Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

GLK12232A-25-SM/GLT12232A-SM

Including GLK12232A-25-SM-USB and GLT12232A-SM-USB

Technical Manual

Revision 3.4

PCB Revision: 3.0 or Higher

Firmware Revision: 8.4 or Higher

Revision History

Revision	Date	Description	Author
3.4	October 19, 2015	Minor command updates	Clark
3.3	September 21, 2015	Updated commands for Firmware Revision 8.6	Clark
3.2	January 30, 2015	Updated Drawings	Clark
3.1	May 21, 2014	Added Commands for Firmware Revision 8.5	Martino
3.0	March 13, 2013	Initial Release	Clark

Contents

1 Introduction 1

2 Quick Connect Guide 2

2.1 Available Headers 2

2.2 Standard Module 3

Recommended Parts 3

Serial Connections 3

I^2C Connections 4

2.3 USB Module 4

Recommended Parts 4

USB Connections 5

3 Software 6

3.1 MOGD# 6

3.2 Firmware Upgrade 7

3.3 Application Notes 7

4 Hardware 8

4.1 Standard Model 8

Extended Communication/Power Header 8

I^2C Communication/Power Header 8

Protocol Select Jumpers 8

4.2 USB Model 9

USB Connector 9

Alternate Power Connector 9

4.3 Common Features 10

General Purpose Outputs 10

Hardware Lock 10

4.4 GLK Model 10

Keypad Header 10

4.5 GLT Model 11

Touch Screen 11

Coordinate Mode 11

Region Mode 11
5 Troubleshooting 12
5.1 Power 12
5.2 Display 12
5.3 Communication 12
5.4 Manual Override 13
6 Commands 14
6.1 Communication 14
6.2 Text 16
6.3 Drawing 19
6.4 Fonts 24
Font File Creation 25
6.5 Bitmaps 26
Bitmap File Creation 27
Bitmap Masking 28
6.6 9-Slices 28
9-Slice File Creation 30
6.7 Animations 30
Animation File Creation 32
6.8 General Purpose Output 32
6.9 Piezo Buzzer 32
6.10 Keypad 33
6.11 Touchpad 35
6.12 Display Functions 37
6.13 Scripting 38
6.14 Filesystem 39
File Upload Protocol 42
XModem Upload Protocol 43
6.15 Data Security 45
6.16 Miscellaneous 45
7 Appendix 47
7.1 Command Summary 47

7.1 Block Diagram 52
7.2 Data Types 52
7.3 Environmental Specifications 53
7.4 Electrical Tolerances 53
7.1 Optical Characteristics 53
7.2 Dimensional Drawings 54
8 Ordering 56
8.1 Part Numbering Scheme 56
8.2 Options 56
8.3 Accessories 57
9 Definitions 58
10 Contact 58

1 Introduction

Figure 1: GLK12232A-25-SM/GLT12232A-SM Display

The GLK12232A-25-SM/GLT12232A-SM 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 I^2C protocols available in the standard model, the USB communication model allows the GLK12232A-25-SM-USB/GLT12232A-SM-USB to be connected to a wide variety of host controllers. Communication speeds of up to 115.2 kbps in serial modes and 400 kHz in I^2C ensure lightning fast text and graphic updates.

The simple command structure permits easy software control of many settings including backlight brightness, screen contrast, and baud rate. On board memory provides a whopping 256KB of customizable fonts and bitmaps to enhance the graphical user experience.

User input on the GLK12232A-25-SM is available through a five by five matrix style keypad or a resistive touch overlay on the GLT12232A-SM. In addition, two general purpose outputs provide simple switchable five volt sources on each model, while a small piezo speaker offers audio feedback for a completely interactive experience.

