LCD Touch Lib v2

Generated by Doxygen 1.8.9.1

Tue Oct 27 2015 18:39:14

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 _background Union Reference

Public Attributes

- uint16_t **color**
- uint8_t * **image**

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.2 _event_queue Struct Reference

Public Attributes

- uint8_t **widgetID** [MAX_EVENTS_ENQUEUED]
- uint8_t **counter**
- uint8_t **read_index**
- uint8_t **write_index**

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.3 _label Struct Reference

Public Attributes

- uint16_t **text_color**
- sFONT * **font**
- char **text** [MAX_LABEL_LEN]

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.4 _textline Struct Reference

Public Attributes

- uint16_t **background_color**
- uint16_t **border_color**
- uint16_t **text_color**
- uint16_t **textline_len**
- sFONT * **font**
- char * **textline**

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.5 _touch_event Struct Reference

Public Attributes

- uint16_t x
- uint16_t y

3.5.1 Member Data Documentation

3.5.1.1 uint16_t _touch_event::x

X component

3.5.1.2 uint16_t _touch_event::y

Y component

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.6 _wdesc Struct Reference

Public Attributes

- WIDGET * **data**
- struct _wdesc * **child**
- struct _wdesc * **sibling**

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.7 _widget Struct Reference

Public Attributes

- WIDGET_TYPE wtype
- TOUCH_AREA bounds
- uint8_t id
- void * wdata
- HANDLER handler
- void * handler_arg

3.7.1 Member Data Documentation

3.7.1.1 TOUCH_AREA _widget::bounds

Widget bounds (top left and bottom right corner points)

3.7.1.2 HANDLER _widget::handler

Handler function of the widget, if any

3.7.1.3 void* _widget::handler_arg

Handler function argument, if any

3.7.1.4 uint8_t _widget::id

Global widget ID

3.7.1.5 void* _widget::wdata

Specific Widget Data

3.7.1.6 WIDGET_TYPE _widget::wtype

Widget Type

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.8 lcd_page Struct Reference

Public Attributes

- PAGE_BACKGROUND **backgroundObj**
- BACKGROUND_TYPE **background_type**

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.9 lcd_page_background Struct Reference

Public Attributes

- BACKGROUND **background**
- BACKGROUND_TYPE **background_type**

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.10 point Struct Reference

Public Attributes

- uint16_t x
- uint16_t y

3.10.1 Member Data Documentation

3.10.1.1 uint16_t point::x

X component

3.10.1.2 uint16_t point::y

Y component

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.11 touch_area Struct Reference

Public Attributes

- POINT tl_corner
- POINT br_corner

3.11.1 Member Data Documentation

3.11.1.1 POINT touch_area::br_corner

Bottom right corner coordinates of the button area

3.11.1.2 POINT touch_area::tl_corner

Top left corner coordinates of the button area

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

3.12 touch_button Struct Reference

Public Attributes

- uint8_t ∗ icon

3.12.1 Member Data Documentation

3.12.1.1 uint8_t∗ touch_button::icon

Null if the icon is included in the background

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

- C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_↩ Types.h

Chapter 4

File Documentation

4.1 C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_↩ Touch_Lib_v2/LCD_Colors.h File Reference

LCD_Lib_Touch color defines.

Macros

- #define **LCD_LIB_WHITE** LCD_COLOR_WHITE
- #define **LCD_LIB_BLACK** LCD_COLOR_BLACK
- #define **LCD_LIB_GREY** LCD_COLOR_GREY
- #define **LCD_LIB_BLUE** LCD_COLOR_BLUE
- #define **LCD_LIB_BLUE2** LCD_COLOR_BLUE2
- #define **LCD_LIB_RED** LCD_COLOR_RED
- #define **LCD_LIB_MAGENTA** LCD_COLOR_MAGENTA
- #define **LCD_LIB_GREEN** LCD_COLOR_GREEN
- #define **LCD_LIB_CYAN** LCD_COLOR_CYAN
- #define **LCD_LIB_YELLOW** LCD_COLOR_YELLOW

4.1.1 Detailed Description

LCD_Lib_Touch color defines.

