GLK24064-16-1U

Including GLK24064-16-1U and GLK24064-16-1U-USB

Technical Manual

Revision 1.1

PCB Revision: 1.0 or Higher

Firmware Revision: 7.0 or Higher

Revision History

Revision	Date	Description	Author
1.1	September 14, 2010	Added Filesystem Upload Command and Heading Numbering	Clark
1.0	August 3, 2010	Initial Release	Clark

Contents

1 Introduction

Figure 1: GLK24064-16-1U Display

The GLK24064-16-1U is an intelligent graphic liquid crystal 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, the USB communication model allows the GLK24064-16-1U to be connected to a wide variety of host controllers. Communication speeds of up to 115.2kbps for serial protocols and 100kbps for I^{2}C 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 sixteen kilobytes of customizable fonts and bitmaps to enhance the graphical user experience.

User input is available through a four by four matrix style keypad. One general purpose output on the standard model or six on the USB provide simple switchable five volt sources. The versatile GLK24064-16-1U, with all the features mentioned above, is available in a variety of colour, voltage, and temperature options to suit almost any application.

2 Quick Connect Guide

2.1 Standard Module
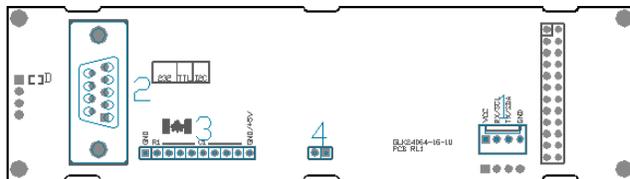
Figure 2: Standard Connections

#	Header	Mate
1	Power/Communication	SCCPC5V/BBC
2	DB9	CSS1FT/CSS4FT
3	Keypad	KPP4x4
4	GPO	None Offered

The standard version of the GLK24064-16-1U allows for user configuration of three 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^{2}C protocol. Connections for each protocol can be accessed through the four pin Communication/Power Header as outlined in the Serial Connections and I^{2}C Connections sections below.

Recommended Parts

Figure 3: Communication/Power Cable (SCCPC5V)

The most common cable choice for any standard Matrix Orbital display, the 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 4: Breadboard Cable (BBC)

For a more flexible interface to the GLK24064-16-1U, 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 GLK24064-16-1U. 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 three jumpers* in the 232 protocol box with the zero ohm jumper resistors provided or an alternate wire or solder solution.
   - TTL: Connect the two 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 four pin female header of the Communication/Power Cable to the Communication/Power Header of your GLK24064-16-1U.
   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 hyperterminal 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 GLK24064-16-1U 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 GLK24064-16-1U 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 GLK24064-16-1U 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.

2.2 USB Module
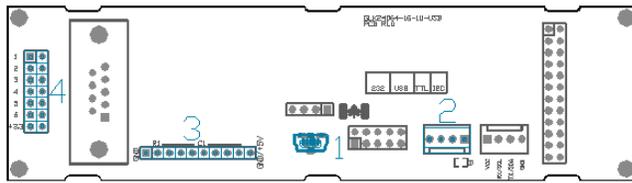
Figure 5: USB Connections

Table 2: Standard Headers

#	Header	Mate
1	Mini USB	EXTMUSB3FT/ INTMUSB3FT
2	Alternate Power	PCS
3	Keypad	KPP4x4
4	GPO	None Offered

The GLK24064-16-1U-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 GLK24064-16-1U-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 GLK24064-16-1U-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 GLK24064-16-1U-USB please follow the steps below.

1. Set the Protocol Select jumpers.
   - USB: The GLK24064-16-1U-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 the USB protocol. 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 yourGLK24064-16-1U-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 hyperterminal 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 GLK24064-16-1U 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. While many software programs are available to communicate with the GLK24064-16-1U, a number of more common samples are detailed in depth below.

Table 3: Reserved Control Characters

Control Characters	
10	Line feed / New line

3.1 Hyperterminal

Installed on older Windows computers, hyperterminal can be run by selecting run and typing 'hypertrm' in the command line. This basic program will allow communication between a PC and your display.

When starting up, a name must be given to your connection, and an icon may be chosen, neither is consequential. Next, it's important to select the appropriate communication port to which your display is connected. Finally, the settings below must be entered to complete the port setup.

Table 4: Hyperterminal Settings

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

Once a port is successfully set up, data can be sent to an attached display by typing on the keyboard. At this point, it may be helpful to echo keys to the monitor by selecting properties from the file menu and opening the ASCII settings from settings tab.

Commands can be sent to an attached display by issuing decimal commands using the number pad. While the ALT key is held down, four digit decimal values can be sent as a single ASCII character. For example, to clear the screen, try the following sequence.

ALT +0254 ALT +0088

Figure 7: Hyperterminal Command

Any commands or text desired can be sent to the communication port using this method to provide total control of any Matrix Orbital display.

3.2 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 monotone bitmaps can easily be created in virtually any image editing program, MOGD# provides an extensive font generation quite 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.
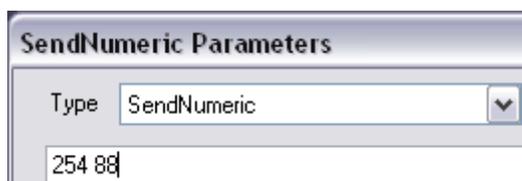
Figure 8: MOGD# Command Example

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.

3.3 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

Power/Communication Header

Figure 9: Communication/Power Header

Table 5: Communication/Power Pinout

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

The Communication/Power Header provides a standard connector for interfacing to the GLK24064-16-1U. Voltage is applied through pins one and four of the four pin Communication/Power Header. Please ensure the correct voltage input for your display by referencing the electrical specifications in Table 51 before connecting power. Pins two and three are reserved for serial transmission, using either the RS-232/TTL or clocking data through the I²C protocol, depending on what has been selected by the Protocol Select Jumpers. The versatile Tyco 640456-4-LF style header employed here can be mated to a wide array of female connectors for a perfect fit in any project.

