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

LK162B-7T

Including LK162B-7T-USB variant

Technical Manual

Revision 1.2

PCB Revision: 1.0 or Higher

Firmware Revision: 7.3 or Higher

Revision History

Revision	Date	Description	Author
1.2	March 31, 2016	Revised Commands for Firmware Revision 7.3	Divino
1.1	March 12, 2014	Revision and correction to Colour in Ordering Options	Martino
1.0	February 20, 2014	Initial Release	Clark

Contents

1 Introduction

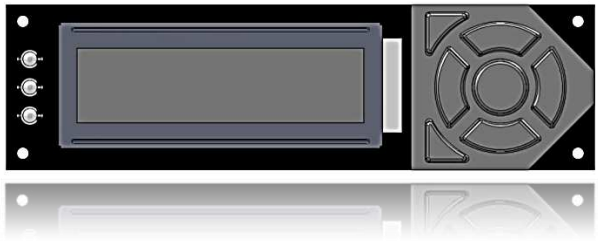

Figure 1: LK162B-7T Display

The LK162B-7T is an intelligent alphanumeric liquid crystal display designed to decrease development time by providing an instant solution to any project. In addition to the RS232, TTL and I2C protocols available in the standard model, the USB communication models allow the LK162B-7T to be connected to a wide variety of host controllers. Communication speeds of up to 115.2kbps for serial protocols and 100kbps for I^2C ensure lightning fast display updates.

The simple command structure permits easy software control of many settings including backlight brightness, screen contrast, and baud rate. On board memory provides up to forty custom characters which can be saved within the unit and recalled for start screens, bar graphs or larger numbers.

User input on the LK162B-7T is available through a seven key, integrated tactile keypad, and three bi-color LEDs provide visual output. An additional two general purpose outputs on the back of the unit provide simple switchable five volt sources

The versatile LK162B-7T, 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

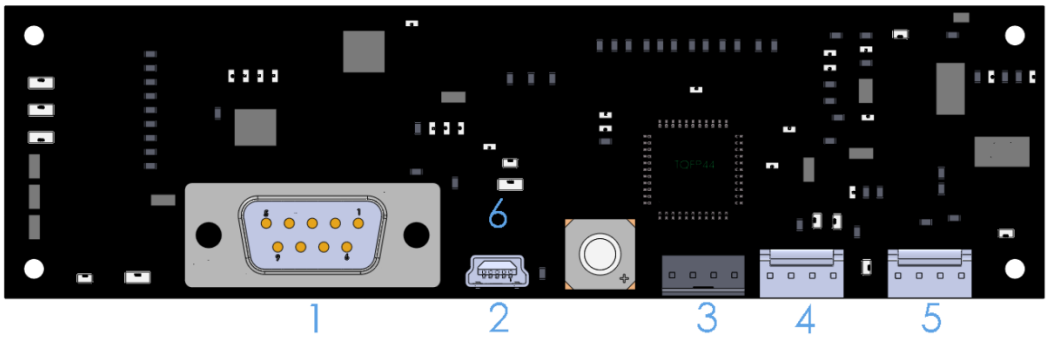

Figure 2: LK162B-7T Standard Module Header Locations

Table 1: List of Available Headers

#	Header	Mate	Population
1	DB9 Serial Header	CSS1FT/CSS4FT	Standard Model Only
2	Mini USB Connector	EXTMUSB3FT/INTMUSB3FT	USB Model Only
3	Alternate Power Connector	PCS	All Models
4	GPO Header	None Offered	All Models
5	Communication/Power Connector	SCCPC5V/BBC	Standard Model Only
6	Alternate USB Connector	None Offered	Custom Only

2.2 Standard Module

The standard version of the LK162B-7T 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^2C protocol. Connections for each protocol can be accessed through the four pin Communication/Power Header as outlined in the Serial Connections and I^2C Connections sections below.