The versatile GLK12232A-25-SM/GLT12232A-SM, 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

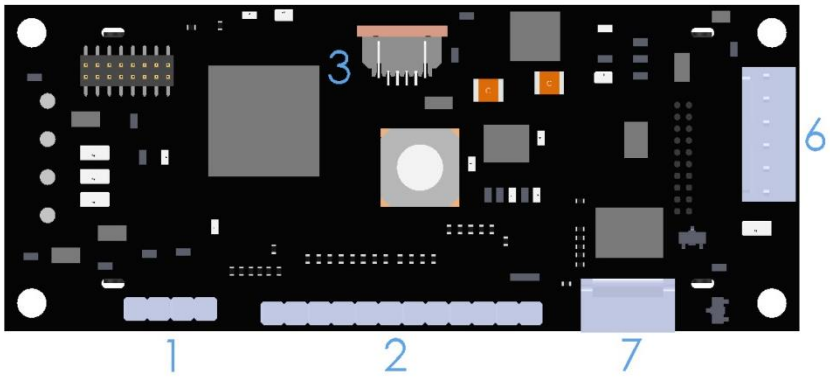


Figure 2: GLK12232A-25-SM/GLT12232A-SM Standard Module Header Locations

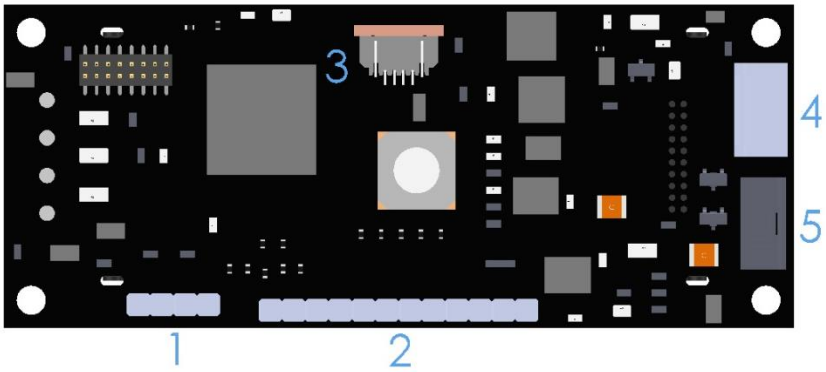


Figure 3: GLK12232A-25-SM/GLT12232A-SM USB Module Header Locations

Table 1: List of Available Headers

#	Header	Mate	Population
1	GPO Header	None Offered	All Models
2	Keypad	KPP4x4	GLK Model Only
3	Touchpad	Touch Panel	GLT Model Only
4	USB Connector	EXTPUSB6FT	USB Model Only
5	Alternate Power Header	PCS	USB Model Only
6	Communication Header	ESCCPC5V	Standard Model Only
7	I2C Communication/Power Header	None Offered	Standard Model Only

2.2 Standard Module

The standard version of the GLK12232A-25-SM/GLT12232A-SM allows for user configuration of two common serial levels. The unit can communicate using serial protocol at either RS323 or TTL voltage levels. Connections for each serial protocol can be accessed through the six pin Communication/Power Header as outlined in the Serial Connections section below.

Recommended Parts

Figure 4: Extended Communication/Power Cable (ESCCPC5V)

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

Figure 5: Breadboard Cable (BBC)

For a more flexible interface to the GLK12232A-25-SM/GLT12232A-SM, 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 GLK12232A-25-SM/GLT12232A-SM. The Communication/Power Cable is most commonly used for this set up as it provides connections for DB9 serial and floppy power cables. To place your board in Serial mode, adhere to the steps laid out below.

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

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

2. Make the connections.
   a. Connect the six pin female header of the Communication/Power Cable to the Communication/Power Header of your GLK12232A-25-SM/GLT12232A-SM.
   b. Insert the male end of your serial cable to the corresponding DB9 header of the Communication/Power Cable and the mate the female connector with the desired communication port of your computer.
   c. Select an unmodified floppy cable from a PC power supply and connect it to the power header of the Communication/Power Cable.

3. Create.
    - MOGD# or a terminal program will serve to get you started, and then you can move on with your own development. Instructions for the former can be found below and a variety of application notes are available for the latter at www.matrixorbital.ca/appnotes.

I²C Connections

A more advanced connection to the GLK12232A-25-SM/GLT12232A-SM is provided by the I²C 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 GLK12232A-25-SM/GLT12232A-SM in I²C mode, get started with the guidelines below.

