GUIslice

0.11.2

Generated by Doxygen 1.8.11

Contents

Chapter 1

GUIslice library

A lightweight GUI framework for embedded displays

Design your GUI with a **drag & drop builder**, then apply the same code to a wide range of displays, libraries and controllers with the **cross-platform framework**. Open source **MIT license** grants free commercial usage.

- Extensive `Documentation` guides available
- `GUIslice API documentation (online)` & `(PDF)`
- Active development: see `latest updates & work in progress`
- `Release history`
- `Website (www.impulseadventure.com)`
- **Support email**: `guislice@gmail.com`

Features

- Pure C library, no dynamic memory allocation
- *Widgets*:
  - text, images, buttons, checkboxes, radio buttons, sliders, keypad, listbox, radial controls, scrolling textbox / terminal, graphs, etc. plus extensions and multiple pages.
- Cross-platform **GUIslice Builder** (beta) desktop application to generate layouts
- *Platform-independent* GUI core currently supports:
  - Adafruit-GFX, TFT_eSPI, SDL1.2, SDL2.0
- *Devices*:
  - Raspberry Pi, Arduino, ESP8266 / NodeMCU, ESP32, M5stack, Teensy 3, Feather M0 (Cortex-M0), nRF52 (Cortex-M4F), LINUX, Beaglebone Black, STM32
- *Typical displays*:
  - PiTFT, Adafruit TFT 3.5" / 2.8" / 2.4" / 2.2" / 1.44", FeatherWing TFT, OLED 0.96", mcufriend, BuyDisplay / EastRising 4.3" 5" 7", Waveshare, 4D Cape

- *Display drivers include*:
  - ILI9341, ST7735, SSD1306, HX8347D, HX8357, PCD8544, RA8875, ILI9341_t3
- *Touchscreen control including*:
  - STMPE610, FT6206, XPT2046, 4-wire, tslib, Adafruit Seesaw
- Foreign characters / UTF-8 encoding (in SDL mode), anti-aliased fonts (in TFT_eSPI mode)
- Dynamic display rotation
- GPIO / pin / keyboard / Adafruit Seesaw control for non-touchscreen devices

Screenshots

GUIslice Builder

- Includes cross-platform (Windows & LINUX) desktop application (beta) to generate GUIslice layouts
- Please refer to `GUIslice Builder wiki` for documentation

Chapter 2

Todo List

Global gslc_CollectFindFocusStep (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, bool bNext, bool ∗pb←Wrapped, int16_t ∗pnElemInd)

Doc. This API is experimental and subject to change

Global gslc_ElemDraw (gslc_tsGui ∗pGui, int16_t nPageId, int16_t nElemId)

Unused?

Global gslc_ElemXRingGaugeSetRange (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nPosMin, int16_t nPosMax)

Global gslc_InitInputMap (gslc_tsGui ∗pGui, gslc_tsInputMap ∗asInputMap, uint8_t nInputMapMax)

Doc. This API is experimental and subject to change

Global gslc_InputMapAdd (gslc_tsGui ∗pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc←_teAction eAction, int16_t nActionVal)

Doc. This API is experimental and subject to change

Global gslc_InputMapLookup (gslc_tsGui ∗pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc_teAction ∗peAction, int16_t ∗pnActionVal)

Doc. This API is experimental and subject to change

Global gslc_PageFocusStep (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, bool bNext)

Doc. This API is experimental and subject to change

Global gslc_SetPinPollFunc (gslc_tsGui ∗pGui, GSLC_CB_PIN_POLL pfunc)

Doc. This API is experimental and subject to change

Chapter 3

Module Index

3.1 Modules

Here is a list of all modules:

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Data Structure Index

5.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

Chapter 7

Module Documentation

7.1 General Functions

General functions for configuring the GUI.

Functions

- char ∗ gslc_GetVer (gslc_tsGui ∗pGui)

  Get the GUIslice version number.
- const char ∗ gslc_GetNameDisp (gslc_tsGui ∗pGui)

  Get the GUIslice display driver name.
- const char ∗ gslc_GetNameTouch (gslc_tsGui ∗pGui)

  Get the GUIslice touch driver name.
- bool gslc_Init (gslc_tsGui ∗pGui, void ∗pvDriver, gslc_tsPage ∗asPage, uint8_t nMaxPage, gslc_tsFont ∗as↩ Font, uint8_t nMaxFont)

  Initialize the GUIslice library.
- void gslc_InitDebug (GSLC_CB_DEBUG_OUT pfunc)

  Initialize debug output.
- void gslc_DebugPrintf (const char ∗pFmt,...)

  Optimized printf routine for GUIslice debug/error output.
- bool gslc_GuiRotate (gslc_tsGui ∗pGui, uint8_t nRotation)

  Dynamically change rotation, automatically adapt touchscreen axes swap/flip.
- void gslc_Quit (gslc_tsGui ∗pGui)

  Exit the GUIslice environment.
- void gslc_Update (gslc_tsGui ∗pGui)

  Perform main GUIslice handling functions.
- bool gslc_SetBkgndImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Configure the background to use a bitmap image.
- bool gslc_SetBkgndColor (gslc_tsGui ∗pGui, gslc_tsColor nCol)

  Configure the background to use a solid color.
- bool gslc_SetClipRect (gslc_tsGui ∗pGui, gslc_tsRect ∗pRect)

  Set the clipping rectangle for further drawing.

7.1.1 Detailed Description

General functions for configuring the GUI.

7.1.2 Function Documentation

7.1.2.1 void gslc_DebugPrintf (const char ∗ *pFmt,* *...*)

Optimized printf routine for GUIslice debug/error output.

- Only supports 's','d','u' tokens
- Calls on the output function configured in gslc_InitDebug()

Parameters

in	*pFmt*	Format string to use for printing
in	...	Variable parameter list

Returns

none

7.1.2.2 const char∗ gslc_GetNameDisp (gslc_tsGui ∗ *pGui*)

Get the GUIslice display driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

7.1.2.3 const char∗ gslc_GetNameTouch (gslc_tsGui ∗ *pGui*)

Get the GUIslice touch driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

7.1.2.4 char∗ gslc_GetVer (gslc_tsGui ∗ *pGui*)

Get the GUIslice version number.

Parameters

`in`	*pGui*	Pointer to GUI

Returns

String containing version number

7.1.2.5 bool gslc_GuiRotate (gslc_tsGui ∗ *pGui,* uint8_t *nRotation*)

Dynamically change rotation, automatically adapt touchscreen axes swap/flip.

The function assumes that the touchscreen settings for swap and flip in the GUIslice config are valid for the configured GSLC_ROTATE.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nRotation*	Screen Rotation value (0, 1, 2 or 3)

Returns

true if success, false otherwise

7.1.2.6 bool gslc_Init (gslc_tsGui ∗ *pGui,* void ∗ *pvDriver,* gslc_tsPage ∗ *asPage,* uint8_t *nMaxPage,* gslc_tsFont ∗ *asFont,* uint8_t *nMaxFont*)

Initialize the GUIslice library.

- Configures the primary screen surface(s)
- Initializes font support

PRE:

- The environment variables should be configured before calling gslc_Init().

Parameters

in	*pGui*	Pointer to GUI
in	*pvDriver*	Void pointer to Driver struct (gslc_tsDriver∗)
in	*asPage*	Pointer to Page array
in	*nMaxPage*	Size of Page array
in	*asFont*	Pointer to Font array
in	*nMaxFont*	Size of Font array

Returns

true if success, false if fail

7.1.2.7 void gslc_InitDebug (GSLC_CB_DEBUG_OUT *pfunc*)

Initialize debug output.

- Defines the user function used for debug/error output
- pfunc is responsible for outputing a single character
- For Arduino, this user function would typically call Serial.print()

Parameters

in	*pfunc*	Pointer to user character-out function

Returns

none

7.1.2.8 void gslc_Quit (gslc_tsGui ∗ *pGui*)

Exit the GUIslice environment.

- Calls lower-level destructors to clean up any initialized subsystems and deletes any created elements or fonts

Parameters

in	*pGui*	Pointer to GUI

Returns

None

7.1.2.9 bool gslc_SetBkgndColor (gslc_tsGui ∗ *pGui,* gslc_tsColor *nCol*)

Configure the background to use a solid color.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*nCol*	RGB Color to use

Returns

true if success, false if fail

7.1.2.10 bool gslc_SetBkgndImage (gslc_tsGui ∗ *pGui,* gslc_tsImgRef *sImgRef*)

Configure the background to use a bitmap image.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*sImgRef*	Image reference

Returns

true if success, false if fail

7.1.2.11 bool gslc_SetClipRect (gslc_tsGui ∗ *pGui,* gslc_tsRect ∗ *pRect*)

Set the clipping rectangle for further drawing.

Parameters

in	*pGui*	Pointer to GUI
in	*pRect*	Pointer to Rect for clipping (or NULL for entire screen)

Returns

true if success, false if error

7.1.2.12 void gslc_Update (gslc_tsGui * *pGui*)

Perform main GUIslice handling functions.

- Handles any touch events
- Performs any necessary screen redraw

Parameters

`in`	*pGui*	Pointer to GUI

Returns

None

7.2 Graphics General Functions

Helper functions that support graphics operations.

Functions

- bool gslc_IsInRect (int16_t nSelX, int16_t nSelY, gslc_tsRect rRect)

    Determine if a coordinate is inside of a rectangular region.
- gslc_tsRect gslc_ExpandRect (gslc_tsRect rRect, int16_t nExpandW, int16_t nExpandH)

    Expand or contract a rectangle in width and/or height (equal amounts on both side), based on the centerpoint of the rectangle.
- bool gslc_IsInWH (int16_t nSelX, int16_t nSelY, uint16_t nWidth, uint16_t nHeight)

    Determine if a coordinate is inside of a width x height region.
- void gslc_UnionRect (gslc_tsRect ∗pRect, gslc_tsRect rAddRect)

    Expand a rect to include another rect.
- void gslc_InvalidateRgnReset (gslc_tsGui ∗pGui)

    Reset the invalidation region.
- void gslc_InvalidateRgnPage (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage)

    Include an entire page (eg.
- void gslc_InvalidateRgnScreen (gslc_tsGui ∗pGui)

    Mark the entire screen as invalidated.
- void gslc_InvalidateRgnAdd (gslc_tsGui ∗pGui, gslc_tsRect rAddRect)

    Add a rectangular region to the invalidation region.
- bool gslc_ClipPt (gslc_tsRect ∗pClipRect, int16_t nX, int16_t nY)

    Perform basic clipping of a single point to a clipping region.
- bool gslc_ClipLine (gslc_tsRect ∗pClipRect, int16_t ∗pnX0, int16_t ∗pnY0, int16_t ∗pnX1, int16_t ∗pnY1)

    Perform basic clipping of a line to a clipping region.
- bool gslc_ClipRect (gslc_tsRect ∗pClipRect, gslc_tsRect ∗pRect)

    Perform basic clipping of a rectangle to a clipping region.
- gslc_tsImgRef gslc_GetImageFromFile (const char ∗pFname, gslc_teImgRefFlags eFmt)

    Create an image reference to a bitmap file in LINUX filesystem.
- gslc_tsImgRef gslc_GetImageFromSD (const char ∗pFname, gslc_teImgRefFlags eFmt)

    Create an image reference to a bitmap file in SD card.
- gslc_tsImgRef gslc_GetImageFromRam (unsigned char ∗pImgBuf, gslc_teImgRefFlags eFmt)

    Create an image reference to a bitmap in SRAM.
- gslc_tsImgRef gslc_GetImageFromProg (const unsigned char ∗pImgBuf, gslc_teImgRefFlags eFmt)

    Create an image reference to a bitmap in program memory (PROGMEM)
- void gslc_PolarToXY (uint16_t nRad, int16_t n64Ang, int16_t ∗nDX, int16_t ∗nDY)

    Convert polar coordinate to cartesian.
- int16_t gslc_sinFX (int16_t n64Ang)

    Calculate fixed-point sine function from fractional degrees.
- int16_t gslc_cosFX (int16_t n64Ang)

    Calculate fixed-point cosine function from fractional degrees.
- gslc_tsColor gslc_ColorBlend2 (gslc_tsColor colStart, gslc_tsColor colEnd, uint16_t nMidAmt, uint16_t n←BlendAmt)

    Create a color based on a blend between two colors.
- gslc_tsColor gslc_ColorBlend3 (gslc_tsColor colStart, gslc_tsColor colMid, gslc_tsColor colEnd, uint16_t n←MidAmt, uint16_t nBlendAmt)

    Create a color based on a blend between three colors.
- bool gslc_ColorEqual (gslc_tsColor a, gslc_tsColor b)

    Check whether two colors are equal.

7.2.1 Detailed Description

Helper functions that support graphics operations.

7.2.2 Function Documentation

7.2.2.1 bool gslc_ClipLine (gslc_tsRect ∗ *pClipRect,* int16_t ∗ *pnX0,* int16_t ∗ *pnY0,* int16_t ∗ *pnX1,* int16_t ∗ *pnY1*)

Perform basic clipping of a line to a clipping region.

- Implements Cohen-Sutherland algorithm
- Coordinates in parameter list are modified to fit the region

Parameters

`in`	*pClipRect*	Pointer to clipping region
`in,out`	*pnX0*	Ptr to X coordinate of line start
`in,out`	*pnY0*	Ptr to Y coordinate of line start
`in,out`	*pnX1*	Ptr to X coordinate of line end
`in,out`	*pnY1*	Ptr to Y coordinate of line end

Returns

true if line is visible, false if it should be discarded

7.2.2.2 bool gslc_ClipPt (gslc_tsRect ∗ *pClipRect,* int16_t *nX,* int16_t *nY*)

Perform basic clipping of a single point to a clipping region.

Parameters

`in`	*pClipRect*	Pointer to clipping region
`in`	*nX*	X coordinate of point
`in`	*nY*	Y coordinate of point

Returns

true if point is visible, false if it should be discarded

7.2.2.3 bool gslc_ClipRect (gslc_tsRect ∗ *pClipRect,* gslc_tsRect ∗ *pRect*)

Perform basic clipping of a rectangle to a clipping region.

- Coordinates in parameter rect are modified to fit the region

Parameters

`in`	*pClipRect*	Pointer to clipping region
`in,out`	*pRect*	Ptr to rectangle

Returns

true if rect is visible, false if it should be discarded

7.2.2.4 gslc_tsColor gslc_ColorBlend2 (gslc_tsColor *colStart*, gslc_tsColor *colEnd*, uint16_t *nMidAmt*, uint16_t *nBlendAmt*)

Create a color based on a blend between two colors.

Parameters

`in`	*colStart*	Starting color
`in`	*colEnd*	Ending color
`in`	*nMidAmt*	Position (0..1000) between start and end color at which the midpoint between colors should appear. Normally set to 500 (half-way).
`in`	*nBlendAmt*	The position (0..1000) between start and end at which we want to calculate the resulting blended color.

Returns

Blended color

7.2.2.5 gslc_tsColor gslc_ColorBlend3 (gslc_tsColor *colStart*, gslc_tsColor *colMid*, gslc_tsColor *colEnd*, uint16_t *nMidAmt*, uint16_t *nBlendAmt*)

Create a color based on a blend between three colors.

Parameters

`in`	*colStart*	Starting color
`in`	*colMid*	Intermediate color
`in`	*colEnd*	Ending color
`in`	*nMidAmt*	Position (0..1000) between start and end color at which the intermediate color should appear.
`in`	*nBlendAmt*	The position (0..1000) between start and end at which we want to calculate the resulting blended color.

Returns

Blended color

7.2.2.6 bool gslc_ColorEqual (gslc_tsColor *a*, gslc_tsColor *b*)

Check whether two colors are equal.

Parameters

in	*a*	First color
in	*b*	Second color

Returns

True iff a and b are the same color.

7.2.2.7 int16_t gslc_cosFX (int16_t *n64Ang*)

Calculate fixed-point cosine function from fractional degrees.

- Depending on configuration, the result is derived from either floating point math library or fixed point lookup table.
- gslc_cosFX(nAngDeg∗64)/32768.0 = cos(nAngDeg∗2pi/360)

Parameters

in	*n64Ang*	Angle (in units of 1/64 degrees)

Returns

Fixed-point cosine result. Signed 16-bit; divide by 32768 to get the actual value.

7.2.2.8 gslc_tsRect gslc_ExpandRect (gslc_tsRect *rRect*, int16_t *nExpandW*, int16_t *nExpandH*)

Expand or contract a rectangle in width and/or height (equal amounts on both side), based on the centerpoint of the rectangle.

Parameters

in	*rRect*	Rectangular region before resizing
in	*nExpandW*	Number of pixels to expand the width (if positive) of contract the width (if negative)
in	*nExpandH*	Number of pixels to expand the height (if positive) of contract the height (if negative)

Returns

gslc_tsRect() with resized dimensions

7.2.2.9 gslc_tsImgRef gslc_GetImageFromFile (const char ∗ *pFname,* gslc_teImgRefFlags *eFmt*)

Create an image reference to a bitmap file in LINUX filesystem.

Parameters

in	*pFname*	Pointer to filename string of image in filesystem
in	*eFmt*	Image format

Returns

Loaded image reference

7.2.2.10 gslc_tsImgRef gslc_GetImageFromProg (const unsigned char ∗ *pImgBuf,* gslc_teImgRefFlags *eFmt*)

Create an image reference to a bitmap in program memory (PROGMEM)

Parameters

in	*pImgBuf*	Pointer to image buffer in memory
in	*eFmt*	Image format

Returns

Loaded image reference

7.2.2.11 gslc_tsImgRef gslc_GetImageFromRam (unsigned char ∗ *pImgBuf,* gslc_teImgRefFlags *eFmt*)

Create an image reference to a bitmap in SRAM.

Parameters

in	*pImgBuf*	Pointer to image buffer in memory
in	*eFmt*	Image format

Returns

Loaded image reference

7.2.2.12 gslc_tsImgRef gslc_GetImageFromSD (const char ∗ *pFname,* gslc_teImgRefFlags *eFmt*)

Create an image reference to a bitmap file in SD card.

Parameters

in	*pFname*	Pointer to filename string of image in SD card
in	*eFmt*	Image format

Returns

Loaded image reference

7.2.2.13 void gslc_InvalidateRgnAdd (gslc_tsGui ∗ *pGui*, gslc_tsRect *rAddRect*)

Add a rectangular region to the invalidation region.

- This is usually called when an element has been modified

Parameters

in	*pGui*	Pointer to GUI
in	*rAddRect*	Rectangle to add to the invalidation region

Returns

none

7.2.2.14 void gslc_InvalidateRgnPage (gslc_tsGui ∗ *pGui*, gslc_tsPage ∗ *pPage*)

Include an entire page (eg.

from a page stack) in the invalidation region

Parameters

in	*pGui*	Pointer to GUI
in	*pPage*	Pointer to page

Returns

none

7.2.2.15 void gslc_InvalidateRgnReset (gslc_tsGui ∗ *pGui*)

Reset the invalidation region.

Parameters

`in`	*pGui*	Pointer to GUI

Returns

none

7.2.2.16 void gslc_InvalidateRgnScreen (gslc_tsGui ∗ *pGui*)

Mark the entire screen as invalidated.

Parameters

`in`	*pGui*	Pointer to GUI

Returns

none

7.2.2.17 bool gslc_IsInRect (int16_t *nSelX,* int16_t *nSelY,* gslc_tsRect *rRect*)

Determine if a coordinate is inside of a rectangular region.

- This routine is useful in determining if a touch coordinate is inside of a button.

Parameters

`in`	*nSelX*	X coordinate to test
`in`	*nSelY*	X coordinate to test
`in`	*rRect*	Rectangular region to compare against

Returns

true if inside region, false otherwise

7.2.2.18 bool gslc_IsInWH (int16_t *nSelX,* int16_t *nSelY,* uint16_t *nWidth,* uint16_t *nHeight*)

Determine if a coordinate is inside of a width x height region.

- This routine is useful in determining if a relative coordinate is within a given W x H dimension

Parameters

`in`	*nSelX*	X coordinate to test
`in`	*nSelY*	X coordinate to test
`in`	*nWidth*	Width to test against
`in`	*nHeight*	Height to test against

Returns

true if inside region, false otherwise

7.2.2.19 void gslc_PolarToXY (uint16_t *nRad,* int16_t *n64Ang,* int16_t ∗ *nDX,* int16_t ∗ *nDY*)

Convert polar coordinate to cartesian.

Parameters

`in`	*nRad*	Radius of ray
`in`	*n64Ang*	Angle of ray (in units of 1/64 degrees, 0 is up)
`out`	*nDX*	X offset for ray end
`out`	*nDY*	Y offset for ray end

Returns

none

7.2.2.20 int16_t gslc_sinFX (int16_t *n64Ang*)

Calculate fixed-point sine function from fractional degrees.

- Depending on configuration, the result is derived from either floating point math library or fixed point lookup table.
- gslc_sinFX(nAngDeg∗64)/32768.0 = sin(nAngDeg∗2pi/360)

Parameters

`in`	*n64Ang*	Angle (in units of 1/64 degrees)

Returns

Fixed-point sine result. Signed 16-bit; divide by 32768 to get the actual value.

7.2.2.21 void gslc_UnionRect (gslc_tsRect ∗ *pRect,* gslc_tsRect *rAddRect*)

Expand a rect to include another rect.

- This routine can be useful to modify an invalidation region to include another modified element

Parameters

`in`	*pRect*	Initial rect region
`in`	*rAddRect*	Rectangle to add to the rect region

Returns

none

7.3 Graphics Primitive Functions

These routines cause immediate drawing to occur on the primary screen.

Functions

- void gslc_DrawSetPixel (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, gslc_tsColor nCol)

    Set a pixel on the active screen to the given color with lock.
- void gslc_DrawLine (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)

    Draw an arbitrary line using Bresenham's algorithm.
- void gslc_DrawLineH (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nW, gslc_tsColor nCol)

    Draw a horizontal line.
- void gslc_DrawLineV (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nH, gslc_tsColor nCol)

    Draw a vertical line.
- void gslc_DrawLinePolar (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nRadStart, uint16_t nRadEnd, int16_t n64Ang, gslc_tsColor nCol)

    Draw a polar ray segment.
- void gslc_DrawFrameRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

    Draw a framed rectangle.
- void gslc_DrawFrameRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

    Draw a framed rounded rectangle.
- void gslc_DrawFillRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

    Draw a filled rectangle.
- void gslc_DrawFillRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

    Draw a filled rounded rectangle.
- void gslc_DrawFrameCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)

    Draw a framed circle.
- void gslc_DrawFillCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor n↩
Col)

    Draw a filled circle.
- void gslc_DrawFrameTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

    Draw a framed triangle.
- void gslc_DrawFillTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

    Draw a filled triangle.
- void gslc_DrawFrameQuad (gslc_tsGui ∗pGui, gslc_tsPt ∗psPt, gslc_tsColor nCol)

    Draw a framed quadrilateral.
- void gslc_DrawFillQuad (gslc_tsGui ∗pGui, gslc_tsPt ∗psPt, gslc_tsColor nCol)

    Draw a filled quadrilateral.

7.3.1 Detailed Description

These routines cause immediate drawing to occur on the primary screen.

7.3.2 Function Documentation

7.3.2.1 void gslc_DrawFillCircle (gslc_tsGui ∗ *pGui*, int16_t *nMidX*, int16_t *nMidY*, uint16_t *nRadius*, gslc_tsColor *nCol*)

Draw a filled circle.

Parameters

in	*pGui*	Pointer to GUI
in	*nMidX*	Center X coordinate
in	*nMidY*	Center Y coordinate
in	*nRadius*	Radius of circle
in	*nCol*	Color RGB value for the fill

Returns

none

7.3.2.2 void gslc_DrawFillQuad (gslc_tsGui ∗ *pGui*, gslc_tsPt ∗ *psPt*, gslc_tsColor *nCol*)

Draw a filled quadrilateral.

