![Chipsmall logo]

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

LCD2041
Technical Manual

Revision: 2.0

Contents

1   Introduction

The LCD2041 comes equipped with the following features;

- 20 column by 4 line text display
- Built in font with provision for up to 8 user defined characters
- Speeds from 1200 bps to a lighting fast 19.2 Kbps over RS-232
- Communication over RS-232 or I^{2}C
- Software controlled contrast
- Backlight with configurable time out setting up to 180 minutes
- One general purpose output for a variety of applications
- Horizontal or Vertical bar graphs
- Variable power options, +5V or +8V to +15V
- Extended temperature option
- Fits Matrix Orbital's Dual PC Bay insert without any modifications

1.1   What to Expect From the LCD2041

The LCD2041 is designed as the display unit for an associated controller. The controller may be anything from a single board, special purpose micro-controller to a PC, depending on the application. This controller is responsible for what is displayed on the screen of the display.

The display provides a simple command structure to allow text and bar graphs to be displayed on the screen. Text fonts are built in, and use standard ASCII mapping. Provision is made for up to 8 user defined characters.

The screen is backlit for low light situations. Backlighting may be turned on or off under program control. Contrast is adjustable to compensate for differing lighting conditions and viewing angles.

A general purpose output allows the controller to switch an electronic or electro-mechanical device by issuing commands to the display unit. This can be used for controlling LEDs, relays, etc.

1.2   What Not to Expect From the LCD2041

The display does not include bitmap graphics capability, except that permitted by defining special characters. The display does not include a keypad interface.

1.3   Setup for Testing

Before setting up the application the user may want to try out the display. This is easily done with a PC. If not equipped with a dual bay PC mounting kit, the following will be required;

- A 4-pin power connector of the type used to connect 3.5" floppy drive. Take care not to connect the display to an unmodified spare power connector in a PC
- A spare RS-232 port. (Either COM1 or COM2)
- A 5V power supply. **For wide voltage models, an 8 - 15 V supply will be required**
- A 9 or 25 pin RS-232 serial cable. If using a 25 conductor cable, a 9 to 25 pin adapter will be required
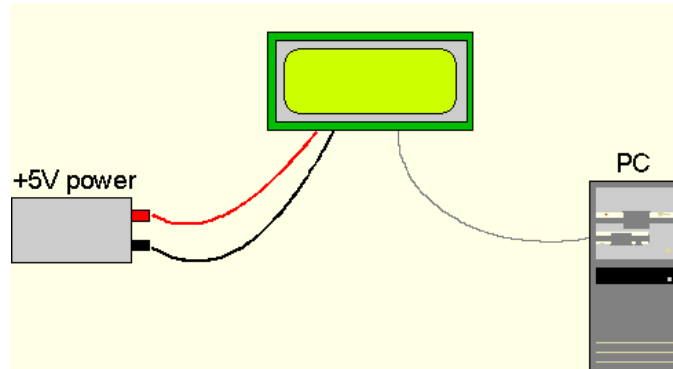
Figure 1: Connections for Testing

1. Refer to the Figure above for the following steps.
2. Wire the connector to the power supply. On most connectors the RED lead will go to +5V and the BLACK lead to GND.

> **NOTE** The Manufacturer's Warranty becomes void if the unit is subjected to over-voltage or reversed polarity.

3. Connect the display to the PC using the serial cable and adapter if required. Make sure the RS-232 cable includes the required ground lead. There must be no voltage differential between the RS-232 ground and the power supply ground.
4. Connect the power connector, making sure that the +5V goes to V+ . Turn on the power; the display backlight should come on and a blinking cursor at the top left should appear.

1.4 Trying out the LCD2041

The unit should be connected to power and the PC and backlight should be on. To experiment with typing text, run a PC program such as Display Tuner or AlphaDemo. Make sure it's configured to use the correct port. Set the baud rate to 19,200.

To exercise some of the other features of the display, a program (in any convenient language such as Basic or C) will need to be written in order to issue the required command strings. Most terminal programs are unable to issue the 0xFE character needed as a command prefix.

