

ezLCD Python Module

1.02

Generated by Doxygen 1.8.3.1

Wed Jul 17 2013 16:10:16

Contents

1	Main Page	1
2	Module Index	3
2.1	Modules	3
3	Namespace Index	5
3.1	Namespace List	5
4	Hierarchical Index	7
4.1	Class Hierarchy	7
5	Class Index	9
5.1	Class List	9
6	Module Documentation	11
6.1	Commands	11
6.1.1	Detailed Description	12
6.1.2	Function Documentation	12
6.1.2.1	backlight	12
6.1.2.2	cfgio	12
6.1.2.3	io	12
6.1.2.4	ping	12
6.1.2.5	play	12
6.1.2.6	reset	13
6.1.2.7	run	13
6.1.2.8	snapshot	13
6.1.2.9	verbose	13
6.1.2.10	wquiet	13
6.1.2.11	xmax	13
6.1.2.12	ymax	14

6.2	Primitive Drawing Commands	15
6.2.1	Detailed Description	15
6.2.2	Function Documentation	15
6.2.2.1	arc	15
6.2.2.2	box	16
6.2.2.3	circle	16
6.2.2.4	clipArea	16
6.2.2.5	clipEnable	16
6.2.2.6	cls	16
6.2.2.7	color	17
6.2.2.8	colorId	17
6.2.2.9	line	17
6.2.2.10	lineType	17
6.2.2.11	lineWidth	17
6.2.2.12	pie	18
6.2.2.13	plot	18
6.2.2.14	xy	18
6.3	Widgets	19
6.3.1	Detailed Description	19
6.3.2	Function Documentation	19
6.3.2.1	ameter	19
6.3.2.2	ameter_color	20
6.3.2.3	button	20
6.3.2.4	choice	20
6.3.2.5	dial	21
6.3.2.6	dmeter	21
6.3.2.7	fontw	21
6.3.2.8	groupBox	22
6.3.2.9	progressBar	22
6.3.2.10	radioButton	22
6.3.2.11	slider	23
6.3.2.12	staticText	23
6.3.2.13	string	23
6.3.2.14	theme	24
6.3.2.15	touchZone	24
6.3.2.16	wstack	24
6.3.2.17	wstate	25

6.3.2.18	wvalue	25
6.4	Bitmaps and Fonts	26
6.4.1	Detailed Description	26
6.4.2	Function Documentation	26
6.4.2.1	font	26
6.4.2.2	fonto	26
6.4.2.3	picture	27
6.4.2.4	printString	27
7	Namespace Documentation	29
7.1	ezLCD3xx Namespace Reference	29
7.1.1	Detailed Description	32
7.1.2	Function Documentation	32
7.1.2.1	__init__	32
7.1.2.2	closeSerial	32
7.1.2.3	WaitForCR	32
8	Class Documentation	33
8.1	ezLCD3xx.ezLCD Class Reference	33
Index		33

Chapter 1

Main Page



Figure 1.1: height=2cm



Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Commands	11
Primitive Drawing Commands	15
Widgets	19
Bitmaps and Fonts	26

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

ezLCD3xx	29
------------------------------------	--------------------

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

object	
ezLCD3xx.ezLCD	33

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ezLCD3xx.ezLCD	33
--	----

Chapter 6

Module Documentation

6.1 Commands

Functions

- def [ezLCD3xx.verbose](#)
The Verbose command will turn on or off more verbose errors.
- def [ezLCD3xx.xmax](#)
The xmax command will return the max x of current display.
- def [ezLCD3xx.ymax](#)
The ymax command will return the max y of current display.
- def [ezLCD3xx.ping](#)
the ping command
- def [ezLCD3xx.backlight](#)
The backlight command will set backlight brightness and timeout.
- def [ezLCD3xx.wquiet](#)
The wquiet command disables the touch event data being sent to the console port.
- def [ezLCD3xx.cfgio](#)
The cfgio command will configure io pins.
- def [ezLCD3xx.io](#)
The io command use to set and clear io pins.
- def [ezLCD3xx.play](#)
The play command will play a macro stored on the drive of the [ezLCD](#).
- def [ezLCD3xx.run](#)
The run command will run a macro stored on the drive of the [ezLCD](#).
- def [ezLCD3xx.reset](#)
The reset command will reset the [ezLCD](#) and run startup.ezm same as power up.
- def [ezLCD3xx.snapshot](#)
The snapshot command will write a copy of the current display to the flash drive as a bmp.