Parameters

in	*pGui*	Pointer to GUI
in	*psPt*	Pointer to array of 4 points
in	*nCol*	Color RGB value for the frame

Returns

true if success, false if error

7.3.2.3 void gslc_DrawFillRect (gslc_tsGui ∗ *pGui*, gslc_tsRect *rRect*, gslc_tsColor *nCol*)

Draw a filled rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nCol*	Color RGB value to fill

Returns

none

7.3.2.4 void gslc_DrawFillRoundRect (gslc_tsGui ∗ *pGui*, gslc_tsRect *rRect*, int16_t *nRadius*, gslc_tsColor *nCol*)

Draw a filled rounded rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nRadius*	Radius for the rounded corners
in	*nCol*	Color RGB value to fill

Returns

none

7.3.2.5 void gslc_DrawFillTriangle (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* int16_t *nX2,* int16_t *nY2,* gslc_tsColor *nCol*)

Draw a filled triangle.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X Coordinate #1
in	*nY0*	Y Coordinate #1
in	*nX1*	X Coordinate #2
in	*nY1*	Y Coordinate #2
in	*nX2*	X Coordinate #3
in	*nY2*	Y Coordinate #3
in	*nCol*	Color RGB value for the fill

Returns

true if success, false if error

7.3.2.6 void gslc_DrawFrameCircle (gslc_tsGui ∗ *pGui,* int16_t *nMidX,* int16_t *nMidY,* uint16_t *nRadius,* gslc_tsColor *nCol*)

Draw a framed circle.

Parameters

in	*pGui*	Pointer to GUI
in	*nMidX*	Center X coordinate
in	*nMidY*	Center Y coordinate
in	*nRadius*	Radius of circle
in	*nCol*	Color RGB value for the frame

Returns

none

7.3.2.7 void gslc_DrawFrameQuad (gslc_tsGui ∗ *pGui,* gslc_tsPt ∗ *psPt,* gslc_tsColor *nCol*)

Draw a framed quadrilateral.

Parameters

in	*pGui*	Pointer to GUI
in	*psPt*	Pointer to array of 4 points
in	*nCol*	Color RGB value for the frame

Returns

true if success, false if error

7.3.2.8 void gslc_DrawFrameRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* gslc_tsColor *nCol*)

Draw a framed rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nCol*	Color RGB value for the frame

Returns

none

7.3.2.9 void gslc_DrawFrameRoundRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* int16_t *nRadius,* gslc_tsColor *nCol*)

Draw a framed rounded rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nRadius*	Radius for the rounded corners
in	*nCol*	Color RGB value for the frame

Returns

none

7.3.2.10 void gslc_DrawFrameTriangle (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* int16_t *nX2,* int16_t *nY2,* gslc_tsColor *nCol*)

Draw a framed triangle.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X Coordinate #1
in	*nY0*	Y Coordinate #1
in	*nX1*	X Coordinate #2
in	*nY1*	Y Coordinate #2
in	*nX2*	X Coordinate #3
in	*nY2*	Y Coordinate #3
in	*nCol*	Color RGB value for the frame

Returns

true if success, false if error

7.3.2.11 void gslc_DrawLine (gslc_tsGui ∗ *pGui*, int16_t *nX0*, int16_t *nY0*, int16_t *nX1*, int16_t *nY1*, gslc_tsColor *nCol*)

Draw an arbitrary line using Bresenham's algorithm.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X coordinate of line startpoint
in	*nY0*	Y coordinate of line startpoint
in	*nX1*	X coordinate of line endpoint
in	*nY1*	Y coordinate of line endpoint
in	*nCol*	Color RGB value for the line

Returns

none

7.3.2.12 void gslc_DrawLineH (gslc_tsGui ∗ *pGui*, int16_t *nX*, int16_t *nY*, uint16_t *nW*, gslc_tsColor *nCol*)

Draw a horizontal line.

- Note that direction of line is in +ve X axis

Parameters

in	*pGui*	Pointer to GUI
in	*nX*	X coordinate of line startpoint
in	*nY*	Y coordinate of line startpoint
in	*nW*	Width of line (in +X direction)
in	*nCol*	Color RGB value for the line

Returns

none

7.3.2.13 void gslc_DrawLinePolar (gslc_tsGui ∗ *pGui*, int16_t *nX*, int16_t *nY*, uint16_t *nRadStart*, uint16_t *nRadEnd*, int16_t *n64Ang*, gslc_tsColor *nCol*)

Draw a polar ray segment.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nX*	X coordinate of line startpoint
`in`	*nY*	Y coordinate of line startpoint
`in`	*nRadStart*	Starting radius of line
`in`	*nRadEnd*	Ending radius of line
`in`	*n64Ang*	Angle of ray (degrees ∗ 64). 0 is up, +90∗64 is to right From -180∗64 to +180∗64
`in`	*nCol*	Color RGB value for the line

Returns

none

7.3.2.14 void gslc_DrawLineV (gslc_tsGui ∗ *pGui*, int16_t *nX*, int16_t *nY*, uint16_t *nH*, gslc_tsColor *nCol*)

Draw a vertical line.

- Note that direction of line is in +ve Y axis

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nX*	X coordinate of line startpoint
`in`	*nY*	Y coordinate of line startpoint
`in`	*nH*	Height of line (in +Y direction)
`in`	*nCol*	Color RGB value for the line

Returns

none

7.3.2.15 void gslc_DrawSetPixel (gslc_tsGui ∗ *pGui*, int16_t *nX*, int16_t *nY*, gslc_tsColor *nCol*)

Set a pixel on the active screen to the given color with lock.

- Calls upon gslc_DrvDrawSetPixelRaw() but wraps with a surface lock lock
- If repeated access is needed, use gslc_DrvDrawSetPixelRaw() instead

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nX*	Pixel X coordinate to set
`in`	*nY*	Pixel Y coordinate to set
`in`	*nCol*	Color pixel value to assign

Returns

none

7.4 Font Functions

Functions that load fonts.

Functions

- bool gslc_FontAdd (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefType eFontRefType, const void ∗pv↩ FontRef, uint16_t nFontSz)

  Load a font into the local font cache and assign font ID (nFontId).
- bool gslc_FontSet (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefType eFontRefType, const void ∗pv↩ FontRef, uint16_t nFontSz)

  Load a font into the local font cache and store as font ID (nFontId)
- gslc_tsFont ∗ gslc_FontGet (gslc_tsGui ∗pGui, int16_t nFontId)

  Fetch a font from its ID value.
- bool gslc_FontSetMode (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefMode eFontMode)

  Set a font's mode.

7.4.1 Detailed Description

Functions that load fonts.

7.4.2 Function Documentation

7.4.2.1 bool gslc_FontAdd (gslc_tsGui ∗ *pGui,* int16_t *nFontId,* gslc_teFontRefType *eFontRefType,* const void ∗ *pvFontRef,* uint16_t *nFontSz*)

Load a font into the local font cache and assign font ID (nFontId).

- Font is stored into next available internal array element
- NOTE: Use FontSet() instead

Parameters

in	*pGui*	Pointer to GUI
in	*nFontId*	ID to use when referencing this font
in	*eFontRefType*	Font reference type (eg. filename or pointer)
in	*pvFontRef*	Reference pointer to identify the font. In the case of SDL mode, it is a filepath to the font file. In the case of Arduino it is a pointer value to the font bitmap array (GFXFont)
in	*nFontSz*	Typeface size to use (only used in SDL mode)

Returns

true if load was successful, false otherwise

7.4.2.2 gslc_tsFont∗ gslc_FontGet (gslc_tsGui ∗ *pGui*, int16_t *nFontId*)

Fetch a font from its ID value.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ FontId*	ID value used to reference the font (supplied originally to gslc_FontAdd()

Returns

A pointer to the font structure or NULL if error

7.4.2.3 bool gslc_FontSet (gslc_tsGui ∗ *pGui*, int16_t *nFontId*, gslc_teFontRefType *eFontRefType*, const void ∗ *pvFontRef*, uint16_t *nFontSz*)

Load a font into the local font cache and store as font ID (nFontId)

- Font is stored into index nFontId, so nFontId must be from separate font enum (0-based).
- Example: enum { E_FONT_BTN, E_FONT_TXT, MAX_FONT };

Parameters

in	*pGui*	Pointer to GUI
in	*nFontId*	ID to use when referencing this font
in	*eFontRefType*	Font reference type (eg. filename or pointer)
in	*pvFontRef*	Reference pointer to identify the font. In the case of SDL mode, it is a filepath to the font file. In the case of Arduino it is a pointer value to the font bitmap array (GFXFont)
in	*nFontSz*	Typeface size to use (only used in SDL mode)

Returns

true if load was successful, false otherwise

7.4.2.4 bool gslc_FontSetMode (gslc_tsGui ∗ *pGui*, int16_t *nFontId*, gslc_teFontRefMode *eFontMode*)

Set a font's mode.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ FontId*	ID value used to reference the font (supplied originally to gslc_FontAdd()

7.5 Page Functions

Functions that operate at the page level.

Functions

- int gslc_GetPageCur (gslc_tsGui ∗pGui)

  Fetch the current page ID.
- void gslc_SetStackPage (gslc_tsGui ∗pGui, uint8_t nStackPos, int16_t nPageId)

  Assign a page to the page stack.
- void gslc_SetStackState (gslc_tsGui ∗pGui, uint8_t nStackPos, bool bActive, bool bDoDraw)

  Change the status of a page in a page stack.
- void gslc_SetPageBase (gslc_tsGui ∗pGui, int16_t nPageId)

  Assigns a page for the base layer in the page stack.
- void gslc_SetPageCur (gslc_tsGui ∗pGui, int16_t nPageId)

  Select a page for the current layer in the page stack.
- void gslc_SetPageOverlay (gslc_tsGui ∗pGui, int16_t nPageId)

  Select a page for the overlay layer in the page stack.
- void gslc_PopupShow (gslc_tsGui ∗pGui, int16_t nPageId, bool bModal)

  Show a popup dialog.
- void gslc_PopupHide (gslc_tsGui ∗pGui)

  Hides the currently active popup dialog.
- void gslc_PageRedrawSet (gslc_tsGui ∗pGui, bool bRedraw)

  Update the need-redraw status for the current page.
- bool gslc_PageRedrawGet (gslc_tsGui ∗pGui)

  Get the need-redraw status for the current page.
- void gslc_PageAdd (gslc_tsGui ∗pGui, int16_t nPageId, gslc_tsElem ∗psElem, uint16_t nMaxElem, gslc_↩ tsElemRef ∗psElemRef, uint16_t nMaxElemRef)

  Add a page to the GUI.
- gslc_tsElemRef ∗ gslc_PageFindElemById (gslc_tsGui ∗pGui, int16_t nPageId, int16_t nElemId)

  Find an element in the GUI by its Page ID and Element ID.

7.5.1 Detailed Description

Functions that operate at the page level.

7.5.2 Function Documentation

7.5.2.1 int gslc_GetPageCur (gslc_tsGui ∗ *pGui*)

Fetch the current page ID.

Parameters

in	*pGui*	Pointer to GUI

Returns

Page ID

7.5.2.2 void gslc_PageAdd (gslc_tsGui ∗ *pGui,* int16_t *nPageId,* gslc_tsElem ∗ *psElem,* uint16_t *nMaxElem,* gslc_tsElemRef ∗ *psElemRef,* uint16_t *nMaxElemRef*)

Add a page to the GUI.

- This call associates an element array with the collection within the page
- Once a page has been added to the GUI, elements can be added to the page by specifying the same page ID

Parameters

in	*pGui*	Pointer to GUI
in	*nPageId*	Page ID to assign
in	*psElem*	Internal element array storage to associate with the page
in	*nMaxElem*	Maximum number of elements that can be added to the internal element array (ie. RAM))
in	*psElemRef*	Internal element reference array storage to associate with the page. All elements, whether they are located in the internal element array or in external Flash (PROGMEM) storage, require an entry in the element reference array.
in	*nMaxElemRef*	Maximum number of elements in the reference array. This is effectively the maximum number of elements that can appear on a page, irrespective of whether it is stored in RAM or Flash (PROGMEM).

Returns

none

7.5.2.3 gslc_tsElemRef∗ gslc_PageFindElemById (gslc_tsGui ∗ *pGui,* int16_t *nPageId,* int16_t *nElemId*)

Find an element in the GUI by its Page ID and Element ID.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ PageId*	Page ID to search
in	*n↩ ElemId*	Element ID to search

Returns

Ptr to an element or NULL if none found

7.5.2.4 bool gslc_PageRedrawGet (gslc_tsGui ∗ *pGui*)

Get the need-redraw status for the current page.

Parameters

in	*pGui*	Pointer to GUI

Returns

True if redraw required, false otherwise

7.5.2.5 void gslc_PageRedrawSet (gslc_tsGui ∗ *pGui,* bool *bRedraw*)

Update the need-redraw status for the current page.

Parameters

in	*pGui*	Pointer to GUI
in	*bRedraw*	True if redraw required, false otherwise

Returns

none

7.5.2.6 void gslc_PopupHide (gslc_tsGui ∗ *pGui*)

Hides the currently active popup dialog.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

7.5.2.7 void gslc_PopupShow (gslc_tsGui ∗ *pGui,* int16_t *nPageId,* bool *bModal*)

Show a popup dialog.

- Popup dialogs use the overlay layer in the page stack

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ PageId*	Page ID to use as the popup dialog
in	*bModal*	If true, popup is modal (other layers won't accept touch). If false, popup is modeless (other layers still accept touch)

Returns

none

7.5.2.8 void gslc_SetPageBase (gslc_tsGui ∗ *pGui,* int16_t *nPageId*)

Assigns a page for the base layer in the page stack.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ PageId*	Page ID to select (or GSLC_PAGE_NONE to disable)

Returns

none

7.5.2.9 void gslc_SetPageCur (gslc_tsGui ∗ *pGui,* int16_t *nPageId*)

Select a page for the current layer in the page stack.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ PageId*	Page ID to select

Returns

none

7.5.2.10 void gslc_SetPageOverlay (gslc_tsGui ∗ *pGui,* int16_t *nPageId*)

Select a page for the overlay layer in the page stack.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ PageId*	Page ID to select (or GSLC_PAGE_NONE to disable)

Returns

none

7.5.2.11 void gslc_SetStackPage (gslc_tsGui ∗ *pGui,* uint8_t *nStackPos,* int16_t *nPageId*)

Assign a page to the page stack.

Parameters

in	*pGui*	Pointer to GUI
in	*nStackPos*	Position to update in the page stack (0..GSLC_STACK__MAX-1)
in	*nPageId*	Page ID to select as current

Returns

none

7.5.2.12 void gslc_SetStackState (gslc_tsGui ∗ *pGui,* uint8_t *nStackPos,* bool *bActive,* bool *bDoDraw*)

Change the status of a page in a page stack.

Parameters

in	*pGui*	Pointer to GUI
in	*nStackPos*	Position to update in the page stack (0..GSLC_STACK__MAX-1)
in	*bActive*	Indicate if page should receive touch events
in	*bDoDraw*	Indicate if page should continue to be redrawn. If pages in the stack are overlapping and an element in a lower layer continues to receive updates, then the element may "show through" the layers above it. In such cases where pages in the stack are overlapping and lower pages contain dynamically updating elements, it may be best to disable redraw while the overlapping page is visible (by setting bDoDraw to false).

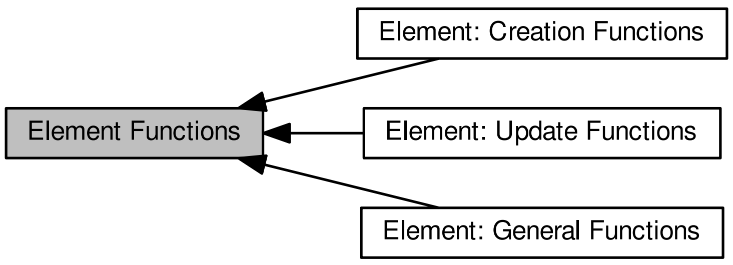
Returns

none

7.6 Element Functions

Functions that are used to create and manipulate elements.

Collaboration diagram for Element Functions:


Element: Creation Functions
Element Functions
Element: Update Functions
Element: General Functions

Modules

- Element: Creation Functions

  Functions that create GUI elements.
- Element: General Functions

  General-purpose functions that operate on Elements.
- Element: Update Functions

  Functions that configure or modify an existing eleemnt.

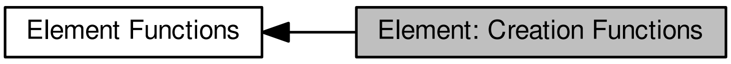
7.6.1 Detailed Description

Functions that are used to create and manipulate elements.

7.7 Element: Creation Functions

Functions that create GUI elements.

Collaboration diagram for Element: Creation Functions:



Functions

- gslc_tsElemRef ∗ gslc_ElemCreateTxt (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

    Create a Text Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBtnTxt (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId, GSLC_CB_TOUCH cbTouch)

    Create a textual Button Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBtnImg (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, gslc_tsImgRef sImgRef, gslc_tsImgRef sImgRefSel, GSLC_CB_TOUCH cbTouch)

    Create a graphical Button Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBox (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect r↩ Elem)

    Create a Box Element.
- gslc_tsElemRef ∗ gslc_ElemCreateLine (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1)

    Create a Line Element.
- gslc_tsElemRef ∗ gslc_ElemCreateImg (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect r↩ Elem, gslc_tsImgRef sImgRef)

    Create an image Element.

7.7.1 Detailed Description

Functions that create GUI elements.

7.7.2 Function Documentation

7.7.2.1 gslc_tsElemRef∗ gslc_ElemCreateBox (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsRect *rElem*)

Create a Box Element.

- Draws a box with frame and fill

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ ElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*rElem*	Rectangle coordinates defining box size

Returns

Pointer to the Element reference or NULL if failure

7.7.2.2 gslc_tsElemRef∗ gslc_ElemCreateBtnImg (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsRect *rElem,* gslc_tsImgRef *sImgRef,* gslc_tsImgRef *sImgRefSel,* GSLC_CB_TOUCH *cbTouch*)

Create a graphical Button Element.

- Creates a clickable element that uses a BMP image with no frame or fill
- Transparency is supported by bitmap color (0xFF00FF)

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*rElem*	Rectangle coordinates defining image size
in	*sImgRef*	Image reference to load (unselected state)
in	*sImgRefSel*	Image reference to load (selected state)
in	*cbTouch*	Callback for touch events

Returns

Pointer to the Element reference or NULL if failure

7.7.2.3 gslc_tsElemRef∗ gslc_ElemCreateBtnTxt (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsRect *rElem,* char ∗ *pStrBuf,* uint8_t *nStrBufMax,* int16_t *nFontId,* GSLC_CB_TOUCH *cbTouch*)

Create a textual Button Element.

- Creates a clickable element that has a textual label with frame and fill

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)

Parameters

in	*nPage*	Page ID to attach element to
in	*rElem*	Rectangle coordinates defining text background size
in	*pStrBuf*	String to copy into element
in	*nStrBufMax*	Maximum length of string buffer (pStrBuf). Only applicable if GSLC_LOCAL_STR=0. Ignored if GSLC_LOCAL_STR=1.)
in	*nFontId*	Font ID to use for text display
in	*cbTouch*	Callback for touch events

Returns

Pointer to the Element reference or NULL if failure

7.7.2.4 gslc_tsElemRef∗ gslc_ElemCreateImg (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsRect *rElem,* gslc_tsImgRef *sImgRef*)

Create an image Element.

- Draws an image

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ ElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*rElem*	Rectangle coordinates defining box size
in	*sImgRef*	Image reference to load

Returns

Pointer to the Element reference or NULL if failure

7.7.2.5 gslc_tsElemRef∗ gslc_ElemCreateLine (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1*)

Create a Line Element.

- Draws a line with fill color

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ ElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)

Parameters

in	*nPage*	Page ID to attach element to
in	*nX0*	X coordinate of line startpoint
in	*nY0*	Y coordinate of line startpoint
in	*nX1*	X coordinate of line endpoint
in	*nY1*	Y coordinate of line endpoint

Returns

Pointer to the Element reference or NULL if failure

7.7.2.6 gslc_tsElemRef∗ gslc_ElemCreateTxt (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsRect *rElem,* char ∗ *pStrBuf,* uint8_t *nStrBufMax,* int16_t *nFontId*)

Create a Text Element.

- Draws a text string with filled background

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*rElem*	Rectangle coordinates defining text background size
in	*pStrBuf*	String to copy into element
in	*nStrBufMax*	Maximum length of string buffer (pStrBuf). Only applicable if GSLC_LOCAL_STR=0. Ignored if GSLC_LOCAL_STR=1.)
in	*nFontId*	Font ID to use for text display

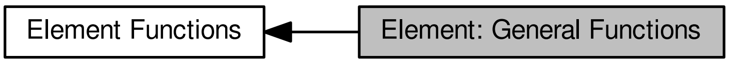
Returns

Pointer to the Element reference or NULL if failure

7.8 Element: General Functions

General-purpose functions that operate on Elements.

Collaboration diagram for Element: General Functions:



Functions

- int gslc_ElemGetId (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

    Get an Element ID from an element structure.

7.8.1 Detailed Description

General-purpose functions that operate on Elements.

7.8.2 Function Documentation

7.8.2.1 int gslc_ElemGetId (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*)

Get an Element ID from an element structure.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference structure

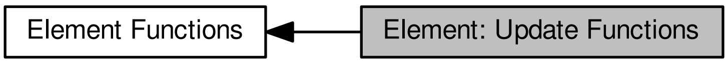
Returns

ID of element or GSLC_ID_NONE if not found

7.9 Element: Update Functions

Functions that configure or modify an existing eleemnt.

Collaboration diagram for Element: Update Functions:



Functions

- void gslc_ElemSetFillEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bFillEn)

  Set the fill state for an Element.
- void gslc_ElemSetFrameEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bFrameEn)

  Set the frame state for an Element.
- void gslc_ElemSetRoundEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bRoundEn)

  Set the rounded frame/fill state for an Element.
- void gslc_ElemSetCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colFrame, gslc_tsColor colFill, gslc_tsColor colFillGlow)

  Update the common color selection for an Element.
- void gslc_ElemSetGlowCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colFrameGlow, gslc_tsColor colFillGlow, gslc_tsColor colTxtGlow)

  Update the common color selection for glowing state of an Element.
- void gslc_ElemSetGroup (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int nGroupId)

  Set the group ID for an element.
- int gslc_ElemGetGroup (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the group ID for an element.
- void gslc_ElemSetTxtAlign (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, unsigned nAlign)

  Set the alignment of a textual element (horizontal and vertical)
- void gslc_ElemSetTxtMargin (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, unsigned nMargin)

  Set the margin around of a textual element.
- void gslc_ElemSetTxtStr (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, const char ∗pStr)

  Update the text string associated with an Element.
- char ∗ gslc_ElemGetTxtStr (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Fetch the current text string associated with an Element.
- void gslc_ElemSetTxtCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colVal)

  Update the text string color associated with an Element ID.
- void gslc_ElemSetTxtMem (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teTxtFlags eFlags)

  Update the text string location in memory.
- void gslc_ElemSetTxtEnc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teTxtFlags eFlags)

  Update the text string encoding mode.
- void gslc_ElemUpdateFont (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int nFontId)

  Update the Font selected for an Element's text.
- void gslc_ElemSetRedraw (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teRedrawType eRedraw)

Update the need-redraw status for an element.

- gslc_teRedrawType gslc_ElemGetRedraw (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the need-redraw status for an element.
- void gslc_ElemSetGlowEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bGlowEn)

  Update the glowing enable for an element.
- void gslc_ElemSetClickEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bClickEn)

  Update the click enable for an element.
- void gslc_ElemSetTouchFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_TOUCH funcCb)

  Update the touch function callback for an element.
- void gslc_ElemSetStyleFrom (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRefSrc, gslc_tsElemRef ∗pElem↩ RefDest)

  Copy style settings from one element to another.
- bool gslc_ElemGetGlowEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the glowing enable for an element.
- void gslc_ElemSetGlow (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bGlowing)

  Update the glowing indicator for an element.
- bool gslc_ElemGetGlow (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the glowing indicator for an element.
- void gslc_ElemSetVisible (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bVisible)

  Update the visibility status for an element.
- bool gslc_ElemGetVisible (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the visibility status for an element.
- void gslc_ElemSetDrawFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_DRAW funcCb)

  Assign the drawing callback function for an element.
- void gslc_ElemSetTickFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_TICK funcCb)

  Assign the tick callback function for an element.
- bool gslc_ElemOwnsCoord (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nX, int16_t nY, bool b↩ OnlyClickEn)

  Determine if a coordinate is inside of an element.

7.9.1 Detailed Description

Functions that configure or modify an existing eleemnt.

7.9.2 Function Documentation

7.9.2.1 bool gslc_ElemGetGlow (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get the glowing indicator for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

True if element is glowing

7.9.2.2 bool gslc_ElemGetGlowEn (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get the glowing enable for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

True if element supports glowing

7.9.2.3 int gslc_ElemGetGroup (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get the group ID for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

Group ID or GSLC_GROUP_ID_NONE if unassigned

7.9.2.4 gslc_teRedrawType gslc_ElemGetRedraw (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get the need-redraw status for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

Redraw status

7.9.2.5 char∗ gslc_ElemGetTxtStr (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Fetch the current text string associated with an Element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

Pointer to character array string

7.9.2.6 bool gslc_ElemGetVisible (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*)

Get the visibility status for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

True if element is shown, false if hidden

7.9.2.7 bool gslc_ElemOwnsCoord (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int16_t *nX*, int16_t *nY*, bool *bOnlyClickEn*)

Determine if a coordinate is inside of an element.

