



# User Manual

**ezLCD-3xx Product Family**

**Version 1.0 Jan 10, 2012**

**Requires Firmware Version 1.1 or Later**

## [1.0 Introduction](#)

## [2.0 About This Manual](#)

### [2.1 One Hundred Dollars - The e.z. way!](#)

## [3.0 How the ezLCD-30x Works](#)

## [4.0 Installation and Getting Started](#)

### [4.1 Connect the ezLCD USB to Your PC](#)

### [4.2 Verify the ezLCD USB Flash Drive Operation](#)

### [4.3a Installing the USB Driver on a Windows 7 Operating System](#)

### [4.3b Installing the USB Driver on a Windows XP Operating System](#)

### [4.4 Run the “Termie” Terminal Program](#)

### [4.5 Verify Connection](#)

### [4.6 Flash Drive Access](#)

## [5.0 Command your ezLCD with EarthSEMP](#)

### [5.1 ezLCD-30x Grammar](#)

### [5.2 Creating and Saving Macros](#)

### [5.3 Comment Always](#)

## [6.0 EarthSEMP Macros](#)

### [6.1 STARTUP.EZM - Your Most Important Macro](#)

## [7.0 Images](#)

### [7.1 Resizing an Image in Photoshop](#)

### [7.2 Resizing an Image in Windows Paint](#)

## [8.0 Colors](#)

### [8.1 Set Current Color](#)

### [8.2 Set Custom Color](#)

## [9.0 Drawing](#)

### [9.1 Drawing a single pixel](#)

### [9.2 Drawing a line](#)

### [9.3 Drawing a box](#)

[9.4 Drawing a circle](#)

[9.5 Drawing a circle section](#)

[9.6 Drawing an arc](#)

[10.0 Fonts and the Print Command](#)

[11.0 Widget Themes](#)

[12.0 Widgets](#)

[12.1 Analog Meter Widget - AMETER](#)

[12.2 Button Widget](#)

[12.3 Check Box Widget - CHECKBOX](#)

[12.4 Choice Widget - CHOICE](#)

[12.5 Dial Widget - DIAL](#)

[12.6 Digital Meter Widget - DMETER](#)

[12.7 Groupbox Widget - GBOX](#)

[12.8 Progress Widget - PROGRESS](#)

[12.9 Radio Button Widget - RADIO](#)

[12.10 Slider Widget - SLIDER](#)

[12.11 Static Text Widget - STATIC](#)

[13.00 Snapshot Widget - SNAPSHOT](#)

[14.0 Flash Drive File Structure](#)

[15.0 Ports & Embedding the ezLCD-30x](#)

[15.1 The ezLCD-30x Command Port](#)

[15.2 Command Port Management](#)

[16.0 Start Something with your ezLCD-30x](#)

[17.0 Warnings, Errata and Gotchas](#)

[18.0 Gratis \(a note from Randy Schafer\)](#)

[Appendix A: ezLCD-30x Connector Pinout](#)

[Appendix B: ezLCD-30x Model Descriptions and Drawings](#)

[Appendix C: EarthSEMPL Colors](#)

[Appendix D: EarthSEMP Command Reference Guide](#)

[TABLE 1](#)

[Appendix E: Upgrading the ezLCD-30x Firmware](#)

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

## 1.0 Introduction

The ezLCD-30x 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-30x into that CNC mill he was always tinkering with in his garage. We hope that you enjoy using your ezLCD-30x 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-30x, 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-30x it is a family manual for the entire ezLCD-3xx family of products. The Appendix will describe the different models. All ezLCD-30x 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-30x 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-30x 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 [store.EarthLCD.com/ezLCD-30x](http://store.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 [store.EarthLCD.com/ezLCD-30x](http://store.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](mailto: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-30x Smart LCD consists of an LCD module and a controller board containing the graphics processor, memory and interfaces. The ezLCD-30x 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-30x, 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-30x into a USB port achieves the following:

- **Powers the ezLCD-30x**
- **Connects the ezLCD USB flash drive to your computer**
- **Opens a USB CDC COM port connection**

The ezLCD-30x 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-30x FlashDrive]

Eventually, the ezLCD-30x will be capable of running as a standalone controller. However, many ezLCD-30x customers will use the ezLCD-30x 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-30x through a serial port. The ezLCD-30x 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-30x 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 **ezLCD.inf** file (which is included on the ezLCD-30x 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-30x Smart LCD**
- **ezLCD-30x 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-30x USB to Your PC

### 4.2 Verify the ezLCD-30x 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.

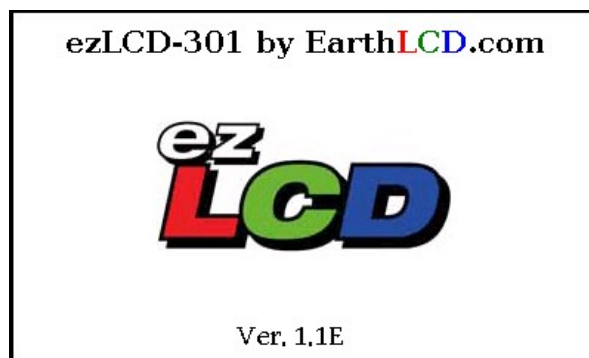


Figure 1



## 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.

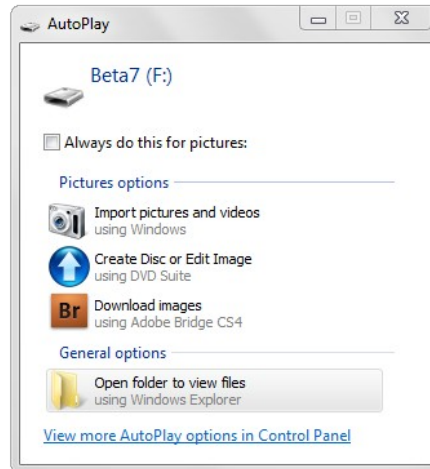


Figure 2

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.