Recommended Parts

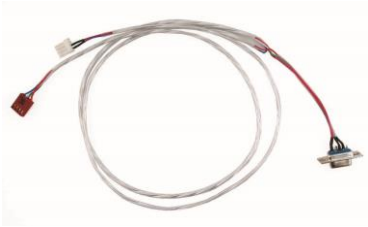

Figure 3: Communication/Power Cable (SCCPC5V)

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

Figure 4: Breadboard Cable (BBC)

For a more flexible interface to the LK162B-7T 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

A serial interface provides a classic connection to the LK162B-7T. The Communication/Power Cable is most commonly used for this set up as it provides connections for DB9 serial and floppy power cables. To place your board in Serial mode, adhere to the steps laid out below.

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

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

2. Make the connections.
    a. Connect the four pin female header of the Communication/Power Cable to the Communication/Power Header of your LK162B-7T.
    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.
    - Use uProject or a terminal program to get started, and then move on with your own development. A number of application notes are available at www.matrixorbital.ca/appnotes.

I^2C Connections

A more advanced connection to the LK162B-7T is provided by the I^2C 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 LK162B-7T in I^2C mode, get started with the guidelines below.

1. Set the Protocol Select switches.
    - I^2C: Ensure that the two I^2C 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 LK162B-7T 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 LK162B-7T-USB offers a single USB protocol for an easy connection to a host computer. This simple and widely available protocol can be accessed using the on board mini B style USB connector as outlined in the USB Connections section.

Recommended Parts

Figure 5: External Mini USB Cable (EXTMUSB3FT)

The External Mini USB cable is recommended for the LK162B-7T-USB display. It will connect to the miniB style header on the unit and provide a connection to a regular A style USB connector, commonly found on a PC.

USB Connections

The USB connection is the quickest, easiest solution for PC development. After driver installation, the LK162B-7T-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 LK162B-7T-USB please follow the steps below.

1. Set the Protocol Select jumpers.
   - USB: The LK162B-7T-USB offers USB protocol only. Model specific hardware prevents this unit from operating in any other protocol, and does not allow other models to operate in USB. Protocol Select jumpers on the USB model cannot be moved.
2. Make the connections.
   - Plug the mini-B header of your External Mini USB cable into your LK162B-7T-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 uProject or a terminal program to get started, and then move on with your own development. A number of application notes are available at www.matrixorbital.ca/appnotes.

3 Software

The multiple communication protocols available and simple command structure of the LK162B-7T 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 number of control characters are also activated. Commands are merely values prefixed with a special command byte, 254 in decimal.

Table 2: Reserved Control Characters

Control Characters							
8	Backspace	10	Line feed / New line	12	Clear screen / New page	13	Carriage return

Once the correct communication port is identified, the following communication settings can be applied to communicate correctly with the LK162B-7T.

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 uProject

The Matrix Orbital alphanumeric display tuner, or uProject, is offered as a free download from the www.matrixorbital.ca support site. It allows the basic functionality of *any display to be tested using a simple graphical user interface system.

While basic functionality can be tested using the GUI portion of the program, more advanced users will enjoy the scripting capability found in the uploader tab. Here commands can be stacked, run, and saved for later use. Although many commands are available to be dragged into the script dialog, perhaps the most powerful is the raw data command found in the other branch.

***Note:** The uProject AutoDetect function will not perform correctly when a USB display is connected. Please manually configure any USB display.

This command allows raw bytes to be sent to the display, permitting many different formats for entry and displaying in decimal notation. Any command from this manual may be entered in decimal notation separated by slashes.

/254/ /88/

Figure 6: uProject Command

Again, the clear screen command is sent to a connected display, this time using uProject raw data command style. Scripts can be run as a whole using the execute command from the script menu, or as single commands by selecting execute once. Before issuing commands, it is a good idea to ensure communication with a display is successful using some of the more basic GUI functions in the main window.