- This routine is useful in determining if a touch coordinate is inside of a button.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Element reference used for boundary test
in	*nX*	X coordinate to test
in	*nY*	Y coordinate to test
in	*bOnlyClickEn*	Only output true if element was also marked as "clickable" (eg. bClickEn=true)

Returns

true if inside element, false otherwise

7.9.2.8 void gslc_ElemSetClickEn (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, bool *bClickEn*)

Update the click enable for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bClickEn*	True if element should support click events

Returns

none

7.9.2.9 void gslc_ElemSetCol (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_tsColor *colFrame*, gslc_tsColor *colFill*, gslc_tsColor *colFillGlow*)

Update the common color selection for an Element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*colFrame*	Color for the frame
in	*colFill*	Color for the fill
in	*colFillGlow*	Color for the fill when glowing

Returns

none

7.9.2.10 void gslc_ElemSetDrawFunc (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, GSLC_CB_DRAW *funcCb*)

Assign the drawing callback function for an element.

- This allows the user to override the default rendering for an element, enabling the creation of a custom element

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*funcCb*	Function pointer to drawing routine (or NULL for default))

Returns

none

7.9.2.11 void gslc_ElemSetFillEn (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, bool *bFillEn*)

Set the fill state for an Element.

- If not filled, the element can support transparency against an arbitrary background, but this can require full screen redraws if the element is updated.
- If filled, the background fill color can be changed by gslc_ElemSetCol()

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bFillEn*	True if filled, false otherwise

Returns

none

7.9.2.12 void gslc_ElemSetFrameEn (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, bool *bFrameEn*)

Set the frame state for an Element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bFrameEn*	True if framed, false otherwise

Returns

none

7.9.2.13 void gslc_ElemSetGlow (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, bool *bGlowing*)

Update the glowing indicator for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bGlowing*	True if element is glowing

Returns

none

7.9.2.14 void gslc_ElemSetGlowCol (**gslc_tsGui** ∗ *pGui,* **gslc_tsElemRef** ∗ *pElemRef,* **gslc_tsColor** *colFrameGlow,* **gslc_tsColor** *colFillGlow,* **gslc_tsColor** *colTxtGlow*)

Update the common color selection for glowing state of an Element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*colFrameGlow*	Color for the frame when glowing
in	*colFillGlow*	Color for the fill when glowing
in	*colTxtGlow*	Color for the text when glowing

Returns

none

7.9.2.15 void gslc_ElemSetGlowEn (**gslc_tsGui** ∗ *pGui,* **gslc_tsElemRef** ∗ *pElemRef,* bool *bGlowEn*)

Update the glowing enable for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bGlowEn*	True if element should support glowing

Returns

none

7.9.2.16 void gslc_ElemSetGroup (**gslc_tsGui** ∗ *pGui,* **gslc_tsElemRef** ∗ *pElemRef,* int *nGroupId*)

Set the group ID for an element.

- Typically used to associate radio button elements together

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nGroupId*	Group ID to assign

Returns

none

7.9.2.17 void gslc_ElemSetRedraw (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_teRedrawType *eRedraw*)

Update the need-redraw status for an element.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*eRedraw*	Redraw state to set

Returns

none

7.9.2.18 void gslc_ElemSetRoundEn (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, bool *bRoundEn*)

Set the rounded frame/fill state for an Element.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*bRoundEn*	True if rounded, false otherwise

Returns

none

7.9.2.19 void gslc_ElemSetStyleFrom (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRefSrc*, gslc_tsElemRef ∗ *pElemRefDest*)

Copy style settings from one element to another.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRefSrc*	Pointer to source Element reference
`in`	*pElemRefDest*	Pointer to destination Element reference

Returns

none

7.9.2.20 void gslc_ElemSetTickFunc (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, GSLC_CB_TICK *funcCb*)

Assign the tick callback function for an element.

- This allows the user to provide background updates to an element triggered by the main loop call to gslc_↩ Update()

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*funcCb*	Function pointer to tick routine (or NULL for none))

Returns

none

7.9.2.21 void gslc_ElemSetTouchFunc (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, GSLC_CB_TOUCH *funcCb*)

Update the touch function callback for an element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*cbTouch*	Pointer to the touch callback function

Returns

none

7.9.2.22 void gslc_ElemSetTxtAlign (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, unsigned *nAlign*)

Set the alignment of a textual element (horizontal and vertical)

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Parameters

in	*nAlign*	Alignment to specify: • GSLC_ALIGN_TOP_LEFT • GSLC_ALIGN_TOP_MID • GSLC_ALIGN_TOP_RIGHT • GSLC_ALIGN_MID_LEFT • GSLC_ALIGN_MID_MID • GSLC_ALIGN_MID_RIGHT • GSLC_ALIGN_BOT_LEFT • GSLC_ALIGN_BOT_MID • GSLC_ALIGN_BOT_RIGHT

Returns

none

7.9.2.23 void gslc_ElemSetTxtCol (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_tsColor *colVal*)

Update the text string color associated with an Element ID.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*colVal*	RGB color to change to

Returns

none

7.9.2.24 void gslc_ElemSetTxtEnc (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_teTxtFlags *eFlags*)

Update the text string encoding mode.

- This function can be used to indicate that the element's text string is encoded in UTF-8, which supports extended / foreign character maps

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*eFlags*	Flags associated with text encoding (GSLC_TXT_ENC_∗)

Returns

none

7.9.2.25 void gslc_ElemSetTxtMargin (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, unsigned *nMargin*)

Set the margin around of a textual element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nMargin*	Number of pixels gap to leave surrounding text

Returns

none

7.9.2.26 void gslc_ElemSetTxtMem (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_teTxtFlags *eFlags*)

Update the text string location in memory.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*eFlags*	Flags associated with text memory location (GSLC_TXT_MEM_∗)

Returns

none

7.9.2.27 void gslc_ElemSetTxtStr (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, const char ∗ *pStr*)

Update the text string associated with an Element.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*pStr*	String to copy into element

Returns

none

7.9.2.28 void gslc_ElemSetVisible (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* bool *bVisible*)

Update the visibility status for an element.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*bVisible*	True if element is shown, false if hidden

Returns

none

7.9.2.29 void gslc_ElemUpdateFont (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int *nFontId*)

Update the Font selected for an Element's text.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*nFontId*	Font ID to select

Returns

none

7.10 Touchscreen Functions

Functions that configure and respond to a touch device.

Macros

- #define TOUCH_ROTATION_DATA

    Additional definitions for Touch Handling These macros define the transforms used in remapping the touchscreen inputs on the basis of the GUI nRotation setting.
- #define TOUCH_ROTATION_DATA

    Additional definitions for Touch Handling These macros define the transforms used in remapping the touchscreen inputs on the basis of the GUI nRotation setting.
- #define TOUCH_ROTATION_SWAPXY(rotation)
- #define TOUCH_ROTATION_SWAPXY(rotation)
- #define TOUCH_ROTATION_FLIPX(rotation)
- #define TOUCH_ROTATION_FLIPX(rotation)
- #define TOUCH_ROTATION_FLIPY(rotation)
- #define TOUCH_ROTATION_FLIPY(rotation)

Functions

- bool gslc_InitTouch (gslc_tsGui ∗pGui, const char ∗acDev)

    Initialize the touchscreen device driver.
- bool gslc_GetTouch (gslc_tsGui ∗pGui, int16_t ∗pnX, int16_t ∗pnY, uint16_t ∗pnPress, gslc_teInputRawEvent ∗peInputEvent, int16_t ∗pnInputVal)

    Initialize the touchscreen device driver.
- void gslc_SetTouchRemapEn (gslc_tsGui ∗pGui, bool bEn)

    Configure touchscreen remapping.
- void gslc_SetTouchRemapCal (gslc_tsGui ∗pGui, uint16_t nXMin, uint16_t nXMax, uint16_t nYMin, uint16_t nYMax)

    Configure touchscreen calibration values.
- void gslc_SetTouchRemapYX (gslc_tsGui ∗pGui, bool bSwap)

    Configure touchscreen XY swap.

7.10.1 Detailed Description

Functions that configure and respond to a touch device.

7.10.2 Macro Definition Documentation

7.10.2.1 #define TOUCH_ROTATION_DATA

Additional definitions for Touch Handling These macros define the transforms used in remapping the touchscreen inputs on the basis of the GUI nRotation setting.

7.10.2.2 #define TOUCH_ROTATION_DATA

Additional definitions for Touch Handling These macros define the transforms used in remapping the touchscreen inputs on the basis of the GUI nRotation setting.

7.10.2.3 #define TOUCH_ROTATION_FLIPX(*rotation*)

7.10.2.4 #define TOUCH_ROTATION_FLIPX(*rotation*)

7.10.2.5 #define TOUCH_ROTATION_FLIPY(*rotation*)

7.10.2.6 #define TOUCH_ROTATION_FLIPY(*rotation*)

7.10.2.7 #define TOUCH_ROTATION_SWAPXY(*rotation*)

7.10.2.8 #define TOUCH_ROTATION_SWAPXY(*rotation*)

7.10.3 Function Documentation

7.10.3.1 bool gslc_GetTouch (gslc_tsGui ∗ *pGui,* int16_t ∗ *pnX,* int16_t ∗ *pnY,* uint16_t ∗ *pnPress,* gslc_teInputRawEvent ∗ *peInputEvent,* int16_t ∗ *pnInputVal*)

Initialize the touchscreen device driver.

Parameters

`in`	*pGui*	Pointer to GUI
`out`	*pnX*	Ptr to int to contain latest touch X coordinate
`out`	*pnY*	Ptr to int to contain latest touch Y coordinate
`out`	*pnPress*	Ptr to int to contain latest touch pressure value
`out`	*peInputEvent*	Indication of event type
`out`	*pnInputVal*	Additional data for event type

Returns

true if touch event, false otherwise

7.10.3.2 bool gslc_InitTouch (gslc_tsGui ∗ *pGui,* const char ∗ *acDev*)

Initialize the touchscreen device driver.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*acDev*	Device path to touchscreen (or "" if not applicable)) eg. "/dev/input/touchscreen"

Returns

true if successful

7.10.3.3 void gslc_SetTouchRemapCal (gslc_tsGui ∗ *pGui,* uint16_t *nXMin,* uint16_t *nXMax,* uint16_t *nYMin,* uint16_t *nYMax*)

Configure touchscreen calibration values.

- Only used if calibration remapping has been enabled

Parameters

in	*pGui*	Pointer to GUI
in	*nXMin*	Resistive touchscreen X_MIN calibration value
in	*nXMax*	Resistive touchscreen X_MAX calibration value
in	*nYMin*	Resistive touchscreen Y_MIN calibration value
in	*nYMax*	Resistive touchscreen Y_MAX calibration value

Returns

none

7.10.3.4 void gslc_SetTouchRemapEn (gslc_tsGui ∗ *pGui,* bool *bEn*)

Configure touchscreen remapping.

Parameters

in	*pGui*	Pointer to GUI
in	*bEn*	Enable touchscreen remapping?

Returns

none

7.10.3.5 void gslc_SetTouchRemapYX (gslc_tsGui ∗ *pGui,* bool *bSwap*)

Configure touchscreen XY swap.

Parameters

in	*pGui*	Pointer to GUI
in	*bSwap*	Enable touchscreen XY swap

Returns

none

7.11 Input Mapping Functions

Functions that handle GPIO / pin and keyboard input.

Functions

- void gslc_SetPinPollFunc (gslc_tsGui ∗pGui, GSLC_CB_PIN_POLL pfunc)
- void gslc_InitInputMap (gslc_tsGui ∗pGui, gslc_tsInputMap ∗asInputMap, uint8_t nInputMapMax)
- void gslc_InputMapAdd (gslc_tsGui ∗pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc_te←Action eAction, int16_t nActionVal)

7.11.1 Detailed Description

Functions that handle GPIO / pin and keyboard input.

7.11.2 Function Documentation

7.11.2.1 void gslc_InitInputMap (**gslc_tsGui** ∗ *pGui,* **gslc_tsInputMap** ∗ *asInputMap,* uint8_t *nInputMapMax*)

Todo Doc. This API is experimental and subject to change

7.11.2.2 void gslc_InputMapAdd (**gslc_tsGui** ∗ *pGui,* **gslc_teInputRawEvent** *eInputEvent,* int16_t *nInputVal,* **gslc_teAction** *eAction,* int16_t *nActionVal*)

Todo Doc. This API is experimental and subject to change

7.11.2.3 void gslc_SetPinPollFunc (**gslc_tsGui** ∗ *pGui,* **GSLC_CB_PIN_POLL** *pfunc*)

Todo Doc. This API is experimental and subject to change

7.12 General Purpose Macros

Macros that are used throughout the GUI for debug.

Macros

- #define GSLC_DEBUG_PRINT(sFmt, ...)

  Macro to enable optional debug output.
- #define GSLC_DEBUG2_PRINT(sFmt, ...)
- #define GSLC_DEBUG_PRINT_CONST(sFmt, ...)
- #define GSLC_DEBUG2_PRINT_CONST(sFmt, ...)

7.12.1 Detailed Description

Macros that are used throughout the GUI for debug.

7.12.2 Macro Definition Documentation

7.12.2.1 #define GSLC_DEBUG2_PRINT(*sFmt, ...*)

7.12.2.2 #define GSLC_DEBUG2_PRINT_CONST(*sFmt, ...*)

7.12.2.3 #define GSLC_DEBUG_PRINT(*sFmt, ...*)

Macro to enable optional debug output.

- Supports printf formatting via gslc_DebugPrintf()
- Supports storing the format string in PROGMEM
- Note that at least one variable argument must be provided to the macro after the format string. This is a limitation of the macro definition. If no parameters are needed, then simply pass 0. For example: GSLC_D←EBUG_PRINT("Loaded OK",0);

Parameters

in	*sFmt*	Format string for debug message

7.12.2.4 #define GSLC_DEBUG_PRINT_CONST(*sFmt, ...*)

7.13 Flash-based Element Macros

Macros that represent element creation routines based in FLASH memory.

Macros

- #define gslc_ElemCreateTxt_P(pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, pFont, colTxt, colFrame, col↩ Fill, nAlignTxt, bFrameEn, bFillEn)

  Create a read-only text element.
- #define gslc_ElemCreateTxt_P_R(pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, strLength, pFont, colTxt, colFrame, colFill, nAlignTxt, bFrameEn, bFillEn)

  Create a read-write text element (element in Flash, string in RAM)
- #define gslc_ElemCreateBox_P(pGui, nElemId, nPage, nX, nY, nW, nH, colFrame, colFill, bFrameEn, bFillEn, pfuncXDraw, pfuncXTick)

  Create a read-only box element.
- #define gslc_ElemCreateLine_P(pGui, nElemId, nPage, nX0, nY0, nX1, nY1, colFill)

  Create a read-only line element.
- #define gslc_ElemCreateBtnTxt_P(pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, pFont, colTxt, colFrame, colFill, colFrameGlow, colFillGlow, nAlignTxt, bFrameEn, bFillEn, callFunc, extraData)

  Create a text button element.

7.13.1 Detailed Description

Macros that represent element creation routines based in FLASH memory.

7.13.2 Macro Definition Documentation

7.13.2.1 #define gslc_ElemCreateBox_P(*pGui, nElemId, nPage, nX, nY, nW, nH, colFrame, colFill, bFrameEn, bFillEn, pfuncXDraw, pfuncXTick*)

Create a read-only box element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Unique element ID to assign
in	*nPage*	Page ID to attach element to
in	*nX*	X coordinate of element
in	*nY*	Y coordinate of element
in	*nW*	Width of element
in	*nH*	Height of element
in	*colFrame*	Color for the frame
in	*colFill*	Color for the fill
in	*bFrameEn*	True if framed, false otherwise
in	*bFillEn*	True if filled, false otherwise
in	*pfuncXDraw*	Pointer to custom draw callback (or NULL if default)
in	*pfuncXTick*	Pointer to custom tick callback (or NULL if default)

7.13.2.2 #define gslc_ElemCreateBtnTxt_P(*pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, pFont, colTxt, colFrame, colFill, colFrameGlow, colFillGlow, nAlignTxt, bFrameEn, bFillEn, callFunc, extraData*)

Create a text button element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Unique element ID to assign
in	*nPage*	Page ID to attach element to
in	*nX*	X coordinate of element
in	*nY*	Y coordinate of element
in	*nW*	Width of element
in	*nH*	Height of element
in	*strTxt*	Text string to display
in	*pFont*	Pointer to font resource
in	*colTxt*	Color for the text
in	*colFrame*	Color for the frame
in	*colFill*	Color for the fill
in	*colFrameGlow*	Color for the frame when glowing
in	*colFillGlow*	Color for the fill when glowing
in	*nAlignTxt*	Text alignment
in	*bFrameEn*	True if framed, false otherwise
in	*bFillEn*	True if filled, false otherwise
in	*callFunc*	Callback function for button press
in	*extraData*	Ptr to extended data structure

7.13.2.3 #define gslc_ElemCreateLine_P(*pGui, nElemId, nPage, nX0, nY0, nX1, nY1, colFill*)

Create a read-only line element.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ ElemId*	Unique element ID to assign
in	*nPage*	Page ID to attach element to
in	*nX0*	X coordinate of line start
in	*nY0*	Y coordinate of line start
in	*nX1*	X coordinate of line end
in	*nY1*	Y coordinate of line end
in	*colFill*	Color for the line

7.13.2.4 #define gslc_ElemCreateTxt_P(*pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, pFont, colTxt, colFrame, colFill, nAlignTxt, bFrameEn, bFillEn*)

Create a read-only text element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Unique element ID to assign
in	*nPage*	Page ID to attach element to
in	*nX*	X coordinate of element
in	*nY*	Y coordinate of element
in	*nW*	Width of element
in	*nH*	Height of element
in	*strTxt*	Text string to display
in	*pFont*	Pointer to font resource
in	*colTxt*	Color for the text
in	*colFrame*	Color for the frame
in	*colFill*	Color for the fill
in	*nAlignTxt*	Text alignment
in	*bFrameEn*	True if framed, false otherwise
in	*bFillEn*	True if filled, false otherwise

7.13.2.5 #define gslc_ElemCreateTxt_P_R(*pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, strLength, pFont, colTxt, colFrame, colFill, nAlignTxt, bFrameEn, bFillEn*)

Create a read-write text element (element in Flash, string in RAM)

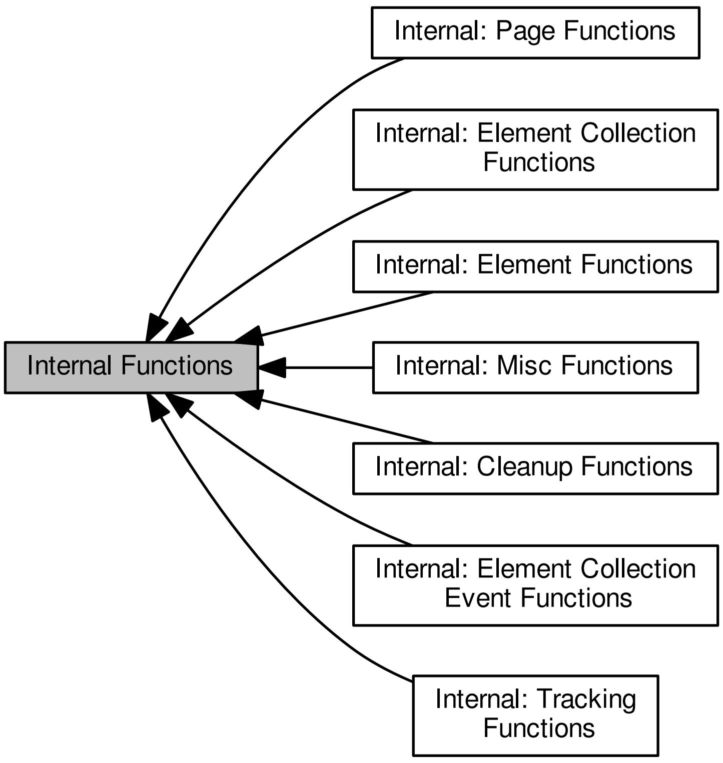
Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Unique element ID to assign
in	*nPage*	Page ID to attach element to
in	*nX*	X coordinate of element
in	*nY*	Y coordinate of element
in	*nW*	Width of element
in	*nH*	Height of element
in	*strTxt*	Text string to display
in	*strLength*	Length of text string
in	*pFont*	Pointer to font resource
in	*colTxt*	Color for the text
in	*colFrame*	Color for the frame
in	*colFill*	Color for the fill
in	*nAlignTxt*	Text alignment
in	*bFrameEn*	True if framed, false otherwise
in	*bFillEn*	True if filled, false otherwise

7.14 Internal Functions

These functions are internal to the GUIslice implementation and are not intended to be called by user code and subject to change even in minor releases.

Collaboration diagram for Internal Functions:


Internal: Page Functions
Internal: Element Collection Functions
Internal: Element Functions
Internal Functions
Internal: Misc Functions
Internal: Cleanup Functions
Internal: Element Collection Event Functions
Internal: Tracking Functions

Modules

- Internal: Misc Functions
- Internal: Element Functions
- Internal: Page Functions
- Internal: Element Collection Functions
- Internal: Element Collection Event Functions
- Internal: Tracking Functions
- Internal: Cleanup Functions

Variables

- int16_t gslc_tsRect::x

  X coordinate of corner.
- int16_t gslc_tsRect::y

Y coordinate of corner.
- uint16_t gslc_tsRect::w

  Width of region.
- uint16_t gslc_tsRect::h

  Height of region.
- int16_t gslc_tsPt::x

  X coordinate.
- int16_t gslc_tsPt::y

  Y coordinate.
- uint8_t gslc_tsColor::r

  RGB red value.
- uint8_t gslc_tsColor::g

  RGB green value.
- uint8_t gslc_tsColor::b

  RGB blue value.
- gslc_teEventType gslc_tsEvent::eType

  Event type.
- uint8_t gslc_tsEvent::nSubType

  Event sub-type.
- void ∗ gslc_tsEvent::pvScope

  Event target scope (eg. Page,Collection,Event)
- void ∗ gslc_tsEvent::pvData

  Generic data pointer for event.
- gslc_teTouch gslc_tsEventTouch::eTouch

  Touch state.
- int16_t gslc_tsEventTouch::nX

  Touch X coordinate (or param1)
- int16_t gslc_tsEventTouch::nY

  Touch Y coordinate (or param2)
- int16_t gslc_tsFont::nId

  Font ID specified by user.
- gslc_teFontRefType gslc_tsFont::eFontRefType

  Font reference type.
- gslc_teFontRefMode gslc_tsFont::eFontRefMode

  Font reference mode.
- const void ∗ gslc_tsFont::pvFont

  Void ptr to the font reference (type defined by driver)
- uint16_t gslc_tsFont::nSize

  Font size.
- const unsigned char ∗ gslc_tsImgRef::pImgBuf

  Pointer to input image buffer in memory [RAM,FLASH].
- const char ∗ gslc_tsImgRef::pFname

  Pathname to input image file [FILE,SD].
- gslc_teImgRefFlags gslc_tsImgRef::eImgFlags

  Image reference flags.
- void ∗ gslc_tsImgRef::pvImgRaw

  Ptr to raw output image data (for pre-loaded images)
- gslc_tsElem ∗ gslc_tsElemRef::pElem

  Pointer to element in memory [RAM,FLASH].
- gslc_teElemRefFlags gslc_tsElemRef::eElemFlags

  Element reference flags.

