User Manual

ezLCD-3xx Product Family
Version 1.4 June 20, 2012
Requires Firmware Version 1.40 or Later

EARTHLCD.COM
We Make LCDs Work™

Table of Contents

1.0 Introduction

The ezLCD-3xx reflects the most intense effort of our 18 year history in the LCD industry and 9th year of ezLCD production. We hope you are as excited about this product as we are! I'd personally like to dedicate this manual to Michal Sieluzycki, our first ezLCD engineer. He started this product line in 2003 with his winning of a Circuit Cellar Design contest, submitting a design that used an 8 bit micro to drive a color TFT display. Michal passed on to the "big lab in the sky" in the spring of 2011. I know he's probably smiling down at us as he adapts the ezLCD-3xx into that CNC mill he was always tinkering with in his garage. We hope that you enjoy using your ezLCD-3xx as much as we've enjoyed creating it!

-Randy Schafer

EarthLCD.com CEO & Fire Starter

2.0 About This Manual

Congratulations on your purchase of your ezLCD-3xx, the easiest way to embed a color LCD with (or without) touchscreen into your existing application, project or new product design. Note while this manual refers to ezLCD-3xx it is a family manual for the entire ezLCD-3xx family of products. The Appendix will describe the different models. All ezLCD-3xx models support the same I/O connector pin out and command set. The difference is the LCD panel size, resolution, number of displayable colors, and whether a touchscreen is included. The ezLCD-3xx is the third generation of ezLCD developed by EarthLCD.com, a dba of Earth Computer Technologies, Inc. This manual contains software, hardware and driver installation instructions and the ezLCD-3xx command list. This manual assumes you are running Microsoft Windows 7 or Windows XP SP3 on your computer system. For and MAC OS X 10.7 (Lion) see Appendix F. For Linux visit **www.EarthLCD.com/ezLCD-30x** for more information.

We've written this manual to introduce a whole new generation of ezLCD products not just to our existing customer base, but also to the Arduino enthusiast, the Maker crowd and engineering students who are excited about making their projects as dynamic and exciting as the smart phone they carry in their pocket. Advanced users may want to go straight to section 4.0. If so, check and see if there is an application note for your host micro on the ezLCD-30x product page at **www.EarthLCD.com/ezLCD-30x**.

2.1 One Hundred Dollars - The e.z. way!

Technical documentation here at EarthLCD is a continuous process. Our goal is to provide easy to use and well documented products. Over our nearly 20 year history our best ideas have come from our customers. We appreciate your suggestions. Please email docs@earthlcd.com with the title of this manual in your subject line and give us suggestions for making the manual better or general corrections and you will be entered into a quarterly drawing for $100 Earth purchase credit!

3.0 How the ezLCD-30x Works

The ezLCD-3xx Smart LCD consists of an LCD module and a controller board containing the graphics processor, memory and interfaces. The ezLCD-3xx contains USB, serial ports, I2C, SPI and I/O pin interfaces. A 4 megabyte USB flash drive on the controller board is used for storing macros, fonts, and images. The drive also includes drivers, utilities and product documentation. To develop projects and configure the ezLCD-3xx, you simply need a terminal program running on a computer set to 115,200 baud rate, 8 data bits, no parity, one stop bit, local echo and CR=CR+LF. Plugging the ezLCD-3xx into a USB port achieves the following:

- **Powers the ezLCD-3xx**
- **Connects the ezLCD USB flash drive to your computer**
- **Opens a USB CDC COM port connection**

The ezLCD-3xx is driven by ASCII commands sent to the Command Port. The Command Port can be either the USB CDC device or one of two serial ports on the ezLCD I/O connector.
[Note: By default the Command Port is set to USB by the STARTUP macro in the \SYS\MACROS directory of the ezLCD-3xx FlashDrive]
Eventually, the ezLCD-3xx will be capable of running as a standalone controller. However, many ezLCD-3xx customers will use the ezLCD-3xx as the user interface in their design and use a dedicated microcontroller chip or board (PIC, ARM, AVR, Arduino, BASIC Stamp, SBC) to do their control functions. The microcontroller would typically communicate to the ezLCD-3xx through a serial port. The ezLCD-3xx is designed to require the least amount of system-dependent software in order to develop programs as quickly as possible. ASCII commands allow any standard terminal program to talk to the ezLCD for demonstrating and learning. By configuring a terminal program to talk to the ezLCD CDC Device (COM Port) you are able to use your PC to send commands directly to the Command Port. The Flash Drive allows for bitmaps, macros and fonts to be stored on the ezLCD-3xx for rapid access. This makes graphics performance independent of host speed.
A standard USB flash drive interface is automatically configured on most computers with a USB port using the built-in MSD driver. The serial interface uses a built-in CDC driver when connected through the USB. The CDC driver is already installed in most computers. Under Windows the driver only requires the **EarthLCD.inf** file (which is included on the ezLCD-3xx flash drive) for configuration.
Unlike LCD's with built in frame buffers, the ezLCD is a full blown smart LCD client. With it's versatile programmability, built in widgets, flash based fonts and bitmaps you can create an analog meter readout for your project in minutes while only using a 100 bytes of your host microcontroller board! Performance is not limited by your host!

4.0 Installation and Getting Started

You will need the following before proceeding:

- **ezLCD-3xx Smart LCD**
- **ezLCD-3xx USB Cable or an ezLCD-30x EDK board with USB cable**
- **A computer with a USB connection (Host)**

The 7 steps to install your ezLCD-30x are:

4.1 Connect the ezLCD-3xx USB to Your PC

4.2 Verify the ezLCD-3xx USB Flash Drive Operation

4.3 Install the USB CDC Driver

4.4 Run the Terminal Program

4.5 Verify Connection

4.6 Flash Drive Access

4.7 "Hello Earth", Your First ezLCD Program

4.1 Connect the ezLCD USB to Your PC

Connect the USB cable to the ezLCD-30x and then to your PC. The ezLCD-30x will power up and display the splash screen (Figure 1). The splash screen appearance will vary depending on your firmware version and ezLCD model. Figure 1 is for a ezLCD-301 with firmware 1.1 and file system E.

Figure 1

4.2 Verify the ezLCD USB Flash Drive Operation

When you plug the ezLCD-30x into your PC, a window labeled **AutoPlay** (Figure 2) will appear on your computer screen. Select the **Open Folder to View Files** option.

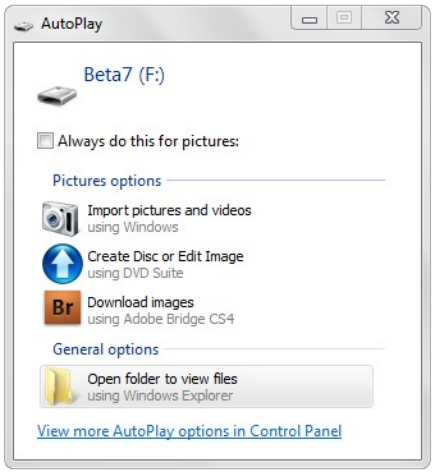


Figure 2

After clicking on **Open Folder to View Files**, you will see a directory for the contents of your ezLCD-30x flash drive. This verifies that the USB flash drive is connected.

4.3a Installing the USB Driver on a Windows 7 Operating System

Note: You must be the computer's administrator or have the password to install windows drivers.

Once the ezLCD-30x fails to install automatically, open up the **Device Manager**. The device should be listed under **Other Devices** with an exclamation mark next to EarthLCD ezLCD-30x. Right click this item and select **Update Driver Software**.

On the next screen, select **Browse my computer for driver software**.

Next, click the **Browse** button and select the flash drive labeled ezLCD-30x that was automatically installed earlier. Click the **OK** button and click the **Next** button. This will begin installing the software.

Windows Security may prompt you that Windows can't verify the publisher of this driver software. Select the **Install this driver software anyway** option. After a moment, the device should be installed successfully.

When you click the **close** button, the device manager should display your device with a COM port in parenthesis next to it (Figure 3). Make a note of this for the next step.

4.3b Installing the USB Driver on a Windows XP Operating System

The **Welcome to Found New Hardware Wizard** will come up first. Click **Close** to exit the install. The Flash Driver will install automatically and the **Autoplay** window will come up. Close the **Autoplay** window and open up the **Device Manager**. The device should be listed under **Other Devices** with an exclamation mark next to Earth LCD ezLCD-30x. Right click this item and select **Update Driver**.

On the following screen, select **No, not this time** and click the **Next** button.
Select **Install from a list or specific location** and click the **Next** button.
Click the **Browse** button and select the flash drive labeled ezLCD-30x that was automatically installed earlier.
Click the **OK** button and click the **Next** button. This will begin installing the software.
The **Hardware Installation** may prompt you that the device has not passed Windows Logo testing to verify its compatibility with Windows XP. Click the **Continue Anyway** button. After a moment, the device should be installed successfully.
When you click the **Finish** button, the device manager will display your device with a COM port in parenthesis next to it (Figure 3). Make a note of this COM Port number to use in configuring the Termie Terminal program in Chapter 4.4.

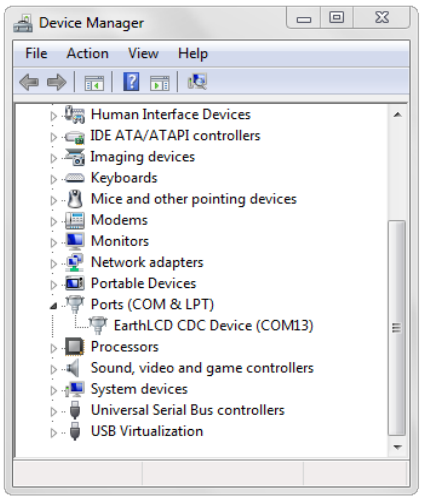


Figure 3

4.4 Run the "Termie" Terminal Program

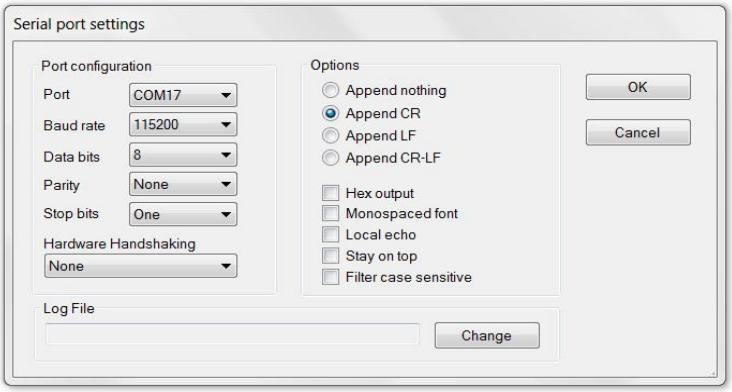


Figure 4

Locate the "TERMIE.EXE" serial terminal software program which is located on the ezLCD-30x USB flash drive and copy the program to your PC then click on it to Run it. Click **settings** and select the COM **Port** number discussed in section **4.3** (Figure 4). Set the **Baud rate** to **115200**, the **Data bits** to **8**,

the **Parity** to **None**, the **Stop bits** to **One,** and make sure to uncheck **Monospaced font** and **Local echo**. Select **OK**.
For purposes of this manual it is assumed that you are using 'Termie', but most other ASCII terminal programs will work as long as you use the same comparable settings .
If you do not remember the COM port, look in **Control Panel/Device Manager/Port** (Figure 3).

4.5 Verify Connection

At the bottom of the Termie program window you will see a **SEND** text box. Type **CLS** in that box and press **Enter**. The ezLCD-30x screen will turn black. Type **PLAY STARTUP** and press **Enter**. The ezLCD-30x splash screen will re-appear. Your connection test is now complete and you can begin programming your ezLCD-30x.

4.6 Flash Drive Access

When accessing the flash drive on the ezLCD-30x you need to be careful with flash drive access.
1) To avoid problems do not have the flash drive open on the PC while you are modifying the contents with serial commands.
2) When you finish updating the flash drive contents on the PC, make sure you eject the drive. The drive may be ejected from the PC by right clicking the ezLCD drive letter and clicking EJECT in Windows Explorer.
3) The flash drive file system uses DOS 8.3 format that allows filenames up to 8 characters followed by a period and a 3 character extension. If you create a file from the PC with a file name longer that the 8 characters, the ezLCD-30x you will not be able to access the file without knowing the DOS filename the PC used to store it. Using the DIR (directory) command from "Termie" will display the DOS file names for you.
4) If files are changed on the flash drive using the internal ezLCD-30x capability, you must press F5 or refresh to see the changes on your PC. Example would be "rename serif48.ezf serif50.ezf". The PC would continue to show serif48.ezf until you press F5, refresh or reboot the ezLCD-30x at which time the PC will then show serif50.ezf.
5) For more information on the Flash File System please see section 13.0.

5.0 Command your ezLCD with EarthSEMPL

Your ezLCD-30x is really a computer and like all computers it has a language in which you can talk to it in. The ezLCD-30x uses the simplest of languages, which we call **Earth Simple Embedded Macro Programming Language**, or **EarthSEMPL** for short. We will use the terms **EarthSEMPL program** and **macro** interchangeably in this manual.

5.1 ezLCD-30x Grammar

The syntax or grammar for EarthSEMPL commands is:

COMMAND {PARAMETER1} {PARAMETER2}...{PARAMETERn}<CR>

EarthSEMPL source code is a free-form ASCII text-line-based language which allows arbitrary use of white space (spaces or tabs) to format code, rather than column-based or text-line-based restrictions. ASCII allows almost any editing program to be used for writing your code. Note: ezLCD commands are not case sensitive (can be upper or lower case letters) except for the 'ezLCD Upgrade' command. Comments may appear either at the beginning of the line or after a command and must be preceded by a single apostrohe (') or the command **COMMENT**.

In the syntax above, **COMMAND** is one of the commands from **Appendix D**.

The **PARAMETER** can be a number, string, index or comment. Between every **COMMAND** and **PARAMETER** you must leave a space, comma or tab.

a) Numbers and indexes inputs are 16 bits and can be decimal, Binary (0b100110111) or Hex (0x3456 or 0h7E54). Any number over 16 bits will be truncated.
b) Strings can be any combination of ASCII characters and should be enclosed by a double quote ("String 1"). A string may also use the back slash as a lead in to an escape character sequence. Current escape characters supported are;
\n Line Feed
\r Carriage Return
\" Double quote
For example to print a word in quotes on the string you would print "\"hello\"".
c) Comments start with single quote (') and continue until the end of the line <CR>. 'this is just a comment

Note: OFF or ON can be used for most commands instead of 0 and 1 for readability.