Author

Daniel Casini

Date

17/10/2015

4.2 C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_↩ Touch_Lib_v2/LCD_Lib_Conf.h File Reference

LCD_Touch_Lib configuration file.

```
#include "STMPE811QTR.h"
#include "stm32f4_discovery_lcd.h"
```

Macros

- #define NUM_PAGES 3

  Maximum number of LCD Pages.
- #define NUM_BUTTONS 16 * NUM_PAGES

  Maximum number of LCD Buttons.
- #define NUM_LABELS 16 * NUM_PAGES

  Maximum number of LCD Labels.
- #define NUM_TEXTLINE 16 * NUM_PAGES

  Maximum number of LCD Text Lines.
- #define NUMWLIST NUM_TEXTLINE + NUM_LABELS + NUM_BUTTONS

  Maximum number of widgets.
- #define MAX_EVENTS_ENQUEUED 10

  Maximum number of touch events in the queue.
- #define MAX_LABEL_LEN 20

  Maximum length for a label content.

4.2.1 Detailed Description

LCD_Touch_Lib configuration file.

Author

Daniel Casini

Date

17/10/2015

4.3 C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_Draw.h File Reference

LCD_Lib_Touch widget draw functions.

```
#include "LCD_Lib_Types.h"
```

Functions

- void drawSingleWidget (WIDGET_DESC *widget)

  calls a specific function to draw a widget, according to the widget type
- void drawLCDPage (WIDGET_DESC *widget)

  draws an LCD page
- void drawLabel (WIDGET_DESC *widget)

  draws a label widget
- void drawButton (WIDGET_DESC *widget)

  draws a button widget
- uint8_t drawTextLine (WIDGET_DESC *widget)

  draw a textline widget

4.3.1 Detailed Description

LCD_Lib_Touch widget draw functions.

Author

Daniel Casini

Date

17/10/2015

4.3.2 Function Documentation

4.3.2.1 uint8_t drawTextLine (WIDGET_DESC * *widget*)

draw a textline widget

Returns

-1 if the the text is too long to be contained in the textbox, 1 otherwise

4.4 C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_↩ Touch_Lib_v2/LCD_Lib_HWParam.h File Reference

HW parameters for the LCD_Touch_Lib.

Macros

- #define TOUCH_AD_VALUE_MAX (4000)

  Analog/Digital converter max value.
- #define TOUCH_AD_VALUE_MIN (100)

  Analog/Digital converter min value.
- #define LCD_HEIGHT LCD_PIXEL_HEIGHT

  Screen height.
- #define LCD_WIDTH LCD_PIXEL_WIDTH

  Screen width.
- #define X_SCALE -0.092348285

  Parameter for the conversion between the AD value to the LCD value (result of previous calibration)
- #define Y_SCALE 0.0662910193

  Parameter for the conversion between the AD value to the LCD value (result of previous calibration)
- #define X_OFFSET -353

  Parameter for the conversion between the AD value to the LCD value (result of previous calibration)
- #define Y_OFFSET 24

  Parameter for the conversion between the AD value to the LCD value (result of previous calibration)
- #define ERR 255

  Error constant.

4.4.1 Detailed Description

HW parameters for the LCD_Touch_Lib.

Author

Daniel Casini

Date

17/10/2015

4.5 C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_↩ Touch_Lib_v2/LCD_Lib_Layers.h File Reference

Layered structure for LCD GUI abstractions implemented using a tree.

```
#include "LCD_Lib_Draw.h"
```

Functions

- uint8_t contains (TOUCH_AREA *area, POINT *point)

  Checks whether a point is contained in a touch area.
- void drawWidgets (WIDGET_DESC *desc)

  Draws the given widget and all the widgets in its sub-tree.
- void drawWidgetSiblings (WIDGET_DESC *widget)