- int16_t gslc_tsElem::nId

    Element ID specified by user.
- uint8_t gslc_tsElem::nFeatures

    Element feature vector (appearance/behavior))
- int16_t gslc_tsElem::nType

    Element type enumeration.
- gslc_tsRect gslc_tsElem::rElem

    Rect region containing element.
- int16_t gslc_tsElem::nGroup

    Group ID that the element belongs to.
- gslc_tsColor gslc_tsElem::colElemFrame

    Color for frame.
- gslc_tsColor gslc_tsElem::colElemFill

    Color for background fill.
- gslc_tsColor gslc_tsElem::colElemFrameGlow

    Color to use for frame when glowing.
- gslc_tsColor gslc_tsElem::colElemFillGlow

    Color to use for fill when glowing.
- gslc_tsImgRef gslc_tsElem::sImgRefNorm

    Image reference to draw (normal)
- gslc_tsImgRef gslc_tsElem::sImgRefGlow

    Image reference to draw (glowing)
- gslc_tsElemRef ∗ gslc_tsElem::pElemRefParent

    Parent element reference.
- char ∗ gslc_tsElem::pStrBuf

    Ptr to text string buffer to overlay.
- uint8_t gslc_tsElem::nStrBufMax

    Size of string buffer.
- gslc_teTxtFlags gslc_tsElem::eTxtFlags

    Flags associated with text buffer.
- gslc_tsColor gslc_tsElem::colElemText

    Color of overlay text.
- gslc_tsColor gslc_tsElem::colElemTextGlow

    Color of overlay text when glowing.
- int8_t gslc_tsElem::eTxtAlign

    Alignment of overlay text.
- uint8_t gslc_tsElem::nTxtMargin

    Margin of overlay text within rect region.
- gslc_tsFont ∗ gslc_tsElem::pTxtFont

    Ptr to Font for overlay text.
- void ∗ gslc_tsElem::pXData

    Ptr to extended data structure.
- GSLC_CB_EVENT gslc_tsElem::pfuncXEvent

    UNUSED: Callback func ptr for event tree (draw,touch,tick)
- GSLC_CB_DRAW gslc_tsElem::pfuncXDraw

    Callback func ptr for custom drawing.
- GSLC_CB_TOUCH gslc_tsElem::pfuncXTouch

    Callback func ptr for touch.
- GSLC_CB_TICK gslc_tsElem::pfuncXTick

    Callback func ptr for timer/main loop tick.
- gslc_tsElem ∗ gslc_tsCollect::asElem

Array of elements.
- uint16_t gslc_tsCollect::nElemMax

  Maximum number of elements to allocate (in RAM)
- uint16_t gslc_tsCollect::nElemCnt

  Number of elements allocated.
- int16_t gslc_tsCollect::nElemAutoIdNext

  Next Element ID for auto-assignment.
- gslc_tsElemRef ∗ gslc_tsCollect::asElemRef

  Array of element references.
- uint16_t gslc_tsCollect::nElemRefMax

  Maximum number of element references to allocate.
- uint16_t gslc_tsCollect::nElemRefCnt

  Number of element references allocated.
- gslc_tsElemRef ∗ gslc_tsCollect::pElemRefTracked

  Element reference currently being touch-tracked (NULL for none)
- int16_t gslc_tsCollect::nElemIndFocused

  Element index currently in focus (eg. by keyboard/pin control), GSLC_IND_NONE for none.
- gslc_tsCollect gslc_tsPage::sCollect

  Collection of elements on page.
- int16_t gslc_tsPage::nPageId

  Page identifier.
- gslc_tsRect gslc_tsPage::rBounds

  Bounding rect for page elements.
- gslc_teInputRawEvent gslc_tsInputMap::eEvent

  The input event.
- int16_t gslc_tsInputMap::nVal

  The value associated with the input event.
- gslc_teAction gslc_tsInputMap::eAction

  Resulting action.
- int16_t gslc_tsInputMap::nActionVal

  The value for the output action.
- uint16_t gslc_tsGui::nDispW

  Width of the display (pixels)
- uint16_t gslc_tsGui::nDispH

  Height of the display (pixels)
- uint16_t gslc_tsGui::nDisp0W

  Width of the display (pixels) in native orientation.
- uint16_t gslc_tsGui::nDisp0H

  Height of the display (pixels) in native orientation.
- uint8_t gslc_tsGui::nDispDepth

  Bit depth of display (bits per pixel)
- uint8_t gslc_tsGui::nRotation

  Adafruit GFX Rotation of display.
- uint8_t gslc_tsGui::nTouchRotation

  Touchscreen rotation offset vs display.
- uint8_t gslc_tsGui::nSwapXY

  Adafruit GFX Touch Swap x and y axes.
- uint8_t gslc_tsGui::nFlipX

  Adafruit GFX Touch Flip x axis.
- uint8_t gslc_tsGui::nFlipY

  Adafruit GFX Touch Flip x axis.

- uint16_t gslc_tsGui::nTouchCalXMin

  Calibration X minimum reading.
- uint16_t gslc_tsGui::nTouchCalXMax

  Calibration X maximum reading.
- uint16_t gslc_tsGui::nTouchCalYMin

  Calibration Y minimum reading.
- uint16_t gslc_tsGui::nTouchCalYMax

  Calibration Y maximum reading.
- gslc_tsFont ∗ gslc_tsGui::asFont

  Collection of loaded fonts.
- uint8_t gslc_tsGui::nFontMax

  Maximum number of fonts to allocate.
- uint8_t gslc_tsGui::nFontCnt

  Number of fonts allocated.
- uint8_t gslc_tsGui::nRoundRadius

  Radius for rounded elements.
- gslc_tsElem gslc_tsGui::sElemTmpProg

  Temporary element for Flash compatibility.
- gslc_teInitStat gslc_tsGui::eInitStatTouch

  Status of touch initialization.
- int16_t gslc_tsGui::nTouchLastX

  Last touch event X coord.
- int16_t gslc_tsGui::nTouchLastY

  Last touch event Y coord.
- uint16_t gslc_tsGui::nTouchLastPress

  Last touch event pressure (0=none))
- bool gslc_tsGui::bTouchRemapEn

  Enable touch remapping?
- bool gslc_tsGui::bTouchRemapYX

  Enable touch controller swapping of X & Y.
- void ∗ gslc_tsGui::pvDriver

  Driver-specific members (gslc_tsDriver∗)
- bool gslc_tsGui::bRedrawPartialEn

  Driver supports partial page redraw.
- gslc_tsImgRef gslc_tsGui::sImgRefBkgnd

  Image reference for background.
- uint8_t gslc_tsGui::nFrameRateCnt

  Diagnostic frame rate count.
- uint8_t gslc_tsGui::nFrameRateStart

  Diagnostic frame rate timestamp.
- gslc_tsPage ∗ gslc_tsGui::asPage

  Array of all pages defined in system.
- uint8_t gslc_tsGui::nPageMax

  Maximum number of pages that can be defined.
- uint8_t gslc_tsGui::nPageCnt

  Current number of pages defined.
- gslc_tsPage ∗ gslc_tsGui::apPageStack [GSLC_STACK__MAX]

  Stack of pages.
- bool gslc_tsGui::abPageStackActive [GSLC_STACK__MAX]

  Whether page in stack can receive touch events.
- bool gslc_tsGui::abPageStackDoDraw [GSLC_STACK__MAX]

Whether page in stack is still actively drawn.
- bool gslc_tsGui::bScreenNeedRedraw

  Screen requires a redraw.
- bool gslc_tsGui::bScreenNeedFlip

  Screen requires a page flip.
- bool gslc_tsGui::bInvalidateEn

  A region of the display has been invalidated.
- gslc_tsRect gslc_tsGui::rInvalidateRect

  The rect region that has been invalidated.
- GSLC_CB_PIN_POLL gslc_tsGui::pfuncPinPoll

  Callback func ptr for pin polling.
- gslc_tsInputMap ∗ gslc_tsGui::asInputMap

  Array of input maps.
- uint8_t gslc_tsGui::nInputMapMax

  Maximum number of input maps.
- uint8_t gslc_tsGui::nInputMapCnt

  Current number of input maps.

7.14.1 Detailed Description

These functions are internal to the GUIslice implementation and are not intended to be called by user code and subject to change even in minor releases.

- The following functions are generally not required for typical users of GUIslice. However, for advanced usage more direct access may be required.

7.14.2 Variable Documentation

7.14.2.1 bool gslc_tsGui::abPageStackActive[GSLC_STACK__MAX]

Whether page in stack can receive touch events.

7.14.2.2 bool gslc_tsGui::abPageStackDoDraw[GSLC_STACK__MAX]

Whether page in stack is still actively drawn.

7.14.2.3 gslc_tsPage∗ gslc_tsGui::apPageStack[GSLC_STACK__MAX]

Stack of pages.

7.14.2.4 gslc_tsElem∗ gslc_tsCollect::asElem

Array of elements.

7.14.2.5 gslc_tsElemRef∗ gslc_tsCollect::asElemRef

Array of element references.

7.14.2.6 gslc_tsFont∗ gslc_tsGui::asFont

Collection of loaded fonts.

7.14.2.7 gslc_tsInputMap∗ gslc_tsGui::asInputMap

Array of input maps.

7.14.2.8 gslc_tsPage∗ gslc_tsGui::asPage

Array of all pages defined in system.

7.14.2.9 uint8_t gslc_tsColor::b

RGB blue value.

7.14.2.10 bool gslc_tsGui::bInvalidateEn

A region of the display has been invalidated.

7.14.2.11 bool gslc_tsGui::bRedrawPartialEn

Driver supports partial page redraw.

If true, only changed elements are redrawn during next page redraw command. If false, entire page is redrawn when any element has been updated prior to next page redraw command.

7.14.2.12 bool gslc_tsGui::bScreenNeedFlip

Screen requires a page flip.

7.14.2.13 bool gslc_tsGui::bScreenNeedRedraw

Screen requires a redraw.

7.14.2.14 bool gslc_tsGui::bTouchRemapEn

Enable touch remapping?

7.14.2.15 bool gslc_tsGui::bTouchRemapYX

Enable touch controller swapping of X & Y.

7.14.2.16 gslc_tsColor gslc_tsElem::colElemFill

Color for background fill.

7.14.2.17 gslc_tsColor gslc_tsElem::colElemFillGlow

Color to use for fill when glowing.

7.14.2.18 gslc_tsColor gslc_tsElem::colElemFrame

Color for frame.

7.14.2.19 gslc_tsColor gslc_tsElem::colElemFrameGlow

Color to use for frame when glowing.

7.14.2.20 gslc_tsColor gslc_tsElem::colElemText

Color of overlay text.

7.14.2.21 gslc_tsColor gslc_tsElem::colElemTextGlow

Color of overlay text when glowing.

7.14.2.22 gslc_teAction gslc_tsInputMap::eAction

Resulting action.

7.14.2.23 gslc_teElemRefFlags gslc_tsElemRef::eElemFlags

Element reference flags.

7.14.2.24 gslc_teInputRawEvent gslc_tsInputMap::eEvent

The input event.

7.14.2.25 gslc_teFontRefMode gslc_tsFont::eFontRefMode

Font reference mode.

7.14.2.26 gslc_teFontRefType gslc_tsFont::eFontRefType

Font reference type.

7.14.2.27 gslc_teImgRefFlags gslc_tsImgRef::eImgFlags

Image reference flags.

7.14.2.28 gslc_teInitStat gslc_tsGui::eInitStatTouch

Status of touch initialization.

7.14.2.29 gslc_teTouch gslc_tsEventTouch::eTouch

Touch state.

7.14.2.30 int8_t gslc_tsElem::eTxtAlign

Alignment of overlay text.

7.14.2.31 gslc_teTxtFlags gslc_tsElem::eTxtFlags

Flags associated with text buffer.

7.14.2.32 gslc_teEventType gslc_tsEvent::eType

Event type.

7.14.2.33 uint8_t gslc_tsColor::g

RGB green value.

7.14.2.34 uint16_t gslc_tsRect::h

Height of region.

7.14.2.35 int16_t gslc_tsInputMap::nActionVal

The value for the output action.

7.14.2.36 uint16_t gslc_tsGui::nDisp0H

Height of the display (pixels) in native orientation.

7.14.2.37 uint16_t gslc_tsGui::nDisp0W

Width of the display (pixels) in native orientation.

7.14.2.38 uint8_t gslc_tsGui::nDispDepth

Bit depth of display (bits per pixel)

7.14.2.39 uint16_t gslc_tsGui::nDispH

Height of the display (pixels)

7.14.2.40 uint16_t gslc_tsGui::nDispW

Width of the display (pixels)

7.14.2.41 int16_t gslc_tsCollect::nElemAutoIdNext

Next Element ID for auto-assignment.

7.14.2.42 uint16_t gslc_tsCollect::nElemCnt

Number of elements allocated.

7.14.2.43 int16_t gslc_tsCollect::nElemIndFocused

Element index currently in focus (eg. by keyboard/pin control), GSLC_IND_NONE for none.

7.14.2.44 uint16_t gslc_tsCollect::nElemMax

Maximum number of elements to allocate (in RAM)

7.14.2.45 uint16_t gslc_tsCollect::nElemRefCnt

Number of element references allocated.

7.14.2.46 uint16_t gslc_tsCollect::nElemRefMax

Maximum number of element references to allocate.

7.14.2.47 uint8_t gslc_tsElem::nFeatures

Element feature vector (appearance/behavior))

7.14.2.48 uint8_t gslc_tsGui::nFlipX

Adafruit GFX Touch Flip x axis.

7.14.2.49 uint8_t gslc_tsGui::nFlipY

Adafruit GFX Touch Flip x axis.

7.14.2.50 uint8_t gslc_tsGui::nFontCnt

Number of fonts allocated.

7.14.2.51 uint8_t gslc_tsGui::nFontMax

Maximum number of fonts to allocate.

7.14.2.52 uint8_t gslc_tsGui::nFrameRateCnt

Diagnostic frame rate count.

7.14.2.53 uint8_t gslc_tsGui::nFrameRateStart

Diagnostic frame rate timestamp.

7.14.2.54 int16_t gslc_tsElem::nGroup

Group ID that the element belongs to.

7.14.2.55 int16_t gslc_tsFont::nId

Font ID specified by user.

7.14.2.56 int16_t gslc_tsElem::nId

Element ID specified by user.

7.14.2.57 uint8_t gslc_tsGui::nInputMapCnt

Current number of input maps.

7.14.2.58 uint8_t gslc_tsGui::nInputMapMax

Maximum number of input maps.

7.14.2.59 uint8_t gslc_tsGui::nPageCnt

Current number of pages defined.

7.14.2.60 int16_t gslc_tsPage::nPageId

Page identifier.

7.14.2.61 uint8_t gslc_tsGui::nPageMax

Maximum number of pages that can be defined.

7.14.2.62 uint8_t gslc_tsGui::nRotation

Adafruit GFX Rotation of display.

7.14.2.63 uint8_t gslc_tsGui::nRoundRadius

Radius for rounded elements.

7.14.2.64 uint16_t gslc_tsFont::nSize

Font size.

7.14.2.65 uint8_t gslc_tsElem::nStrBufMax

Size of string buffer.

7.14.2.66 uint8_t gslc_tsEvent::nSubType

Event sub-type.

7.14.2.67 uint8_t gslc_tsGui::nSwapXY

Adafruit GFX Touch Swap x and y axes.

7.14.2.68 uint16_t gslc_tsGui::nTouchCalXMax

Calibration X maximum reading.

7.14.2.69 uint16_t gslc_tsGui::nTouchCalXMin

Calibration X minimum reading.

7.14.2.70 uint16_t gslc_tsGui::nTouchCalYMax

Calibration Y maximum reading.

7.14.2.71 uint16_t gslc_tsGui::nTouchCalYMin

Calibration Y minimum reading.

7.14.2.72 uint16_t gslc_tsGui::nTouchLastPress

Last touch event pressure (0=none))

7.14.2.73 int16_t gslc_tsGui::nTouchLastX

Last touch event X coord.

7.14.2.74 int16_t gslc_tsGui::nTouchLastY

Last touch event Y coord.

7.14.2.75 uint8_t gslc_tsGui::nTouchRotation

Touchscreen rotation offset vs display.

7.14.2.76 uint8_t gslc_tsElem::nTxtMargin

Margin of overlay text within rect region.

7.14.2.77 int16_t gslc_tsElem::nType

Element type enumeration.

7.14.2.78 int16_t gslc_tsInputMap::nVal

The value associated with the input event.

7.14.2.79 int16_t gslc_tsEventTouch::nX

Touch X coordinate (or param1)

7.14.2.80 int16_t gslc_tsEventTouch::nY

Touch Y coordinate (or param2)

7.14.2.81 gslc_tsElem∗ gslc_tsElemRef::pElem

Pointer to element in memory [RAM,FLASH].

7.14.2.82 gslc_tsElemRef∗ gslc_tsElem::pElemRefParent

Parent element reference.

Used during redraw to notify parent elements that they require redraw as well. Primary usage is in compound elements. NOTE: Although this field is only used in GLSC_COMPOUND mode, it is not wrapped in an ifdef because the ElemCreate∗_P() function macros currently initialize this field.

7.14.2.83 gslc_tsElemRef∗ gslc_tsCollect::pElemRefTracked

Element reference currently being touch-tracked (NULL for none)

7.14.2.84 const char∗ gslc_tsImgRef::pFname

Pathname to input image file [FILE,SD].

7.14.2.85 GSLC_CB_PIN_POLL gslc_tsGui::pfuncPinPoll

Callback func ptr for pin polling.

7.14.2.86 GSLC_CB_DRAW gslc_tsElem::pfuncXDraw

Callback func ptr for custom drawing.

7.14.2.87 GSLC_CB_EVENT gslc_tsElem::pfuncXEvent

UNUSED: Callback func ptr for event tree (draw,touch,tick)

7.14.2.88 GSLC_CB_TICK gslc_tsElem::pfuncXTick

Callback func ptr for timer/main loop tick.

7.14.2.89 GSLC_CB_TOUCH gslc_tsElem::pfuncXTouch

Callback func ptr for touch.

7.14.2.90 const unsigned char∗ gslc_tsImgRef::pImgBuf

Pointer to input image buffer in memory [RAM,FLASH].

7.14.2.91 char∗ gslc_tsElem::pStrBuf

Ptr to text string buffer to overlay.

7.14.2.92 gslc_tsFont∗ gslc_tsElem::pTxtFont

Ptr to Font for overlay text.

7.14.2.93 void∗ gslc_tsEvent::pvData

Generic data pointer for event.

This member is used to either pass a pointer to a simple data datatype (such as Element or Collection) or to a another structure that contains multiple fields.

7.14.2.94 void∗ gslc_tsGui::pvDriver

Driver-specific members (gslc_tsDriver∗)

7.14.2.95 const void∗ gslc_tsFont::pvFont

Void ptr to the font reference (type defined by driver)

7.14.2.96 void∗ gslc_tsImgRef::pvImgRaw

Ptr to raw output image data (for pre-loaded images)

7.14.2.97 void∗ gslc_tsEvent::pvScope

Event target scope (eg. Page,Collection,Event)

7.14.2.98 void∗ gslc_tsElem::pXData

Ptr to extended data structure.

7.14.2.99 uint8_t gslc_tsColor::r

RGB red value.

7.14.2.100 gslc_tsRect gslc_tsPage::rBounds

Bounding rect for page elements.

7.14.2.101 gslc_tsRect gslc_tsElem::rElem

Rect region containing element.

7.14.2.102 gslc_tsRect gslc_tsGui::rInvalidateRect

The rect region that has been invalidated.

7.14.2.103 gslc_tsCollect gslc_tsPage::sCollect

Collection of elements on page.

7.14.2.104 gslc_tsElem gslc_tsGui::sElemTmpProg

Temporary element for Flash compatibility.

7.14.2.105 gslc_tsImgRef gslc_tsGui::sImgRefBkgnd

Image reference for background.

7.14.2.106 gslc_tsImgRef gslc_tsElem::sImgRefGlow

Image reference to draw (glowing)

7.14.2.107 gslc_tsImgRef gslc_tsElem::sImgRefNorm

Image reference to draw (normal)

7.14.2.108 uint16_t gslc_tsRect::w

Width of region.

7.14.2.109 int16_t gslc_tsRect::x

X coordinate of corner.

7.14.2.110 int16_t gslc_tsPt::x

X coordinate.

7.14.2.111 int16_t gslc_tsRect::y

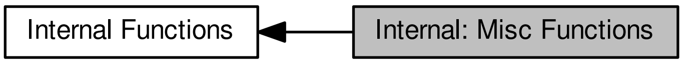
Y coordinate of corner.

7.14.2.112 int16_t gslc_tsPt::y

Y coordinate.

7.15 Internal: Misc Functions

Collaboration diagram for Internal: Misc Functions:



Functions

- gslc_tsImgRef gslc_ResetImage ()

  Create a blank image reference structure.

7.15.1 Detailed Description

7.15.2 Function Documentation

7.15.2.1 gslc_tsImgRef gslc_ResetImage ( )

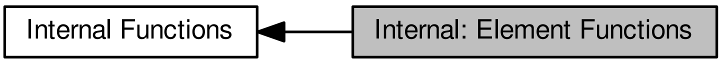
Create a blank image reference structure.

Returns

Image reference struct

7.16 Internal: Element Functions

Collaboration diagram for Internal: Element Functions:



Functions

- gslc_tsElem gslc_ElemCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPageId, int16_t nType, gslc_ts←Rect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

  Create a new element with default styling.
- gslc_tsElemRef ∗ gslc_ElemAdd (gslc_tsGui ∗pGui, int16_t nPageId, gslc_tsElem ∗pElem, gslc_teElem←RefFlags eFlags)

  Add the Element to the list of generated elements in the GUI environment.
- uint8_t gslc_GetElemRefFlag (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nFlagMask)

  Get the flags associated with an element reference.
- void gslc_SetElemRefFlag (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nFlagMask, uint8_t n←FlagVal)

  Set the flags associated with an element reference.
- gslc_tsElem ∗ gslc_GetElemFromRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Returns a pointer to an element from an element reference, copying from FLASH to RAM if element is stored in PROGMEM.
- gslc_tsElem ∗ gslc_GetElemFromRefD (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nLineNum)

  Returns a pointer to an element from an element reference.
- void ∗ gslc_GetXDataFromRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nType, int16_t nLine←Num)

  Returns a pointer to the data structure associated with an extended element.
- void gslc_ElemSetImage (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsImgRef sImgRef, gslc_ts←ImgRef sImgRefSel)

  Set an element to use a bitmap image.
- bool gslc_ElemDrawByRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teRedrawType eRedraw)

  Draw an element to the active display.
- void gslc_ElemDraw (gslc_tsGui ∗pGui, int16_t nPageId, int16_t nElemId)

  Draw an element to the active display.
- void gslc_DrawTxtBase (gslc_tsGui ∗pGui, char ∗pStrBuf, gslc_tsRect rTxt, gslc_tsFont ∗pTxtFont, gslc←_teTxtFlags eTxtFlags, int8_t eTxtAlign, gslc_tsColor colTxt, gslc_tsColor colBg, int16_t nMarginW, int16_t nMarginH)

  Draw text with full text justification.
- void gslc_SetRoundRadius (gslc_tsGui ∗pGui, uint8_t nRadius)

  Set the global rounded radius.

7.16.1 Detailed Description

7.16.2 Function Documentation

7.16.2.1 void gslc_DrawTxtBase (gslc_tsGui ∗ *pGui,* char ∗ *pStrBuf,* gslc_tsRect *rTxt,* gslc_tsFont ∗ *pTxtFont,* gslc_teTxtFlags *eTxtFlags,* int8_t *eTxtAlign,* gslc_tsColor *colTxt,* gslc_tsColor *colBg,* int16_t *nMarginW,* int16_t *nMarginH*)

Draw text with full text justification.

- This function is usually only required by internal GUIslice rendering operations but is made available for custom element usage as well

Parameters

in	*pGui*	Pointer to GUI
in	*pStrBuf*	Pointer to text string buffer
in	*rTxt*	Rectangle region to contain the text
in	*pTxtFont*	Pointer to the font
in	*eTxtFlags*	Text string attributes
in	*eTxtAlign*	Text alignment / justification mode
in	*colTxt*	Text foreground color
in	*colBg*	Text background color
in	*nMarginW*	Horizontal margin within rect region to keep text away
in	*nMarginH*	Vertical margin within rect region to keep text away

Returns

none

7.16.2.2 gslc_tsElemRef∗ gslc_ElemAdd (gslc_tsGui ∗ *pGui,* int16_t *nPageId,* gslc_tsElem ∗ *pElem,* gslc_teElemRefFlags *eFlags*)

Add the Element to the list of generated elements in the GUI environment.

- NOTE: The content of pElem is copied so the pointer can be released after the call.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ PageId*	Page ID to add element to (GSLC_PAGE_NONE to skip in case of temporary creation for compound elements)
in	*pElem*	Pointer to Element to add
in	*eFlags*	Flags describing the element (eg. whether the element should be stored in internal RAM array or is located in Flash/PROGMEM).