Serial DB9 Connector
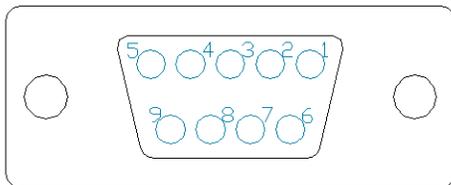
Figure 10: Serial DB9 Connector

Table 6: Serial DB9 Pinout

Pin	Function
5	Gnd
3	Rx
2	Tx
9	NC/Vcc*

The GLK24064-16-1U provides a DB-9 Connector to readily interface with serial devices using EIA232 standard signal levels. It is also possible to communicate at TTL levels of 0 to +5V by setting the Protocol Select Jumpers to TTL. As an added feature it is also possible to apply power through pin 9 of the DB-9 Connector in order to reduce cable clutter. A standard male DB9 header will provide the perfect mate for this connector.

*Note: Do not apply voltage through pin 9 of the DB-9 Connector AND through the Communication/Power Header at the same time.

Power Through DB9 Jumper

In order to provide power through pin 9 of the DB-9 Connector you must connect the Power Through DB-9 Jumper labelled D, as illustrated below. This connection can be made using a zero ohm resistor, recommended size 0603, or a solder bridge. The GLK24064-16-1U allows all voltage models to use the power through DB-9 option, see the specifications in Table 51 for voltage requirements.
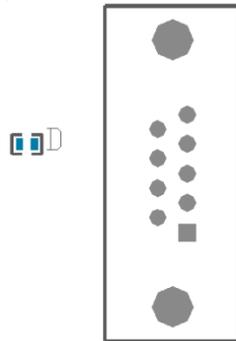
Power Through DB9 Jumper

Protocol Select Jumpers

The Protocol Select Jumpers provide the means necessary to toggle the GLK24064-16-1U between RS-232, TTL and I²C protocols. As a default, the jumpers are set to RS-232 mode with solder jumps on the 232 jumpers. In order to place the display module in I²C mode you must first remove the solder jumps from the 232 jumpers and then place them on the I^{2}C jumpers. The display will now be in I²C mode and have a default slave address of 0x50, unless it has been changed. Similarly, in order to change the display to TTL mode, simply remove the zero ohm resistors from the 232 or I²C jumpers and solder them to the TTL jumpers. Protocol tables are shown below where an `X` designates a connected jump while an 'O' signifies an open connection.

Table 7: RS232 Protocol Settings

RS232			TTL		I²C	
X	X	X	0	0	0	0

Table 8: TTL Protocol Settings

RS232			TTL		I²C	
O	O	O	X	X	O	O

Table 9: I^{2}C Protocol Settings

RS232			TTL		I²C	
O	O	O	O	O	X	X

4.2 USB Model

Mini USB Connector
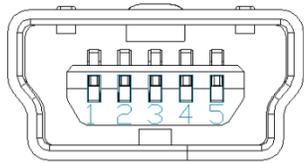
Figure 11: Mini USB Connector

Table 10: Mini USB Pinout

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

The GLK24064-16-1U-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 Power Connector
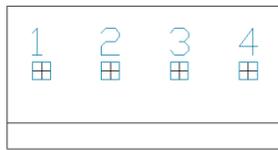
Figure 12: Alternate Power Connector

Table 11: Alternate Power Pinout

Pin	Function
1	NC
2	Gnd
3	Gnd
4	Vcc

The Alternate Power Connector provides the ability to power the GLK24064-16-1U-USB using a second cable. The Tyco 171825-4 style header is particularly useful for connecting to an unmodified floppy power cable from a PC power supply for a simple bench power solution.

4.3 Common Features

Keypad Header

Table 12: Keypad Pinout

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

Figure 13: Keypad Header

To facilitate user input, the GLK24064-16-1U provides a Keypad Interface Connector which allows matrix style keypad of up to sixteen 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, which is associated with the particular key press, is automatically sent on the Tx communication line. If the display module is running in I²C mode, the "Auto Transmit Keypress" function may 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 ten pin header of the Keypad Interface Connector will interface to a variety of different devices including the four by four Matrix Orbital keypad.

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

General Purpose Outputs

Figure 14: Standard Model GPO
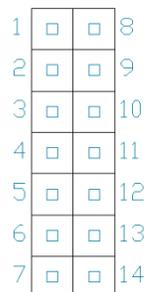
Figure 15: USB Model GPO

Table 13: GPO Pinout

Pin	Function	Pin	Function
1	GPO 1	8	Gnd
2	GPO 2	9	Gnd
3	GPO 3	10	Gnd
4	GPO 4	11	Gnd
5	GPO 5	12	Gnd
6	GPO 6	13	Gnd
7	Vcc	14	Gnd

A unique feature of the GLK24064-16-1U is the ability to control relays** and other external devices using either one or six General Purpose Outputs. Each can source up to 10mA of current at five volts when on or sink 20mA at zero volts when off. The two row, fourteen 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.

Troubleshooting

4.4 Power

In order for your Matrix Orbital display to function correctly, it must be supplied with the appropriate power. If the screen does not illuminate, power may not applied correctly. Try following the tips below.

- First, make sure that you are using the correct power connector. Standard floppy drive power cables from your PC power supply may fit on the Communication/Power Header; however they do not have the correct pin out to provide power. Matrix Orbital supplies power cable adapters for connecting to a PC, which can be found in the Accessories section.
- Next, 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.

4.5 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.