Finally the **<CR>** represents a carriage return. Note that in your terminal program the carriage return is sent when you press enter and is not shown on the screen. If you use a microcontroller it should send the carriage return byte (13 decimal or 0D hex) after each command. The carriage return tells EarthSEMPL to immediately execute the command you just typed. Your terminal program must send a carriage return after each line of a command and when you write macros in a text editing program your

editor must insert a carriage return after each line of text (Notepad, Wordpad and almost all editors do). In the included Termie program, it's important that you select the **Append CR** option in **SETTINGS**. For Firmware 1.1 and after, you should select the **Append CR-LF** option.

5.2 Creating and Saving Macros

EarthSEMPL is an interpreter. This means that the code you write is executed immediately which allows for testing and changing your program immediately. Although commands run instantly in command line, they are not saved unless you assemble them into a file. There are two ways to save a macro. One way is to **RECORD** it with the ezLCD-30x and the other is to type or paste the commands into an editor on your PC and save as an .ezm file on the ezLCD Flash drive in the \EZUSER\MACROS directory. See **Section 6.0** for detailed information on writing macros.

5.3 Always Comment

While the **Hello Earth** program may be simple to read and understand for an experienced programmer, courteous programmers put a comment on every line. A single quote in a command line tells the ezLCD-30x command processor to ignore all text after the quote in that line. Your comment should be a brief description of what that line of code does. So, the **Hello Earth** program should like this:

```
'ezLCD-30x Hello Earth Program

'Written 9/10/11 by James Harrell

CLS BLACK 'Clear screen to black

COLOR WHITE 'Set drawing color to white

FONT SANS72 'Set font to SANS72

XY 100 40 'Set cursor x=100 y=40

PRINT "HELLO" 'Print Hello

XY 100 110 'Set cursor x=100 y=110

PRINT "EARTH" 'Print Earth
```

6.0 EarthSEMPL Macros

A macro is a group of commands and can be as short or as long as you'd like. Macros can be created on your ezLCD-30x by using the **RECORD** command. When all the required commands are typed in for a macro use the **STOP** command to stop recording the macro and to save the file.

Studying existing EarthSEMPL macros is the easiest way to learn about your ezLCD. Your ezLCD includes many examples stored in the \EZSYS\MACROS directory demonstrating the various commands. The latest can be downloaded as part of the file system for your ezLCD-3xx model on it's product page at the EarthLCD website. The ezLCD-30x latest file system is located at http://www.earthlcd.com/ezLCD-30x . Note: Substitute your model number for ezLCD-3xx.

The macros can also be created in a text editing program. The Windows application **Wordpad** works great for this. To create a macro, open a new document in your text editor and type commands just as you would in the terminal window. When you think you've got it right, save the file to the **USER/MACROS** folder on your ezLCD-30x USB flash drive as a .txt file, but use .ezm as the file suffix. **Make sure that the file name is 8 characters or less**, (not including the "ezm" file suffix). For example, if you wanted to save the **Hello Earth** program from earlier as a macro, you would enter the lines of code as they appear in the manual in your text editor and save it. Calling it **Hello.ezm** is a good choice, since the word **Hello** has only 5 characters.

To run the macro, go back to your terminal program and type **PLAY HELLO**. **HELLO EARTH** will appear on your screen exactly as it did when you typed the program in line-by-line. To run other macros just type **PLAY** and the macro name.

One benefit of creating macros with a text editor is that it gives you the opportunity to test your programs with the trial-and-error method. For example, if you're working out the placement of an item on screen, you can enter the **XY** values, save the macro and run it. If the item placement is off, adjust the **XY** values, save the macro and try again.

Once you've written a program and saved it as a macro, you can use that macro as a starting point or template for other macros. You can open up a macro that you've created in your text editor, modify the code, save it under a new name and you've got a brand new macro.

In addition, there are a number of factory-supplied macros on your ezLCD-30x USB flash drive. Some are demos and some are tools to help understand the features and capabilities of your ezLCD-30x. Remember, when you start creating new macros from existing ones, **always** make a copy of the macro into the USER directory before you change anything.

IMPORTANT NOTE: Sometimes bad macros or not stopping macros by using STOP or the RESET command will cause the USB port to crash. If you do a lot of development this way a separate USB to serial adapter talking to the ezLCD serial port is recommended. One has been built into the optional ezLCD-30x-EDK development board.

6.1 STARTUP.EZM - Your Most Important Macro

The most important macro on your ezLCD-30x is the startup macro, **startup.ezm,** which automatically runs every time the ezLCD-30x is powered on. (If you are familiar with MSDOS it is similar to autoexec.bat!)

It may also be used to set default fonts, themes, colors and other ezLCD parameters. Application notes by EarthLCD will assume you are using the factory default macro.

Never change the default startup macro.

Instead copy the original **\EZSYS\MACROS\STARTUP.EZM** into the **\EZUSER\MACROS** directory and then customize it for your application. For images, fonts and macros, including startup.ezm files, the ezLCD-30x will check the **\EZUSER\MACROS** directory first. If it does not find it there it will then look in the **\EZUSER** directory.

In rare cases you may make your ezLCD inoperable by what you put in startup.ezm, so we highly suggest that when you make a copy of it in the **\EZUSER\MACROS** directory and name it to test.ezm and run it manually a few times before changing the name to startup.ezm. Typically when you develop an application you will put it in a macro such as myprog.ezm and during testing run it manually (type 'play myprog' in termie). When done and you are ready to distribute the program you would add this line to your startup.ezm in the user directory: play myprog.

7.0 Images

The ezLCD-3xx can display .jpg, .gif and .bmp image files. Example image files are located in your **\EZSYS\IMAGES** directory. Image files must be kept in your **\EZUSER\IMAGES** directory. To display an image file, type the command **PICTURE** or **IMAGE** into your terminal window followed by the image name, **including the file suffix**. The image should match the pixel width and height and number of colors of the display characteristics of your ezLCD30x model. To display properly at full-screen on an ezLCD-301 all images should be 400 pixels wide by 240 pixels tall, 16 bit color. Images saved in .gif format offer the smallest file size and fastest load time.

7.1 Resizing an Image in Photoshop

This example is for the ezLCD-301. Other displays would be slightly different.

Open the image in Photoshop and select the **crop** tool from the toolbar. With the **crop** tool selected, set the width and height ratio of your crop. At the top of the screen are two boxes labeled **width** and **height**. Set the width value to **4** and the height value to **2.4**.

Drag the **crop** tool across your photo diagonally and resize the crop window by grabbing the corner handles. When you're happy with the selected crop area, press **Enter** to crop the image.

Go to the **Image** drop-down menu and select **Image Size**. A dialog box will appear.

Set the **Width** value to **400 pixels** and the **Height** value to **240 pixels** and select **OK** (Figure 5).

Go to the **File** menu and select **Save for Web & Devices.** A dialog box will appear.

Set **image type** to **GIF**, leave the **Transparency** button unchecked and set colors to **128**. Leave all other settings at default.

Select **Save** and save to your **\EZUSER\IMAGES** folder on the ezLCD-30x USB flash drive.

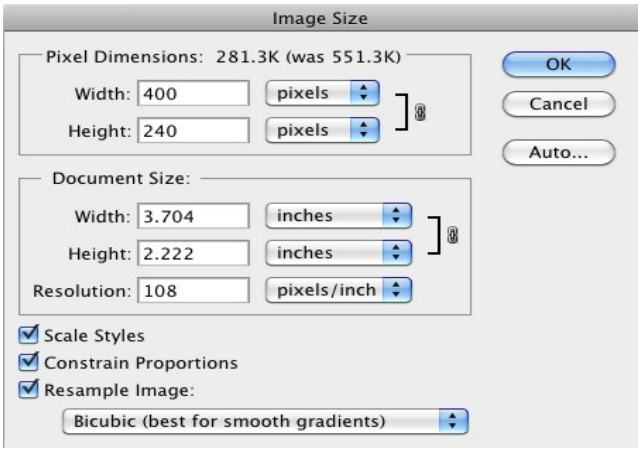


Figure 5

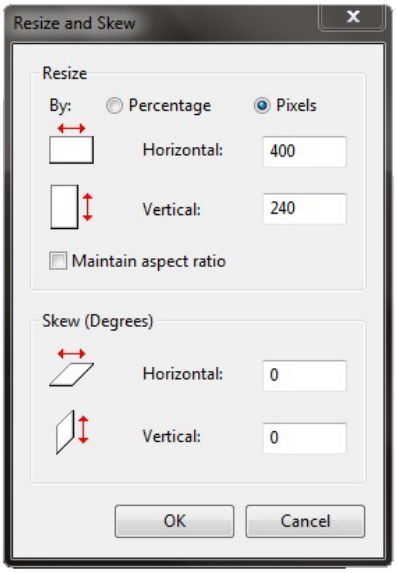
7.2 Resizing an Image in Windows Paint

Open the image in Windows Paint. Make sure that under the **Paint** dropdown menu, **Image Properties** the **Units** option is set to pixels.

Use the **Rectangular Selection tool** to select the area of the image that you want to crop. Paint does not allow for fixed cropping ratios. To overcome this, you'll have to watch the pixel dimensions of your selection box (displayed at the bottom of the window) as you size it. To avoid image distortion during the resize process, do your best to achieve a ratio of 4:2.4. Once you've achieved the desired size, select the **Crop** button.

Next, select the **Resize** button. A dialog box will appear entitled **Resize and Skew** (Figure 6). Select **Pixels** and uncheck the **Maintain aspect ratio** box. Enter 400 as the horizontal value and 240 as the vertical value. Select **OK**.

Figure 6



To confirm that your image has been sized properly, open the **Image Properties** dialog box again. When you're happy with the appearance of your image, go to the **Paint** dropdown menu and select **Save As**. Remember that your file name is limited to 8 characters. Save the image to your **\EZUSER\IMAGES** folder as a .gif to ensure the smallest file size possible and fastest load time on your ezLCD-30x.

8.0 Colors

The ezLCD-30x has the ability to display up to 65,535 colors simultaneously. This is referred to as 16-bit color. You may input 24-bit colors which the ezLCD-30x converts internally to 16-bit. To ease color selection, an index of 200 customizable colors are provided internally (see COLORID). The first 168 colors have been preset in your ezLCD-30x, with the remaining colors available for your own custom colors. Each color has been assigned an index number. The first 16 colors can be referenced by name or by index. For example, the following two commands will both clear the screen to red:
CLS RED Clear screen to red using color name **RED**
CLS 4 Clear screen to red using color index number **4**

8.1 Set Current Color

To set the current color, type **COLOR [C]** into your terminal program, where **{C}** is either the color name or color index number. The current color will remain active until you change it.

8.2 Set Custom Color

The **COLORID** command allows you to set custom colors by entering their RGB values. The first 16 COLORIDs are not changeable.

COLORID [id][R][G][B]

To set a custom color, an index number **[id]** and RGB values **[R][G][B]** must be assigned. Use IBM blue as an example. We know that the RGB values of IBM blue are Red=83, Green=120, and Blue=179 on the RGB scale. To assign IBM blue to color index number 180, type the following command:
COLORID 180 83 120 179

To test it enter:

CLS 180

The first 168 color values are pre-defined. While it is possible to change the color values from 16 through 168 with the **COLORID** command, it is not recommended. Instead, use index 169 through 199 for your custom colors. A full list of preset colors with their respective index numbers and color names can be found in **Appendix C.**

A macro has been included on your ezLCD-30x which shows the 16 most commonly used colors along with their color index numbers. Type **PLAY COLORS** in your terminal program to run this macro.

9.0 Drawing

The ezLCD-3xx has the ability to draw individual pixels, lines, boxes, circles, circle sections and arcs. Successive shapes will appear to be layered on top one another if drawn in the same location. All shapes are drawn with the current color. To see a demo of some shape examples, type **PLAY SHAPES** into your terminal program. For more detailed options see the command table in **Appendix B**.

To draw on the display you must first set the current position of the pointer [**x][y]**. This is done with the **XY [x][y] or XY [align]** command. **[align]** options are LT (Left Top), CT (Center Top), RT (Right Top), LC (Left Center), CC (Center Center), RC (Right Center), LB (Left Bottom), CB (Center Bottom), RB (Right Bottom). You can type XY <cr> and it will tell you the current XY position so try the different Align options and see what XY is set to. For the ezLCD-301, XY CC follow by XY will return 199 119. This the center of the screen vertically and horizontally.

Before any drawing, the position must be set. It also can be set from a previous command. Default is 0,0 (upper left corner) after CLS.

9.1 Drawing a single pixel

Typing the command **PLOT** will modify the pixel at the current position to the current color.
Typing the command **PLOT [x][y]** will modify the pixel at the **[x][y]** position to the current color.

9.2 Drawing a line

Typing the command **LINE [x][y]** draws a line from the current **xy** position to **[x][y],** using the current color and line type.

The **LINETYPE[option]** command gives you the option to draw a sold, dotted, or dashed line.

Options: 0 = solid, 1= dotted (1 pixel spacing between dots), **2 = dashed** (2 pixel spacing between dashes)

The **LINEWIDTH[width]** command allows you to draw either a **thin** line (width = 1) or a **thick** line (width = 3). Only [width] = 1 or 3 are available.

9.3 Drawing a box

Typing the command **BOX [w][h][f]** draws a box at current **xy** position. Replace **[w]** and **[h]** with the desired width and height of the box, in pixels. Replace **[f]** with either a **0, 1 or f**. **0** or no value dictates an open box, **1** or **F** dictates a filled box.

9.4 Drawing a circle

Typing the command **CIRCLE [r][f]** draws a circle at current **XY** position. Replace **[r]** with the desired radius, in pixels. Replace **[f]** with either a **0, 1 or f**. **0** or no value dictates an open circle, **1** or **f** dictates a filled circle.

9.5 Drawing a circle section

Typing the command **PIE [r][s][e]** draws a section of a circle (pie slice) at current **xy** position. Replace **[R]** with the desired radius of the section, in pixels. Replace **[S]** with the start angle at which you want the section to start. Replace **[e]** with the end angle at which you want the section to end.

9.6 Drawing an arc

Typing the command **ARC [R][S][E][F]** draws an arc at current XY position. Replace **{R}** with the desired radius of the arc, in pixels. Replace **[S]** with the start angle at which you want the arc to start. Replace **[E]** with the end angle at which you want the arc to end. Replace **[F]** with either a **0, 1 or F**. **0** or no value dictates an open circle, **1 or F** dictates a filled arc.