6.1.1 Detailed Description

6.1.2 Function Documentation

6.1.2.1 `def ezLCD3xx.backlight (self, brightness, timeout=None, level=None)`

The backlight command will set backlight brightness and timeout.

Parameters

<i>brightness</i>	1
<i>timeout</i>	2
<i>level</i>	3

6.1.2.2 `def ezLCD3xx.cfgio (self, pin, function)`

The cfgio command will configure io pins.

Parameters

<i>pin</i>	
<i>function</i>	

6.1.2.3 `def ezLCD3xx.io (self, pin, level=None)`

The io command use to set and clear io pins.

Parameters

<i>pin</i>	
<i>level</i>	

Returns

io level

6.1.2.4 `def ezLCD3xx.ping (self)`

the ping command

Returns

0

6.1.2.5 `def ezLCD3xx.play (self, filename)`

The play command will play a macro stored on the drive of the [ezLCD](#).

Parameters

<i>macro</i>	filename
--------------	----------

6.1.2.6 def ezLCD3xx.reset (self)

The reset command will reset the ezLCD and run startup.ezm same as power up.

6.1.2.7 def ezLCD3xx.run (self, filename)

The run command will run a macro stored on the drive of the ezLCD.

Parameters

<i>macro</i>	filename
--------------	----------

6.1.2.8 def ezLCD3xx.snapshot (self, x, y, w, h, filename)

The snapshot command will write a copy of the current display to the flash drive as a bmp.

Parameters

<i>x</i>	starting x position
<i>y</i>	starting y position
<i>w</i>	width
<i>h</i>	height
<i>filename.bmp</i>	Make sure you have space on the internal flash drive !

6.1.2.9 def ezLCD3xx.verbose (self, state)

The Verbose command will turn on or off more verbose errors.

Parameters

<i>state</i>	0=off 1=on
--------------	------------

6.1.2.10 def ezLCD3xx.wquiet (self, state)

The wquiet command disables the touch event data being sent to the console port.

Parameters

<i>state</i>	0=off 1=on
--------------	------------

6.1.2.11 def ezLCD3xx.xmax (self)

The xmax command will return the max x of current display.

Returns

x-horizontal resolution in pixels starting from 0

6.1.2.12 `def ezLCD3xx.ymax (self)`

The ymax command will return the max y of current display.

Returns

y-vertical resolution in pixels starting from 0

6.2 Primitive Drawing Commands

Functions

- def [ezLCD3xx.cls](#)
The cls command will clear the screen to black if no color is given.
- def [ezLCD3xx.color](#)
The color command see [ezLCD3xx](#) manual for colors.
- def [ezLCD3xx.colorId](#)
The colorId command.
- def [ezLCD3xx.xy](#)
The xy command will set or return the x y coordinates.
- def [ezLCD3xx.plot](#)
The plot command will set a pixel to current color and if used x y.
- def [ezLCD3xx.lineType](#)
The lineType Command will set the line type for the line command.
- def [ezLCD3xx.lineWidth](#)
The lineWidth Command will set the line width for the line command.
- def [ezLCD3xx.line](#)
The line command will draw a line from current xy to line(x,y)
- def [ezLCD3xx.box](#)
The box command will draw a box starting from the current xy in width and height with option for filled.
- def [ezLCD3xx.circle](#)
The circle command will draw a circle in the current xy with radius and optional filled.
- def [ezLCD3xx.pie](#)
The pie command will draw a pie slice at current xy.
- def [ezLCD3xx.arc](#)
The arc command will draw a arc i the current xy optional filled.
- def [ezLCD3xx.clipArea](#)
The cliparea command allows you to designate a rectangular/box area that you can draw in.
- def [ezLCD3xx.clipEnable](#)
The clipenable command enables or disables cliparea.

