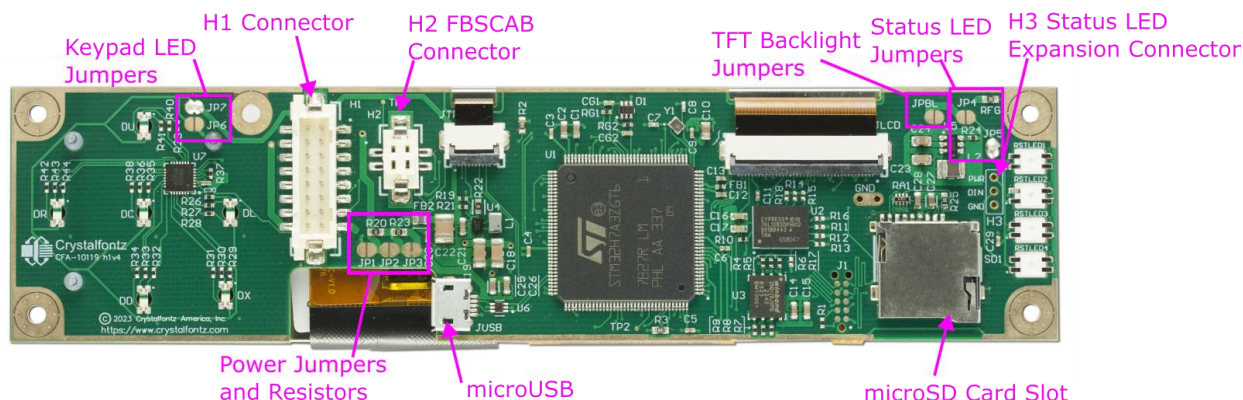
DC Characteristics	Symbol	Minimum	Maximum	Unit
Supply Voltage	$V_{CC}$	2.5	5.5	V
GPIO Input High Voltage	$V_{IH}$	2.3	5.5	V
GPIO Input Low Voltage	$V_{IL}$	-0.3	0.9	V
GPIO Output High Voltage	$V_{OH}$	2.9	-	
GPIO Output Low Voltage	$V_{OL}$	-	0.4	V

### 4.3. GPIO Current Limits

Typical GPIO Current Limits	Max	Unit
Output current sunk by any I/O	20	mA
Total output current sunk by sum of I/Os	130	mA
Total output current sourced by sum of I/Os	135	mA

## 5. Connection Information

### 5.1. Location of Connectors



### 5.2. H1 Connector Details

H1 is a 16-pin header. The part number is DF11Z-16DP-2V(27). A typical connecting header is the DF11-16DS-2C.

The H1 connector provides an interface for controlling or monitoring external devices with the CFA039A0-N-VDxT. Thirteen of the H1 pins may be configured as general-purpose inputs/outputs (GPIOs) or for specific control, communications, or ADC use (depending on the pin). The remaining three pins are for power-supply and external CFA039A0-N-VDxT reset control.

Pin functions are configured using a combination of commands 33 (0x21): Interface Options, 34 (0x22): GPIO Pin Configuration and 28 (0x1C): ATX Functionality. See the [CFA039A0 Command Reference](#) for more information on these commands.

All pins are 5V tolerant, but as the microcontroller used on the CFA039A0-N-VDxT is 3.3V, outputs are limited to 3.3V high-level. See [H1 GPIO Pins](#) for details.

The architecture of the CFA039A0-N-VDxT allows flexibility in the configuration of the GPIO pins. When pins are not used for a communication interface, they can be set as an input or output. When configured as a GPIO output, they can output constant high or low signals or a variable duty cycle 100 Hz PWM signal. When configured as a GPIO input, the CFA039A0-N-VDxT continuously polls the pins at 200 Hz. The host can query the previously polled level using command 34 (0x22): GPIO Pin Configuration.

Pull-up/pull-down resistance values are approximately 40kΩ. Typical GPIO current limits when sinking or sourcing all five GPIO pins simultaneously are 8 mA. See the ST-Micro STM32H7 datasheets for additional information. The GPIOs do not have under/over voltage or over current protection.

Communications interface settings override GPIO settings. If a communications interface using H1 pins is enabled, GPIO configuration of those same pins will be ignored. The Serial interface is enabled by default. To use the H1 connector pins H1.1 and H1.2 as GPIOs the Serial interface must first be disabled. See command 33 (0x21): Interface Options.

### 5.2.1.H1 Connector Pinout

H1 Pin Number	Default Function	GPIO Number	Communications Function
1	Serial RX	GPIO[7]	Serial RX
2	Serial TX	GPIO[8]	Serial TX
3	GPIO[9]	GPIO[9]	I2C SDA
4	GPIO[10]	GPIO[10]	I2C SCL
5	ADC 0	GPIO[5]	
6	ADC 1	GPIO[6]	
7	GPIO[11]	GPIO[11]	SPI CIPO
8	GPIO[12]	GPIO[12]	SPI COPI
9	ATX Power Control	GPIO[2]	SPI SS/CS
10	ATX Reset Control	GPIO[3]	SPI SCK
11	GPIO[0]	GPIO[0]	
12	ATX Power Sense	GPIO[1]	SPI INT (Data Ready)
13	GPIO[4]	GPIO[4]	I2C INT (Data Ready)
14	CFA039A0-N-VDxT Reset		
15	Power GND		
16	Power V <sub>CC</sub>		

### 5.2.2.H1 Pins 5 and 6 - ADC

H1 pins 5 and 6 are a special case as these are configured for ADC use. In default mode, these pins are configured as 0 to 3.3V analog-to-digital converter (ADC) inputs and are sampled continuously at 11kHz. When using command 34 (0x22): GPIO Pin Configuration, the sampled ADC values are averaged between the host reading the values. The averaged value is multiplied by 16 to increase value accuracy over long sample periods. The minimum and maximum ADC values are also tracked between the host reading values using this command.

The ADC has 12-bit resolution, and uses a 3.3V reference voltage (min=3.27V max=3.39V). To calculate the approximate (uncalibrated) voltage at the H1 ADC pins:

Average voltage = (returned-average-value) / 16 / 4096 \* 3.3

Minimum voltage = (returned-minimum-value) / 4096 \* 3.3

Maximum voltage = (returned-maximum-value) / 4096 \* 3.3

These two pins have an extra inline protection resistor, and power steering diodes. These pins can tolerate ±8V for a short amount of time. These pins also have a low-pass filter with a -3db roll-off at 27kHz.

## 5.3. H2 Connector

H2 is a 4-pin connector to connect the CFA039A0-N-VDxT module to a separately purchased CFA-FBSCAB using the WR-EXT-Y37.



#### 5.4. H3 Connector

H3 can connect up to four external Innolux N-PI32TATPRGPB.ZR LEDs (or LEDs which use the same communications interface).

Pin on H3 Header	Symbol	Description
1	PWR	This pin is tied to $V_{CC}$ . It is an output and external power should not be applied.
2	DOUT*	Data out. This pin is for single wire coms and should be connected to DIN on the peripheral LEDs.
3	GND	Ground.

\*Note: On hardware version h1v4 this pin is marked DIN on the silkscreen of the module.

#### 5.5. MicroSD Card Slot

The microSD card slot can expand the storage capabilities of the CFA039A0-N-VDxT. See [section 9.2: MicroSD Card](#) for more information on formatting the SD card for use.

#### 5.6. Jumper Configurations

The module has eight jumpers that configure the module as described below.

Jumper	Feature	Description
JP1	Power	<b>Open:</b> (default) No effect. <b>Closed:</b> Connects H1 pin 16 to 3.3V Rail.
JP2/ R20	Power	Open: H1 does not power the module via 3.3V regulator. <b>Closed:</b> (default, by R20) Connects H1 pin 16 to 3.3V regulator input. <i>Note: JP2 and R20 serve the same purpose. Closing either will result in a closed state.</i>
JP3/ R23	Power	Open: USB does not power the module. <b>Closed:</b> (default, by R23) USB +5V is connected to 3.3V regulator input. <i>Note: JP3 and R23 serve the same purpose. Closing either will result in a closed state.</i>
JP4	Status LED	<b>Open:</b> (default) Status LEDs not powered via $V_{CC}$ . <b>Closed:</b> Status LEDs are powered via $V_{CC}$ .
JP 5	Status LED	Open: Status LEDs not powered via 3.3V regulator output. <b>Closed:</b> (default) Status LEDs are powered via 3.3V regulator output (3.3V).
JP 6	Keypad LED	<b>Open:</b> (default) Keypad LEDs not powered via 3.3V regulator output. <b>Closed:</b> Keypad LEDs are powered via the 3.3V regulator output (3.3V).  Recommend to open JP7 and close JP6 if $V_{CC}$ is less than 3.3V.
JP 7	Keypad LED	Open: Keypad LEDs not powered via $V_{CC}$ . <b>Closed:</b> (default) Keypad LEDs are powered via $V_{CC}$ .
JPBL	TFT Backlight	<b>Open:</b> (default) No effect. <b>Closed:</b> TFT backlight always at full brightness when power is supplied to the module.