10.0 Fonts and the Print Command

Your ezLCD-3xx comes with a selection of different fonts pre-installed. Type the command **PLAY FONTS** into your terminal program to run a macro that displays the factory installed fonts in their available sizes. The number designation in the font name refers to the height of the font in pixels. Therefore, the font **SANS48** is 48 pixels tall when displayed on the screen of the ezLCD-3xx.

The ezLCD-30X font converter for windows will allow you to create and customize new ezLCD fonts from TrueType and OpenType fonts. It is available for free download at **http://www.earthlcd.com/ezLCD-301.**

To set the current font, type the command **FONT [font]** into your terminal program, where **[font]** is the name of the font. It is not necessary to include the font suffix (.ezf). Only *.ezf can be used.

To test your font, type the following into your terminal program:

```
XY CC              'goto the center of the screen

FONT NEURO72       'select your font file to use

PRINT "HELLO"      'display the string to screen
```

The word **HELLO** will appear in the center of the screen and display in the **NEURO** font 72 pixels tall. The **PRINT** command always displays text on the screen of the ezLCD-30x at the current XY position. When using the **PRINT** command, be sure to place double quotes around the text you want to appear.

In addition to printing at the current XY position, text can also be positioned automatically to 9 positions relative to the current XY: LT (Left Top), CT (Center Top), RT (Right Top), LC (Left Center), CC (Center Center), RC (Right Center), LB (Left Bottom), CB (Center Bottom), RB (Right Bottom).

Note: To understand this command visualize your text string as a bitmap or box with characters in it and this command is positioning the box. Default position is LT. To use this feature, enter the position information at the end of the **PRINT** command. For example, we can set the current position to the center of the screen (XY CC) and then print the text using text justified to the center (PRINT "Hello" CC):

```
XY CC 'set current position to center of screen

PRINT "HELLO" CC 'print to screen with CC option
```

will print the text at the horizontal and vertical center of the screen. This command will justify the text referencing the center of the text because of the CC option. You can also justify text placement with the other 8 options. It is an easy way of placing text without computing font heights and widths. To see a macro that demonstrates this, type **PLAY ALIGN** in your terminal program.

The command, **fonto [option],** will change the orientation or direction the text prints.

[option] 0 = horizontal orientation, **1** = vertical orientation

Since the vertical orientation is relatively uncommon, it is a good idea to set the orientation back to horizontal, when exiting a macro, else other macros might behave incorrectly For more detailed **FONT** and **PRINT** options, see the command table in **Appendix D**.

11.0 Widget Themes

The THEME command is a way of describing color and font characteristics of widgets. Themes are introduced here but you may want to proceed to the 12.0 Widgets and play with widgets and refer back to here when you want to customize the themes for the examples. Different widget types can use the same color theme to make your GUI look consistent. The THEME command sets the theme but when we discuss them we may use the term themes in the manual. The ezLCD-3xx supports 16 themes (0-15). The first eight widget themes (0-7) and widget fonts are preset in the startup macro:

```
fontw 0 serif24          'set theme 0 font for widget to serif24 (serif24.ezf font file in \SYS\FONTS\)
fontw 1 serif24
fontw 2 serif24
fontw 3 serif24
fontw 4 serif24
fontw 5 serif24
fontw 6 serif24
fontw 7 serif24
```

'THEME	A	B	C	D	E	F	G	H	I	J	K
theme	0	1	2	0	0	0	3	3	1	0	0
theme	1	155	152	3	3	3	3	4	4	50	1
theme	2	5	20	3	3	3	4	4	5	0	2
theme	3	9	3	0	0	0	8	8	9	0	3
theme	4	7	3	0	0	0	6	6	6	6	4
theme	5	126	118	3	3	3	35	35	36	0	5
theme	6	111	106	3	3	3	12	12	101	0	6
theme	7	58	48	3	3	3	14	14	54	0	7

(Note: The above themes were shipped starting with Firmware version 1.1 and File System Release E. Please confirm the themes in your startup file to avoid confusion and remember ezLCD-30x will be using themes from the \USER\MACROS\STARTUP.EZM directory if it exists otherwise themes set in \SYS\MACROS\STARTUP.EZM.)

You must set the widget font with the FONTW command before setting the theme. By having themes set by the startup.ezm macro it avoids having to send them from your host and saves memory on your host. These themes will work with any widget that you create. Changing the default themes 0-7 may cause examples in this manual and demo macros on your flash drive to not display correctly. If it's ever necessary to reset to the default themes simply type RESET and **startup.ezm** macro automatically runs reloading the default themes. If you create new themes for your project it is recommended you use theme id's 8-15. The widget themes contain values for:

Theme [ID][EmbossDkColor][EmbossLtColor][TextColor0][TextColor1][TextColorDisabled] [Color0][Color1][ColorDisabled][CommonBkColor][Fontw].

To see this in context, type the following into your terminal program:

THEME 5 126 118 3 3 3 35 35 35 35 2
Don't worry about damaging the default settings - these are the default values for theme 5. You can change the settings and see the results by using the **BUTTON** command as outlined in **Section 12.0**.

The command **THEME 5** references theme 5.

The **[EmbossDkColor] and [EmbossLtColor]** values, **126** and **118**, designate the colors that act as the highlight on the upper left edge and the shadow on the lower right edge of each button, respectively. These look best when the highlight is a few shades lighter than the main button color and the shadow is a few shades darker.

The **[TextColor0], [TextColor1]** and **[TextColorDisabled]** values, **3**, **3** and **3**, designate the color of the text when the button is at rest, being touched or is disabled. They're set by default to be the same colors, but can be changed to give a visual indication of the button's state.

The [**Color0], [Color1]** and **[ColorDisabled]** values, **35**, **35** and **35**, designate the color of the face of the button when it is at rest, being touched or is disabled. They're set by default to be the same colors, but can be changed to give a visual indication of the button's state.

The [**CommonBkColor]** value, **35**, designates the common background color. This is also set by default to match the button's face.

The [**Fontw]** value, **2**, specifies the font to be used with the theme. The font MUST be defined **BEFORE** defining the theme using the FontW command.

For more details on widget themes, see the command list in **Appendix B**. Further information can be found by opening the **buttons.ezm** macro in a text editor.

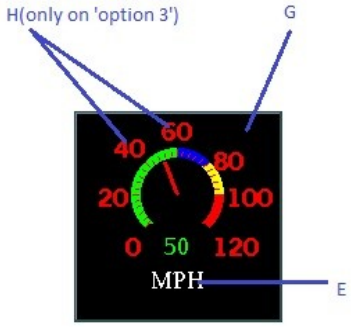
11.1 Diagrams of Widget Themes

Theme [ID] [EmbossDkColor] [EmbossLtColor] [TextColor0] [TextColor1] [TextColorDisabled] [Color0] [Color1]

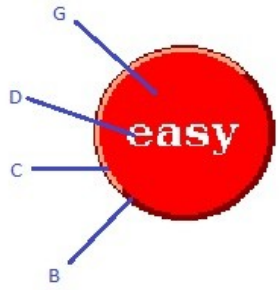
[A] [B] [C] [D] [E] [F] [G] [H]

[ColorDisabled] [CommonBkColor] [Fontw]

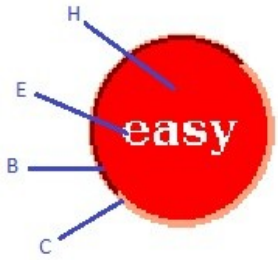
[I] [J] [K]



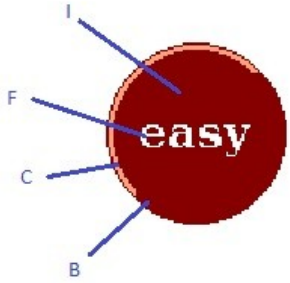
Analog Meter



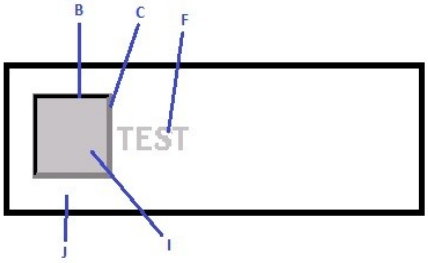
Button



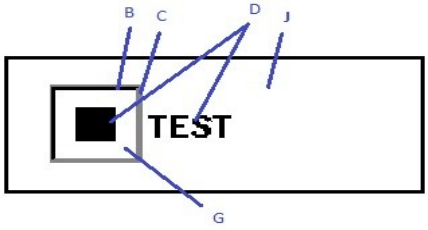
Pressed Button



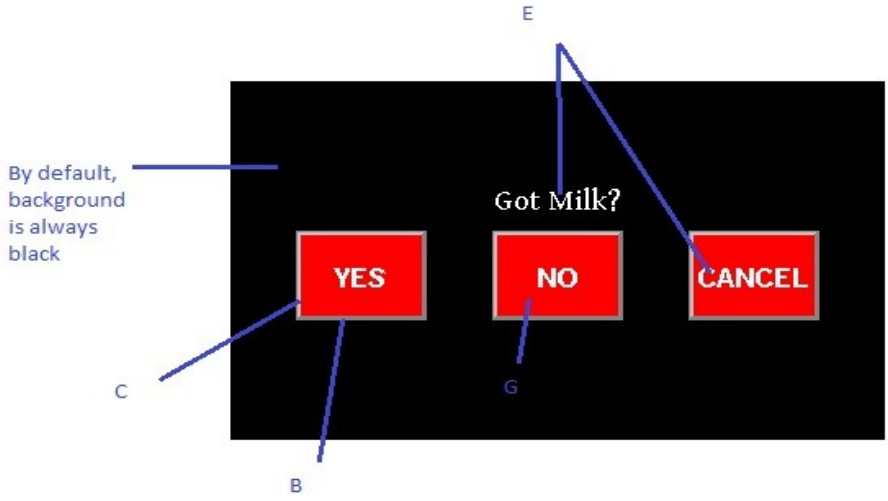
Disabled Button



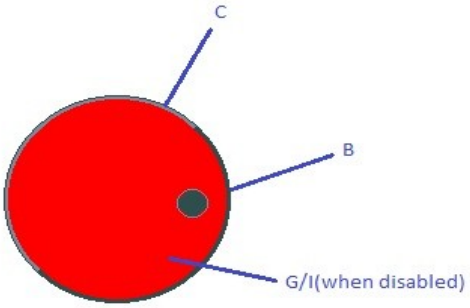
Disabled Checkbox



"Filled/Checked" Checkbox

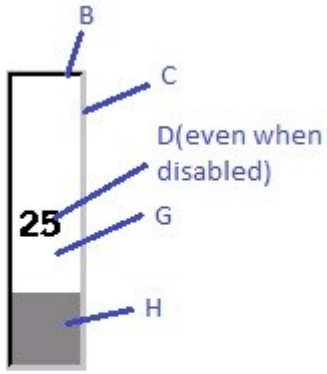


Choice

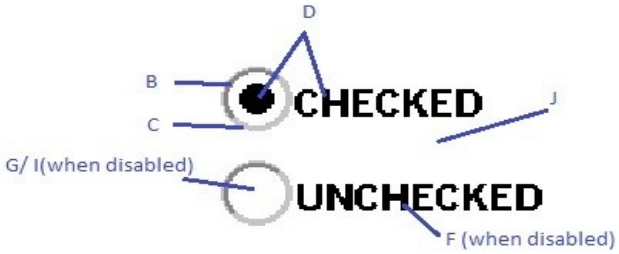


Dial

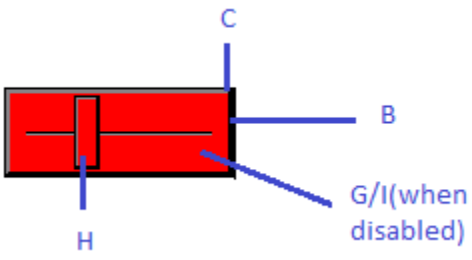
Digital Meter



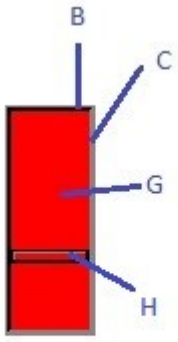
Progress Bar



Radio Button

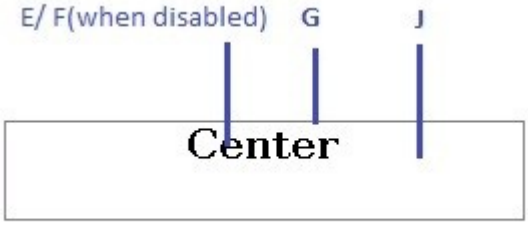


Slider



Slider (scroll bar option)

Static Box



Framed Static Box

11.1 Some Non-Widget Commands

CLIPENAREA[left]|[top]|[right]|[bottom] allows you to designate a rectangular/box area that you can draw in. Any surrounding area will be protected and no changes can be made to it. (ex. CLIPAREA 50 30 200 150)

CLIPENABLE[enable] is a command to turn on or off clipenable. 0/off = disable, 1/on = enable

RECORD[name] allows you to record your commands to a macro, **[name]**. You will have to use this command first and then all the following commands will be recorded. To stop recording, use command **STOP**.

LOOP[on/off] command allows you to run a macro over and over. To stop loop, either hit Ctrl+C or reset your ezLCD. When you activate loop in the console/terminal, you will not be able to input any commands until you have stopped the macro. Therefore, you will need to put LOOP OFF in the macro you are looping.

PAUSE[ms] stops the macro for any [ms] milliseconds and then continues. It is important to use pauses in between widget commands, such as changing values. For a widget, you would draw an initial state and without pauses, the changes will be visually instantaneous. By adding pauses, you will be able to see the changes in value.

SPEED[ms] The command allows you to determine how fast the commands are processed in milliseconds. The smaller [ms] is, the faster the commands are processed. 100 = 1/10 second, 1000 = 1 second

WAIT[option] stops a macro and waits for an event to happen.

[options] are T = touch, !T = No touch, TR = Touch and Release, IO[1-9] = wait for 1, !IO [1-9] = wait for 0.

For options **IO[1-9]** or **!IO[1-9]** you can assign the IOs to a button or such for the wait command. Then once the event happens, the macro will continue to run. Ios 10-33 can be used with the expander board.