6.2.1 Detailed Description

6.2.2 Function Documentation

6.2.2.1 def ezLCD3xx.arc(self, radius, start, end, fill = 0)

The arc command will draw a arc i the current xy optional filled.

Parameters

<i>radius</i>	radius of arc
<i>start</i>	start angle
<i>end</i>	end angle
<i>fill</i>	1=filled arc 0=outline only *optional defaults to outline

6.2.2.2 `def ezLCD3xx.box (self, width, height, fill = 0)`

The box command will draw a box starting from the current xy in width and height with option for filled.

Parameters

<i>width</i>	width of box in pixels
<i>height</i>	height of box in pixels
<i>fill</i>	1=filled box 0=outline only *optional defaults to outline

6.2.2.3 `def ezLCD3xx.circle (self, radius, fill = 0)`

The circle command will draw a circle in the current xy with radius and optional filled.

Parameters

<i>radius</i>	radius of circle
<i>fill</i>	1=filled circle 0=outline only *optional defaults to outline

6.2.2.4 `def ezLCD3xx.clipArea (self, left, top, right, bottom)`

The cliparea command allows you to designate a rectangular/box area that you can draw in.

Any surrounding area will be protected and no changes can be made to it

Parameters

<i>left</i>	
<i>top</i>	
<i>right</i>	
<i>bottom</i>	

6.2.2.5 `def ezLCD3xx.clipEnable (self, enable)`

The clipenable command enables or disables cliparea.

Parameters

<i>enable</i>	0=off 1=on
---------------	------------

6.2.2.6 `def ezLCD3xx.cls (self, Color = None)`

The cls command will clear the screen to black if no color is given.

Parameters

<i>Color</i>	color to clear screen to
--------------	--------------------------

6.2.2.7 `def ezLCD3xx.color (self, color = None)`

The color command see [ezLCD3xx](#) manual for colors.

Parameters

<i>color</i>	number
--------------	--------

Returns

color as a tuple

6.2.2.8 `def ezLCD3xx.colorId (self, ID, R=None, G=None, B=None)`

The colorId command.

Parameters

<i>R</i>	Red Value
<i>G</i>	Green Value
<i>B</i>	Blue Value

Returns

color as a tuple if r g b is None

6.2.2.9 `def ezLCD3xx.line (self, x, y)`

The line command will draw a line from current xy to line(x,y)

Parameters

<i>x</i>	
<i>y</i>	

6.2.2.10 `def ezLCD3xx.lineType (self, option)`

The lineType Command will set the line type for the line command.

Parameters

<i>option</i>	0 = solid, 1= dotted (1 pixel spacing between dots), 2 = dashed (2 pixel spacing between dashes)
---------------	--

6.2.2.11 `def ezLCD3xx.lineWidth (self, width)`

The lineWidth Command will set the line width for the line command.

Parameters

<i>width</i>	thin line (width = 1) or a thick line (width =3). Only [width] = 1 or 3 are available.
--------------	--

6.2.2.12 `def ezLCD3xx.pie (self, radius, start, end)`

The pie command will draw a pie slice at current xy.

Parameters

<i>radius</i>	radius of pie
<i>start</i>	start angle
<i>end</i>	end angle

6.2.2.13 `def ezLCD3xx.plot (self, x=None, y=None)`

The plot command will set a pixel to current color and if used x y.

Parameters

<i>x</i>	optional
<i>y</i>	optional

6.2.2.14 `def ezLCD3xx.xy (self, x=None, y=None)`

The xy command will set or return the x y coordinates.

Parameters

<i>x</i>	x position
<i>y</i>	y position

Returns

x y if x and y not supplied

```

1 # Set x y to 100 100
2 LCD.xy(100,100)
3 # Get Current x y
4 (x,y)=LCD.xy()
```