1. Set the Protocol Select switches.
    - I²C: Ensure that the two I²C 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 GLK12232A-25-SM/GLT12232A-SM and plug the four leads into your breadboard. The red lead will require power, while the black should be connected to ground, and the green and yellow should be connected to your controller clock and data lines respectively.
    b. Pull up the clock and data lines to five volts using a resistance between one and ten kilohms on your breadboard.
3. Create.
    - This time you're on your own. While there are many examples within the Matrix Orbital AppNote section, www.matrixorbital.ca/appnotes, too many controllers and languages exist to cover them all. If you get stuck in development, it is possible to switch over to another protocol on the standard board, and fellow developers are always on our forums for additional support.

2.3 USB Module

The GLK12232A-25-SM-USB/GLT12232A-SM-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 4pin friction-lock style connector as outlined in the USB Connections section.

Recommended Parts

Figure 6: 4pin USB Cable (EXT4PUSB3FT)

The External 4pin USB cable is recommended for the GLK12232A-25-SM-USB/GLT12232A-SM-USB display. It will connect to the friction-locking header on the unit and provide a connection to a regular A style USB connector, up to six feet away.

USB Connections

The USB connection is the quickest, easiest solution for PC development. After driver installation, the GLK12232A-25-SM-USB/GLT12232A-SM-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 GLK12232A-25-SM-USB/GLT12232A-SM-USB please follow the steps below.

1. Set the Protocol Select jumpers.
   - USB: The GLK12232A-25-SM-USB/GLT12232A-SM-USB offers USB protocol only. Model specific hardware prevents this unit from operating in any other protocol, and does not allow other models to operate in USB. Protocol Select jumpers on the USB model cannot be moved.
2. Make the connections.
   - Plug the friction lock header of your External 4pin USB cable into your GLK12232A-25-SM-USB/GLT12232A-SM-USB and the regular USB header into your computer USB jack.
3. Install the drivers.
   a. Download the latest drivers at www.matrixorbital.ca/drivers, and save them to a known location.
   b. When prompted, install the USB bus controller driver automatically
   c. If asked, continue anyway, even though the driver is not signed
   d. When the driver install is complete, your display will turn on, but communication will not yet be possible.
   e. At the second driver prompt, install the serial port driver automatically
   f. Again, if asked, continue anyway
4. Create.
   - Use MOGD# or a terminal program to get started, and then move on with your own development. Instructions for the former can be found below and a number of application notes are available for the latter at www.matrixorbital.ca/appnotes.

3 Software

The communication protocol available and simple command structure of the GLK12232A-25-SM/GLT12232A-SM means that a variety of applications can be used to communicate with the display. Text is sent to the display as a character string, for example, sending the decimal value 41 will result in an 'A' appearing on the screen. A single control character is also available. Commands are merely values prefixed with a special command byte, 254 in decimal.

Table 2: Reserved Control Characters

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

Once the correct communication port is identified, the following communication settings can be applied to communicate correctly with the GLK12232A-25-SM/GLT12232A-SM.

Table 3: Communication Settings

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

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

3.1 MOGD#

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

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

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

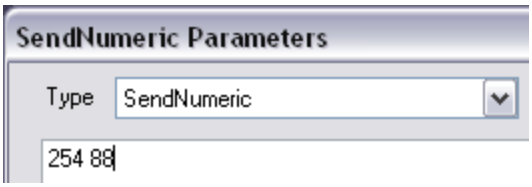


Figure 7: MOGD# Command Example

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

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

3.2 Firmware Upgrade

The firmware of the GLK12232A-25-SM/GLT12232A-SM can be upgraded in the field. All firmware revisions can be installed using software found at www.matrixorbital.ca/software/GLT Series.

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.