4.6 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 the GLK24064-16-1U to the transmitting pin from your host and the Tx pin to the receiving pin.
- If you are communicating to the display via I²C***** please ensure that the data is being sent to the correct address. The default slave address for the display module is 80.
- In I^{2}C 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: I²C communication will always require pull up resistors on SCL and SDA of one to ten kilohms.

4.7 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 next to the keypad header.
3. Reconnect power to your unit, and wait for the start screen before removing the override jumper.
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^{2}C Address	80

Table 14: Manual Override Settings

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

5 Commands

5.1 Communications

1.1 Changing the I2C Slave Address	Dec	**254 51**	**Address**
	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	1 byte, even value

1.2 Changing the Baud Rate	Dec	**254 57**	**Speed**
	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	1 byte, valid settings shown below

Table 15: Accepted Baud Rate Values

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

1.3 Setting a Non-Standard Baud Rate	Dec	**254 164**	**Speed**
	Hex	**FE A4**	**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	2 byte, calculations shown below

Equation 1: Speed Byte Calculation   *Equation 2: Actual Baud Rate Calculation*

Equation 3: Baud Rate Error Calculation

1.4 Transmission Protocol Select	Dec	**254 160**	**Protocol**
	Hex	**FE A0**	**Protocol**

Selects the protocol used for data transmission to the host. Data transmission from the host is not affected. Must be set to the protocol in use to receive data correctly.

Protocol	1 byte, 1 for Serial (RS232/RS422/TTL/USB) or 0 for I2C

1.5 Turn Flow Control On	Dec	**254 58**	**Full Empty**
	Hex	**FE 3A**	**Full Empty**
	ASCII	**■ :**	**Full Empty**

Enables simple flow control. The display will return a single byte (254) to the host when the display buffer is almost full and a different byte (255) 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 128 bytes. Not available in I^2C. Default off.

Full	1 byte, number of bytes remaining before buffer is completely full, 0 < Full < Empty < 128
Empty	1 byte, number of bytes remaining before buffer can be considered empty enough to accept more data

1.6 Turn Flow Control Off	Dec	**254 59**
	Hex	**FE 3B**
	ASCII	**■ ;**

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

5.2 Text

2.1 Auto Scroll On	Dec	254 81
	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.

2.2 Auto Scroll Off	Dec	254 82
	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.

2.3 Clear Screen	Dec	254 88
	Hex	FE 58
	ASCII	■ X

Clears the contents of the screen.

2.4 Set Cursor Position	Dec	254 71	Column Row
	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	1 byte, value between 1 and number of character columns
Row	1 byte, value between 1 and number of character rows

2.5 Set Cursor Coordinate	Dec	254 121	X Position Y Position
	Hex	FE 79	X Position Y Position
	ASCII	■ y	X Position Y Position

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

X Position	1 byte, value between 1 and screen width, represents leftmost character position
Y Position	1 byte, value between 1 and screen height, represents topmost character position

2.6 Go Home	Dec	254 72
	Hex	FE 48
	ASCII	■ H

Returns the cursor to the top left of the screen.

5.3 Fonts

3.1 Upload a Font File	Dec	**254 36**	**ID Size Data**
	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 Upload Protocol or XModem Upload Protocol entries.

ID	1 byte, unique font identification number, must be less than 128
Size	2 bytes, LSB followed by MSB, size of the entire font file
Data	variable length, font file data, see Font File Creation for example

3.2 Set the Current Font	Dec	**254 49**	**ID**
	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	1 byte, unique font identification number

3.3 Set Font Metrics	Dec	**254 50**	**Line Margin**	**Top Margin**	**Character Spacing**	**Line Spacing**	**Scroll Start**
	Hex	**FE 32**	**Line Margin**	**Top Margin**	**Character Spacing**	**Line Spacing**	**Scroll Start**
	ASCII	**■ 2**	**Line Margin**	**Top Margin**	**Character Spacing**	**Line Spacing**	**Scroll Start**

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

Line Margin	1 byte, space between left of display and first column of text. Default 0.
Top Margin	1 byte, space between top of display area and first row of text. Default 0.
Character Spacing	1 byte, space between characters. Default 0.
Line Spacing	1 byte, space between character rows. Default 1.
Scroll Start	1 byte, point at which text scrolls up screen to display additional rows. Default height-1.

3.4 Set Box Space Mode	Dec	**254 172**	**Switch**
	Hex	**FE AC**	**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	1 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 16: Example Font File Header

Maximum Width	Character Height	ASCII Start Value	ASCII End Value
5	7	72	74

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 17: 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 its 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 18: 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 19: 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 20: Example Font File

Header		5 7 72 74
Character Table	h	0 13 5
	i	0 18 3
	j	0 21 4
Character Data	h	132 45 152 198 32
	i	67 36 184
	j	16 49 25 96

5.4 Bitmaps

4.1 Upload a Bitmap File	Dec	**254 94**	ID	Size	Data
	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 Upload Protocol or XModem Upload Protocol entries.

ID	1 byte, unique bitmap identification number, must be less than 128
Size	2 bytes, width and height of the bitmap
Data	variable length, bitmap file data, see Bitmap File Creation example

4.2 Draw a Bitmap from Memory	Dec	**254 98**	ID	X Position	Y Position
	Hex	**FE 62**	ID	X Position	Y Position
	ASCII	**■ b**	ID	X Position	Y Position

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

ID	1 byte, unique bitmap identification number
X Position	1 byte, leftmost coordinate of bitmap
Y Position	1 byte, topmost coordinate of bitmap

4.3 Draw a Bitmap Directly	Dec	**254 100**	X Position	Y Position	Width	Height	Data
	Hex	**FE 64**	X Position	Y Position	Width	Height	Data
	ASCII	**■ d**	X Position	Y Position	Width	Height	Data

Draw a bitmap directly to the graphic display without saving to memory.