Returns

Pointer to Element reference or NULL if fail

7.16.2.3 gslc_tsElem gslc_ElemCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPageId,* int16_t *nType,* gslc_tsRect *rElem,* char ∗ *pStrBuf,* uint8_t *nStrBufMax,* int16_t *nFontId*)

Create a new element with default styling.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	User-supplied ID for referencing this element (or GSLC_ID_AUTO to auto-generate)
in	*nPageId*	The page ID on which this page should be associated
in	*nType*	Enumeration that indicates the type of element that is requested for creation. The type adjusts the visual representation and default styling.
in	*rElem*	Rectangle region framing the element
in	*pStrBuf*	String to copy into element
in	*nStrBufMax*	Maximum length of string buffer (pStrBuf). Only applicable if GSLC_LOCAL_STR=0. Ignored if GSLC_LOCAL_STR=1.)
in	*nFontId*	Font ID for textual elements

Returns

Initialized structure

7.16.2.4 void gslc_ElemDraw (gslc_tsGui ∗ *pGui,* int16_t *nPageId,* int16_t *nElemId*)

Draw an element to the active display.

- Element is referenced by a page ID and element ID

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ PageId*	ID of page containing element
in	*n↩ ElemId*	ID of element

Returns

none

Todo Unused?

7.16.2.5 bool gslc_ElemDrawByRef (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_teRedrawType *eRedraw*)

Draw an element to the active display.

- Element is referenced by an element pointer

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Ptr to Element reference to draw
`in`	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

7.16.2.6 void gslc_ElemSetImage (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_tsImgRef *sImgRef,* gslc_tsImgRef *sImgRefSel*)

Set an element to use a bitmap image.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference to update
`in`	*sImgRef*	Image reference (normal state)
`in`	*sImgRefSel*	Image reference (glowing state)

Returns

none

7.16.2.7 gslc_tsElem∗ gslc_GetElemFromRef (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Returns a pointer to an element from an element reference, copying from FLASH to RAM if element is stored in PROGMEM.

This function enables all APIs to work with Elements irrespective of whether they were created in RAM or Flash.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element Reference

Returns

Pointer to Element after ensuring that it is accessible from RAM

7.16.2.8 gslc_tsElem∗ gslc_GetElemFromRefD (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nLineNum*)

Returns a pointer to an element from an element reference.

This is a wrapper for GetElemFromRef() including debug checking for invalid pointers.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element Reference
`in`	*nLineNum*	Line number from calling function (ie. **LINE**)

Returns

Pointer to Element after ensuring that it is accessible from RAM

7.16.2.9 uint8_t gslc_GetElemRefFlag (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* uint8_t *nFlagMask*)

Get the flags associated with an element reference.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Element reference pointer
`in`	*nFlagMask*	Flags to read

Returns

Values associated with the element reference flags (subject to the flag mask)

7.16.2.10 void∗ gslc_GetXDataFromRef (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nType,* int16_t *nLineNum*)

Returns a pointer to the data structure associated with an extended element.

- Example usage: gslc_tsXListbox∗ pListbox = (gslc_tsXListbox∗)gslc_GetXDataFromRef(pGui, pElemRef, GSLC_TYPEX_LISTBOX, **LINE**);

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element Reference
`in`	*nType*	Expected type indicator (ie. GSLC_TYPEX_∗)
`in`	*nLineNum*	Line number from calling function (ie. **LINE**)

Returns

Void pointer to extended data (pXData), or NULL if error. Needs to be typecasted accordingly.

7.16.2.11 void gslc_SetElemRefFlag (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, uint8_t *nFlagMask*, uint8_t *nFlagVal*)

Set the flags associated with an element reference.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Element reference pointer
in	*nFlagMask*	Flags to read
in	*nFlagVal*	Values to assign to masked flags

Returns

none

7.16.2.12 void gslc_SetRoundRadius (gslc_tsGui ∗ *pGui*, uint8_t *nRadius*)

Set the global rounded radius.

- Used for rounded rectangles

Parameters

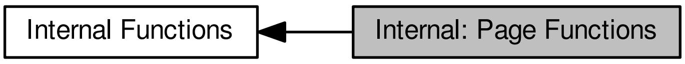
in	*pGui*	Pointer to GUI
in	*nRadius*	Radius for rounded elements

Returns

none

7.17 Internal: Page Functions

Collaboration diagram for Internal: Page Functions:



Functions

- bool gslc_PageEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for a page.
- void gslc_PageRedrawGo (gslc_tsGui ∗pGui)

  Redraw all elements on the active page.
- void gslc_PageFlipSet (gslc_tsGui ∗pGui, bool bNeeded)

  Indicate whether the screen requires page flip.
- bool gslc_PageFlipGet (gslc_tsGui ∗pGui)

  Get state of pending page flip state.
- void gslc_PageFlipGo (gslc_tsGui ∗pGui)

  Update the visible screen if page has been marked for flipping.
- gslc_tsPage ∗ gslc_PageFindById (gslc_tsGui ∗pGui, int16_t nPageId)

  Find a page in the GUI by its ID.
- void gslc_PageRedrawCalc (gslc_tsGui ∗pGui)

  Perform a redraw calculation on the page to determine if additional elements should also be redrawn.
- int16_t gslc_PageFocusStep (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, bool bNext)
- gslc_tsEvent gslc_EventCreate (gslc_tsGui ∗pGui, gslc_teEventType eType, uint8_t nSubType, void ∗pv←Scope, void ∗pvData)

  Create an event structure.
- bool gslc_ElemEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for an element.
- bool gslc_ElemSendEventTouch (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRefTracked, gslc_teTouch e←Touch, int16_t nX, int16_t nY)

  Trigger an element's touch event.

7.17.1 Detailed Description

7.17.2 Function Documentation

7.17.2.1 bool gslc_ElemEvent (void ∗ *pvGui,* gslc_tsEvent *sEvent*)

Common event handler function for an element.

Parameters

in	*pvGui*	Void pointer to GUI
in	*sEvent*	Event data structure

Returns

true if success, false if fail

7.17.2.2 bool gslc_ElemSendEventTouch (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRefTracked*, gslc_teTouch *eTouch*, int16_t *nX*, int16_t *nY*)

Trigger an element's touch event.

This is an optional behavior useful in some extended element types.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRefTracked*	Pointer to tracked Element reference (or NULL for none))
in	*eTouch*	Touch event type
in	*nX*	X coordinate of event (absolute coordinate)
in	*nY*	Y coordinate of event (absolute coordinate)

Returns

true if success, false if error

7.17.2.3 gslc_tsEvent gslc_EventCreate (gslc_tsGui ∗ *pGui*, gslc_teEventType *eType*, uint8_t *nSubType*, void ∗ *pvScope*, void ∗ *pvData*)

Create an event structure.

Parameters

in	*pGui*	Pointer to GUI
in	*eType*	Event type (draw, touch, tick, etc.)
in	*nSubType*	Refinement of event type (or 0 if unused)
in	*pvScope*	Void ptr to object receiving event so that the event handler will have the context
in	*pvData*	Void ptr to additional data associated with the event (eg. coordinates for touch events)

Returns

None

7.17.2.4 bool gslc_PageEvent (void ∗ *pvGui,* gslc_tsEvent *sEvent*)

Common event handler function for a page.

Parameters

in	*pvGui*	Void pointer to GUI
in	*sEvent*	Event data structure

Returns

true if success, false if fail

7.17.2.5 gslc_tsPage∗ gslc_PageFindById (gslc_tsGui ∗ *pGui,* int16_t *nPageId*)

Find a page in the GUI by its ID.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ PageId*	Page ID to search

Returns

Ptr to a page or NULL if none found

7.17.2.6 bool gslc_PageFlipGet (gslc_tsGui ∗ *pGui*)

Get state of pending page flip state.

Parameters

in	*pGui*	Pointer to GUI

Returns

True if screen requires page flip

7.17.2.7 void gslc_PageFlipGo (gslc_tsGui ∗ *pGui*)

Update the visible screen if page has been marked for flipping.

- On some hardware this can trigger a double-buffering page flip.

Parameters

`in`	*pGui*	Pointer to GUI

Returns

None

7.17.2.8 void gslc_PageFlipSet (gslc_tsGui ∗ *pGui,* bool *bNeeded*)

Indicate whether the screen requires page flip.

- This is generally called with bNeeded=true whenever drawing has been done to the active page. Page flip is actually performed later when calling PageFlipGo().

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*bNeeded*	True if screen requires page flip

Returns

None

7.17.2.9 int16_t gslc_PageFocusStep (gslc_tsGui ∗ *pGui,* gslc_tsPage ∗ *pPage,* bool *bNext*)

Todo Doc. This API is experimental and subject to change

7.17.2.10 void gslc_PageRedrawCalc (gslc_tsGui ∗ *pGui*)

Perform a redraw calculation on the page to determine if additional elements should also be redrawn.

This routine checks to see if any transparent elements have been marked as needing redraw. If so, the whole page may be marked as needing redraw (or at least the other elements that have been exposed underneath).

Parameters

`in`	*pGui*	Pointer to GUI

Returns

none

7.17.2.11 void gslc_PageRedrawGo (gslc_tsGui ∗ *pGui*)

Redraw all elements on the active page.

Only the elements that have been marked as needing redraw are rendered unless the entire page has been marked as needing redraw (in which case everything is drawn)

Parameters

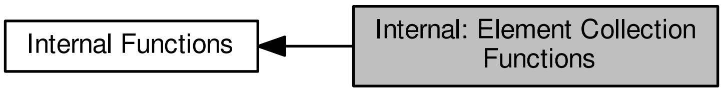
in	*pGui*	Pointer to GUI

Returns

none

7.18 Internal: Element Collection Functions

Collaboration diagram for Internal: Element Collection Functions:



Functions

- void gslc_CollectReset (gslc_tsCollect ∗pCollect, gslc_tsElem ∗asElem, uint16_t nElemMax, gslc_tsElemRef ∗asElemRef, uint16_t nElemRefMax)

  Reset the members of an element collection.
- gslc_tsElemRef ∗ gslc_CollectElemAdd (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, const gslc_tsElem ∗p←Elem, gslc_teElemRefFlags eFlags)

  Add an element to a collection.
- bool gslc_CollectGetRedraw (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Determine if any elements in a collection need redraw.
- gslc_tsElemRef ∗ gslc_CollectFindElemById (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nElemId)

  Find an element in a collection by its Element ID.
- gslc_tsElemRef ∗ gslc_CollectFindElemFromCoord (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nX, int16_t nY)

  Find an element in a collection by a coordinate coordinate.
- int gslc_CollectGetNextId (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Allocate the next available Element ID in a collection.
- gslc_tsElemRef ∗ gslc_CollectGetElemRefTracked (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Get the element within a collection that is currently being tracked.
- void gslc_CollectSetElemTracked (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsElemRef ∗pElemRef)

  Set the element within a collection that is currently being tracked.
- int16_t gslc_CollectGetFocus (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Get the element index within a collection that is currently in focus.
- void gslc_CollectSetFocus (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nElemInd)

  Set the element index within a collection that is currently in focus.
- bool gslc_CollectFindFocusStep (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, bool bNext, bool ∗pbWrapped, int16_t ∗pnElemInd)
- void gslc_CollectSetParent (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsElemRef ∗pElemRefParent)

  Assign the parent element reference to all elements within a collection.

7.18.1 Detailed Description

7.18.2 Function Documentation

7.18.2.1 gslc_tsElemRef∗ gslc_CollectElemAdd (gslc_tsGui ∗ *pGui,* gslc_tsCollect ∗ *pCollect,* const gslc_tsElem ∗ *pElem,* gslc_teElemRefFlags *eFlags*)

Add an element to a collection.

- Note that the contents of pElem are copied to the collection's element array so the pElem pointer can be discarded are the call is complete.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection
in	*pElem*	Ptr to the element to add
in	*eFlags*	Flags describing the element (eg. whether the element should be stored in internal RAM array or is located in Flash/PROGMEM).

Returns

Pointer to the element reference in the collection that has been added or NULL if there was an error

7.18.2.2 gslc_tsElemRef∗ gslc_CollectFindElemById (gslc_tsGui ∗ *pGui,* gslc_tsCollect ∗ *pCollect,* int16_t *nElemId*)

Find an element in a collection by its Element ID.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection
in	*n←→ ElemId*	Element ID to search for

Returns

Pointer to the element reference in the collection that was found or NULL if no matches found

7.18.2.3 gslc_tsElemRef∗ gslc_CollectFindElemFromCoord (gslc_tsGui ∗ *pGui,* gslc_tsCollect ∗ *pCollect,* int16_t *nX,* int16_t *nY*)

Find an element in a collection by a coordinate coordinate.

- A match is found if the element is "clickable" (bClickEn=true) and the coordinate falls within the element's bounds (rElem).

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection
in	*nX*	Absolute X coordinate to use for search
in	*nY*	Absolute Y coordinate to use for search

Returns

Pointer to the element reference in the collection that was found or NULL if no matches found

7.18.2.4 bool gslc_CollectFindFocusStep (**gslc_tsGui** ∗ *pGui,* **gslc_tsCollect** ∗ *pCollect,* bool *bNext,* bool ∗ *pbWrapped,* int16_t ∗ *pnElemInd*)

Todo Doc. This API is experimental and subject to change

7.18.2.5 **gslc_tsElemRef**∗ gslc_CollectGetElemRefTracked (**gslc_tsGui** ∗ *pGui,* **gslc_tsCollect** ∗ *pCollect*)

Get the element within a collection that is currently being tracked.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection

Returns

Pointer to the element reference in the collection that is currently being tracked or NULL if no elements are being tracked

7.18.2.6 int16_t gslc_CollectGetFocus (**gslc_tsGui** ∗ *pGui,* **gslc_tsCollect** ∗ *pCollect*)

Get the element index within a collection that is currently in focus.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection

Returns

Element index or GSLC_IND_NONE for none

7.18.2.7 int gslc_CollectGetNextId (gslc_tsGui ∗ *pGui,* gslc_tsCollect ∗ *pCollect*)

Allocate the next available Element ID in a collection.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection

Returns

Element ID that is reserved for use

7.18.2.8 bool gslc_CollectGetRedraw (gslc_tsGui ∗ *pGui,* gslc_tsCollect ∗ *pCollect*)

Determine if any elements in a collection need redraw.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to Element collection

Returns

True if redraw required, false otherwise

7.18.2.9 void gslc_CollectReset (gslc_tsCollect ∗ *pCollect,* gslc_tsElem ∗ *asElem,* uint16_t *nElemMax,* gslc_tsElemRef ∗ *asElemRef,* uint16_t *nElemRefMax*)

Reset the members of an element collection.

Parameters

in	*pCollect*	Pointer to the collection
in	*asElem*	Internal element array storage to associate with the collection
in	*nElemMax*	Maximum number of elements that can be added to the internal element array (ie. RAM))
in	*asElemRef*	Internal element reference array storage to associate with the collection. All elements, whether they are located in the internal element array or in external Flash (PROGMEM) storage, require an entry in the element reference array.
in	*nElemRefMax*	Maximum number of elements in the reference array. This is effectively the maximum number of elements that can appear in the collection, irrespective of whether it is stored in RAM or Flash (PROGMEM).

Returns

none

7.18.2.10 void gslc_CollectSetElemTracked (gslc_tsGui ∗ *pGui*, gslc_tsCollect ∗ *pCollect*, gslc_tsElemRef ∗ *pElemRef*)

Set the element within a collection that is currently being tracked.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection
in	*pElemRef*	Ptr to element reference to mark as being tracked

Returns

none

7.18.2.11 void gslc_CollectSetFocus (gslc_tsGui ∗ *pGui*, gslc_tsCollect ∗ *pCollect*, int16_t *nElemInd*)

Set the element index within a collection that is currently in focus.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection
in	*nElemInd*	Element index to set in focus, GSLC_IND_NONE for none

Returns

none

7.18.2.12 void gslc_CollectSetParent (gslc_tsGui ∗ *pGui*, gslc_tsCollect ∗ *pCollect*, gslc_tsElemRef ∗ *pElemRefParent*)

Assign the parent element reference to all elements within a collection.

- This is generally used in the case of compound elements where updates to a sub-element should cause the parent (compound element) to be redrawn as well.)

Parameters

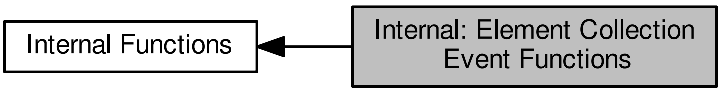
in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to the collection
in	*pElemRefParent*	Ptr to element reference that is the parent

Returns

none

7.19 Internal: Element Collection Event Functions

Collaboration diagram for Internal: Element Collection Event Functions:



Functions

- bool gslc_CollectEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for an element collection.
- void gslc_CollectTouch (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsEventTouch ∗pEventTouch)

  Handle touch events within the element collection.
- bool gslc_CollectTouchCompound (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY, gslc_tsCollect ∗pCollect)

  Handle dispatch of touch (up,down,move) events to compound elements sub elements.
- void gslc_CollectInput (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsEventTouch ∗pEventTouch)

  Handle direct input events within the element collection.

7.19.1 Detailed Description

7.19.2 Function Documentation

7.19.2.1 bool gslc_CollectEvent (void ∗ *pvGui,* gslc_tsEvent *sEvent*)

Common event handler function for an element collection.

Parameters

in	*pvGui*	Void pointer to GUI
in	*sEvent*	Event data structure

Returns

true if success, false if fail

7.19.2.2 void gslc_CollectInput (gslc_tsGui ∗ *pGui,* gslc_tsCollect ∗ *pCollect,* gslc_tsEventTouch ∗ *pEventTouch*)

Handle direct input events within the element collection.

Parameters

`in`	*pGui*	Pointer to the GUI
`in`	*pCollect*	Ptr to the element collection
`in`	*pEventTouch*	Ptr to the touch event structure

Returns

none

7.19.2.3 void gslc_CollectTouch (gslc_tsGui ∗ *pGui,* gslc_tsCollect ∗ *pCollect,* gslc_tsEventTouch ∗ *pEventTouch*)

Handle touch events within the element collection.

Parameters

`in`	*pGui*	Pointer to the GUI
`in`	*pCollect*	Ptr to the element collection
`in`	*pEventTouch*	Ptr to the touch event structure

Returns

none

7.19.2.4 bool gslc_CollectTouchCompound (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teTouch *eTouch,* int16_t *nRelX,* int16_t *nRelY,* gslc_tsCollect ∗ *pCollect*)

Handle dispatch of touch (up,down,move) events to compound elements sub elements.

Parameters

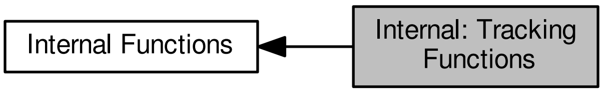
`in`	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
`in`	*pvElem*	Void ptr to Element (typecast to gslc_tsElem∗)
`in`	*eTouch*	Touch event type
`in`	*nRelX*	Touch X coord relative to element
`in`	*nRelY*	Touch Y coord relative to element
`in`	*pCollect*	Collection containing sub elements

Returns

true if success, false otherwise

7.20 Internal: Tracking Functions

Collaboration diagram for Internal: Tracking Functions:



Functions

- void gslc_TrackTouch (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, int16_t nX, int16_t nY, uint16_t nPress)

  Handles a touch event and performs the necessary tracking, glowing and selection actions depending on the press state.
- void gslc_TrackInput (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, gslc_teInputRawEvent eInputEvent, int16_↩ t nInputVal)

  Handles a direct input event and performs the necessary tracking, glowing and selection actions depending on the state.
- bool gslc_InputMapLookup (gslc_tsGui ∗pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc↩ _teAction ∗peAction, int16_t ∗pnActionVal)

7.20.1 Detailed Description

7.20.2 Function Documentation

7.20.2.1 bool gslc_InputMapLookup (**gslc_tsGui** ∗ *pGui*, **gslc_teInputRawEvent** *eInputEvent*, int16_t *nInputVal*, **gslc_teAction** ∗ *peAction*, int16_t ∗ *pnActionVal*)

Todo Doc. This API is experimental and subject to change

7.20.2.2 void gslc_TrackInput (**gslc_tsGui** ∗ *pGui*, **gslc_tsPage** ∗ *pPage*, **gslc_teInputRawEvent** *eInputEvent*, int16_t *nInputVal*)

Handles a direct input event and performs the necessary tracking, glowing and selection actions depending on the state.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pPage*	Pointer to current page
`in`	*eInputEvent*	Indication of event type
`in`	*nInputVal*	Additional data for event type

Returns

none

7.20.2.3 void gslc_TrackTouch (gslc_tsGui ∗ *pGui,* gslc_tsPage ∗ *pPage,* int16_t *nX,* int16_t *nY,* uint16_t *nPress*)

Handles a touch event and performs the necessary tracking, glowing and selection actions depending on the press state.

Parameters

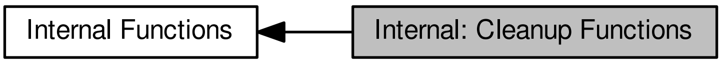
`in`	*pGui*	Pointer to GUI
`in`	*pPage*	Pointer to current page
`in`	*nX*	X coordinate of touch event
`in`	*nY*	Y coordinate of touch event
`in`	*nPress*	Pressure level of touch event (0 for none, else touch)

Returns

none

7.21 Internal: Cleanup Functions

Collaboration diagram for Internal: Cleanup Functions:



Functions

- void gslc_GuiDestruct (gslc_tsGui ∗pGui)

    Free up any surfaces associated with the GUI, pages, collections and elements.
- void gslc_PageDestruct (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage)

    Free up any members associated with a page.
- void gslc_CollectDestruct (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

    Free up any members associated with an element collection.
- void gslc_ElemDestruct (gslc_tsElem ∗pElem)

    Free up any members associated with an element.
- void gslc_ResetFont (gslc_tsFont ∗pFont)

    Initialize a Font struct.
- void gslc_ResetElem (gslc_tsElem ∗pElem)

    Initialize an Element struct.

7.21.1 Detailed Description

7.21.2 Function Documentation

7.21.2.1 void gslc_CollectDestruct (gslc_tsGui ∗ *pGui,* gslc_tsCollect ∗ *pCollect*)

Free up any members associated with an element collection.

Parameters

in	*pGui*	Pointer to GUI
in	*pCollect*	Pointer to collection

Returns

none

7.21.2.2 void gslc_ElemDestruct (gslc_tsElem ∗ *pElem*)

Free up any members associated with an element.

Parameters

in	*pElem*	Pointer to element

Returns

none

7.21.2.3 void gslc_GuiDestruct (gslc_tsGui ∗ *pGui*)

Free up any surfaces associated with the GUI, pages, collections and elements.

Also frees up any fonts.

- Called by gslc_Quit()

Parameters

in	*pGui*	Pointer to GUI

Returns

none

7.21.2.4 void gslc_PageDestruct (gslc_tsGui ∗ *pGui,* gslc_tsPage ∗ *pPage*)

Free up any members associated with a page.

Parameters

in	*pGui*	Pointer to GUI
in	*pPage*	Pointer to Page

Returns

none

7.21.2.5 void gslc_ResetElem (gslc_tsElem ∗ *pElem*)

Initialize an Element struct.

Parameters

in	*pElem*	Pointer to Element

Returns

none

7.21.2.6 void gslc_ResetFont (gslc_tsFont ∗ *pFont*)

Initialize a Font struct.

Parameters

in	*pFont*	Pointer to Font

Returns

none

Chapter 8

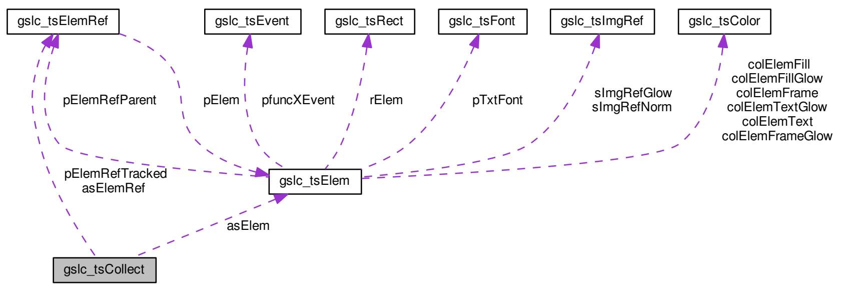
Data Structure Documentation

8.1 gslc_tsCollect Struct Reference

Element collection struct.

`#include <GUIslice.h>`

Collaboration diagram for gslc_tsCollect:



Data Fields

- gslc_tsElem ∗ asElem

  Array of elements.
- uint16_t nElemMax

  Maximum number of elements to allocate (in RAM)
- uint16_t nElemCnt

  Number of elements allocated.
- int16_t nElemAutoIdNext

  Next Element ID for auto-assignment.
- gslc_tsElemRef ∗ asElemRef

  Array of element references.
- uint16_t nElemRefMax

  Maximum number of element references to allocate.