  Draws the given widget and all the siblings widgets.
- void insertWidget (WIDGET *widget, WIDGET *father)

  Inserts a widget in the tree: if father is specified, widget becomes a child of father, otherwise the father widget is searched with the findContainerWidget() method.
- void insertChild (WIDGET_DESC *father, WIDGET_DESC *newChild)

  Inserts the widget newChild as child of father.
- WIDGET_DESC * findPressedWidget (POINT *detected)

  Finds the pressed widget from a point, i.e. finds the deepest widget in the widget layer tree that has an handler.
- POINT getWidgetCenter (WIDGET *widget)
- WIDGET_DESC * findContainerWidget (WIDGET *widget)

  Returns the 'father' widget of the given widget, i.e., a widget with a bounding box that entirely contains the one of the given widget.

4.5.1 Detailed Description

Layered structure for LCD GUI abstractions implemented using a tree.

Author

Daniel Casini

Date

17/10/2015

4.5.2 Function Documentation

4.5.2.1 uint8_t contains (TOUCH_AREA * *area*, POINT * *point*)

Checks whether a point is contained in a touch area.

Parameters

area	Area of the screen
point	Point to check

Returns

1 if point is contained in area, 0 otherwise

4.5.2.2 void drawWidgets (WIDGET_DESC * *desc*)

Draws the given widget and all the widgets in its sub-tree.

Parameters

desc	widget descriptor

4.5.2.3 void drawWidgetSiblings (WIDGET_DESC * *widget*)

Draws the given widget and all the siblings widgets.

Parameters

widget	widget descriptor

4.5.2.4 WIDGET_DESC* findPressedWidget (POINT * *detected*)

Finds the pressed widget from a point, i.e. finds the deepest widget in the widget layer tree that has an handler.

Parameters

detected	detected point in the touchscreen

4.5.2.5 POINT getWidgetCenter (WIDGET * *widget*)

Parameters

widget	widget object

Returns

widget bounding box center

4.5.2.6 void insertChild (WIDGET_DESC * *father,* WIDGET_DESC * *newChild*)

Inserts the widget newChild as child of father.

Parameters

father	father widget
newChild	newChild widget

4.5.2.7 void insertWidget (WIDGET * *widget*, WIDGET * *father*)

Inserts a widget in the tree: if father is specified, widget becomes a child of father, otherwise the father widget is searched with the findContainerWidget() method.

Parameters

widget	widget to insert in the layer tree
father	father widget

4.6 C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Lib_Types.h File Reference

Types for LCD_Touch_Lib.

```
#include "LCD_Lib_HWParam.h"
#include "LCD_Lib_Conf.h"
```

Classes

- struct point
- struct _touch_event
- struct touch_area
- struct _event_queue
- union _background
- struct lcd_page_background
- struct _widget
- struct _wdesc
- struct lcd_page
- struct touch_button
- struct _textline
- struct _label

Macros

- #define POINTS_TO_PIXEL 1.3

  Converts points to pixels (for fonts)

Typedefs

- typedef void(∗ HANDLER) (void ∗arg)
- typedef struct point **POINT**
- typedef struct _touch_event **TOUCH_EVENT**
- typedef struct touch_area **TOUCH_AREA**
- typedef struct _event_queue **EVENT_QUEUE_T**
- typedef enum _widgetType **WIDGET_TYPE**
- typedef union _background **BACKGROUND**
- typedef enum _backgroundType **BACKGROUND_TYPE**
- typedef struct lcd_page_background **PAGE_BACKGROUND**
- typedef struct _widget **WIDGET**
- typedef struct _wdesc **WIDGET_DESC**
- typedef struct lcd_page **LCD_PAGE**
- typedef struct touch_button **BUTTON**
- typedef struct _textline **TEXT_LINE**
- typedef struct _label **LABEL**

Enumerations

- enum **_widgetType** { **LCD_PAGE_TYPE**, **BUTTON_TYPE**, **LABEL_TYPE**, **TEXTLINE_TYPE**, **NUM_WIDGET_TYPE** }
- enum **_backgroundType** { **BACKGROUND_COLOR**, **BACKGROUND_IMAGE**, **NUM_BACKGROUND_TYPES** }

4.6.1 Detailed Description

Types for LCD_Touch_Lib.