2 Connections

2.1 Connector Pinout

Refer to the Figure below for this chapter.
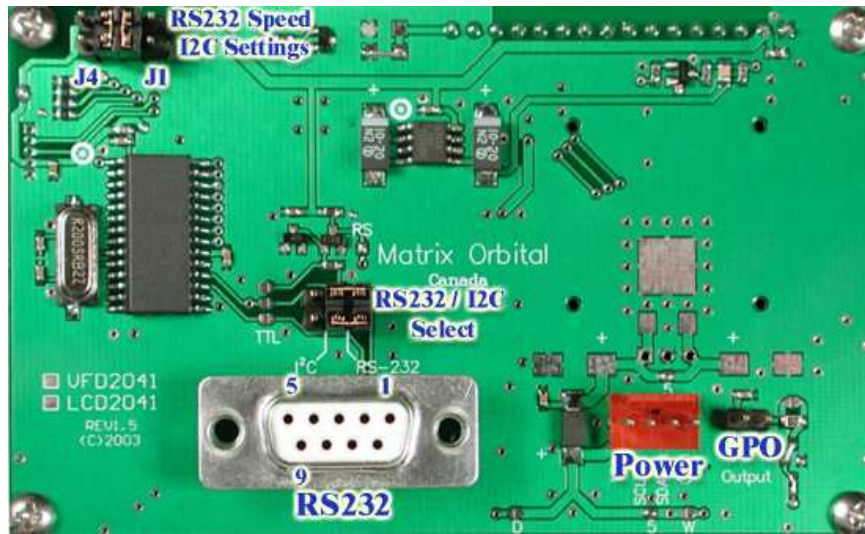
Figure 2: Electrical Connections

The display has three connectors;

Table 1: Connectors & Functions

Connector	Function
2 pin	General purpose output
4 pin	Power and I^2C communication
DB-9F	RS-232/power

2.1.1 Power Connection

Power is applied via pins 1 and 4. Power requirement for standard units is +5 VDC $\pm$0.25V . Units with the wide voltage range option require 8 - 15 VDC.

WARNINGS

- Do not apply any power with reversed polarity. Do not apply any voltage other than the specified voltage. Do not use any cables other than the cables supplied by Matrix Orbital, unless aware of the modifications required. Do not apply voltage to the DB-9 connector AND power connector Do not apply more than +5Vdc to pin #9 on the DB-9 connector.

Connector pinout is as follows;

Figure 3: Power Connector

Table 2: Pinout Connections

Pin 4	Ground
Pin 3	SDA (I^2C data) / Rx
Pin 2	SCL (I^2C clock) / Tx
Pin 1	Vdc

2.1.2 Five Volt Modules

If the display is used in a PC it is tempting to plug a spare power connector into the unit. **Don't do this!** Wiring for the PC power connector and that required for the display are different.
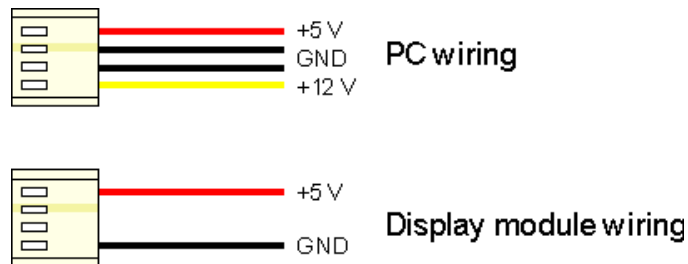
Figure 4: Wiring for 5V Modules

Matrix Orbital is equipped to supply an adapter cable designed to use with the LCD2041 when it's installed in a PC. The cable is wired as shown in the Figure below.

NOTE   This cable does not provide connections for I^2C.

Figure 5: Five Volt Power Cable

2.1.3 Wide Voltage Range Modules

NOTE   Do not use this cable unless the display module has the "Wide voltage range" option. Use of the 12 volt power cable with 5 volt modules will damage the module.

The 12 volt power cable is designed for use with wide voltage range display modules mounted in a PC. Wiring required for the 12 volt power connector is shown in the Figure below.
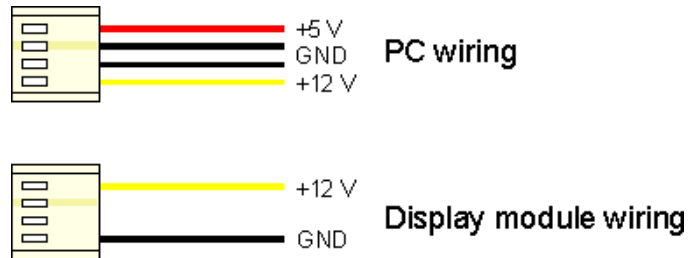
Figure 6: Wiring for 12 Volt Modules

Matrix Orbital is equipped to supply an adapter cable designed to use with the display module when it's installed in a PC. The cable is wired as shown in the Figure below.