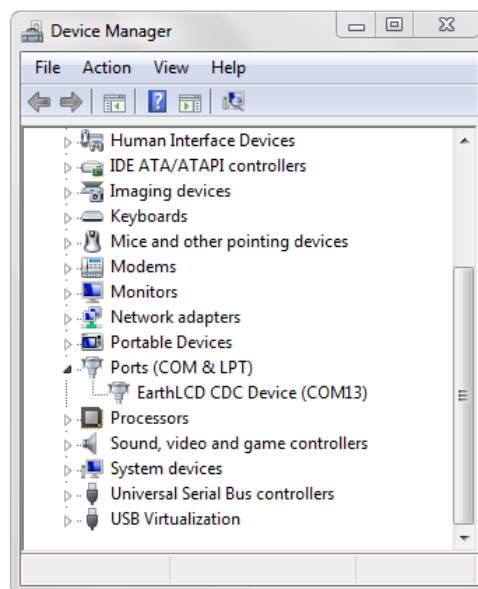


Figure 3

#### 4.4 Run the “Termie” Terminal Program

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).

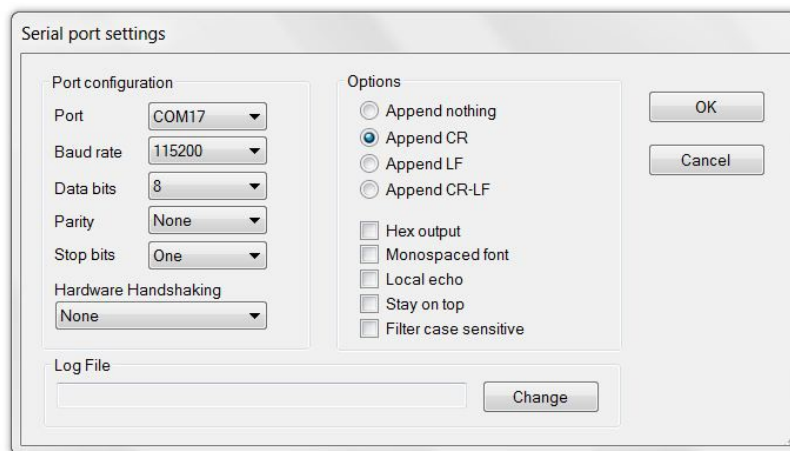


Figure 4

#### 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 than 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 EarthSEMPLE

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 **EarthSEMPLE** for short. We will use the terms **EarthSEMPLE program** and **macro** interchangeably in this manual.

### 5.1 ezLCD-30x Grammar

The syntax or grammar for EarthSEMPLE commands is:

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

EarthSEMPLE 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 apostrophe ( ' ) 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 EarthSEMPLE 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

EarthSEMPLE 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 Comment Always

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 EarthSEMPLE 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 EarthSEMPLE 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-30x model on it's product page at the EarthLCD website. The ezLCD-30x latest file system is located at <http://store.earthlcd.com/ezLCD-30x> . Note: Substitute your model number for ezLCD-30x.**

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-30x 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.

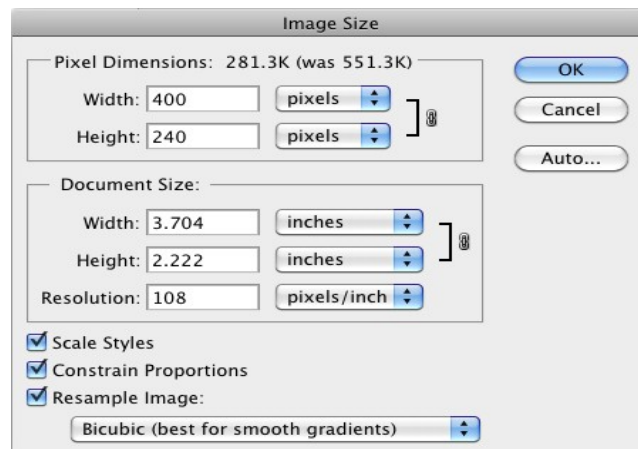


Figure 5

## 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**.

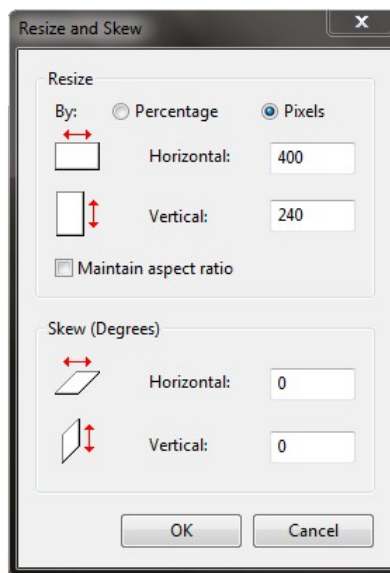


Figure 6

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-30x 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 {Just}** command. **{Just}** 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.

### 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-30x 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-30x.

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://store.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. 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-30x 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

'	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.

## 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 take user input and output 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-30x 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-30x 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 customisable) 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-30x 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. Some of the widgets require touch screen input to be useful and may have limited use for ezLCD-30x displays with no touchscreen.*
- 6. 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 that looks like a car speedometer. It's companion command, AMETER\_VALUE, allows you to set the needle value without redrawing the whole 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}**

#### EXAMPLE:

<b>STRING 1 "MPH"</b>	<b>'Label Widget MPH</b>
<b>FONTW 5 "Sans24"</b>	<b>'Set Widget font 5 to small font</b>
<b>AMETER 1 0 0 150 150 1 100 0 120 5 1</b>	<b>'draw ameter widget</b>
<b>AMETER_VALUE 1 60</b>	<b>'ameter to 60 (does not redraw widget, just needle)</b>

**Result:**



The AMETER command contains ten different values.

The **{id}** value **1**, 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, **0** and **0**, designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **{Width}** and **{Height}** values, **150** and **50**, designate the width and height of the widget in pixels.

The **{Options}**, **1**, designates the DRAW option of the analog meter.

[It is recommended that only option 1 be used at this time, and to delete the widget, use CLS]

Option choices: **1=draw, 2=disabled, 3=ring, 4=accuracy**.

The **{Value}** value, **100**, designates the initial value setting of the needle on the meter.

The **{Min}** value, **0**, designates the minimum value on the meter scale.

The **{Max}** value, **120**, designates the maximum value on the meter scale.

The **{Theme}** value, **5**, sets widget to theme 5.

The **{StringID}**, **1**, designates the id number of the text string that you'd like displayed below the meter.

## 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. This is the ideal replacement for a switch in an embedded application. Your ezLCD-30x has the ability to display custom-themed buttons.

**BUTTON {id} {X} {Y} {Width} {Height} {Options} {Align} {Radius} {Theme} {StringID}**

**Example:**     **STRING 0 STOP**

**'Set button label to STOP**

**BUTTON 1 10 10 100 100 1 0 10 2 0**

**Result:**



**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**, 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, **10** and **10**, designate the location of the button on the screen as the **XY** coordinate of the upper left corner.