Author

Daniel Casini

Date

17/10/2015

4.6.2 Typedef Documentation

4.6.2.1 typedef void(* HANDLER) (void *arg)

Object handler type

4.7 C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_Touch_Lib_v2/LCD_Touch_Lib.c File Reference

LCD and Touchscreen library for STM32F4Discovery with SSD2119 (LCD) and STMPE811QTR (Touch)

```
#include "LCD_Touch_Lib.h"
#include "stm32f4_discovery_lcd.h"
#include "string.h"
#include "stdio.h"
```

Functions

- void initLCD_Touch ()

  Touch event queue.
- WIDGET_DESC * newButton (WIDGET *btn, uint16_t x_tl, uint16_t y_tl, uint16_t width, uint16_t height, uint8_t *icon, HANDLER handler, void *handler_arg, WIDGET *father)

  Creates a new button.
- WIDGET_DESC * newPage (WIDGET *page, uint16_t color)

  Creates a new page.
- WIDGET_DESC * newTextLine (WIDGET *textLine, uint16_t x_tl, uint16_t y_tl, uint16_t width, uint16_t height, uint16_t textColor, uint16_t borderColor, uint16_t backgroundColor, sFONT *font, char *input_port, uint16_t input_port_size, WIDGET *father)

  Creates a new textline.
- WIDGET_DESC * newLabel (WIDGET *label, uint16_t x, uint16_t y, uint16_t textColor, sFONT *font, char *text, WIDGET *father)

  Creates a new label.

- void enqueueTouchEvents (POINT detected)

  Enqueue a touch event, after having been detected the pressed widget.
- void dispatchTouchEvents ()

  Dispatches a touch event taken from the queue, if any.
- void pollingTouch (POINT *pnt)

  Waits for a pression on the touchscreen and return the point in which is pressed (blocking)
- void drawWidget (WIDGET *widget)

  Draws the given widget and all the widgets in its sub-tree.
- void getTouch (POINT *pnt)

  Waits for a pression on the touchscreen and return the point in which is pressed (not blocking)
- uint8_t textLine_setText (WIDGET *textLine, char newText[ ])

  Set the text of a given textline.
- uint8_t textLine_setAndDisplayText (WIDGET *textLine, char newText[ ])

  Set the text of a given textline and updates the screen.
- uint8_t textLine_addText (WIDGET *textLine, char newText[ ])

  Concatenates the given text with the one already contained in the given textline.
- uint8_t textLine_removeLastChar (WIDGET *textLine)

  Remove the last character in the textline.
- uint8_t textLine_removeLastCharAndDisplayText (WIDGET *textLine)

  Remove the last character in the textline and updates the screen.
- uint8_t textLine_addAndDisplayText (WIDGET *textLine, char newText[ ])

  Concatenates the given text with the one already contained in the given textline and updates the screen.
- void textLine_eraseText (WIDGET *textLine)

  Erases the textline widget text.
- void textLine_eraseAndDisplayText (WIDGET *textLine)

  Erases the textline widget text and updates the screen.
- void drawCurrentPage ()

  Draws the current page.
- uint8_t changePage (uint8_t page_index)

  Changes and draw the page with index page_index.
- uint8_t setPage (uint8_t page_index)

  Sets the page with index equal to pageIndex as activePage.

Variables

- uint8_t **activePage**
- uint8_t numPages

  Index of the currently active page.
- uint8_t numWidgets

  Current number of LCD pages.
- uint8_t numButtons

  Current number of Widgets.
- uint8_t numLabels

  Current number of Buttons.
- uint8_t numTextLines

  Current number of Labels.
- WIDGET_DESC * LCD_pages [NUM_PAGES]

  Current number of TextLines.
- WIDGET_DESC widgetList [NUMWLIST]

  LCD Pages pointer.

