GLK12232-25

Including GLK12232-25-USB

Technical Manual

Revision 3.1

PCB Revision: 3.0 or Higher

Firmware Revision: 5.8 or Higher

Revision History

Revision	Date	Description	Author
3.1	November 23, 2015	Initial Release	Divino

Contents

1 Introduction

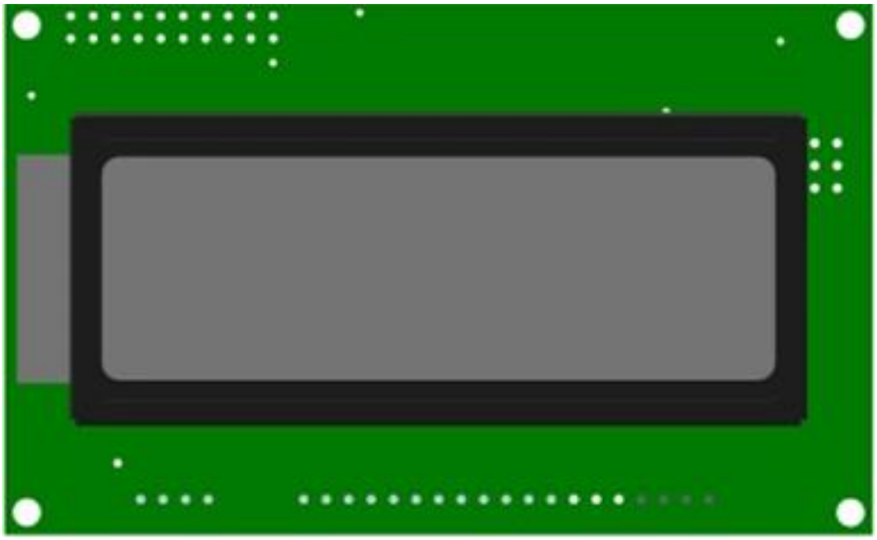


Figure 1: GLK12232-25 Display

The GLK12232-25 is an intelligent graphic liquid crystal display engineered to quickly and easily add an elegant creativity to any application. In addition to the RS232, TTL and I2C protocols available in the standard model, a USB communication model allows the GLK12232-25 to be connected to a wide variety of host controllers. Communication speeds of up to 115.2kbps for serial protocols and 100kbps for I^2C ensure lightning fast text and graphic display.

The simple command structure permits easy software control of many settings including backlight brightness, screen contrast, and baud rate. On board memory provides 16KB of customizable fonts and bitmaps to enhance the graphical user experience.

User input on the GLK12232-25 is available through a five by five matrix style keypad. Two general purpose outputs provide simple switchable five volt sources on each model.

The versatile GLK12232-25, with all the features mentioned above, is available in a variety of colour and temperature options to suit almost any application.

2 Quick Connect Guide

2.1 Available Headers

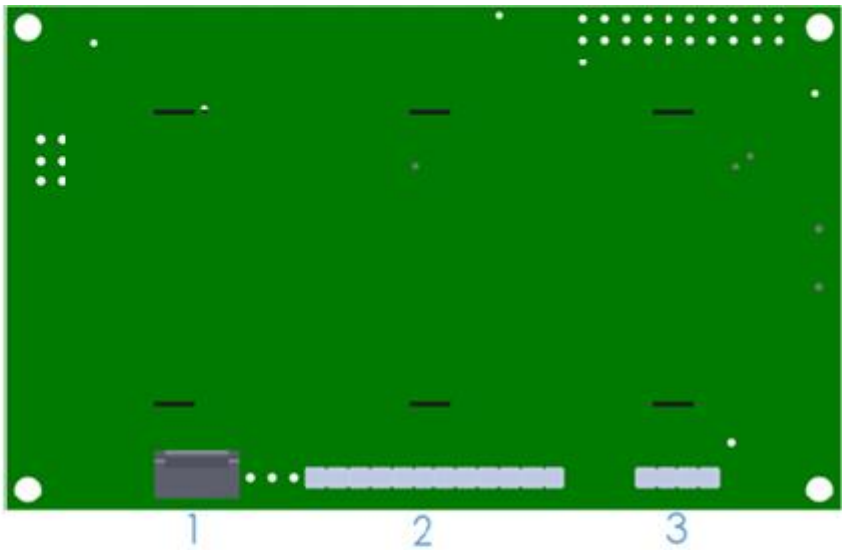


Figure 2: GLK12232-25 Standard Module Header Locations

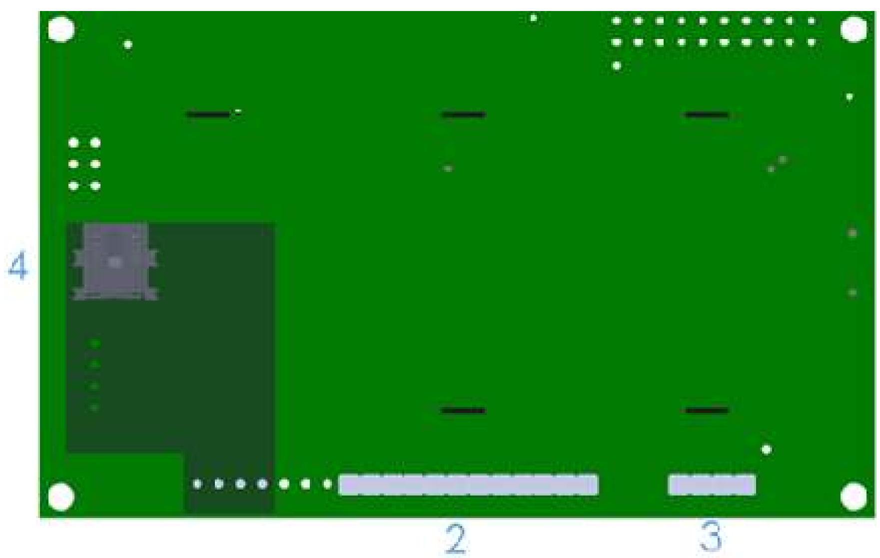


Figure 3: GLK12232-25 USB Model Header Locations

Table 1: List of Available Headers

#	Header	Mate	Population
1	Communication/Power Connector	ESCCPC5V/BBC	Standard Model Only
2	Keypad	KPP4x4	USB Model Only
3	GPO Header	None Offered	All Models
4	Mini USB Connector	EXTMUSB3FT/INTMUSB3FT	All Models

2.2 Standard Module

The standard version of the GLK12232-25 allows for user configuration of two common communication protocols. First, the unit can communicate using serial protocol at either RS323 or TTL voltage levels. Second, it can communicate using the Inter-Integrated Circuit connect, or I^2C protocol. Connections for each protocol can be accessed through the four pin Communication/Power Header as outlined in the Serial Connections and I^2C Connections sections below.

Recommended Parts

Figure 4: Extended Communication/Power Cable (ESCCPC5V)

The most common cable choice for any standard Matrix Orbital display, the Extended Communication/ Power Cable offers a simple connection to the unit with familiar interfaces. DB9 and floppy power headers provide all necessary input to drive your display.

Figure 5: Breadboard Cable (BBC)

For a more flexible interface to the GLK12232-25 , a Breadboard Cable may be used. This provides a simple four wire connection that is popular among developers for its ease of use in a breadboard environment.

Serial Connections

Serial protocol provides a classic connection to the GLK12232-25 . The Communication/Power Cable is most commonly used for this set up as it provides connections for DB9 serial and floppy power cables. To place your board in Serial mode, adhere to the steps laid out below.

1. Set the Protocol Select jumpers.
   - RS232: Connect the five jumpers* in the 232 protocol box with the zero ohm jumper resistors provided or an alternate wire or solder solution.
   - TTL: Connect the four jumpers* in the TTL protocol box.

***Note:** Jumpers must be removed from all protocol boxes save for the one in use.

2. Make the connections.
   a. Connect the six pin female header of the Communication/Power Cable to the Communication/Power Header of your GLK12232-25 .
   b. Insert the male end of your serial cable to the corresponding DB9 header of the Communication/Power Cable and the mate the female connector with the desired communication port of your computer.
   c. Select an unmodified floppy cable from a PC power supply and connect it to the power header of the Communication/Power Cable.
3. Create.
   - MOGD# or a terminal program will serve to get you started, and then you can move on with your own development. Instructions for the former can be found below and a variety of application notes are available for the latter at www.matrixorbital.ca/appnotes.

I^{2}C Connections

A more advanced connection to the GLK12232-25 is provided by the I^{2}C protocol setting. This is best accomplished using a breadboard and the Breadboard Cable. Power must be supplied from your breadboard or another external source. To dive right into your application and use the GLK12232-25 in I^{2}C mode, get started with the guidelines below.

1. Set the Protocol Select switches.
   - I^{2}C: Ensure that the two I^{2}C jumpers in the corresponding protocol box are connected while all others are open.
2. Make the connections.
   a. Connect the Breadboard Cable to the Communication/Power Header on your GLK12232-25 and plug the four leads into your breadboard. The red lead will require power, while the black should be connected to ground, and the green and yellow should be connected to your controller clock and data lines respectively.
   b. Pull up the clock and data lines to five volts using a resistance between one and ten kilohms on your breadboard.
3. Create.
   - This time you're on your own. While there are many examples within the Matrix Orbital AppNote section, www.matrixorbital.ca/appnotes, too many controllers and languages exist to cover them all. If you get stuck in development, it is possible to switch over to another protocol on the standard board, and fellow developers are always on our forums for additional support.