The {**Width**} and {**Height**} values, **100** and **100**, designate the width and height of the button in pixels.

The {**Options**}, **1**, designates the option of the button.

Option choices: **1=draw, 2=disabled, 3=pressed, 4=toggle.**

The {**Align**} value, **0**, 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, **10**, 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.

The {**Theme**} value, **2**, 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**}, **0**, 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 stringID contents. Example: string 5 "Wrap\nText" will appear on 2 lines.

### 12.3 Check Box Widget - CHECKBOX

The CHECKBOX widget allows you to display a check box with a string next to it that 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-30x has the ability to display custom-themed CHECKBOX's.

**CHECKBOX {id} {X} {Y} {Width} {Height} {Options} {Theme} {StringID}**

**Example:**

**STRING 1 TEST**

**CHECKBOX 1 30 30 225 50 1 1 1**

**Result:**



**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 **1**, 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, **30** and **30**, designate the location of the widget on the screen as the XY coordinate of the upper left corner.

The {Width} and {Height} values, **225** and **50**, designate the width and height of the widget in pixels.

The {Options}, **1**, designates the option "unchecked".

Option choices: **1=draw unchecked, 2=draw disabled, 3=draw checked, 4=redraw**

The {Theme} value, **1**, sets widget to theme 1.

The {StringID}, **1**, designates the id number of the text string that you'd like displayed next to the text string.

### 12.4 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-30x has the ability to display custom-themed CHOICE’s.

**CHOICE {String} {Theme}**

**Example:**

**CHOICE "Got Milk?" 1**

**‘quote around string**

**Result:**



**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.

The {Theme} value, **1**, sets widget to theme 1.

{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**

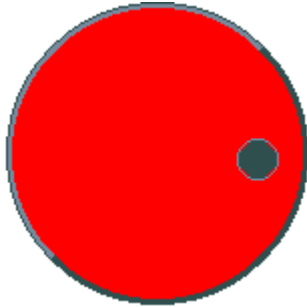
### 12.5 Dial Widget - DIAL

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} {Options} {Resolution} {Value} {Max} {Theme}**

**Example:**

**DIAL 1 100 85 75 1 1 25 100 1**

**Result:**

**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**, 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, **100** and **85**, designate the location of the widget on the screen as the **XY** coordinate of the center of the dial.

The **{Radius}** values, **75**, means that the radius of the dial is 75, which the diameter will be 150.

The **{Options}**, **1**, means the dial is drawn. Option choices: **1=draw, 2=disabled**.

The **{Resolution}** value, **1**, designates the increments in the range. So the indicator will be a value of every number, such as, 10,11,12,13,14,15....

The **{Value}** value, **25**, designates the initial dial value.

The **{Max}** value, **100**, designates the largest value of the dial's input.

The **{Theme}** value, **1**, sets widget to theme 1. Note: Themes 0-7 are predefined in the STARTUP.EZM macro.

### 12.6 Digital Meter Widget - DMETER

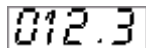
The DMETER widget allows you to display a digital meter as in a panel meter. It's companion command the DMETER\_VALUE command allows you to set the read out value without redrawing the meter. The DMETER.EZM file in the \EZSYS\MACROS directory contains an example of this widget. It can be invoked by typing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

**DMETER {id} {X} {Y} {Width} {Height} {Options} {Value} {Digits} {DotPosition} {Theme}**

**Dmeter\_Value {id} {Value}**

**Example:**

```
CLS WHITE
FONTW 2 LCD24
THEME 1 0 2 0 0 0 0 1 3 3 2
DMETER 1 50 50 70 22 14 0120 4 1 1
DMETER_VALUE 1 123
```

**Result:**


The DMETER command contains ten different values.

The **{id}** value **1**, 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, **50** and **50**, designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **{Width}** and **{Height}** values, **70** and **50**, designate the width and height of the widget in pixels.

The **{Options}**, **14**, centers the value in the frame.

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}** **0123**, designates and displays the initial setting of the readout as it appears on the meter.

The **{Digits}** value, **4**, designates the number of digits displayed on the meter.

The **{DotPos}** value, **1**, designates the position of the decimal point from the 'right' most number.

The **{Theme}** value, **1**, sets widget to theme 1.

### 12.7 Groupbox Widget - GBOX

The GBOX widget generates a border/box and by changing the Options allows the strings within the box to be printed 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**

**STRING 1 FRUITS**

**GBOX 1 20 30 300 200 1 2 1**

**XY 25 60**

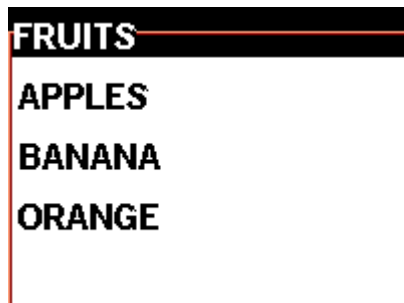
**PRINT "APPLES"**

**XY 25 90**

**PRINT "BANANA"**

**XY 25 120**

**PRINT "ORANGE"**

**Result:**

The GBOX command contains eight different values.

The **{id}** value **1**, 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, **20** and **30**, designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **{Width}** and **{Height}** values, **300** and **200**, designate the width and height of the widget in pixels.

The **{Options}**, **1**, designates the option of the analog meter.

Option choices: **1=left,2=disabled,3=right,4=center**

The **{Theme}** value, **2**, sets widget to theme 2.



The **{StringID}**, **1**, designates the id number of the text string that you'd like displayed below the meter.

### 12.8 Progress Widget - PROGRESS

The PROGRESS widget allows you to display a percent 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-30x 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} {Options} {Max} {Value} {Theme}**

**PROGRESS\_VALUE {id} {Value}**

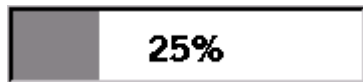
**Example:**

**THEME 1 0 2 0 0 0 3 1 3 3 2**

**PROGRESS 1 50 100 180 40 1 100 25 1**

**PROGRESS\_VALUE 1 26**

**Result:**



The PROGRESS command contains nine different values and PROGRESS\_VALUE contains two values.

The **{id}** value **1**, 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, **50 100**, designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **{Width}** **{Height}** values, **180 40**, designate the width and height of the widget in pixels.

The **{Options}**, **1**, 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 **{Max}** value, **100**, designates the maximum value that can be reached.

The **{Value}** value, **25**, designates the initial percentage value. By using the **PROGRESS\_VALUE** command changes the value from **25** to **26**.