Figure 7: Twelve Volt Power Cable

2.1.4 Applying Power through the DB-9 Connector

Power may be provided to the module by pin 9 of the DB-9 connector instead of through the 4-pin SIP. If power is to be applied using the DB-9, it must be a regulated 5Vdc supply.

NOTE This applies to wide voltage units, V and VPT extensions, as well as to standard 5 volt units.
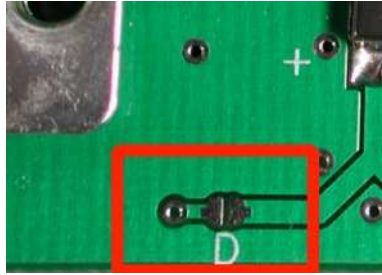
Figure 8: DB-9 Power

To use pin 9 as the power source, the user must solder the 5 volt jumper pad beside the DB-9 connector.

WARNING Application of a voltage to pin 9 greater than 5.25 volts may cause immediate destruction of unit and void the Manufacturer's Warranty.

2.2 RS-232 Communications

The RS-232 connector on the PC cable is wired so that a standard 'straight through' 9 pin D-sub cable may be used to connect the module to a standard serial port such as COM ports on PCs. Please note that this device complies with the EIA-232 standard in that it uses signal levels from $\pm$ 12V to $\pm$ 12V and can be converted to TTL levels as well. The display is set at 19.2Kbps default speed. Other settings are;

- 8 bits, no parity, 1 stop bit.

2.2.1 DB9 Connections

A standard DB-9F is provided for RS-232 communications. Power may also be supplied via this connector if desired.
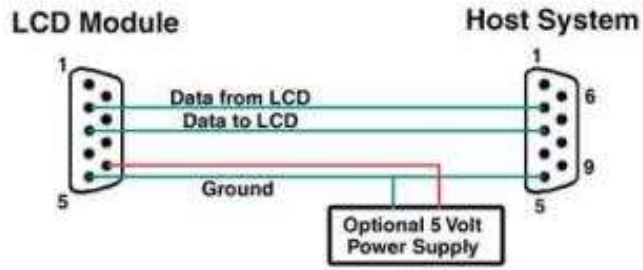
Figure 9: RS-232 and Power Connector

Table 3: RS-232 Pinout

Pin Number	Direction	Description	LCD	Host
2	Data from LCD	Data out (LCD	Tx	Rx
3	Data to LCD	Data in (LCD)	Rx	Tx
5	-	Ground	gnd	gnd

2.2.2 Alternate Serial Connection

An optional alternate connection header is provided to allow one cable for data and power. Two jumpers will have to be set into the RS-232 position above the DB-9 connector.

Figure 10: RS232 Configuration

Connector pinout is as follows;

Pin 4	Ground
Pin 3	Rx
Pin 2	Tx
Pin 1	Vdc

Figure 11: Power Connector

2.2.3 TTL communications

The display can be talked to at TTL (logic 0V to +5V) levels. The following three jumps will need to be modified as shown in figure below. This will allow TTL communications through the 4-pin SIP header or the DB9 connector.
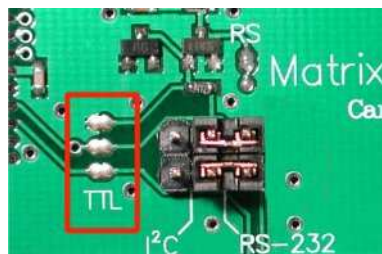
Figure 12: TTL Communications

