# ezLCD Python Module 1.02



Generated by Doxygen 1.8.4

Sat Jul 20 2013 13:53:46

# **Contents**

1	Insta	alling th	e Module												1
2	Intro	duction	n To The H	ardwai	re										3
3	Coo	rdinates	s System												5
4	Intro	duction	n To The S	oftwar	е										7
5	Mod	ule Inde	ex												13
	5.1	Module	es				 	13							
6	Nam	espace	Index												15
	6.1	Names	space List				 	15							
7	Hier	archica	l Index												17
	7.1	Class	Hierarchy				 	17							
8	Clas	s Index													19
	8.1	Class	List				 	19							
9	Mod	ule Dod	umentatio	on											21
	9.1	Comm	ands				 	21							
		9.1.1	Detailed	Descrip	tion		 	21							
		9.1.2	Function	Docum	entatio	on .	 	21							
			9.1.2.1	backli	ght .		 	21							
			9.1.2.2	cfgio			 	22							
			9.1.2.3	io			 	22							
			9.1.2.4	ping			 	22							
			9.1.2.5	play .			 	22							
			9.1.2.6	reset			 	22							
			9.1.2.7	run .			 	22							
			9.1.2.8	snaps	hot .		 	23							
			9.1.2.9	verbos	se .		 	23							
			9.1.2.10	wquie	t		 	23							

iv CONTENTS

		9.1.2.11	xmax	23
		9.1.2.12	ymax	23
9.2	Primitv	e Drawing	Commands	24
	9.2.1	Detailed	Description	24
	9.2.2	Function	Documentation	24
		9.2.2.1	arc	24
		9.2.2.2	box	24
		9.2.2.3	circle	25
		9.2.2.4	clipArea	25
		9.2.2.5	clipEnable	25
		9.2.2.6	cls	25
		9.2.2.7	color	25
		9.2.2.8	colorld	26
		9.2.2.9	line	27
		9.2.2.10	lineType	27
		9.2.2.11	lineWidth	27
		9.2.2.12	pie	27
		9.2.2.13	plot	27
		9.2.2.14	xy	28
9.3	Widget	s		29
	9.3.1	Detailed	Description	29
	9.3.2	Function	Documentation	29
		9.3.2.1	ameter	29
		9.3.2.2	ameter_color	30
		9.3.2.3	button	30
		9.3.2.4	choice	30
		9.3.2.5	dial	31
		9.3.2.6	dmeter	31
		9.3.2.7	fontw	31
		9.3.2.8	groupBox	32
		9.3.2.9	progressBar	32
		9.3.2.10	radioButton	32
		9.3.2.11	slider	32
		9.3.2.12	staticText	33
		9.3.2.13	string	33
		9.3.2.14	theme	33
		9.3.2.15	touchZone	34
		9.3.2.16	wstack	34
		9.3.2.17	wstate	35

CONTENTS

9.4	Bitmap	s and Fon	ts			 	 	 	 		 		 		36
	9.4.1	Detailed	Descripti	on .		 	 	 	 		 		 		36
	9.4.2	Function	Docume	ntatio	n	 	 	 	 		 				36
		9.4.2.1	font			 	 	 	 		 				36
		9.4.2.2	fonto .			 	 	 	 		 				36
		9.4.2.3	picture			 	 	 	 		 				37
		9.4.2.4	printStr	ing .		 	 	 	 		 				37
10 Nam	nespace	Documer	ntation												39
10.1	ezLCD	3xx Name	space Re	eferen	ce .	 	 	 	 		 				39
	10.1.1	Detailed	Descripti	on .		 	 	 	 		 				42
	10.1.2	Function	Docume	ntatio	n	 	 	 	 		 				42
		10.1.2.1	init			 	 	 	 		 				42
		10.1.2.2	closeSe	erial .		 	 	 	 		 				42
		10.1.2.3	WaitFor	CR .		 	 	 	 		 				42
11 Clas	s Docui	mentation	1												43
11.1	ezLCD	3xx.ezLC[	O Class F	Refere	nce .	 	 	 	 		 				43
Index															44

# **Installing the Module**

install info here

requires pySerial http://pyserial.sourceforge.net/

Installing the Module

## **Introduction To The Hardware**

The ezLCD modules contains a GPU an related circutry to drive a LCD display, USB interface

Internal 4mb MSD flash drive for storage of fonts, bitmaps and macros.