The **{Theme}** value, **1**, sets widget to theme 1.

### 12.9 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, while checkboxes can have many filled in. Therefore, radio buttons work as a group; they 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} {Options} {Theme} {StringID}**

**Example:**

```
CLS WHITE

STRING 0 "HIGH"

STRING 1 "MEDIUM"

STRING 2 "LOW"

THEME 1 1 2 0 0 0 3 1 1 3 0

RADIO 0 25 50 230 35 5 1 0          'HIGH
RADIO 1 25 95 230 35 1 1 1          'MED
RADIO 2 25 140 230 35 1 1 2         'LOW
```

**Results:**



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 **0 /1/ 2**, 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, **25** and **50**, designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **{Width}** and **{Height}** values, **230** and **35**, 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}**, **5**, designates the option “**first checked**”. Should be set to 4 or 5 for first button in the first group on the screen then the rest in that group follow with option set to 1 or 3. More than one group of radio buttons can occupy the screen by using the option 4 ‘**first unchecked**’ or 5 ‘**first checked**’ in the first line of the second group. For example, if you want the 4th button on the screen to be in a new group, then you would put the **Option** value as **4** or **5**, then all the following buttons will be in that second group until a later button is flagged ‘first’ by setting 4 or 5 again.

Option choices: **1=unchecked, 2=disabled, 3=checked, 4=FIRST unchecked, 5=FIRST checked**

The **{Theme}** value, **1**, sets widget to theme 1.

The **{StringID}**, **2**, designates the id number of the text string that you’d like displayed by the button.

### 12.10 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.

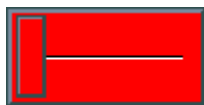
**SLIDER {id} {X} {Y} {Width} {Height} {Options} {Range} {Resolution} {Value} {Theme}**

**Example:**

**CLS WHITE**

**SLIDER 1 20 30 100 50 1 75 5 25 1**

**Result:**



**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**, 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, **20** and **30**, designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The **{Width}** and **{Height}** values, **100** and **50**, designate the width and height of the widget in pixels.

The **{Options}**, **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, **75**, designates that the minimum and maximum value is 0-75.

The **{Resolution}** value, **5**, 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, **25**, designates the initial value of the indicator.

The **{Theme}** value, **1**, sets widget to theme 1.

### 12.11 Static Text Widget - STATIC

The STATIC widget generates a framed text box with a string at different alignments. If you want to change the text in this box it changes without having to overwrite background color first. It displays a label that the user can read, but not change. 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} {Options} {Theme} {StringID}**

**Example:**

**STRING 1 "HELLO WORLD"**

**STATIC 1 10 25 220 25 5 1 1**

**Result:**

**HELLO WORLD**

The STATIC command contains eight different values.

The **{id}** value **1**, 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, **10** and **25**, designate the location of the widget on the screen as the **XY** coordinate of the upper left corner.

The {Width} and {Height} values, **220** and **25**, designate the width and height of the widget in pixels.

The {Options}, **1**, designates the **Left** option of the STATIC widget.

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} value, **1**, sets widget to theme 1.

The {StringID}, **1**, 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 press F5 on your PC to refresh windows explorer with its contents. You can then open the USER IMAGES file to see your snapshot on the PC.

***REMEMBER*** to press F5 before trying to open the image. The file will not appear until you have done so.

***NOTE:*** This command is **BETA** and was created for convenience in capturing images for this manual. It is provided “as is” for now.

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, **0** and **0**, designate the starting XY coordinate at the upper left corner.

The {Width} and {Height} values, **200** and **150**, designate the area of the screen for SNAPSHOT to capture.

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.

## 14.0 Flash Drive File Structure

The ezLCD-30x USB flash drive appears as a removable storage device on the **Host** computer. In Windows, click on **Start**, then **Computer** and your ezLCD-30x USB flash drive will appear. By double clicking on it you may access the ezLCD-30x 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)

**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-30x

While the ezLCD-30x 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 microcontroller like an AVR, PIC or ARM microcontroller. 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-30x to other microcontrollers can be found on the ezLCD-30x product page at [store.EarthLCD.com/ezLCD-30x](http://store.EarthLCD.com/ezLCD-30x). We suggest that Newbies take a look at the Arduino application notes.

### 15.1 The ezLCD-30x Command Port

Ports are the way the ezLCD-30x talks to the outside world. Ports on the ezLCD-30x include USB, Serial, SPI and I2C. 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 microcontroller 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 TWO which is set by the following command:

**CMD SERIAL2 115200 1 N81**

Note: You cannot communicate with the ezLCD with any terminal program unless the command port is set to the correct COM port. If you want to use the USB port type the command:

**CMD CDC**

In this manual we showed you how to use the ezLCD-30x using the full ASCII **long** command. Every ezLCD-30x command has two formats: **long** and **short**. On your microcontroller, 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 microcontroller host in the **\EZUSER\MACROS** directory. The **\EZUSER\MACRO** directory will be searched at **powerup** 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-30x

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-30x 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 webpage at [store.earthlcd.com/ezLCD-30x](http://store.earthlcd.com/ezLCD-30x). File updates, firmware and examples will also be provided there, so bookmark it and check back often.