X Position	1 byte, leftmost coordinate of bitmap
Y Position	1 byte, topmost coordinate of bitmap
Width	1 byte, width of bitmap
Height	1 byte, height of bitmap
Data	bitmap dependent, see 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 only a single glyph, no header or table is required. 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 affect of the Matrix Orbital graphic stylization ability.

Table 21: Smiley Face Bitmap

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

Table 22: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

5.5 Drawing

5.1 Set Drawing Colour	Dec	**254 99**	Colour			
	Hex	**FE 63**	Colour			
	ASCII	**■ c**	Colour			

Change the drawing colour used for all subsequent drawing commands that do not implicitly specify colour.

Colour	1 byte, 0 for background or 1 to 255 for text colour

5.2 Draw Pixel	Dec	**254 112**	X Position	Y Position
	Hex	**FE 70**	X Position	Y Position
	ASCII	**■ p**	X Position	Y Position

Draw a single pixel on the graphic display using the current drawing colour.

X Position	1 byte, horizontal position of pixel, value between 0 and 239
Y Position	1 byte, vertical position of pixel, value between 0 and 63

5.3 Draw a Line	Dec	**254 108**	X1 Position	Y1 Position	X2 Position	Y2 Position
	Hex	**FE 6C**	X1 Position	Y1 Position	X2 Position	Y2 Position
	ASCII	**■ l**	X1 Position	Y1 Position	X2 Position	Y2 Position

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

X1 Position	1 byte, horizontal coordinate of first terminus, value between 0 and 239
Y1 Position	1 byte, vertical coordinate of first terminus, value between 0 and 63
X2 Position	1 byte, horizontal coordinate of second terminus, value between 0 and 239
Y2 Position	1 byte, vertical coordinate of second terminus, value between 0 and 63

5.4 Continue a Line	Dec	**254 101**	X Position	Y Position
	Hex	**FE 65**	X Position	Y Position
	ASCII	**■ e**	X Position	Y Position

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

X Position	1 byte, left coordinate of terminus, value between 0 and 239
Y Position	1 byte, top coordinate of terminus, value between 0 and 63

5.5 Draw a Rectangle	Dec	**254 114**	Colour	X1 Position	Y1 Position	X2 Position	Y2 Position
	Hex	**FE 72**	Colour	X1 Position	Y1 Position	X2 Position	Y2 Position
	ASCII	**■ r**	Colour	X1 Position	Y1 Position	X2 Position	Y2 Position

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

Colour	1 byte, 0 for background or 1 to 255 for text colour
X1 Position	1 byte, leftmost coordinate, value between 0 and 239
Y1 Position	1 byte, topmost coordinate, value between 0 and 63
X2 Position	1 byte, rightmost coordinate, value between X1 and 239
Y2 Position	1 byte, bottommost coordinate, value between Y1 and 63

5.6 Draw a Solid Rectangle	Dec	254 120	Colour X1 Position Y1 Position X2 Position Y2 Position
	Hex	FE 78	Colour X1 Position Y1 Position X2 Position Y2 Position
	ASCII	■ x	Colour X1 Position Y1 Position X2 Position Y2 Position

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

Colour	1 byte, 0 for background or 1to 255 for text colour
X1 Position	1 byte, leftmost coordinate, value between 0 and 239
Y1 Position	1 byte, topmost coordinate, value between 0 and 63
X2 Position	1 byte, rightmost coordinate, value between 0 and 239
Y2 Position	1 byte, bottommost coordinate, value between 0 and 63

5.7 Initialize a Bar Graph	Dec	254 103	ID Type X1 Position Y1 Position X2 Position Y2 Position
	Hex	FE 67	ID Type X1 Position Y1 Position X2 Position Y2 Position
	ASCII	■ g	ID Type X1 Position Y1 Position X2 Position Y2 Position

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 Bar Graph command below.

ID	1 byte, unique bar identification number, between 0 and 15
Type	1 byte, graph style, see Table 23
X1 Position	1 byte, leftmost coordinate, value between 0 and 239
Y1 Position	1 byte, topmost coordinate, value between 0 and 63
X2 Position	1 byte, rightmost coordinate, value between 0 and 239
Y2 Position	1 byte, bottommost coordinate, value between 0 and 63

Table 23: Bar Graph Types

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

5.8 Draw a Bar Graph	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 | 1 byte, unique bar identification number, between 0 and 15 |
| Value | 1 byte, portion of graph to fill in pixels, will not exceed display bounds |

<table>
<tr><td rowspan="3">5.9 Initialize a
Strip Chart</td><td>Dec</td><td>254 106</td><td>ID X1 Position Y1 Position X2 Position Y2 Position</td></tr>
<tr><td>Hex</td><td>FE 6A</td><td>ID X1 Position Y1 Position X2 Position Y2 Position</td></tr>
<tr><td>ASCII</td><td>■ j</td><td>ID X1 Position Y1 Position X2 Position Y2 Position</td></tr>
</table>

Designate a portion of the screen for horizontal scrolling. Can be used to create scrolling graphs or marquee text.

ID	1 byte, unique chart identification number, between 0 and 6
X1 Position	1 byte, leftmost coordinate, value between 0 and 239
Y1 Position	1 byte, topmost coordinate, value between 0 and 63
X2 Position	1 byte, rightmost coordinate, must be separated from 0 by a multiple of eight
Y2 Position	1 byte, bottommost coordinate, value between 0 and 63

<table>
<tr><td rowspan="3">5.10 Shift a
Strip Chart</td><td>Dec</td><td>254 107</td><td>Direction & ID</td></tr>
<tr><td>Hex</td><td>FE 6B</td><td>Direction & ID</td></tr>
<tr><td>ASCII</td><td>■ k</td><td>Direction & ID</td></tr>
</table>

Shift a designated strip chart area eight bits left or right. All text and fonts within the area are shifted.