Display can be controlled through USB CDC Serial or TTL 3.3v Serial .

Once power is applied to the display it starts up and executes startup.ezm, it will look in /EZUSER/MACROS and if not found will look in /EZSYS/USERS .

What this file does in set all defaults for the Display and communcations port.

Including some default widget fonts and themes.

Its best to have a minimal one in the /EZUSER/MACROS directory with only the relevent settings in it .

Sample minimal startup.ezm.

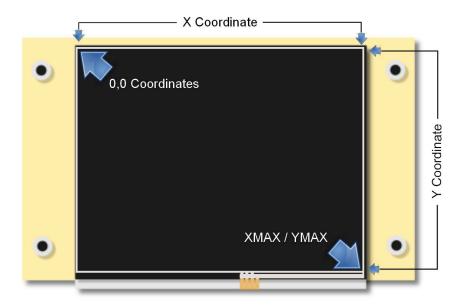
```
'minimal.ezm
verbose off
cmd cdc
'set some fonts for widgets
fontw 0 0
fontw 1 0
fontw 2 0
fontw 3 serif24
fontw 4 serif24
fontw 5 serif24
fontw 6 serif24
fontw 7 serif24
'Set some themes for widgets
theme 5 126 118 3 3 3 35 35 36 0 5
theme 6 111 106 3 3 3 12 12 101 0 6 theme 7 58 48 3 3 3 14 14 54 0 7
color white
print "Python CDC Mode 115200 Baud "
'print device model
print 65
print " "
'print firmware version
```

Commands are sent to the ezLCD though the serial interface, Commands are text based and end with a carrage return **cr**.

So if you send **cls** ending with a **cr** the device will clear the screen and return a **cr** when the command is complete, some widgets take a bit of time (in the millsecond range) to complete so after sending a command allways wait for a **cr** to comeback before sending another command.

Introd	uction	To 1	ΓhΔ	Hard	war
IIIIII	ucuon	101		ııaıu	wait

# **Coordinates System**



The ezLCD uses a X Y coordinates system to specify the location for all graphics commands .

One thing to note is that the displays X Y start at 0, so even though you have a display that is 480x272 pixels wide XMAX is 479 and YMAX is 271.

X direction is horizontal across the display starting at the left 0 and ending at the max width of the display.

Y direction is vertical starting at the top 0 and ending at the bottom of the display.

XMAX and YMAX Values for the ezLCD 3xx Line

arLCD 319 239

ezLCD-301 399 239

ezLCD-302

ezLCD-303 319 239

ezLCD-313 319 239

ezLCD-304 479 271

Coord	linates S	vstem

6

# **Introduction To The Software**

Minimal example will open the ezLCD port clear the screen and print 'Hello From Python' in red



```
1 # Minimal ezLCD Python demo
4 import platform
8 sys.path.append("C:\Users\codeman\Documents\GitHub\ezLCD3xxPython\module")
9 from ezLCD3xx import *
11 #check what OS we are on
12 #Windows
13 if platform.system() == 'Windows':
14     LCD = ezLCD('com6')
15 #Mac
16 elif platform.system() == 'Dawrwin':
17 LCD = ezLCD('/dev/tty.usbsomething')
18 # Bail out if comport error
19 if LCD.openSerial() == False:
20    print 'Error Opening Port'
21    raise SystemExit
23 # Turn verbose off
24 LCD.verbose('off')
25 # Turn off button press info from ezLCD
26 LCD.wquiet(ON)
27 # CLear screen
28 LCD.cls()
29 # Set draw color to red
```

```
30 LCD.color(RED)
31 # Print string at coordinates x=80 and y=100
32 LCD.printString("Hello From Python",80,100)
```