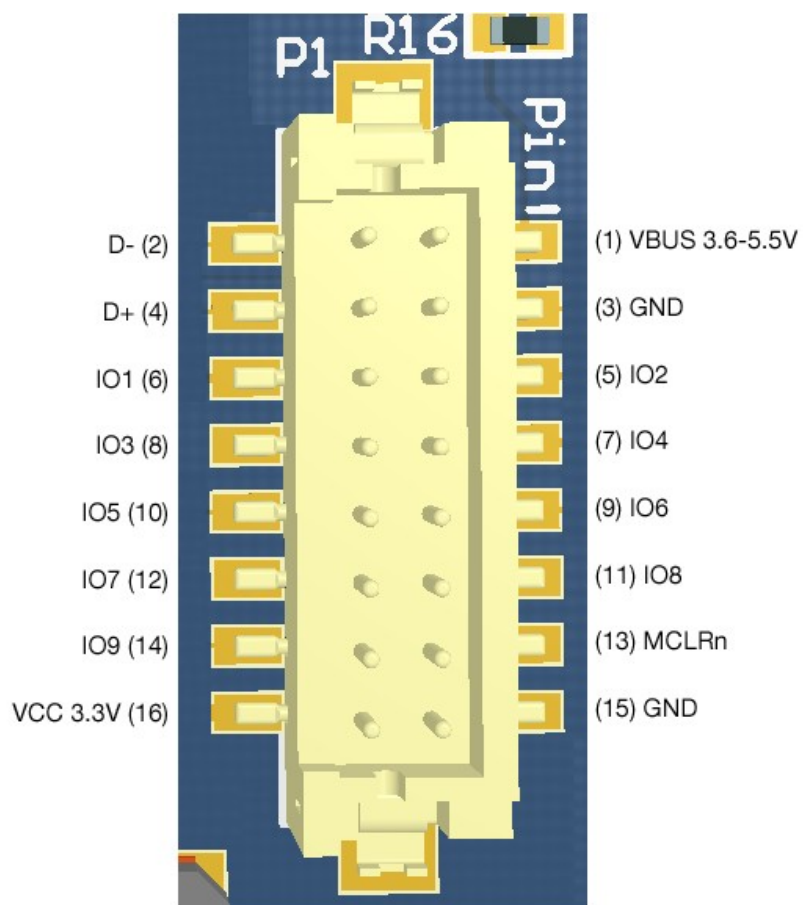
## 17.0 Warnings, Errata and Gotchas

- **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-30x!

## Appendix A: ezLCD-30x Connector Pinout



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

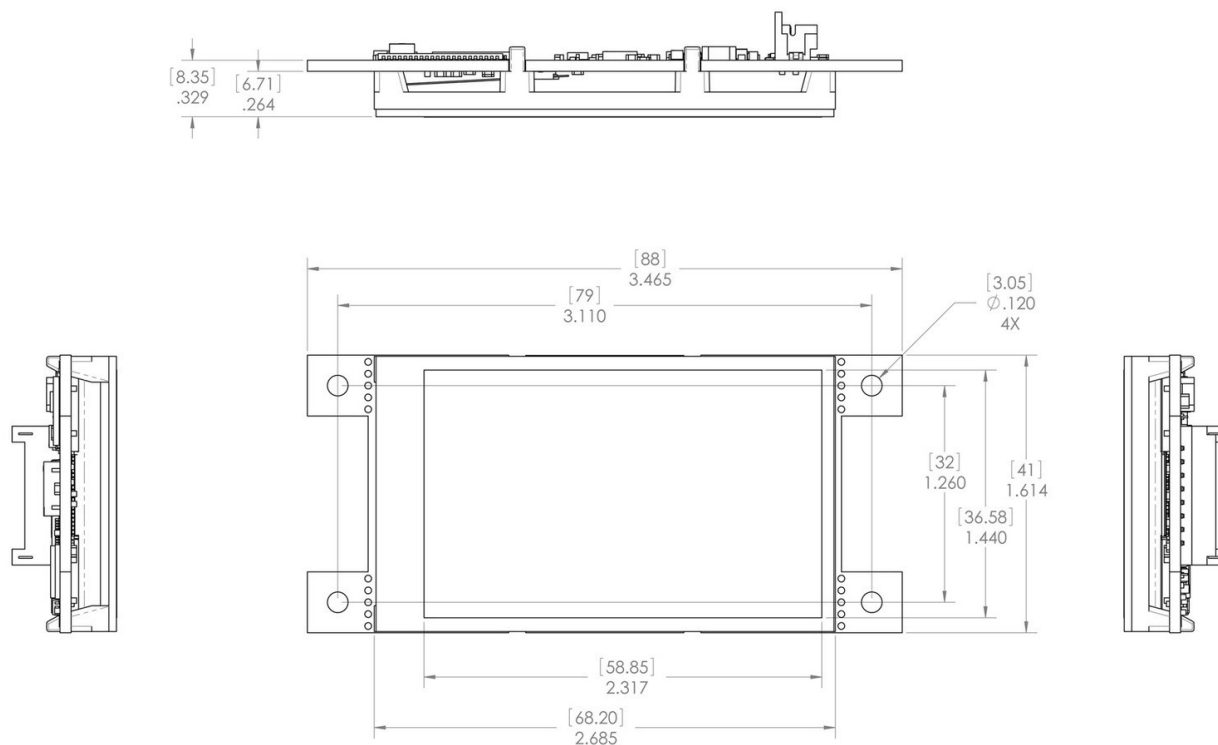
The I/O pins are programmable for different I/O functions in the future but the current firmware supports the following command to set the command port to serial port 2 on IO3 & IO4 :

`CMD SERIAL2 115200 1 N81` ‘ Sets 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



**ezLCD-302:** (Available Jan 2012) 240 x 160 (4:3) 4096 Transflective Color TFT (Sunlight Readable - NO Touchscreen)

## Appendix C: EarthSEMP Colors

<pre> Color_ID[ 0 ] = BLACK;      Color_ID[ 1 ] = GRAY;     Color_ID[ 2 ] = SILVER;     Color_ID[ 3 ] = WHITE;     Color_ID[ 4 ] = RED;     Color_ID[ 5 ] = MAROON;     Color_ID[ 6 ] = YELLOW;     Color_ID[ 7 ] = OLIVE;     Color_ID[ 8 ] = LIME;     Color_ID[ 9 ] = GREEN;     Color_ID[ 10 ] = AQUA;     Color_ID[ 11 ] = TEAL;     Color_ID[ 12 ] = BLUE;     Color_ID[ 13 ] = NAVY;     Color_ID[ 14 ] = FUCHSIA;//Magenta     Color_ID[ 15 ] = PURPLE;  //red colors      Color_ID[ 16 ] = INDIANRED;     Color_ID[ 17 ] = LIGHTCORAL;     Color_ID[ 18 ] = SALMON;     Color_ID[ 19 ] = DARKSALMON;     Color_ID[ 20 ] = LIGHTSALMON;     Color_ID[ 21 ] = RED;     Color_ID[ 22 ] = CRIMSON;     Color_ID[ 23 ] = FIREBRICK; </pre>	<pre> //Yellow colors (continued)      Color_ID[ 41 ] =LIGHTGOLDENRODYELLOW      Color_ID[ 42 ] = PAPAYAWHIP;     Color_ID[ 43 ] = MOCCASIN;     Color_ID[ 44 ] = PEACHPUFF;     Color_ID[ 45 ] = PALEGOLDENROD;     Color_ID[ 46 ] = KHAKI;     Color_ID[ 47 ] = DARKKHAKI;  //Purple colors      Color_ID[ 48 ] = LAVENDER;     Color_ID[ 49 ] = THISTLE;     Color_ID[ 50 ] = PLUM;     Color_ID[ 51 ] = VIOLET;     Color_ID[ 52 ] = ORCHID;     Color_ID[ 53 ] = FUCHSIA;     Color_ID[ 54 ] = MEDIUMORCHID;     Color_ID[ 55 ] = MEDIUMPURPLE;     Color_ID[ 56 ] = BLUEVIOLET;     Color_ID[ 57 ] = DARKVIOLET;     Color_ID[ 58 ] = DARKORCHID;     Color_ID[ 59 ] = DARKMAGENTA;     Color_ID[ 60 ] = PURPLE;     Color_ID[ 61 ] = INDIGO;     Color_ID[ 62 ] = DARKSLATEBLUE; </pre>
--	--