- LCD_PAGE pageData [NUM_PAGES]

  Widget List variables.
- BUTTON buttonData [NUM_BUTTONS]

  LCD Pages variables.
- LABEL labelData [NUM_LABELS]

  Button Data variables.
- TEXT_LINE textLineData [NUM_TEXTLINE]

  Label Data variables.
- EVENT_QUEUE_T eventQueue

  Textline Data variables.

4.7.1 Detailed Description

LCD and Touchscreen library for STM32F4Discovery with SSD2119 (LCD) and STMPE811QTR (Touch)

Author

Daniel Casini

Date

05/03/2014

4.7.2 Function Documentation

4.7.2.1 void changePage (uint8_t *page_index*)

Changes and draw the page with index page_index.

Returns

-1 if the given page does not exists

4.7.2.2 void drawWidget (WIDGET * *widget*)

Draws the given widget and all the widgets in its sub-tree.

Parameters

widget	widget object

4.7.2.3 void enqueueTouchEvents (POINT *detected*)

Enqueue a touch event, after having been detected the pressed widget.

Parameters

detected	point returned by the touchscreen

4.7.2.4 void getTouch (POINT * *pnt*)

Waits for a pression on the touchscreen and return the point in which is pressed (not blocking)

Parameters

area	Area of the screen
point	Point to check

4.7.2.5 void initLCD_Touch ( )

Touch event queue.

Initializes the LCD display and the touchscreen.

4.7.2.6 WIDGET_DESC * newButton (WIDGET * *btn*, uint16_t *x_tl*, uint16_t *y_tl*, uint16_t *width*, uint16_t *height*, uint8_t * *icon*, HANDLER *handler*, void * *handler_arg*, WIDGET * *father*)

Creates a new button.

Parameters

btn	Button to create
x_tl	Top right corner x coordinate
y_tl	Top right corner y coordinate
height	Height
width	Width
icon	Icon of the widget, array of characters in BMP565 format: if null, the widget's area is associated to the page background
handler	Handler function of the widget
handler_arg	Handler function argument of the widget
father	Father widget (if not specified, the father is automatically detected according to the widgets coordinates)

Returns

Pointer to the widget descriptor

4.7.2.7 WIDGET_DESC * newLabel (WIDGET * *label*, uint16_t *x*, uint16_t *y*, uint16_t *textColor*, sFONT * *font*, char * *text*, WIDGET * *father*)

Creates a new label.

Parameters

label	Label to create
x	Top right corner x coordinate
y	Top right corner y coordinate
textColor	Textcolor, in 565 format (16 bit integer)
font	Text font
text	Text of the label
father	Father widget (if not specified, the father is automatically detected according to the widgets coordinates)

Returns

Pointer to the widget descriptor

4.7.2.8 WIDGET_DESC * newPage (WIDGET * *page*, uint16_t *color*)

Creates a new page.

Parameters

page	Page widget
color	background color

Returns

Pointer to the widget descriptor

4.7.2.9 WIDGET_DESC * newTextLine (WIDGET * *textLine,* uint16_t *x_tl,* uint16_t *y_tl,* uint16_t *width,* uint16_t *height,* uint16_t *textColor,* uint16_t *borderColor,* uint16_t *backgroundColor,* sFONT * *font,* char * *input_port,* uint16_t *input_port_size,* WIDGET * *father*)

Creates a new textline.

Parameters

label	Label to create
x_tl	Top right corner x coordinate
y_tl	Top right corner y coordinate
height	Height
width	Width
textColor	Textcolor, in 565 format (16 bit integer)
borderColor	Textcolor, in 565 format (16 bit integer)
backgroundColor	Textcolor, in 565 format (16 bit integer)
font	Text font
input_port	Pointer to an input port string
input_port_size	Maximum size of the input port string
father	Father widget (if not specified, the father is automatically detected according to the widgets coordinates)

Returns

Pointer to the widget descriptor

4.7.2.10 void pollingTouch (POINT * *pnt*)

Waits for a pression on the touchscreen and return the point in which is pressed (blocking)

Parameters

area	Area of the screen
point	Point to check

4.7.2.11 uint8_t setPage (uint8_t *pageIndex*)

Sets the page with index equal to pageIndex as activePage.