Direction & ID	1 byte, MSB is direction, 0 for left or 1 for right, remaining bits indicate chart number

Table 24: Strip Chart Shift Example

Direction	ID	Byte	Value	Description
0	000 0001	01	1	Shift chart 1 left
1	000 0001	81	129	Shift chart 1 right

5.6 General Purpose Output

<table>
<tr><td rowspan="3">6.1 General Purpose
Output Off</td><td>Dec</td><td>254 86</td><td>Number</td></tr>
<tr><td>Hex</td><td>FE 56</td><td>Number</td></tr>
<tr><td>ASCII</td><td>■ V</td><td>Number</td></tr>
</table>

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

Number	1 byte, GPO to be turned off, value between 1 and 6

<table>
<tr><td rowspan="3">6.2 General Purpose
Output On</td><td>Dec</td><td>254 87</td><td>Number</td></tr>
<tr><td>Hex</td><td>FE 57</td><td>Number</td></tr>
<tr><td>ASCII</td><td>■ W</td><td>Number</td></tr>
</table>

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

Number	1 byte, GPO to be turned on, value between 1 and 6

<table>
<tr><td rowspan="2">6.3 Set Start Up
GPO State</td><td>Dec</td><td>254 195</td><td>Number State</td></tr>
<tr><td>Hex</td><td>FE C3</td><td>Number State</td></tr>
</table>

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

Number	1 byte, GPO to be controlled, value between 1 and 6
State	1 byte, 1 for on or 0 for off

5.7 Keypad

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

Key presses are automatically sent to the host when received by the display. Default is Auto Transmit on.

7.2 Auto Transmit Key Presses Off	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. Use this mode for I2C transactions. Default is Auto Transmit on.

7.3 Poll Key Press	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.

Response	1 byte, value of key pressed (MSB determines additional keys to be read)

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

Clears all key presses from the key buffer.

7.5 Set Debounce Time	Dec	254 85	Time
	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 a debounce time of approximately 52ms.

Time	1 byte, debounce increment (debounce time = Time * 6.554ms)

7.6 Set Auto Repeat Mode	Dec	**254 126**	**Mode**
	Hex	**FE 7E**	**Mode**

Sets key press repeat mode to typematic or hold. In typematic mode if a key press is held, 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	1 byte, 1 for hold mode or 0 for typematic

7.7 Auto Repeat Mode Off	Dec	**254 96**
	Hex	**FE 60**

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

7.8 Assign Keypad Codes	Dec	**254 213**	**Key Down Key Up**
	Hex	**FE D5**	**Key Down Key Up**

Assigns the key down and key up values sent to the host when a key press is detected. Defaults are shown below and can be reset by setting all values to 255.

Key Down	16 bytes, key down values
Key Up	16 bytes, key up values

Table 25: 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)

Table 26: 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)

5.8 Display Functions

8.1 Display On	Dec	254 66	Minutes
	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	1 byte, number of minutes to leave backlight on, a value of 0 leaves the display on indefinitely

8.2 Display Off	Dec	254 70
	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
	Hex	FE 99	Brightness

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

Brightness	1 byte, brightness level from 0(Dim) to 255(Bright)

8.4 Set and Save Brightness	Dec	254 152	Brightness
	Hex	FE 98	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	1 byte, brightness level from 0(Dim) to 255(Bright)

8.5 Set Backlight Colour	Dec	254 130	Red Green Blue
	Hex	FE 82	Red Green Blue

Immediately sets the backlight colour of a tricolour display. Output is a mix of the three colours and can be controlled with brightness and contrast commands. Default is 255, 255, 255, or white.

Red	1 byte, red brightness level from 0(Dim) to 255(Bright)
Green	1 byte, green brightness level from 0(Dim) to 255(Bright)
Blue	1 byte, blue brightness level from 0(Dim) to 255(Bright)

8.6 Set Contrast	Dec	254 80	Contrast
	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	1 byte, contrast level from 0(Light) to 255(Dark)

8.7 Set and Save Contrast	Dec	254 145	Contrast
	Hex	FE 91	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	1 byte, contrast level from 0(Light) to 255(Dark)

5.9 Filesystem

9.1 Wipe Filesystem	Dec	254 33 89 33
	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
	Hex	FE AD	Type ID

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

Type	1 byte, 0 for font or 1 for bitmap
ID	1 byte, unique identification number of font or bitmap to be deleted

9.3 Get Filesystem Space	Dec	254 175
	Hex	FE AF

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

Response	4 bytes, number of bytes remaining in 32KB memory, LSB to MSB

9.4 Get Filesystem Directory	Dec	254 179
	Hex	FE B3

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

Response	variable length, 1 byte representing number of entries followed by 4 identification bytes for each entry

Table 27: Filesystem Identification Bytes

Byte	3	2	1	0
Description	Size MSB	Size LSB	Type/ID	Flag

Table 28: Extended Byte Descriptions

Size MSB	most significant byte of the file size
Size LSB	least significant byte of the file size
Type/ID	MSB designates file type, 0 for font or 1 for bitmap, remaining bits indicate ID number
Flag	a value of 0 indicates entry is not in use

9.5 Filesystem Upload	Dec	254 176	Size Data		
	Hex	FE B0	Size Data		

This command will upload a filesystem image to the display. The size used is almost always the entire 16kB memory.   Filesystem data can be uploaded LSB to MSB in the same manner as a font or bitmap file.