<pre> Color_ID[ 24 ] = DARKRED;  //Pink colors  Color_ID[ 25 ] = PINK;  Color_ID[ 26 ] = LIGHTPINK;  Color_ID[ 27 ] = HOTPINK;  Color_ID[ 28 ] = DEEPPINK;  Color_ID[ 29 ] =MEDIUMVIOLETRED;  Color_ID[ 30 ] = PALEVIOLETRED;  //Orange colors  Color_ID[ 31 ] = LIGHTSALMON;  Color_ID[ 32 ] = CORAL;  Color_ID[ 33 ] = TOMATO;  Color_ID[ 34 ] = ORANGERED;  Color_ID[ 35 ] = DARKORANGE;  Color_ID[ 36 ] = ORANGE;  //Yellow colors  Color_ID[ 37 ] = GOLD;  Color_ID[ 38 ] = YELLOW;  Color_ID[ 39 ] = LIGHTYELLOW;  Color_ID[ 40 ] = LEMONCHIFFON; </pre>	<pre> Color_ID[ 63 ] = SLATEBLUE;  Color_ID[ 64 ] = MEDIUMSLATEBLUE;  //Green colors  Color_ID[ 65 ] = GREENYELLOW;  Color_ID[ 66 ] = CHARTREUSE;  Color_ID[ 67 ] = LAWNGREEN;  Color_ID[ 68 ] = LIME;  Color_ID[ 69 ] = LIMEGREEN;  Color_ID[ 70 ] = PALEGREEN;  Color_ID[ 71 ] = LIGHTGREEN;  Color_ID[ 72 ] = MEDIUMSPRINGGREEN;  Color_ID[ 73 ] = SPRINGGREEN;  Color_ID[ 74 ] = MEDIUMSEAGREEN;  Color_ID[ 75 ] = SEAGREEN;  Color_ID[ 76 ] = FORESTGREEN;  Color_ID[ 77 ] = GREEN;  Color_ID[ 78 ] = DARKGREEN;  Color_ID[ 79 ] = YELLOWGREEN;  Color_ID[ 80 ] = OLIVEDRAB;  Color_ID[ 81 ] = OLIVE;  Color_ID[ 82 ] = DARKOLIVEGREEN; </pre>

<pre> Color_ID[ 83 ] = MEDIUMAQUAMARINE; Color_ID[ 84 ] = DARKSEAGREEN;  Color_ID[ 85 ] = LIGHTSEAGREEN; </pre>	<pre> //White colors continued  Color_ID[ 133 ] = AZURE;  Color_ID[ 134 ] = ALICEBLUE; </pre>
---	---



Color_ID[ 86 ] = DARKCYAN;	Color_ID[ 135 ] = GHOSTWHITE;
Color_ID[ 87 ] = TEAL; //Blue/Cyan colors	Color_ID[ 136 ] = WHITESMOKE;
Color_ID[ 88 ] = AQUA;	Color_ID[ 137 ] = SEASHELL;
Color_ID[ 89 ] = CYAN;	Color_ID[ 138 ] = BEIGE;
Color_ID[ 90 ] = LIGHTCYAN;	Color_ID[ 139 ] = OLDLACE;
Color_ID[ 91 ] = PALETURQUOISE;	Color_ID[ 140 ] = FLORALWHITE;
Color_ID[ 92 ] = AQUAMARINE;	Color_ID[ 141 ] = IVORY;
Color_ID[ 93 ] = TURQUOISE;	Color_ID[ 142 ] = ANTIQUEWHITE;
Color_ID[ 94 ] = MEDIUMTURQUOISE;	Color_ID[ 143 ] = LINEN;
Color_ID[ 95 ] = DARKTURQUOISE;	Color_ID[ 144 ] = LAVENDERBLUSH;
Color_ID[ 96 ] = CADETBLUE;	Color_ID[ 145 ] = MISTYROSE;
Color_ID[ 97 ] = STEELBLUE;	//Gray colors
Color_ID[ 98 ] = LIGHTSTEELBLUE;	Color_ID[ 146 ] = GAINSBORO;
Color_ID[ 99 ] = POWDERBLUE;	Color_ID[ 147 ] = LIGHTGREY;
Color_ID[ 100 ] = LIGHTBLUE;	Color_ID[ 148 ] = SILVER;
Color_ID[ 101 ] = SKYBLUE;	Color_ID[ 149 ] = DARKGRAY;
Color_ID[ 102 ] = LIGHTSKYBLUE;	Color_ID[ 150 ] = GRAY;
Color_ID[ 103 ] = DEEPSKYBLUE;	Color_ID[ 151 ] = DIMGRAY;
Color_ID[ 104 ] = DODGERBLUE;	Color_ID[ 152 ] = LIGHTSLATEGRAY;
Color_ID[ 105 ] = CORNFLOWERBLUE;	Color_ID[ 153 ] = SLATEGRAY;
Color_ID[ 106 ] = ROYALBLUE;	Color_ID[ 154 ] = MEDIUMTURQUOISE;
Color_ID[ 107 ] = BLUE;	Color_ID[ 155 ] = DARKSLATEGRAY;
Color_ID[ 108 ] = MEDIUMBLUE;	Color_ID[ 156 ] = BLACK;
Color_ID[ 109 ] = DARKBLUE;	//extra colors
Color_ID[ 110 ] = NAVY;	Color_ID[ 157 ] = GRAY7;
Color_ID[ 111 ] = MIDNIGHTBLUE;	Color_ID[ 158 ] = GRAY20;
//Brown colors	Color_ID[ 159 ] = GRAY40;
Color_ID[ 112 ] = CORNSILK;	Color_ID[ 160 ] = GRAY80;