2.3 USB Module

The GLK12232-25-USB offers a single USB protocol for easy connection to a host computer. The simple and widely available protocol can be accessed using the on board mini B style USB connector as outlined in the USB Connections section.

Recommended Parts

Figure 6: Mini USB Cable (EXTMUSB3FT)

The External Mini USB cable is recommended for the GLK12232-25-USB display. It will connect to the miniB style header on the unit and provide a connection to a regular A style USB connector, commonly found on a PC.

USB Connections

The USB connection is the quickest, easiest solution for PC development. After driver installation, the GLK12232-25-USB will be accessible through a virtual serial port, providing the same result as a serial setup without the cable hassle. To connect to your GLK12232-25-USB please follow the steps below.

1. Set the Protocol Select jumpers.
   - USB: The GLK12232-25-USB offers USB protocol only. Model specific hardware prevents this unit from operating in any other protocol, and does not allow other models to operate in USB. Protocol Select jumpers on the USB model cannot be moved.
2. Make the connections.
   - Plug the mini-B header of your External Mini USB cable into your GLK12232-25-USB and the regular USB header into your computer USB jack.
3. Install the drivers.
   a. Download the latest drivers at www.matrixorbital.ca/drivers, and save them to a known location.
   b. When prompted, install the USB bus controller driver automatically
   c. If asked, continue anyway, even though the driver is not signed
   d. When the driver install is complete, your display will turn on, but communication will not yet be possible.
   e. At the second driver prompt, install the serial port driver automatically
   f. Again, if asked, continue anyway
4. Create.
   - Use MOGD# or a terminal program to get started, and then move on with your own development. Instructions for the former can be found below and a number of application notes are available for the latter at www.matrixorbital.ca/appnotes.

3 Software

The multiple communication protocols available and simple command structure of the GLK12232-25 means that a variety of applications can be used to communicate with the display. Text is sent to the display as a character string, for example, sending the decimal value 41 will result in an 'A' appearing on the screen. A single control character is also available. Commands are merely values prefixed with a special command byte, 254 in decimal.

Table 2: Reserved Control Characters

Control Characters			
7	Bell / Sound Buzzer	10	Line feed / New line

Once the correct communication port is identified, the following communication settings can be applied to communicate correctly with the GLK12232-25 .

Table 3: Communication Settings

BPS	Data Bits	Parity	Stop Bits	Flow Control
19200	8	None	1	None

Finally, with a communication port identified and correctly setup simple text strings or even command bytes can easily be transmitted to control your display.

3.1 MOGD#

The Matrix Orbital Graphic Display interface, MOGD#, is offered as a free download from www.matrixorbital.ca/software/software_graphic. It provides a simple graphical interface that allows settings, fonts, and bitmaps to be easily customised for any application.

While monochromatic bitmaps can easily be created in virtually any image editing program, MOGD# provides an extensive font generation suite to stylize your display to any project design. In addition to standard font wide modifications, character ranges can be specified by start and end values to eliminate unused symbols, and individual glyphs can be modified with a double click. Finally, text spacing can be tailored and a complete font library built with your Matrix Orbital graphic display.

Like uProject, MOGD# offers a scripting capability that provides the ability to stack, run, and save a series of commands. The most basic function is the Send Numeric tool which is used to transmit a string of values to the display to write text or execute a command.

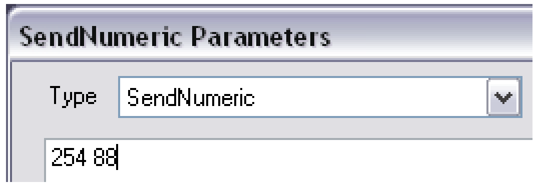


Figure 7: MOGD# Command Example

Again, the clear screen command is sent to a connected display, this time using the MOGD# Send Numeric function command style. Scripts can be run as a whole using the Play button from the toolbar or as single commands by selecting Step; once executed it must be Reset. Before issuing commands, it is a good idea to ensure communication with a display is successful using the autodetect button.

This program provides both a staging areas for your graphics display and a proving ground that will prepare it for any application environment.

3.2 Application Notes

Full demonstration programs and code are available for Matrix Orbital displays in the C# language from Simple C# AppNote Pack in the Application Note section at www.matrixorbital.ca/appnotes. Difficulty increases from beginner, with the Hello World program, to advanced with the Dallas One-Wire temperature reading application.

Many additional applications are available in a number of different programming languages. These programs are meant to showcase the capability of the display and are not intended to be integrated into a final design. For additional information regarding code, please read the On Code document also found on the support site.

4 Hardware

4.1 Standard Model

I^2C Communication/Power Header

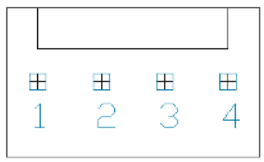


Figure 8: I2C Communication/Power Header

Table 4: I^2C Communication/Power Pinout

Pin	Function
1	Vcc
2	Rx (SCL)
3	Tx (SDA)
4	Gnd

Voltage is applied through pins one and four of the header, please reference the electrical specifications before applying power. Pins two and three are reserved for I^2C clock and data signals respectively, both of which should be pulled up to five volts using a resistance between one and ten kilohms. The Tyco 640456-4-LF style header used can be mated to a number of connectors, including Molex 22-01-3047.

4.2 USB Model

Mini USB Connector

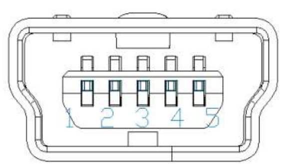


Figure 9: Mini USB Connector

Table 5: Mini USB Pinout

Pin	Function
1	Vcc
2	D-
3	D+
5	Gnd

The GLK12232-25-USB -USB comes with a familiar Mini USB Connector to fulfill both communication and power needs. The standard MiniB style header can be connected to any other USB style using the appropriate cable. Most commonly used with a PC, this connection creates a virtual com port that offers a simple power solution with a familiar communication scheme.

Alternate USB Header

Some advanced applications may prefer the straight four pin connection offered through the Optional Alternate USB Header. This header offers power and communication access in a simple interface package. The Optional Alternate USB Header may be added to the GLK12232-25-USB for an added charge as part of a custom order. Please use the Contact section to request more information from the friendly Matrix Orbital sales team.

4.3 Common Features

General Purpose Outputs

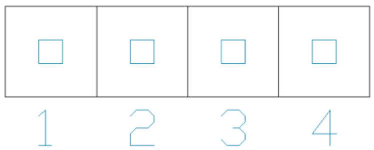


Figure 10: GPO Header

Table 6: GPO Pinout

Pin	Function
1	GPO 1
2	GND
3	GPO 2
4	GND

A unique feature of the GLK12232-25 is the ability to control relays* and other external devices using either one of two General Purpose Outputs. Each can source up to 20mA of current at five volts when on or sink 20mA at zero volts when off. The straight, four pin header can be interfaced to a number of female connectors to provide control to any peripheral devices required.

*Note: If connecting a relay, be sure that it is fully clamped using a diode and capacitor in order to absorb any electro-motive force (EMF) which will be generated.

Keypad Header

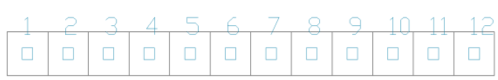


Figure 11: Keypad Header

Table 7: Keypad Pinout

Pin	Function	Pin	Function
1	Gnd	7	Column 1
2	Row 1	8	Column 2
3	Row 2	9	Column 3
4	Row 3	10	Column 4
5	Row 4	11	Column 5
6	Row 5	12	Gnd/Vcc*

To facilitate user input, the GLK12232-25 provides a Keypad Interface Connector which allows a matrix style keypad of up to twenty-five keys to be directly connected to the display module. Key presses are generated when a short is detected between a row and a column. When a key press is generated, a character specific to that key press is automatically sent on the Tx communication line. If a synchronous read method is desired in serial mode*, the "Auto Transmit Keypress" function can be turned off to allow the key presses to remain in the buffer so that they may be polled. The character that is associated with each key press may also be altered using the "Assign Key Codes" command. The straight twelve pin header of the Keypad Interface Connector will interface to a variety of different devices including the Matrix Orbital KPP4x4 keypad.

*Note: In I^{2}C mode, the "Auto Transmit Keypress" function should always be on, keypresses should not be polled.

**Note: The Ground / +5V pin is toggled by the jumper above the right of the keypad connector. Jump pads 1 & 2 for GND or 2 & 3 for +5V.

Protocol Select Jumpers

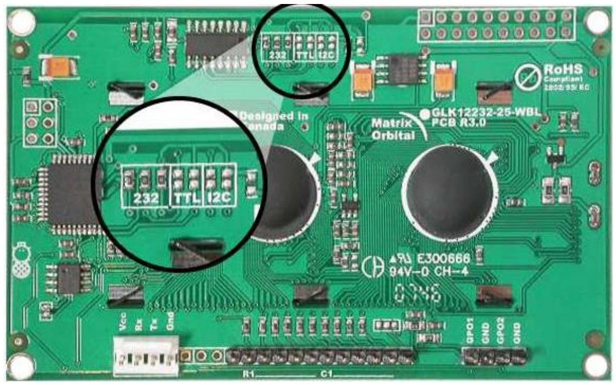


Figure 12: Protocol select Jumpers

The Protocol Select Jumpers provide the means necessary to toggle the GLK12232-25 between RS-232, TTL and I²C protocols. As a default, the jumpers are set to RS-232 mode with solder jumps on the RS232 jumpers. In order to place the display module in I²C mode you must first remove the solder jumps from the RS232 jumpers and then place them on the I²C jumpers. The display will now be in I²C mode and have a default slave address of 80, unless changed with the appropriate command. Similarly, in order to change the display to TTL mode, simply remove the zero ohm resistors from the RS232 or I²C jumpers and solder them to the TTL jumpers. Protocol resistors should be set to TTL for USB, and cannot be moved.