## 6. ATX Power Supply and Control Connections

ATX power supply control functionality allows the buttons on the CFA039A0-N-VDxT to act as the power and reset buttons on a system, simplifying front panel design.

### GPIO[1] ATX Host Power Sense

The CFA039A0-N-VDxT acts differently depending on the host's power state. Thus, the host's "switched +5v" must be connected to GPIO[1]. This GPIO line functions as POWER SENSE. The POWER SENSE pin is an input with a pull-down, 5k $\Omega$  nominal.

### GPIO[2] ATX Host Power Control

The motherboard's power switch input is connected to GPIO[2]. This GPIO line functions as POWER CONTROL. The POWER CONTROL pin is a high impedance input until the LCD module instructs the host to turn on or off. Then it will change momentarily to low impedance output, driving either low or high depending on the setting of POWER INVERT. See command [28 \(0x1C\): Set ATX Power Switch Functionality](#).

### GPIO[3] ATX Host Restart Control

The motherboard's reset switch input is connected to GPIO[3]. This GPIO line functions as RESTART. The RESTART pin is a high-impedance input until the LCD module wants to reset the host. Then it will change momentarily to low impedance output, driving either low or high depending on the setting of RESTART\_INVERT. See command [28 \(0x1C\): ATX Functionality](#). This connection is also used for the hardware watchdog.

ATX Power Supply & Control Connections	Pin on H1 Connector*
V <sub>SB</sub> (+5v)	Pin 16
Ground	Pin 15
GPIO[1] ATX Host Power Sense	Pin 12
GPIO[2] ATX Host Power Control	Pin 9
GPIO[3] ATX Host Reset Control	Pin 10
*For "Full Swing" RS232 using the optional CFA-RS232 Level Translator Board, the H1 pins are passed through to the CFA-RS232's J1 connector.	

**IMPORTANT:** The GPIO pins above must be configured to their default drive mode for the ATX functions to work. These settings are factory default but may be changed by the user. The GPIO pins used for ATX control must **not** be configured as user GPIO or for use as a communications interface. See commands 34 (0x22): Set or Set and Configure GPIO Pin and 33 (0x21): Interface Options in the [CFA039A0 Command Reference](#) for more information.

**NOTE:** The CFA039A0-N-VDxT cannot control ATX functionality via a connected CFA-FBSCAB's GPIO connector. ATX control must be performed via the CFA039A0-N-VDxT's H1 connector.

### ATX Connection to H1 Using WR-PWR-Y25/38 Cable

The illustration below shows a Crystalfontz [WR-PWR-Y25](#) or [WR-PWR-Y38](#) ATX cable connected to the CFA039A0-N-VDxT H1 connector and a system's host and ATX Power Supply:

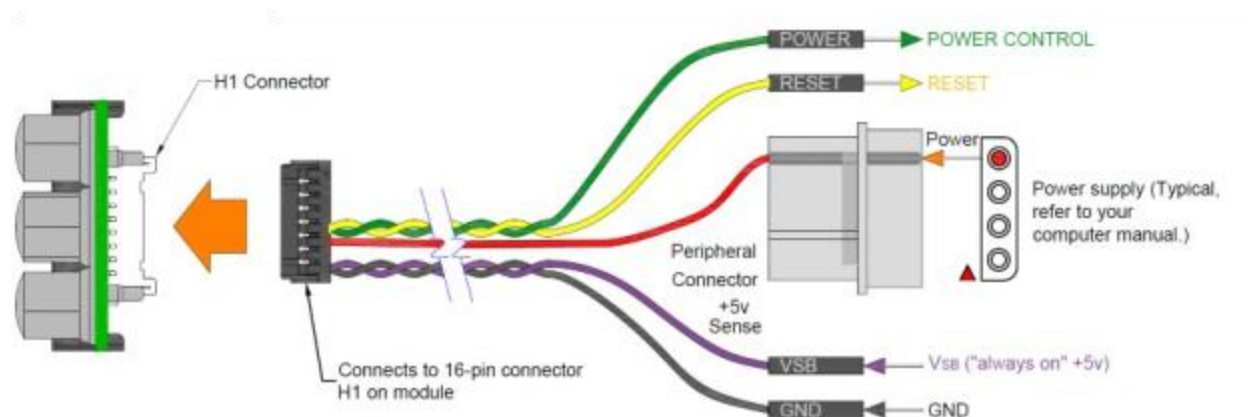


Figure 10. ATX Connection to H1 with WR-PWR-Y25 or WR-PWR-Y38 Cable

## 7. Firmware

### 7.1. How to Identify Firmware Revision Number

There are three ways to have the module identify its firmware revision:

1. Before applying power to the CFA039A0-N-VDxT, press the right arrow key on the keypad. Apply power, keeping the right arrow key depressed until the firmware revision displays.
2. During a restart, keep the right arrow key depressed until the firmware revision displays.
3. Use command 1 (0x01): Get Module Information.

### 7.2. Firmware Protections

#### 7.2.1. Watchdog

To recover from any possible main firmware faults or lock-ups the module has a "watchdog" feature enabled that will reset the module after 15 seconds of the firmware becoming unresponsive.

#### 7.2.2. Firmware Program Storage

The main firmware is supplied by Crystalfontz as an encrypted file and may be updated or downgraded at any time by the user using the module bootloader (see the instructions in section [7.6. Firmware Update \(From On-board Flash, microSD Card, or Host\)](#)).

Custom or third-party firmware can be developed and loaded onto the module. [Contact Crystalfontz](#) for further information.

### 7.3. Module Bootloader

The module bootloader software runs on power-on or reset of the module. The bootloader will always run the main firmware immediately unless the main firmware is not present or the bootloader is instructed to show its selection menu.

The module bootloader can perform the following functions:

- starts the main firmware (automatic)
- allows updating of the main firmware (see [section 7.4 Future Firmware Updates](#))
- performs some hardware test functions (see [section 7.3.2 Bootloader Hardware Testing](#))

The bootloader cannot be updated by the user or in the field. The module bootloader is programmed by Crystalfontz during production. If the module bootloader becomes compromised or inoperable the only way to recover the module is to return it to Crystalfontz for evaluation and repair.

#### 7.3.1. Entering the Bootloader

There are two methods available for entering the bootloader main menu:

1. Turn off the power supply to the module. While holding the UP and DOWN keys on the module keypad, turn on the power supply to the module.
2. Send command 5 (Restart into Bootloader Mode) to the module.

#### 7.3.2. Bootloader Hardware Testing

Testing of the on-board flash, graphics RAM IC, and touchscreen can be performed by selecting the appropriate option with the keypad.

**NOTE:** Performing on-board flash testing will erase the contents of the on-board flash. Module settings, and any saved user files will be removed.

## 7.4. Future Firmware Updates

CFA039A0-N-VDxT display modules include pre-installed firmware that performs the command functions described in the [CFA039A0 Command Reference](#). Crystalfontz may make updates to the firmware in the future. Firmware updates are announced via [PCN \(Part Change Notices\)](#).

Any firmware updates will be available as a free download in the “Files and Downloads” section on the [product’s webpage](#). Detailed instructions on how to update the firmware will be included in the field upgrade kit provided.

### 7.4.1. Firmware Update (From microSD or On-board Flash)

Once presented with the bootloader main menu, select the firmware loader by pressing the TICK button on the module keypad.

The bootloader will check for the presence of the firmware update file on the on-board flash filesystem, followed by the microSD card. The firmware file must be present in the root directory of the filesystem and be named “cfa039a0.blf”. If using a microSD card, format it as FAT12, FAT16 or FAT32.

