GLK24064-25/GLT24064

Technical Manual

Revision 2.2

PCB Revision: 4.0 or Higher

Firmware Revision: 8.1 or Higher

Revision History

Revision	Date	Description	Author
2.2	March 8, 2011	Updated Electrical Specifications	Clark
2.1	January 27, 2011	Filesystem Command Updates for Firmware Revision 8.1	Clark
2.0	November 3, 2010	Initial Release	Clark

Contents

1 Introduction

Figure 1: GLK24064-25/GLT24064 Display

The GLK24064-25/GLT24064 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, USB and RS422 communication models allow the GLK24064-25/GLT24064 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 thirty-two kilobytes of customizable fonts and bitmaps to enhance the graphical user experience.

User input on the GLK24064-25 is available through a five by five matrix style keypad or a resistive touch overlay on the GLT24064. Six general purpose outputs provide simple switchable five volt sources on each model. In addition, a versatile Dallas One-Wire header provides a communication interface for up to thirty-two devices.

The versatile GLK24064-25/GLT24064, 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 Available Headers
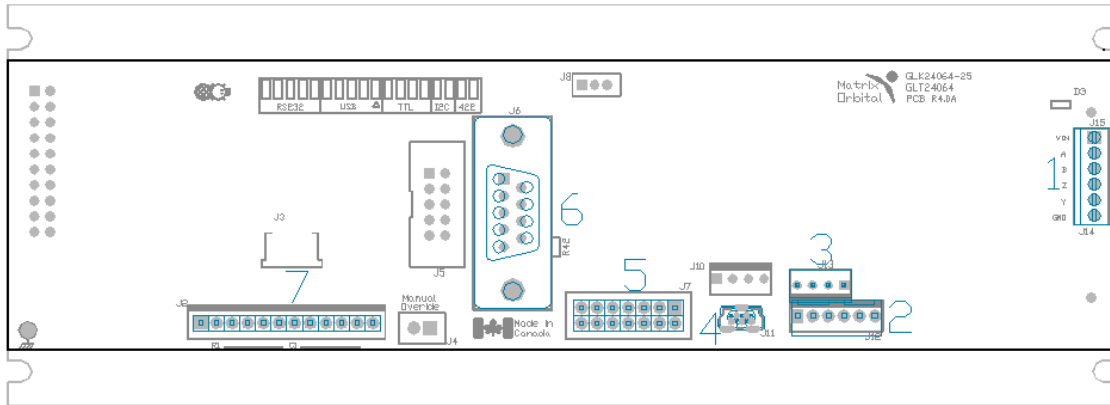
Figure 2: GLK24064-25/GLT24064 Header Locations

Table 1: List of Available Headers

#	Header	Mate	Population
1	RS422 Terminal Block	16-30 AWG Wire	422 Model Only
2	Extended Communication/Power Connector	ESCCPC5V/BBC	Standard Model Only
3	Alternate Power Connector	PCS	422 and USB Models Only
4	Mini USB Connector	EXTMUSB3FT/INTMUSB3FT	USB Model Only
5	GPO Header	None Offered	All Models
6	DB9 Serial Header	CSS1FT/CSS4FT	Standard Model Only
7	Keypad/Touchpad	KPP4x4/Touch Panel	GLK/GLT Model Only

2.2 Standard Module

The standard version of the GLK24064-25/GLT24064 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

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 3: Extended Communication/Power Cable (ESCCPC5V)

For a more flexible interface to the GLK24064-25/GLT24064, 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.

Figure 4: Breadboard Cable (BBC)

2.3 Serial Connections

Serial protocol provides a classic connection to the GLK24064-25/GLT24064. The Extended 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 Extended Communication/Power Cable to the Communication/Power Header of your GLK24064-25/GLT24064.
    b. Insert the male end of your serial cable to the corresponding DB9 header of the Extended 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 Extended 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 these programs can be found below and a variety of application notes are available at www.matrixorbital.ca/appnotes.

I^{2}C Connections

A more advanced connection to the GLK24064-25/GLT24064 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 GLK24064-25/GLT24064 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 GLK24064-25/GLT24064 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.

2.4 USB Module

The GLK24064-25-USB/GLT24064-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

The External Mini USB cable is recommended for the GLK24064-25-USB /GLT24064-USB display. It will connect to the mini-B style header on the unit and provide a connection to a regular A style USB connector, commonly found on a PC.

Figure 5: Mini USB Cable (EXTMUSB3FT)

USB Connections

The USB connection is the quickest, easiest solution for PC development. After driver installation, the GLK24064-25-USB/GLT24064-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 GLK24064-25-USB/GLT24064-USB, please follow the steps below.

1. Set the Protocol Select jumpers.
   * USB: The GLK24064-25-USB/GLT24064-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 your GLK24064-25-USB/GLT24064-USB and the regular USB header into your computer USB jack.   Additional power may be required, especially for YG displays, and can be supplied to the Alternate Power Connector.
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 these programs can be found below and a number of application notes are available at www.matrixorbital.ca/appnotes.