Hardware Lock

Figure 13: FileSystem Lock Jumper

The Hardware Lock allows fonts, bitmaps, and settings to be saved, unaltered by any commands. By connecting the two pads near the memory chip with a zero ohm resistor, the display will be locked. This supersedes the data lock command and cannot be circumvented by any software means. To unlock the display and make changes simply remove the jumper.

5 Troubleshooting

5.1 Power

In order for your Matrix Orbital display to function correctly, it must be supplied with the appropriate power. If the power LED near the top right corner of the board is not illuminated, power is not applied correctly. Try following the tips below.

- First, check the power cable which you are using for continuity. If you don't have an ohm meter, try using a different power cable, if this does not help try using a different power supply.
- If power is applied through the DB9 connector, ensure that the Power Through DB9 Jumper is connected.
- If changes have been made to the protocol select block, ensure all the appropriate protocol select jumpers are connected and all unused protocol jumpers are disconnected.
- The last step will be to check the interface connector in use on your display. If the power connections have become loose, or you are unable to resolve the issue, please Contact Matrix Orbital for more information.

5.2 Display

If your display is powered successfully, the Matrix Orbital logo, or user created screen should display on start up. If this is not the case, check out these tips.

- Ensure the contrast is not too high or too low. This can result in a darkened or blank screen respectively. See the Manual Override section to reset to default.
- Make sure that the start screen is not blank. It is possible to overwrite the Matrix Orbital logo start screen, if this happens the screen may be blank. Try writing to the display to ensure it is functional, after checking the contrast above.

5.3 Communication

When communication of either text or commands is interrupted, try the steps below.

- First, check the communication cable for continuity. If you don't have an ohm meter, try using a different communication cable. If you are using a PC try using a different Com/USB Port.
- Next, please ensure that the display module is set to communicate on the protocol that you are using, by checking the Protocol Select Jumpers.
- In serial and USB protocols, ensure that the host system and display module are both communicating on the same baud rate. The default rate for the display module is 19200 bps.
- Match Rx from your display to the transmitting pin from your host and the Tx pin to the receiving pin.
- If you are communicating to the display via I^2C* please ensure that the data is being sent to the correct address. The default slave address for the display module is 80.

- In I^2C mode, connect Rx to the clock line of your controller and Tx to the data output.
- Unlock the display. See the Set and Save Data Lock command for more info.
- Finally, you may reset the display to its default settings using the Manual Override procedure outlined below.

***Note:** I^2C communication will always require pull up resistors on SCL and SDA of one to ten kilohms.

5.4 Manual Override

Should the settings of your display become altered in a way that dramatically impacts usability, the default settings can be temporarily restored. To override the display, please follow the steps below.

1. Disconnect power from your display.
2. Place a jumper on the two manual override pins, for the GLK12232-25 model these are the middle two keypad pins.
3. Reconnect power to your unit, and wait for the start screen before removing the jumper. Please note the jumper will adversely affect performance if left in place during use.
4. Settings will be temporarily** overridden to the defaults listed in the Manual Override Settings table. At this point any important settings, such as contrast, backlight, or baud rate, should not only be set but saved so they remain when the override is removed.

Parameter	Value
Backlight	255
Contrast	128
Baud Rate	19200
I^2C Address	80

Table 8: Manual Override Settings

****Note:** The display module will revert back to the old settings once turned off, unless desired settings are saved.

6 Commands

6.1 Communication

1.1 Change Baud Rate	Dec	254 57	Speed	v5.8
	Hex	FE 39	Speed	
	ASCII	■ 9	Speed	

Immediately changes the baud rate. Not available in I2C. Baud rate can be temporarily forced to 19200 by a manual override.

Speed	Byte	Valid settings shown below.

Table 9: Accepted Baud Rate Values

Rate	9600	14400	19200	28800	38400	57600	76800	115200
Speed	207	138	103	68	51	34	25	16

1.2 Change I2C Slave Address	Dec	254 51	Address	v5.8
	Hex	FE 33	Address	
	ASCII	■ 3	Address	

Immediately changes the I2C write address. Only even values are permitted as the next odd address will become the read address. Default is 80.

Address	Byte	Even value.

1.3 Set a Non-Standard Baud Rate	Dec	254 164	Speed	v5.0
	Hex	FE A4	Speed	
	ASCII	■ ñ	Speed	

Immediately changes the baud rate to a non-standard value. Speed must be a whole number between 977 and 153800. Due to rounding, error increases with baud rate, actual baud must be within 3% of desired baud to ensure accurate communication. Not available in I2C. Can be temporarily forced to 19200 by a manual override.

Speed	Short	Calculations shown below, standard crystal speed is 16MHz.

$$Speed = \frac{CrystalSpeed}{(8 \times DesiredBaud)} - 1$$

Equation 1: Speed Byte Calculation

$$ActualBaud = \frac{CrystalSpeed}{(8 \times (Speed + 1))}$$

Equation 2: Actual Baud Rate Calculation

$$\frac{|DesiredBaud - ActualBaud|}{DesiredBaud} < 0.03$$

Equation 3: Baud Rate Error Calculation

1.4 Turn Software Flow Control On	Dec	254 58	AlmostFull AlmostEmpty	v5.8
	Hex	FE 3A	AlmostFull AlmostEmpty	
	ASCII	■ :	AlmostFull AlmostEmpty	

Enables simple flow control. The display will return a single, Xoff, byte to the host when the display buffer is almost full and a different, Xon, byte when the buffer is almost empty. Full value should provide enough room for the largest data packet to be received without buffer overflow. No data should be sent to the display between full and empty responses to permit processing. Buffer size is 256* bytes. Not available in I2C. Default off.

AlmostFull	Byte	Number of bytes remaining before buffer is completely full. Value between 0 and 128.
AlmostEmpty	Byte	Number of bytes before buffer can be considered empty enough to accept data.

1.5 Turn Software Flow Control Off	Dec	254 59	v5.8
	Hex	FE 3B	
	ASCII	■ ;	

Disables flow control. Bytes sent to the display may be permitted to overflow the buffer resulting in data loss.

6.2 Text

2.1 Clear Screen	Dec	254 88	v5.8
	Hex	FE 58	
	ASCII	■ X	

Clears the contents of the screen.

2.2 Go Home	Dec	254 72	v5.8
	Hex	FE 48	
	ASCII	■ H	

Returns the cursor to the top left of the screen.

2.3 Set Cursor Position	Dec	254 71	Column Row	v5.8
	Hex	FE 47	Column Row	
	ASCII	■ G	Column Row	

Sets the cursor to a specific cursor position where the next transmitted character is printed.

Column	Byte	Value between 1 and number of character columns.
Row	Byte	Value between 1 and number of character rows.

2.4 Set Cursor Coordinate	Dec	254 121	X Y	v5.8
	Hex	FE 79	X Y	
	ASCII	■ y	X Y	

Sets the cursor to an exact pixel position where the next transmitted character is printed.

X	Byte	Value between 1 and screen width, represents leftmost character position.
Y	Byte	Value between 1 and screen height, represents topmost character position.

2.5 Auto Scroll On	Dec	254 81	v5.8
	Hex	FE 51	
	ASCII	■ Q	

The entire contents of screen are shifted up one line when the end of the screen is reached. Display default is on.

2.6 Auto Scroll Off	Dec	254 82	v5.8
	Hex	FE 52	
	ASCII	■ R	

New text is written over the top line when the end of the screen is reached. Display default is Auto Scroll on.

6.3 Drawing

3.1 Set Drawing Colour	Dec	254 99	Colour	v5.8
	Hex	FE 63	Colour	
	ASCII	■ c	Colour	

Set the colour to be used for all future drawing commands that do not implicitly specify colour.

Colour	Byte	0 for background or any other value for text colour.

3.2 Draw Pixel	Dec	254 112	X Y	v5.8
	Hex	FE 70	X Y	
	ASCII	■ p	X Y	

Draw a single pixel at the specified coordinate using the current drawing colour.

X	Byte	Horizontal position of pixel to be drawn, zero indexed from left.
Y	Byte	Vertical position of pixel to be drawn, zero indexed from top.

3.3 Draw a Line	Dec	254 108	X1 Y1 X2 Y2	v5.8
	Hex	FE 6C	X1 Y1 X2 Y2	
	ASCII	■ l	X1 Y1 X2 Y2	

Draw a line connecting two termini. Lines may be rendered differently when drawn right to left versus left to right.

X1	Byte	Horizontal coordinate of the first terminus, zero indexed from left.
Y1	Byte	Vertical coordinate of the first terminus, zero indexed from top.
X2	Byte	Horizontal coordinate of second the terminus, zero indexed from left.
Y2	Byte	Vertical coordinate of second the terminus, zero indexed from top.