This program provides scratch pad upon which a tome of display projects and ideas can be assembled.

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

Communication/Power Header

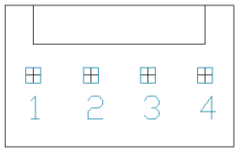

Figure 7: Communication/Power Header

Table 4: Communication/Power Pinout

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

The Communication/Power Header provides a standard connector for interfacing to the LK162B-7T. 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 Voltage Specifications before connecting power. Pins two and three are reserved for serial transmission, using either the RS-232/TTL or clocking data through the I²C protocol, depending on what has been selected by the Protocol Select Jumpers. The versatile Tyco 640456-4-LF style header used can be mated to a number of connectors, the Molex 22-01-3047 for example.

Serial DB9 Connector

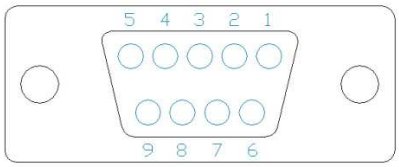

Figure 8: Serial DB9 Connector

Table 5: Serial DB9 Pinout

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

The LK162B-7T provides a DB-9 Connector to readily interface with serial devices using EIA232 standard signal levels. It is also possible to communicate at TTL levels of 0 to +5V by setting the Protocol Select Jumpers to TTL. As an added feature it is also possible to apply power through pin 9 of the DB-9 Connector in order to reduce cable clutter. A standard male DB9 header will provide the perfect mate for this connector.

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

Power Through DB9 Jumper

In order to provide power through pin 9 of the DB-9 Connector you must connect the Power Through DB-9 Jumper labelled D, as illustrated below. This connection can be made using a zero ohm resistor, recommended size 0603, or a solder bridge. The LK162B-7T allows all voltage models to use the power through DB-9 option, see the Voltage Specifications for power requirements.

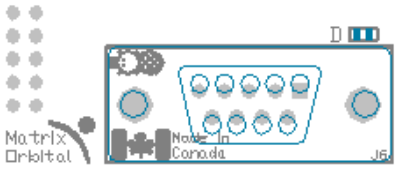

Figure 9: Power Through DB9 Jumper

Protocol Select Jumpers

The Protocol Select Jumpers provide the means necessary to toggle the LK162B-7T 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.

4.2 USB Model

Mini USB Connector

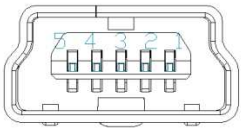

Figure 10: Mini USB Connector

Table 6: Mini USB Pinout

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

The LK162B-7T-USB comes with a familiar Mini USB Connector to fulfill both communication and power needs. The standard MiniB style header can be connected to any other USB style using the appropriate cable. Most commonly used with a PC, this connection creates a virtual com port that offers a simple power solution with a familiar communication scheme.

Alternate USB Header

Some advanced applications may prefer the straight four pin connection offered through the Optional Alternate USB Header. This header offers power and communication access in a simple interface package. The Optional Alternate USB Header may be added to the LK162B-7T-USB for an added charge as part of a custom order. Please use the Contact section to request more information from the friendly Matrix Orbital sales team.

4.3 Common Features

General Purpose Outputs

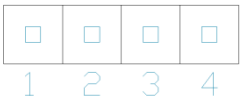

Figure 11: GPO Header

Table 8: GPO Pinout

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

A unique feature of the LK162B-7T is the ability to control relays* and other external devices using either one or six General Purpose Outputs. Each can source up to 10mA of current at five volts when on or sink 20mA at zero volts when off. The two row, fourteen pin header can be interfaced to a number of female connectors to provide control to any peripheral devices required.

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

Alternate Power Connector

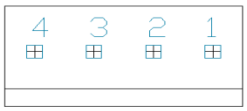

Figure 12: Alternate Power Connector

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 LK162B-7T 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.

***Note:** To apply power via pin 4, move the 0 ohm resistor from R51 to R52. The R52 configuration is better suited to applying 12V power to an extended voltage display.