- uint16_t nElemRefCnt

  Number of element references allocated.
- gslc_tsElemRef ∗ pElemRefTracked

  Element reference currently being touch-tracked (NULL for none)
- int16_t nElemIndFocused

  Element index currently in focus (eg. by keyboard/pin control), GSLC_IND_NONE for none.

8.1.1 Detailed Description

Element collection struct.

- Collections are used to maintain a list of elements and any touch tracking status.
- Pages and Compound Elements both instantiate a Collection

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

- src/GUIslice.h

8.2 gslc_tsColor Struct Reference

Color structure. Defines RGB triplet.

```
#include <GUIslice.h>
```

Data Fields

- uint8_t r

  RGB red value.
- uint8_t g

  RGB green value.
- uint8_t b

  RGB blue value.

8.2.1 Detailed Description

Color structure. Defines RGB triplet.

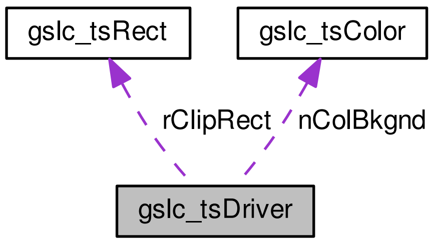
The documentation for this struct was generated from the following file:

- src/GUIslice.h

8.3 gslc_tsDriver Struct Reference

```
#include <GUIslice_drv_adagfx.h>
```

Collaboration diagram for gslc_tsDriver:



Data Fields

- gslc_tsColor nColBkgnd

  Background color (if not image-based)
- gslc_tsRect rClipRect

  Clipping rectangle.

8.3.1 Field Documentation

8.3.1.1 gslc_tsColor gslc_tsDriver::nColBkgnd

Background color (if not image-based)

8.3.1.2 gslc_tsRect gslc_tsDriver::rClipRect

Clipping rectangle.

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

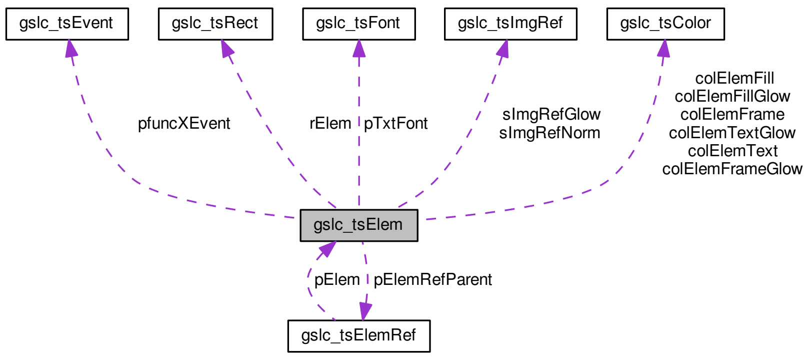
- src/GUIslice_drv_adagfx.h
- src/GUIslice_drv_m5stack.h
- src/GUIslice_drv_tft_espi.h

8.4 gslc_tsElem Struct Reference

Element Struct.

```
#include <GUIslice.h>
```

Collaboration diagram for gslc_tsElem:



Data Fields

- int16_t nId

    Element ID specified by user.
- uint8_t nFeatures

    Element feature vector (appearance/behavior))
- int16_t nType

    Element type enumeration.
- gslc_tsRect rElem

    Rect region containing element.
- int16_t nGroup

    Group ID that the element belongs to.
- gslc_tsColor colElemFrame

    Color for frame.
- gslc_tsColor colElemFill

    Color for background fill.
- gslc_tsColor colElemFrameGlow

    Color to use for frame when glowing.
- gslc_tsColor colElemFillGlow

    Color to use for fill when glowing.
- gslc_tsImgRef sImgRefNorm

    Image reference to draw (normal)
- gslc_tsImgRef sImgRefGlow

    Image reference to draw (glowing)
- gslc_tsElemRef ∗ pElemRefParent

    Parent element reference.
- char ∗ pStrBuf

Ptr to text string buffer to overlay.

- uint8_t nStrBufMax

  Size of string buffer.
- gslc_teTxtFlags eTxtFlags

  Flags associated with text buffer.
- gslc_tsColor colElemText

  Color of overlay text.
- gslc_tsColor colElemTextGlow

  Color of overlay text when glowing.
- int8_t eTxtAlign

  Alignment of overlay text.
- uint8_t nTxtMargin

  Margin of overlay text within rect region.
- gslc_tsFont ∗ pTxtFont

  Ptr to Font for overlay text.
- void ∗ pXData

  Ptr to extended data structure.
- GSLC_CB_EVENT pfuncXEvent

  UNUSED: Callback func ptr for event tree (draw,touch,tick)
- GSLC_CB_DRAW pfuncXDraw

  Callback func ptr for custom drawing.
- GSLC_CB_TOUCH pfuncXTouch

  Callback func ptr for touch.
- GSLC_CB_TICK pfuncXTick

  Callback func ptr for timer/main loop tick.

8.4.1 Detailed Description

Element Struct.

- Represents a single graphic element in the GUIslice environment
- A page is made up of a number of elements
- Each element is created with a user-specified ID for further accesses (or GSLC_ID_AUTO for it to be auto-generated)
- Display order of elements in a page is based upon the creation order
- Extensions to the core element types is provided through the pXData reference and pfuncX∗ callback functions.

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

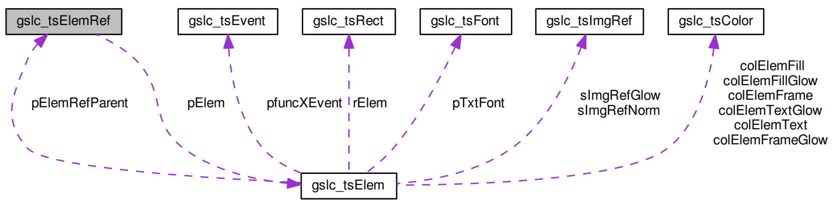
- src/GUIslice.h

8.5 gslc_tsElemRef Struct Reference

Element reference structure.

`#include <GUIslice.h>`

Collaboration diagram for gslc_tsElemRef:



Data Fields

- gslc_tsElem ∗ pElem

    Pointer to element in memory [RAM,FLASH].
- gslc_teElemRefFlags eElemFlags

    Element reference flags.

8.5.1 Detailed Description

Element reference structure.

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

- src/GUIslice.h

8.6 gslc_tsEvent Struct Reference

Event structure.

`#include <GUIslice.h>`

Data Fields

- gslc_teEventType eType

    Event type.
- uint8_t nSubType

    Event sub-type.
- void ∗ pvScope

    Event target scope (eg. Page,Collection,Event)
- void ∗ pvData

    Generic data pointer for event.

8.6.1 Detailed Description

Event structure.

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

- src/GUIslice.h

8.7 gslc_tsEventTouch Struct Reference

Structure used to pass touch data through event.

```
#include <GUIslice.h>
```

Data Fields

- gslc_teTouch eTouch

  Touch state.
- int16_t nX

  Touch X coordinate (or param1)
- int16_t nY

  Touch Y coordinate (or param2)

8.7.1 Detailed Description

Structure used to pass touch data through event.

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

- src/GUIslice.h

8.8 gslc_tsFont Struct Reference

Font reference structure.

```
#include <GUIslice.h>
```

Data Fields

- int16_t nId

  Font ID specified by user.
- gslc_teFontRefType eFontRefType

  Font reference type.
- gslc_teFontRefMode eFontRefMode

  Font reference mode.
- const void ∗ pvFont

  Void ptr to the font reference (type defined by driver)
- uint16_t nSize

  Font size.

8.8.1 Detailed Description

Font reference structure.

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

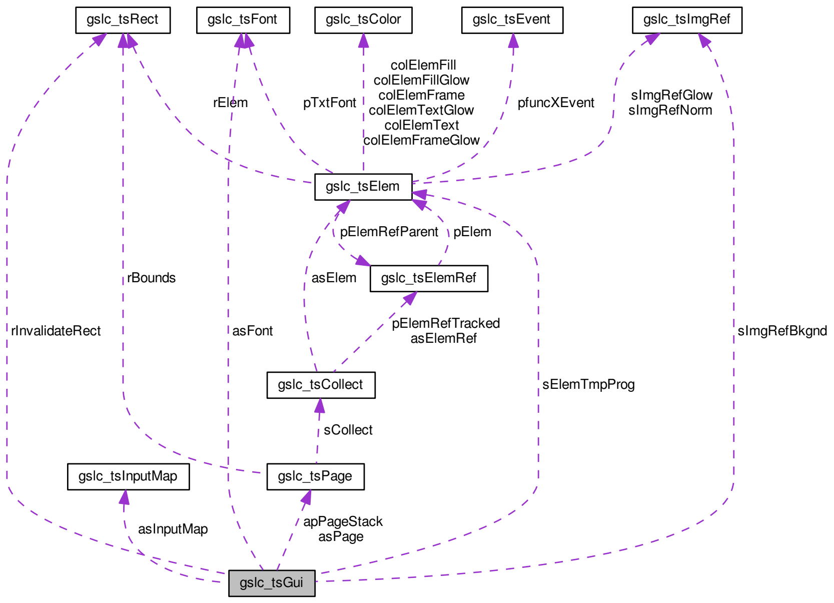
- src/GUIslice.h

8.9 gslc_tsGui Struct Reference

GUI structure.

```
#include <GUIslice.h>
```

Collaboration diagram for gslc_tsGui:



Data Fields

- uint16_t nDispW

  Width of the display (pixels)
- uint16_t nDispH

  Height of the display (pixels)
- uint16_t nDisp0W

  Width of the display (pixels) in native orientation.
- uint16_t nDisp0H

  Height of the display (pixels) in native orientation.
- uint8_t nDispDepth

  Bit depth of display (bits per pixel)

- uint8_t nRotation

  Adafruit GFX Rotation of display.
- uint8_t nTouchRotation

  Touchscreen rotation offset vs display.
- uint8_t nSwapXY

  Adafruit GFX Touch Swap x and y axes.
- uint8_t nFlipX

  Adafruit GFX Touch Flip x axis.
- uint8_t nFlipY

  Adafruit GFX Touch Flip x axis.
- uint16_t nTouchCalXMin

  Calibration X minimum reading.
- uint16_t nTouchCalXMax

  Calibration X maximum reading.
- uint16_t nTouchCalYMin

  Calibration Y minimum reading.
- uint16_t nTouchCalYMax

  Calibration Y maximum reading.
- gslc_tsFont ∗ asFont

  Collection of loaded fonts.
- uint8_t nFontMax

  Maximum number of fonts to allocate.
- uint8_t nFontCnt

  Number of fonts allocated.
- uint8_t nRoundRadius

  Radius for rounded elements.
- gslc_tsElem sElemTmpProg

  Temporary element for Flash compatibility.
- gslc_teInitStat eInitStatTouch

  Status of touch initialization.
- int16_t nTouchLastX

  Last touch event X coord.
- int16_t nTouchLastY

  Last touch event Y coord.
- uint16_t nTouchLastPress

  Last touch event pressure (0=none))
- bool bTouchRemapEn

  Enable touch remapping?
- bool bTouchRemapYX

  Enable touch controller swapping of X & Y.
- void ∗ pvDriver

  Driver-specific members (gslc_tsDriver∗)
- bool bRedrawPartialEn

  Driver supports partial page redraw.
- gslc_tsImgRef sImgRefBkgnd

  Image reference for background.
- uint8_t nFrameRateCnt

  Diagnostic frame rate count.
- uint8_t nFrameRateStart

  Diagnostic frame rate timestamp.
- gslc_tsPage ∗ asPage

Array of all pages defined in system.
- uint8_t nPageMax

  Maximum number of pages that can be defined.
- uint8_t nPageCnt

  Current number of pages defined.
- gslc_tsPage ∗ apPageStack [GSLC_STACK__MAX]

  Stack of pages.
- bool abPageStackActive [GSLC_STACK__MAX]

  Whether page in stack can receive touch events.
- bool abPageStackDoDraw [GSLC_STACK__MAX]

  Whether page in stack is still actively drawn.
- bool bScreenNeedRedraw

  Screen requires a redraw.
- bool bScreenNeedFlip

  Screen requires a page flip.
- bool bInvalidateEn

  A region of the display has been invalidated.
- gslc_tsRect rInvalidateRect

  The rect region that has been invalidated.
- GSLC_CB_PIN_POLL pfuncPinPoll

  Callback func ptr for pin polling.
- gslc_tsInputMap ∗ asInputMap

  Array of input maps.
- uint8_t nInputMapMax

  Maximum number of input maps.
- uint8_t nInputMapCnt

  Current number of input maps.

8.9.1 Detailed Description

GUI structure.

- Contains all GUI state and content
- Maintains list of one or more pages

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

- src/GUIslice.h

8.10 gslc_tsImgRef Struct Reference

Image reference structure.

```
#include <GUIslice.h>
```

Data Fields

- const unsigned char ∗ pImgBuf

  Pointer to input image buffer in memory [RAM,FLASH].
- const char ∗ pFname

  Pathname to input image file [FILE,SD].
- gslc_teImgRefFlags eImgFlags

  Image reference flags.
- void ∗ pvImgRaw

  Ptr to raw output image data (for pre-loaded images)

8.10.1 Detailed Description

Image reference structure.

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

- src/GUIslice.h

8.11 gslc_tsInputMap Struct Reference

Input mapping.

```
#include <GUIslice.h>
```

Data Fields

- gslc_teInputRawEvent eEvent

  The input event.
- int16_t nVal

  The value associated with the input event.
- gslc_teAction eAction

  Resulting action.
- int16_t nActionVal

  The value for the output action.

8.11.1 Detailed Description

Input mapping.

- Describes mapping from keyboard or GPIO input to a GUI action (such as changing the current element focus)
- This is generally used to support keyboard or GPIO control over the GUI operation

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

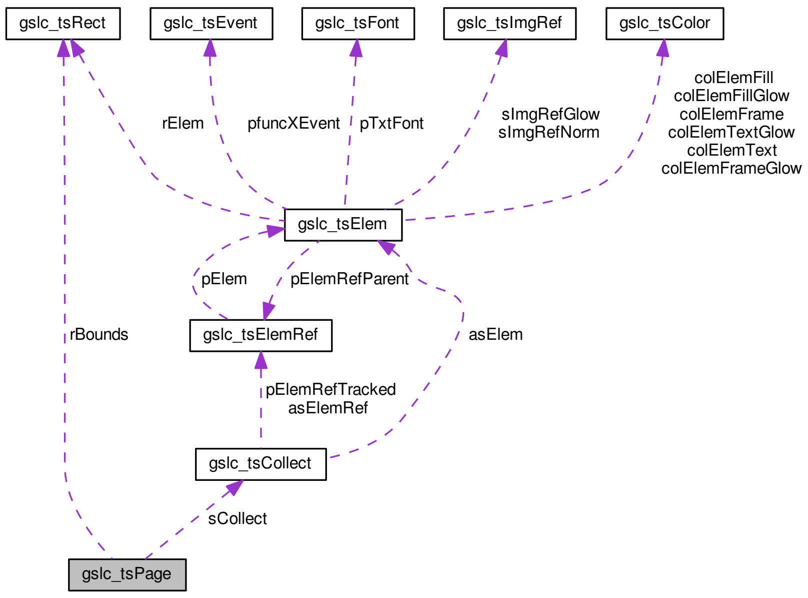
- src/GUIslice.h

8.12 gslc_tsPage Struct Reference

Page structure.

```
#include <GUIslice.h>
```

Collaboration diagram for gslc_tsPage:



Data Fields

- gslc_tsCollect sCollect

    Collection of elements on page.
- int16_t nPageId

    Page identifier.
- gslc_tsRect rBounds

    Bounding rect for page elements.

8.12.1 Detailed Description

Page structure.

- A page contains a collection of elements
- Many redraw functions operate at a page level
- Maintains state as to whether redraw or screen flip is required

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

- src/GUIslice.h

8.13 gslc_tsPt Struct Reference

Define point coordinates.

```
#include <GUIslice.h>
```

Data Fields

- int16_t x

  X coordinate.
- int16_t y

  Y coordinate.

8.13.1 Detailed Description

Define point coordinates.

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

- src/GUIslice.h

8.14 gslc_tsRect Struct Reference

Rectangular region. Defines X,Y corner coordinates plus dimensions.

```
#include <GUIslice.h>
```

Data Fields

- int16_t x

  X coordinate of corner.
- int16_t y

  Y coordinate of corner.
- uint16_t w

  Width of region.
- uint16_t h

  Height of region.

8.14.1 Detailed Description

Rectangular region. Defines X,Y corner coordinates plus dimensions.

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

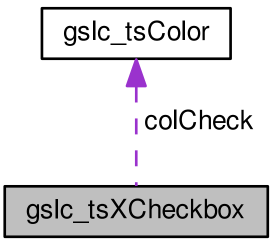
- src/GUIslice.h

8.15 gslc_tsXCheckbox Struct Reference

Extended data for Checkbox element.

```
#include <XCheckbox.h>
```

Collaboration diagram for gslc_tsXCheckbox:



Data Fields

- bool bRadio

    Radio-button operation if true.
- gslc_teXCheckboxStyle nStyle

    Drawing style for element.
- bool bChecked

    Indicates if it is selected (checked)
- gslc_tsColor colCheck

    Color of checked inner fill.
- GSLC_CB_XCHECKBOX pfuncXToggle

    Callback event to say element has changed.

8.15.1 Detailed Description

Extended data for Checkbox element.

8.15.2 Field Documentation

8.15.2.1 bool gslc_tsXCheckbox::bChecked

Indicates if it is selected (checked)

8.15.2.2 bool gslc_tsXCheckbox::bRadio

Radio-button operation if true.

8.15.2.3 gslc_tsColor gslc_tsXCheckbox::colCheck

Color of checked inner fill.

8.15.2.4 gslc_teXCheckboxStyle gslc_tsXCheckbox::nStyle

Drawing style for element.

8.15.2.5 GSLC_CB_XCHECKBOX gslc_tsXCheckbox::pfuncXToggle

Callback event to say element has changed.

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

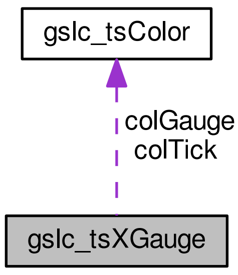
- src/elem/XCheckbox.h

8.16 gslc_tsXGauge Struct Reference

Extended data for Gauge element.

```
#include <XGauge.h>
```

Collaboration diagram for gslc_tsXGauge:



Data Fields

- int16_t nMin

  Minimum control value.
- int16_t nMax

  Maximum control value.
- int16_t nVal

  Current control value.
- int16_t nValLast

  Last value.
- bool bValLastValid

  Last value valid?
- gslc_teXGaugeStyle nStyle

  Gauge sub-type.
- gslc_tsColor colGauge

  Color of gauge fill bar.
- gslc_tsColor colTick

  Color of gauge tick marks.
- uint16_t nTickCnt

  Number of gauge tick marks.
- uint16_t nTickLen

  Length of gauge tick marks.
- bool bVert

  Vertical if true, else Horizontal.
- bool bFlip

  Reverse direction of gauge.
- uint16_t nIndicLen

  Indicator length.
- uint16_t nIndicTip

  Size of tip at end of indicator.
- bool bIndicFill

  Fill the indicator if true.

8.16.1 Detailed Description

Extended data for Gauge element.

8.16.2 Field Documentation

8.16.2.1 bool gslc_tsXGauge::bFlip

Reverse direction of gauge.

8.16.2.2 bool gslc_tsXGauge::bIndicFill

Fill the indicator if true.

8.16.2.3 bool gslc_tsXGauge::bValLastValid

Last value valid?

8.16.2.4 bool gslc_tsXGauge::bVert

Vertical if true, else Horizontal.

8.16.2.5 gslc_tsColor gslc_tsXGauge::colGauge

Color of gauge fill bar.

8.16.2.6 gslc_tsColor gslc_tsXGauge::colTick

Color of gauge tick marks.

8.16.2.7 uint16_t gslc_tsXGauge::nIndicLen

Indicator length.

8.16.2.8 uint16_t gslc_tsXGauge::nIndicTip

Size of tip at end of indicator.

8.16.2.9 int16_t gslc_tsXGauge::nMax

Maximum control value.

8.16.2.10 int16_t gslc_tsXGauge::nMin

Minimum control value.

8.16.2.11 gslc_teXGaugeStyle gslc_tsXGauge::nStyle

Gauge sub-type.

8.16.2.12 uint16_t gslc_tsXGauge::nTickCnt

Number of gauge tick marks.

8.16.2.13 uint16_t gslc_tsXGauge::nTickLen

Length of gauge tick marks.

8.16.2.14 int16_t gslc_tsXGauge::nVal

Current control value.

8.16.2.15 int16_t gslc_tsXGauge::nValLast

Last value.

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

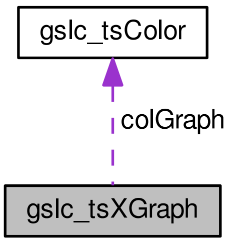
- src/elem/XGauge.h

8.17 gslc_tsXGraph Struct Reference

Extended data for Graph element.

```
#include <XGraph.h>
```

Collaboration diagram for gslc_tsXGraph:



Data Fields

- int16_t ∗ pBuf

    Ptr to the data buffer (circular buffer))
- uint8_t nMargin

    Margin for graph area within element rect.
- gslc_tsColor colGraph

    Color of the graph.
- gslc_teXGraphStyle eStyle

    Style of the graph.
- uint16_t nBufMax

    Maximum number of points in buffer.
- bool bScrollEn

    Enable for scrollbar.
- uint16_t nScrollPos

    Current scrollbar position.
- uint16_t nWndHeight

    Visible window height.
- uint16_t nWndWidth

    Visible window width.
- int16_t nPlotValMax

    Visible window maximum value.
- int16_t nPlotValMin

    Visible window minimum value.
- uint16_t nPlotIndMax

    Number of data points to show in window.
- uint16_t nBufCnt

    Number of points in buffer.
- uint16_t nPlotIndStart

    First row of current window.

8.17.1 Detailed Description

Extended data for Graph element.

8.17.2 Field Documentation

8.17.2.1 bool gslc_tsXGraph::bScrollEn

Enable for scrollbar.

8.17.2.2 gslc_tsColor gslc_tsXGraph::colGraph

Color of the graph.

8.17.2.3 gslc_teXGraphStyle gslc_tsXGraph::eStyle

Style of the graph.

8.17.2.4 uint16_t gslc_tsXGraph::nBufCnt

Number of points in buffer.

8.17.2.5 uint16_t gslc_tsXGraph::nBufMax

Maximum number of points in buffer.

8.17.2.6 uint8_t gslc_tsXGraph::nMargin

Margin for graph area within element rect.

8.17.2.7 uint16_t gslc_tsXGraph::nPlotIndMax

Number of data points to show in window.

8.17.2.8 uint16_t gslc_tsXGraph::nPlotIndStart

First row of current window.

8.17.2.9 int16_t gslc_tsXGraph::nPlotValMax

Visible window maximum value.

8.17.2.10 int16_t gslc_tsXGraph::nPlotValMin

Visible window minimum value.

8.17.2.11 uint16_t gslc_tsXGraph::nScrollPos

Current scrollbar position.

8.17.2.12 uint16_t gslc_tsXGraph::nWndHeight

Visible window height.

8.17.2.13 uint16_t gslc_tsXGraph::nWndWidth

Visible window width.