When the bootloader finds the firmware update file, it will ask for confirmation and then proceed to update the main firmware.

### 7.4.2. Firmware Update (From Host)

Once presented with the bootloader main menu, select the firmware loader by pressing the TICK button on the module keypad.

The bootloader will check for the presence of the firmware update file on the on-board flash filesystem, followed by the microSD card. When the file cannot be found, it will wait for connection from the host via USB or serial connection.

## 7.5. Main Firmware Special Modes

### 7.5.1. Check Module Version

To check the module version information:

1. Turn off the power supply to the module
2. Hold the RIGHT key on the module keypad
3. Turn on the power supply to the module

The module will display version information until the TICK key is pressed. Then the module will reset itself and continue normal operation.

### 7.5.2. Module Settings Reset

A complete settings reset can be performed by:

1. Turn off the power supply to the module
2. Hold the RIGHT and CROSS keys on the module keypad
3. Turn on the power supply to the module

After the instructions on the display are followed, the module will reset itself and continue normal operation.

### 7.5.3. USB Mass Storage Access Mode

The module can start in a special USB mass-storage access mode ([USB Mass Storage Device Class](#)) that allows the host OS to directly access both the on-board flash and microSD filesystems.



Microsoft Windows has built-in support for accessing the FAT16 or FAT32 filesystem on the microSD card (if present). It does not, however, have the necessary drivers to access the LittleFS filesystem used on the on-board flash.

Popular Linux distributions also have built-in support for accessing the FAT16 or FAT32 filesystem on the microSD card (if present). The LittleFS filesystem used on the on-board flash may also be accessed by using the [littlefs-fuse wrapper available here](#).

**Warning:**

Accessing the on-board flash and microSD by this method can potentially corrupt or erase the contents of the filesystems. Access to the filesystems is quite slow so patience is required, and the filesystems must be unmounted / ejected by the host OS before the module is powered off or reset.

To start the USB mass-storage access mode:

1. Turn off the power supply to the module
2. Hold the LEFT and TICK keys on the module keypad
3. Plug the module into the host USB port

After the instructions on the display are followed, the module will reset itself and continue normal operation.

#### **7.5.4. USB Debugging Output**

The module can enable a second USB virtual-com port for debugging output.

To enable the debugging output virtual-com port:

1. Turn off the power supply to the module
2. Hold the LEFT and CROSS keys on the module keypad
3. Plug the module into the host USB port

The module will appear to start normally but will also create a second USB virtual-com port on the host. Data sent from the host to the module on the debugging virtual-com port is ignored by the module. Use command 62 to set the debug output options.

The debugging output USB virtual-com port may also be semi-permanently enabled by creating an empty file named "CDCDEBUG\_MODE" in the root directory of the on-board flash filesystem. If this file exists during firmware start, the debugging output USB virtual-com port will be enabled.

The debugging output is color encoded using ANSI escape codes, so a compatible serial console viewer must be used (e.g., PuTTY on Windows, or GTKTerm on Linux).



## 8. Communication Interfaces

The module supports USB, Serial, I2C, and SPI interfaces for command packet communications with one or more host devices. Each of the communication interfaces has its own input/output buffers and may be used simultaneously. While interface data is handled synchronously, internally command packets are handled asynchronously.

If module communication problems occur, use the debugging via Command 62: Debugging on a different interface. See the [CFA039A0 Command Reference](#) for additional information.

### 8.1. USB Interface

The module supports USB communications using a [“communication device class abstract control model”](#) (CDC ACM) virtual-serial interface through the microUSB connector.

Most modern common host operating systems include generic drivers for this USB interface type and will present a serial communications interface (virtual COM port) for host software use. A Microsoft certified Windows USB driver package is available from the [Crystalfontz website](#). Any virtual serial port settings that can be adjusted on the host such as baud rate, stop bits, etc. are ignored.

On Linux hosts, the virtual serial device file will be named “/dev/ttyACMx” (where x is the next available device number).

### 8.2. UART Serial Interface (Logic Level, Inverted)

The module supports one logic-level, inverted serial interface on the H1 connector, pins H1.1 (RX) and H1.2 (TX).

The serial interface is enabled by default with settings of 115200 baud, 8 data bits, no parity, 1 stop bit. If the Logic Level, Inverted (LLI) interface is enabled, GPIO use of the RX and TX pins will be unavailable. See command 33 in the [CFA039A0 Command Reference](#) for configuration options.

### 8.3. UART Serial Interface (Full-Swing RS232)

The module can be customized with a [CFA-RS232](#) interface board to provide a “full-swing” industry standard, ESD protected RS232 interface. When the CFA-RS232 board is fitted, the LLI serial interface becomes inaccessible as both interfaces use the same H1.1 and H1.2 pins.

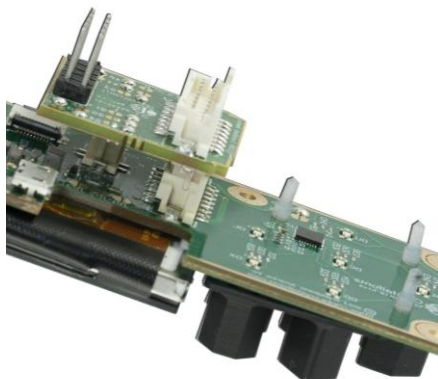


Figure 1. Angled View of CFA-RS232 Level Translator Mounted on CFA039A0-N-VDxT

The CFA-RS232 Level Translator has a 16-pin socket connector J3 that mates with the 16-pin connector H1 on the back of the module. The CFA-RS232 converts the 0v to +5v (logic level), Rx and Tx signals from the module's microcontroller to RS232 levels.

#### **8.4. I2C Target Interface**

The module supports one target I2C interface on the H1 connector, pins H1.3 (SDA), H1.4 (SCL) and H1.13 (INT).

When the module has packet data available to read by the host in its outgoing I2C buffer, the H1.13 data-ready/interrupt pin will be driven low. The H1.13 pin is open-drain so must be pulled-up externally. The host device should monitor the state of this pin and initiate an I2C data read sequence while it is low. The pin will return to a hi-Z state when the outgoing buffer is empty.

The I2C interface is not enabled by default. If the interface is enabled, GPIO use of the same pins will be unavailable. See command 33 in the [CFA039A0 Command Reference](#) for configuration options.

#### **8.5. SPI Peripheral Interface**

The module supports one peripheral SPI interface on the H1 connector, pins H1.7 (CIPO), H1.8 (COPI), H1.9 (CS/SS), H1.10 (SCK) and H1.12 (INT).

The CS (chip-select) H1.9 pin is active low and must be pulled low before the host initiates SPI communications. It can be tied low permanently if the SPI bus is not shared with other devices.

When the module has packet data available to read in its outgoing SPI buffer, the H1.12 data-ready/interrupt pin will be driven low. The H1.12 pin is open-drain so must be pulled-up externally. The host device should monitor the state of this pin and initiate a SPI data read sequence while it is low. The pin will return to a hi-Z state when the outgoing buffer is empty.

There is no effective flow-control when reading or writing on the SPI interface. In order to avoid dropping packets, waiting for packet responses is recommended. Data may be dropped or overflow the module's incoming data buffer if data is sent over the SPI interface without waiting for packet responses.

The SPI interface is not enabled by default. If the interface is enabled, GPIO use of the same pins will be unavailable including the use of ATX power control. See command 33 in the [CFA039A0 Command Reference](#) for configuration options.

## 9. Non-Volatile Data Storage

The module has two non-volatile data storage devices available: on-board flash and a microSD card socket.

All storage related commands supported by the module can use both the on-board flash and microSD card (if a microSD card is inserted and operational).

Both devices use a conventional MS-DOS/Windows style filename/directory structure, with the two storage devices using drive letters for identification. The on-board flash is “A” drive (for example “a:\logo\image.jpg”), and the microSD card is “B” drive (for example “b:\logo\image.jpg”).

If no drive letter is given in a filename/path (for example, “logo\image.jpg”), the on-board flash is used.

### 9.1. On-Board Flash

The on-board flash is a fast and reliable 128M-bit (16M-byte) NOR flash IC located on the module. Data is stored on the on-board flash using the robust [LittleFS filesystem](#).