2.5 RS422 Module

The GLK24064-25-422/GLT24064-422 provides an industrial alternative to the standard RS232 communication protocol. Rather than single receive and transmit lines, the RS422 model uses a differential pair for each of the receive and transmit signals to reduce degradation and increase transmission lengths. Power can be transmitted at distance to a -VPT module or supplied from the immediate vicinity to a regular unit. RS422 signals are available in a six pin connector as described in the RS422 Connections section.

RS422 Connections

The GLK24064-25-422/GLT24064-422 provides a robust RS422 interface to the display line. For this interface, a series of six wires are usually screwed into the RS422 terminal block provided. An alternate header is also available to provide local power to a regular unit. To connect to your GLK24064-25-422/GLT24064-422, adhere to the steps laid out below.

1. Set the Protocol Select jumpers.
    - RS422: The GLK24064-25-422/GLT24064-422 offers only RS422 protocol and does not require any jumper changes.
2. Make the connections.
    a. Screw one wire; sized 16 to 30 on the American Wire Gauge, into each of the six terminal block positions. When local power is supplied, a floppy cable may link to the alternate power header.
    b. Connect the Vcc wire to the positive terminal of your power supply and the GND terminal to the negative or ground lead to provide appropriate power as in Table 62.
    c. Secure the A and B wires to your non-inverting and inverting output signals respectively, while attaching the Z and Y wires to your inverting and non-inverting inputs.
3. Create.
    - In a PC environment, MOGD# or hyperterminal will serve to get you started. In addition, a variety of application notes are available online in a number of different languages to aid in the development of a host controller. Instructions for these programs can be found below and the simple C# example at www.matrixorbital.ca/appnotes is a great first programming reference.

3 Software

The multiple communication protocols available and simple command structure of the GLK24064-25/GLT24064 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	
10	Line feed / New line

Once the correct communication port is identified, the following communication settings can be applied to communicate correctly with the GLK24064-25/GLT24064.

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 monotone 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.
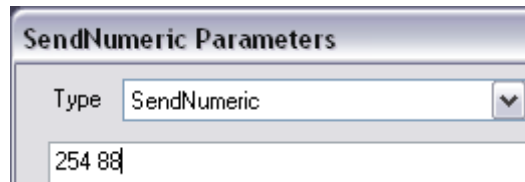
Figure 6: MOGD# Command Example

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.

3.1 Firmware Upgrade

Beginning with revision 8.1, the firmware of the GLK24064-25/GLT24064 can be upgraded in the field. Alternatively, the changes to the filesystem and subsequent commands can also be reverted by downgrading the firmware to revision 8.0 using the same process.

All firmware revisions of the GLK24064-25/GLT24064 can be downloaded from www.matrixorbital.ca/software/GLTSeries and installed using MOGD#.

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

Extended Communication/Power Header
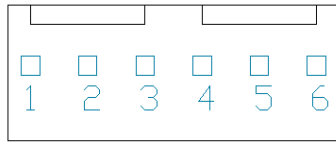
Figure 7: Extended Communication/Power Header

Table 4: Extended Communication/Power Pinout

Pin	Function
1	Vcc
2	Rx (SCL)
3	Tx (SDA)
4	Gnd
5	RTS
6	CTS

The Extended Communication/Power Header provides a standard connector for interfacing to the GLK24064-25/GLT24064. 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 62 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. Pins five and six can be used for serial transmission hardware flow control, and are ignored for I²C communications. The Molex 22-04-1061 style header used can be mated to a number of connectors.

Serial DB9 Connector
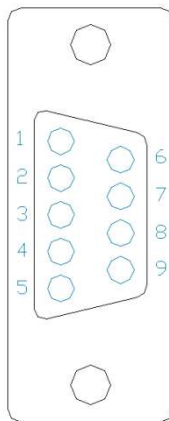
Figure 8: Serial DB9 Connector

Table 5: Serial DB9 Pinout

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

The GLK24064-25/GLT24064 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 R42, as illustrated below. This connection can be made using a zero ohm resistor, recommended size 0603, or a solder bridge. The GLK24064-25/GLT24064 allows all voltage models to use the power through DB-9 option, see the specifications in Table 62 for voltage requirements.
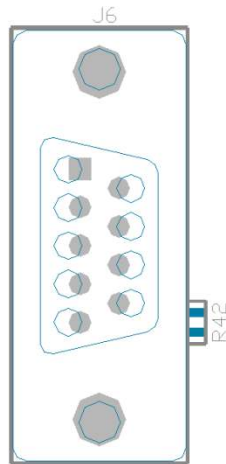
Figure 9: Power Through DB9 Jumper

Alternate Serial Header

Some advanced applications may prefer the straight two by five pin connection offered through the optional Alternate Serial Header. This header offers power and communication access in a simple interface package. The Alternate Serial Header may be added to the GLK24064-25/GLT24064 Standard model for an added charge as part of a custom order. Please Contact sales for details.