3.4 Continue a Line	Dec	254 101	X Y	v5.8
	Hex	FE 65	X Y	
	ASCII	■ e	X Y	

Draw a line from the last point drawn to the coordinate specified using the current drawing colour.

X	Byte	Left coordinate of the terminus, zero indexed from left.
Y	Byte	Top coordinate of the terminus, zero indexed from top.

3.5 Draw a Rectangle	Dec	254 114	Colour X1 Y1 X2 Y2	v5.8
	Hex	FE 72	Colour X1 Y1 X2 Y2	
	ASCII	■ r	Colour X1 Y1 X2 Y2	

Draw a rectangular frame one pixel wide using the colour specified; current drawing colour is ignored.

Colour	Byte	0 for background or any other value for text colour.
X1	Byte	Leftmost coordinate of the rectangle, zero indexed from left.
Y1	Byte	Topmost coordinate of the rectangle, zero indexed from top.
X2	Byte	Rightmost coordinate of the rectangle, zero indexed from left.
Y2	Byte	Bottommost coordinate of the rectangle, zero indexed from top.

3.6 Draw a Filled Rectangle				v5.8
	Dec	254 120	Colour X1 Y1 X2 Y2	
	Hex	FE 78	Colour X1 Y1 X2 Y2	
	ASCII	■ x	Colour X1 Y1 X2 Y2	

Draw a filled rectangle using the colour specified; current drawing colour is ignored.

Colour	Byte	0 for background or any other value for text colour.
X1	Byte	Leftmost coordinate of the filled rectangle, zero indexed from left.
Y1	Byte	Topmost coordinate of the filled rectangle, zero indexed from top.
X2	Byte	Rightmost coordinate of the filled rectangle, zero indexed from left.
Y2	Byte	Bottommost coordinate of the filled rectangle, zero indexed from top.

3.7 Initialize a Bar Graph				V5.8
	Dec	254 103	ID Type X1 Y1 X2 Y2	
	Hex	FE 67	ID Type X1 Y1 X2 Y2	
	ASCII	■ g	ID Type X1 Y1 X2 Y2	

Initialize a bar graph in memory for later implementation. Graphs can be located anywhere on the screen, but overlapping may cause distortion. Graph should be filled using the Draw a Bar Graph command.

ID	Byte	Unique bar identification number, between 0 and 255.
Type	Byte	Graph style, see Bar Graph Types.
X1	Byte	Leftmost coordinate of the bar, zero indexed from left.
Y1	Byte	Topmost coordinate of the bar, zero indexed from top.
X2	Byte	Rightmost coordinate of the bar, zero indexed from left.
Y2	Byte	Bottommost coordinate of the bar, zero indexed from top.

Table 10: Bar Graph Types

	Direction	Base
0	Vertical	Bottom
1	Horizontal	Left
2	Vertical	Top
3	Horizontal	Right

3.8 Draw a Bar Graph				V5.8
	Dec	254 105	ID Value	
	Hex	FE 69	ID Value	
	ASCII	■ i	ID Value	

Fill in a portion of a bar graph after initialization. Any old value will be overwritten by the new. Setting a value of zero before setting a new value will restore a graph should it become corrupted.

ID	Byte	Unique bar identification number, value between 0 and 255.
Value	Byte	Portion of graph to fill in pixels, will not exceed display bounds.

3.9 Initialize a Strip Chart	Dec	254 110	ID X1 Y1 X2 Y2	V5.8
	Hex	FE 6E	ID X1 Y1 X2 Y2	
	ASCII	■ n	ID X1 Y1 X2 Y2	

Designate a portion of the screen for a chart. Visual changes will occur when the update command is issued.

ID	Byte	Unique chart identification number, value between 0 and 7.
X1	Byte	Leftmost coordinate of the strip chart, zero indexed from left.
Y1	Byte	Topmost coordinate of the strip chart, zero indexed from top.
X2	Byte	Rightmost coordinate of the strip chart, zero indexed from left.
Y2	Byte	Bottommost coordinate of the strip chart, zero indexed from top.

Table 11: Strip Chart Types (Bytes 3-0)

Type	Description
0	Bar
1	Line
2	Step
3	Box

3.10 Update a Strip Chart	Dec	254 111	ID Value	v5.8
	Hex	FE 6F	ID Value	
	ASCII	■ o	ID Value	

Shift the specified strip chart and draw a new value.

ID	Byte	Chart identification number, value between 0 and 7.
Value	Short	Value to add to the chart.

6.4 Fonts

4.1 Upload a Font File	Dec	254 36	ID Size Data	v5.8
	Hex	FE 24	ID Size Data	
	ASCII	■ $	ID Size Data	

Upload a font to a graphic display. To create a font see the Font File Creation section, for upload protocol see the File Transfer Protocol entry. Default font is ID 1.

ID	Byte	Unique font identification number, value between 0 and 255.
Size	Short	Size of the entire font file.
Data	Byte(s)	Font file data, see the Font File Creation example.

4.2 Set the Current Font	Dec	254 49	ID	v5.8
	Hex	FE 31	ID	
	ASCII	■ 1	ID	

Set the font in use by specifying a unique identification number. Characters sent after the command will appear in the font specified; previous text will not be affected. Default is 1.

ID	Byte	Unique font identification number, value between 0 and 255.

4.3 Set Font Metrics	Dec	254 50	LineMargin TopMargin CharSpace LineSpace Scroll	v5.8
	Hex	FE 32	LineMargin TopMargin CharSpace LineSpace Scroll	
	ASCII	■ 2	LineMargin TopMargin CharSpace LineSpace Scroll	

Set the font spacing, or metrics, used with the current font. Changes only appear in text sent after command.

LineMargin	Byte	Space between left of display and first column of text. Default 0.
TopMargin	Byte	Space between top of display area and first row of text. Default 0.
CharSpace	Byte	Space between characters. Default 0.
LineSpace	Byte	Space between character rows. Default 1.
Scroll	Byte	Point at which text scrolls up screen to display additional rows. Default 1.

4.4 Set Box Space Mode	Dec	254 172	Switch	v5.8
	Hex	FE AC	Switch	
	ASCII	■ ¼	Switch	

Toggle box space on or off. When on, a character sized box is cleared from the screen before a character is written. This eliminates any text or bitmap remnants behind the character. Default is on.

Switch	Byte	1 for on or 0 for off.

Font File Creation

Matrix Orbital graphic displays are capable of displaying text in a wide variety of styles customizable to suit any project design. Front files alter the style of text and appearance of the display.

By default, a Matrix Orbital graphic display is loaded with a small filled font in slot one and a future bk bt 16 style in slot two. Both are available at www.matrixorbital.ca/software/graphic_fonts.

The easiest way to create, add, or modify the fonts of any graphic display is through the MOGD# tool. This provides a simple graphic interface that hides the more complex intricacies of the font file.

Table 12: Example Font File Header

Maximum Width	Character Height	ASCII Start Value	ASCII End Value
5	7	104	106

The font file header contains four bytes: First, the number of columns in the widest character; usually 'w', second, the pixel height of each character, and finally, the start and end values of the character range. The range represents the values that must be sent to the display to trigger the characters to appear on the screen. In the example, the decimal values corresponding to the lowercase letters 'h' through 'j' will be used resulting in the range shown.

Table 13: Example Character Table

	MSB	LSB	Width
h	0	13	5
i	0	18	3
j	0	21	4

The character table contains information that allows the display to locate each individual character in a mass of character data. Each character has three bytes; two indicating it's offset in the character data and one indicating its width. The offset takes into account the header and table bytes to point to the first byte of the character data it references. The first byte of the file, maximum width, has an offset of zero. The width byte of each character can be identical as in a fixed width font, or in our case, variable. The character table will become clearer after analyzing the final part of the font file, character data.

Table 14: Character 'h' Bitmap

1	0	0	0	0
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1

Table 15: Character 'h' Data

1	0	0	0	0	1	0	0	84	**132**
0	0	1	0	1	1	0	1	2D	**45**
1	0	0	1	1	0	0	0	98	**152**
1	1	0	0	0	1	1	0	C6	**198**
0	0	1	0	0	0	0	0	20	**32**

The character data is a binary graphical representation of each glyph in a font. Each character is drawn on a grid containing as many rows as the height specified in the header and as many columns as the width specified in the character table. Cells are drawn by writing a one in their location and cleared by setting a value of zero. Starting at the top left, moving right, then down, eight of these cells form a character data byte. When all cells are accounted for, zeroes may be added to the last byte to complete it. A sample of an 'h' glyph is shown above. The data for the 'i' and 'j' characters will follow to complete the custom font file displayed below.

Table 16: Example Font File

Header	5 7 104 106
Character Table	0 13 5
	0 18 3
	0 21 4
Character Data	132 45 152 198 32
	67 36 184
	16 49 25 96

6.5 Bitmaps

5.1 Upload a Bitmap File	Dec	**254 94**	ID Size Data	**v5.8**
	Hex	**FE 5E**	ID Size Data	
	ASCII	**■ ^**	ID Size Data	

Upload a bitmap to a graphic display. To create a bitmap see the Bitmap File Creation section, for upload protocol see the File Transfer Protocol entry. Start screen is ID 1.