The on-board flash stores module settings (as saved by command 4), user data (see command 2 and command 3), and user files saved by command 39, or the USB mass-storage access mode (see module start-up options).

The file system can be cleared / re-formatted by using command 39, subcommand 6.

The flash IC can be error-checked using the module bootloader (this will erase any data files present on the on-board flash). See [7.3.2. Bootloader Hardware Testing](#).

See Command 39 Storage Command Group in the [CFA039A0 Command Reference](#) for more details on on-board flash use.

### 9.2. MicroSD Card

The module supports the use of an optional microSD card for extended data storage.

The microSD card must be of SDHC type, and formatted to FAT12, FAT16, or preferably the FAT32 filesystem. Use a high-endurance / industrial type microSD card for mid/long term reliability.

If long-term continual file access is required (for example looping video playback on the display) use the on-board flash.

The file system can be cleared and re-formatted by using command 39, subcommand 6.

See Command 39 Storage Command Group in the [CFA039A0 Command Reference](#) for more details on microSD Card use.

## 10. Command Packet Communications

Communication between the module and the host takes place in the form of simple and robust CRC checked packets. The packet format allows for reliable communications between the module and the host without the 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.

For more information on the command packets, see the [CFA039A0 Command Reference](#).

### 10.1. Handshaking / Flow Control

The module's packet structure makes traditional hardware or software handshaking unnecessary. Reconcile packets rather than using delays when communicating with the module. To reconcile packets, ensure that the acknowledgement packet has been received from the most recently sent packet before sending any additional packets to the module. This practice avoids dropped packets or missed communication between the host and module.

The module 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 guarantee that it can process and remove packets from its input buffer faster than the packets can arrive given the data rate and the reporting configuration of the module.

Report packets are sent asynchronously with respect to command packets received from the host. The host should not assume 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.

### 10.2. Packet Types

The host uses **host-to-module command packets** (“command packets”) to request the module perform the specified command function.

If the module executes the command successfully, it will reply with a **command success response** packet. If the command packet contains an invalid request, or the module has a problem carrying out the command, the module will reply with a **command error response**.

The module uses **module-to-host report packets** when a specific event occurs at the module (for example, a key is pressed, the capacitive touchscreen is touched).

### 10.3. Debugging Communications

If module communication problems occur, use the debugging output enabled using Command 62: Debugging on a different interface.

Enable the debug-low or debug-high level setting. If data is dropped from an incoming data buffer due to corruption, too high data-rate, or other problems a debugging output message will be sent.

## 11. Graphics

The module features a 480x128 pixel, 16-bit color TFT display with wide-ranging, customizable, and easy to use graphical abilities. A 5-point capacitive touchscreen (CFA039A0-N-VDCT only) allows for configurable inputs from a user. The powerful STM32H7 microcontroller on the module, performs the heavy lifting associated with composing graphics and updating the display.

### 11.1. Graphical Objects

The graphics system is based on graphical objects, each having a user-defined unique identification number ("ID number") ranging from a value of 1 to 255. Graphical objects may not share a ID number. The graphical object ID number is set by the user on object creation and is used to refer to the object until its removal. If a graphical object is removed the ID number becomes unused and may then be re-used for a newly created graphics object.

The user lays out the intended display design using graphical objects, which may be individually created, removed, moved, and configured at any time by using command 40: Graphics Core Command Group. See the [CFA039A0 Command Reference](#) for more information.

### 11.2. Graphical Object Types

Below is a list of the available graphical object types and the key features of each.

- Text
  - True type font support using industry standard TTF files
  - Sub-pixel rendering, with anti-aliasing
  - ASCII/UTF8 and UTF16 character set support
  - Includes one built-in TTF font ([JetBrains Mono NL](#))
  - Load up to 15 custom fonts from the on-board flash or microSD card
- Image
  - Images can operate as touch operated buttons with three states: up, down, and disabled
  - JPG, BMP and PNG images from the on-board flash, or microSD card
  - Hardware JPG decoding, for fast loading/display of JPG images
  - Support for PNG images with an alpha/opacity layer
  - Number of images loaded is only limited by available graphics memory
- Video
  - Play, pause, and skip through videos, controlled using Command 41
  - Supports motion-jpeg AVI files loaded from the on-board flash or microSD card
  - Supports framerates up to 30fps
  - Plays multiple videos simultaneously
- Button
  - Simple touch-operated push or toggle button control
  - Supports three states - up, down, and disabled
  - Different style and label text settings for each state
  - Configurable report packets for simple host alerts/signaling
- Slider
  - Touch-operated draggable slider control for numerical value adjustment
  - Optional text label and value display on left or right of the control
  - Supports two states - enabled and disabled
  - Configurable report packets for simple host alerts/signaling
- Number Edit
  - Touch-operated numerical value adjustment using up and down buttons
  - Optional text label and value display
  - Supports two states - enabled and disabled
  - Configurable report packets for simple host alerts/signaling

- **Checkbox**
  - Touch-operated checkbox control
  - Optional text label
  - Supports four states - unchecked, checked, crossed, and disabled
  - Configurable report packets for simple host alerts/signaling
- **Progress Bar**
  - Graphical progress bar for indicating the progressor value of a task or operation
  - Configurable minimum/maximum values and current position
  - Optional text label and value display
  - Supports two states - enabled and disabled
- **Sketch Surface**
  - A general-purpose graphics object for pixel level art
  - Draw pixels, lines, rectangles, circles onto the surface

### 11.3. Graphical Object Styling

To reduce the amount of styling options (font, colors, etc.) that must be passed to the module in an object creation command packet, the module uses a default style. When a new graphics object is created it adopts the default style options. After creation, the graphics objects style options are unique to the object and can be modified at any time without affecting other graphical objects.

Properties of the default style may be changed by using command 40 subcommands, with the object ID number of 0 (zero). Changes to the default style will persist if the module's settings are saved by command 4. See the [CFA039A0 Command Reference](#) for more information on these commands

The modules “factory default” style settings are as follows:

Style Parameter	Value
Font Slot	0
Font Size	28
Object Opacity	0xFF (100%)
Corner Radius	10

Style Parameter	Color Name	HTML Code (Hex RRGGBB)	RGB565 Code
Fill A Color	Mid Blue / Purple	#380093	0x3812
Border A Color	Light Blue	#6617E6	0x60DC
Control A Color	Pale Blue	#BA17E6	0xB8DC
Text A Color	Pale Yellow	#FFFD95	0xFFFF2
Fill B Color	Dark Blue / Purple	#210057	0x200B
Border B Color	Pale Blue	#6617E6	0x60DC
Control B Color	Dark Blue / Grey	#5F4982	0x6250
Text B Color	Pale Green	#85FF96	0x87F2
Fill C Color	28% Grey	#484848	0x4A49
Border C Color	50% Grey	#808080	0x8410
Control C Color	42% Grey	#6D6D6D	0x6B6D
Text C Color	50% Grey	#808080	0x8410



#### 11.4. Display Composition

Graphical objects are layered on the display in z-index order which is specified on object creation. A graphics object with a higher z-index will cover an element with a lower z-index.

All graphical objects also have an opacity value which sets the transparency of the object. A value of 0 (zero) is completely transparent, through to a value of 255 which is completely opaque (fully visible).

The image displayed on the modules display is composed in this order:

1. The framebuffer is cleared/filled using the color set by command 40, subcommand 2
2. Graphical objects are drawn onto the framebuffer in the order of the z-index attribute
3. The legacy CFA635/CFA735/CFA835 text layer is overlaid on the framebuffer
4. Module status information (for example, ATX system status) is overlaid on the framebuffer
5. The framebuffer is sent to the display

#### 11.5. Display and Object Coordinates

The module's display is 480x128 pixels in size (480 pixels wide, 128 pixels high). The 0,0 (X, Y) coordinate is at the top-left of the display and 479,127 at the bottom-right.

The graphical object X,Y position specifies the top-left of the object, except for text objects where the Y position is the base-line of the text. Graphical objects X and Y positions are a signed 16-bit integer, they may range from -10,000 to +10,000.

If a graphical object lies outside of the visible 0,0 to 479,127 pixel region, it will be clipped, or not visible. For example, if an image object is 120x120 pixels in size, and has an X,Y location of -40,90 only the top-right part of the image will be seen on the display. This feature allows for graphical objects to be scrolled across the display. Moving a graphical object out of the visible display area is a valid method of temporarily hiding an object (another option is setting its opacity to 0).