Returns

-1 if the page does not exists, 1 otherwise

4.7.2.12 uint8_t textLine_addAndDisplayText (WIDGET * *textLine,* char *newText[ ]*)

Concatenates the given text with the one already contained in the given textline and updates the screen.

Parameters

textLine	textline widget
newText	text to be concatenated

Returns

-1 if the text becomes to long for the textline, 1 otherwise

4.7.2.13 uint8_t textLine_addText (WIDGET * *textLine,* char *newText[ ]*)

Concatenates the given text with the one already contained in the given textline.

Parameters

textLine	textline widget
newText	text to be concatenated

Returns

-1 if the text becomes to long for the textline, 1 otherwise

4.7.2.14 void textLine_eraseAndDisplayText (WIDGET * *textLine*)

Erases the textline widget text and updates the screen.

Parameters

textLine	textline widget

4.7.2.15 void textLine_eraseText (WIDGET * *textLine*)

Erases the textline widget text.

Parameters

textLine	textline widget

4.7.2.16 uint8_t textLine_removeLastChar (WIDGET * *textLine*)

Remove the last character in the textline.

Parameters

textLine	textline widget

Returns

-1 if there is no character to remove, 1 otherwise

4.7.2.17 uint8_t textLine_removeLastCharAndDisplayText (WIDGET * *textLine*)

Remove the last character in the textline and updates the screen.

Parameters

textLine	textline widget

Returns

-1 if there is no character to remove, 1 otherwise

4.7.2.18 uint8_t textLine_setAndDisplayText (WIDGET * *textLine,* char *newText[ ]*)

Set the text of a given textline and updates the screen.

Parameters

textLine	textline widget
newText	text to be copied

Returns

-1 if the given text is too long for the textline, 1 otherwise

4.7.2.19 uint8_t textLine_setText (WIDGET * *textLine,* char *newText[ ]*)

Set the text of a given textline.

Parameters

textLine	textline widget
newText	text to be copied

Returns

-1 if the given text is too long for the textline, 1 otherwise

4.8 C:/Users/Daniel/Desktop/IndustrialGEN/IndustrialGUIGenerationProject/LCD_↩ Touch_Lib_v2/LCD_Touch_Lib.h File Reference

New LCD and Touchscreen library for STM32F4Discovery with SSD2119 (LCD) and STMPE811QTR (Touch) (v2) Implements screen layers, touch management. The currently supported widgets are pages, buttons, textlines, labels.

```
#include "STMPE811QTR.h"
#include "stm32f4_discovery_lcd.h"
#include "fonts.h"
#include "LCD_Lib_Layers.h"
#include "LCD_Colors.h"
```

Macros

- #define RGB_2_565(r, g, b) (((r&0xF8)<<8)|((g&0xFC)<<3)|((b&0xF8)>>3))

  RGB 888 to RGB 565 Conversion Macro.

Functions

- void initLCD_Touch ()

  Touch event queue.
- WIDGET_DESC * newPage (WIDGET *page, uint16_t color)

  Creates a new page.
- WIDGET_DESC * newButton (WIDGET *btn, uint16_t x_tl, uint16_t y_tl, uint16_t width, uint16_t height, uint8_t *icon, HANDLER handler, void *handler_arg, WIDGET *father)

  Creates a new button.
- WIDGET_DESC * newLabel (WIDGET *label, uint16_t x, uint16_t y, uint16_t textColor, sFONT *font, char *text, WIDGET *father)

  Creates a new label.
- WIDGET_DESC * newTextLine (WIDGET *textLine, uint16_t x_tl, uint16_t y_tl, uint16_t width, uint16_t height, uint16_t textColor, uint16_t borderColor, uint16_t backgroundColor, sFONT *font, char *input_port, uint16_t input_port_size, WIDGET *father)