8.17.2.14 int16_t∗ gslc_tsXGraph::pBuf

Ptr to the data buffer (circular buffer))

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

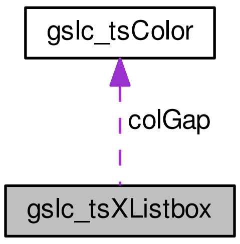
- src/elem/XGraph.h

8.18 gslc_tsXListbox Struct Reference

Extended data for Listbox element.

```
#include <XListbox.h>
```

Collaboration diagram for gslc_tsXListbox:



Data Fields

- uint8_t ∗ pBufItems

    Buffer containing items.
- uint16_t nBufItemsMax

    Max size of buffer containing items.
- uint16_t nBufItemsPos

    Current buffer position.
- int16_t nItemCnt

    Number of items in the list.
- int8_t nCols

    Number of columns.
- int8_t nRows

    Number of columns (or XLSITBOX_SIZE_AUTO to calculate)

- bool bNeedRecalc

  Determine if sizing may need recalc.
- int8_t nMarginW

  Margin inside main listbox area (X offset)
- int8_t nMarginH

  Margin inside main listbox area (Y offset)
- int16_t nItemW

  Width of listbox item.
- int16_t nItemH

  Height of listbox item.
- int8_t nItemGap

  Gap between listbox items.
- gslc_tsColor colGap

  Gap color.
- int8_t nItemMarginW

  Text margin inside listbox items (X offset)
- int8_t nItemMarginH

  Text margin inside listbox items (Y offset)
- bool bItemAutoSizeW

  Enable auto-sizing of items (in width)
- bool bItemAutoSizeH

  Enable auto-sizing of items (in height)
- int16_t nItemCurSel

  Currently selected item (XLISTBOX_SEL_NONE for none)
- int16_t nItemCurSelLast

  Old selected item to redraw (XLISTBOX_SEL_NONE for none)
- int16_t nItemSavedSel

  Persistent selected item (ie. saved selection)
- int16_t nItemTop

  Item to show at top of list after scrolling (0 is default)
- GSLC_CB_XLISTBOX_SEL pfuncXSel

  Callback func ptr for selection update.

8.18.1 Detailed Description

Extended data for Listbox element.

8.18.2 Field Documentation

8.18.2.1 bool gslc_tsXListbox::bItemAutoSizeH

Enable auto-sizing of items (in height)

8.18.2.2 bool gslc_tsXListbox::bItemAutoSizeW

Enable auto-sizing of items (in width)

8.18.2.3 bool gslc_tsXListbox::bNeedRecalc

Determine if sizing may need recalc.

8.18.2.4 gslc_tsColor gslc_tsXListbox::colGap

Gap color.

8.18.2.5 uint16_t gslc_tsXListbox::nBufItemsMax

Max size of buffer containing items.

8.18.2.6 uint16_t gslc_tsXListbox::nBufItemsPos

Current buffer position.

8.18.2.7 int8_t gslc_tsXListbox::nCols

Number of columns.

8.18.2.8 int16_t gslc_tsXListbox::nItemCnt

Number of items in the list.

8.18.2.9 int16_t gslc_tsXListbox::nItemCurSel

Currently selected item (XLISTBOX_SEL_NONE for none)

8.18.2.10 int16_t gslc_tsXListbox::nItemCurSelLast

Old selected item to redraw (XLISTBOX_SEL_NONE for none)

8.18.2.11 int8_t gslc_tsXListbox::nItemGap

Gap between listbox items.

8.18.2.12 int16_t gslc_tsXListbox::nItemH

Height of listbox item.

8.18.2.13 int8_t gslc_tsXListbox::nItemMarginH

Text margin inside listbox items (Y offset)

8.18.2.14 int8_t gslc_tsXListbox::nItemMarginW

Text margin inside listbox items (X offset)

8.18.2.15 int16_t gslc_tsXListbox::nItemSavedSel

Persistent selected item (ie. saved selection)

8.18.2.16 int16_t gslc_tsXListbox::nItemTop

Item to show at top of list after scrolling (0 is default)

8.18.2.17 int16_t gslc_tsXListbox::nItemW

Width of listbox item.

8.18.2.18 int8_t gslc_tsXListbox::nMarginH

Margin inside main listbox area (Y offset)

8.18.2.19 int8_t gslc_tsXListbox::nMarginW

Margin inside main listbox area (X offset)

8.18.2.20 int8_t gslc_tsXListbox::nRows

Number of columns (or XLSITBOX_SIZE_AUTO to calculate)

8.18.2.21 uint8_t∗ gslc_tsXListbox::pBufItems

Buffer containing items.

8.18.2.22 GSLC_CB_XLISTBOX_SEL gslc_tsXListbox::pfuncXSel

Callback func ptr for selection update.

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

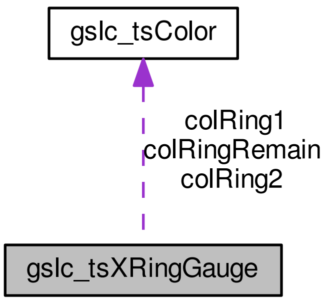
- src/elem/XListbox.h

8.19 gslc_tsXRingGauge Struct Reference

Extended data for XRingGauge element.

```
#include <XRingGauge.h>
```

Collaboration diagram for gslc_tsXRingGauge:



Data Fields

- int16_t nPosMin
- int16_t nPosMax
- uint16_t nDeg64PerSeg
- int8_t nThickness
- bool bGradient
- uint8_t nSegGap
- gslc_tsColor colRing1
- gslc_tsColor colRing2
- gslc_tsColor colRingRemain
- int16_t nPos

  Current position value.
- int16_t nPosLast

  Previous position value.
- char acStrLast [XRING_STR_MAX]

8.19.1 Detailed Description

Extended data for XRingGauge element.

8.19.2 Field Documentation

8.19.2.1 char gslc_tsXRingGauge::acStrLast[XRING_STR_MAX]

8.19.2.2 bool gslc_tsXRingGauge::bGradient

8.19.2.3 gslc_tsColor gslc_tsXRingGauge::colRing1

8.19.2.4 gslc_tsColor gslc_tsXRingGauge::colRing2

8.19.2.5 gslc_tsColor gslc_tsXRingGauge::colRingRemain

8.19.2.6 uint16_t gslc_tsXRingGauge::nDeg64PerSeg

8.19.2.7 int16_t gslc_tsXRingGauge::nPos

Current position value.

8.19.2.8 int16_t gslc_tsXRingGauge::nPosLast

Previous position value.

8.19.2.9 int16_t gslc_tsXRingGauge::nPosMax

8.19.2.10 int16_t gslc_tsXRingGauge::nPosMin

8.19.2.11 uint8_t gslc_tsXRingGauge::nSegGap

8.19.2.12 int8_t gslc_tsXRingGauge::nThickness

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

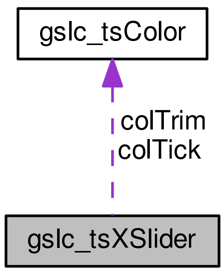
- src/elem/XRingGauge.h

8.20 gslc_tsXSlider Struct Reference

Extended data for Slider element.

`#include <XSlider.h>`

Collaboration diagram for gslc_tsXSlider:



Data Fields

- bool bVert

  Orientation: true if vertical, else horizontal.
- int16_t nThumbSz

  Size of the thumb control.
- int16_t nPosMin

  Minimum position value of the slider.
- int16_t nPosMax

  Maximum position value of the slider.
- uint16_t nTickDiv

  Style: number of tickmark divisions (0 for none)
- int16_t nTickLen

  Style: length of tickmarks.
- gslc_tsColor colTick

  Style: color of ticks.
- bool bTrim

  Style: show a trim color.
- gslc_tsColor colTrim

  Style: color of trim.
- int16_t nPos

  Current position value of the slider.
- GSLC_CB_XSLIDER_POS pfuncXPos

  Callback func ptr for position update.

8.20.1 Detailed Description

Extended data for Slider element.

8.20.2 Field Documentation

8.20.2.1 bool gslc_tsXSlider::bTrim

Style: show a trim color.

8.20.2.2 bool gslc_tsXSlider::bVert

Orientation: true if vertical, else horizontal.

8.20.2.3 gslc_tsColor gslc_tsXSlider::colTick

Style: color of ticks.

8.20.2.4 gslc_tsColor gslc_tsXSlider::colTrim

Style: color of trim.

8.20.2.5 int16_t gslc_tsXSlider::nPos

Current position value of the slider.

8.20.2.6 int16_t gslc_tsXSlider::nPosMax

Maximum position value of the slider.

8.20.2.7 int16_t gslc_tsXSlider::nPosMin

Minimum position value of the slider.

8.20.2.8 int16_t gslc_tsXSlider::nThumbSz

Size of the thumb control.

8.20.2.9 uint16_t gslc_tsXSlider::nTickDiv

Style: number of tickmark divisions (0 for none)

8.20.2.10 int16_t gslc_tsXSlider::nTickLen

Style: length of tickmarks.

8.20.2.11 GSLC_CB_XSLIDER_POS gslc_tsXSlider::pfuncXPos

Callback func ptr for position update.

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

- src/elem/XSlider.h

8.21 gslc_tsXTemplate Struct Reference

Callback function for slider feedback.

```
#include <XTemplate.h>
```

8.21.1 Detailed Description

Callback function for slider feedback.

Extended data for Slider element

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

- src/elem/XTemplate.h

8.22 gslc_tsXTextbox Struct Reference

Extended data for Textbox element.

`#include <XTextbox.h>`

Data Fields

- char ∗ pBuf

  Ptr to the text buffer (circular buffer))
- int8_t nMarginX

  Margin for text area within element rect (X)
- int8_t nMarginY

  Margin for text area within element rect (Y)
- bool bWrapEn

  Enable for line wrapping.
- uint16_t nBufRows

  Number of rows in buffer.
- uint16_t nBufCols

  Number of columns in buffer.
- bool bScrollEn

  Enable for scrollbar.
- uint16_t nScrollPos

  Current scrollbar position.
- uint8_t nChSizeX

  Width of characters (pixels)
- uint8_t nChSizeY

  Height of characters (pixels)
- uint8_t nWndCols

  Window X size.
- uint8_t nWndRows

  Window Y size.
- uint8_t nCurPosX

  Cursor X position.
- uint8_t nCurPosY

  Cursor Y position.
- uint8_t nBufPosX

  Buffer X position.
- uint8_t nBufPosY

  Buffer Y position.
- uint8_t nWndRowStart

  First row of current window.
- int16_t nRedrawRow

  Specific row to update in redraw (if not -1)

8.22.1 Detailed Description

Extended data for Textbox element.

8.22.2 Field Documentation

8.22.2.1 bool gslc_tsXTextbox::bScrollEn

Enable for scrollbar.

8.22.2.2 bool gslc_tsXTextbox::bWrapEn

Enable for line wrapping.

8.22.2.3 uint16_t gslc_tsXTextbox::nBufCols

Number of columns in buffer.

8.22.2.4 uint8_t gslc_tsXTextbox::nBufPosX

Buffer X position.

8.22.2.5 uint8_t gslc_tsXTextbox::nBufPosY

Buffer Y position.

8.22.2.6 uint16_t gslc_tsXTextbox::nBufRows

Number of rows in buffer.

8.22.2.7 uint8_t gslc_tsXTextbox::nChSizeX

Width of characters (pixels)

8.22.2.8 uint8_t gslc_tsXTextbox::nChSizeY

Height of characters (pixels)

8.22.2.9 uint8_t gslc_tsXTextbox::nCurPosX

Cursor X position.

8.22.2.10 uint8_t gslc_tsXTextbox::nCurPosY

Cursor Y position.

8.22.2.11 int8_t gslc_tsXTextbox::nMarginX

Margin for text area within element rect (X)

8.22.2.12 int8_t gslc_tsXTextbox::nMarginY

Margin for text area within element rect (Y)

8.22.2.13 int16_t gslc_tsXTextbox::nRedrawRow

Specific row to update in redraw (if not -1)

8.22.2.14 uint16_t gslc_tsXTextbox::nScrollPos

Current scrollbar position.

8.22.2.15 uint8_t gslc_tsXTextbox::nWndCols

Window X size.

8.22.2.16 uint8_t gslc_tsXTextbox::nWndRows

Window Y size.

8.22.2.17 uint8_t gslc_tsXTextbox::nWndRowStart

First row of current window.

8.22.2.18 char∗ gslc_tsXTextbox::pBuf

Ptr to the text buffer (circular buffer))

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

- src/elem/XTextbox.h

8.23 THPoint Class Reference

```
#include <GUIslice_th.h>
```

Public Member Functions

- THPoint (void)
- THPoint (uint16_t x, uint16_t y, uint16_t z)
- bool operator== (THPoint)
- bool operator!= (THPoint)

Data Fields

- uint16_t x
- uint16_t y
- uint16_t z

8.23.1 Constructor & Destructor Documentation

8.23.1.1 THPoint::THPoint (void)

8.23.1.2 THPoint::THPoint (uint16_t *x*, uint16_t *y*, uint16_t *z*)

8.23.2 Member Function Documentation

8.23.2.1 bool THPoint::operator!= (**THPoint** *p1*)

8.23.2.2 bool THPoint::operator== (**THPoint** *p1*)

8.23.3 Field Documentation

8.23.3.1 uint16_t THPoint::x

8.23.3.2 uint16_t THPoint::y

8.23.3.3 uint16_t THPoint::z

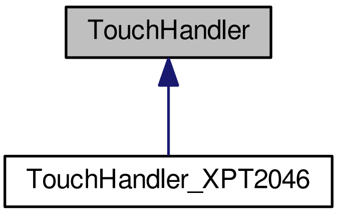
The documentation for this class was generated from the following files:

- src/GUIslice_th.h
- src/GUIslice_th.cpp

8.24 TouchHandler Class Reference

```
#include <GUIslice_th.h>
```

Inheritance diagram for TouchHandler:



Public Member Functions

- TouchHandler ()
- void setSize (uint16_t _disp_xSize, uint16_t _disp_ySize)
- void setCalibration (uint16_t ts_xMin, uint16_t ts_xMax, uint16_t ts_yMin, uint16_t ts_yMax)
- void setSwapFlip (bool _swapXY, bool _flipX, bool _flipY)
- THPoint scale (THPoint pIn)
- virtual void begin (void)
- virtual THPoint getPoint (void)

8.24.1 Constructor & Destructor Documentation

8.24.1.1 TouchHandler::TouchHandler ( ) `[inline]`

8.24.2 Member Function Documentation

8.24.2.1 void TouchHandler::begin (void) `[virtual]`

Reimplemented in TouchHandler_XPT2046.

8.24.2.2 THPoint TouchHandler::getPoint (void) `[virtual]`

Reimplemented in TouchHandler_XPT2046.

8.24.2.3 THPoint TouchHandler::scale (THPoint *pIn*)

8.24.2.4 void TouchHandler::setCalibration (uint16_t *ts_xMin,* uint16_t *ts_xMax,* uint16_t *ts_yMin,* uint16_t *ts_yMax*)

8.24.2.5 void TouchHandler::setSize (uint16_t *_disp_xSize,* uint16_t *_disp_ySize*)

8.24.2.6 void TouchHandler::setSwapFlip (bool *_swapXY,* bool *_flipX,* bool *_flipY*)

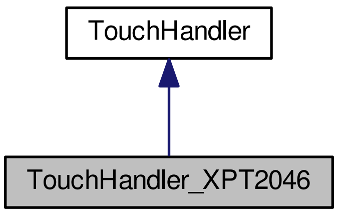
The documentation for this class was generated from the following files:

- src/GUIslice_th.h
- src/GUIslice_th.cpp

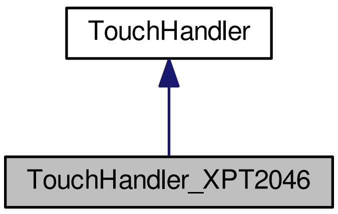
8.25 TouchHandler_XPT2046 Class Reference

```
#include <GUIslice_th_XPT2046.h>
```

Inheritance diagram for TouchHandler_XPT2046:



Collaboration diagram for TouchHandler_XPT2046:



Public Member Functions

- TouchHandler_XPT2046 (SPIClass &spi, uint8_t spi_cs_pin)
- void begin (void)
- THPoint getPoint (void)

Data Fields

- SPIClass spi
- XPT2046_touch touchDriver

8.25.1 Constructor & Destructor Documentation

8.25.1.1 TouchHandler_XPT2046::TouchHandler_XPT2046 (SPIClass & *spi,* uint8_t *spi_cs_pin*) `[inline]`

8.25.2 Member Function Documentation

8.25.2.1 void TouchHandler_XPT2046::begin (void) `[inline], [virtual]`

Reimplemented from TouchHandler.

8.25.2.2 THPoint TouchHandler_XPT2046::getPoint (void) `[inline], [virtual]`

Reimplemented from TouchHandler.

8.25.3 Field Documentation

8.25.3.1 SPIClass TouchHandler_XPT2046::spi

8.25.3.2 XPT2046_touch TouchHandler_XPT2046::touchDriver

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

- src/GUIslice_th_XPT2046.h

Chapter 9

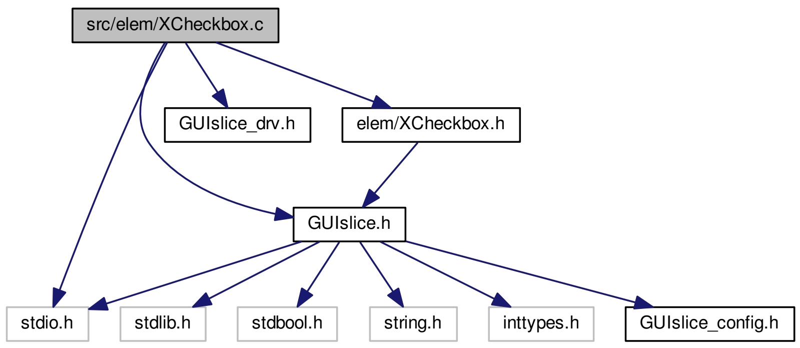
File Documentation

9.1 README.md File Reference

9.2 src/elem/XCheckbox.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XCheckbox.h"
#include <stdio.h>
```

Include dependency graph for XCheckbox.c:



Functions

- gslc_tsElemRef ∗ gslc_ElemXCheckboxCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_↩ tsXCheckbox ∗pXData, gslc_tsRect rElem, bool bRadio, gslc_teXCheckboxStyle nStyle, gslc_tsColor col↩ Check, bool bChecked)

    Create a Checkbox or Radio button Element.
- bool gslc_ElemXCheckboxGetState (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

    Get a Checkbox element's current state.
- gslc_tsElemRef ∗ gslc_ElemXCheckboxFindChecked (gslc_tsGui ∗pGui, int16_t nGroupId)

Find the checkbox within a group that has been checked.

- void gslc_ElemXCheckboxSetStateFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_XCH←ECKBOX pfuncCb)

  Assign the state callback function for a checkbox/radio button.
- void gslc_ElemXCheckboxSetStateHelp (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bChecked)
- void gslc_ElemXCheckboxSetState (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bChecked)

  Set a Checkbox element's current state.
- void gslc_ElemXCheckboxToggleState (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Toggle a Checkbox element's current state.
- bool gslc_ElemXCheckboxDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a Checkbox element on the screen.
- bool gslc_ElemXCheckboxTouch (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)

  Handle touch events to Checkbox element.

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.2.1 Function Documentation

9.2.1.1 gslc_tsElemRef∗ gslc_ElemXCheckboxCreate (gslc_tsGui ∗ *pGui*, int16_t *nElemId*, int16_t *nPage*, gslc_tsXCheckbox ∗ *pXData*, gslc_tsRect *rElem*, bool *bRadio*, gslc_teXCheckboxStyle *nStyle*, gslc_tsColor *colCheck*, bool *bChecked*)

Create a Checkbox or Radio button Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*bRadio*	Radio-button functionality if true
in	*nStyle*	Drawing style for checkbox / radio button
in	*colCheck*	Color for inner fill when checked
in	*bChecked*	Default state

Returns

Pointer to Element reference or NULL if failure

9.2.1.2 bool gslc_ElemXCheckboxDraw (void ∗ *pvGui*, void ∗ *pvElemRef*, gslc_teRedrawType *eRedraw*)

Draw a Checkbox element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.2.1.3 gslc_tsElemRef∗ gslc_ElemXCheckboxFindChecked (gslc_tsGui ∗ *pGui*, int16_t *nGroupId*)

Find the checkbox within a group that has been checked.

Parameters

in	*n↩ GroupId*	Group ID to search

Returns

Element Ptr or NULL if none checked

9.2.1.4 bool gslc_ElemXCheckboxGetState (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*)

Get a Checkbox element's current state.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

Current state

9.2.1.5 void gslc_ElemXCheckboxSetState (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, bool *bChecked*)

Set a Checkbox element's current state.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bChecked*	New state

Returns

none

9.2.1.6 void gslc_ElemXCheckboxSetStateFunc (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* GSLC_CB_XCHECKBOX *pfuncCb*)

Assign the state callback function for a checkbox/radio button.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*pfuncCb*	Function pointer to callback routine (or NULL for none)

Returns

none

9.2.1.7 void gslc_ElemXCheckboxSetStateHelp (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* bool *bChecked*)

9.2.1.8 void gslc_ElemXCheckboxToggleState (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Toggle a Checkbox element's current state.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

none

9.2.1.9 bool gslc_ElemXCheckboxTouch (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teTouch *eTouch,* int16_t *nRelX,* int16_t *nRelY*)

Handle touch events to Checkbox element.

- Called from gslc_ElemSendEventTouch()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eTouch*	Touch event type
in	*nRelX*	Touch X coord relative to element
in	*nRelY*	Touch Y coord relative to element

Returns

true if success, false otherwise

9.2.2 Variable Documentation

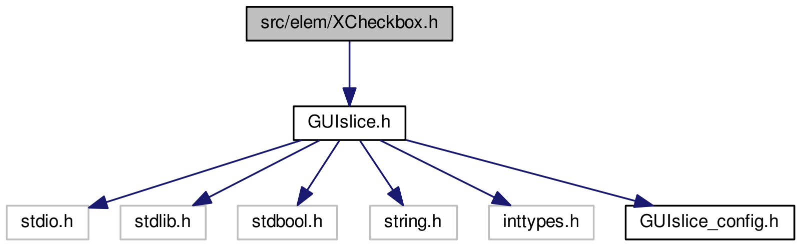
9.2.2.1 const char ERRSTR_NULL

9.2.2.2 const char **GSLC_PMEM** ERRSTR_PXD_NULL[ ]

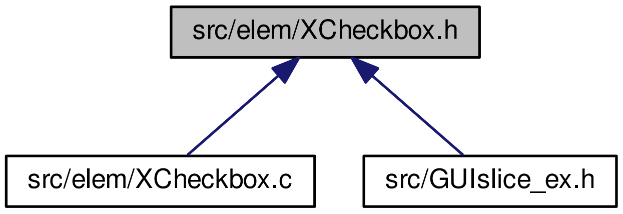
9.3 src/elem/XCheckbox.h File Reference

`#include "GUIslice.h"`

Include dependency graph for XCheckbox.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsXCheckbox

  Extended data for Checkbox element.

Macros

- #define GSLC_TYPEX_CHECKBOX
- #define gslc_ElemXCheckboxCreate_P(pGui, nElemId, nPage, nX, nY, nW, nH, colFill, bFillEn, nGroup, b↩ Radio_, nStyle_, colCheck_, bChecked_)

  Create a Checkbox or Radio button Element in Flash.

Typedefs

- typedef bool(∗ GSLC_CB_XCHECKBOX) (void ∗pvGui, void ∗pvElemRef, int16_t nSelId, bool bChecked)

  Callback function for checkbox/radio element state change.

Enumerations

- enum gslc_teXCheckboxStyle { GSLCX_CHECKBOX_STYLE_BOX, GSLCX_CHECKBOX_STYLE_X, G↩ SLCX_CHECKBOX_STYLE_ROUND }

  Checkbox drawing style.

Functions

- gslc_tsElemRef ∗ gslc_ElemXCheckboxCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_↩ tsXCheckbox ∗pXData, gslc_tsRect rElem, bool bRadio, gslc_teXCheckboxStyle nStyle, gslc_tsColor col↩ Check, bool bChecked)

  Create a Checkbox or Radio button Element.
- bool gslc_ElemXCheckboxGetState (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get a Checkbox element's current state.
- void gslc_ElemXCheckboxSetState (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bChecked)

  Set a Checkbox element's current state.
- gslc_tsElemRef ∗ gslc_ElemXCheckboxFindChecked (gslc_tsGui ∗pGui, int16_t nGroupId)

  Find the checkbox within a group that has been checked.
- void gslc_ElemXCheckboxToggleState (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Toggle a Checkbox element's current state.
- void gslc_ElemXCheckboxSetStateFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_XCH↩ ECKBOX pfuncCb)

  Assign the state callback function for a checkbox/radio button.
- bool gslc_ElemXCheckboxDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a Checkbox element on the screen.
- bool gslc_ElemXCheckboxTouch (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)

  Handle touch events to Checkbox element.

9.3.1 Macro Definition Documentation

9.3.1.1 #define gslc_ElemXCheckboxCreate_P(*pGui, nElemId, nPage, nX, nY, nW, nH, colFill, bFillEn, nGroup, bRadio_, nStyle_, colCheck_, bChecked_*)

Create a Checkbox or Radio button Element in Flash.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Unique element ID to assign
in	*nPage*	Page ID to attach element to
in	*nX*	X coordinate of element
in	*nY*	Y coordinate of element
in	*nW*	Width of element
in	*nH*	Height of element
in	*colFill*	Color for the control background fill
in	*bFillEn*	True if background filled, false otherwise (recommend True)
in	*nGroup*	Group ID that radio buttons belong to (else GSLC_GROUP_NONE)
in	*bRadio_*	Radio-button functionality if true
in	*nStyle_*	Drawing style for checkbox / radio button
in	*col←↩ Check_*	Color for inner fill when checked
in	*b←↩ Checked←↩ _*	Default state

Returns

none

9.3.1.2 #define GSLC_TYPEX_CHECKBOX

9.3.2 Typedef Documentation

9.3.2.1 typedef bool(∗ GSLC_CB_XCHECKBOX) (void ∗pvGui, void ∗pvElemRef, int16_t nSelId, bool bChecked)

Callback function for checkbox/radio element state change.

- nSelId: Selected element's ID or GSLC_ID_NONE
- bChecked: Element was selected if true, false otherwise

9.3.3 Enumeration Type Documentation

9.3.3.1 enum gslc_teXCheckboxStyle

Checkbox drawing style.

Enumerator

GSLCX_CHECKBOX_STYLE_BOX	Inner box.
GSLCX_CHECKBOX_STYLE_X	Crossed.
GSLCX_CHECKBOX_STYLE_ROUND	Circular.

9.3.4 Function Documentation

9.3.4.1 gslc_tsElemRef∗ gslc_ElemXCheckboxCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXCheckbox ∗ *pXData,* gslc_tsRect *rElem,* bool *bRadio,* gslc_teXCheckboxStyle *nStyle,* gslc_tsColor *colCheck,* bool *bChecked*)

Create a Checkbox or Radio button Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*bRadio*	Radio-button functionality if true
in	*nStyle*	Drawing style for checkbox / radio button
in	*colCheck*	Color for inner fill when checked
in	*bChecked*	Default state

Returns

Pointer to Element reference or NULL if failure

9.3.4.2 bool gslc_ElemXCheckboxDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a Checkbox element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.3.4.3 gslc_tsElemRef∗ gslc_ElemXCheckboxFindChecked (gslc_tsGui ∗ *pGui,* int16_t *nGroupId*)

Find the checkbox within a group that has been checked.

Parameters

in	*pGui*	Pointer to GUI
in	*n↩ GroupId*	Group ID to search

Returns

Element Ptr or NULL if none checked

9.3.4.4 bool gslc_ElemXCheckboxGetState (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get a Checkbox element's current state.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

Current state

9.3.4.5 void gslc_ElemXCheckboxSetState (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* bool *bChecked*)

Set a Checkbox element's current state.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bChecked*	New state

Returns

none

9.3.4.6 void gslc_ElemXCheckboxSetStateFunc (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* GSLC_CB_XCHECKBOX *pfuncCb*)

Assign the state callback function for a checkbox/radio button.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*pfuncCb*	Function pointer to callback routine (or NULL for none)

Returns

none

9.3.4.7 void gslc_ElemXCheckboxToggleState (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Toggle a Checkbox element's current state.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference

Returns

none

9.3.4.8 bool gslc_ElemXCheckboxTouch (void ∗ *pvGui*, void ∗ *pvElemRef*, gslc_teTouch *eTouch*, int16_t *nRelX*, int16_t *nRelY*)

Handle touch events to Checkbox element.

- Called from gslc_ElemSendEventTouch()

Parameters

`in`	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
`in`	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
`in`	*eTouch*	Touch event type
`in`	*nRelX*	Touch X coord relative to element
`in`	*nRelY*	Touch Y coord relative to element

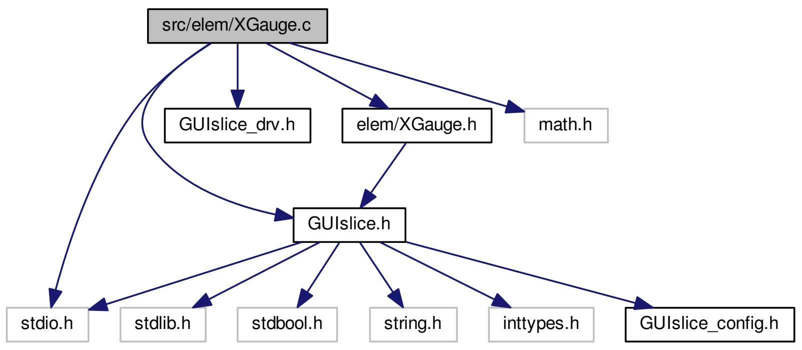
Returns

true if success, false otherwise

9.4 src/elem/XGauge.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XGauge.h"
#include <stdio.h>
#include <math.h>
```