6.3 Widgets

Functions

- def `ezLCD3xx.ameter`
The ameter widget.
- def `ezLCD3xx.ameter_color`
The ameter_color command.
- def `ezLCD3xx.dmeter`
The dmeter widget.
- def `ezLCD3xx.button`
The button command.
- def `ezLCD3xx.choice`
The choice widget allows you to print a string and display buttons for the user to choose a response.
- def `ezLCD3xx.groupBox`
The groupBox widget.
- def `ezLCD3xx.radioButton`
The radioButton widget.
- def `ezLCD3xx.staticText`
The staticText widget.
- def `ezLCD3xx.slider`
The slider command.
- def `ezLCD3xx.progressBar`
The progressBar command.
- def `ezLCD3xx.touchZone`
The touchZone command.
- def `ezLCD3xx.dial`
The dial command.
- def `ezLCD3xx.theme`
The theme command sets the colors for widgets.
- def `ezLCD3xx.fontw`
The fontW command will set the font for widget.
- def `ezLCD3xx.string`
The string command will set or return a internal string.
- def `ezLCD3xx.wstack`
The wstack command will return the stack of widgets pressed 32 levels.
- def `ezLCD3xx.wvalue`
The wvalue command will set or return a value to or from a widget.
- def `ezLCD3xx.wstate`
The wstate command.

6.3.1 Detailed Description

6.3.2 Function Documentation

6.3.2.1 `def ezLCD3xx.ameter (self, ID, x, y, width, height, options, value, minV, maxV, theme, stringID, meterType = 0)`

The ameter widget.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	
<i>value</i>	
<i>minV</i>	
<i>maxV</i>	
<i>theme</i>	
<i>stringID</i>	
<i>meter- Type@verbatim</i>	ameter [ID][x][y][width][height][options][value][minV][maxV][theme][stringID][type]

6.3.2.2 `def ezLCD3xx.ameter_color (self, ID, color1, color2, color3, color4, color5, color6)`

The ameter_color command.

Parameters

<i>ID</i>	
<i>color1</i>	
<i>color2</i>	
<i>color3</i>	
<i>color4</i>	
<i>color5</i>	
<i>color6</i>	

6.3.2.3 `def ezLCD3xx.button (self, ID, x, y, width, height, options, align, radius, theme, stringID, text=None)`

The button command.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	
<i>align</i>	
<i>radius</i>	
<i>theme</i>	
<i>stringID</i>	
<i>text</i>	optional text for button

6.3.2.4 `def ezLCD3xx.choice (self, string, theme, string1=None, string2=None, string3=None)`

The choice widget allows you to print a string and display buttons for the user to choose a response.

Parameters

<i>string</i>	the text about the buttons
<i>theme</i>	the theme ID
<i>string1</i>	string for left button *optional defaults to YES
<i>string2</i>	string for center button *optional defaults to NO
<i>string3</i>	string for right button *optional defaults to CANCEL

Returns

1=left button
 0=center button
 -1=right button

6.3.2.5 `def ezLCD3xx.dial (self, ID, x, y, radius, option, resolution, value, maxx, theme)`

The dial command.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>radius</i>	
<i>option</i>	
<i>resolution</i>	
<i>value</i>	
<i>maxx</i>	
<i>theme@verbatim</i>	dial [ID][x][y][radius][option][resolution][value][max][theme]

6.3.2.6 `def ezLCD3xx.dmeter (self, ID, x, y, width, height, options, value, digits, dp, theme)`

The dmeter widget.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	
<i>value</i>	
<i>digits</i>	
<i>dp</i>	
<i>theme</i>	

6.3.2.7 `def ezLCD3xx.fontw (self, fontnumber, name)`

The fontW command will set the font for widget.

Parameters

<i>fontnumber</i>	number of the font
<i>name</i>	filename of font '0' and '1' are internal fonts

6.3.2.8 `def ezLCD3xx.groupBox (self, ID, x, y, width, height, options, theme, stringID)`

The groupBox widget.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	
<i>theme</i>	
<i>stringID</i>	

6.3.2.9 `def ezLCD3xx.progressBar (self, ID, x, y, width, height, options, value, mmax, theme, stringID)`

The progressBar command.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	
<i>value</i>	
<i>mmax</i>	
<i>theme</i>	
<i>stringID</i>	

6.3.2.10 `def ezLCD3xx.radioButton (self, ID, x, y, width, height, options, theme, stringID)`

The radioButton widget.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	Options: 1=draw , 2=disabled, 3=checked, 4=first, 5=first and checked.
<i>theme</i>	
<i>stringID</i>	