  Creates a new textline.
- void pollingTouch (POINT *pnt)

  Waits for a pression on the touchscreen and return the point in which is pressed (blocking)
- void getTouch (POINT *pnt)

  Waits for a pression on the touchscreen and return the point in which is pressed (not blocking)
- void enqueueTouchEvents (POINT detected)

  Enqueue a touch event, after having been detected the pressed widget.
- void dispatchTouchEvents ()

  Dispatches a touch event taken from the queue, if any.
- void drawWidget (WIDGET *widget)

  Draws the given widget and all the widgets in its sub-tree.
- void drawCurrentPage ()

  Draws the current page.
- uint8_t setPage (uint8_t pageIndex)

  Sets the page with index equal to pageIndex as activePage.
- uint8_t changePage (uint8_t page_index)

  Changes and draw the page with index page_index.
- uint8_t textLine_setText (WIDGET *textLine, char newText[ ])

  Set the text of a given textline.
- uint8_t textLine_setAndDisplayText (WIDGET *textLine, char newText[ ])

  Set the text of a given textline and updates the screen.
- uint8_t textLine_addText (WIDGET *textLine, char newText[ ])

  Concatenates the given text with the one already contained in the given textline.
- uint8_t textLine_addAndDisplayText (WIDGET *textLine, char newText[ ])

  Concatenates the given text with the one already contained in the given textline and updates the screen.
- void textLine_eraseText (WIDGET *textLine)

  Erases the textline widget text.
- void textLine_eraseAndDisplayText (WIDGET *textLine)

  Erases the textline widget text and updates the screen.
- uint8_t textLine_removeLastChar (WIDGET *textLine)

  Remove the last character in the textline.
- uint8_t textLine_removeLastCharAndDisplayText (WIDGET *textLine)

  Remove the last character in the textline and updates the screen.

4.8.1 Detailed Description

New LCD and Touchscreen library for STM32F4Discovery with SSD2119 (LCD) and STMPE811QTR (Touch) (v2) Implements screen layers, touch management. The currently supported widgets are pages, buttons, textlines, labels.

Author

Daniel Casini

Date

17/10/2015

4.8.2 Function Documentation

4.8.2.1 uint8_t changePage (uint8_t *page_index*)

Changes and draw the page with index page_index.

Returns

-1 if the given page does not exists

4.8.2.2 void drawWidget (WIDGET * *widget*)

Draws the given widget and all the widgets in its sub-tree.

Parameters

widget	widget object

4.8.2.3 void enqueueTouchEvents (POINT *detected*)

Enqueue a touch event, after having been detected the pressed widget.

Parameters

detected	point returned by the touchscreen

4.8.2.4 void getTouch (POINT * *pnt*)

Waits for a pression on the touchscreen and return the point in which is pressed (not blocking)

Parameters

area	Area of the screen
point	Point to check

4.8.2.5 void initLCD_Touch ( )

Touch event queue.

Initializes the LCD display and the touchscreen.

4.8.2.6 WIDGET_DESC* newButton (WIDGET * *btn,* uint16_t *x_tl,* uint16_t *y_tl,* uint16_t *width,* uint16_t *height,* uint8_t * *icon,* HANDLER *handler,* void * *handler_arg,* WIDGET * *father*)

Creates a new button.

Parameters

btn	Button to create
x_tl	Top right corner x coordinate
y_tl	Top right corner y coordinate
height	Height
width	Width
icon	Icon of the widget, array of characters in BMP565 format: if null, the widget's area is associated to the page background
handler	Handler function of the widget
handler_arg	Handler function argument of the widget
father	Father widget (if not specified, the father is automatically detected according to the widgets coordinates)

Returns

Pointer to the widget descriptor

4.8.2.7 WIDGET_DESC* newLabel (WIDGET * *label,* uint16_t *x,* uint16_t *y,* uint16_t *textColor,* sFONT * *font,* char * *text,* WIDGET * *father*)

Creates a new label.