5 Troubleshooting

5.1 Power

In order for your LK162B-7T 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, make sure that you are using the correct power connector. Standard floppy drive power cables from your PC power supply may fit on the Communication/Power Header; however they do not have the correct pin out to provide power. Matrix Orbital supplies power cable adapters for connecting to a PC, which can be found in the accessories section.
- Next, check the power cable which you are using for continuity. If you don't have an ohm meter, try using a different power cable, if this does not help try using a different power supply.
- If power is applied through the DB9 connector, ensure that the Power Through DB9 Jumper is connected.
- If changes have been made to the protocol select block, ensure all the appropriate protocol select jumpers are connected and all unused protocol jumpers are disconnected.
- The last step will be to check the interface connector in use on your display. If the power connections have become loose, or you are unable to resolve the issue, please Contact Matrix Orbital for more information.

5.2 Display

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

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

5.3 Communication

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

- First, check the communication cable for continuity. If you don't have an ohm meter, try using a different communication cable. If you are using a PC try using a different Com/USB Port.
- Next, please ensure that the display module is set to communicate on the protocol that you are using, by checking the Protocol Select Jumpers.
- In serial and USB protocols, ensure that the host system and display module are both communicating on the same baud rate. The default rate for the display module is 19200 bps.
- Match Rx from your display to the transmitting pin from your host and the Tx pin to the receiving pin.
- If you are communicating to the display via I²C* please ensure that the data is being sent to the correct address. The default slave address for the display module is 80.
- In I²C mode, connect Rx to the clock line of your controller and Tx to the data output.
- Unlock the display. See the Set and Save Data Lock command for more info.
- Finally, you may reset the display to its default settings using the Manual Override procedure outlined below.

***Note:** I²C communication will always require pull up resistors on SCL and SDA of one to ten kilohms.

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. Hold down the bottom left dot key.
3. Reconnect power to your unit, and wait for the start screen before releasing the key.
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
I²C Address	80

Table 8: Manual Override Settings

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

6 Commands

6.1 Communication

1.1 Change Baud Rate				v5.0
	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	Byte	Valid settings shown below.		

Table 9: Accepted Baud Rate Values

Rate	1200	2400	4800	9600	19200	28800	38400	57600	*76800	*115200
Speed	83	41	207	103	51	34	25	16	12	8

1.2 Change I^2C Slave Address				v5.0
	Dec	254 51	Address	
	Hex	FE 33	Address	
	ASCII	■ 3	Address	
Immediately changes the I^2C 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				v5.0
	Dec	254 160	Protocol	
	Hex	FE A0	Protocol	
	ASCII	■ 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 I^2C.		

1.4 Set a Non-Standard Baud Rate				v5.0
	Dec	254 164	Speed	
	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	Word	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 Software Reset	Dec	254 253 77 79 117 110	V7.3
	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	Short	Successful reset response, 254 212.	

6.2 Text

2.1 Clear Screen	Dec	254 88	v5.0
	Hex	FE 58	
	ASCII	■ X	
Clears the contents of the screen.			

2.2 Change the Start Up Screen	Dec	254 64	Characters	v5.0
	Hex	FE 40	Characters	
	ASCII	■ @	Characters	
Changes the message displayed on start up. Custom characters can be included by adding their decimal value (0-7). Characters will automatically wrap on the display.				
Characters	80 bytes, space characters can be added as needed			

2.3 Auto Scroll On	Dec	254 81	v5.0
	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.			

2.4 Auto Scroll Off	Dec	254 82	v5.0
	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.			

2.5 Set Auto Line Wrap On	Dec	254 67	v5.0
	Hex	FE 43	
	ASCII	■ C	
Text will wrap to the next consecutive line once a row becomes full. Default is Auto Line Wrap on.			