2.2.4 Configuring RS-232 and I^{2}C

RS-232 baud rate and I^{2}C address are configured by means of jumpers.
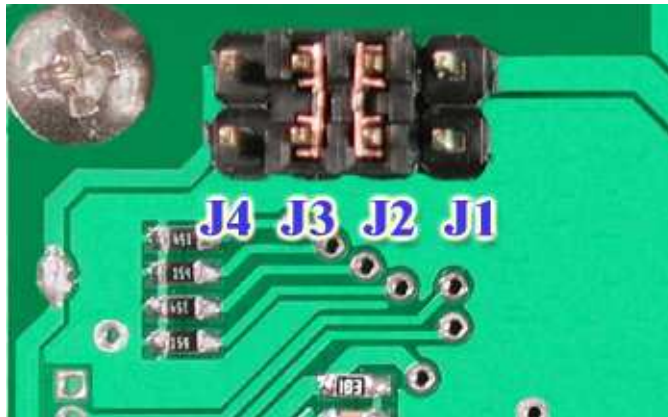
Figure 13: RS-232 Jumpers

The module is supplied with jumpers J1 and J2 installed, which gives an RS-232 baud rate of 19200 and an I^2C address of 0x5C.

- RS-232 port: J0, J1, J2 - control baud rate. RS-232 format is 8N1, 8 bits, no parity, one stop bit
- I^2C port: J0, J1, J2, J3 - sets slave peripheral address

Table 4: Serial Baud Rates

Baud Rate	Slave Address	J3	J2	J1	J0
1200	50H	out	out	out	out
	52H	out	out	out	in
2400	54H	out	out	in	out
	56H	out	out	in	in
9600	58H	out	in	out	out
	5AH	out	in	out	in
19200	5CH	out	in	in	out
	5EH	out	in	in	in
1200	60H	in	out	out	out
	62H	in	out	out	in
2400	64H	in	out	in	out
	66H	in	out	in	in
9600	68H	in	in	out	out
	6AH	in	in	out	in
19200	6CH	in	in	in	out
	6EH	in	in	in	in

2.3 I²C Communications

I²C communications runs at 100Kbps and supports up to 16 units on a single communications line. The I²C data line operates on 5 volt CMOS levels.

2.3.1 ACK

The idea of ACK is to indicate when the data has been received correctly. ACK does not indicate data incorrectly received. ACK simply fails to indicate when data is correctly received. Clearly, this is of limited usefulness and even less so with Matrix Orbital modules. Matrix orbital modules are not capable of failing to acknowledge an incorrectly received byte in response to that bytes transition. They are only capable of failing to acknowledge the bytes following the byte, which was not received. To fully understand the reasons for this one needs to understand something about how a Matrix Orbital module processes data. Basically the reason why a Matrix Orbital module might fail to receive a byte correctly is that it was unable to process the byte previous before the failed byte was transmitted. Because the module cannot possibly know that it would be unable to store the byte before the next byte was received it cannot know to not ACK. The reason for this situation in deference to situations one might be familiar with (i.e., memory chips, etc) is that the Matrix Orbital module employs a micro-processor to perform these data storage functions. A memory chip takes care of these things entirely with in hardware subsystems that operate at the same speed as the transmission themselves.

The display uses a standard Phillips 7bit address as defined by Phillips. However, Matrix Orbital specifies I²C address in 8bits. The 8th bit, or least significant bit, LSB or Low Order Bit of the 8bit address is a read / write bit. If we take a standard Phillips 7bit address of 45hex this would be in binary, 1000101. This is 7bits. Matrix Orbital would describe the Phillips I²C address of 45hex as 8Ahex. The read address would be 8Bhex.

For more information on Phillips I²C please visit;

`http://www.ping.be/~ping0751/i2cfaq/i2cindex.htm`

And for more information specific to our displays visit;

`http://www.lcdforums.com/forums/viewtopic.phpt=910`

To communicate by I²C two jumpers have to be set into the I²C position above the DB-9 connector.
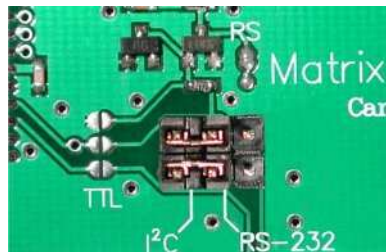
Figure 14: I²C Communications

Connector pinout is as follows;