Size	4 bytes, size of the filesystem to upload, LSB to MSB, usually 0x00 0x40 0x00 0x00 for 16kB
Data	variable length, data to upload

9.6 Download a File	Dec	254 178	Type ID
	Hex	FE B2	Type ID

Downloads a single font or bitmap file from the display to the host.

Type	1 byte, 0 for font or 1 for bitmap
ID	1 byte, unique identification number of font or bitmap to download
Response	variable length, first 2(font) or 4(bitmap) bytes represent file size followed by file data

9.7 Move a File	Dec	254 180	Old Type	Old ID	New Type	New ID
	Hex	FE B4	Old Type	Old ID	New Type	New ID

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.

Old Type	1 byte, original file type, 0 for font or 1 for bitmap
Old ID	1 byte, original unique file identification number
New Type	1 byte, new file type, 0 for font or 1 for bitmap
New ID	1 byte, new unique file identification number, must be less than 128

9.8 Dump the Filesystem	Dec	254 48
	Hex	FE 30
	ASCII	∎ 0

Downloads complete filesystem containing all fonts and bitmaps stored in the display. A veritable heap of data.

Response	16388 bytes, 4 bytes of size LSB to MSB followed by entire 16KB filesystem

File Upload 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 29: Upload Protocol Responses

Value	Action	Description
1	Confirm	Transfer successful, upload continues
8	Decline	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 30: Font Upload Protocol

Host	Display	Comments
254		Command Prefix
36		Upload Font File Command
1		Reference ID
31		Font File LSB
0		Font File MSB
	1	Size Confirmation
5		First Font Data Byte
	5	Echo Data Byte
1		Confirm Data Byte
...	...	...
96		Last Font Data Byte
	96	Echo Data Byte
1		Confirm Data Byte

Table 31: Bitmap Upload Protocol

Host	Display	Comments
254		Command Prefix
94		Upload Bitmap File Command
1		Reference ID
5		Bitmap File LSB
0		Bitmap File MSB
	1	Size Confirmation
5		First Bitmap Data Byte
	5	Echo Data Byte
1		Confirm Data Byte
...	...	...
224		Last Bitmap Data Byte
	224	Echo Data Byte
1		Confirm 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.

XModem Upload Protocol

In addition to its original simple upload format, Matrix Orbital has added an XModem based protocol. This facilitates much faster download speeds by increasing the packet size from 1 byte to 128 bytes greatly increasing throughput. Though a protocol similar to the original upload scheme is used, a two byte CRC check is preformed at the end of each packet in place of the byte echo system. To begin the upload, a series of command bytes are sent, much however, no distinction is made between bitmap and font as the XModem protocol is used to upload bin or ebin files that contain all the bitmaps and fonts required for the unit. Once the command bytes are sent, the size of the file is sent in two bytes, least significant byte first. Then two additional bytes are sent of the value zero. At this point the display will respond with an ACK if the file fits or a NAK otherwise. Please note that these values are different than those of the original protocol as seen in the table below. If a NAK is seen at any point by the host, the upload is to be aborted in the same fashion as the regular protocol. If the file will fit, the start of header byte will be sent by the host, followed by a block count, in regular and inverted format, representing the number of 128 byte blocks remaining to. The display will then check to make sure the block count value matches its own before ACKing. The host can then send a 128 byte block of data followed by that blocks high and low CRC16 bytes. The display then performs a CRC check on the data receive and ACKs if it matches that which was sent. Transfer continues with a block count and continues in this way until the end of file is reached. Once the end of the upload file is reached, the host should transmit a single end of transmission byte. If the end of file is expected, the display will ACK one last time. This EOT byte along with the other special characters mentioned above is listed in the table below.

Table 32: XModem Upload Protocol:

Host	Display	Comments
254		Command Prefix
219		XModem Upload Command
133		Command Byte One
6		Command Byte Two
48		Command Byte Three
0		Size LSB
64		Size MSB
0		Zero
0		Zero
	6	ACK (NAK if file is too big)
1		Start of Header
128		Block Count
127		Inverted Block Count (255-Count)
	6	ACK (NAK if counts don't match)
<128 B>		128 Byte Data Block
30		CRC MSB
71		CRC LSB
	6	ACK (NAK if CRCs don't match)
…	…	…
4		End of Transmission
	6	ACK (NAK if EOT is not expected)

Table 33: XModem Protocol Message Bytes

Value	Action	Description
6	Acknowledged	Transfer successful, upload continues
33	Not Acknowledged	Transfer failed, upload aborted
1	Start of Header	Begin upload transfer
4	End of Transmission	End completed upload transfer

5.10 Data Security

10.1 Set Remember	Dec	254 147	Switch
	Hex	FE 93	Switch

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 approximately 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.

Switch	1 byte, 1 for on or 0 for off

10.2 Set Data Lock	Dec	254 202 245 160	Level
	Hex	FE CA F5 A0	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	1 byte, each bit representing a level, see Table 34

Table 34: Data Lock Bits

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

Table 35: Lock Parameters

Reserved	Place holders only, should be 0
Address	Locks the Baud Rate and I^{2}C 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
	Hex	FE CB F5 A0	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	1 byte, see data lock table

5.11 Miscellaneous

11.1 Write Customer Data	Dec	**254 52**	Data
	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	16 bytes, user defined data

11.2 Read Customer Data	Dec	**254 53**
	Hex	**FE 35**
	ASCII	**■ 5**

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

Response	16 bytes, previously saved user defined data

11.3 Read Version Number	Dec	**254 54**
	Hex	**FE 36**
	ASCII	**■ 6**

Causes display to respond with its firmware version number.

Response	1 byte, convert to hexadecimal to view major and minor revision numbers

11.4 Read Module Type	Dec	**254 55**
	Hex	**FE 37**
	ASCII	**■ 7**

Causes display to respond with its module number.