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

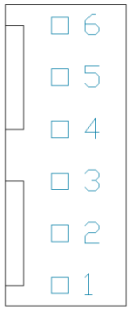


Figure 8: Extended Communication/Power Header

Table 4: Extended Communication/Power Pinout

Pin	Function
1	Vcc
2	Rx
3	Tx
4	Gnd
5	CTS
6	RTS

The Extended Communication/Power Header provides a standard connector for interfacing to the GLK12232A-25-SM/GLT12232A-SM. Voltage is applied through pins one and four of the six pin Extended Communication/Power Header. Please ensure the correct voltage input for your display by referencing the Voltage Specifications before connecting power. Pins two and three are reserved for serial transmission, using either the RS-232/TTL, depending on what has been selected by the Protocol Select Jumpers. Pins five and six can be used for serial transmission hardware flow control. The Molex 22-04-1061 style header used can be mated to a number of connectors, a 22-01-1062 for example.

I²C Communication/Power Header

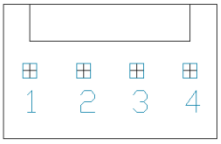


Figure 9: I2C Communication/Power Header

Table 5: I²C Communication/Power Pinout

Pin	Function
1	Vcc
2	SCL
3	SDA
4	Gnd

Voltage is applied through pins one and four of the header, please reference the electrical specifications before applying power. Pins two and three are reserved for I²C clock and data signals respectively, both of which should be pulled up to five volts using a resistance between one and ten kilohms. The Tyco 640456-4-LF style header used can be mated to a number of connectors, including Molex 22-01-3047.

Protocol Select Jumpers

The Protocol Select Jumpers provide the means necessary to toggle the GLK12232A-25-SM/GLT12232A-SM between RS-232 and TTL protocols. As a default, the jumpers are set to RS-232 mode with solder jumps on the RS232 jumpers. In order to change the display to TTL mode, simply remove the zero ohm resistors from the RS232 jumpers and solder them to the TTL jumpers.

4.2 USB Model

USB Connector

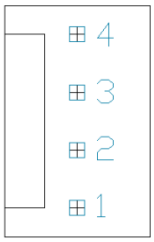


Figure 10: USB Connector

Table 6: USB Pinout

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

The GLK12232A-25-SM-USB/GLT12232A-SM-USB comes with a friction-locking straight pin Connector to fulfill both communication and power needs. Most commonly used with a PC, this connection creates a virtual com port that offers a simple power solution with a familiar communication scheme. The Molex 22-04-1061 style header used can be mated to a number of connectors, a 22-01-1062 for example.

Alternate Power Connector

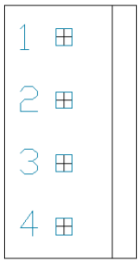


Figure 11: Alternate Power Connector

Table 7: Alternate Power Pinout

Pin	Function
1	Vcc
2	Gnd
3	Gnd
4	NC/Vcc

The Alternate Power Connector provides the ability to power the GLK12232A-25-SM-USB/GLT12232A-SM-USB using a second cable. The Tyco 171825-4 style header is particularly useful for connecting to an unmodified floppy power cable, a 171822-4 for example, from a PC power supply for a simple bench power solution.

4.3 Common Features

General Purpose Outputs

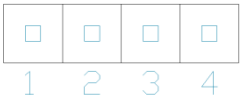


Figure 12: GPO Header

Table 8: GPO Pinout

Pin	Function
1	GPO 1
2	Gnd
3	GPO 2
4	Gnd

A unique feature of the GLK12232A-25-SM/GLT12232A-SM is the ability to control relays* and other external devices using one of two General Purpose Outputs. Each can source up to 3mA of current at three volts when on or sink 3mA at zero volts when off. The four pin header can be interfaced to a number of female connectors to provide control to any peripheral devices required.

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

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 Resistor, 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.

4.4 GLK Model

Keypad Header

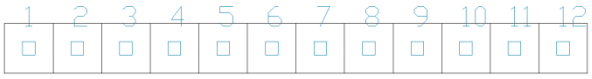


Figure 13: Keypad Header

Table 9: 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 GLK12232A-25-SM 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. The character that is associated with each key press may 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.