#### 11.6. Graphics Memory

The module has 16M-bytes of volatile memory (RAM) allocated for graphics object use. This is separate from the 16M-bytes of on-board flash which is used for non-volatile settings and user-data storage.

When a graphics object is created its settings, data, and graphics buffer use some of the available graphics memory. How much it uses depends on the type of graphics object and its dimensions. See command 42: Image Object Command Group in the [CFA039A0 Command Reference](#) for details.

If the graphics memory becomes too full to allow the creation of a new object, an error packet will be sent to the host (see [command error codes](#) for details).

The amount of graphics memory remaining can be read by using command 39, subcommand 7 or by monitoring the debug log using command 62.

The module state is volatile (is lost on power-off / reboot), unless saved by command 4.

#### 11.7. Touchscreen Features

The module (CFA039A0-N-VDCT only) features a 5-point capacitive touchscreen for interaction with graphics object controls or for general purpose use by the host system. The touchscreen uses the same coordinate system as the display, with point 0,0 (X,Y) at the top-left and point 479,127 at the bottom-right of the touchscreen.

Global touchscreen events can be reported to the host system by enabling touch reporting with command 25. Localized touch events to graphics objects can be enabled using the graphics object creation command, or enabled at a later time using command 40, subcommand 12.

## 12. Command Codes

For convenience, command codes are grouped by type in the following list. Refer to the separately provided [CFA039A0 Command Reference](#) for additional details on the command set. Experiment with these commands using the free download of [cfTest](#).

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 are the same as the low 6-bits of the type field of the command packet being acknowledged.

General / Communications
Command 0 (0x0): Ping
Command 1 (0x01): Get Module Information
Command 4 (0x04): Store Current State as Boot State
Command 5 (0x05): Module Reset / Restart
Command 28 (0x1C): ATX Functionality
Command 29 (0x1D): Watchdog
Command 33 (0x21): Interface Options
Command 62 (0x3E): Debugging

Display / LCD
Command 6 (0x06): Clear Display
Command 9 (0x09): Legacy Special Character Bitmaps
Command 11 (0x0B): Legacy Cursor Position
Command 12 (0x0C): Legacy Cursor Style
Command 13 (0x0D): Display Contrast
Command 14 (0x0E): LED Control
Command 31 (0x1F): Legacy Write Text to the Display
Command 32 (0x20): Legacy Read Text from the Display
Command 40 (0x28): Display Graphic Options includes:
Command 41 (0x29): Video Playback Control includes:
Command 42 (0x2A): Image Object Command Group
Command 43 (0x2B): True Type Fonts Command Group
Command 44 (0x2C): Sketch Surface Command Group
Command 47 (0x2F): Button Command Group
Command 48 (0x30) Slider Control Command Group

Command 49 (0x31) Number-Edit Control Command Group
Command 50 (0x32) Checkbox Control Command Group
Command 51 (0x33) Progress Bar Command Group

<b>GPIOs and Keypad</b>
Command 14 (0x0E): LED Control
Command 23 (0x17): Keypad Reporting
Command 24 (0x18): Keypad Poll State
Command 25 (0x19): Touchscreen Reporting
Command 28 (0x1C): ATX Functionality
Command 34 (0x22): GPIO Pin Levels
Command 37(0x25) Subcommand 5: CFA-FBSCAB GPIO Pin Levels

<b>Fan and Temperature Control / Monitoring</b>
Command 37 (0x25): CFA-FBSCAB Command Group

<b>Micro-SD Operations</b>
Command 39 (0x27): Storage Command Group
Command 41 (0x29), Subcommand 0: Load A Video from MicroSD Card

<b>On-Board Flash Operations</b>
Command 2 (0x02): Write User Flash Area
Command 3 (0x03): Read User Flash Area
Command 4 (0x04): Store Current State as Boot State

## 13. Backwards Compatibility

The CFA039A0-N-VDxT has been designed with backwards compatibility in mind, but due to changes in hardware and new features some CFA635/CFA735/CFA835 features and commands have changed or are no longer supported.

In the following table, if a command exists in the CFA039A0-N-VDxT, the “CFA039A0-N-VDxT Backwards Compatible” column will have an entry. If a column has a “-”, that command is not present for that module.

### 13.1. Functional Differences

Command Number	Command Name	CFA635 Command Present	CFA735 Command Present	CFA835 Command Present	CFA039A0-N-VDxT Backwards Compatible	Notes
0	Ping	Yes	Yes	Yes	Yes	
1	Module Information	Yes	Yes	Yes	Yes	
2	Write User Flash Area	Yes	Yes	Yes	Yes	
3	Read User Flash Area	Yes	Yes	Yes	Yes	
4	Store Current State as Boot State	Yes	Yes	Yes	Yes	
5	Module Reset/Restart	Yes	Yes	Yes	Yes	
6	Clear Display	Yes	Yes	Yes	Yes	
9	Special Character Bitmaps	Yes	Yes	Yes	Yes	
10	Read LCD Memory	Yes	Yes	-	-	Partially supported by the “Read Text from the Display” command
11	Cursor Position	Yes	Yes	Yes	Yes	
12	Cursor Style	Yes	Yes	Yes	Yes	
13	Display Contrast	Yes	Yes	Yes	Partial	Module responds to the command, but the setting has no effect.
14	LED Control	Yes	Yes	Yes	Partial	The one data-byte length command does not modify keypad brightness.

Command Number	Command Name	CFA635 Command Present	CFA735 Command Present	CFA835 Command Present	CFA039A0-N-VDxT Backwards Compatible	Notes
16	Fan Reporting	Yes	Yes	-	-	This is now a subcommand in the "CFA-FBSCAB Command Group"
17	Fan Power	Yes	Yes	-	-	This is now a subcommand in the "CFA-FBSCAB Command Group"
18	DOW Information	Yes	Yes	-	-	This is now a subcommand in the "CFA-FBSCAB Command Group"
19	Temperature Reporting	Yes	Yes	-	-	This is now a subcommand in the "CFA-FBSCAB Command Group"
20	DOW Transaction	Yes	-	-	-	This is now a subcommand in the "CFA-FBSCAB Command Group"
22	LCD Direct Command	Yes	-	-	-	
23	Keypad Reporting	Yes	Yes	Yes	Yes	
24	Keypad Poll State	Yes	Yes	Yes	Yes	
25	Set Fan Power-Safe	Yes	Yes	-	-	This is now a subcommand in the "CFA-FBSCAB Command Group"
25	Touchscreen Reporting	-	-	-	(New Command)	This command replaces the old "Set Fan Power-Safe" command number
26	Set Fan Glitch	Yes	Yes	-	-	This is now a subcommand in the "CFA-FBSCAB Command Group"
27	Query Fan Power	Yes	Yes	-	-	This is now a subcommand in the "CFA-FBSCAB Command Group"
28	ATX Functionality	Yes	Yes	Yes	Yes	

Command Number	Command Name	CFA635 Command Present	CFA735 Command Present	CFA835 Command Present	CFA039A0-N-VDxT Backwards Compatible	Notes
29	Watchdog	-	Yes	Yes	Yes	
31	Write Text to the Display	Yes	Yes	Yes	Yes	Custom font options from the CFA835 are not backwards compatible
32	Read Text from the Display	-	-	Yes	Yes	
33	Interface Options	Yes	Yes	Yes	Yes	
34	GPIO Pin Configuration	Yes	Yes	Yes	Yes	
35	Read GPIO Pin Configuration	Yes	Yes	-	-	
36	Interface Bridge	-	-	Yes	-	(may be supported in a future firmware version)
37	CFA-FBSCAB Command Group	-	-	Yes	Yes	
38	Custom Fonts	-	-	Yes	-	Replaced by the "True-Type Fonts Command Group"
39	Storage Command Group	-	-	Yes	Partial	Subcommand 0 and 4 – With no storage device specified in the filename, CFA039A0 will default to the on-board flash. The CFA835 will only read from the microSD card. Subcommands 1 to 3 – Fully backwards-compatible with the CFA835.
40	Graphics Core Command Group	-	-	Yes	Partial	Subcommand 0 – bit0 is backwards compatible, bit1 has no effect. Subcommand 1 –