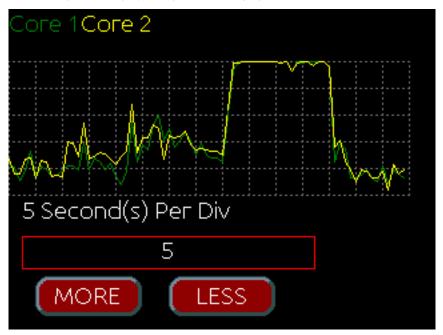
Button example will display a button widget then poll for button presses and update screen



```
1 # Button ezLCD Python demo
4 import platform
7 sys.path.append("C:\Users\segler\Documents\GitHub\ezLCD3xxPython\module")
8 from ezLCD3xx import *
10 #check what OS we are on
11 #Windows
12 if platform.system() == 'Windows':
       LCD = ezLCD('com4')
13
14 #Mac
15 elif platform.system() == 'Dawrwin':
       LCD = ezLCD('/dev/tty.usbsomething')
17 # Bail out if comport error
18 if LCD.openSerial() == False:
     print 'Error Opening Port'
19
       raise SystemExit
20
22 # Turn verbose off
23 LCD.verbose('off')
24 \# Turn off button press info from ezLCD
25 LCD.wquiet(ON)
26 # CLear screen
27 LCD.cls()
28 # Set draw color to red
29 LCD.color(RED)
30 # Set widget font 0
31 LCD.fontw(0,'1')
32 # Set wodget font 1
33 LCD.fontw(1,'0')
34 # Set theme #1
35 LCD.theme(1, 155, 152, 3, 0, 3, 24, 4, 5, 0, 1)
36 # Print string at coordinates x=80 and y=100
37 LCD.printString("Hello From Python", 80,100)
38 # Draw button widget with a ID of 1
39 LCD.button( 1, 80, 150, 155, 50, 1, 0, 10, 6, 3, 'Press Here')
40 # Draw a staticText box
41 LCD.staticText(2, 35, 30, 250, 30, 8, 1, 1, 'Press Button')
42 # Clear widget stack
43 LCD.wstack(CLEAR)
44
45 while True:
     # check widget stack this will return widget updates (button press ect.) last in first out order
46
       (ID, Info, Data) = LCD.wstack(LIFO)
```

```
# check if ID = 1 widget 1 and info = pressed
        if ID == 1 and Info == 4:
50
             # clear the stack just to be safe
             LCD.wstack(CLEAR)
51
52
             # change draw color to yellow
             LCD.color(YELLOW)
53
             \# change change string 1 for text on static text ID 2
55
             LCD.string(1,'Button Pressed')
56
             # redraw static text box ID 2 3=redraw
        LCD.wstate(2, 3)
# check if ID = 1 widget 1 and info = pressed and released
if ID == 1 and Info == 1:
57
58
59
             # clear the stack just to be safe
60
            LCD.wstack(CLEAR)
             # change draw color to yellow
             LCD.color(YELLOW)
             # change change string 1 for text on static text ID 2
LCD.string(1,'Button Pressed and Released')
# redraw static text box ID 2 3=redraw
64
65
66
             LCD.wstate(2, 3)
69
```