Pin 4	Ground
Pin 3	SDA (I^2C Data)
Pin 2	SLC (I^2C Clock)
Pin 1	Vdc

Figure 15: Power Connector

2.4 General Purpose Output

The display has a general purpose output which can be used to control relays or other electronic devices. This allows external devices to be turned on or off using your PC or controller and software commands. The + terminal is connected to the module positive supply, the - terminal is connected through a 240ohm current limiting resistor and the electronic switch to ground.
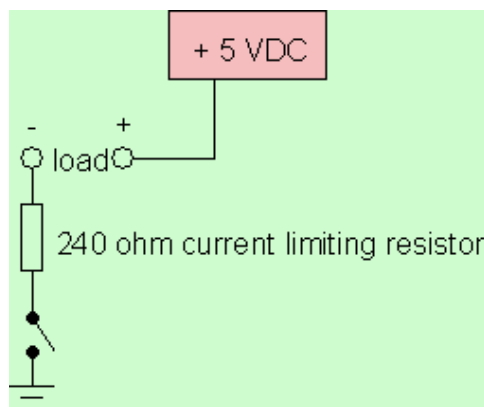
Figure 16: General Purpose Output

Maximum allowable current is 20mA, which is enforced by the current limiting resistor. If the device being switched has a resistance of 240ohms or more the corresponding resistor may be shorted. Solder a small jumper wire or wirewrap wire between the two feed through holes.

Matrix Orbital   LCD2041   **11**
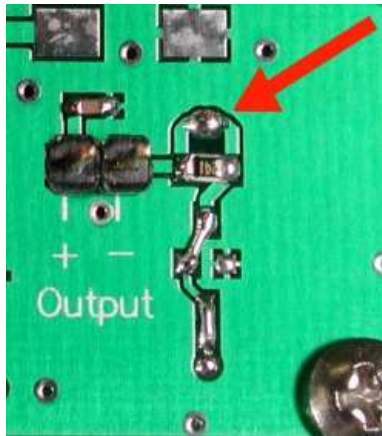
Figure 17: Bypassing 240 ohm Resistor

NOTE The GPOs do not have any over current or over / under voltage protection so care must be taken when using them. For instance if the external device is a relay it must be fully clamped using a diode and capacitor to absorb any generated back electro-motive force or "EMF".
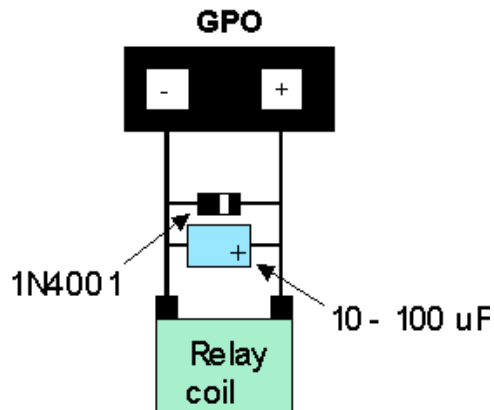
Figure 18: Clamping a Relay

3 Displaying Text

This chapter describes the various text-display commands in detail. **Before issuing commands to the LCD2041 please read sections 6.2 and 6.3**.

3.1 General

Text is displayed on the display using the built in 5x7 dot matrix font, in addition to up to 8 user defined characters.

3.2 The built in Character Font

The display includes a built in 5x7 dot matrix font with the full range of ASCII characters plus a variety of extended characters, as shown in the Figure below.

Figure 19: Character Set

In addition to the built in characters, users may define up to 8 special characters which, once defined, occupy positions 0x00 to 0x07 in the above chart. The display does not have provision to download other

fonts.

3.3 Writing Text to the Display

When the display receives a character, it displays that character at the position currently defined. The next character sent to the module then advances to the following position on the display. Characters are drawn using the built in font, and only characters defined in the font are actually displayed. Characters that are not defined by the built in font print as a space. The position where text is to be inserted is a character location stored in the display's volatile memory and maintained internally by the display's firmware. This position is manipulated by the commands shown in the following section.

3.4 Text Commands

In this section commands are identified by their names and decimal values.

3.4.1 Auto line wrap on (254 67)

Enables automatic line wrapping. This is not 'word wrapping' and wraps may occur in the middle of a word. If auto line wrap and auto scroll are both off (default) text will wrap from line 1 to line 3 then 2 then 4. Factory default is OFF.

3.4.2 Auto line wrap off (254 68)

Disables automatic line wrapping.