Protocol Select Jumpers

The Protocol Select Jumpers provide the means necessary to toggle the standard GLK24064-25/GLT24064 model 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.

Hardware Lock

The Hardware Lock allows fonts, bitmaps, and settings to be saved, unaltered by any commands. By connecting the two pads near the memory chip, designated as R13, 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.

4.2 USB Model

Mini USB Connector
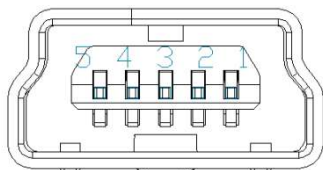
Figure 10: Mini USB Connector

Table 6: Mini USB Pinout

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

The GLK24064-25-USB/GLT24064-USB comes with a familiar Mini USB Connector to fulfill both communication and power needs. The standard Mini-B 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 GLK24064-25-USB/GLT24064-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.

Alternate Power Connector
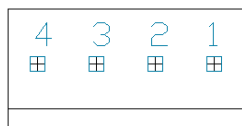
Figure 11: Alternate Power Connector

Table 7: Alternate Power Pinout

Pin	Function
1	Vcc
2	Gnd
3	Gnd
4	NC

The Alternate Power Connector provides the ability to power the GLK24064-25-USB/GLT24064-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 RS422 Model

RS422 Header
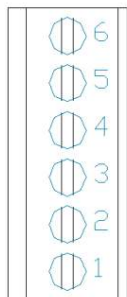
Figure 12: RS422 Header

Table 8: RS422 Pinout

Pin	Function
1	Gnd
2	Rx (Y)
3	Inv Rx (Z)
4	Inv Tx (B)
5	Tx (A)
6	Vcc

The six pin RS422 interface header of the GLK24064-25-422/GLT24064-422 offers power and ground connections as well as two differential pair communication lines. Regular and inverted lines are provided for both receive and transmit signals. Power is supplied locally to the regular variant while the –VPT can receive power over a distance. The Tyco 282834-6 style header is most suited to a simple wire connection.

Alternate Power Connector
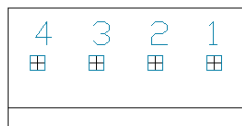
Figure 13: Alternate Power Connector

Table 9: Alternate Power Pinout

Pin	Function
1	Vcc
2	Gnd
3	Gnd
4	NC

The Alternate Power Connector provides the ability to power the GLK24064-25-422/GLT24064-422 using a second cable. This is particularly useful for the regular module that is to be powered locally. The Tyco 171825-4 style header will fit a floppy power cable from a PC power supply for a simple bench power solution.

4.4 GLK Model

Keypad Header
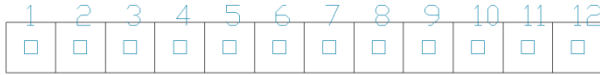
Figure 14: Keypad Header

Table 10: 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 GLK24064-25 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 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 twelve pin header of the Keypad Interface Connector will interface to a variety of different devices including the Matrix Orbital KPP4x4 keypad.

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

4.5 GLT Model

Touch Screen

The GLT24064 facilitates user touch input in one of two distinct ways. Coordinate mode will report events by supplying their exact position on the screen. Region mode will report events within defined boundaries on the screen. Both modes are outlined below.

Coordinate Mode
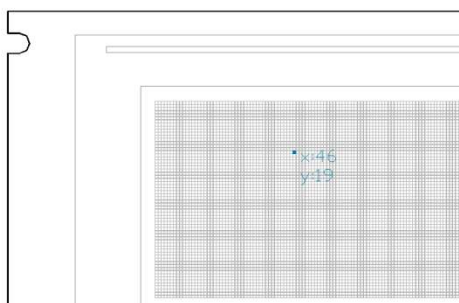
In coordinate mode all touch events are reported using three single byte values. First, the type of event is transmitted, followed by the x and y coordinates of its position. Pressure and drag thresholds must be exceeded for an event to be registered. A low drag threshold will result in greater tracking accuracy but transmits much more data to the host. Care should be taken to find balance. This mode offers a great degree of flexibility and creativity.

Table 11: Coordinate Mode Event Prefixes

Return Value	1	2	4
Touch Event	Press	Release	Drag

Region Mode
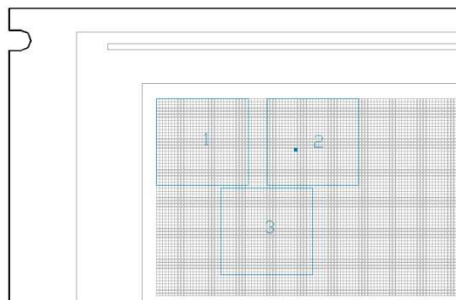
A simpler, keypad style alternative to coordinate mode, region mode offers only a single byte for each touch event. Unique regions are created by specifying a position, size, and return values. A value corresponding to a specific region is returned when an event occurs within its bounds. Events outside of regions result in transmission of the value 255. Regions can be deleted individually or collectively when no longer needed. This mode allows quick and easy set up.