6.3.2.11 `def ezLCD3xx.slider(self, ID, x, y, width, height, options, rrange, resolution, value, theme)`

The slider command.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	
<i>rrange</i>	
<i>resolution</i>	
<i>value</i>	
<i>theme</i>	

6.3.2.12 `def ezLCD3xx.staticText(self, ID, x, y, width, height, options, theme, stringID, text=None)`

The staticText widget.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	Options: 1=left, 2=disabled , 3=right , 4=center, 5=left framed, 6=disabled framed, 7=right framed, 8=center framed , 9=redraw text.
<i>theme</i>	theme
<i>stringID</i>	stringID number
<i>text</i>	text to display *optional

6.3.2.13 `def ezLCD3xx.string(self, stringNumber, string=None)`

The string command will set or return a internal string.

Parameters

<i>stringNumber</i>	number of string to set or return
<i>string</i>	string to set optional internal strings are used for text on buttons and other widgets Strings are defined as 128 characters. There are 64 strings (0 to 63). String 61-63 are used by the CHOICE command. String 64 is temp location. String 65 is the product string String 66 is the firmware string

```
6.3.2.14 def ezLCD3xx.theme ( self, ID, EmbossDkColor, EmbossLtColor, TextColor0, TextColor1, TextColorDisabled, Color0,
                             Color1, ColorDisabled, CommonBkColor, Fontw )
```

The theme command sets the colors for widgets.

Parameters

<i>ID</i>	Theme ID
<i>EmbossDkColor</i>	Dark color for 3d effect
<i>EmbossLtColor</i>	Light color for 3d effect
<i>TextColor0</i>	
<i>TextColor1</i>	
<i>TextColor-Disabled</i>	
<i>Color0</i>	
<i>Color1</i>	
<i>ColorDisabled</i>	
<i>CommonBkColor</i>	
<i>Fontw</i>	widget font for theme

```
6.3.2.15 def ezLCD3xx.touchZone ( self, ID, x, y, width, height, options )
```

The touchZone command.

Parameters

<i>ID</i>	
<i>x</i>	
<i>y</i>	
<i>width</i>	
<i>height</i>	
<i>options</i>	

```
6.3.2.16 def ezLCD3xx.wstack ( self, option )
```

The wstack command will return the stack of widgets pressed 32 levels.

Parameters

<i>option</i>	0=FIFO 1=LIFO 2=CLEAR FIFO First in First out LIFO Last in First out CLEAR Clear the stack
---------------	---

Returns

tuple of ID, Info, Data

Button Widget Values

- ID = widgetID of widget pressed
- Info 1=Pressed and released 2=Cancel 4=Pressed
- Data button state

TouchZone Widget Vaules

- ID = widgetID of widget pressed
- Info 1=Pressed and released 2=Cancel 4=Pressed
- Data button state

Slider Widget Values

- ID = widgetID of widget pressed
- Info 1 = value incremented 2 = value decremented
- Data slider value

CheckBox Widget Values

- ID = widgetID of widget pressed
- Info 4 = checked 1 = unchecked
- Data state

Dial Widget Values

- ID = widgetID of widget pressed
- Info 1 = turned clockwise 2 = turned counter-clockwise
- Data dial value

```
1 # check wstack for button presses
2 (ID, Info, Data) = LCD.wstack(LIFO)
```

6.3.2.17 def ezLCD3xx.wstate (self, ID, option)

The wstate command.

Parameters

<i>ID</i>	widget ID
<i>option</i>	0 = delete, 1 = enable, 2 = disable, 3 = redraw

6.3.2.18 def ezLCD3xx.wvalue (self, ID, value = None)

The wvalue command will set or return a value to or from a widget.