SECURITY[option][password] allows you to to lock the flash drive so that it will be inaccessible from the PC until you unlock it.

[option] Set = set the password, Reset = to enter password/unlock it

For the [option], it is important that you capitalize the first letter otherwise it will not work. Also, it is important that you put quotes around [password]. (ex. **SECURITY Set "ezlcd"**)

CALIBRATE is used to make sure your "touches" on the ezLCD will be precise. This command will prompt you to press at different positions on the screen to help align the coordinates with the display.

FORMAT "ezLCD" "EarthRules" is a command that will format and erase all your files on your flash drive. This is useful when to clean up corrupt files you may have. Be sure to type the command exactly as it is shown. The parameters need to be in quotes as well as have the appropriate letters capitalized.

CWD displays the current directory you are in. One of its uses is that before making any changes to it, you would type CWD in the console port to make sure your are in the correct one. The default directory you are in is \EZSYS\FONTS.

CHDIR[name] or CD[name] changes to a directory, [name]. This is useful when you want to make

new files or changes to them, because you need to be or located at that directory the files are in. [name] needs to be in double quotes. If you want to move to a directory that is within another directory, you will need to show each directory paths with a \ in front. (ex. Chdir "\EZUSER\MACROS")

MKDIR[name] or MD[name] makes a directory. On your ezLCD, there are already directories named "EZSYS" and "EZUSER". You can create your own using these commands. However, if you want to make a directory/folder within another, you will need to show the paths. (ex. MD "\EZUSER\new") Before every directory, include \ in front. It is also important to put [name] in quotes.

RMDIR[name] or RD[name] removes/deletes a directory. [name] needs to be in double quotes. To delete the directory, you need to either be in the directory it is located in, or show the paths. (ex. \EZUSER\TEST).

TYPE[name] or MORE[name] command shows you the contents of the file you specified to the console port. This is useful when you want to look at a file without having to go into the flash drive and finding the file. The name needs to have its file extension and must have quotes around it. (ex. more "ameter.ezm") Make sure to change to the directory the file is in before trying to view it.

DEL[name] or ERASE[name] is used to delete or erase files that are in a directory without having to locate them in the flash drive. [name] needs to be in double quotes. You will also need to make sure you change to the directory the file is in. Also, include the file type/extension.

REN[original][new] changes the file's [original] name to [new] name. It names to be in double quotes as well as have file extension/type at the end. With the rename command, not only can you change the name, but also the file type. (ex. REN "droid.bin" "droid.ezm")

HELP[command] command displays the help file for the command you specified to the console port. This is useful when you want to look up the syntax for a command without a manual handy. The command file assumes the help file is in \ezSYS\Help and has an extension of .EZH. The commands is case insensitive.

TOUCHX Returns the last touchscreen X position.

TOUCHY Returns the last touchscreen Y position.

TOUCHS Returns the current touchscreen status.

0 = not currently pressed

3= pressed

4 = released

LIGHT brightness[,delay][,sleep brightness]

brightness will set the backlight level (0-100%). If present, delay will setup a timer to change the sleep brightness after delay in seconds.

LIGHT 100,5,50

Will set backlight to 100% and after 5 minutes will turn the backlight to 50%. Any touchscreen or serial activity on the command port will wake up the display. To turn off the timeout use a delay of 255.

BEEP [frequency][,duration]

Will generate a simple tone of the frequency and duration provided.

If frequency and duration are not provided it will default to 4000, 1000.

Before using the beeper it must be configured to match the hardware you are using.

For the expander board IO7 is used.
CFGIO 7,BEEPER
BEEP 4000,2000

BRIDGE [port]

The bridge command will bridge one peripheral to another.
Options are:
USBSERIAL1 Connect USB to UART1
USBSERIAL2 Connect USB to UART2
USBSERIAL3 Connect USB to UART3
DEBUGSERIAL1 Connects Console IO to UART1
DEBUGSERIAL2 Connects Console IO to UART2
DEBUGSERIAL3 Connects Console IO to UART3
DEBUGUSB Connects Console IO to USB
Its up to the user to avoid bridging IO improperly. The CFGIO command should be used to assign port pins to the peripherals as needed.

Example1:
```
CMD CDC       'Configure Command Port to USB CDC
cfgio 4 serial2_tx 115200 N81         'Configure Command Port to SERIAL2
cfgio 3 serial2_rx 115200 N81
BRIDGE DEBUGSERIAL2
```

This configuration will allow you to use either the USB or UART2 as command port. It is most useful when debugging a connection to a microprocessor thorough UART2. You can see what commands the microprocessor is sending to the ezLCD-30x.

If you were using the SERIAL2 as your console connection you could use
BRIDGE DEBUGUSB
This would allow the user to DEBUG the UART2 interface.

Example2:
```
cfgio 4 serial2_tx 115200 N81         'Configure Command Port to SERIAL2
cfgio 3 serial2_rx 115200 N81
cfgio 6 serial1_tx 9600 N81  'Configure Command Port to SERIAL1
cfgio 2 serial1_rx 9600 N81
CMD SERIAL2
BRIDGE USBSERIAL1
```

This would allow the UART2 as the console port and USB and UART1 are bridged together (not console). This configuration can be used to program an Arduino or Basic Stamp in circuit. You must also enable DTR and RTS to reset those products during programming. Appnote to follow.

12.0 Widgets

A widget is a reusable screen object of a graphical user interface that displays an information arrangement and provides standardized data manipulation. Widgets allow you to create an interactive user interface that is small, fast and easy!

There are user input widgets and output widgets. Input widgets takes the user input and outputs a result to the command port. These include the button and slider widget. Most of the input widgets require touch screen input to be useful and may have limited use for ezLCD-3xx displays with no touchscreen. Output widgets send out status events when changed to the command port. They usually have an element that can take a separate input from a widget value command to update just the part of the widget that shows the value like the needle on the AMETER widget. Control widgets like checkbox, radio button, and slider will send out status events when changed to the command port. This provides a faster and more dynamic user interface.

You could program your ezLCD with a startup macro that draws a meter widget in one line and then your host would only need to send 6 bytes to update the needle position! Button and checkbox widgets send 3 bytes when a button is pressed/released or checkbox is checked/unchecked. The ezLCD-3xx has various widgets that simplify the creation of objects on the screen. The CLS command clears all current widget references. Widgets are powerful in that they can be drawn typically by just issuing the command followed by the parameters and this can take as little as 11 bytes! Widgets use preset (but customizable) themes. Widgets that have values like progress bar and meters can be updated by sending the appropriate widget value command. Every widget has a unique ID and this value should be different for every widget you use in an application regardless of the type. The current widgets are Analog Meter, Button, Check Box, Choice, Dial, Digital Meter, Group Box, Progress Bar, Radio Button, Slider and Static Text.

Your ezLCD-3xx has the ability to display custom-themed buttons. There are 8 preset widget themes installed that can be used for various button shapes. See **Section 11.0** for more information about widget themes.

IMPORTANT WIDGET NOTES & ERRATA:
*1. You must be sure that the **[Width]** and **[Height]** parameter values fit on your screen or the widget will not be drawn!*
2. Widgets do not support internal fonts at this time.
3. Redraw function not supported so use CLS to clear widgets.
4. Your ezLCD contains documented examples of all the widgets in the \EZSYS\MACROS directory of its flash drive. The file will have the same name as the command such as DIAL.EZM and AMETER.EZM.

5. Widget Ids must be a number [1-99] and unique from other widgets.
6. *Some of the widgets require touch screen input to be useful and may have limited use for ezLCD-3xx displays with no touchscreen.*
7. *If you decide to modify or create a theme remember you need to send the Fontw command before sending the theme!*

12.1 Analog Meter Widget – AMETER

The AMETER widget allows you to display an analog meter which looks like a car speedometer. It's companion command, AMETER_VALUE, allows you to set the needle value without redrawing the whole meter. Also, the AMETER_COLOR command allows you to change the colors of the number and line indicators that form an arc around the meter. The AMETER.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing PLAY AMETER from your terminal program.

ameter [ID][x][y][width][height][options][value][min][max][theme][stringID][type]

ameter_value [ID][value]

ameter_color [ID][color1][color2][color3][color4][color5][color6]

EXAMPLE:

```
cls white                          'clears the screen to white
string 1 "testing"                 'the word will appear at the bottom of the widget
fontw 0 sans24                     'widget font needs to be set before the theme
theme 0 0 1 2 3 4 5 6 7 8 0        'colors are to distinguish different parts of widget
ameter_color 101 3 2 1 151 155 156 'ID = 101 when stated before widget
ameter 1 50 30 200 200 1 10 0 120 0 1  'ameter
pause 2000                         'pauses 2 secons before changing value from 10 to 50
ameter_value 1 50                  'ID must be same as the ID of ameter you're changing
```

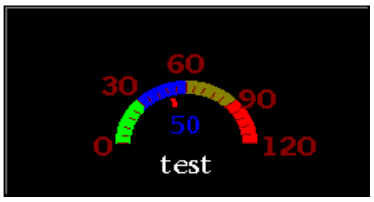
Images of Widget Options:

Option 1= Draw
Option 2 = Disabled

Option 3 = Ring

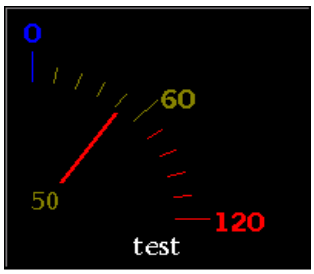
Option 4 = Accuracy

Option 1= Draw
Option 2 = Disabled

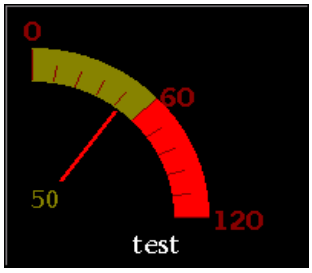


Option 3 = Ring

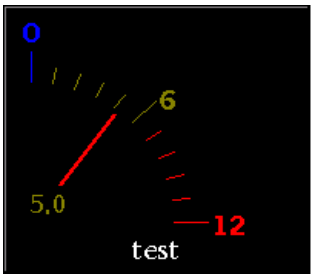
Option 4 = Accuracy



Option 1= Draw
Option 2 = Disabled



Option 3 = Ring



Option 4 = Accuracy

The AMETER command contains ten different values.

The **[ID]** value **1-99**, is the id number of this particular widget. You can create many different widgets as long as each ID is unique regardless of widget type. For **ameter_color,** the ID should be the same as the ameter you want to change. However if you start ameter_color before the ameter itself, use **ameter_color ID = 101**.

The **[x]** and **[y]** values designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the widget in pixels.

The **[options]**, **1**, designates the DRAW option of the analog meter.
To delete the widget, use command CLS. For Option 2 (disabled), the widget looks as it does for Option 1. Option choices: **1=draw, 2=disabled, 3=ring, 4=accuracy**.
Draw prints the widget to screen. **Disabled** draws a widget that can no be affected or changed. **Ring** draws the widget with an arc'ed bar around numbers. **Accuracy** allows you to display numbers with a decimal point for more exact numbers.

The **[value]** designates the initial value setting of the needle on the meter. For **ameter_value**, **[value]** is the value you want the ameter to change to.

The **[min]** designates the minimum value on the meter scale.

The **[max]** designates the maximum value on the meter scale.

The **[theme]** sets the widget to theme 5.

The **[stringID]** designates the id number of the text string that you'd like displayed below the meter.

[color 1-6] changes the colors of the 6 "zones" of indicating lines and numbers of the meter arc. The zones starts at **1** at the left end of the arc and **6** being the right end of the arc.
For half meter type only colors 3-6 are used. For quarter meter type only colors 5 and 6 are used.

The **[type]** is the meter type/style you want to use. By default, the meter type is set to full. For the half sized ameter, you will need to adjust **[width]** to make the meter proportional. For **full** and **quarter**
0=full, 1=half, 2=quarter

12.2 Button Widget

A button widget (sometimes known as a push button or command button) is a user interface element that provides the user a simple way to trigger an event. You can draw/make a variety of buttons making them round or square by adjusting the **[radius]** parameter. This is the ideal replacement for a switch in an embedded application. Your ezLCD-3xx has the ability to display custom-themed buttons.

button [ID][x][y][width][height][options][align][radius][theme][stringID]

Example:

```
cls white                          'clears the screen to white
string 1 "testing"                 'the word will appear at the center of the widget
fontw 0 sans24                     'widget font needs to be set before the theme
theme 0 0 1 2 3 4 5 6 7 8 0        'colors are to distinguish different parts of widget
button 1 10 10 100 100 1 0 50 0 1  'red when unpressed and yellow when pressed
```

Images of Widget Options:

Option 1 = Draw

Option 3 = Pressed

Option 2 = Disabled

Align 0 = Center

Align 1 = Right

Align 2 = Left

Align 4 = Top

Align 3 = Bottom

Pressing the button will send BP1 to the Command Port and releasing it will send BR1 to the Command Port.

The button command contains ten different values.

The **[ID]** value **1-99**, is the id number of this particular button. You can create many different buttons, and therefore button ids, as you'd like.

The **[x]** and **[y]** values designate the location of the button on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the button in pixels.

The **[options]** designates the state of the button, whether it is pressed, disabled, and et al.
Option choices: **1=draw, 2=disabled, 3=toggle pressed, 4=toggle not pressed, 5=toggle pressed disabled, 6=toggle not pressed disabled**.

The **[align]** value, designates the alignment of the text as it appears on the button.
Alignment choices: **0=centered, 1=right, 2=left, 3=bottom, 4=top**.

The **[radius]** value, designates the corner radius of the button corners in pixels. A value of **0** achieves a square corner, while a value that is half the length of one side will give a round button. To see some different shapes for buttons, run demo buttons.ezm.
Example for a square button: **button 1 10 10 100 100 1 0 10 0 1**

The **[theme]** value designates the widget theme. Type **PLAY BUTTONS** into your terminal program to see a macro example of the 8 included widget themes as well as some different button shapes.

The **[stringID]**, designates the id number of the text string that you'd like displayed on the button. You

can save as many different text strings as you'd like. To write a text string, type the following: **STRING 0 "HELLO"** and the word **HELLO** will appear on any button that designates text string **0**.

Note: To create multi-line text on buttons, use \n in the string contents. Example: string 5 "Wrap\nText" will appear on 2 lines.