Response	1 byte, module number, see partial list below

Table 36: Sample Module Type Responses

25	GLK24064-16-1U-USB	26	GLK24064-16-1U

6 Appendix

6.1 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 37: Communication Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Changing the I2C Slave Address	51	33	3	Address	None	Always
Changing the Baud Rate	57	39	9	BaudRate	None	Always
Setting a Non-Standard Baud Rate	164	A4	ñ	Speed	None	Always
Transmission Protocol Select	160	A0	á	Protocol	None	Remember On
Turn Flow Control On	58	3A	:	Full, Empty	None	Remember On
Turn Flow Control Off	59	3B	;	None	None	Remember On

Table 38: Text Command Summary

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

Table 39: Font Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Upload a Font File	36	24	$	ID, Size [2], Data []	See Font File Creation	Always
Set the Current Font	49	31	1	ID	None	Remember On
Set Font Metrics	50	32	2	LineMargin, TopMargin, CharSpace, LineSpace, ScrollStart	None	Remember On
Set Box Space Mode	172	AC	¼	Switch	None	Remember On

Table 40: Bitmap Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Upload a Bitmap File	94	5E	^	ID, Size [2], Data []	See Bitmap File Creation	Always
Draw a Bitmap from Memory	98	62	b	ID, X, Y	None	Never
Draw a Bitmap Directly	100	64	d	X, Y, Width, Height, Data []	None	Never

Table 41: Drawing Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Set Drawing Colour	99	63	c	Colour	None	Remember On
Draw Pixel	112	70	p	X, Y	None	Never
Draw a Line	108	6C	l	X1, Y1, X2, Y2	None	Never
Continue a Line	101	65	e	X, Y	None	Never
Draw a Rectangle	114	72	r	Colour, X1, Y1, X2, Y2	None	Never
Draw a Solid Rectangle	120	78	x	Colour, X1, Y1, X2, Y2	None	Never
Initialize a Bar Graph	103	67	g	ID, Type, X1, Y1, X2, Y2	None	Remember On
Draw a Bar Graph	105	69	i	ID, Value	None	Never
Initialize a Strip Chart	106	6A	j	ID, X1, Y1, X2, Y2	None	Remember On
Shift a Strip Chart	107	6B	k	DirectionID	None	Never

Table 42: General Purpose Output Command Summary

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

Table 43: 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	O	None	None	Remember On
Poll Key Press	38	26	&	None	KeyPress	Never
Clear Key Buffer	69	45	E	None	None	Never
Set Debounce Time	85	55	U	Time	None	Remember On
Set Auto Repeat Mode	126	7E	~	Mode	None	Remember On
Auto Repeat Mode Off	96	60	`	None	None	Remember On
Assign Keypad Codes	213	D5	⌐	KeyUp [16], KeyDown [16]	None	Always

Table 44: Display Functions Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Display On	66	42	B	Minutes	None	Remember On
Display Off	70	46	F	None	None	Remember On
Set Brightness	153	99	Ö	Brightness	None	Remember On
Set and Save Brightness	152	98	ÿ	Brightness	None	Always
Set Backlight Colour	130	82	'	Red, Green, Blue	None	Remember On
Set Contrast	80	50	P	Contrast	None	Remember On
Set and Save Contrast	145	91	æ	Contrast	None	Always

Table 45: Filesystem Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Wipe Filesystem	33, 89, 33	21, 59, 21	!, Y, !	None	None	Always
Delete a File	173	AD	¡	Type, ID	None	Always
Get Filesystem Space	175	AF	»	None	Space [4]	Never
Get Filesystem Directory	179	B3	¦	None	Entries []	Never
Filesystem Upload	176	B0	°	Size [4], Data[]	None	Always
Download a File	178	B2	▓	Type, ID	Data []	Never
Move a File	180	B4	⊣	Old Type, Old ID, New Type, New ID	None	Always
Dump the Filesystem	48	30	0	None	Size [4], Data [32768]	Never

Table 46: Data Security Command Summary

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

Table 47: Miscellaneous Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Write Customer Data	52	34	4	Data [16]	None	Always
Read Customer Data	53	35	5	None	Data [16]	Never
Read Version Number	54	36	6	None	Version	Never
Read Module Type	55	37	7	None	Module	Never

6.2 Environmental Specifications

Table 48: Environmental Limits

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

6.3 Electrical Tolerances

Current Consumption

Table 49: Current Consumption

Board		Backlight		GPOs
45mA	+	75 to 205 mA	+	20mA each maximum

Table 50: Backlight Current Draw

GW & WB	TC & TCI
75mA	205mA

Input Voltage Specifications

Table 51: Voltage Specifications

Low Voltage (-LV)	Standard	Wide Voltage (-V)	Extended Wide Voltage (-VPT)
3.3V	4.75-5.25V	9.0-15.0V	9.0-35.0V

6.4 Optical Characteristics

Table 52: Display Optics

| Module Size	112.00 x 38.00 x 28.9	mm
Viewing Area	98.0 x 28.4	mm
Active Area	93.57 x 24.93	mm
Pixel Size	0.36 x 0.36	mm
Pixel Pitch	0.39 x 0.39	mm
Viewing Direction	12	O'clock
Viewing Angle	-30 to +30	°
Contrast Ratio	3	
Backlight Half-Life	20,000	Hours