#### Load example will display the cpu load as a graph



```
1 #!/usr/bin/env python
2 # Python Serial library for ezLCD3xx
3 # http://www.ezlcd.com/
5 # You need the pySerial Library by Chris Liechti
6 # http://pyserial.wiki.sourceforge.net/pySerial
10 # END SerLCD Class Definition -----
12 # Start Test Program -----
13 import commands
14 import os
15 import re
16 import time as timer
17 import sys
18 import platform
19 import time
20 import psutil
21
22 sys.path.append("C:\Users\codeman\Documents\GitHub\ezLCD3xxPython\module")
23 from ezLCD3xx import *
25 def drawGrid():
2.6
       LCD.lineType(2)
27
       LCD.xv(0,30)
28
       LCD.color(BLACK)
       LCD.box(300,110,1)
```

```
30
        LCD.xy(0,0)
        LCD.color(GREEN)
31
        LCD.printString('Core 1')
32
33
        LCD.color(YELLOW)
        LCD.printString('Core 2')
34
        LCD.color(155)
35
        LCD.color(LIME)
36
37
        LCD.font('1')
38
        LCD.font('0')
39
        LCD.color(151)
       for y in range(6):
    LCD.xy(0,(y*20)+39)
40
41
            LCD.line(300, (y*20)+39)
42
43
        for x in range (16):
44
         LCD.xy(x*20,39)
45
             LCD.line(x*20,139)
        LCD.xy(300,39)
46
        LCD.line(300,139)
47
48
        LCD.lineType(0)
49
50 def drawTime(res):
51
        LCD.xy(10,140)
52
        LCD.color(BLACK)
        LCD.box(300.30, FILLED)
5.3
        LCD.color(WHITE)
54
        Time=str(res)+' Second(s) Per Div'
55
56
        LCD.printString(Time)
57
        LCD.string(5, str(res))
LCD.wstate(7,REDRAW)
58
59
60
61
62 if platform.system() == 'Windows':
63
        LCD = ezLCD('com6')
print 'Error Opening Port'
        raise SystemExit
69
70 LCD.verbose('OFF')
71 LCD.wquiet (ON)
72 LCD.cls()
73 LCD.fontw(0,'1')
74 LCD.fontw(1,'0')
75 LCD.fontw(2,'serif24')
76 LCD.theme(1, 155, 152, 3, 0, 3, 24, 4, 5, 0, 1)
77 LCD.backlight(100, 5, 10)
78 LCD.cls()
79 LCD.font('0')
80 LCD.fonto(0)
81 info = ' '
82 LCD.string( 1, '%')
83 LCD.color(WHITE)
84 LCD.cfgio(8,'analog')
85 print LCD.xmax()
86 print LCD.ymax()
87 LCD.xy(100,100)
88 (x,y) = LCD.xy()
89 print int(x), int(y)
90 (r,q,b) = LCD.colorId(3)
91 print r,g,b
92 print LCD.string(65)
93 print LCD.string(66)
94 print LCD.color()
95 print LCD.io(8)
96
98 LCD.button( 5, 20, 200, 80, 30 , 1, 0, 10, 1, 2, 'MORE')
99 LCD.button( 6, 120, 200, 80, 30 , 1, 0, 10, 1, 3, 'LESS')
100 LCD.staticText(7, 10, 170, 220, 25, 8, 1, 5, 'test')
101 drawGrid()
102 x=0
103 y1=239
104 y2=239
105 1x=0
106 ly1=239
107 ly2=239
108 res=5
109 drawTime(res)
110 LCD.wstack(CLEAR)
111 while True:
112
113
         oldinfo = info
114
         cores=psutil.cpu_percent(interval=1, percpu=True)
         y1 = 139 - cores[0]

y2 = 139 - cores[1]
115
116
```

```
117
           if x!=0:
             LCD.color(GREEN)
118
                 LCD.xy(lx,ly1)
119
                LCD.line(x, y1)
LCD.color(YELLOW)
LCD.xy(lx,ly2)
LCD.line(x, y2)
120
121
122
123
           ly1 = y1
ly2 = y2
lx = x
x += 20/res
124
125
126
127
128
           if x >= 300:
x=0
129
130
                 y1=239
131
                 y2=239
1x =0
1y1 =239
1y2 =239
132
133
134
135
136
                 drawGrid()
           (ID, info, data) = LCD.wstack(LIFO)
LCD.wstack(CLEAR)
137
138
           if ID == 5 and info==1:
    res +=1
139
140
141
                 drawTime(res)
142
           if ID == 6 and info==1:
    if res > 1:
        res -=1
143
144
145 drawT.
146 LCD.closeSerial()
                      drawTime(res)
147 # End Test Program ---
```

Introdu	iction	To The	Software
IIIIIOUL	<b>JULIOII</b>	TO THE	SULWALE

# **Module Index**

## 5.1 Modules

Hara is a list of all modules	
	_

Commands	??
Primitve Drawing Commands	??
Widgets	??
Bitmaps and Fonts	??

14 **Module Index** 

# Namespace Index

6.1	Namespace List	
Here	is a list of all documented namespaces with brief descriptions:	
0.7	ol CD3vv	2

16 Namespace Index

# **Hierarchical Index**

7.1 Class Hierarch
--------------------

This inheritance list is sort	ted roughly	but r	ot co	omp	lete	ly, a	llph	abe	tica	ılly:							
object																	
ezLCD3xx.ezLCD											 	 					 ??

18 **Hierarchical Index** 

# **Class Index**

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

20 Class Index

# **Module Documentation**

#### 9.1 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.

• def ezLCD3xx.calibrate

The calibrate command will re calibrate the touch screen.

### 9.1.1 Detailed Description

#### 9.1.2 Function Documentation

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

The backlight command will set backlight brightness and timeout.

22 Module Documentation

#### **Parameters**

brightness	1
timeout	2
level	3

#### 9.1.2.2 def ezLCD3xx.cfgio ( self, pin, function )

The cfgio command will configure io pins.

#### **Parameters**

pin	
function	

### 9.1.2.3 def ezLCD3xx.io ( self, pin, level = None )

The io command use to set and clear io pins.