12.3 Touchzone Widget

A touchzone widget is a user interface element that provides the user a simple way to trigger an event just by press an area of the screen defined by a box. You can also draw graphics of any kind onto the screen and place a hot spot around it with touchzone.

touchzone [ID][x][y][width][height][options]

Example:

touchzone 1 10 10 100 100 1

Pressing the touchzone will send TZP1 to the Command Port and releasing it will send TZR1 to the Command Port. Moving off the touchzone before releasing will generate TZC1.

The touchzone command contains six different values.

The **[ID]** value **1-99**, is the id number of this particular touchzone. You can create many different touchzones but they should not over lap.

The **[x]** and **[y]** values designate the location of the touchzone on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the touchzone in pixels.

The **[options]** designates the state of the touchzone, whether it is pressed, disabled, and et al.
Option choices: **1=enable, 2=disabled**.

12.4 Check Box Widget – CHECKBOX

The CHECKBOX widget allows you to display a check box with a string next to it. This permits the user to make a choice. When a CHECKBOX state changes (it is checked or unchecked) a status change is sent to the host via the current Command Port. When a check box with widget ID 1 is checked a CC1 is transmitted and when it is unchecked a CU1 is sent to the Command Port. A check box can also be viewed as a single state switch that can be set on (checked) or off (un-checked).

The CHECKBOX.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be implemented by typing PLAY CHECKBOX from your terminal program. The ezLCD-3xx has the ability to display custom-themed CHECKBOX's.

checkbox [ID][x][y][width][height][options][theme][stringID]

Example:

cls white	**'clears the screen to white**
string 1 "testing"	**'the word will appear at the bottom of the widget**
fontw 0 sans24	**'widget font needs to be set before the theme**
theme 0 0 1 2 3 4 5 6 7 8 0	**'colors are to distinguish different parts of widget**
checkbox 1 30 30 225 50 1 0 1	**'draws unchecked initially, checked when pressed**

Images of Widget Options:

Option 1 = Unchecked

Option 2 = Disabled

Option 3 = Checked

If you press the box it will check it and look like this:
and send "CC1" to the command port. Press again to uncheck and "CU1" is sent to command port.

The CHECKBOX command contains eight different values.

The **[ID]** value is the id number of this particular widget. You can create many different widgets as long as each ID is unique regardless of widget type.

The **[x]** and **[y]** values designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the widget in pixels.

The **[options]**, designates the initial state of the checkboxes.
Option choices: **1=draw unchecked, 2=draw disabled, 3=draw checked, 4=redraw**

The **[theme]** changes the colors on the widget.

The **[stringID]** designates the id number of the text string that you'd like displayed next to the text string.

12.5 Choice Widget – CHOICE

The CHOICE widget allows you to print a string and display buttons for the user to choose a response. CHOICE reply buttons are “yes”, “no”, or “cancel”. This widget is useful for asking simple “yes or no” questions without having to tediously figure out coordinates, sizes, and et al. for buttons and strings. The ezLCD-3xx has the ability to display custom-themed CHOICE’s. However, you will not be able to change the shapes of the buttons.

choice [string][theme]

Example:

fontw 0 sans24	'widget font needs to be set before the theme
theme 0 0 1 2 3 4 5 6 7 8 0	'theme for buttons; has no affect on bacground
CHOICE "Got Milk?" 0	‘quotes around string

Images of Widget Options:

Pressing the YES, NO or CANCEL button will output a 1, 0 or -1 respectively to the Command Port.

The CHOICE command contains two different values.

[string] “Got Milk?”, will be printed above the buttons. Please make sure to put the string in quotations marks. Also, the string cannot be substituted with a String id.
Responses/Return Values: **0=no, 1=yes, -1=cancel**

The **[theme]** only affects or will only change the colors for the buttons of the widget.

12.6 Dial Widget – DIAL

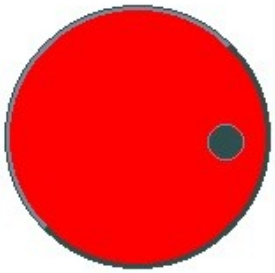
The DIAL widget allows you to display a dial that looks like an analog volume control found in modern cars. The DIAL.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing PLAY DIAL in TERMIE.

dial [ID][x][y][radius][option][resolution][value][max][theme]

Example:

```
cls white                          'clears the screen to white
theme 0 0 1 2 3 4 5 6 7 8 0        'colors are to distinguish different parts of widget
dial 1 100 85 75 1 1 25 100 0      'draws a dial
```

Images of Widget Options:



Dial

Pressing the handle and spinning it will move the DIAL in increments of VALUE and output the VALUE to the Command Port.

The DIAL command contains nine different parameters.

The **[ID]** value **1-99**, is the id number of this particular widget. You can create many different widgets as long as each ID is unique regardless of widget type.

The **[x]** and **[y]** values designate the location of the widget on the screen as the **XY** coordinate of the center of the dial.

The **[radius]** values means that the radius of the dial is 75, which the diameter will be 150.

The **[option]** designates the state of the dial. Option choices: **1=draw, 2=disabled.**

The **[resolution]** designates the increments in the range. So the indicator will be a value of every number, such as, 10,11,12,13,14,15.... has a resolution of **1**.

The **[value]** designates the initial dial value.

The **[max]** value designates the largest value of the dial's input.

The **[theme]** value sets widget theme. Note: Themes 0-7 are predefined in the STARTUP.EZM macro.

12.7 Digital Meter Widget – DMETER

The DMETER widget allows you to display a digital meter as in a panel meter. It's companion command the DMETER_VALUE command allows you to set the read out value without redrawing the meter. The DMETER.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing PLAY DMETER in your terminal program.

Note: If using negative numbers make sure your font has a minus sign. Some larger fonts only have numbers 0-9.

dmeter [ID][x][y][width][height][option][value][digits][dp][theme]
dmeter_value [ID][value]

Example:

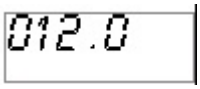
```
cls white                          'clears the screen to white
fontw 0 sans24                     'widget font needs to be set before the theme
theme 0 0 1 2 3 4 5 6 7 8 0        'colors are to distinguish different parts of widget
dmeter 1 50 50 200 160 14 0120 4 1 0   'background is green with a thin yellow frame
pause 2000                         'pauses for 2 seconds
dmeter_value 1 123                 'changes value
```

Images of Widget Options:

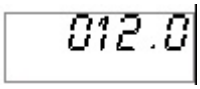
Option 1 = Left

Option 3 = Right

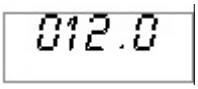
Option 4 = Center



Option 11 = Left Framed



Option 13 = Right Framed



Option 14 = Center Framed

The DMETER command contains ten different values.

The **[ID]** value **1-99**, is the id number of this particular widget. You can create many different widgets

as long as each ID is unique regardless of widget type. For **dmeter_value** the **ID** must be the same ID as the dmeter you want to change.

The **[x]** and **[y]** values designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the widget in pixels.

The **[option]** determines the alignment of the digits and whether the box is framed.
Option choices: **1=left, 2=disabled, 3=right, 4=center, 11=left framed, 12=disable framed, 13=right framed, 14=center framed, 6=redraw.**

The **[value]** designates and displays the initial setting of the readout as it appears on the meter.

The **[digits]** value designates the number of digits displayed on the meter.

The **[dp]** value designates the position of the decimal point from the 'right' most number.

The **[theme]** value sets widget theme.

12.8 Groupbox Widget - GBOX

The GBOX widget generates a border/box and by changing the **options** positions the header text at different alignments. Group boxes help visually distinguish related items by framing them. The Groupbox consists **only** of the frame, title, and a title background. The GBOX.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing PLAY GBOX from your terminal program.

gbox [ID][x][y][width][height][options][theme][stringID]

Example:

cls white	**'clears the screen to white**
string 1 "testing"	**'the word will appear at the bottom of the widget**
fontw 0 sans24	**'widget font needs to be set before the theme**
theme 0 0 1 2 3 4 5 6 7 8 0	**'colors are to distinguish different parts of widget**
gbox 1 20 30 300 200 1 0 1	

Images of Widget Options:

Option 1 = Left

Option 3 = Right

Option 4 = Center

The GBOX command contains eight different values.

The **[ID]** is the id number **1-99** of this particular widget. You can create many different widgets as long as each ID is unique regardless of widget type.

The **[x]** and **[y]** values designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the widget in pixels.

The **[options]** determines the header alignments. (The options do not affect the Contents' alignment) Option choices: **1=left,2=disabled,3=right,4=center**

The **[theme]** value sets widget to theme 2.

The **[stringID] d**esignates the id number of the string you want as a header of the box.

12.9 Progress Widget – PROGRESS

The PROGRESS widget allows you to display a progression bar at an initial state. To change the values to show progression or regression, use the PROGRESS_VALUE command. PROGRESS_VALUE command does not re-draw the entire percentage bar, but changes the value. The PROGRESS.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing PLAY PROGRESS in Termie. The ezLCD-3xx has the ability to display custom-themed progress bars. Changing the PROGRESS_VALUE setting (25) will update the BAR and PERCENT without having to redraw the whole widget.

progress [ID][x][y][width][height][option][value][max][theme]

progress_value [ID][value]

Example:

```
cls white                          'clears the screen to white
fontw 0 sans24                     'widget font needs to be set before the theme
theme 0 0 1 2 3 4 5 6 7 8 0        'colors are to distinguish different parts of widget
progress 1 50 100 180 40 1 10 100 0   'draws a progress bar, background = red
pause 2000                         'pauses for 2 seconds
progress_value 1 26                'changes value from 10 to 26
```

Images of Widget Options:

Option 1 = Horizontal

Option 3 = Vertical

The PROGRESS command contains nine different values and PROGRESS_VALUE contains two values.

The **[ID]** value **1-99**, is the id number of this particular widget. You can create many different widgets as long as each ID is unique regardless of widget type. In order to change the value of a specific progress bar, you must use its id number. If the progress id is **1** then the Progress_Value id must be **1**.

The **[x] [y]** values designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **[width][height]** values designate the width and height of the widget in pixels.

The **[option]** designates the option of the progress bar.
Option choices: **1=draw horizontal, 2=horizontal disabled, 3=vertical, 4=vertical disabled, 5=redraw horizontal, 6=redraw horizontal disabled, 7=redraw vertical, 8=redraw vertical disabled**

The **[value]** value designates the initial percentage value. By using the PROGRESS_VALUE command changes the initial value to a different one.

The **[max]** value, **100**, designates the maximum value that can be reached.

The **[theme]** value, **1**, sets widget to theme 1.

12.10 Radio Button Widget – RADIO

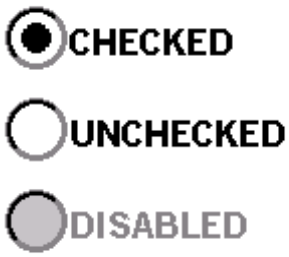
The RADIO widget allows you to display buttons for making a selection. Radio buttons differ from checkboxes in that only one button can be filled in at a time, while checkboxes can have many filled in. Therefore, radio buttons are interconnected. If one button is checked then the others will go to or remain as an 'unchecked' state. The RADIO.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing PLAY RADIO from your terminal program.

radio [ID][x][y][width][height][option][theme][stringID]

Example:

```
cls white                          'clears the screen to white
string 1 "testing"                 'the word will appear at the bottom of the widget
fontw 0 sans24                     'widget font needs to be set before the theme
theme 0 0 1 2 3 4 5 6 7 8 0        'colors are to distinguish different parts of widget
radio 1 25  50 230 35 5 0 1
radio 2 25  95 230 35 1 0 1
```

Images of Widget Options:



Pressing high, medium or low will send either a RB0, RB1 or RB2, respectively, to the Command Port.

The RADIO command contains eight different values.

The **[ID]** value **1-99** is the id number of this particular widget. Although radio buttons are connected as a group, each button still needs its own id number.

The **[x]** and **[y]** values designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the widget in pixels. The **[width]** and **[height]** are not the dimensions for the radio button, but it is the area which the button and string will be in. The size of the radio button itself is defined by the height of the button.

The **[options]** available allow you to draw radio buttons checked, unchecked, or disabled. By disabling a button, the user will not be able to change its state. Options 4 (first checked) and Options 5(first unchecked) help specify that it is the first button in a group. Options 4 & 5, therefore, allows you to have more than one group of buttons occupying the screen at the same time. When Options 4 or 5 are specified in a button command, the following buttons are in the same group as the first until another "first" button is defined. Then the buttons created after will be in the second group. If you make a button “first unchecked” remember to draw one button in the group as “checked”
Option choices: **1=unchecked, 2=disabled, 3=checked, 4=FIRST unchecked, 5=FIRST checked**

The **[theme]** is the theme ID you want to use. Theme will change the colors of the buttons and text of the widget

The **[stringID]** designates the id number of the text string that you’d like displayed by the button.

12.11 Slider Widget - SLIDER

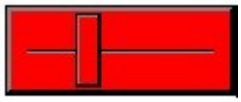
The SLIDER widget allows you to display a vertical or horizontal slider bar that looks like a light dimmer. The SLIDER widget components are the slider and a handle, also known as the thumb or indicator. The SLIDER.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing PLAY SLIDER in TERMIE.

slider [ID][x][y][width][height][option][range][resolution][value][theme]

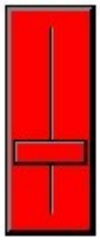
Example:

```
cls white                          'clears the screen to white
fontw 0 sans24                     'widget font needs to be set before the theme
theme 0 0 1 2 3 4 5 6 7 8 0        'colors are to distinguish different parts of widget
slider 1 20 30 100 50 1 75 5 25 0
```

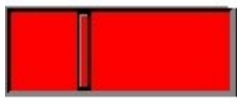
Images of Widget Options:



Option 1 = Horizontal



Option 3 = Vertical



Option 5 = Horizontal Scroll Bar

Option 7 = Vertical Scroll Bar

Pressing and sliding the slider thumb (handle) will update the slider value and image (without redrawing the whole widget) and output the setting to the Command Port.

The SLIDER command contains ten different values.

The **[ID]** value **1-99** is the id number of this particular widget. You can create many different widgets as long as each ID is unique regardless of widget type.

The **[x]** and **[y]** values designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the widget in pixels.

The **[option]**, **1**, designates the options of the slider.
Option choices: **1=draw horizontal, 2=horizontal disabled, 3=vertical, 4=vertical disabled, 5=horizontal slider, 6=horizontal slider disabled, 7=vertical slider, 8=vertical slider disabled**

The **[range]** value designates that the minimum and maximum value is 0-75.