ID	**Byte**	Unique bitmap identification number, value between 0 and 255.
Size	**Short**	Size of the entire bitmap file.
Data	**Byte(s)**	Bitmap file data, see the Bitmap File Creation example.

5.2 Draw a Bitmap from Memory	Dec	254 98	ID X Y	v5.8
	Hex	FE 62	ID X Y	
	ASCII	■ b	ID X Y	

Draw a previously uploaded bitmap from memory. Top left corner must be specified for drawing.

ID	**Byte**	Unique bitmap identification number, value between 0 and 255.
X	**Byte**	Leftmost coordinate of bitmap.
Y	**Byte**	Topmost coordinate of bitmap.

5.3 Draw a Bitmap Directly	Dec	254 100	X1 Y1 Data	v5.8
	Hex	FE 64	X1 Y1 Data	
	ASCII	■ d	X1 Y1 Data	

Draw a bitmap directly to the graphic display without saving to memory. Cannot be implemented in a script.

X1	**Byte**	Leftmost coordinate of bitmap.
Y1	**Byte**	Topmost coordinate of bitmap.
Data	**Byte(s)**	Bitmap file data, see the Bitmap File Creation example.

Bitmap File Creation

In addition to fonts, Matrix Orbital graphic displays can also hold a number of customizable bitmaps to provide further stylistic product integration. Like font files, bitmaps files are most easily uploaded to a display using MOGD#. However, the critical data component of the bitmap upload command is detailed below for reference.

The bitmap data block is similar to that of a font. However, as a bitmap is a single glyph, only a simple two byte header is required. First, one byte representing the bitmap width is sent, then one byte for the height. Each bitmap is merely encoded in binary fashion using a series of ones and zeroes. Again a grid can be created using the width and height specified in the upload command, populated in the manner above, and converted into byte values. A smiley face example is shown below to indicate the ultimate effect of the Matrix Orbital graphic stylization ability.

Table 17: Smiley Face Bitmap

0	**1**	0	**1**	0
0	0	0	0	0
1	0	0	0	**1**
0	**1**	**1**	**1**	0

Table 18:Smiley Face Data

0	**1**	0	**1**	0	0	0	0	50	80
0	0	**1**	0	0	0	**1**	0	22	34
1	**1**	**1**	0	0	0	0	0	E0	224

Table 19: Example Bitmap File

Header	5 4
Bitmap Data	80 34 224

6.6 General Purpose Output

6.1 General Purpose Output On				v5.8
	Dec	254 87	Number	
	Hex	FE 57	Number	
	ASCII	■ W	Number	

Turns the specified GPO on, sourcing current from an output of five volts.

Number	Byte	GPO to be turned on.

6.2 General Purpose Output Off				v5.8
	Dec	254 86	Number	
	Hex	FE 56	Number	
	ASCII	■ V	Number	

Turns the specified GPO off, sinking current to an output of zero volts.

Number	Byte	GPO to be turned off.

6.3 Set Start Up GPO State				v5.8
	Dec	254 195	Number State	
	Hex	FE C3	Number State	
	ASCII	■ ├	Number State	

Sets and saves the start-up state of the specified GPO in non-volatile memory. Changes will be seen on start up.

Number	Byte	GPO to be controlled.
State	Byte	1 for on or 0 for off.

6.7 Keypad

7.1 Auto Transmit Key Presses On			v5.8
	Dec	254 65	
	Hex	FE 41	
	ASCII	■ A	

Key presses are automatically sent to the host when received by the display. Use this mode for I2C transactions.

7.2 Auto Transmit Key Presses Off			v5.8
	Dec	254 79	
	Hex	FE 4F	
	ASCII	■ O	

Key presses are held in the 10 key buffer to be polled by the host using the Poll Key Press command. Default is Auto Transmit on.

7.3 Poll Key Press			v5.8
	Dec	254 38	
	Hex	FE 26	
	ASCII	■ &	

Reads the last unread key press from the 10 key display buffer. If another key is stored in the buffer the MSb will be 1, the MSb will be 0 when the last key press is read. If there are no stored key presses a value of 0 will be returned. Auto transmit key presses must be turned off for this command to be successful, do not use with I^2C.

Response	Byte	Value of key pressed (MSb determines additional keys to be read).

7.4 Clear Key Buffer	Dec	254 69	v5.8
	Hex	FE 45	
	ASCII	■ E	

Clears all key presses from the key buffer.

7.5 Set Debounce Time	Dec	254 85	Time	v5.8
	Hex	FE 55	Time	
	ASCII	■ U	Time	

Sets the time between a key press and a key read by the display. Most switches will bounce when pressed; the debounce time allows the switch to settle for an accurate read. Default is 8 representing approximately 52ms.

Time	Byte	Debounce increment (debounce time = Time * 6.554ms).

7.6 Set Auto Repeat Mode	Dec	254 126	Mode	v5.8
	Hex	FE 7E	Mode	
	ASCII	■ DEL	Mode	

Sets key press repeat mode to typematic or hold. In typematic mode if a key press is held, by default the key value is transmitted immediately, then 5 times a second after a 1 second delay. In hold mode, the key down value is transmitted once when pressed, and then the key up value is sent when the key is released. Default is typematic.

Mode	Byte	1 for hold mode or 0 for typematic.

7.7 Auto Repeat Mode Off	Dec	254 96	v5.8
	Hex	FE 60	
	ASCII	■ `	

Turns auto repeat mode off. Default is on (typematic).

7.8 Assign Keypad Codes	Dec	254 213	Key Down	Key Up	V5.8
	Hex	FE D5	Key Down	Key Up	
	ASCII	■ Ｆ	Key Down	Key Up	

Assigns the key down and key up values sent to the host when a key press is detected. A key up and key down value must be sent for every key, a value of 255 will leave the key unaltered. Defaults are shown below.

Key Down	Bytes [25]	Key down values, beginning at row one column one moving right then down.
Key Up	Bytes [25]	Key up values, beginning at row one column one moving right then down.

Table 20: Default Key Down Values

Key Down				
A(65)	B(66)	C(67)	D(68)	E(69)
F(70)	G(71)	H(72)	I(73)	J(74)
K(75)	L(76)	M(77)	N(78)	O(79)
P(80)	Q(81)	R(82)	S(83)	T(84)
U(85)	V(86)	W(87)	X(88)	Y(89)

Table 21: Default Key Up Values

Key Up				
a(97)	b(98)	c(99)	d(100)	e(101)
f(102)	g(103)	h(104)	i(105)	j(106)
k(107)	l(108)	m(109)	n(110)	o(111)
p(112)	q(113)	r(114)	s(115)	t(116)
u(117)	v(118)	w(119)	x(120)	y(121)

6.8 Display Functions

8.1 Backlight On	Dec	254 66	Minutes	v5.8
	Hex	FE 42	Minutes	
	ASCII	■ B	Minutes	

Turns the display backlight on for a specified length of time. If an inverse display color is used this command will essentially turn on the text.

Minutes	Byte	Number of minutes to leave backlight on, a value of 0 leaves the display on indefinitely.

8.2 Backlight Off	Dec	254 70	v5.8
	Hex	FE 46	
	ASCII	■ F	

Turns the display backlight off. If an inverse display colour is used this command will turn off the text.

8.3 Set Brightness	Dec	254 153	Brightness	v5.8
	Hex	FE 99	Brightness	
	ASCII	■ Ö	Brightness	

Immediately sets the backlight brightness. If an inverse display color is used this represents the text colour intensity instead. Default is 255.

Brightness	Byte	Brightness level from 0(Dim) to 255(Bright).

8.4 Set and Save Brightness	Dec	254 152	Brightness	v5.8
	Hex	FE 98	Brightness	
	ASCII	■ ÿ	Brightness	

Immediately sets and saves the backlight brightness. Although brightness can be changed using the set command, it is reset to this saved value on start up. Default is 255.

Brightness	Byte	Brightness level from 0(Dim) to 255(Bright).

8.5 Set Contrast	Dec	254 80	Contrast	v5.8
	Hex	FE 50	Contrast	
	ASCII	■ P	Contrast	

Immediately sets the contrast between background and text. If an inverse display color is used this also represents the text brightness. Default is 128.

Contrast	Byte	Contrast level from 0(Light) to 255(Dark).

8.6 Set and Save Contrast	Dec	254 145	Contrast	v5.8
	Hex	FE 91	Contrast	
	ASCII	■ æ	Contrast	

Immediately sets and saves the contrast between background and text. Although contrast can be changed using the set command, it is reset to this saved value on start up. Default is 128.

Contrast	Byte	Contrast level from 0(Light) to 255(Dark).

6.9 Filesystem

9.1 Delete Filesystem	Dec	254 33 89 33	v5.8
	Hex	FE 21 59 21	
	ASCII	■ ! Y !	

Completely erase all fonts and bitmaps from a graphic display. Extended length of the command is intended to prevent accidental execution. To ensure filesystem integrity, cycle power to the display after erasure.

9.2 Delete a File	Dec	254 173	Type ID	v5.8
	Hex	FE AD	Type ID	
	ASCII	■ ¡	Type ID	

Removes a single font or bitmap file given the type and unique identification number. Cycle power after deletion.

Type	Byte	0 for font or 1 for bitmap.
ID	Byte	Unique identification number of font or bitmap to be deleted, value between 0 and 255.