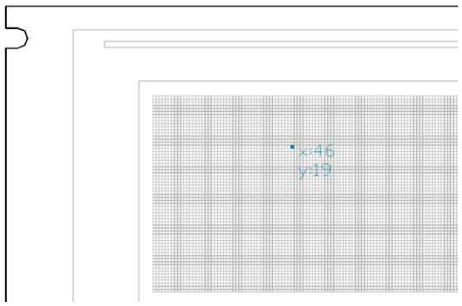
***Note:** The Ground / +3.3V pin is toggled by the jumper to the right of the keypad connector. Jump pads 1 & 2 for +3.3V or 2 & 3 for GND.

4.5 GLT Model

Touch Screen

The GLT12232A-SM 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

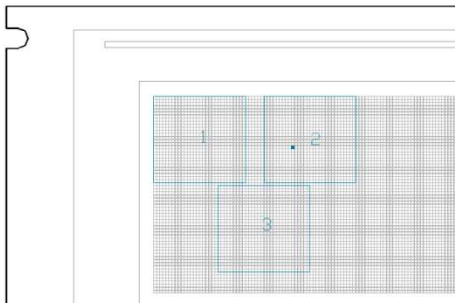


In coordinate mode all touch events are reported using three single byte values. First, the type of event is transmitted, followed by the x and y coordinates of its position. Pressure and drag thresholds must be exceeded for an event to be registered. A low drag threshold will result in greater tracking 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 10: 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 11: Region Mode Event Responses

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

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.
- 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.

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 USB Port.
- In USB protocol, 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.
- 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.

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 only two pins on 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 GLT12232A-SM 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

Table 12: Manual Override Settings

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

6 Commands

6.1 Communication

1.1 Change Baud Rate	Dec	254 57	Speed	v8.0
	Hex	FE 39	Speed	
	ASCII	■ 9	Speed	
Immediately changes the baud rate. Baud rate can be temporarily forced to 19200 by a manual override.				
Speed	Byte	Valid settings shown below.		

Table 13: Accepted Baud Rate Values

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

1.2 Change I2C Slave Address	Dec	254 51	Address	v8.0
	Hex	FE 33	Address	
	ASCII	■ 3	Address	
Immediately changes the I2C write address. Only even values are permitted as the next odd address will become the read address. Default is 80.				
Address	Byte	Even value.		

1.3 Transmission Protocol Select	Dec	254 160	Protocol	v8.0
	Hex	FE A0	Protocol	
		■ á	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	Byte	1 for Serial (RS232/RS422/TTL/USB) or 0 for I2C.		

1.4 Set a Non-Standard Baud Rate	Dec	254 164	Speed	v5.0
	Hex	FE A4	Speed	
	ASCII	■ ñ	Speed	
Immediately changes the baud rate to a non-standard value. Speed must be a whole number between 977 and 153800. Due to rounding, error increases with baud rate, actual baud must be within 3% of desired baud to ensure accurate communication. Not available in I2C. Can be temporarily forced to 19200 by a manual override.				
Speed	Short	Calculations shown below, standard crystal speed is 16MHz.		

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

Equation 1: Speed Byte Calculation

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

Equation 2: Actual Baud Rate Calculation

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

Equation 3: Baud Rate Error Calculation

1.5 Set Flow Control Mode	Dec	254 63	Mode	v8.0
	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 Off, or 0.

Mode	Byte	Flow control setting as below.

Table 14: Hardware Flow Control Trigger Levels

Bytes	1	4	8	14
Level	0	1	2	3

Table 15: Flow Control Settings

Flow Control	None	Software	Hardware
Mode	0	1	2

1.6 Set Hardware Flow Control Trigger Level	Dec	254 62	Level	v8.0
	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	Byte	Trigger level as above.

1.7 Turn Software Flow Control On	Dec	254 58	Almost Full	Almost Empty	v8.0
	Hex	FE 3A	Almost Full	Almost Empty	
	ASCII	■ :	Almost Full	Almost 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 256* bytes. Not available in I^2C. Default off.

Almost Full	Byte	Number of bytes remaining before buffer is completely full, 0 < Full < Empty < 256*.
Almost Empty	Byte	Number of bytes before buffer can be considered empty enough to accept data.