2.6 Set Auto Line Wrap Off	Dec	254 68	v5.0
	Hex	FE 44	
	ASCII	■ D	
Text will skip one line when wrapping once a row becomes full. Writing order will be rows 1, 3, 2, and then 4. Default is Auto Line Wrap on.			

2.7 Set Cursor Position	Dec	254 71	Column	Row	v5.0
	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.

2.8 Go Home	Dec	254 72	v5.0
	Hex	FE 48	
	ASCII	■ H	

Returns the cursor to the top left of the screen.

2.9 Move Cursor Back	Dec	254 76	v5.0
	Hex	FE 4C	
	ASCII	■ L	

Moves cursor one position to the left. Cursor will obey wrap settings.

2.10 Move Cursor Forward	Dec	254 77	v5.0
	Hex	FE 4D	
	ASCII	■ M	

Moves cursor one position to the right. Cursor will obey wrap settings.

2.11 Underline Cursor On	Dec	254 74	v5.0
	Hex	FE 4A	
	ASCII	■ J	

Displays a line under the current cursor position. Can be used with block cursor.

2.12 Underline Cursor Off	Dec	254 75	v5.0
	Hex	FE 4B	
	ASCII	■ K	

Removes line under current cursor position.

2.13 Blinking Block Cursor On	Dec	254 83	v5.0
	Hex	FE 53	
	ASCII	■ S	

Displays a blinking block over the current cursor position. Can be used with underline.

2.14 Blinking Block Cursor Off	Dec	254 84	v5.0
	Hex	FE 54	
	ASCII	■ T	

Removes blinking block over current cursor position.

6.3 Special Characters

3.1 Create a Custom Character	Dec	254 78	ID Data	v5.0
	Hex	FE 4E	ID Data	
	ASCII	■ N	ID Data	

Creates a custom character. Each character is divided into 8 rows of 5 pixels; each data byte represents one row. Each byte is padded by three zero bits followed by five bits representing each pixel state. A one represents an on condition while a zero is off. Characters are lost when a new memory bank is loaded, unless they are saved.

ID	Byte	Character ID, value between 0 and 7.
Data	Byte[8]	Character pixel data as shown below.

Table 10: Custom Degree Character

Data[1]	000	p1	p2	p3	p4	p5	00001000	8
Data[2]	000	p1	p2	p3	p4	p5	00010100	20
Data[3]	000	p1	p2	p3	p4	p5	00001000	8
Data[4]	000	p1	p2	p3	p4	p5	00000011	3
Data[5]	000	p1	p2	p3	p4	p5	00000100	4
Data[6]	000	p1	p2	p3	p4	p5	00000100	4
Data[7]	000	p1	p2	p3	p4	p5	00000011	3
Data[8]	000	p1	p2	p3	p4	p5	00000000	0

3.2 Save Custom Characters	Dec	254 193	Bank ID Data	v5.0
	Hex	FE C1	Bank ID Data	
	ASCII	■ ñ	Bank ID Data	

Provides access to all memory banks to create and save custom characters, graph bars, and large digits. Any new characters saved will overwrite the old, so care should be taken when writing to any bar or digit memory bank. Bank structure is shown below.

Bank	Byte	1 byte, memory bank ID, value between 0 and 4, as below.
ID	Byte	1 byte, value between 0 and 7.
Data	Byte[8]	8 bytes, character pixel data as above.

Table 11: Custom Character Banks

0	Start-up Characters	1	Horizontal Bars	2	Vertical Bars	3	Medium Digits	4	Large Digits
---	---	---	---	---	---	---	---	---	---

3.3 Load Custom Characters	Dec	254 192	Bank	v5.0
	Hex	FE C0	Bank	
	ASCII	■ └	Bank	

Loads a bank of custom characters into memory for use. Must be issued before using a bank of characters. Alternatively, an appropriate initialize command can be used.

Bank	Byte	Memory bank ID, value between 0 and 4, as above.