#### **Parameters**

pin	
level	

#### Returns

io level

### 9.1.2.4 def ezLCD3xx.ping ( self )

the ping command

Returns

0

#### 9.1.2.5 def ezLCD3xx.play ( self, filename )

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

#### **Parameters**

filename	macro filename

#### 9.1.2.6 def ezLCD3xx.reset ( self )

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

### 9.1.2.7 def ezLCD3xx.run ( self, filename )

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

9.1 Commands 23

#### **Parameters**

filename	macro filename
----------	----------------

#### 9.1.2.8 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**

X	starting x position
у	starting y position
W	width
h	height
filename	filename.bmp Make sure you have space on the internal flash drive!

### 9.1.2.9 def ezLCD3xx.verbose ( self, state )

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

#### **Parameters**

state	0=off 1=on

### 9.1.2.10 def ezLCD3xx.wquiet ( self, state )

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

#### **Parameters**

state	0=off 1=on

#### 9.1.2.11 def ezLCD3xx.xmax ( self )

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

#### Returns

x-horizontal resolution in pixels starting from 0

## 9.1.2.12 def ezLCD3xx.ymax ( self )

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

#### Returns

y-vertical resolution in pixels starting from 0

24 Module Documentation

## 9.2 Primitve Drawing Commands

#### **Functions**

· def ezLCD3xx.cls

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

• def ezLCD3xx.color

The color command see ezLCD3xx manual for colors.

· def ezLCD3xx.colorld

The colorld 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 line Type 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.

#### 9.2.1 Detailed Description

#### 9.2.2 Function Documentation

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

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

#### **Parameters**

	radius	radius of arc
	start	start angle
	end	end angle
Ī	fill	1=filled arc 0=outline only *optional defaults to outline

9.2.2.2 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**

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

#### 9.2.2.3 def ezLCD3xx.circle ( self, radius, fill = 0 )

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

#### **Parameters**

radius	radius of circle
fill	1=filled circle 0=outline only *optional defaults to outline

#### 9.2.2.4 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**

left	
top	
right	
bottom	

#### 9.2.2.5 def ezLCD3xx.clipEnable ( self, enable )

The clipenable command enables or disables cliparea.

#### **Parameters**

enable	0=off 1=on

#### 9.2.2.6 def ezLCD3xx.cls ( self, Color = None )

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

#### **Parameters**

Color	color to clear screen to

### 9.2.2.7 def ezLCD3xx.color ( self, color = None )

The color command see ezLCD3xx manual for colors.

#### **Parameters**

color	number

#### Returns

color as a tuple

26 Module Documentation

9.2.2.8 def ezLCD3xx.colorld ( self, ID, R = None, G = None, B = None )

The colorld command.

#### **Parameters**

ID	color ID number
R	Red Value
G	Green Value
В	Blue Value

#### Returns

color as a tuple if r g b is None

### 9.2.2.9 def ezLCD3xx.line ( self, x, y )

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

#### **Parameters**

X	
У	

#### 9.2.2.10 def ezLCD3xx.lineType ( self, option )

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

#### **Parameters**

option	0 = solid, 1= dotted (1 pixel spacing between dots), 2 = dashed (2 pixel spacing between
	dashes)

### 9.2.2.11 def ezLCD3xx.lineWidth ( self, width )

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

#### **Parameters**

width	thin line (width = 1) or a thick line (width =3). Only [width] = 1 or 3 are available.