Parameters

<i>ID</i>	
<i>value</i>	

6.4 Bitmaps and Fonts

Functions

- def `ezLCD3xx.picture`
The picture command will display a bitmap in bmp, jpg, gif formats with optional coordinates.
- def `ezLCD3xx.font`
The font command will set current font to use for printString fonts are located in the /EZSYS/FONTS and /EZUSER/FONTS use the ezLCD-3xx Font Converter from earthlcd.com to convert truetype fonts to `ezLCD` format internal fonts will display faster than external fonts.
- def `ezLCD3xx.fonto`
The FONTO command will change the orientation or direction the text prints.
- def `ezLCD3xx.printString`
print string in current color and font and optional coordinates

6.4.1 Detailed Description

6.4.2 Function Documentation

6.4.2.1 def `ezLCD3xx.font (self, font)`

The font command will set current font to use for printString fonts are located in the /EZSYS/FONTS and /EZUSER/FONTS

use the ezLCD-3xx Font Converter from earthlcd.com

to convert truetype fonts to `ezLCD` format

internal fonts will display faster than external fonts.

Parameters

<i>font</i>	font name '0' and '1' are internal fonts '0' is medium and '1' is small 1 # Set font to internal medium font 2 LCD.font('0') 3 # Set font to LCD24 4 LCD.font('LCD24')
-------------	---

6.4.2.2 def `ezLCD3xx.fonto (self, orientation = None)`

The FONTO command will change the orientation or direction the text prints.

Parameters

<i>orientation</i>	0 90 180 270
--------------------	--------------

Returns

orientation current orientation if orientation is not supplied

```

1 LCD.fonto(0)
2 LCD.color(YELLOW)
3 LCD.printString('Hello',100,100)
4 LCD.fonto(90)
5 LCD.color(RED)
6 LCD.printString('Hello',100,100)
7 LCD.fonto(180)
8 LCD.color(BLUE)
9 LCD.printString('Hello',100,100)
10 LCD.fonto(270)
11 LCD.color(GREEN)
12 LCD.printString('Hello',100,100)

```

6.4.2.3 def ezLCD3xx.picture (self, image, x=None, y=None)

The picture command will display a bitmap in bmp, jpg, gif formats with optional coordinates.

Parameters

<i>image</i>	filename of image 'logo.gif'
<i>x</i>	x coordinates
<i>y</i>	y coordinates x y are optional and if not supplied will display image at current xy <pre> 1 # display python.gif at 10 10 2 LCD.picture('python.gif',10,10) 3 # display python.gif at current x y 4 LCD.picture('python.gif') </pre>

6.4.2.4 def ezLCD3xx.printString (self, string, x=None, y=None, orientation=None)

print string in current color and font and optional coordinates

Parameters

<i>string</i>	string to print
<i>x</i>	x coordinates
<i>y</i>	y coordinates
<i>orientation</i>	rotate text direction x y are optional and if not supplied will print string at current xy orientation is optional but if used x y must be supplied ** orientation will be restored to previous orientation after printing string ** <pre> 1 # display string 'Hello World' at 10 10 2 LCD.printString('Hello World',10,10) 3 # display string 'Hello World' at current x y 4 LCD.printString('Hello World') 5 # display string 'Hello World' at 10 10 rotated 90 6 LCD.printString('Hello World',10,10,90) </pre>

Chapter 7

Namespace Documentation

7.1 ezLCD3xx Namespace Reference

Classes

- class [ezLCD](#)

Functions

- def [__init__](#)
ezLCD object
- def **openSerial**
- def [closeSerial](#)
- def [WaitForCR](#)
This is a internal use function.
- def [verbose](#)
The Verbose command will turn on or off more verbose errors.
- def [xmax](#)
The xmax command will return the max x of current display.
- def [ymax](#)
The ymax command will return the max y of current display.
- def [ping](#)
the ping command
- def [backlight](#)
The backlight command will set backlight brightness and timeout.
- def [wquiet](#)
The wquiet command disables the touch event data being sent to the console port.
- def [cfgio](#)
The cfgio command will configure io pins.
- def [io](#)
The io command use to set and clear io pins.
- def [play](#)
The play command will play a macro stored on the drive of the [ezLCD](#).
- def [run](#)

- The run command will run a macro stored on the drive of the [ezLCD](#).
- def [reset](#)

The reset command will reset the [ezLCD](#) and run startup.ezm same as power up.
- def [snapshot](#)

The snapshot command will write a copy of the current display to the flash drive as a bmp.
- def [cls](#)

The cls command will clear the screen to black if no color is given.
- def [color](#)

The color command see [ezLCD3xx](#) manual for colors.
- def [colorId](#)