3.4.3 Auto scroll on (254 81)

When auto scrolling is on the display shifts the entire display's contents up to make room for a new line of text when the text reaches the scroll position at the bottom right character position.

3.4.4 Auto scroll off (254 82)

When auto scrolling is disabled, text will wrap to the top left corner of the display area. Existing text in the display area is not erased before new text is placed. A series of 'spaces' followed by a "Cursor home" command may be used to erase the top line of text.

3.4.5 Set cursor position (254 71 [column] [row])

This command sets the cursor position (text insertion point) to the [column] and [row] specified. Columns have values from 1 to 20 (0x01 to 0x14) and rows have values of 1 to 4 (0x01 to 0x04).

3.4.6 Send cursor home (254 72)

This command moves the cursor position (text insertion point) to the top left of the display area.

3.4.7 Turn on underline cursor (254 74)

Turns on the underline cursor. The cursor shows the current text insertion point. Both underline and blinking cursors may be turned on or off independently. The cursor is off by default.

3.4.8 Turn off underline cursor (254 75)

Turns off the underline cursor. Does not affect the blinking block cursor.

3.4.9 Turn on block (blinking) cursor (254 83)

Turns on the blinking block cursor. The cursor shows the current text insertion point. Both blinking and underline cursors may be turned on or off independently. The cursor is off by default.

3.4.10 Turn off block (blinking) cursor (254 84)

Turns off the blinking block cursor. Does not affect the underline cursor.

3.4.11 Cursor left (254 76)

Moves the cursor one position to the left but does not erase any character that may be in that position. Note that this command moves the text insertion point even if the cursor is turned off.

NOTE A 'destructive backspace' which erases the character to the left of the original position, may be done by issuing the following sequence: cursor left, space, cursor left.

3.4.12   Cursor right (254 77)

Moves the cursor one position to the right but does not erase any character that may be in that position. This command moves the text insertion point even if the cursor is turned off.

4   Bar Graphs and Special Characters

The display includes the ability to draw bar graphs, either horizontally or vertically, large numbers, and allows users to define up to eight special characters. Eight characters (ASCII values 0x00 to 0x07) are set aside for use with bar graphs, user defined characters, and large digits. Since the same 8 characters are used for each function, the functions may not be used simultaneously. The characters may be defined or redefined at any time by issuing the commands shown in this section. Once defined, they may be used either by means of the bar graph commands, or by simply issuing one of the ASCII values 0x00 to 0x07 which is not prefixed by the command byte, 254.

4.1   Command List

4.1.1   Initialize wide vertical bar graph (254 118)

This command defines the 8 special / user characters to be blocks suitable for use in drawing wide vertical bar graphs, such as 5 pixels. Any previously existing definitions will be lost. Once this command has been issued, any number of vertical bar graphs may be drawn unless the characters are redefined by another command.

4.1.2   Initialize narrow vertical bar graph (254 115)

This command defines the 8 special / user characters to be blocks suitable for use in drawing narrow vertical bar graphs, such as 2 pixels. Any previously existing definitions will be lost. Once this command has been issued, any number of vertical bar graphs may be drawn unless the characters are re-defined by another command.

4.1.3   Draw vertical bar graph (254 61 [column] [height])

Draws a vertical bar graph in [column] having a height of [height] pixels. The height may range from 0 to 20 (0x00 to 0x14) pixels. The necessary characters must first be initialized by either of the commands shown in section 5.1.1 or 5.1.2, which will determine the width of the graph drawn. Graph may be erased by drawing a bar graph of height = 0 in the same column.

4.1.4   Initialize horizontal bar graph (254 104)

This command defines the 8 special / user characters to be blocks suitable for use in drawing horizontal bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of horizontal bar graphs may be drawn unless the characters are redefined by another command.

4.1.5   Draw horizontal bar graph (254 124 [column] [row] [dir] [length])

Draws a horizontal bar graph in [row] starting at [column] with a length of [length] pixels. [row] may have a value of 0x01 or 0x02, column may range from 0x01 to 0x14 and length may be from 0x00 to 0x64 (0 to 100) if the graph can extend the full width of the screen. Each column is 5 pixels wide, as spaces between the columns don't count.

[dir] specifies the direction: 0x00 goes from left to right, 0x01 goes from right to left.