### 9.2.2.12 def ezLCD3xx.pie ( self, radius, start, end )

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

## **Parameters**

radius	radius of pie
start	start angle
end	end angle

## 9.2.2.13 def ezLCD3xx.plot ( self, x = None, y = None)

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

28 Module Documentation

#### **Parameters**

X	optional
у	optional

## 9.2.2.14 def ezLCD3xx.xy ( self, x = None, y = None)

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

#### **Parameters**

X	x position
у	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()
```

9.3 Widgets 29

## 9.3 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.

### 9.3.1 Detailed Description

#### 9.3.2 Function Documentation

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

The ameter widget.

30 Module Documentation

#### **Parameters**

ID	
X	
у	
width	
height	
options	
value	
minV	
maxV	
theme	
stringID meterType	
meterType	

9.3.2.2 def ezLCD3xx.ameter\_color ( self, ID, color1, color2, color3, color4, color5, color6 )

The ameter\_color command.

#### **Parameters**

ID	
color1	
color2	
color3	
color4	
color5	
color6	

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

The button command.

#### **Parameters**

ID	
X	
у	
width	
height	
options	
align	
radius	
theme	
stringID	
text	optional text for button

9.3.2.4 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** 

Generated on Sat Jul 20 2013 13:53:46 for My Project by Doxygen

9.3 Widgets 31

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

#### Returns

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

 $9.3.2.5 \quad \text{def ezLCD3xx.dial (} \quad \textit{self, ID, x, y, radius, option, resolution, value, } \quad \textit{maxx, theme )} \\$ 

The dial command.

#### **Parameters**

ID	
X	
У	
radius	
option	
resolution	
value	
maxx	
theme	

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

The dmeter widget.

#### **Parameters**

ID	
X	
y	
width	
height	
options	
value	
digits	
dp	
theme	

9.3.2.7 def ezLCD3xx.fontw ( self, fontnumber, name )

The fontW command will set the font for widget.

#### **Parameters**

fontnumber	number of the font
------------	--------------------

name	filename of font
	'0' and '1' are internal fonts

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

The groupBox widget.

#### **Parameters**

ID	
X	
у	
width	
height	
options	
theme	
stringID	

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

The progressBar command.

#### **Parameters**

ID	
X	
у	
width	
height	
options	
value	
mmax	
theme	
stringID	

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

The radioButton widget.

# Parameters

ID	
X	
у	
width	
height	
options	Options: 1=draw, 2=disabled, 3=checked, 4=first, 5=first and checked.
theme	
stringID	

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

The slider command.

9.3 Widgets 33

#### **Parameters**

ID	
X	
У	
width	
height	
options	
rrange	
resolution	
value	
theme	

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

The staticText widget.

## **Parameters**

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

9.3.2.13 def ezLCD3xx.string ( self, stringNumber, string = None )

The string command will set or return a internal string.

#### **Parameters**

stringNumber	number of string to set or return
string	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

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

The theme command sets the colors for widgets.

Parameters
------------

Generated on Sat Jul 20 2013 13:53:46 for My Project by Doxygen

ID	Theme ID
EmbossDkColor	Dark color for 3d effect
EmbossLtColor	Light color for 3d effect
TextColor0	
TextColor1	
TextColor-	
Disabled	
Color0	
Color1	
ColorDisabled	
CommonBk-	
Color	
Fontw	widget font for theme

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

The touchZone command.

#### **Parameters**

ID	
X	
у	
width	
height	
options	

9.3.2.16 def ezLCD3xx.wstack ( self, option )

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

#### **Parameters**

option	0=FIFO 1=LIFO 2=CLEAR
	FIFO Fist in Fist out
	LIFO Last in First out
	CLEAR Clear the stack

### Returns

truple 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 Vaules**

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

9.3 Widgets 35

# **Dial Widget Vaules**

- 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)
```

## 9.3.2.17 def ezLCD3xx.wstate ( self, ID, option )

The wstate command.

#### **Parameters**

ID	widget ID
option	0 = delete, 1 = enable, 2 = disable, 3 = redraw

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

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

#### **Parameters**

ID	
value	

# 9.4 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

#### 9.4.1 Detailed Description

#### 9.4.2 Function Documentation

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

The font command will set current font to use for printString fonts are located in the /EZSYS/FONTS and /EZUSE-B/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**

```
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')
```

#### 9.4.2.2 def ezLCD3xx.fonto ( self, orientation = None )

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

#### **Parameters**

```
orientation 0 90 180 270
```

#### Returns

orientation current orientation if orientation is not suppled

```
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)
```

# 9.4.2.3 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**

image	filename of image 'logo.gif'
X	x coordinates
У	y coordinates x y are optional and if not supplied will display image at current xy  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')

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

print string in current color and font and optional coordinates

#### **Parameters**

string	string to print
X	x coordinates
у	y coordinates
orientation	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 # diplay string 'Hello World' at 10 10 rotated 90 6 LCD.printString('Hello World',10,10,90)</pre>

# **Chapter 10**

# **Namespace Documentation**

# 10.1 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 calibrate

The calibrate command will re calibrate the touch screen.

• def cls

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

· def color

The color command see ezLCD3xx manual for colors.