<pre>Color_ID[ 113 ] = BLANCHEDALMOND; Color_ID[ 114 ] = BISQUE; Color_ID[ 115 ] = NAVAJOWHITE; Color_ID[ 116 ] = WHEAT; Color_ID[ 117 ] = BURLYWOOD; Color_ID[ 118 ] = TAN; Color_ID[ 119 ] = ROSYBROWN; Color_ID[ 120 ] = SANDYBROWN; Color_ID[ 121 ] = GOLDENROD;  Color_ID[ 125 ] = SADDLEBROWN; //White colors Color_ID[ 129 ] = WHITE; Color_ID[ 130 ] = SNOW; Color_ID[ 131 ] = HONEYDEW; Color_ID[ 132 ] = MINTCREAM;</pre>	<pre>Color_ID[ 161 ] = GRAY90; Color_ID[ 162 ] = GRAY95; Color_ID[ 163 ] = RED4; Color_ID[ 164 ] = FIREBRICK1; Color_ID[ 165 ] = DARKGREEN; Color_ID[ 166 ] = PALEGREEN; Color_ID[ 167 ] = LIGHTYELLOW;</pre>

## Appendix D: EarthSEMP Command Reference Guide

A quick reference guide of the EarthSEMP command set that can run on the ezLCD-30x are listed below.

Input values can be up to 65535. Strings can be upto 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
RESET	29	RESET	RESET	Reset ezLCD as if just turned on
CLS	2	CLS index	CLS 1	Clear screen to COLORID[color] and clear widgets. Bad name picks BLACK
	2	CLS color	CLS RED	Clear screen to COLORID[color] and clear widgets. Bad name picks BLACK
PING	3	PING	PING	Asks LCD to send acknowledge with PONG
VERBOSE	106	VERBOSE [ON] or [OFF] or [0] or [1]	VERBOSE ON	Sets verbose (command echo) mode on(1) or off(0)
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
COLOR	6	COLOR index	COLOR 0	Set current color to COLORID[index] (0-199)
	6	COLOR	COLOR	Return current color
COLORID	7	COLORID index R G B	COLORID 1 45 56 64	Set color id n to R, G, B
	7	COLORID index	COLORID 1	Return color values of COLORID index
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
FONTW	11	FONTW index font	FONTW 0 "med_36"	Use programmable widget font (ezf file) from flash drive for each of 16 Themes.
FONTO	12	FONTO orientation	FONTO 0	Set font orientation 0=horizontal, 1=vertical
	12	FONTO	FONTO	Return font orientation
THRESHOLD	105	THRESHOLD value	THRESHOLD 256	Value used for sensitivity of touch screen.

				Default is 256
<b>LINEWIDTH</b>	13	LINEWIDTH pixel	LINEWIDTH 1	Sets line width to pixel width. 1 or 3 pixels wide
	13	LINEWIDTH	LINEWIDTH	Returns line width
<b>LINETYPE</b>	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
<b>XY</b>	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
<b>XYID</b>	41	XYID index 1	XYID 1 1	Save X and Y into XY array using index
	41	XYID index	XYID 1	Restore X and Y from XY array using index
<b>STRING</b>	16	STRING index string	STRING 1 "Randy"	Store string in the string array using index. Index=0-61
	16	STRING index	STRING 1	Recall string from string array using index
<b>PLOT</b>	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
<b>LINE</b>	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
<b>BOX</b>	19	BOX width height fill	BOX 50 50 F	Place a box from current XY with specified width and height. Fill=1 or F to fill box or string starting with F. Box must be > 1 pixel wide
<b>CIRCLE</b>	20	CIRCLE radius fill	CIRCLE 75 1	Draw circle at current XY with radius. Fill=1 or F to fill circle.
<b>ARC</b>	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.
<b>PIE</b>	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.
<b>PICTURE</b>	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
<b>PRINT</b>	25	PRINT index [just]	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 [just]	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

<b>CLIPENABLE</b>	46	CLIPENABLE enable	CLIPENABLE ON	Turn on clip enable area
<b>CLIPAREA</b>	47	CLIPAREA	CLIPAREA left top right bottom	ClipArea to protect the surrounding area from change.
<b>RECORD</b>	30	RECORD name	RECORD "Droid"	Record macro "name" to internal flash drive under \EZUSER\MACROS
<b>PLAY</b>	31	PLAY name	PLAY "Droid"	Play macro "name", Looks in directory \EZUSER\MACROS first and if not found it looks in \EZSYS\MACROS
<b>STOP</b>	32	STOP	STOP	Stop playing macro and close the file.  You may have to restart the display to see the files from the PC side.
<b>WAIT</b>	102	WAIT {Option}	WAIT T	Wait for events. Options: "T" = Touch, "IT" =No Touch, "TR" = Touch and release, "IO1-9" = Wait for 1, "IO1-9" = Wait for 0. No option = Touch and Release
<b>PAUSE</b>	33	PAUSE delay	PAUSE 500	Pause delay msec
<b>LOOP</b>	34	LOOP [ON]/[OFF] or [0]/[1]	LOOP ON	Loop any macro that run. Loop should not be used in a macro.
<b>SPEED</b>	35	SPEED delay	SPEED 100	Delay between macro line processing (msec)
<b>REM</b>	50	REM	REM Does not matter	Remark also ' , " , #
<b>COMMENT</b>	50	COMMENT	COMMENT Anything	Same as remark
<b>ERASE</b>	58	ERASE name	ERASE "Droid.ezm"	Delete file name
<b>DEL</b>	58	DEL name	DEL "Droid.ezm"	Delete file name
<b>REN</b>	57	REN name1 name2	REN "Droid.bin" "Droid.ezm"	Rename file name1 to name2
<b>CHDIR</b>	52	CHDIR directory	CHDIR "MACROS"	Change to directory
<b>CD</b>	52	CD directory	CD "FONTS"	Change to directory
<b>CWD</b>	51	CWD	CWD	Display current working directory
<b>MD</b>	53	MD directory	MD "MAPS"	Make directory
<b>MKDIR</b>	53	MKDIR directory	MKDIR "FARMS"	Make directory
<b>RD</b>	58	RD directory	RD "FARMS"	Remove directory
<b>RMDIR</b>	54	RMDIR directory	RMDIR "MAPS3"	Remove directory from current directory
<b>MORE</b>	59	MORE name	MORE "Droid.ezm"	Sends the content of the file name to the current console port
<b>TYPE</b>	59	TYPE name	TYPE "Droid.ezm"	Sends the content of the file name to the current console port
<b>COPY</b>	56	COPY name1 name2	COPY "Droid.bin" "Droid.ezm"	Copy file name1 to file name2
<b>DIR</b>	55	DIR mask	DIR *.ezm"	Returns directory using mask
<b>FORMAT</b>	60	FORMAT password serial [volume_label]	FORMAT "ezLCD" "EarthRules"	Format "ezLCD" "volumeid"
<b>SECURITY</b>	40	SECURITY password string	SECURITY "ezLCD" lockdrive123	Lock the flash drive so it can not be accessed from the PC
<b>CMD</b>	62	CMD Interface Baudrate Cmd Modebits	CMD SERIAL2 115200 1 N81	Select command to interface. Baud rate=110 to 230400. com=1 for command port.