***Note:** Buffer size was increased to 256 bytes from 128 bytes at firmware revision 8.3.

1.8 Turn Software Flow Control Off	Dec	254 59	v8.0
	Hex	FE 3B	
	ASCII	■ ;	

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

1.9 Set Software Flow Control Response	Dec	254 60	Xon	Xoff	v8.0
	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	Byte	Value returned when display buffer is almost empty, permitting transmission to resume.
Xoff	Byte	Value returned when display buffer is almost full, signaling transmission to halt.

1.10 Echo	Dec	254 255	Length	Data	v8.3
	Hex	FE FF	Length	Data	
	ASCII	■	Length	Data	

Send data to the display that it will echo. Useful to confirm communication or return information from scripts.

Length	Word	Length of data array to be echoed.
Data	Byte(s)	An arbitrary array of data that the module will return.
Response	Byte(s)	The same arbitrary array of data originally sent.

1.11 Delay	Dec	254 251	Time	v8.3
	Hex	FE FB	Time	
	ASCII	■ √	Time	

Pause command execution to and responses from the display for the specified length of time.

Time	Word	Length of delay in ms, maximum 2000.

1.12 Software Reset	Dec	254 253 77 79 117 110	v8.4
	Hex	FE FD 4D 4F 75 6E	
	ASCII	■ ² M O u n	

Reset the display as if power had been cycled via a software command. No commands should be sent while the unit is in the process of resetting; a response will be returned to indicate the unit has successfully been reset.

Response	Word	Successful reset response, 254 214.

6.2 Text

2.1 Clear Screen	Dec	254 88	v8.0
	Hex	FE 58	
	ASCII	■ X	

Clears the contents of the screen.

2.2 Go Home	Dec	254 72	v8.0
	Hex	FE 48	
	ASCII	■ H	

Returns the cursor to the top left of the screen.

2.3 Set Cursor Position	Dec	254 71	Column	Row	v8.0
	Hex	FE 47	Column	Row	
	ASCII	■ G	Column	Row	

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

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

2.4 Set Cursor Coordinate	Dec	254 121	X Y	v8.0
	Hex	FE 79	X Y	
	ASCII	■ y	X Y	

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

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

2.5 Get String Extents	Dec	254 41	Text	v8.6
	Hex	FE 29	Text	
	ASCII	■)	Text	

Read the size of the rectangle that the specified string would occupy if it was rendered with the current font.

Text	String	String on which to preform extents calculation. A single line of text is assumed.
Response	Byte(s)	Width and height of the string in pixels. A width greater than the screen will return 0.

2.6 Initialize Text Window	Dec	254 43	ID X1 Y1 X2 Y2 Font CharSpace LineSpace Scroll	v8.3
	Hex	FE 2B	ID X1 Y1 X2 Y2 Font CharSpace LineSpace Scroll	
	ASCII	■ +	ID X1 Y1 X2 Y2 Font CharSpace LineSpace Scroll	

Designates a portion of the screen to which text can be confined. Font commands affect only the current window, default (entire screen) is window 0.

ID	Byte	Unique text window identification number, value between 0 and 15.
X1	Byte	Leftmost coordinate.
Y1	Byte	Topmost coordinate.
X2	Byte	Rightmost coordinate.
Y2	Byte	Bottommost coordinate.
*Font	Short	Unique font ID to use for this window, value between 0 and 1023.
CharSpace	Byte	Spacing between characters to use for this window.
LineSpace	Byte	Spacing between lines to use for this window.
Scroll	Byte	Number of pixel rows to write to before scrolling text.

***Note:** Font was changed from a Byte length at firmware revision 8.5

2.7 Set Text Window	Dec	254 42	ID	v8.3
	Hex	FE 2A	ID	
	ASCII	■ *	ID	

Sets the text window to which subsequent text and commands will apply. Default (entire screen) is window 0.

ID	Byte	Unique text window to use.

2.8 Clear Text Window	Dec	254 44	ID	v8.3
	Hex	FE 2C	ID	
	ASCII	■ ,	ID	

Clears the contents of a specific text window, similar to the clear screen command.

ID	Byte	Unique text window to clear.