The **[resolution]** value designates the increments in the range. So the indicator, or the thumb, will be a value of every other 5 numbers, such as, 10,15,20,25,30,35...75

The **[value]** value designates the initial value of the indicator.

The **[theme]** is the ID of the theme you want to use.

12.12 Static Text Widget – STATIC

The STATIC widget generates a framed text box with a header string at different alignments. This command changes text within a box without having to overwrite its background The STATIC.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing PLAY STATIC in your terminal program.

static [ID][x][y][width][height][option][theme][stringID]

static_value [ID][string]

Example:

```
cls white                          'clears the screen to white
string 1 "testing"                 'the word will appear at the bottom of the widget
fontw 0 sans24                     'widget font needs to be set before the theme
theme 0 0 1 2 3 4 5 6 7 8 0        'colors are to distinguish different parts of widget
static 1 10 25 220 25 5 0 1        'static box w/ green background & yellow frame
```

Images of Widget Options:

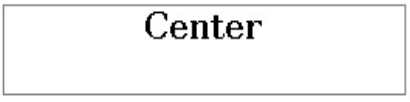
Option 1 = Left

Option 5 = Left Framed

Option 3 = Right

Option 7 = Right Framed

Option 4 = Center



Option 8 = Center Framed

The STATIC command contains eight different values.

The **[ID]** value **1-99**, is the ID number of this particular widget. You can create many different widgets as long as each ID is unique regardless of widget type. For **static_value** the **ID** needs to be the same ID as the static box you want to change.

The **[x]** and **[y]** values designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **[width]** and **[height]** values designate the width and height of the widget in pixels.

The **[option]**, designates the mostly the alignments of the STATIC widget. **Redraw** clears the background of the assigned area then rewrites the text.
Option choices: **1=left, 2=disabled, 3=right, 4=center, 5=left framed, 6=disabled framed, 7=right framed, 8=center framed, 9=redraw**

The **[theme]** is the ID of the theme you want to use.

The **[stringID]** designates the id number of the text string that you'd like displayed.

13.0 Snapshot Command

The Snapshot command is a lot like taking a screen shot. First, you need to display onto the screen what you would like to take a snapshot of. Then you use the command SNAPSHOT to capture and save it to a file. This command always saves the file as a 24bit BMP file regardless of how it was placed on the screen. This command may take as much as 10 seconds to capture the image if its a large area. After the command completes you should reset your ezLCD. You can then open the \USER\IMAGES file to see your snapshot on the PC.

REMEMBER to reset your ezLCD which is most easily done by unplugging the USB from the computer or pressing the reset button on the EDK board.

NOTE: As a precaution, take one picture and reset, continue this cycle however needed.

snapshot [x][y][width][height][filename]

Example:
PLAY FACE
SNAPSHOT 0 0 400 240 PIC

A file named PIC.BMP is saved in the \EZUSER\IMAGES directory of the ezLCD flash drive.
The SNAPSHOT command has five different values.

The **[x]** and **[y]** values designate the starting **XY** coordinate at the upper left corner.

The **[width]** and **[height]** values designate the area of the screen for SNAPSHOT to capture. Dimensions have to be even number, excluding hundred's digit. When taking an image of a widget, be sure to leave plenty room around the widget in the dimensions. **ex**. if the dimensions are not big enough for a button, the image will be distorted to fit the dimension.

The **[filename]**, **PIC**, is the name of the saved image. You do not have to specify the type of image it will be. It will be saved as a 24bit bitmap image in the \EZUSER\IMAGES directory. Be careful to not use the same name as other images. Otherwise the image will save over the other.

14.0 Widget State Command

The widget state command is used to adjust the state of a widget.

WSTATE [ID] [options]

[ID] must be the same as the **ID** of the widget you want to change.

[Options] are: **0 = delete, 1 = enable, 2 = disable, 3 = redraw**
0 = Delete the widget. This option redraws the widget to the common background color and then unlinks the widget ID from further processing. Once a widget is deleted its state can no longer be modified.
1 = Enable the widget. This option will enable a previously created widget that has been disabled. The Widget is redrawn with the enable colors in the Theme.
2 = Disable the widget. This option will disable a previously created widget that is enabled. The Widget is redrawn with the disable colors in the Theme.
3 = Redraw the widget. This option will redraw a previously created widget. This is useful if the widget has been over written by other text or if the string has been modified and needs to be redrawn on the widget.

Examples:

cls white	**'clears the screen to white**
string 1 "testing"	**'the word will appear at the bottom of the widget**
fontw 0 sans24	**'widget font needs to be set before the theme**
theme 0 0 1 2 3 4 5 6 7 8 0	**'colors are to distinguish different parts of widget**
static 1 10 25 220 25 5 0 1	**'draws a static widget**
pause 2000	**'waits for 2 seconds**
string 1 “tested”	**'change the string to “test”**
wstate 1 3	**'changes the string to “test” without having to redraw entire**

14.0 Flash Drive File Structure

The ezLCD-3xx USB flash drive appears as a removable storage device on the **Host** computer. In Windows, click on **Start**, then **Computer** and your ezLCD-3xx USB flash drive will appear. By double clicking on it you may access the ezLCD-3xx memory content which includes two directories:

EZSYS - system configuration default files (do not alter)

EZUSER - storage of user's custom fonts, images and macros

Note: It is recommended to make a copy of the **EZSYS** directory on your PC hard drive in case of accidental alteration of that directory on your flash drive. The **EZSYS** default directory is also available for download on the EarthLCD website.

EZSYS\FONTS - default font files (.ezf)

EZSYS\IMAGES - default image files (.gif, .jpg, .bmp)

EZSYS\MACRO - demonstration EarthSEMPL macro files (.ezm)

EZSYS\HELP – default help files (.ezh)

EZUSER\FONTS - storage of user-added fonts

EZUSER\IMAGES - storage of user-added images

EZUSER\MACRO - storage of user-added EarthSEMPL macro files

15.0 Ports & Embedding the ezLCD-3xx

While the ezLCD-3xx can be used as a display running off a PC or even as a standalone device, the ezLCD family of products were primarily designed to work as an embedded client. When used as a client the ezLCD will be controlled and communicated to by a host through one of its ports, sometimes referred to as I/O (input/output) ports. One of the ezLCD ports will be hooked to the compatible port on a host.

The host can be a micro-controller like an AVR, PIC or ARM micro-controller The host can also be a PC as it is in the examples we've shown up to now. More specific examples of embedding and connecting the ezLCD-3xx to other micro-controllers can be found on the ezLCD-3xx product page at **www.EarthLCD.com/ezLCD-30x**. We suggest that beginners take a look at the Arduino application notes.

15.1 The ezLCD-3xx Command Port

Ports are the way the ezLCD-3xx talks to the outside world. Ports on the ezLCD-3xx currently include USB and Serial. In the examples provided earlier in the manual, your command port is set to USB. As with previous generations of the ezLCD, most customers will use the provided tools and their PC to develop their user interface.
In a typical application, the ezLCD is connected to a micro-controller through one of its ports. The CMD command will let you set the command port to another besides the factory default USB. Most ezLCD application notes will use serial port 2 which is set by the following command:

First configure the GPIOs to the pins you are using for the serial. Typically this is;

cfgio 4 serial2_tx 115200 N81 'Configure Command Port to SERIAL2 TX

cfgio 3 serial2_rx 115200 N81

Now change the command port to the serial2 uart

cmd serial2 '115200 BAUD No Parity, 8 bits 1 Stop Bit

You could have used serial1,2 or 3 in this example.
Note: You cannot communicate with the ezLCD with any terminal program unless the command port is set to the correct COM port and baud rate. If you want to use the USB port type the command:

CMD CDC

In this manual we showed you how to use the ezLCD-3xx using the full ASCII **long** command. Every ezLCD-3xx command has two formats: **long** and **short**. On your micro-controller, which may have a limited amount of memory, you may want to use the short form which is a numeric ASCII string taking only one to three bytes (1-999) as shown in **Appendix B**.

Warning: Do not change the **CMD** port in the **startup.ezm** file located in the **\EZSYS\MACROS** folder. Instead, make a copy of it in **\EZUSER\MACROS** and modify it there.

15.2 Command Port Management

Setting the command port to another port besides USB can cause problems if you cannot set it back. That is the reason we recommend to **never change the startup.ezm file** in the **\EZSYS\MACROS** directory or any files in the \EZSYS directory.
To change your command port when you have the terminal program hooked to the USB port (when your ezLCD is plugged into a PC) you can simply type the appropriate **CMD** command shown above to switch back and forth.

For testing without USB hooked up just create a startup.ezm file with the **CMD** to switch the command port to your **micro-controller** host in the **\EZUSER\MACROS** directory. The **\EZUSER\MACRO** directory will be searched at power up or **reset** first and that startup.ezm will be run.
To get the ezLCD back to the USB command port host simply delete or rename the **\EZUSER\MACROS\startup.ezm** file causing the default startup.ezm (**SYS\MACROS\startup.ezm**) to switch your command port back to USB!

16.0 Start Something with your ezLCD-3xx

The best way to get familiar with a new computer program or product is to look at other people's examples. Look at the included macros in the **\EZSYS\MACROS** directory. Study them and run them to see what they do. Copy them to the **\EZUSER\MACROS** directory and then rename and modify them. We went through extra effort to document them as examples to learn from.

The possibilities and applications for the ezLCD-3xx are well beyond the scope of this manual. However, in the coming months the staff, consultants, customers and maybe even **you** will develop application notes that will be available on the product web page at **www.earthlcd.com/ezLCD-30x**. File updates, firmware and examples will also be provided there, so bookmark it and check back often.

17.0 Warnings, Errata and Gotchas

• **Always eject the flash drive before unplugging your ezLCD from your P.C.** Also eject after copying or modifying any flash drive files from your PC.

• **Close COM port before closing your terminal program.**

• **Never open Termie from the flash drive.**

• **As with any new technology product there will be bugs or opportunities for improvement ;-) :** If you find something that you think should be changed, fixed or enhanced, send it to support301@earthlcd.com and it will be addressed ASAP.

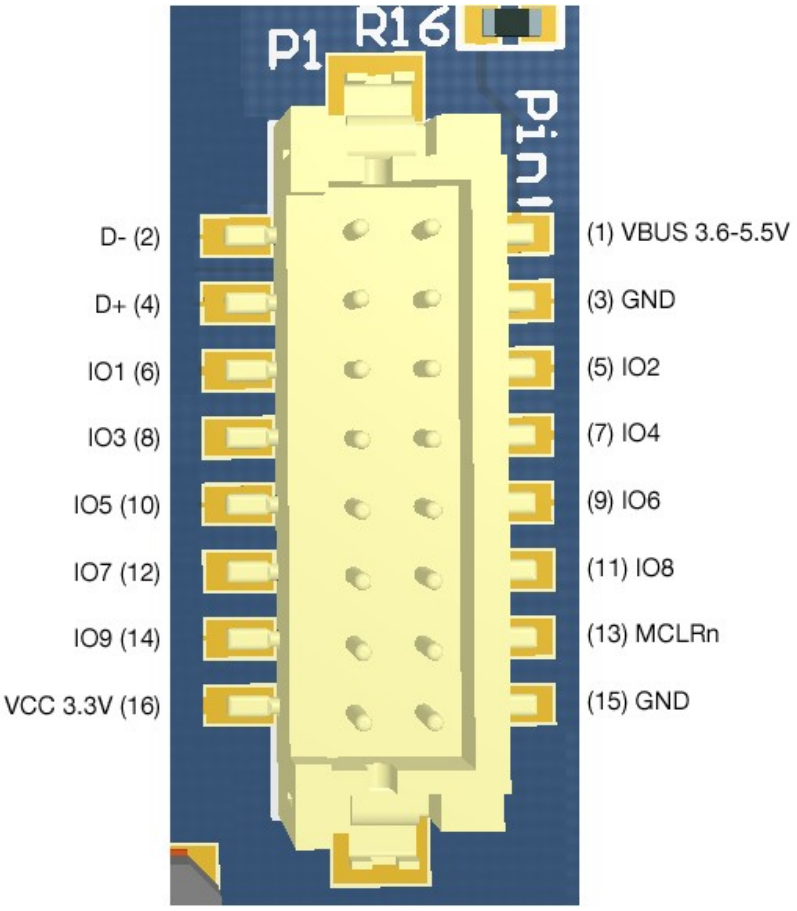
• **Do not play a macro of the same name within itself.**

• **When modifying the startup file, copy it from the SYS/MACRO directory to the USER/MACRO directory and name it something like TEST.ezm. Modify this file as you want. Then run the macro with PLAY TEST. After your changes are confirmed working like you want, then rename the macro to STARTUP.ezm. Debugging a live startup file can cause you serious grief if you modify something that locks you out of the comm port or flash drive or causes a crash. You may no longer get the chance to repair the problem in the startup. Updating firmware can not repair a bad startup file.**

18.0 Gratis (a note from Randy Schafer)

There are more than a few people who put up with my continuous banter about making this product right: Mark Eck, our VP of Sales and Marketing who's never short of new ideas. Our graphics artist and Maker enthusiast James Harrell. Rich Obermeyer, our renaissance engineer and VP of Engineering who left the ASIC world to come to Earth and make one more great product because I guaranteed him it would be fun. Also my wife Kate and Rich's wife Phyllis are to be commended for having patience with their geek husbands' absence while this product was developed. To all the employees, consultants and interns that help wring out the bugs on the prototypes, thank you! And last but not least, the customers of the last two generations of ezLCD who always held us accountable and inspired us to increase the passion in our work and make ezLCD better and ez-ER. Enjoy your new ezLCD-3xx!

Appendix A: ezLCD-30x Connector Pinout



Note: Connector is DF11-16DP-2V. Mating Connector DF11-16DS-2C or DF11-16DS-2DSA

The I/O pins are programmable for different I/O functions in the future but the current firmware supports the following command to set the command port to serial port 2 on IO3 & IO4 :

CMD SERIAL2 115200 1 N81

For firmware 1.4 on use

cfgio 4 serial2_tx 115200 N81 'Configure Command Port to SERIAL2

cfgio 3 serial2_rx 115200 N81

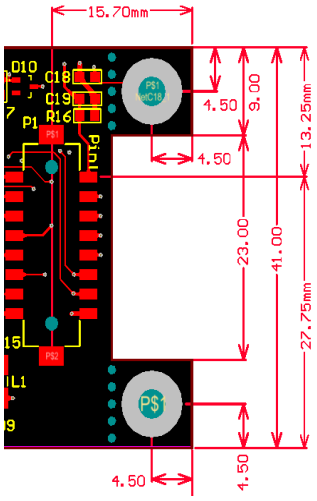
cmd serial2 '115200 BAUD No Parity, 8 bits 1 Stop Bit

‘ Set**s** command port IO3(Pin 8) to RX and IO4(Pin 7) to TX.