Table 12: Region Mode Event Responses

Return Value	Key Down	Key Up	Key Down	255
Touch Event	Press	Release	Drag	Out of Region

4.6 Common Features

General Purpose Outputs
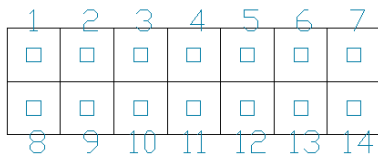
Figure 15: GPO Header

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 GLK24064-25/GLT24064 is the ability to control relays***** and other external devices using one of six General Purpose Outputs. Each can source up to 19mA of current at 4.6V when on or sink 19mA at 0V 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.

Dallas One-Wire Connector
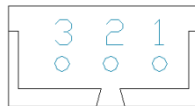
Figure 16: Dallas One-Wire Connector

Table 14: Dallas One-Wire Pinout

Pin	Function
1	Vcc
2	D
3	Gnd

In addition to the six general purpose outputs the GLK24064-25/GLT24064 offers an Optional Dallas One-Wire bridge, to allow for an additional thirty two one-wire devices to be connected to the display. This header can be populated with a Tyco 173979 connector at an added cost by custom order only. Please use the Contact section to request more information from the Matrix Orbital sales team.

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 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 protocol, ensure that the host system and display module are both communicating on the same baud rate. The default baud rate for the display module is 19200 bps.
- Match Rx from the GLK24064-25/GLT24064 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.

*Note: I²C 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 GLK model these are the middle two keypad pins, for the GLT these are the two pins near the keypad header.
3. Reconnect power to your unit, and wait for the start screen before removing the jumper. Please note the jumper will adversely affect GLT 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^{2}C Address	80

Table 15: 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 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 16: Accepted Baud Rate Values

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

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

Selects the protocol used for data transmission from the display. Data transmission to the display 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

1.4 Turn Software 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, 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 128 bytes. Not available in I^{2}C. 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

1.5 Turn Software 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.

1.6 Set Software Flow Control Response	Dec	**254 60**	Xon Xoff
	Hex	**FE 3C**	Xon Xoff
	ASCII	**■ <**	Xon Xoff

Sets the values returned for almost full and almost empty messages when in flow control mode. This command permits the display to utilize standard flow control values of 0x11 and 0x13, note that defaults are 0xFF and 0xFE.

Xon	1 byte, value returned when display buffer is almost empty, permitting transmission to resume
Xoff	1 byte, value returned when display buffer is almost full, signaling transmission to halt

1.7 Set Hardware Flow Control Trigger Level	Dec	**254 62**	Level
	Hex	**FE 3E**	Level
	ASCII	**■ >**	Level

Sets the hardware flow control trigger level. The Clear To Send signal will be deactivated once the number of characters in the display buffer reaches the level set; it will be reactivated once all data in the buffer is handled.

Level	1 byte, trigger level as below

Table 17: Hardware Flow Control Trigger Levels   *Table 18: Flow Control Settings*

Bytes	1	4	8	14
Level	0	1	2	3

Flow Control	None	Software	Hardware
Mode	0	1	2

1.8 Set Flow Control Mode	Dec	**254 63**	Mode
	Hex	**FE 3F**	Mode
	ASCII	**■ ?**	Mode

Toggles flow control between hardware, software and off settings. Software and Hardware control can be further tuned using the settings above. Default is Hardware, or 2.

Mode	1 byte, flow control setting as above

6.2 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.

6.3 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. Default font is ID 1.

ID	2 bytes*, unique font identification number, LSB first
Size	4 bytes*, size of the entire font file, LSB first
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 | 2 bytes*, 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 |

*Note: To accommodate additional memory, Font and Bitmap IDs have been increased to 2 bytes and size to 4 from firmware revision 8.1 onward. Please query your display revision to ensure this command is issued correctly.

Font File Creation

Matrix Orbital graphic displays are capable of displaying text in a wide variety of styles customizable to suit any project design. Font 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 in the software download section at www.matrixorbital.ca. 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.

Manually created font files will have three parts: the file header, character table, and character data.

Table 19: 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 20: 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 21: 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 22: 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 23: 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

6.4 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. Start screen is ID 1.

ID	2 bytes*, unique bitmap identification number, LSB first
Size	4 bytes*, size of the entire bitmap file, LSB first
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	2 bytes*, unique bitmap identification number
X Position	1 byte, leftmost coordinate of bitmap
Y Position	1 byte, topmost coordinate of bitmap

Note: To accommodate additional memory, Font and Bitmap IDs have been increased to 2 bytes and size to 4 from firmware revision 8.1 onward. Please query your display revision to ensure this command is issued correctly.