The colorId command.
- def [xy](#)

The xy command will set or return the x y coordinates.
- def [plot](#)

The plot command will set a pixel to current color and if used x y.
- def [lineType](#)

The lineType Command will set the line type for the line command.
- def [lineWidth](#)

The lineWidth Command will set the line width for the line command.
- def [line](#)

The line command will draw a line from current xy to line(x,y)
- def [box](#)

The box command will draw a box starting from the current xy in width and height with option for filled.
- def [circle](#)

The circle command will draw a circle in the current xy with radius and optional filled.
- def [pie](#)

The pie command will draw a pie slice at current xy.
- def [arc](#)

The arc command will draw a arc in the current xy optional filled.
- def [clipArea](#)

The cliparea command allows you to designate a rectangular/box area that you can draw in.
- def [clipEnable](#)

The clipenable command enables or disables cliparea.
- def [ameter](#)

The ameter widget.
- def [ameter_color](#)

The ameter_color command.
- def [dmeter](#)

The dmeter widget.
- def [button](#)

The button command.
- def [choice](#)

The choice widget allows you to print a string and display buttons for the user to choose a response.
- def [groupBox](#)

The groupBox widget.
- def [radioButton](#)

The radioButton widget.

- def [staticText](#)
The staticText widget.
- def [slider](#)
The slider command.
- def [progressBar](#)
The progressBar command.
- def [touchZone](#)
The touchZone command.
- def [dial](#)
The dial command.
- def [theme](#)
The theme command sets the colors for widgets.
- def [fontw](#)
The fontW command will set the font for widget.
- def [string](#)
The string command will set or return a internal string.
- def [wstack](#)
The wstack command will return the stack of widgets pressed 32 levels.
- def [wvalue](#)
The wvalue command will set or return a value to or from a widget.
- def [wstate](#)
The wstate command.
- def [picture](#)
The picture command will display a bitmap in bmp, jpg, gif formats with optional coordinates.
- def [font](#)
The font command will set current font to use for printString fonts are located in the /EZSYS/FONTS and /EZUSER/FONTS use the ezLCD-3xx Font Converter from earthlcd.com to convert truetype fonts to ezLCD format internal fonts will display faster than external fonts.
- def [fonto](#)
The FONTO command will change the orientation or direction the text prints.
- def [printString](#)
print string in current color and font and optional coordinates

Variables

- int **BLACK** = 0
- int **GRAY** = 1
- int **SILVER** = 2
- int **WHITE** = 3
- int **RED** = 4
- int **MAROON** = 5
- int **YELLOW** = 6
- int **OLIVE** = 7
- int **LIME** = 8
- int **GREEN** = 9
- int **AQUA** = 10
- int **TEAL** = 11

- int **BLUE** = 12
- int **NAVY** = 13
- int **FUCHISA** = 14
- int **PURPLE** = 15
- int **FILLED** = 1
- int **ON** = 1
- int **OFF** = 0
- int **FIFO** = 0
- int **LIFO** = 1
- int **CLEAR** = 2
- int **DELETE** = 0
- int **ENABLE** = 1
- int **DISABLE** = 2
- int **REDRAW** = 3
- [interface](#)
 - open serial port*
- **ser**
- **sio**

7.1.1 Detailed Description

Python Module for earthlcd.com ezLCD 3xx line of displays
<http://earthlcd.com>

(c)2013 ken segler
ken@earthlcd.com
requires pySerial <http://pyserial.sourceforge.net/>

7.1.2 Function Documentation

7.1.2.1 `def ezLCD3xx.__init__(self, interface)`

[ezLCD](#) object

7.1.2.2 `def ezLCD3xx.closeSerial (self)`

close

7.1.2.3 `def ezLCD3xx.WaitForCR (self)`

This is a internal use function.