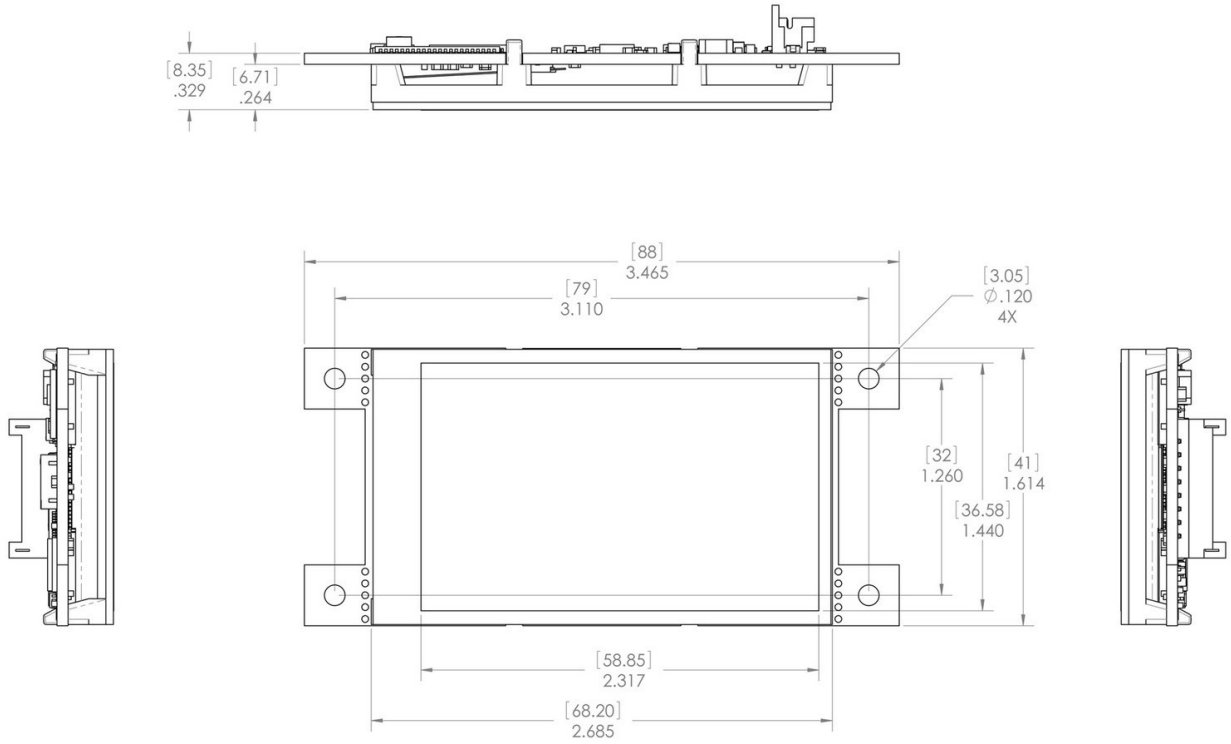
Note: RX is receive Input signal and TX is Transmit Output signal and the signals are 3.3 volts but 5 volt tolerant.

Appendix B: ezLCD-30x Model Descriptions and Drawings

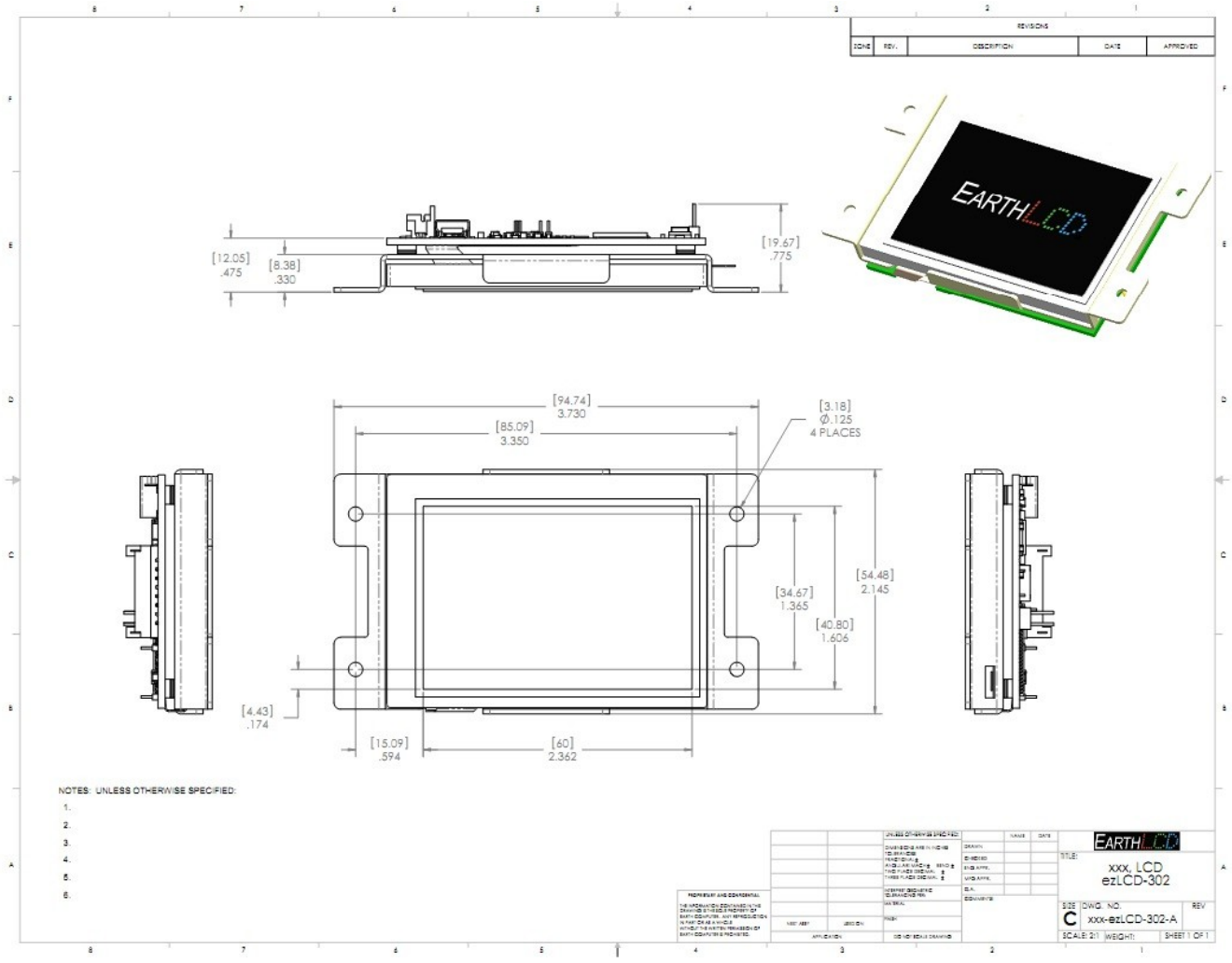
ezLCD-301: 400 x 240 Wide (10:6) 65.536 Transmissive Color TFT with Resistive TouchScreen
Connector Drawing (See 16 Pin): http://www.hirose.co.jp/cataloge_hp/e54305002.pdf
Connector Placement:



Mechanical Drawing:



ezLCD-302: 240 x 160 (4:3) 4096 Reflective Color TFT (Sunlight Readable - NO Touchscreen)



Appendix C: EarthSEMPL Colors

color_ID [x]	Color
0	black
1	grey
2	silver
3	white
4	red
5	maroon
6	yellow
7	olive
8	lime
9	green
10	aqua
11	teal
12	blue
13	navy
14	fuchsia
15	purple
16	indian red
17	light coral
18	salmon
19	dark salmon
20	light salmon
21	red
22	crimson
23	fire brick
24	dark red
25	pink
26	light pink
27	hot pink
28	deep pink
29	medium violet red
30	pale violet red
31	light salmon
32	coral
33	tomato
34	orange red
35	dark orange
36	orange
37	gold
38	yellow
39	light yellow
40	lemon chiffon
41	light golden rod yellow
42	papaya whip
43	moccasin
44	peach puff
45	pale golden rod
46	khaki
47	dark khaki
48	lavendar
49	thistle
50	plum
51	violet
52	orchid
53	fuchsia
54	medium orchid
55	medium purple
56	blue violet
57	dark violet
58	dark orchid
59	dark magenta
60	purple
61	indigo
62	dark slate blue
63	slate blue
64	medium slate blue
65	green yellow
66	chartreuse
67	lawn green
68	lime
69	lime green
70	pale green
71	light green
72	medium spring green
73	spring green
74	medium sea green
75	sea green
76	forest green
77	green
78	dark green
79	yellow green
80	olive drab
81	olive
82	dark olive green
83	medium aquamarine
84	dark sea green
85	light sea green
86	dark cyan
87	teal
88	aqua
89	cyan
90	light cyan
91	pale turquoise

92	aquamarine
93	turquoise
94	medium turquoise
95	dark turquoise
96	cadet blue
97	steel blue
98	light steel blue
99	powder blue
100	light blue
101	sky blue
102	light sky blue
103	deep sky blue
104	dodger blue
105	cornflower blue
106	royal blue
107	blue
108	medium blue
109	dark blue
110	navy
111	midnight blue
112	corn silk
113	blanched almond
114	bisque
115	navajo white
116	wheat
117	burly wood
118	tan
119	rosy brown
120	sandy brown
121	golden rod
122	dark golden rod
123	peru
124	chocolate
126	saddle brown
127	sienna
128	brown
129	white
130	snow
131	honey dew
132	mint cream
133	azure
134	alice blue
135	ghost white
136	white smoke
137	sea shell
138	beige
139	old lace
140	floral white
141	ivory
142	antique white
143	linen
144	lavender blush
145	misty rose
146	gainsboro
147	light gray
148	silver
149	dark gray
150	gray
151	dim gray
152	light slate gray
153	slate gray
154	medium turquoise
155	dark slate gray
156	black
157	gray7
158	gray20
159	gray40
160	gray80
161	gray90
162	gray95
163	red4
164	firebrick1
165	dark green
166	pale green
167	light yellow

Appendix D: EarthSEMPL Command Reference Guide

A quick reference guide of the EarthSEMPL command set that can run on the ezLCD-30x are listed below.
Input values can be an integer between -32768 and 32767. Strings can be up to 64 characters.
Examples and descriptions are provided for each command in the table. Note that this command list is updated from time to time and you should check the ezLCD-30x product page at EarthLCD.com for the latest documentation.

TABLE 1

COMMAND	SHORT FORM	SYNTAX	EXAMPLE	DESCRIPTION & OPTIONS
ARC	21	ARC [radius][start][end][fill]	ARC 50 100 120 1	Draw ARC with Radius, Start angle and End angle. Fill=1 or F to fill arc. Angle of 0 is on the right.
BOX	19	BOX [width][height][fill]	BOX 50 50 F	Place a box from current XY with specified width and height. Fill=1or F to fill box or string starting with F. Box must be > 1 pixel wide
CALIBRATE		calibrate	calibrate	This is used to make sure your touches will be precise.
CD	52	CD [directory]	CD "\EZUSER\MACROS"	Change to directory
CFGIO	37	CFGIO [GPIO][type]	CFGIO 1 0	Configure GPIO as input (0) or output (1), serial, SPI, I2C
	37		CFGIO 1 IN	Can use IN for direction
	37		CFGIO 4 OUT	Can use OUT for direction
CHDIR	52	CHDIR [directory]	CHDIR "\EZUSER\MACROS"	Change to directory
CIRCLE	20	CIRCLE [radius][fill]	CIRCLE 75 1	Draw circle at current XY with radius. Fill=1 or F to fill circle.
CLIPAREA	47	CLIPAREA [left][top][right][bottom]	CLIPAREA 50 30 200 180	ClipArea to protect the surrounding area from change.
CLIPENABLE	46	CLIPENABLE [enable]	CLIPENABLE ON	Turn on clip enable area
CLS	2	CLS [color]	CLS RED	Clear screen to COLORID[color]. This command also clears widgets.
CMD	62	CMD [interface]	CMD SERIAL2	Select command to interface. CDC, USB, SERIAL1, SERIAL2, SERIAL3
COLOR	6	COLOR [ID]	COLOR 0	Set current color to COLORID[index] (0-199)
	6	COLOR	COLOR	Return current color
COLORID	7	COLORID [ID][R][G][B]	COLORID 1 45 56 64	[R][G][B] can be values of 0-255
	7	COLORID [ID]	COLORID 1	Return color values of COLORID index
COMMENT	50	COMMENT	COMMENT Anything	Same as remark
COPY	56	COPY [name1][name2]	COPY "Droid.bin" "Droid.ezm"	Copy file name1 to file name2
CWD	51	CWD	CWD	Display current working directory