4.1.6   Initialize large digits (254 110)

This command defines the 8 special / user characters to be blocks suitable for use in drawing large digits. Any previously existing definitions will be lost. Once this command has been issued, any number of large characters may be placed until the characters are redefined by another command.

4.1.7   Place large digit 254 35 [col] [digit]

This command allows the large digits to be drawn on the display screen. Numbers of almost full display height may be placed along side regular text on four row displays. The column number has a maximum value which is less than the display width because the digits are all three columns wide.

Before using this command the "Initialize large digits" command must be issued to define the blocks necessary to make up the digits. If regular text and large digits are mixed on one screen, the user should always set the display cursor position before placing regular text because the creation of a large digit will leave the cursor position to the bottom right of the large digit and not at the last regular text write position.

[col] can have values from 0x01 to 0x12 (1 to 18). [digit] has values from 0x00 to 0x09 (0 to 9).

4.1.8   Define custom character (254 78 [c] [8 bytes])

The display allows up to 8 user defined 'custom' characters. These characters occupy the first 8 (0x00 to 0x07) places in the character set.

Custom characters occupy a 5x8 pixel matrix. Built in characters are 5x7: the bottom row of pixels is normally reserved for the underline cursor. The underline cursor should be turned off if the bottom row of pixels forms part of a custom character.

The characters are defined by issuing the command 254 78 [c] followed by 8 bytes to define the character. [c] is the character number (0x00 to 0x07). The 8 bytes are mapped as shown below;

Table 5: 8 Byte Map

MSB LSB								
*	*	*	1	2	3	4	5	Data Byte 1
*	*	*	6	7	8	9	10	Data Byte 2
*	*	*	11	12	13	14	15	Data Byte 3
*	*	*	16	17	18	19	20	Data Byte 4
*	*	*	21	22	23	24	25	Data Byte 5
*	*	*	26	27	28	29	30	Data Byte 6
*	*	*	31	32	33	34	35	Data Byte 7
*	*	*	36	37	38	39	40	Data Byte 8

A "1" bit indicates an on (black) pixel, a "0" bit indicates an off (clear) pixel.

Once defined, a character is displayed simply by issuing a value (0x00 to 0x07) corresponding to the character number. The character will be laid out as follows;

Table 6: Character Values

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
Cursor Line				

NOTE   Custom characters will be erased if any of the "Initialize bar graph" commands are issued.

Example of a degree symbol;

```
txUart.sendByte(0xFE); //command prefix
txUart.sendByte('N'); //custom character command
txUart.sendByte(0x00); //custom character value 0-7
txUart.sendByte(12); //8 bytes to create
txUart.sendByte(18); //the custom character
txUart.sendByte(18);
txUart.sendByte(12);
txUart.sendByte(0);
txUart.sendByte(0);
txUart.sendByte(0);
```

```
txUart.sendByte(0);
txUart.sendByte(0x00); //display custom character 0
```

5   Miscellaneous Commands

The commands listed in this chapter don't readily fit in any of the other categories, or are used in more than one category.

5.1   Command List

5.1.1   Clear display (254 88)

This command clears the display and resets the text insertion point to the top left of the screen.

5.1.2   Set contrast (254 80 [contrast])

This command sets the display's contrast to [contrast], where [contrast] is a value between 0x00 and 0xFF (between 0 and 255). Lower values cause 'on' elements in the display area to appear lighter, while higher values cause 'on' elements to appear darker. Lighting conditions will affect the actual value used for optimal viewing. Individual display modules will also differ slightly from each other in appearance. In addition, values for optimal viewing while the display backlight is on may differ from values used when backlight is off.

5.1.3   Backlight on (254 66 [minutes])

This command turns on the backlight for a time of [minutes] minutes. If [minutes] is zero (0), the backlight will remain on indefinitely. The factory default for backlight is on.

5.1.4   Backlight off (254 70)

This command turns the backlight of the display off.

5.1.5   General purpose output off (254 86)

This command turns OFF any of the general purpose outputs. [gpo #] is 1 to 8. Please note that OFF means the GPO pin floats (high impedance).

5.1.6   Set backlight brightness (254 153 [brightness])

This command sets the display's brightness to [brightness], where [brightness] is a value between 0x00 and 0xFF (between 0 and 255). The higher the number is, the brighter the backlight will appear.