				mode=N91/2,O81/2,E81/2,N81/2
<b>CFGIO</b>	<b>37</b>	CFGIO GPIO TYPE	CFGIO 1 0	Configure GPIO as input (0) or output (1), serial, SPI, I2C
	<b>37</b>		CFGIO 1 IN	Can use IN for direction
	<b>37</b>		CFGIO 4 OUT	Can use OUT for direction
<b>IO</b>	<b>38</b>	IO GPIO DATA	IO 1 0	IO write GPIO (1-9) with DATA. Write data to peripheral if configured.
	<b>38</b>	IO GPIO DATA		IO READ GPIO. Return DATA from peripheral if configured.
<b>Widget Commands</b>		<b>Syntax</b>	<b>Example</b>	<b>Options</b>
<b>THEME</b>	<b>90</b>	Theme index, EmbossDkColor, EmbossLtColor, TextColor0, TextColor1, TextColorDisabled, Color0, Color1, ColorDisabled, CommonBkColor, Fontw	Theme 0 1 2 3 4 5 6 7 9 10	Set Widget Theme Info. Note: Fontw requires Firmware Version 1.06 or later!
<b>AMETER_VALUE</b>	<b>77</b>	Ameter id Value	Ameter_Value 1 55	Update the value of an Analog Meter and redraw it.
<b>ANALOG METER</b>	<b>76</b>	Ameter id X Y Width Height Options Initial Min Max Theme StringID	Ameter 1 25 25 400 240 1 200 0 500 1 2	Options: 1=draw, 2=disabled, 3=ring, 4=accuracy.
<b>BUTTON</b>	<b>70</b>	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.
<b>CHECKBOX</b>	<b>71</b>	Checkbox id X Y Width Height options Theme StringID	Checkbox 1 30 30 225 50 1 2 0 3	Options: 1=draw, 2= disabled, 3=checked, 4=redraw.
<b>CHOICE</b>	<b>89</b>	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
<b>DIGITAL METER</b>	<b>74</b>	Dmeter id X Y Width Height Options Value Digits DotPos 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.
<b>DMETER_VALUE</b>	<b>75</b>	Dmeter_Value Id Value	Dmeter_Value 2 57	Update the value of a Digital Meter and redraw it.
<b>GROUPBOX</b>	<b>72</b>	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
<b>PROGRESS</b>	<b>85</b>	Progress id X Y Width Height Options Position 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.
<b>PROGRESS_VALUE</b>	<b>86</b>	Progress_Value Id Value	Progress_Value 1 46	Update the Value of a progress bar and redraw it.
<b>RADIO BUTTON</b>	<b>73</b>	Radio id X Y Width Height OptionsTheme StringID	Radio 4 50 50 100 50 4 2 1	Options: 1=draw, 2=disabled, 3=checked, 4=first, 5=first and checked.
<b>STATIC TEXT</b>	<b>87</b>	Static id X Y Width Height	STATIC 1 25 25 200 75 1 0 2 1	Options: 1=left, 2=disabled , 3=right ,

		Options    Frame    Theme StringID		4=center, 5=left framed, 6=disabled framed, 7=right framed, 8=center framed ,9=redraw text.
<b>STATICTEXT_VALUE</b>	88	StaticText_Value id String	StaticText_Value 1 "Hello"	Update the text of a Static Text Box and redraw it
<b>AMETER_COLOR</b>	78	Ameter 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.
<b>DIAL</b>	80	Dial id X Y Radius Options Resolution Initial Max Theme StringID	Dial 1 200 120 75 1 1 15 100 2	Options: 1=draw, 2=disable.
<b>SLIDER</b>	82	Slider id X Y Width 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.

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** [store.EarthLCD.com/ezLCD-30x](http://store.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:

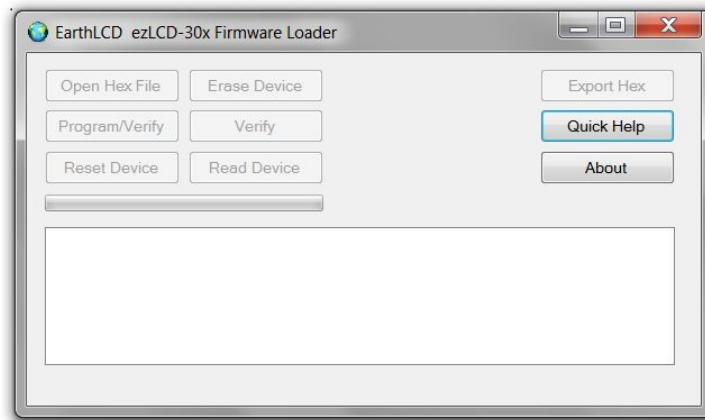
**Upgrade Firmware Enabled.**

**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**.

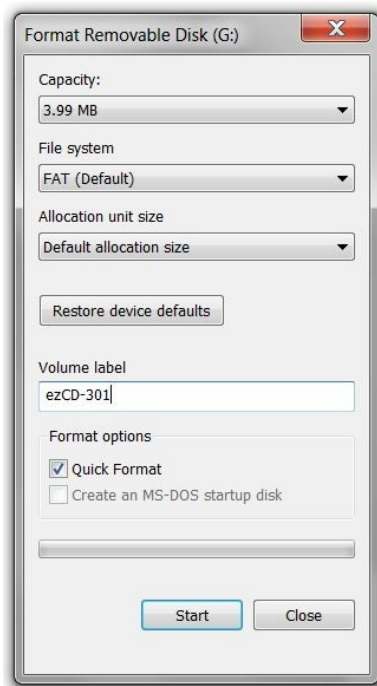
**Step 6.** Click **Open Hex File** in the ezLCD-30x Firmware Loader program.

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

**Step 8.** Click **Program/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 than 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.