DEL	58	DEL [name]	DEL "Droid.ezm"	Delete file name
DIR	55	DIR [mask]	DIR *.ezm"	Returns directory using mask
ECHO		ECHO [mode]	ECHO off	Sets mode 0 = OFF or 1 = ON
ERASE	58	ERASE [name]	ERASE "Droid.ezm"	Delete file name. Must be in the directory the file is located in.
FONT	10	FONT [font]	FONT 0	Use internal font (factory) Currently 0 and 2 = default medium font
	10	FONT [font]	FONT "med_36"	Use programmable font (ezf file) from flash drive
FONTO	12	FONTO [orientation]	FONTO 0	Set font orientation 0=horizontal, 1=vertical
	12	FONTO	FONTO	Return font orientation
FONTW	11	FONTW [ID][font]	FONTW 0 “med_36”	Use programmable widget font (ezf file) from flash drive for each of 16 Themes.
FORMAT	60	FORMAT [password] [volume_label]	FORMAT "ezLCD" "EarthRules"	Formats by removing files in flash drive. Type the command exactly as it is shown in example.
HELP		HELP [filename]	HELP ameter	Prints command's syntax and some information about it to the terminal.
IO	38	IO [GPIO][DATA]	IO 1 0	IO write GPIO (1-9) with DATA. Write data to peripheral if configured.
	38	IO [GPIO][DATA]		IO READ GPIO. Return DATA from peripheral if configured.
LIGHT	5	LIGHT [brightness]	LIGHT 87	Backlight level to brightness when brightness is 0 to 100%. Default is 75%
	5	LIGHT	LIGHT	Return current backlight level setting
LINE	18	LINE [x][y]	LINE 75 70	Place a line from current XY to X,Y with current color and current width and current type
LINETYPE	14	LINETYPE [type]	LINETYPE 2	Sets line type to solid, dot or dash. 0=solid, number increases spacing between dots
	14	LINETYPE	LINETYPE	Returns line type
LINEWIDTH	13	LINEWIDTH [pixel]	LINEWIDTH 1	Sets line width to pixel width. 1 or 3 pixels wide
	13	LINEWIDTH	LINEWIDTH	Returns line width
LOOP	34	LOOP [ON]/[OFF] or [0]/[1]	LOOP ON	Crtl+C to stop loop or put LOOP OFF in macro
MD	53	MD [directory]	MD "MAPS"	Make directory
MKDIR	53	MKDIR [directory]	MKDIR "FARMS"	Make directory
MORE	59	MORE [name]	MORE "Droid.ezm"	Sends the content of the file name to the current console port
PAUSE	33	PAUSE [ms]	PAUSE 500	Delay in milliseconds
PICTURE	24	PICTURE [x][y][options][index]	PICTURE 0 0 3 1	Display PICTURE ID index on the LCD. File can be JPG, GIF or BMP X Y is the upper left corner of the image when no options are applied. Options of 1=align to center of screen, 2=down scale image to screen, 3=both.
	24	PICTURE [x][y][options][file]	PICTURE 0 0 3 "Boats.gif"	Display PICTURE file on the LCD. File can be JPG, GIF or BMP

PIE	22	PIE [radius][start][end]	PIE 55 120 140	Draw PIE with Radius, Start Angle, and End angle Angle of 0 is on the right.
PING	3	PING	PING	Asks LCD to send acknowledge with PONG
PLAY	31	PLAY [name]	PLAY "Droid"	Play macro "name", Looks in directory \EZUSER\MACROS first and if not found it looks in \EZSYS\MACROS
PLOT	17	PLOT	PLOT	Place a pixel at current XY with current color
	17	PLOT [x][y]	PLOT 12 44	Place a pixel at X Y with current color
PRINT	25	PRINT [ID][alignment]	PRINT 1	Print string in String Array pointed to by index to the display. 9 font justifications: LT, CT, RT, LC, CC, RC, LB, CB, RB
	25	PRINT [string][alignment]	PRINT "Hello Earth" CC	Print string to the display at current X Y. 9 font justifications: LT, CT, RT, LC, CC, RC, LB, CB, RB
RD	58	RD [directory]	RD "FARMS"	Remove directory
RECORD	30	RECORD [name]	RECORD "Droid"	Record macro "name" to internal flash drive under \EZUSER\MACROS
REM	50	REM	REM Does not matter	Remark also ' ," ,#
REN	57	REN [name1][name2]	REN "Droid.bin" "Droid.ezm"	Rename file name1 to name2
RESET	29	RESET	RESET	Reset ezLCD as if just turned on
RMDIR	54	RMDIR [directory]	RMDIR "MAPS3"	Remove directory from current directory
SECURITY	40	SECURITY [option][password]	security Set "ezlcd123"	Lock the flash drive so it can not be accessed from the PC
			security Reset "ezlcd123"	Set = set password and locks the flash drive. Reset = put in password to unlock. Set/Reset must have the first letter capitalized.
SNAPSHOT	48	SNAPSHOT [x][y][width][height] [filename]	SNAPSHOT 30 30 50 50 “test”	Takes a picture of the specified area. File name must be in double quotes.
SPEED	35	SPEED [ms]	SPEED 100	Delay between macro line processing in milliseconds
STOP	32	STOP	STOP	Stops recording you macro and closes file
STRING	16	STRING [ID][string]	STRING 1 "Randy"	Store string in the string array using index. Index=0-61
	16	STRING [ID]	STRING 1	Recall string from string array using index
THRESHOLD	105	THRESHOLD [value]	THRESHOLD 256	Value used for sensitivity of touch screen. Default is 256
TYPE	59	TYPE [name]	TYPE "Droid.ezm"	Sends the content of the file name to the current console port
VERBOSE	106	VERBOSE [ON] or [OFF] or [0] or [1]	VERBOSE ON	Sets verbose (command echo) mode on(1) or off(0)

WAIT	102	WAIT [Option]	WAIT T	Wait for events. Options: “T” = Touch, “!T” =No Touch, “TR” = Touch and release, “IO[1-9]” = Wait for 1 (choose a pin number [1-9]), “!IO[1-9]” = Wait for 0. No option = Touch and Release
XY	15	XY [x][y]	XY 50 50	Set drawing cursor to location x,y on screen. x and y are checked to make sure they can fit on the screen. 9 font justifications: LT, CT, RT, LC, CC, RC, LB, CB, RB offers convenient screen placement options.
	15	XY	XY	Return current x,y location
XYID	41	XYID [ID][1]	XYID 1 1	Save X and Y into XY array using index
	41	XYID [ID]	XYID 1	Restore X and Y from XY array using index
Widget Commands	**SHORT FORM**	**SYNTAX**	**EXAMPLE**	**DESCRIPTION & OPTIONS**
AMETER	76	AMETER [ID][x][y][width][height] [options][value][min][max][theme] [tringID]	AMETER 1 25 25 400 240 1 200 0 500 1 2	Options: 1=draw , 2=disabled, 3=ring, 4=accuracy.
AMETER_ COLOR	78	AMETER_COLOR [ID][color1] [color2][color3][color4][color5] [color6]	AMETER_COLOR 1 4 5 6 blue green yellow	Change the colors used for the 6 analog meter quadrants.
AMETER_ VALUE	77	AMETER_VALUE [ID][Value]	AMETER_VALUE 1 55	Update the value of an Analog Meter and redraw it.
BUTTON	70	BUTTON [ID][x][y][width][height] [options][align][radius][theme] [stringID]	BUTTON 1 25 25 75 75 1 0 0 2 1	Options: 1=draw , 2=disabled, 3=pressed, 4=toggle. Align 0=centered, 1=right, 2=left, 3=bottom, 4=top.
CHECKBOX	71	CHECKBOX [ID][x][y][width] [height][option][theme][stringID]	CHECKBOX 1 30 30 225 50 1 2 0 3	Options: 1=draw , 2= disabled, 3=checked, 4=redraw .
CHOICE	89	CHOICE [string][theme]	CHOICE “Ready to fire!” 1	Question String to display with scheme and get response. Response: 1=yes, 0=no and -1=cancel
DIAL	80	DIAL [ID][x][y][radius][options] [resolution][value][max][theme] [stringID]	DIAL 1 200 120 75 1 1 15 100 2	Options: 1=draw , 2=disable.
DMETER	74	DMETER [ID][x][y][width][height] [option][value][digits][DP][theme]	DMETER 1 50 50 100 50 1 3 2 1 2	Options: 1=left, 2=disabled, 3=right, 4=center, 11=left framed, 12=disabled framed, 13=right framed, 14=center framed, 6=redraw .
DMETER_VALUE	75	DMETER_VALUE [ID][value]	DMETER_VALUE 2 57	Update the value of a Digital Meter and redraw it.
GROUPBOX	72	GBOX [ID][x][y][width][height] [options][theme][stringID]	GBOX 3 0 0 300 200 4 2 1	Options: 1=left, 2=disabled, 3=right. 4=center aligned
PROGRESS	85	PROGRESS [ID][x][y][width][height] [options][value][range][theme]	PROGRESS 4 0 100 399 139 3 45 50 2	Options: 1=horizontal, 2=H disabled, 3=vertical, 4=V disabled, 5=redraw bar.
PROGRESS_VALUE	86	PROGRESS_VALUE [ID][value]	PROGRESS_VALUE 1 46	Update the Value of a progress bar and redraw it.

RADIO BUTTON	73	RADIO [ID][x][y][w idth][height] [option][theme][stringID]	RADIO 4 50 50 100 50 4 2 1	Options: 1=draw , 2=disabled, 3=checked, 4=first, 5=first and checked.
SLIDER	82	SLIDER [ID][x][y][w idth][height] [options][range][resolution][value] [theme]	SLIDER 1 20 30 100 50 1 75 5 25 1	Options: 1=draw hori, 2=hori disabled, 3=vert, 4=vert disabled, 5=hori scrollbar, 6=hori scrollbar disabled, 7=vert scrollbar, 8=vert scrollbar disable.
STATIC	87	STATIC [ID][x][y][w idth][height] [option][theme][stringID]	STATIC 1 25 25 200 75 1 2 1	Options: 1=left, 2=disabled , 3=right , 4=center, 5=left framed, 6=disabled framed, 7=right framed, 8=center framed , 9=redraw text.
STATIC_VALUE	88	STATIC_VALUE [ID][string]	STATIC_VALUE 1 “Hello”	Update the text of a Static Text Box and redraw it
THEME	90	[EmbossLtColor][TextColor0] [TextColor1][TextColorDisabled] [Color0][Color1][ColorDisabled]	THEME 0 1 2 3 4 5 6 7 9 10	Set Widget Theme Info. Note: Fontw requires Firmw are Version 1.06 or later!
WSTATE	92	WSTATE [ID][options]	WSTATE 1 3	Options: 0 = delete, 1 = enable, 2 = disable, 3 = redraw

THE ezLCD-30x SOFTWARE AND HARDWARE IS PROVIDED TO YOU "AS IS," AND WE MAKE NO EXPRESS OR IMPLIED WARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNCTIONALITY, OPERABILITY, OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR INFRINGEMENT. WE EXPRESSLY DISCLAIM ANY LIABILITY WHATSOEVER FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST REVENUES, LOST PROFITS, LOSSES RESULTING FROM BUSINESS INTERRUPTION OR LOSS OF DATA, REGARDLESS OF THE
FORM OF ACTION OR LEGAL THEORY UNDER WHICH THE LIABILITY MAY BE ASSERTED, EVEN IF ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

Appendix E: Upgrading the ezLCD-30x Firmware

A Windows PC is required to upgrade the firmware on an ezLCD-30x. There are two parts to upgrading the ezLCD-30x firmware**.**

1) Putting the ezLCD in firmware upgrade mode**.**

2) Run the Firmware Loader to load the firmware from your P.C. to the ezLCD-30x using the USB port.

Before starting **an upgrade** be sure you have downloaded the ezLCD30x Firmware Loader **and installed it. The latest** firmware **can be found at** www.EarthLCD.com/ezLCD-30x.

IMPORTANT: Never use any upgrade firmware that is not designed for the display you have. Only ezLCD-30x firmware should be installed. Using the wrong firmware will make your unit inoperable and leave no way to install the correct firmware.

Before upgrading you ezLCD firmware you should backup any macros you have created by copying them from the ezLCD-30x flash drive to your computer.

Have your ezLCD-30x installed and running with the terminal program as shown in the ezLCD-30x **Getting Started** section of this manual.

NOTE: Once you put the ezLCD in firmware upgrade mode it cannot come out of this state until new firmware is programmed using the provided program even if you unplug it!

Step 1. Put the ezLCD in **Firmware Upgrade Mode**. Type in the following command line: **Upgrade ezLCD.** The command must be typed exactly and is case sensitive. You should receive the message: **Upgrade Firmware Enabled**.

Step 2. Close your terminal program.

Step 3. Unplug the ezLCD from the USB port.

Step 4. Run the ezLCD-30x Firmware Loader program (should already be running).

Step 5. Plug the ezLCD-30x back into USB. It will only display a dim gray screen when in upgrade mode. The ezLCD-30x Firmware Loader program will beep and the text box should display **Device attached**.

Step 6. Click **Open Hex File** in the ezLCD-30x Firmware Loader program.

Step 7. Navigate to your ezLCD-30x firmware file and click on it (does not show file is loaded).

Step 8. Click P**rogram/Verify** in the ezLCD-30x Firmware Loader program. The ezLCD-30x Firmware Loader text box should display several status messages followed by **Erase/Program/Verify**

completed Successfully.

Step 9. Click **Reset Device** in the ezLCD-30x Firmware Loader program. It should sign back on with the firmware version you loaded displayed in the bottom left corner of the ezLCD-30x splash screen. If you get the ‘ **FSINIT FAILED’** instead of the splash screen you will need to reformat the ezLCD flash drive. Format the ezLCD, using quick format by right clicking the drive in file manager and selecting the button "Restore Device Defaults".

Step 10. Load the new file system if you re-formatted or downloaded a new file system from the EarthLCD website.

Step 11. Reconnect your terminal program and enjoy your firmware upgrade.

Appendix F: Installing & Using the ezLCD-30x on a MAC (OS X Lion (10.7)

The ezLCD-30x requires OS X version 10.7 or later to run on a MAC. The good news there are no drivers or utilities to install.

Step 1. Run the Mac Terminal program
Use spotlight or navigate in Finder to the Applications/Utilities folder and run the Terminal application.

Step 2. Determine the ezLCD-30x USB device name.
Plug in your ezLCD-30x to the USB port. At the Terminal command prompt type LS /dev/tty.* (note: /dev/tty MUST be lower case). All your tty compatible devices will list including one that starts with 'usbmodem'. That is your usb device name for the ezLCD-30x. It will be different on different computers like /dev/tty.usbmodemfa132 for example. If you see more then one USB device you can unplug your ezLCD-30x to see which
one goes away and then plug it back in to get the device name.

Step 3. Set the terminal mode to serial port mode using the screen command (usbmodemfa132 should be replaced with the result of Step 2):

screen /dev/tty.usbmodemfa132

Step 4. Type CLS and the ezLCD-30x screen should clear and you can goto section 4.6 of this manual to continue learning how to use your ezLCD-30x.

Editing Macros with TextEdit program on your MAC, You may use TextEdit that comes with your MAC to create and modify ezLCD macro files but you need to be sure that you use text format not rich text format (rtf). To assure this, navigate in Finder to any file (like demo.ezm in \EZSYS\MACROS and press Option on your keyboard and right click your mouse at the same time and choose 'Open With' and select TextEdit.