3.4 Save Start Up Screen Custom Characters	Dec	254 194	ID Data	v5.0
	Hex	FE C2	ID Data	
	ASCII	■ ┬	ID Data	

Saves a custom character to memory for the start up screen or repeated use. Start up characters are displayed by sending their ID to the screen.

ID	Byte	Value between 0 and 7.
Data	Byte[8]	Character pixel data, see Custom Degree Character example.

3.5 Initialize Medium Numbers	Dec	254 109	v5.0
	Hex	FE 6D	
	ASCII	■ m	

Loads the medium number custom character bank into memory. Medium numbers must be initialized before use.

3.6 Place Medium Numbers	Dec	254 111	Row Column Digit	v5.0
	Hex	FE 6F	Row Column Digit	
	ASCII	■ o	Row Column Digit	

Places a single medium decimal digit of 2 row height and 1 column width on the display at the position specified. Medium numbers must be initialized before being placed.

Row	Byte	Value between 1 and 20.
Column	Byte	Value between 1 and 4.
Digit	Byte	Single decimal digit to display.

3.7 Initialize Horizontal Bar	Dec	254 104	v5.0
	Hex	FE 68	
	ASCII	■ h	

Loads the horizontal bar graph custom character bank into memory. Horizontal bar characters must be initialized before a graph is displayed.

3.8 Place Horizontal Bar Graph	Dec	254 124	Column Row Direction Length	v5.0
	Hex	FE 7C	Column Row Direction Length	
	ASCII	■ |	Column Row Direction Length	

Places a horizontal bar graph on the screen beginning at the column and row specified. The bar extends either right or left to the length indicated. New bars will overwrite old.

Column	Byte	1 byte, value between 1 and 16
Row	Byte	1 byte, value between 1 and 2
Direction	Byte	1 byte, 0 for right and 1 for left
Length	Byte	1 byte, length in pixels of the graph, value between 0 and 100

3.9 Initialize Narrow Vertical Bar	Dec	254 115		v5.0
	Hex	FE 73		
	ASCII	■ s		
Loads the narrow horizontal bar graph custom character bank into memory. A narrow bar is 2 pixels wide. Horizontal bar characters must be initialized before a graph is displayed.				

3.10 Initialize Wide Vertical Bar	Dec	254 118		v5.0
	Hex	FE 76		
	ASCII	■ v		
Loads the wide horizontal bar graph custom character bank into memory. A wide bar is 5 pixels wide. Horizontal bar characters must be initialized before a graph is displayed.				

3.11 Place Vertical Bar	Dec	254 61	Column Length	v5.0
	Hex	FE 3D	Column Length	
	ASCII	■ =	Column Length	
Places a vertical bar graph on the screen extending from the first row of the column specified. The bar extends upwards to the length indicated. A new bar will over write the old.				
Column	Byte	Value between 1 and 16.		
Length	Byte	Height in pixels of the graph, value between 0 and 16.		

6.4 General Purpose Output

4.1 General Purpose Output On	Dec	254 87	Number	v5.0
	Hex	FE 57	Number	
	ASCII	■ W	Number	
Turns the specified GPO on, sourcing current from an output of five volts.				
Number	Byte	GPO to be turned on.		

4.2 General Purpose Output Off	Dec	254 86	Number	v5.0
	Hex	FE 56	Number	
	ASCII	■ V	Number	
Turns the specified GPO off, sinking current to an output of zero volts.				
Number	Byte	GPO to be turned off.		

4.3 Set Start Up GPO State	Dec	254 195	Number State	v5.0
	Hex	FE C3	Number State	
	ASCII	■ ├	Number State	
Sets and saves the start up state of the specified GPO in non volatile memory. Changes will be seen on start up.				
Number	Byte	GPO to be controlled.		
State	Byte	1 for on or 0 for off.		