Command Number	Command Name	CFA635 Command Present	CFA735 Command Present	CFA835 Command Present	CFA039A0-N-VDxT Backwards Compatible	Notes
						Fully backwards-compatible with the CFA835. All other subcommands are NOT backwards compatible or are new for the CFA039A0.
41	Video Object Command Group	-	-	Yes	(New Command)	This CFA039A0 command has the same command number as the "Video Playback Control Command Group" as the CFA835, but the subcommands and their options are new and not backwards-compatible.
42	Image Object Command Group	-	-	-	(New Command)	
43	True-Type Fonts Command Group	-	-	-	(New Command)	
44	Sketch Surface Command Group	-	-	-	(New Command)	
47	Button Command Group	-	-	-	(New Command)	
48	Slider Control Command Group	-	-	-	(New Command)	
49	Number-Edit Control Command Group	-	-	-	(New Command)	
50	Checkbox Control	-	-	-	(New Command)	



Command Number	Command Name	CFA635 Command Present	CFA735 Command Present	CFA835 Command Present	CFA039A0-N-VDxT Backwards Compatible	Notes
	Command Group					
51	Progress Bar Command Group	-	-	-	(New Command)	
62	Debugging				(New Command)	

### 13.2. Functional Differences

- **ATX Power Control**
  - The CFA039A0-N-VDxT no longer resets itself after a “Keypad Restart”, “Keypad Power On”, “Keypad Power-Off” or resuming from a “Lighting Mimics Host Power” off state. Instead, it loads the saved boot state.
- **Startup Interface Configuration**
  - The CFA039A0-N-VDxT does not currently support a power-on interface configuration screen, as is present on the CFA835 series modules.
- **Interface Bridge**
  - The CFA039A0-N-VDxT does not currently support the interface bridging feature as present on the CFA835 series modules.
- **CFA-FBSCAB Live Display**
  - The CFA039A0-N-VDxT does not currently support the CFA-FBSCAB live display feature as present on the CFA835 series modules.

[Contact us](#) if you require these features.

### 13.3. Legacy Character Generator ROM (CGROM)

The following character map is included for backwards compatibility with the CFAx35 modules when using command 9, command 11, command 12, command 31, and command 32. To find the code for a given character, add the two numbers that are shown in bold for its row and column. For instance, to display a superscript 9, add together the decimal column and row headers – 128<sub>d</sub> and 9<sub>10</sub> to get 137<sub>10</sub> or combine the upper and lower 4 bits (1000 and 1001 become 1000 1001).

upper 4 bits lower 4 bits	0 <sub>d</sub> 0000 <sub>2</sub>	16 <sub>d</sub> 0001 <sub>2</sub>	32 <sub>d</sub> 0010 <sub>2</sub>	48 <sub>d</sub> 0011 <sub>2</sub>	64 <sub>d</sub> 0100 <sub>2</sub>	80 <sub>d</sub> 0101 <sub>2</sub>	96 <sub>d</sub> 0110 <sub>2</sub>	112 <sub>d</sub> 0111 <sub>2</sub>	128 <sub>d</sub> 1000 <sub>2</sub>	144 <sub>d</sub> 1001 <sub>2</sub>	160 <sub>d</sub> 1010 <sub>2</sub>	176 <sub>d</sub> 1011 <sub>2</sub>	192 <sub>d</sub> 1100 <sub>2</sub>	208 <sub>d</sub> 1101 <sub>2</sub>	224 <sub>d</sub> 1110 <sub>2</sub>	240 <sub>d</sub> 1111 <sub>2</sub>
0 <sub>d</sub> 0000 <sub>2</sub>	CGRAM [0]															
1 <sub>d</sub> 0001 <sub>2</sub>	CGRAM [1]															
2 <sub>d</sub> 0010 <sub>2</sub>	CGRAM [2]															
3 <sub>d</sub> 0011 <sub>2</sub>	CGRAM [3]															
4 <sub>d</sub> 0100 <sub>2</sub>	CGRAM [4]															
5 <sub>d</sub> 0101 <sub>2</sub>	CGRAM [5]															
6 <sub>d</sub> 0110 <sub>2</sub>	CGRAM [6]															
7 <sub>d</sub> 0111 <sub>2</sub>	CGRAM [7]															
8 <sub>d</sub> 1000 <sub>2</sub>	CGRAM [0]															
9 <sub>d</sub> 1001 <sub>2</sub>	CGRAM [1]															
10 <sub>d</sub> 1010 <sub>2</sub>	CGRAM [2]															
11 <sub>d</sub> 1011 <sub>2</sub>	CGRAM [3]															
12 <sub>d</sub> 1100 <sub>2</sub>	CGRAM [4]															
13 <sub>d</sub> 1101 <sub>2</sub>	CGRAM [5]															
14 <sub>d</sub> 1110 <sub>2</sub>	CGRAM [6]															
15 <sub>d</sub> 1111 <sub>2</sub>	CGRAM [7]															

Figure 12. Character Generator (CGROM)

## 14. Module Reliability and Longevity

We work to continuously improve our products, including backlights that are brighter and last longer.

ITEM	SPECIFICATION
TFT Backlight	50,000 hours (typical)
Keypad	1,000,000 keystrokes
RGB LEDs	30,000 hours

**NOTE:** LED lifetime is defined as hours until max brightness has dimmed to 50% the original brightness. Dim or turn off LEDs during periods of inactivity to conserve the lifetime

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

Crystalfontz is committed to making all of our modules available for as long as possible. For each module that 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 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 do our best to find an acceptable replacement module with the same fit, form, and function.

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

Although the replacement module is within the stated specifications and tolerances of the discontinued module, changes may require modifications 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 changes in your code.
- 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 PCN on the product's website page as soon as possible. If interested, subscribe to future [Part Change Notices](#).

## 15. Care and Handling Precautions

For optimal operation of the CFA039A0-N-VDxT and to prolong its life, follow the precautions below.

### 15.1. ESD (Electrostatic Discharge)

Use industry standard antistatic precautions as for any other static sensitive devices e.g., expansion cards, motherboards, or integrated circuits. Ground personnel, work surfaces, and equipment.

### 15.2. Design and Mounting

- The exposed surface of the CFA039A0-N-VDNT “glass” is actually a polarizer laminated on top of the glass. When handling the module, avoid touching the polarizer. Finger oils are difficult to remove. To protect the soft plastic polarizer from damage, place a transparent plate (e.g., acrylic, polycarbonate or glass), in front of the module, leaving a small gap between the plate and the display surface.
- To protect the display from damage, the module ships with a protective film over the polarizer or capacitive touchscreen. Peel off the protective film slowly. Peeling off the protective film abruptly may generate static electricity.
- Do not disassemble or modify the module.
- Do not reverse polarity to the power supply connections. This will immediately ruin the module.

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

- Do not expose the CFA039A0-N-VDxT to strong mechanical shock, impact, torque, or tension.
- Do not drop, toss, bend, or twist the CFA039A0-N-VDxT.
- Do not place weight or pressure on the CFA039A0-N-VDxT.

### 15.4. If LCD Panel Breaks

- If the LCD panel breaks, do 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.

### 15.5. Cleaning

- The polarizer (laminated to the glass), is soft plastic that can easily be scratched or damaged, so use extra care cleaning.
- Do not clean the polarizer with liquids. Do not wipe the polarizer with any type of cloth or swab.
- Use the removable protective film or standard transparent office tape (for example, Scotch® brand “Crystal Clear Tape”) to remove smudges and any foreign matter.
- 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.