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, 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 affect of the Matrix Orbital graphic stylization ability.

Table 24: Smiley Face Bitmap

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

Table 25: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 26: Example Bitmap File

Header	5 4
Bitmap Data	80 34 224

6.5 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 27
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 27: Bar Graph Types

	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

5.9 Initialize a Strip Chart	Dec	**254 106**	ID	X1 Position	Y1 Position	X2 Position	Y2 Position
	Hex	**FE 6A**	ID	X1 Position	Y1 Position	X2 Position	Y2 Position
	ASCII	**■ j**	ID	X1 Position	Y1 Position	X2 Position	Y2 Position

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

5.10 Shift a Strip Chart	Dec	**254 107**	Direction & ID
	Hex	**FE 6B**	Direction & ID
	ASCII	**■ k**	Direction & ID

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 28: 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

6.6 General Purpose Output

6.1 General Purpose Output Off	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	1 byte, GPO to be turned off, value between 1 and 6

6.2 General Purpose Output On	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	1 byte, GPO to be turned on, value between 1 and 6

6.3 Set Start Up GPO State	Dec	**254 195**	Number State
	Hex	**FE C3**	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	1 byte, GPO to be controlled, value between 1 and 6
State	1 byte, 1 for on or 0 for off

6.7 Dallas One-Wire

7.1 Search for a One-Wire Device	Dec	254 200 2
	Hex	FE C8 02

Sends a search query to each of the up to 32 devices on the one wire bus. Any connected device will respond with an identification packet.

Response	14 bytes, identification packet as shown below

Table 29: Dallas One-Wire Packet Information

Offset	Length	Value	Description
0	2	9002	Preamble
2	1	138	Another device packet will follow OR
		10	Last device packet
3	1	49	Packet Type
4	1	0	Error Code (0 indicates success)
5	8		Device Address
13	1	0	CRC8 address check (0 indicates validity)

7.2 Dallas One-Wire Transaction	Dec	254 200 1	Flags Send Bits Receive Bits Data
	Hex	FE C8 01	Flags Send Bits Receive Bits Data

Performs a single Dallas 1-Wire transaction. Consult your device documentation for information regarding device specific protocols. If an error is encountered, a corresponding value will be returned by the device.

Flags	1 byte, flags for transaction, see below
Send Bits	1 byte, number of bytes to be sent to the device
Receive Bits	1 byte, number of bytes expected to be received from the device
Data	Variable, data to be transmitted LSB to MSB

Table 30: Dallas One-Wire Flag Table

Bit	Flag Description
7	
6	Unused
5	
4	0 (Future Compatibility)
3	Add CRC8 to transaction
2	0 (Future Compatibility)
1	Read CRC8 from transaction
0	Reset Bus prior to transaction

Table 31: Dallas One-Wire Error Table

Code	Error Description
0	Success
1	Unknown Command
2	No Devices Found
3	Fatal Search Error

6.8 Piezo Buzzer

8.1 Activate Piezo Buzzer	Dec	254 187	Frequency Time
	Hex	FE BB	Frequency Time

Activates a buzz of specific frequency from the onboard piezo buzzer for a specified length of time.

Frequency	2 bytes, frequency of buzz in hertz
Time	2 bytes, length of buzzer sound in milliseconds

6.9 Keypad

9.1 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.

9.2 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.

9.3 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)

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

Clears all key presses from the key buffer.

9.5 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)

9.6 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

9.7 Auto Repeat Mode Off	Dec	254 96
	Hex	FE 60

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

9.8 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. 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	25 bytes, key down values
Key Up	25 bytes, key up values

Table 32: 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 33: 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)

6.10 Touchpad

Set Touch Region	Dec	254 132	ID X Position Y Position Width Height Key Down Key Up
	Hex	FE 84	ID X Position Y Position Width Height Key Down Key Up

Creates a region of the screen that responds when pressed and released with a defined single byte.

ID	1 byte, unique region identification number, maximum 32 regions
X Position	1 byte, leftmost coordinate, value between 0 and 239
Y Position	1 byte, topmost coordinate, value between 0 and 63
Width	1 byte, width of region, must be within screen bounds
Height	1 byte, height of region, must be within screen bounds
Key Down	1 byte, value returned when region is pressed
Key Up	1 byte, value returned when region is released

Delete a Touch Region	Dec	254 133	ID
	Hex	FE 85	ID

Deletes a previously created touch region. Events from undefined regions return the value 255 by default.

ID	1 byte, unique region identification number

Delete All Touch Regions	Dec	254 134
	Hex	FE 86

Deletes all previously created touch regions. Recommended for use before dividing the screen into new regions.