LED Indicators

The LK162B-7T has 6 General Purpose Outputs which control 3 bi-colour LEDs. Red, green, and orange-yellow colours can be created using these software controlled GPOs. Odd numbered GPOs control red while even numbers switch the green aspects of the LEDs, as shown in the table below.

Table 12: LED Output

Colour	GPO_O	GPO_E
Yellow	0	0
Green	0	1
Red	1	0
Off	1	1

4.4 Set LED Indicators	Dec	**254 90**	Number Colour	**V7.3**
	Hex	**FE 5A**	Number Colour	
	ASCII	**■ Z**	Number Colour	
Immediately sets the state of the specified LED indicator to a specific colour. Temporary unless remember is on.				
Number	**Byte**	LED indicator to be controlled.		
Colour	**Byte**	LED colour state as below.		

Table 13: LED Indicator Colour

State	Colour
Off	0
Green	1
Red	2
Yellow	3

6.5 Piezo Buzzer

5.1 Activate Piezo Buzzer	Dec	**254 140**	Time	**V7.2**
	Hex	**FE 8C**	Time	
	ASCII	**■ î**	Time	
Activates a 500Hz buzz from the onboard piezo buzzer for a specified length of time.				
Time	**Byte**	Duration of the beep in 100 millisecond increments.		

5.2 Set Keypad Buzzer Beep	Dec	**254 182**	Setting	**V7.2**
	Hex	**FE B6**	Setting	
	ASCII	**■ ╢**	Setting	
Set activation of a 50ms buzz when specific keypad events occur.				
Setting	**Byte**	Keypad event(s) that trigger a 50ms buzzer beep as shown below. Default is 0.		

Table 14: Keypad Buzzer Beep Settings

Setting	Description
0	No Buzz
1	Buzz on Key Press
2	Buzz on Key Release
3	Buzz on Press and Release

6.6 Keypad

6.1 Auto Transmit Key Presses On				v5.0
	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.

6.2 Auto Transmit Key Presses Off				v5.0
	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.

6.3 Poll Key Press				v5.0
	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	Byte	Value of key pressed (MSb determines additional keys to be read).

6.4 Clear Key Buffer				v5.0
	Dec	254 69		
	Hex	FE 45		
	ASCII	■ E		

Clears all key presses from the key buffer.

6.5 Set Debounce Time				v5.0
	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 approximately 52ms.

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

6.6 Set Auto Repeat Mode				v5.0
	Dec	254 126	Mode	
	Hex	FE 7E	Mode	
	ASCII	■ DEL	Mode	

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

Mode	Byte	1 for hold mode or 0 for typematic.

6.7 Auto Repeat Mode Off				v5.0
	Dec	254 96		
	Hex	FE 60		
	ASCII	■ `		

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

6.8 Assign Keypad Codes	Dec	254 213	Key Down Key Up	v5.0
	Hex	FE D5	Key Down Key Up	
	ASCII	■ Ƒ	Key Down Key Up	

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

Key Down	Bytes [9]	Key down values, beginning at row one column one moving right then down.
Key Up	Bytes [9]	Key up values, beginning at row one column one moving right then down.

		Key Down		
		Columns		
		1	2	3
Rows	1	A	B	C
	2	D	E	N/A*
	3	G	H	N/A*

Figure 134: Default Tactile Key Down Values

		Key Up		
		Columns		
		1	2	3
Rows	1	a	b	c
	2	d	e	N/A*
	3	g	h	N/A*

Figure 15: Default Tactile Key Up Values

***Note:** Values are not mapped to a physical key.

6.9 Keypad Backlight Off	Dec	254 155	v5.0
	Hex	FE 9B	
	ASCII	■ ¢	

Turns the keypad backlight off.

6.10 Set Keypad Brightness	Dec	254 156	Brightness	v5.0
	Hex	FE 9C	Brightness	
	ASCII	■ £	Brightness	

Immediately sets the keypad brightness. On time is set using the Backlight On command. Default is 255.

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