· def colorId

The colorld 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 i 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** = 11int **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

## 10.1.1 Detailed Description

10.1.2.3 def ezLCD3xx.WaitForCR ( self )

This is a internal use function.

```
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/

10.1.2 Function Documentation

10.1.2.1 def ezLCD3xx.__init__( self, interface )

ezLCD object

10.1.2.2 def ezLCD3xx.closeSerial( self )

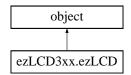
close
```

# **Chapter 11**

# **Class Documentation**

# 11.1 ezLCD3xx.ezLCD Class Reference

Inheritance diagram for ezLCD3xx.ezLCD:



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

 $\bullet \ \ C:/Users/codeman/Documents/GitHub/ezLCD3xxPython/module/ezLCD3xx.py$ 

# Index

init	wquiet, 25		
ezLCD3xx, 44	xmax, 25		
,	ymax, 26		
ameter	•		
Widgets, 31	dial		
ameter_color	Widgets, 33		
Widgets, 32	dmeter		
arc	Widgets, 33		
Primitve Drawing Commands, 27			
	ezLCD3xx, 41		
backlight	init, 44		
Commands, 24	closeSerial, 44		
Bitmaps and Fonts, 39	WaitForCR, 44		
font, 39	ezLCD3xx.ezLCD, 45		
fonto, 39	fami		
picture, 40	font		
printString, 40	Bitmaps and Fonts, 39		
box	fonto		
Primitve Drawing Commands, 28	Bitmaps and Fonts, 39		
button	fontw		
Widgets, 32	Widgets, 33		
otaio	groupBox		
cfgio	Widgets, 34		
Commands, 24	-		
choice	io		
Widgets, 32 circle	Commands, 24		
Primitve Drawing Commands, 28	line		
clipArea	Primitve Drawing Commands, 29		
Primitve Drawing Commands, 28	lineType		
clipEnable	Primitve Drawing Commands, 29		
Primitve Drawing Commands, 28	lineWidth		
closeSerial	Primitve Drawing Commands, 29		
ezLCD3xx, 44	mint		
CIS  Drimitus Drawing Commands 39	picture		
Primitve Drawing Commands, 28	Bitmaps and Fonts, 40		
Color  Drimitus Drawing Commands 20	pie		
Primitve Drawing Commands, 29	Primitve Drawing Commands, 30		
colorid	ping		
Primitve Drawing Commands, 29	Commands, 24		
Commands, 23	play		
backlight, 24	Commands, 24		
cfgio, 24	plot		
io, 24	Primity Drawing Commands, 30		
ping, 24	Primitve Drawing Commands, 27		
play, 24	arc, 27		
reset, 25	box, 28		
run, 25	circle, 28		
snapshot, 25	clipArea, 28		
verbose, 25	clipEnable, 28		

INDEX 45

	cls, 28	wqui	et
	color, 29		Commands, 25
	colorld, 29	wsta	
	line, 29		Widgets, 36
	lineType, 29	wsta	
	lineWidth, 29		Widgets, 37
	pie, 30	wval	
	plot, 30		Widgets, 37
	xy, 30	xmax	×
print	String		Commands, 25
	Bitmaps and Fonts, 40	ху	Commands, 25
	ressBar	-	Primitve Drawing Commands, 30
	Widgets, 34		Trimitive Drawing Commands, 30
rodio	Dutton	ymax	x
	Button		Commands, 26
	Widgets, 34		,
rese			
	Commands, 25		
run			
	Commands, 25		
slide	•		
	Widgets, 35		
snap	shot		
	Commands, 25		
	cText		
	Widgets, 35		
strin			
	Widgets, 35		
ــــــــــــــــــــــــــــــــــــــ	_		
them			
	Widgets, 36		
	nZone		
	Widgets, 36		
رمیات			
verb			
	Commands, 25		
\Mait	ForCR		
vvaii	ezLCD3xx, 44		
\\/ida			
	gets, 31		
	ameter, 31		
	ameter_color, 32		
	button, 32		
	choice, 32		
	dial, 33		
	dmeter, 33		
	fontw, 33		
	groupBox, 34		
	progressBar, 34		
	radioButton, 34		
	slider, 35		
	staticText, 35		
	string, 35		
	theme, 36		
	touchZone, 36		
	wstack, 36		
	wstate, 37		
	wvalue, 37		
	<del></del>		