Set Touch Mode	Dec	254 135	Mode
	Hex	FE 87	Mode

Sets the method used to return touch events. Region mode will return a single value for events in defined areas. Coordinate mode will return event, x position, and y position bytes for each press, drag, or release.

Mode	1 byte, touch reporting mode, 0 for region or 1 for coordinate mode. Default is coordinate.

10.1 Set Region Reporting Mode	Dec	254 136	Mode
	Hex	FE 88	Mode

Defines the events transmitted in region mode. Allows only events specified to return a value to the host. Key down values are transmitted for press and drag events, key up for release, and the value 255 for out of region.

Mode	1 byte, region reporting mode, see table below. Default reporting returns all events.

Table 34: Region Reporting Mode Byte

Byte	7-4	3	2	1	0
Event	Reserved	Out of Region	Drag	Release	Press

10.2 Set Dragging Threshold	Dec	254 137	Threshold
	Hex	FE 89	Threshold

Sets the distance a press is required to travel before a drag event is reported. Precision will vary inversely to data transmitted; care should be taken to find a suitable balance. Distance is calculated as $\Delta x^2 + \Delta y^2 = d^2$.

Threshold	1 byte, threshold value between 1 and 255. Default is 8.

10.3 Set Pressure Threshold	Dec	254 138	Threshold
	Hex	FE 8A	Threshold

Sets the pressure required to trigger a touch event.

Threshold	2 bytes, threshold value between 1 and 65535. Default is 1000.

10.4 Run Touchpad Calibration	Dec	254 139
	Hex	FE 8B

Triggers an interactive calibration of the touchpad. User will be required to touch various points on the screen during calibration. This command is recommended for use when environmental or user conditions change to ensure correct operation.

Response	2 bytes, command byte 254, then 21 for success or 20 for failure.

6.11 Display Functions

11.1 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

11.2 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.

11.3 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)

11.4 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)

11.5 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)

11.6 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)

6.12 Filesystem

12.1 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.

12.2 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	2 bytes, unique identification number of font or bitmap to be deleted

12.3 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 memory, LSB to MSB

12.4 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, 2 bytes representing number of entries plus 8 identification bytes for each entry

Table 35: 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 36: Extended Byte Descriptions

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

12.5 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 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
Data	variable length, data to upload

12.6 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	2 bytes, unique identification number of font or bitmap to download
Response	variable length, first 4 bytes represent file size followed by file data

12.7 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	2 bytes, original unique file identification number
New Type	1 byte, new file type, 0 for font or 1 for bitmap
New ID	2 bytes, new unique file identification number

12.8 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	4 bytes of size LSB to MSB followed by entire 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 37: 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 38: Font Upload Protocol				*Table 39: Bitmap Upload Protocol*		
Host	**Display**	**Comments**		**Host**	**Display**	**Comments**
254		Command Prefix		254		Command Prefix
36		Upload Font File Command		94		Upload Bitmap File Command
1		Reference ID LSB		1		Reference ID LSB
0		Reference ID MSB		0		Reference ID MSB
31		Font File Size LSB		5		Bitmap File Size LSB
0		Font File Size		0		Bitmap File Size
0		Font File Size		0		Bitmap File Size
0		Font File MSB		0		Bitmap File MSB
	1	Size Confirmation			1	Size Confirmation
5		First Font Data Byte		5		First Bitmap Data Byte
	5	Echo Data Byte			5	Echo Data Byte
1		Confirm Data Byte		1		Confirm Data Byte
…	…	…		…	…	…
96		Last Font Data Byte		224		Last Bitmap Data Byte
	96	Echo Data Byte			224	Echo Data Byte
1		Confirm 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.

<div style="display:flex">

Table 40: 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
0		Size
1		Size
0		Size MSB
	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 41: 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

</div>

6.13 Data Security

| 13.1 Set | Dec | **254 147** | Switch |
| Remember | 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 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.

Switch	1 byte, 1 for on or 0 for off

| 13.2 Set Data | Dec | **254 202 245 160** | Level |
| Lock | 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 42

Table 42: Data Lock Bits

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

Table 43: 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

| 13.3 Set and Save | Dec | **254 203 245 160** | Level |
| Data Lock | 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

6.14 Miscellaneous

14.1 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

14.2 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

14.3 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

14.4 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 44: Sample Module Type Responses

21	GLK24064-25	105	GLT24064
107	GLK24064-25-USB	106	GLT24064-USB
109	GLK24064-25-422	110	GLT24064-422

14.5 Read Screen	Dec	**254 184**
	Hex	**FE B8**

Return the current commanded state of each pixel on the screen.