9.3 Get Filesystem Space	Dec	254 175	v5.8
	Hex	FE AF	
	ASCII	■ »	

Returns the amount of space remaining in the display for font or bitmap uploads.

Response	Integer	Number of bytes remaining in memory.

9.4 Get Filesystem Directory	Dec	254 179	v5.8
	Hex	FE B3	
	ASCII	■ |	

Returns a directory to the contents of the filesystem. The total number and type of each entry will be provided.

Response	Short	Number of entries.
	Byte(s)	One entry for every file, 8 identification bytes for each entry.

Table 22: Filesystem Identification Bytes

Byte	7	6	5	4	3	2	1	0
Description	Size(MSB)	Size	Size	Size(LSB)	Type(4)/ID(4)	ID (LSB)	Start Page (MSB)	Start Page (LSB)

Table 23: Extended Byte Descriptions

Size	The complete file size.
Type/ID	First four bits designate file type, 0 for font or 1 for bitmap, remaining 12 bits indicate ID number.
Start Page	Memory start page, a value of 0 indicates entry is not in use.

9.5 Filesystem Upload	Dec	254 176	Size Data	v5.8
	Hex	FE B0	Size Data	
	ASCII	■ ░	Size Data	

This command will upload a filesystem image to the display. The size used is almost always the entire memory. Filesystem data can be uploaded LSB to MSB using the File Transfer Protocol.

Size	Integer	Size of the filesystem to upload.
Data	Byte(s)	Filesystem data to upload.

9.6 Filesystem Download	Dec	254 48		v5.8
	Hex	FE 30		
	ASCII	■ 0		
Downloads complete filesystem containing all fonts and bitmaps stored in the display using the File Transfer Protocol. A veritable heap of data.				
Response	Integer	Size of the filesystem to download.		
	Byte(s)	Filesystem data to download.		

9.7 File Download	Dec	254 178	Type ID	v5.8
	Hex	FE B2	Type ID	
	ASCII	■ ▓	Type ID	
Downloads a single font or bitmap file from the display to the host using the File Transfer Protocol.				
Type	Byte	Variable length, see File Types.		
ID	Byte	Unique identification number of font or bitmap to download, value between 0 and 1023.		
Response	Short	File size.		
	Byte(s)	File data.		

9.8 File Move	Dec	254 180	OldType OldID NewType NewID	v5.8
	Hex	FE B4	OldType OldID NewType NewID	
	ASCII	■ ┤	OldType OldID NewType NewID	
Used to move a single file and/or alter the type of an existing file. Old ID location must be valid and new ID empty.				
OldType	Byte	Original file type, value between 0 and 1023, see File Types.		
OldID	Byte	Original unique file identification number, value between 0 and 1023.		
NewType	Byte	New file type, see File Types.		
NewID	Byte	New unique file identification number.		

Table 24: File Types

Font	Bitmap	Script	Animation
0	1	2	3

File Transfer Protocol

Once a bitmap or font file has been created and paired to its command it must be sent using a file protocol developed specifically for Matrix Orbital displays. Once a file upload command has been sent requesting a unique reference number and specifying the file size required, the display will respond indicating whether it has enough room to save the file or not. As is the case throughout the upload protocol, a response of 1 will indicate confirmation while an 8 corresponds to rejection and will terminate the session.

Table 25: Upload Protocol Responses

Value	Action	Description
1	Acknowledged	Transfer successful, upload continues
8	Not Acknowledged	Transfer failed, abort upload

Once a file is confirmed to fit within the display, the upload will begin. A protocol is used here to ensure each byte is uploaded successfully. After each byte is sent, the module will echo it back to the host. It should then be checked against the value originally sent before a confirmation byte of 1 is returned. If the transmitted and echoed values do not match the upload should be aborted by sending a value of 8 instead. The upload will continue in this manner as indicated by the examples below which utilize familiar font and bitmap files.

Table 26: Font Upload Protocol

Host	Display	Comments
254		Command Prefix
36		Upload Font File Command
1		Reference ID LSB
0		Reference ID MSB
31		Font File Size LSB
0		Font File Size
0		Font File Size
0		Font File MSB
	1	Acknowledge Size
5		First Font Data Byte
	5	Echo Data Byte
1		Acknowledge Data Byte
7		Second Font Data Byte
…	…	…
96		Last Font Data Byte
	96	Echo Data Byte
1		Acknowledge Data Byte

Table 27: Bitmap Upload Protocol

Host	Display	Comments
254		Command Prefix
94		Upload Bitmap File Command
1		Reference ID LSB
0		Reference ID MSB
5		Bitmap File Size LSB
0		Bitmap File Size
0		Bitmap File Size
0		Bitmap File MSB
	1	Acknowledge Size
5		First Bitmap Data Byte
	5	Echo Data Byte
1		Acknowledge Data Byte
4		Second Bitmap Data Byte
…	…	…
224		Last Bitmap Data Byte
	224	Echo Data Byte
1		Acknowledge Data Byte

It should be noted that the display has a timeout setting of 2.1 seconds before it resets to prevent it from hanging during the upload process. Upon reset, the values 254 and 212 will be returned to indicate an error or lengthy delay has occurred in the upload process. If everything goes smoothly, the protocol will end with the host transmitting a final confirmation byte and the font will be stored in the display ready for any application.

6.10 Data Security

10.1 Set Remember	Dec	**254 147**	Mode	**v5.8**
	Hex	**FE 93**	Mode	
	ASCII	**■ ô**	Mode	

Allows changes to specific settings to be saved to the display memory. Writing to non-volatile memory can be slow and each change consumes 1 write of at least 100,000 available. The Command Summary outlines which commands are saved always, never, and when this command is on only. Remember is off by default.

Mode	**Byte**	1 for on or 0 for off.

10.2 Set Data Lock	Dec	254 202 245 160	Level	v5.8
	Hex	FE CA F5 A0	Level	
	ASCII	■ ╩ ⌡ á	Level	

Temporarily locks certain aspects of the display to ensure no inadvertent changes are made. The lock is released after a power cycle. A new level overrides the old, and levels can be combined. Default is 0.

Level	Byte	Lock level, see Data Lock Bits table.

Table 28: Data Lock Bits

Display	Command	Filesystem	Setting	Address	Reserved	Reserved	Reserved
7	6	5	4	3	2	1	0

Table 29: Lock Parameters

Reserved	Place holders only, should be 0
Address	Locks the Baud Rate and I2C address
Setting	Locks all settings from being saved
Filesystem	Locks all bitmaps and fonts
Command	Locks all commands, text can still be written
Display	Locks entire display, no new text can be displayed

10.3 Set and Save Data Lock	Dec	254 203 245 160	Level	v5.8
	Hex	FE CB F5 A0	Level	
	ASCII	■ ╦ ⌡ á	Level	

Locks certain aspects of the display to ensure no inadvertent changes are made. The lock is not affected by a power cycle. A new level overrides the old, and levels can be combined. Default is 0.

Level	Byte	See Data Lock Bits table.

6.11 Miscellaneous

11.1 Write Customer Data	Dec	254 52	Data	v5.8
	Hex	FE 34	Data	
	ASCII	■ 4	Data	

Saves a user defined block of data to non-volatile memory. Useful for storing display information for later use.

Data	Byte(s)	User defined data, 16 bytes.

11.2 Read Customer Data	Dec	254 53	v5.8
	Hex	FE 35	
	ASCII	■ 5	

Reads data previously written to non-volatile memory. Data is only changed when written, surviving power cycles.

Response	Byte(s)	Previously saved user defined data, 16 bytes.

11.3 Read Version Number	Dec	254 54	v5.8
	Hex	FE 36	
	ASCII	■ 6	

Causes display to respond with its firmware version number. Often used as a communication test.

Response	Byte	Convert to hexadecimal to view major and minor revision numbers.

11.4 Read Module Type	Dec	254 55	v8.0
	Hex	FE 37	
	ASCII	■ 7	

Causes display to respond with its module number.

Response	Byte	Module number, see Sample Module Type Response for a partial list.

Table 30: Sample Module Type Responses

33	GLK12232-25-USB
34	GLK12232-25

7 Appendix

7.1 Command Summary

Available commands below include identifying number, required parameters, the returned response and an indication of whether settings are remembered always, never, or with remember set to on.