### 15.6. Operation

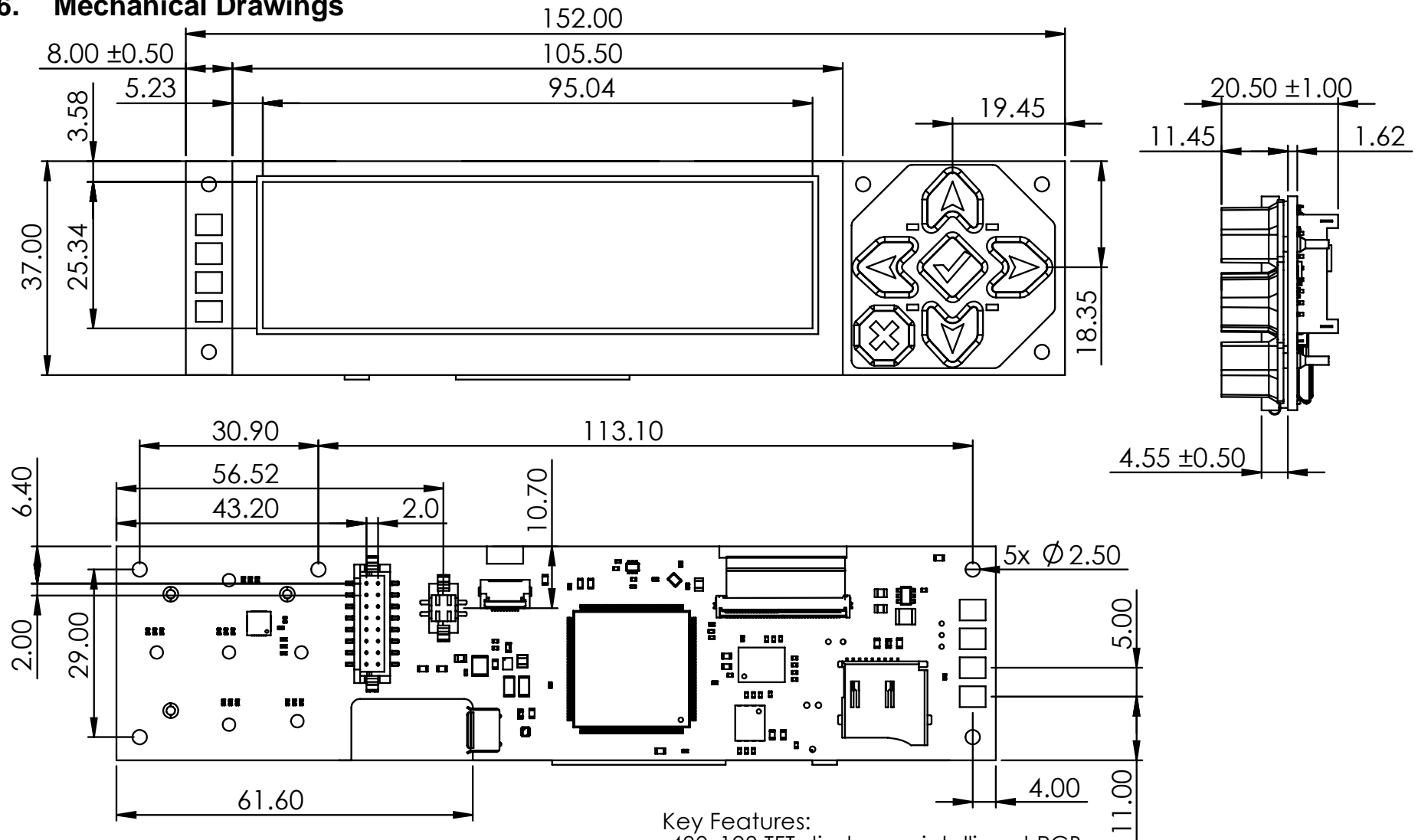
- Protect the CFA039A0-N-VDxT from ESD and power supply transients.
- Observe the operating temperature limitations: a minimum of -20°C to a maximum of +70°C with minimal fluctuation. Operation outside of these limits may shorten life and/or harm display.
- At lower temperatures of this range, response time is delayed.
- Operate away from dust, moisture, and direct sunlight.
- Dim or turn off the backlight during periods of inactivity to conserve the backlight lifetime.

### 15.7. 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 can cause moisture, resulting in permanent damage.
- Do not allow weight to be placed on the CFA039A0-N-VDxT while in storage.
- Recycle outdated Crystalfontz modules at an approved facility.



## 16. Mechanical Drawings



### Key Features:

- 480x128 TFT display on intelligent PCB
- Individually controllable keypad and status RGB LEDs
- Fully decoded keypad
- Show attached storage devices as drives
- µSD slot
- 128Mb onboard flash
- Built in graphical features such as sliders and buttons

Units: millimeters  
Tolerance: ±0.3mm



**CrystalFontz**  
©2025 CrystalFontz America, Inc.

Part Number:

**CFA039A0-N-VDNT**

Date:

5/6/2025

Filename:

CFA039A0-N-VDNT.pdf

Revision:

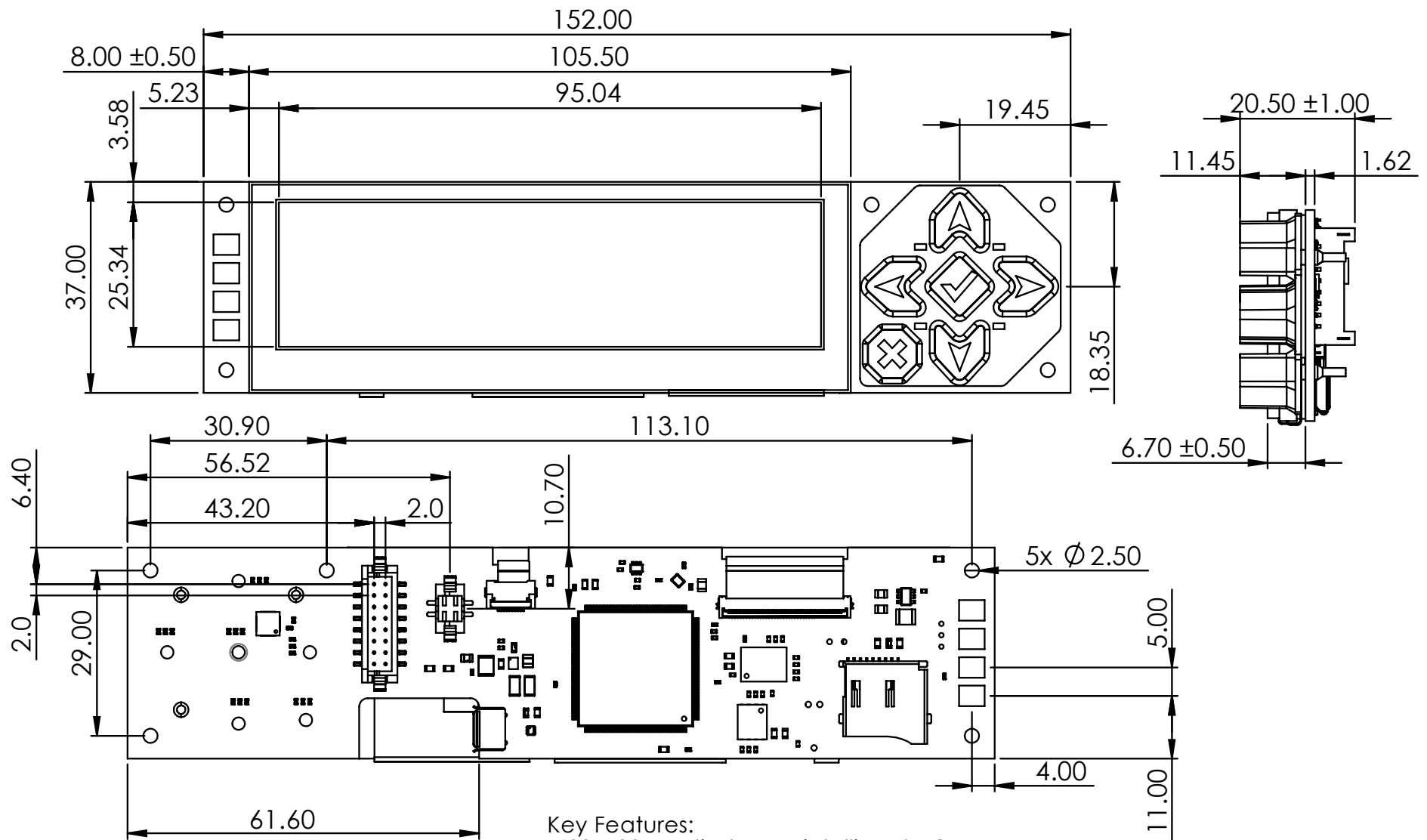
0v1

Web:

[www.crystallfontz.com/product/CFA039A0-N-VDNT](http://www.crystallfontz.com/product/CFA039A0-N-VDNT)

Sheet:

1 of 1



#### Key Features:

- 480x128 TFT display on intelligent PCB
- Individually controllable keypad and status RGB LEDs
- Fully decoded keypad
- Show attached storage devices as drives
- µSD slot
- 128Mb onboard flash
- Built in graphical features such as sliders and buttons