Response	1920 bytes, 30 bytes per display row, 64 rows of data representing the Boolean value of each pixel

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 45: 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
Transmission Protocol Select	160	A0	á	Protocol	None	Remember On
Turn Software Flow Control On	58	3A	:	Full, Empty	None	Remember On
Turn Software Flow Control Off	59	3B	;	None	None	Remember On
Set Software Flow Control Response	60	3C	<	Xon, Xoff	None	Remember On
Set Hardware Flow Control Trigger Level	62	3E	>	Level	None	Remember On
Set Flow Control Mode	63	3F	?	Mode	None	Remember On

Table 46: 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 47: Font Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Upload a Font File	36	24	$	ID [2], Size [4], Data []	See Font File Creation	Always
Set the Current Font	49	31	1	ID [2]	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 48: Bitmap Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Upload a Bitmap File	94	5E	^	ID [2], Size [4], Data []	See Bitmap File Creation	Always
Draw a Bitmap from Memory	98	62	b	ID [2], X, Y	None	Never
Draw a Bitmap Directly	100	64	d	X, Y, Width, Height, Data []	None	Never

Table 49: 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 50: 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 51: Dallas One-Wire Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Search for a One-Wire Device	200, 2	C8, 02	└, ☻	None	Data [14]	Never
Dallas One-Wire Transaction	200, 1	C8, 01	└, ☺	Flags, Send, Receive, Data []	Data []	Never

Table 52: Piezo Buzzer Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Activate Piezo Buzzer	187	BB	»	Frequency [2], Time [2]	None	Never

Table 53: 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 [25], KeyDown [25]	None	Always

Table 54: Touchpad Command Summary

Name	Dec	Hex	ASCII	Parameters	Response	Remembered
Set Touch Region	132	84	ä	ID, X, Y, Width, Height, KeyUp, KeyDown	None	Remember On
Delete a Touch Region	133	85	à	ID	None	Remember On
Delete All Touch Regions	134	86	å	None	None	Remember On
Set Touch Mode	135	87	ç	Mode	None	Remember On
Set Region Reporting Mode	136	88	ê	Mode	None	Remember On
Set Dragging Threshold	137	89	ë	Threshold	None	Remember On
Set Pressure Threshold	138	8A	è	Threshold	None	Remember On
Run Touchpad Calibration	139	8B	ï	None	Outcome [2]	Always

Table 55: 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 Contrast	80	50	P	Contrast	None	Remember On
Set and Save Contrast	145	91	æ	Contrast	None	Always

Table 56: 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 [2]	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 [2]	Data []	Never
Move a File	180	B4	⊣	Old Type, Old ID [2], New Type, New ID [2]	None	Always
Dump the Filesystem	48	30	0	None	Size [4], Data []	Never

Table 57: 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 58: 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
Read Screen	184	B8	⌐	None	Pixels [1920]	Never

7.2 Environmental Specifications

Table 59: Environmental Limits

	Standard	*Extended (-E)
Operating Temperature	0°C to +50°C	-20°C to +70°C
Storage Temperature	-10°C to +60°C	-30°C to +80°C
Operating Relative Humidity	Maximum 90% non-condensing	

*Note: The Extended Temperature option is not available for any variant of the GLT24064.

7.3 Electrical Tolerances

Current Consumption

Table 60: Current Consumption

Board		Backlight		GPOs
85mA	+	55 to 440 mA	+	20mA each maximum

Table 61: Backlight Current Draw

YG	GW & WB
440mA	55mA

Input Voltage Specifications

Table 62: Voltage Specifications

Standard**	Extended Wide Voltage (-VPT)
4.75-5.25V	9.0-35.0V

**Note: The Standard Voltage variant of the RS422 model should be powered from a local source only.

7.4 Optical Characteristics

Table 63: Display Optics

Module Size	180.00 x 65.00 x 30.5	mm
Viewing Area	132.2 x 39.2	mm
Active Area	127.16 x 33.88	mm
Pixel Size	0.49 x 0.49	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	50,000	Hours