Chapter 8

Class Documentation

8.1 ezLCD3xx.ezLCD Class Reference

Inheritance diagram for ezLCD3xx.ezLCD:

The documentation for this class was generated from the following file:

- C:/Users/Segler/Documents/GitHub/ezLCD3xxPython/module/ezLCD3xx.py

Index

- `__init__`
 - `ezLCD3xx`, [32](#)
- `ameter`
 - Widgets, [19](#)
- `ameter_color`
 - Widgets, [20](#)
- `arc`
 - Primitive Drawing Commands, [15](#)
- `backlight`
 - Commands, [12](#)
- Bitmaps and Fonts, [26](#)
 - `font`, [26](#)
 - `fonto`, [26](#)
 - `picture`, [27](#)
 - `printString`, [27](#)
- `box`
 - Primitive Drawing Commands, [15](#)
- `button`
 - Widgets, [20](#)
- `cfgio`
 - Commands, [12](#)
- `choice`
 - Widgets, [20](#)
- `circle`
 - Primitive Drawing Commands, [16](#)
- `clipArea`
 - Primitive Drawing Commands, [16](#)
- `clipEnable`
 - Primitive Drawing Commands, [16](#)
- `closeSerial`
 - `ezLCD3xx`, [32](#)
- `cls`
 - Primitive Drawing Commands, [16](#)
- `color`
 - Primitive Drawing Commands, [16](#)
- `colorId`
 - Primitive Drawing Commands, [17](#)
- Commands, [11](#)
 - `backlight`, [12](#)
 - `cfgio`, [12](#)
 - `io`, [12](#)
 - `ping`, [12](#)
 - `play`, [12](#)
 - `reset`, [13](#)
 - `run`, [13](#)
 - `snapshot`, [13](#)
 - `verbose`, [13](#)
 - `wquiet`, [13](#)
 - `xmax`, [13](#)
 - `ymax`, [14](#)
- `dial`
 - Widgets, [21](#)
- `dmeter`
 - Widgets, [21](#)
- `ezLCD3xx`, [29](#)
 - `__init__`, [32](#)
 - `closeSerial`, [32](#)
 - `WaitForCR`, [32](#)
- `ezLCD3xx.ezLCD`, [33](#)
- `font`
 - Bitmaps and Fonts, [26](#)
- `fonto`
 - Bitmaps and Fonts, [26](#)
- `fontw`
 - Widgets, [21](#)
- `groupBox`
 - Widgets, [22](#)
- `io`
 - Commands, [12](#)
- `line`
 - Primitive Drawing Commands, [17](#)
- `lineType`
 - Primitive Drawing Commands, [17](#)
- `lineWidth`
 - Primitive Drawing Commands, [17](#)
- `picture`
 - Bitmaps and Fonts, [27](#)
- `pie`
 - Primitive Drawing Commands, [18](#)
- `ping`
 - Commands, [12](#)
- `play`
 - Commands, [12](#)

- plot
 - Primitive Drawing Commands, 18
- Primitive Drawing Commands, 15
 - arc, 15
 - box, 15
 - circle, 16
 - clipArea, 16
 - clipEnable, 16
 - cls, 16
 - color, 16
 - colorId, 17
 - line, 17
 - lineType, 17
 - lineWidth, 17
 - pie, 18
 - plot, 18
 - xy, 18
- printString
 - Bitmaps and Fonts, 27
- progressBar
 - Widgets, 22
- radioButton
 - Widgets, 22
- reset
 - Commands, 13
- run
 - Commands, 13
- slider
 - Widgets, 23
- snapshot
 - Commands, 13
- staticText
 - Widgets, 23
- string
 - Widgets, 23
- theme
 - Widgets, 23
- touchZone
 - Widgets, 24
- verbose
 - Commands, 13
- WaitForCR
 - ezLCD3xx, 32
- Widgets, 19
 - ameter, 19
 - ameter_color, 20
 - button, 20
 - choice, 20
 - dial, 21
 - dmeter, 21
 - fontw, 21
 - groupBox, 22
 - progressBar, 22
 - radioButton, 22
 - slider, 23
 - staticText, 23
 - string, 23
 - theme, 23
 - touchZone, 24
 - wstack, 24
 - wstate, 25
 - wvalue, 25
- wquiet
 - Commands, 13
- wstack
 - Widgets, 24
- wstate
 - Widgets, 25
- wvalue
 - Widgets, 25
- xmax
 - Commands, 13
- xy
 - Primitive Drawing Commands, 18
- ymax
 - Commands, 14