Units: millimeters  
Tolerance: ±0.3mm



**Crystalfontz**  
©2025 Crystalfontz America, Inc.

Part Number:

**CFA039A0-N-VDCT**

Date:

5/6/2025

Filename:

CFA039A0-N-VDCT.pdf

Revision:

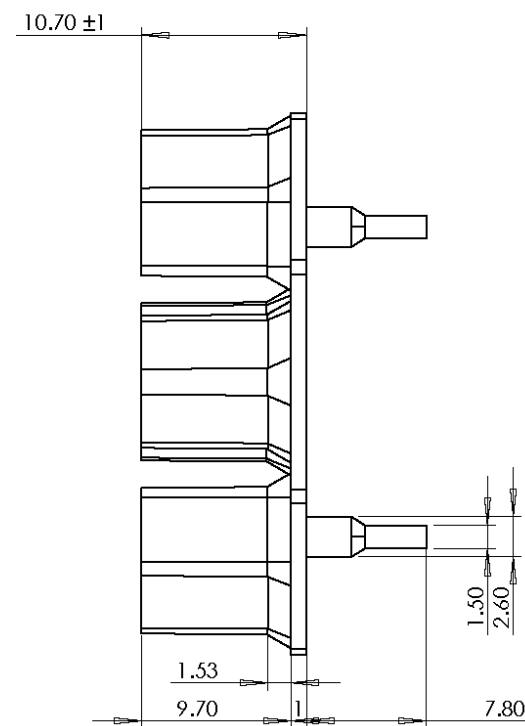
0v1

Web:

[www.crystalfontz.com/product/CFA039A0-N-VDCT](http://www.crystalfontz.com/product/CFA039A0-N-VDCT)

Sheet:

1 of 1



1 of 1



## Appendix A: Demonstration Software and Sample Code

### a. Crystalfontz cfTest

[cfTest](#) for Windows is a testing and configuration software that works on all Crystalfontz Intelligent LCD modules. This software allows exploration of the command set for all Crystalfontz Smart LCDs.

### b. Linux CLI Examples

[CLI Example Software](#) is a Linux compatible command-line demonstration program with C source code.

**NOTE:** It will show as /dev/ttyACMx instead of /dev/ttyUSBx.

[LCDproc](#) is an open-source project that supports many of the Crystalfontz displays. The CFA635 configuration should work with the CFA039A0-N-VDxT.

### c. Sample Code for RPM Calculation Information

This C function decodes the fan speed from a Fan Speed Report packet into RPM (fan tach speed):

```
bool HandleFanRPMReplyPacket (COMMAND_PACKET *packet, char *output)
{
    uint8_t fbscab_index;
    uint8_t fan_index;
    uint8_t cycles;
    uint8_t data_offset;
    uint8_t timer_lsb;
    uint8_t timer_msb;
    uint8_t pulses_per_revolution;
    uint16_t timer_ticks;
    uint8_t output_offset;
    float fan_rpm;

    /*
    fan rpm query command response packet has the format of:
    type = 0x40 | 0x25 = 0x65 = 101
    data length = 14
    data[0] = 2 (read fan tachometer speed)
    data[1] = FBSCAB module index
    data[2] = fan 1 number of fan tach cycles
    data[3] = fan 1 LSB of fan timer ticks
    data[4] = fan 1 MSB of fan timer ticks
    data[5] = fan 2 number of fan tach cycles
    data[6] = fan 2 LSB of fan timer ticks
    data[7] = fan 2 MSB of fan timer ticks
    data[8] = fan 3 number of fan tach cycles
    data[9] = fan 3 LSB of fan timer ticks
    data[10] = fan 3 MSB of fan timer ticks
    data[11] = fan 4 number of fan tach cycles
    data[12] = fan 4 LSB of fan timer ticks
    data[13] = fan 4 MSB of fan timer ticks
    */

    //check packet length
    if (packet->length != 14)
    {
        //unexpected packet length, should be 14 bytes
        return false;
    }
    //check the packets command number and type
    // 0x25 | 0x40 = FBSCAB Command Group | Reply Packet
    if (packet->command != (0x25 | 0x40))
    {
        //wrong packet command/type
        return false;
    }
    //check the packets subcommand type
    // 2 = Read fan tachometer speed
    if (packet->data[0] != 2)
```



```

    {
        //wrong packet subcommand value
        return false;
    }
    //get fbscab index from the packet
    fbscab_index = packet->data[1];

    //prepare output string
    output_offset = 0;
    output_offset += sprintf(&output[output_offset], "FBSCAB:%d - ",
fbscab_index);

    //process packet data for the 4 fans
    for (fan_index = 0; fan_index < 4; fan_index++)
    {
        //data offset for fan_index data in the packet
        data_offset = 2 + (fan_index * 3);
        //prepare output string
        output_offset += sprintf(&output[output_offset], "FAN%d: ",
fan_index);
        //get the fan data from the packet
        cycles = packet->data[data_offset];
        timer_lsb = packet->data[data_offset+1];
        timer_msb = packet->data[data_offset+2];
        timer_ticks = timer_lsb | (timer_msb << 8);
        //check fan cycles value
        if (cycles < 3)
        {
            //fan has stopped
            output_offset += sprintf(&output[output_offset], "STOPPED ");
            //next fan
            continue;
        }
        if (cycles < 4)
        {
            //fan is turning too slow to count RPM
            output_offset += sprintf(&output[output_offset], "SLOW ");
            //next fan
            continue;
        }
        if (cycles == 0xFF)
        {
            //unknown value
            output_offset += sprintf(&output[output_offset], "UNKNOWN ");
            //next fan
            continue;
        }

        //if we get to here, we have valid fan tach data
        //calculate fan RPM
        pulses_per_revolution = 2; //specific to each fan, most commonly 2
        fan_rpm = ((27692308L / pulses_per_revolution) * (cycles - 3)) /
(float)timer_ticks;

        //add RPM to output string
        output_offset += sprintf(&output[output_offset], "%5.2f ", fan_rpm);
        //done, next fan
    }
    //all done
    return true;
}

```

#### d. Sample Code for Temperature Sensor Report

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

```

bool HandleTempReplyPacket (COMMAND_PACKET *packet, char *output)
{
    uint8_t fbscab_index;

```



```
uint8_t sensor_index;
uint8_t temp_lsb;
uint8_t temp_msb;
uint16_t temp_raw;
uint8_t crc_status;
float deg_c;
float deg_f;

/*
temperature query command response packet has the format of:
type = 0x40 | 0x25 = 0x65 = 101
data_length = 5
data[0] = 4 (read WR-DOW-Y17 temperature)
data[1] = FBSCAB module index
data[2] = DOW device index (0-15)
data[3] = LSB of temperature data
data[4] = MSB of temperature data
*/

//check the packets command number and type
// 0x25 | 0x40 = FBSCAB Command Group | Reply Packet
if (packet->command != (0x25 | 0x40))
{
    //wrong packet command/type
    return false;
}
//check the packets subcommand type
// 4 = Read WR-DOW-Y17 temperature
if (packet->data[0] != 4)
{
    //wrong packet type
    return false;
}

//get fbscab & temp sensor index from the packet
fbscab_index = packet->data[1];
sensor_index = packet->data[2];
//get raw temperature data from the packet
temp_lsb = packet->data[3];
temp_msb = packet->data[4];
temp_raw = temp_lsb | (temp_msb << 8);

//check temperature data CRC flags
crc_status = temp_raw << 14;
if (crc_status == 1)
{
    //CRC check failed
    return false;
}
if (crc_status == 2)
{
    //no sensor in this location
    //this should never happen
    return false;
}
if (crc_status == 3)
{
    //no valid data from this sensor yet
    return false;
}
//if we get to here, crc status==0, so temperature data is valid
//calculate temperature
deg_c = temp_raw / (float)16.0;
deg_f = (deg_c * 9.0) / 5.0 + 32.0;
//return text
sprintf(output, "FBSCAB:%d SENSOR:%d TEMP_DEGC:%0.2f
TEMP_DEGF:%0.2f", fbscab_index, sensor_index, deg_c, deg_f);
//done
return true;
}
```

**Datasheet Revision Table**

2025-03-06	Initial Release
2025-05-06	Correct mechanical drawings