7.5 Dimensional Drawings

Figure 17: Display Dimensional Drawing

Figure 18: Standard Model Dimensional Drawing
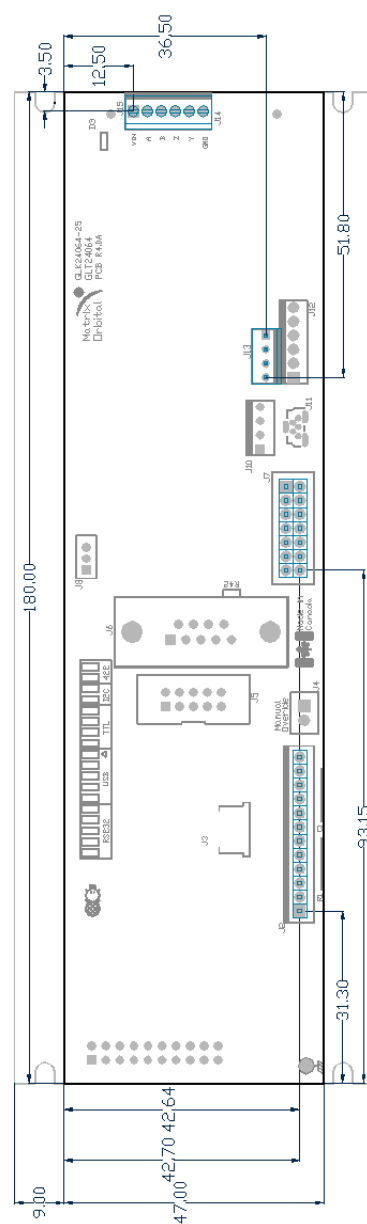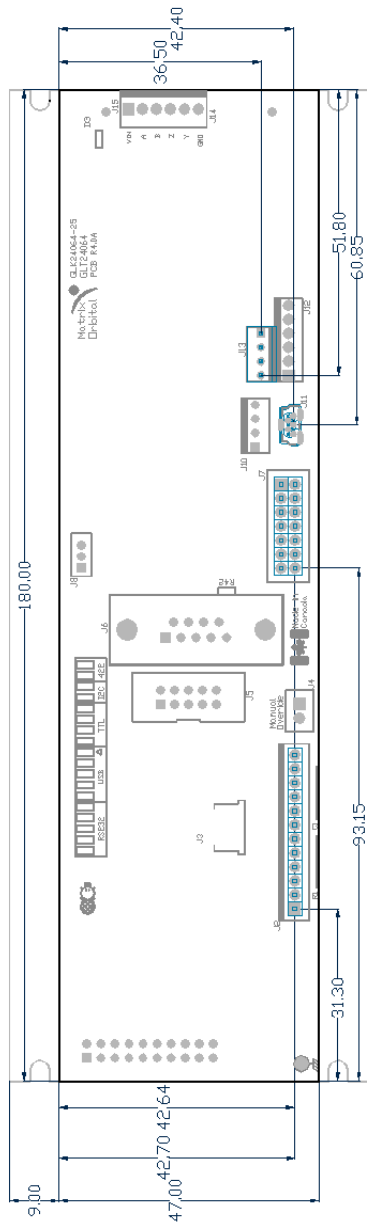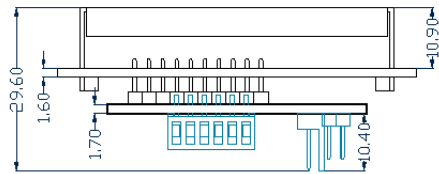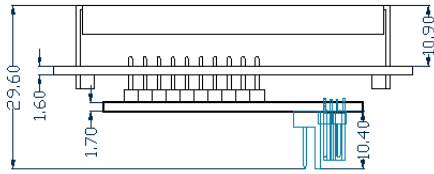
Figure 19: USB Model Dimensional Drawing

Figure 20: RS422 Model Dimensional Drawing

8 Ordering

8.1 Part Numbering Scheme

Table 64: Part Numbering Scheme

GLK	-24064			-GW	-VPT	-E
1	2	3	4	5	6	7

8.2 Options

Table 65: Display Options

#	Designator	Options
1	Product Type	GLK: Graphic Liquid Crystal Display with Keypad Input GLT: Graphic Liquid Crystal Display with Touchpad Input
2	Display Size	24064: 240 pixel columns by 64 rows
3	Keypad Size	NP: No keypad 25: 25 key maximum
4	Protocol	NP: Standard Model -USB: USB Only Model -422: RS422 Only Model*
5	Colour	NP: Standard (Grey Text with Yellow-Green Background) GW: Grey Text with White Background WB: White Test with Blue Background
6	Voltage	NP: Standard Voltage -VPT: Wide Voltage with Efficient Switching Power Supply
7	Temperature	NP: Standard -E: Extended Temperature**

*Note: The RS422 model should only be powered from a local source, unless the –VPT variant is used.

**Note: The Extended Temperature option is not available for any variant of the GLT24064.

8.3 Accessories
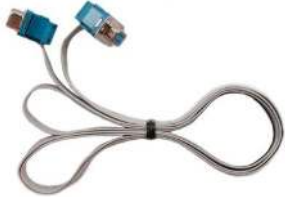
Power

PCS	Standard Power Cable	

Communication

CSS4FT	4 ft. Serial Cable	
EXTMUSB3FT	Mini-USB Cable	
INTMUSB3FT	Internal Mini-USB Cable	
ESCCPC5V	Extended Serial Communication/5V Power Cable	
BBC	Breadboard Cable	

Peripherals

KPP4x4	16 Button Keypad	
Temperature Probe	Dallas One-Wire Temperature Probe	

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.

DOW: Dallas One-Wire protocol, similar to I^2C, provides reduced data rates at a greater distance. One wire carries data, while two others supply power and ground. Matrix Orbital tests non-parasitic devices only, those that do not draw power from the data line; however, some parasitic devices may work.

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 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 high level is -30V, a low is +30V.

RS422: Recommended standard 422, a more robust differential pair serial protocol.

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.

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

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

10 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