Table 31: Communication Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Change Baud Rate	57	39	9	Byte	None	Always
Change I2C Slave Address	51	33	3	Byte	None	Always
Set a Non-Standard Baud Rate	164	A4	ñ	Short	None	Always
Turn Software Flow Control On	58	3A	:	Byte[2]	None	Remember On
Turn Software Flow Control Off	59	3B	;	None	None	Remember On

Table 32: Text Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Clear Screen	88	58	X	None	None	Never
Go Home	72	48	H	None	None	Never
Set Cursor Position	71	47	G	Byte[2]	None	Never
Set Cursor Coordinate	121	79	y	Byte[2]	None	Never
Auto Scroll On	81	51	Q	None	None	Remember On
Auto Scroll Off	82	52	R	None	None	Remember On

Table 33: Drawing Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Set Drawing Colour	99	63	c	Byte	None	Remember On
Draw Pixel	112	70	p	Byte[2]	None	Never
Draw a Line	108	6C	l	Byte[4]	None	Never
Continue a Line	101	65	e	Byte[2]	None	Never
Draw a Rectangle	114	72	r	Byte[5]	None	Never
Draw a Filled Rectangle	120	78	x	Byte[5]	None	Never
Initialize a Bar Graph	103	67	g	Byte[6]	None	Remember On
Draw a Bar Graph	105	69	i	Byte[2]	None	Never
Initialize a Strip Chart	106	6A	n	Byte[5], Short[2], Byte[2], Short	None	Remember On
Update a Strip Chart	107	6B	o	Byte, Short	None	Never

Table 34: Font Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Upload a Font File	36	24	$	Short, Integer, Byte[]	See Font File Creation	Always
Set the Current Font	49	31	1	Short	None	Never
Set Font Metrics	50	32	2	Byte[5]	None	Remember On
Set Box Space Mode	172	AC	¼	Byte	None	Remember On

Table 35: Bitmap Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Upload a Bitmap File	94	5E	^	Short, Integer, Byte[]	See Bitmap File Creation	Always
Draw a Bitmap from Memory	98	62	b	Short, Byte[2]	None	Never
Draw a Bitmap Directly	100	64	d	Byte[2], Byte[]	None	Never

Table 36: General Purpose Output Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
General Purpose Output On	86	56	V	Byte	None	Never
General Purpose Output Off	87	57	W	Byte	None	Never
Set Start Up GPO State	195	C3	├	Byte[2]	None	Always

Table 37: Keypad Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Auto Transmit Key Presses On	65	41	A	None	None	Remember On
Auto Transmit Key Presses Off	79	4F	`	None	None	Remember On
Poll Key Press	38	26	&	None	Byte	Never
Clear Key Buffer	69	45	E	None	None	Never
Set Debounce Time	85	55	U	Byte	None	Remember On
Auto Repeat Mode Off	96	60	`	None	None	Remember On
Assign Keypad Codes	213	D5	┌	Byte[25], Byte[25]	None	Always

Table 38: Display Functions Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Backlight On	66	42	B	Byte	None	Remember On
Backlight Off	70	46	F	None	None	Remember On
Set Brightness	153	99	Ö	Byte	None	Remember On
Set and Save Brightness	152	98	ÿ	Byte	None	Always
Set Contrast	80	50	P	Byte	None	Remember On
Set and Save Contrast	145	91	æ	Byte	None	Always

Table 39: Filesystem Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Delete Filesystem	33, 89, 33	21, 59, 21	!, Y, !	None	None	Always
Delete a File	173	AD	¡	Byte, Short	None	Always
Get Filesystem Space	175	AF	»	None	Integer	Never
Get Filesystem Directory	179	B3	│	None	Byte[][8]	Never
Filesystem Upload	176	B0	░	Integer, Byte[]	None	Always
Filesystem Download	48	30	0	None	Integer, Byte[]	Never
File Download	178	B2	▓	Byte, Short	Integer, Byte[]	Never
File Move	180	B4	┤	Byte, Integer, Byte, Integer	None	Always

Table 40: Data Security Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Set Remember	147	93	ô	Byte	None	Always
Set Data Lock	202, 245, 160	CA, F5, A0	╩, ⌡, á	Byte	None	Remember On
Set and Save Data Lock	203, 245, 160	CB, F5, A0	╦, ⌡, á	Byte	None	Always

Table 41: Miscellaneous Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Write Customer Data	52	34	4	Byte[16]	None	Always
Read Customer Data	53	35	5	None	Byte[16]	Never
Read Version Number	54	36	6	None	Byte	Never
Error! Reference source not found.	55	37	7	None	Byte	Never

7.2 Block Diagram

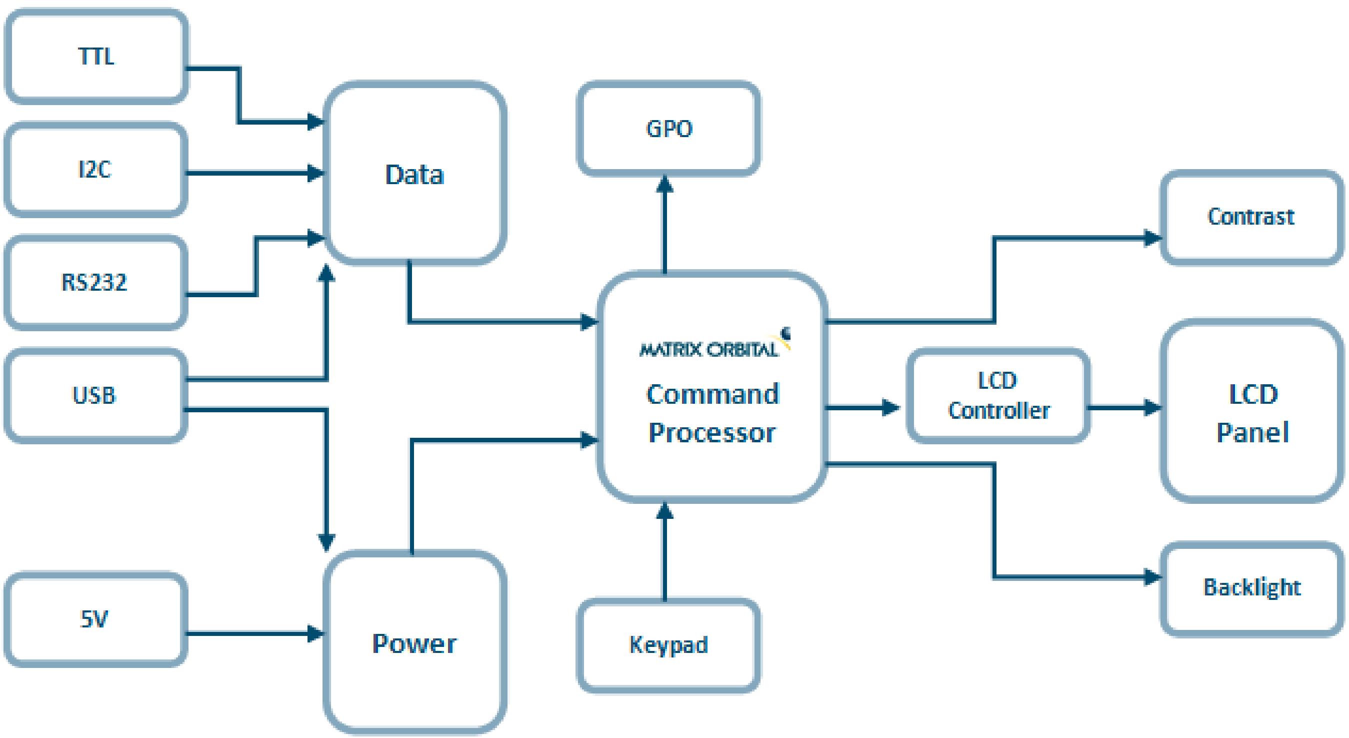


Figure 14: Functional Diagram

7.3 Environmental Specifications

Table 42: Environmental Limits

	Standard	*Extended (-E)
Operating Temperature	0°C to +50°C	-20°C to +60°C
Storage Temperature	-10°C to +60°C	-20°C to +70°C
Operating Relative Humidity	Maximum 90% non-condensing	