6.5 Dimensional Drawings
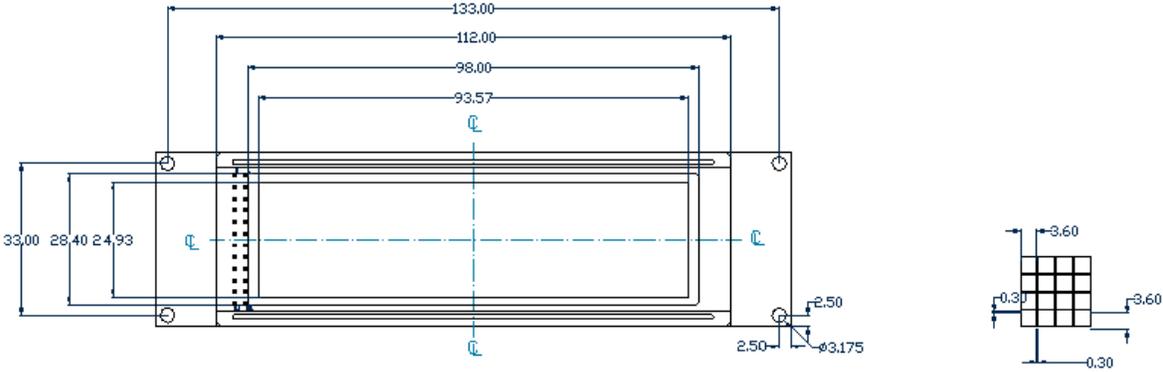
*Figure 16: Display Dimensional Drawing*
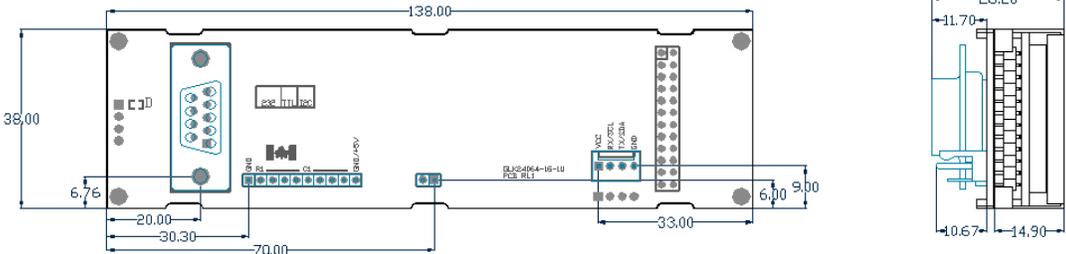
Figure 17: Standard Model Dimensional Drawing
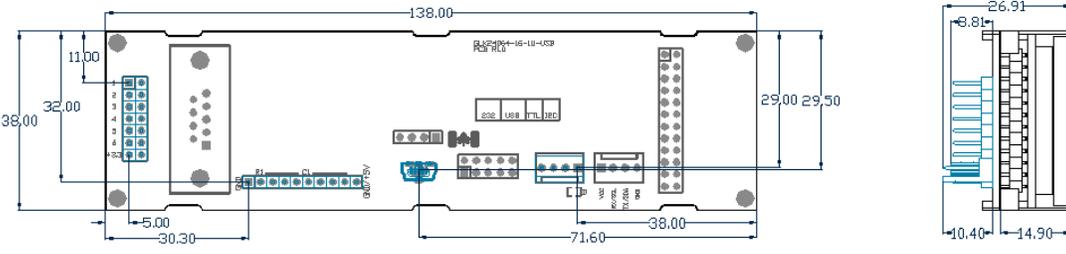
Figure 18: USB Model Dimensional Drawing

7 Ordering

7.1 Part Numbering Scheme

Table 53: Part Numbering Scheme

GLK	-24064	-16	-1U	-USB	-FGW		-E
1	2	3	4	5	6	7	8

7.2 Options

Table 54: Display Options

#	Designator	Options
1	Product Type	GLK: Graphic Liquid Crystal Display with Keypad Input
2	Display Size	24064: 240 pixel columns by 64 rows
3	Keypad Size	16: 16 key maximum
4	Form Factor	1U: Designed to 1U, or PC bay insert, dimensions
5	Protocol	NP: Standard Model -USB: USB Only Model
6	Colour	FGW: Grey Text with White Background WB: White Test with Blue Background *TCI: Tricolour Text with Black Background
7	Voltage	NP: Standard Voltage -LV: Low Voltage -V: Wide Voltage -VPT: Wide Voltage with Efficient Switching Power Supply
8	Temperature	NP: Standard -E: Extended Temperature

Note: Tricolour options are available for the Standard, Non-USB model only.

7.3 Accessories

Power

Table 55: Power Accessories

PCS	Standard Power Cable	

Communication

Table 56: Communication Accessories

CSS4FT	4 ft. Serial Cable	
EXTMUSB3FT	Mini-USB Cable	
INTMUSB3FT	Internal Mini-USB Cable	
SCCPC5V	Serial Communication/5V Power Cable	
BBC	Breadboard Cable	
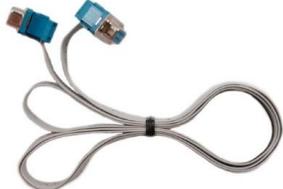
Peripherals

KPP4x4	16 Button Keypad	

Mounting

B24064-BK	24064-1U Black Mounting Bracket	

8 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.

FFSTN: Double film super-twisted nematic in reference to an LCD. The addition of two layers of film between the STN display and polarizer improves contrast.

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

GUI: Graphical user interface.

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

I^2C: Inter-integrated circuit protocol employing a clock and data line to communicate a short distance at slow speeds between a master and up to 128 addressable slave devices. A Matrix Orbital 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 2K 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 2K and 10K Ω.

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.

USB: Universal Serial Bus protocol widely used in PCs.

9 Contact

Sales	Support	Online
Phone: 403.229.2737	Phone: 403.204.3750	Purchasing: www.matrixorbital.com
Email: sales@matrixorbital.ca	Email: support@matrixorbital.ca	Support: www.matrixorbital.ca