2.9 Initialize Label	Dec	254 45	ID X1 Y1 X2 Y2 Vert Hor Font Background CharSpace	v8.3
	Hex	FE 2D	ID X1 Y1 X2 Y2 Vert Hor Font Background CharSpace	
	ASCII	■ -	ID X1 Y1 X2 Y2 Vert Hor Font Background CharSpace	

Designates a portion of the screen that can be easily updated with one line of text, often used to display variables.

ID	Byte	Unique label identification number, value between 0 and 15.
X1	Byte	Leftmost coordinate.
Y1	Byte	Topmost coordinate.
X2	Byte	Rightmost coordinate.
Y2	Byte	Bottommost coordinate.
Vert	Byte	Vertical justification of the label text; 0 for top, 1 for middle, or 2 for bottom.
Hor	Byte	Horizontal justification of the label text; 0 for left, 1 for centre, or 2 for right.
Font	Short	Unique font ID to use for this label, value between 0 and 1023.
Background	Byte	State of the pixels in the label region that is not occupied by text; 0 for off or 1 for on.
CharSpace	Byte	Spacing between characters to use for this label.

***Note:** Font was changed from a Byte length at firmware revision 8.5

2.10 Initialize Scrolling Label	Dec	254 47	ID X1 Y1 X2 Y2 Vert Dir Font Background CharSpace Delay	v8.6
	Hex	FE 2F	ID X1 Y1 X2 Y2 Vert Dir Font Background CharSpace Delay	
	ASCII	■ /	ID X1 Y1 X2 Y2 Vert Dir Font Background CharSpace Delay	

Designates a portion of the screen that can be easily updated with one line of text, often used to display variables.

ID	Byte	Unique label identification number, value between 0 and 15.
X1	Byte	Leftmost coordinate.
Y1	Byte	Topmost coordinate.
X2	Byte	Rightmost coordinate.
Y2	Byte	Bottommost coordinate.
Vert	Byte	Vertical justification of the label text; 0 for top, 1 for middle, or 2 for bottom.
Dir	Byte	Direction of the scrolling behavior; 0 for left, 1 for right, or 2 for bounce.
Font	Short	Unique font ID to use for this label, value between 0 and 1023.
Background	Byte	State of the pixels in the label region that is not occupied by text; 0 for off or 1 for on.
CharSpace	Byte	Spacing between characters to use for this label.
Delay	Short	Time in milliseconds to elapse between characters printed.

2.11 Update Label	Dec	254 46	ID Data	v8.3
	Hex	FE 2E	ID Data	
	ASCII	■ .	ID Data	

Update a previously created label with new text. Send a null character (empty string) to clear a label.

ID	Byte	Unique label to update, between 0 and 15.
Data	String	Information to display in the label, must be terminated with a null (value of zero) byte.

2.12 Auto Scroll On	Dec	254 81	v8.0
	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.

2.13 Auto Scroll Off	Dec	254 82		v8.0
	Hex	FE 52		
	ASCII	■ R		

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

6.3 Drawing

3.1 Set Drawing Colour	Dec	254 99	Colour	v8.0
	Hex	FE 63	Colour	
	ASCII	■ c	Colour	

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

Colour	Byte	0 for inactive (background) colour or any other value for active (text) colour.

3.2 Draw Pixel	Dec	254 112	X Y	v8.0
	Hex	FE 70	X Y	
	ASCII	■ p	X Y	

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

X	Byte	Horizontal position of pixel to be drawn.
Y	Byte	Vertical position of pixel to be drawn.

3.3 Draw a Line	Dec	254 108	X1 Y1 X2 Y2	v8.0
	Hex	FE 6C	X1 Y1 X2 Y2	
	ASCII	■ l	X1 Y1 X2 Y2	

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

X1	Byte	Horizontal coordinate of first terminus.
Y1	Byte	Vertical coordinate of first terminus.
X2	Byte	Horizontal coordinate of second terminus.
Y2	Byte	Vertical coordinate of second terminus.

3.4 Continue a Line	Dec	254 101	X Y	v8.0
	Hex	FE 65	X Y	
	ASCII	■ e	X Y	

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

X	Byte	Left coordinate of terminus.
Y	Byte	Top coordinate of terminus.