7.4 Electrical Tolerances

Current Consumption

Table 43: Current Consumption

Board		Backlight		GPOs
57mA	+	92mA	+	20mA each maximum

Table 44: Backlight Current Draw

Backlight
92mA

Input Voltage Specifications

Table 45: Voltage Specifications

Standard
4.75-5.25V

7.5 Dimensional Drawings

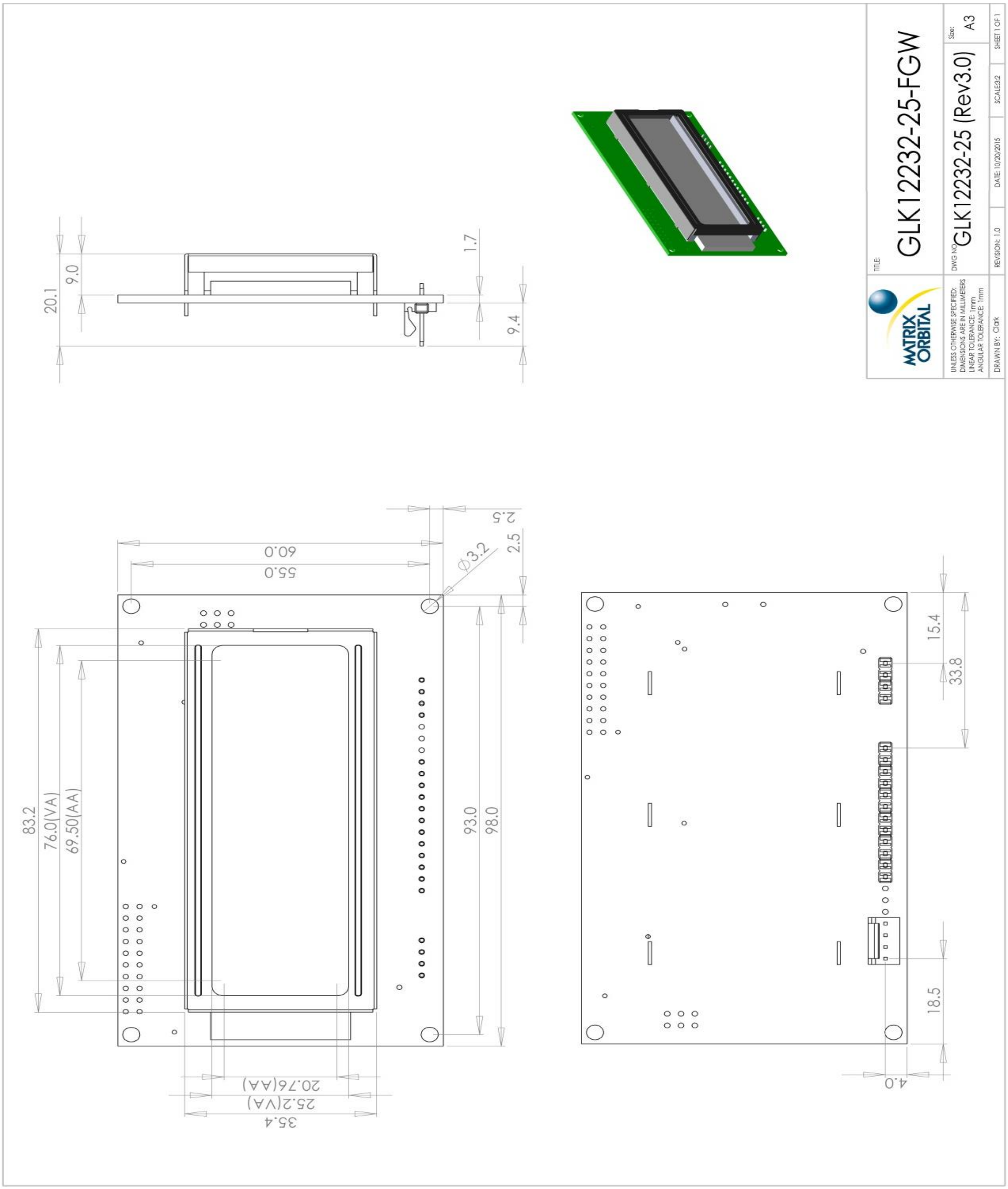


Figure 15: Standard Model Dimensional Drawing

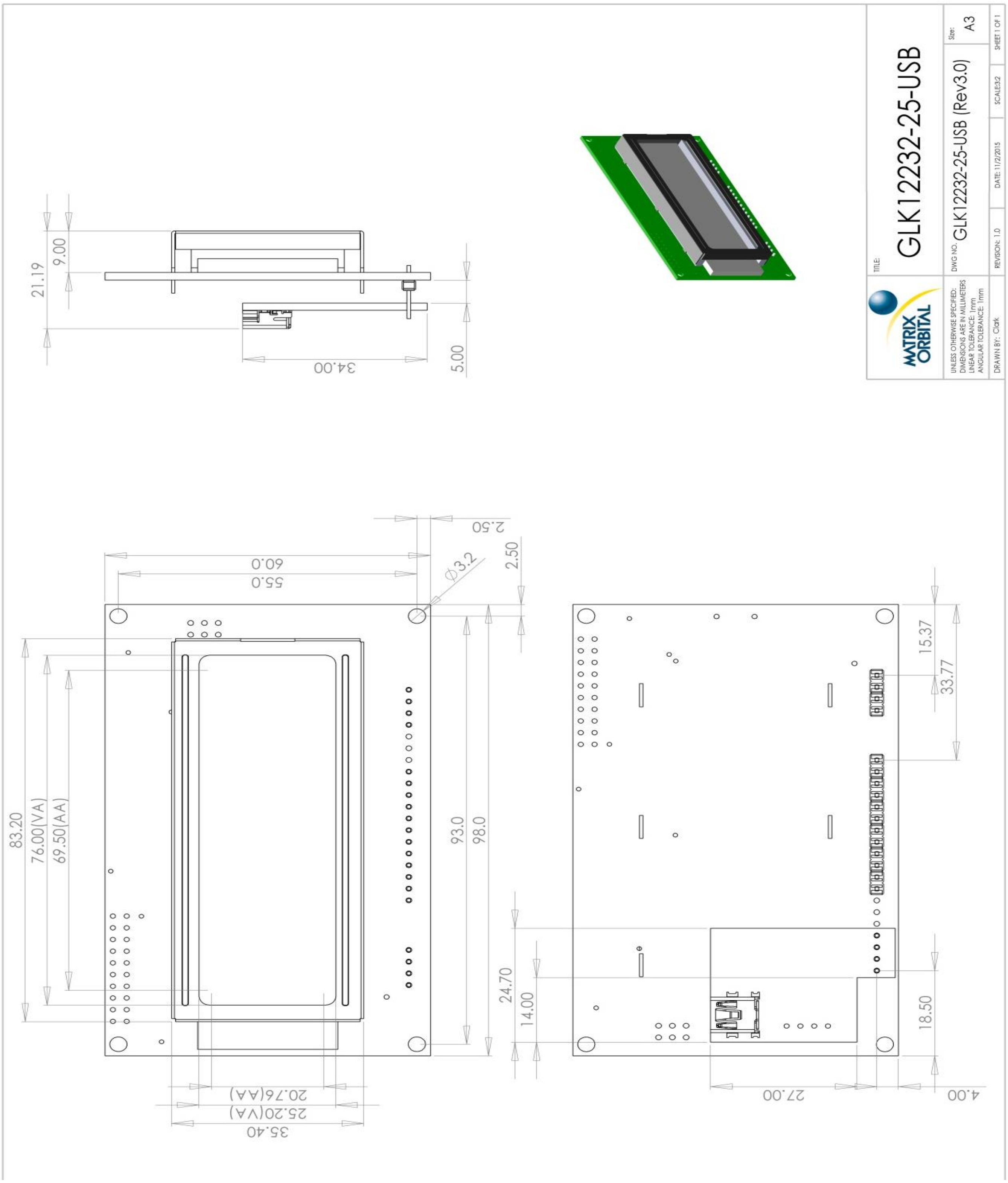


Figure 16: USB Model Dimensional Drawing

7.6 Optical Characteristics

Table 46: Display Optics

Module Size	98.0 x 60.0 x 20.1	mm
Viewing Area	76.0 x 25.2	mm
Active Area	69.50 x 20.76	mm
Pixel Size	0.52 x 0.62	mm
Pixel Pitch	0.53 x 0.53	mm
Viewing Direction	12	O'clock
Viewing Angle	-30 to +30	°
Contrast Ratio	3	
Backlight Half-Life	20,000	Hours

8 Ordering

8.1 Part Numbering Scheme

Table 47: Part Numbering Scheme

GLK	-12232	-25	-USB	-FGW	-E
1	2	3	5	6	7

8.2 Options

Table 48: Display Options

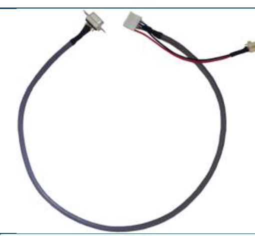
#	Designator	Options
1	Product Type	GLK: Graphic Liquid Crystal Display with Keypad Input
2	Display Size	12232: 122 pixel columns by 32 rows
3	Keypad Size	25: 25 key maximum
5	Protocol	*NP: Standard Model -USB: USB Only Model
6	Colour	FGW: Grey Text with White Background WB: White Text with Blue Background
7	Temperature	*NP: Standard -E: Extended Temperature

***Note:** NP means No Populate; skip this designator in the part number and move to the next option.

8.3 Accessories

Communication

Table 49: Communication Accessories

EXTMUSB3FT	Mini-USB Cable	
INTMUSB3FT	Internal Mini-USB Cable	
SCCPC5V	Extended Serial Communication/5V Power Cable	
BBC	Breadboard Cable	

Peripherals

Table 50: Peripheral Accessories

KPP4x4	16 Button Keypad	

9 Definitions

ASCII: American standard code for information interchange used to give standardized numeric codes to alphanumeric characters.

BPS: Bits per second, a measure of transmission speed.

Byte: An unsigned data packet that is eight bits long.

GPO: General purpose output, used to control peripheral devices from a display.

GUI: Graphical user interface.

Hexadecimal: A base 16 number system utilizing symbols 0 through F to represent the values 0-15.

I^2C: Inter-integrated circuit protocol uses clock and data lines to communicate short distances at slow speeds from a master to up to 128 addressable slave devices. A display is a slave device.

LSB: Least significant bit or byte in a transmission, the rightmost when read.

MSB: Most significant bit or byte in a transmission, the leftmost when read.

RS232: Recommended standard 232, a common serial protocol. A low level is -30V, a high is +30V.

SDA: Serial data line used to transfer data in I^2C protocol. This open drain line should be pulled high through a resistor. Nominal values are between 1K and 10K Ω.

SCL: Serial clock line used to designate data bits in I^2C protocol. This open drain line should be pulled high through a resistor. Nominal values are between 1K and 10K Ω.

Short: An unsigned data packet that is sixteen bits long, in little Endian format.

STN: Super-twisted nematic in reference to an LCD. In a relaxed or nematic state, crystals orientate themselves in the same direction and allow light to pass. In an excited state these crystals align to block light. Super-twisted crystals move from 180 to 270 degrees providing greater contrast than TN models.

TTL: Transistor-transistor logic applied to serial protocol. Low level is 0V while high logic is 5V.

10 Contact

Sales
Phone: 403.229.2737
Email: sales@matrixorbital.ca

Support
Phone: 403.204.3750
Email: support@matrixorbital.ca

Online
Purchasing: www.matrixorbital.com
Support: www.matrixorbital.ca