5.1.7   General purpose output on (254 87)

This command turns ON the general purpose output. ON means that the output is pulled low or, ground via 240ohms.

5.1.8   Read module type

This command will return, over the RS-232 interface in TTL levels only, the model type value of the module. This command returns a 1-byte hex value. Values for various modules at the time of this publication are as follows;

Table 7: Module Values

LCD0821 - 0x01	LCD2021 - 0x03	**LCD2041 - 0x05**
LCD4041 - 0x07	LK202-25 - 0x08	LK204-25 - 0x09
LK404-55 - 0x0A	VFD2021 - 0x0B	VFD2041 - 0x0C
VK204-25PC - 0x0E	VK204-25 - 0x0F	GLC12232 - 0x10
GLC24064 - 0x13	GLK24064-25 - 0x15	GLK12232-25 - 0x22
LK404-AT - 0x31	LK402-12 - 0x33	LK162-12 - 0x34

6   Appendix; Command Summary

6.1   General

The operation of the display is controlled by a simple and consistent command set. Commands control;

- Text
- Display graphics
- Display keypad interface
- Miscellaneous operating parameters

6.2   Issuing Commands

Commands are issued to the display by the controller. In a test setup, commands can be issued to the display by means of a BASIC program, using the chr$( ) function. In the tables below, we've shown

commands in hex, ASCII and decimal form. All commands begin with the prefix character 0xFE (254 decimal). These commands are issued on the serial communications link (I^{2}C or RS-232) at the currently defined baud rate.

For example (using BASIC in a test setup), the user could issue the command to clear the screen on the display by including the line:

```
PRINT#1,chr$(254);chr$(88)
```

in the BASIC program.

```
ZComm1->WriteCommByte(0xfe);
ZComm1->WriteCommByte('X');
```

6.3   On Numbers

Like all computerized devices, the display operates with commands and values in the form of binary numbers. These binary numbers are arranged in 8 digit (i.e., 8 bit) groups called bytes. The decimal value of a byte may have any value from 0 to 255. Bytes are usually specified in either decimal or hexadecimal (base 16) form for convenience, since binary numbers are confusing to deal with directly. Hexadecimal (hex) numbers are particularly convenient because exactly two hexadecimal digits make up one byte, each hex digit representing 4 binary digits (4 bits) as shown here;

Table 8: Hex Value Table

Binary	Hex	Decimal	Binary	Hex	Decimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

Based on the table, the byte 01001011 can be represented in hex as 4B, which is usually written as any of 4Bh, 4BH, 4B hex or 0x4B. The numbers can also be expressed in decimal form if preferred.

6.3.1   ASCII Characters

Since computers deal internally with numbers only, but externally with both letters and numbers, several schemes were developed to 'map' written characters to numeric values. One such scheme has become universal, the American Standard Code for Information Interchange, or ASCII. ASCII tables are readily available from a number of sources. A few examples will do here;

Table 9: Example of an ASCII Table

Letter/ Number	Decimal Value	Hex Value
A	65	41
a	97	61
0	48	30
9	57	39

This gives rise to the possibility of confusion when parameters are being set on the display. For example, the GPO ON and OFF commands use a number to indicate which GPO is being controlled. We're told that acceptable values are 0 to 8. All such parameters must use numeric values (i.e., the actual byte values). If we send the ASCII number 0 by mistake it will actually give the value 48 decimal (30 hex) to the parameter, which is wrong.

In the tables given in the following sections ASCII characters are shown as 'A', with single quotes.

6.4 Text Commands

Syntax in the tables below is given in hex, decimal and decimal with ASCII, in that order, one per line.

Table 10: Text Commands

Command	Syntax	Default	Notes
Auto line wrap on	FE 43 254 67 254'C'	off	Enables line wrapping (not word wrap).
Auto line wrap off	FE 44 254 68 254 'D'	off	Disables line wrapping.
Auto scroll on	FE 51 254 81 254 'Q'	off	Enables scroll at bottom of screen. Text will push display up one line to make room for new line.
Auto scroll off	FE 52 254 82 254 'R'	off	Disables auto scroll. Text will wrap to top left and overwrite existing text.