Parameters

label	Label to create
x	Top right corner x coordinate
y	Top right corner y coordinate
textColor	Textcolor, in 565 format (16 bit integer)
font	Text font
text	Text of the label
father	Father widget (if not specified, the father is automatically detected according to the widgets coordinates)

Returns

Pointer to the widget descriptor

4.8.2.8 WIDGET_DESC* newPage (WIDGET * *page,* uint16_t *color*)

Creates a new page.

Parameters

page	Page widget
color	background color

Returns

Pointer to the widget descriptor

4.8.2.9 WIDGET_DESC* newTextLine (WIDGET * *textLine,* uint16_t *x_tl,* uint16_t *y_tl,* uint16_t *width,* uint16_t *height,* uint16_t *textColor,* uint16_t *borderColor,* uint16_t *backgroundColor,* sFONT * *font,* char * *input_port,* uint16_t *input_port_size,* WIDGET * *father*)

Creates a new textline.

Parameters

label	Label to create
x_tl	Top right corner x coordinate
y_tl	Top right corner y coordinate
height	Height
width	Width
textColor	Textcolor, in 565 format (16 bit integer)
borderColor	Textcolor, in 565 format (16 bit integer)
backgroundColor	Textcolor, in 565 format (16 bit integer)
font	Text font
input_port	Pointer to an input port string
input_port_size	Maximum size of the input port string
father	Father widget (if not specified, the father is automatically detected according to the widgets coordinates)

Returns

Pointer to the widget descriptor

4.8.2.10 void pollingTouch (POINT * *pnt*)

Waits for a pression on the touchscreen and return the point in which is pressed (blocking)

Parameters

area	Area of the screen
point	Point to check

4.8.2.11 uint8_t setPage (uint8_t *page_index*)

Sets the page with index equal to pageIndex as activePage.

Returns

-1 if the page does not exists, 1 otherwise

4.8.2.12 uint8_t textLine_addAndDisplayText (WIDGET * *textLine,* char *newText[ ]*)

Concatenates the given text with the one already contained in the given textline and updates the screen.

Parameters

textLine	textline widget
newText	text to be concatenated

Returns

-1 if the text becomes to long for the textline, 1 otherwise

4.8.2.13 uint8_t textLine_addText (WIDGET * *textLine,* char *newText[ ]*)

Concatenates the given text with the one already contained in the given textline.

Parameters

textLine	textline widget
newText	text to be concatenated

Returns

-1 if the text becomes to long for the textline, 1 otherwise

4.8.2.14 void textLine_eraseAndDisplayText (WIDGET * *textLine*)

Erases the textline widget text and updates the screen.

Parameters

textLine	textline widget

4.8.2.15 void textLine_eraseText (WIDGET * *textLine*)

Erases the textline widget text.

Parameters

textLine	textline widget

4.8.2.16 uint8_t textLine_removeLastChar (WIDGET * *textLine*)

Remove the last character in the textline.

Parameters

textLine	textline widget

Returns

-1 if there is no character to remove, 1 otherwise

4.8.2.17 uint8_t textLine_removeLastCharAndDisplayText (WIDGET * *textLine*)

Remove the last character in the textline and updates the screen.

Parameters

textLine	textline widget

Returns

-1 if there is no character to remove, 1 otherwise

4.8.2.18 uint8_t textLine_setAndDisplayText (WIDGET * *textLine,* char *newText[ ]*)

Set the text of a given textline and updates the screen.

Parameters

textLine	textline widget
newText	text to be copied

Returns

-1 if the given text is too long for the textline, 1 otherwise

4.8.2.19 uint8_t textLine_setText (WIDGET * *textLine*, char *newText[ ]*)

Set the text of a given textline.

Parameters

textLine	textline widget
newText	text to be copied

Returns

-1 if the given text is too long for the textline, 1 otherwise

Index