Include dependency graph for XGauge.c:



Functions

- gslc_tsElemRef ∗ gslc_ElemXGaugeCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsX←Gauge ∗pXData, gslc_tsRect rElem, int16_t nMin, int16_t nMax, int16_t nVal, gslc_tsColor colGauge, bool bVert)

  Create a Gauge Element.
- void gslc_ElemXGaugeSetStyle (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teXGaugeStyle nStyle)

  Configure the style of a Gauge element.
- void gslc_ElemXGaugeSetIndicator (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colGauge, uint16_t nIndicLen, uint16_t nIndicTip, bool bIndicFill)

  Configure the appearance of the Gauge indicator.
- void gslc_ElemXGaugeSetTicks (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colTick, uint16_t nTickCnt, uint16_t nTickLen)

  Configure the appearance of the Gauge ticks.
- void gslc_ElemXGaugeUpdate (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nVal)

  Update a Gauge element's current value.
- void gslc_ElemXGaugeSetFlip (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bFlip)

  Set a Gauge element's fill direction.
- bool gslc_ElemXGaugeDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a gauge element on the screen.
- bool gslc_ElemXGaugeDrawProgressBar (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teRedraw←Type eRedraw)

  Helper function to draw a gauge with style: progress bar.

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.4.1 Function Documentation

9.4.1.1 gslc_tsElemRef∗ gslc_ElemXGaugeCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXGauge ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nMin,* int16_t *nMax,* int16_t *nVal,* gslc_tsColor *colGauge,* bool *bVert*)

Create a Gauge Element.

- Draws a gauge element that represents a proportion (nVal) between nMin and nMax.
- Support gauge sub-types:
  - GSLC_TYPEX_GAUGE_PROG_BAR: Horizontal or vertical box with filled region
  - GSLC_TYPEX_GAUGE_RADIAL: Radial / compass indicator
- Default appearance is a horizontal progress bar, but can be changed with gslc_ElemXGaugeSetStyle())

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining gauge size
in	*nMin*	Minimum value of gauge for nVal comparison
in	*nMax*	Maximum value of gauge for nVal comparison
in	*nVal*	Starting value of gauge
in	*colGauge*	Color for the gauge indicator
in	*bVert*	Flag to indicate vertical vs horizontal action (true = vertical, false = horizontal)

Returns

Pointer to Element reference or NULL if failure

9.4.1.2 bool gslc_ElemXGaugeDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a gauge element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.4.1.3 bool gslc_ElemXGaugeDrawProgressBar (gslc_tsGui * *pGui*, gslc_tsElemRef * *pElemRef*, gslc_teRedrawType *eRedraw*)

Helper function to draw a gauge with style: progress bar.

- Called from gslc_ElemXGaugeDraw()

Parameters

in	*pGui*	Ptr to GUI
in	*pElemRef*	Ptr to Element reference
in	*eRedraw*	Redraw status

Returns

true if success, false otherwise

9.4.1.4 void gslc_ElemXGaugeSetFlip (gslc_tsGui * *pGui*, gslc_tsElemRef * *pElemRef*, bool *bFlip*)

Set a Gauge element's fill direction.

- Setting bFlip reverses the default fill direction
- Default fill direction for horizontal gauges: left-to-right
- Default fill direction for vertical gauges: bottom-to-top

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bFlip*	If set, reverse direction of fill from default

Returns

none

9.4.1.5 void gslc_ElemXGaugeSetIndicator (gslc_tsGui * *pGui*, gslc_tsElemRef * *pElemRef*, gslc_tsColor *colGauge*, uint16_t *nIndicLen*, uint16_t *nIndicTip*, bool *bIndicFill*)

Configure the appearance of the Gauge indicator.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*colGauge*	Color of the indicator
`in`	*nIndicLen*	Length of the indicator
`in`	*nIndicTip*	Size of the indicator tip
`in`	*bIndicFill*	Fill in the indicator if true

Returns

none

9.4.1.6 void gslc_ElemXGaugeSetStyle (**gslc_tsGui** ∗ *pGui,* **gslc_tsElemRef** ∗ *pElemRef,* **gslc_teXGaugeStyle** *nType*)

Configure the style of a Gauge element.

- This function is used to select between one of several gauge types (eg. progress bar, radial dial, etc.)

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*nType*	Gauge style enumeration

Returns

none

9.4.1.7 void gslc_ElemXGaugeSetTicks (**gslc_tsGui** ∗ *pGui,* **gslc_tsElemRef** ∗ *pElemRef,* **gslc_tsColor** *colTick,* uint16_t *nTickCnt,* uint16_t *nTickLen*)

Configure the appearance of the Gauge ticks.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*colTick*	Color of the gauge ticks
`in`	*nTickCnt*	Number of ticks to draw around / along gauge
`in`	*nTickLen*	Length of the tick marks to draw

Returns

none

9.4.1.8 void gslc_ElemXGaugeUpdate (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nVal*)

Update a Gauge element's current value.

- Note that min & max values are assigned in create()

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nVal*	New value to show in gauge

Returns

none

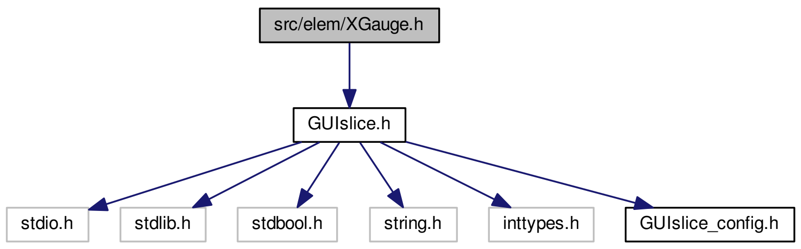
9.4.2 Variable Documentation

9.4.2.1 const char GSLC_PMEM ERRSTR_NULL[ ]

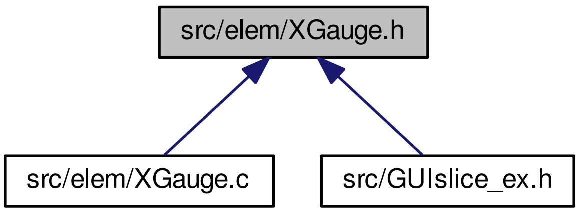
9.4.2.2 const char GSLC_PMEM ERRSTR_PXD_NULL[ ]

9.5 src/elem/XGauge.h File Reference

```
#include "GUIslice.h"
```
Include dependency graph for XGauge.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsXGauge

  Extended data for Gauge element.

Macros

- #define GSLC_TYPEX_GAUGE
- #define gslc_ElemXGaugeCreate_P(pGui, nElemId, nPage, nX, nY, nW, nH, nMin_, nMax_, nVal_, col↩ Frame_, colFill_, colGauge_, bVert_)

  Create a Gauge Element in Flash.

Enumerations

- enum gslc_teXGaugeStyle { GSLCX_GAUGE_STYLE_PROG_BAR, GSLCX_GAUGE_STYLE_RADIAL, GSLCX_GAUGE_STYLE_RAMP }

  Gauge drawing style.

Functions

- gslc_tsElemRef ∗ gslc_ElemXGaugeCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsX↩ Gauge ∗pXData, gslc_tsRect rElem, int16_t nMin, int16_t nMax, int16_t nVal, gslc_tsColor colGauge, bool bVert)

  Create a Gauge Element.
- void gslc_ElemXGaugeSetStyle (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teXGaugeStyle nType)

  Configure the style of a Gauge element.
- void gslc_ElemXGaugeSetIndicator (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colGauge, uint16_t nIndicLen, uint16_t nIndicTip, bool bIndicFill)

  Configure the appearance of the Gauge indicator.
- void gslc_ElemXGaugeSetTicks (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colTick, uint16_t nTickCnt, uint16_t nTickLen)

  Configure the appearance of the Gauge ticks.
- void gslc_ElemXGaugeUpdate (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nVal)

  Update a Gauge element's current value.

- void gslc_ElemXGaugeSetFlip (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bFlip)

  Set a Gauge element's fill direction.
- bool gslc_ElemXGaugeDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a gauge element on the screen.
- bool gslc_ElemXGaugeDrawProgressBar (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teRedraw←Type eRedraw)

  Helper function to draw a gauge with style: progress bar.

9.5.1 Macro Definition Documentation

9.5.1.1 #define gslc_ElemXGaugeCreate_P(*pGui, nElemId, nPage, nX, nY, nW, nH, nMin_, nMax_, nVal_, colFrame_, colFill_, colGauge_, bVert_*)

Create a Gauge Element in Flash.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Unique element ID to assign
in	*nPage*	Page ID to attach element to
in	*nX*	X coordinate of element
in	*nY*	Y coordinate of element
in	*nW*	Width of element
in	*nH*	Height of element
in	*nMin_*	Minimum value of gauge for nVal comparison
in	*nMax_*	Maximum value of gauge for nVal comparison
in	*nVal_*	Starting value of gauge
in	*col←Frame_*	Color for the gauge frame
in	*colFill_*	Color for the gauge background fill
in	*col←Gauge_*	Color for the gauge indicator
in	*bVert_*	Flag to indicate vertical vs horizontal action (true = vertical, false = horizontal)

Returns

none

9.5.1.2 #define GSLC_TYPEX_GAUGE

9.5.2 Enumeration Type Documentation

9.5.2.1 enum gslc_teXGaugeStyle

Gauge drawing style.

Enumerator

GSLCX_GAUGE_STYLE_PROG_BAR	Progress bar.
GSLCX_GAUGE_STYLE_RADIAL	Radial indicator.
GSLCX_GAUGE_STYLE_RAMP	Ramp indicator.

9.5.3 Function Documentation

9.5.3.1 gslc_tsElemRef∗ gslc_ElemXGaugeCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXGauge ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nMin,* int16_t *nMax,* int16_t *nVal,* gslc_tsColor *colGauge,* bool *bVert*)

Create a Gauge Element.

- Draws a gauge element that represents a proportion (nVal) between nMin and nMax.
- Support gauge sub-types:
  - GSLC_TYPEX_GAUGE_PROG_BAR: Horizontal or vertical box with filled region
  - GSLC_TYPEX_GAUGE_RADIAL: Radial / compass indicator
- Default appearance is a horizontal progress bar, but can be changed with gslc_ElemXGaugeSetStyle())

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining gauge size
in	*nMin*	Minimum value of gauge for nVal comparison
in	*nMax*	Maximum value of gauge for nVal comparison
in	*nVal*	Starting value of gauge
in	*colGauge*	Color for the gauge indicator
in	*bVert*	Flag to indicate vertical vs horizontal action (true = vertical, false = horizontal)

Returns

Pointer to Element reference or NULL if failure

9.5.3.2 bool gslc_ElemXGaugeDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a gauge element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.5.3.3 bool gslc_ElemXGaugeDrawProgressBar (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_teRedrawType *eRedraw*)

Helper function to draw a gauge with style: progress bar.

- Called from gslc_ElemXGaugeDraw()

Parameters

in	*pGui*	Ptr to GUI
in	*pElemRef*	Ptr to Element reference
in	*eRedraw*	Redraw status

Returns

true if success, false otherwise

9.5.3.4 void gslc_ElemXGaugeSetFlip (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, bool *bFlip*)

Set a Gauge element's fill direction.

- Setting bFlip reverses the default fill direction
- Default fill direction for horizontal gauges: left-to-right
- Default fill direction for vertical gauges: bottom-to-top

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bFlip*	If set, reverse direction of fill from default

Returns

none

9.5.3.5 void gslc_ElemXGaugeSetIndicator (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_tsColor *colGauge*, uint16_t *nIndicLen*, uint16_t *nIndicTip*, bool *bIndicFill*)

Configure the appearance of the Gauge indicator.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*colGauge*	Color of the indicator
in	*nIndicLen*	Length of the indicator
in	*nIndicTip*	Size of the indicator tip
in	*bIndicFill*	Fill in the indicator if true

Returns

none

9.5.3.6 void gslc_ElemXGaugeSetStyle (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_teXGaugeStyle *nType*)

Configure the style of a Gauge element.

- This function is used to select between one of several gauge types (eg. progress bar, radial dial, etc.)

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nType*	Gauge style enumeration

Returns

none

9.5.3.7 void gslc_ElemXGaugeSetTicks (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_tsColor *colTick*, uint16_t *nTickCnt*, uint16_t *nTickLen*)

Configure the appearance of the Gauge ticks.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*colTick*	Color of the gauge ticks
in	*nTickCnt*	Number of ticks to draw around / along gauge
in	*nTickLen*	Length of the tick marks to draw

Returns

none

9.5.3.8 void gslc_ElemXGaugeUpdate (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nVal*)

Update a Gauge element's current value.

- Note that min & max values are assigned in create()

Parameters

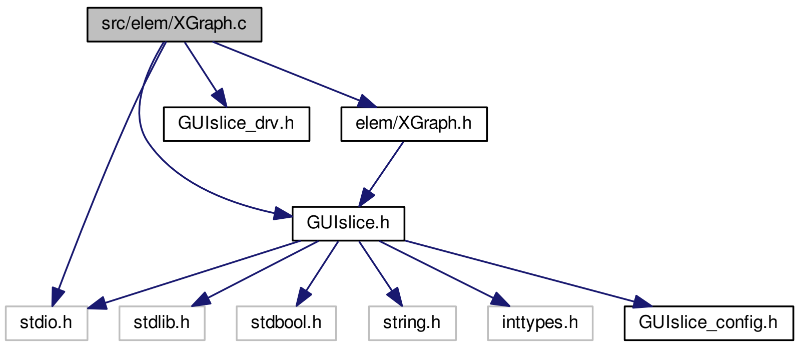
in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nVal*	New value to show in gauge

Returns

none

9.6 src/elem/XGraph.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XGraph.h"
#include <stdio.h>
```
Include dependency graph for XGraph.c:



Functions

- gslc_tsElemRef ∗ gslc_ElemXGraphCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsX←Graph ∗pXData, gslc_tsRect rElem, int16_t nFontId, int16_t ∗pBuf, uint16_t nBufMax, gslc_tsColor colGraph)

Create a Graph Element.

- void gslc_ElemXGraphSetStyle (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teXGraphStyle eStyle, uint8_t nMargin)

  Set the graph's additional drawing characteristics.
- void gslc_ElemXGraphSetRange (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nYMin, int16_t n←YMax)

  Set the graph's drawing range.
- void gslc_ElemXGraphScrollSet (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nScrollPos, uint8_t nScrollMax)

  Set the graph scroll position (nScrollPos) as a fraction of nScrollMax.
- void gslc_ElemXGraphAdd (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nVal)

  Add a value to the graph at the latest position.
- bool gslc_ElemXGraphDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a Graph element on the screen.

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.6.1 Function Documentation

9.6.1.1 void gslc_ElemXGraphAdd (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nVal*)

Add a value to the graph at the latest position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nVal*	Data value to add

Returns

none

9.6.1.2 gslc_tsElemRef∗ gslc_ElemXGraphCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXGraph ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nFontId,* int16_t ∗ *pBuf,* uint16_t *nBufRows,* gslc_tsColor *colGraph*)

Create a Graph Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure

Parameters

in	*rElem*	Rectangle coordinates defining checkbox size
in	*nFontId*	Font ID to use for graph area
in	*pBuf*	Ptr to data buffer (already allocated) with size (nBufMax) int16_t
in	*nBufRows*	Maximum number of points in buffer
in	*colGraph*	Color of the graph

Returns

Pointer to Element reference or NULL if failure

9.6.1.3 bool gslc_ElemXGraphDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a Graph element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.6.1.4 void gslc_ElemXGraphScrollSet (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* uint8_t *nScrollPos,* uint8_t *nScrollMax*)

Set the graph scroll position (nScrollPos) as a fraction of nScrollMax.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nScrollPos*	New scroll position
in	*nScrollMax*	Maximum scroll position

Returns

none

9.6.1.5 void gslc_ElemXGraphSetRange (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nYMin,* int16_t *nYMax*)

Set the graph's drawing range.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nYMin*	Minimum Y value to draw
in	*nYMax*	Maximum Y value to draw

Returns

none

9.6.1.6 void gslc_ElemXGraphSetStyle (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_teXGraphStyle *eStyle,* uint8_t *nMargin*)

Set the graph's additional drawing characteristics.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*eStyle*	Drawing style for the graph
in	*nMargin*	Margin to provide around graph area inside frame

Returns

none

9.6.2 Variable Documentation

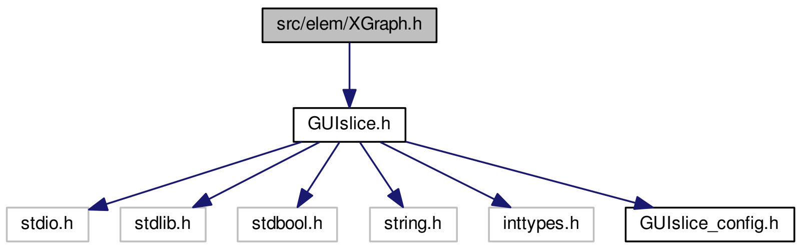
9.6.2.1 const char GSLC_PMEM ERRSTR_NULL[ ]

9.6.2.2 const char GSLC_PMEM ERRSTR_PXD_NULL[ ]

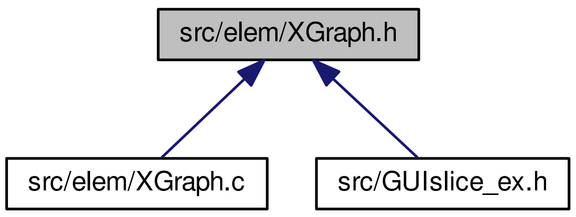
9.7 src/elem/XGraph.h File Reference

```
#include "GUIslice.h"
```

Include dependency graph for XGraph.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsXGraph

    Extended data for Graph element.

Macros

- #define GSLC_TYPEX_GRAPH

Enumerations

- enum gslc_teXGraphStyle { GSLCX_GRAPH_STYLE_DOT, GSLCX_GRAPH_STYLE_LINE, GSLCX_GR↵ APH_STYLE_FILL }

    Gauge drawing style.

Functions

- gslc_tsElemRef ∗ gslc_ElemXGraphCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsX↩ Graph ∗pXData, gslc_tsRect rElem, int16_t nFontId, int16_t ∗pBuf, uint16_t nBufRows, gslc_tsColor col↩ Graph)

    Create a Graph Element.
- void gslc_ElemXGraphSetStyle (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teXGraphStyle eStyle, uint8_t nMargin)

    Set the graph's additional drawing characteristics.
- void gslc_ElemXGraphSetRange (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nYMin, int16_t n↩ YMax)

    Set the graph's drawing range.
- bool gslc_ElemXGraphDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

    Draw a Graph element on the screen.
- void gslc_ElemXGraphAdd (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nVal)

    Add a value to the graph at the latest position.
- void gslc_ElemXGraphScrollSet (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nScrollPos, uint8_t nScrollMax)

    Set the graph scroll position (nScrollPos) as a fraction of nScrollMax.

9.7.1 Macro Definition Documentation

9.7.1.1 #define GSLC_TYPEX_GRAPH

9.7.2 Enumeration Type Documentation

9.7.2.1 enum gslc_teXGraphStyle

Gauge drawing style.

Enumerator

GSLCX_GRAPH_STYLE_DOT	Dot.
GSLCX_GRAPH_STYLE_LINE	Line.
GSLCX_GRAPH_STYLE_FILL	Filled.

9.7.3 Function Documentation

9.7.3.1 void gslc_ElemXGraphAdd (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nVal*)

Add a value to the graph at the latest position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nVal*	Data value to add

Returns

none

9.7.3.2 gslc_tsElemRef∗ gslc_ElemXGraphCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXGraph ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nFontId,* int16_t ∗ *pBuf,* uint16_t *nBufRows,* gslc_tsColor *colGraph*)

Create a Graph Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*nFontId*	Font ID to use for graph area
in	*pBuf*	Ptr to data buffer (already allocated) with size (nBufMax) int16_t
in	*nBufRows*	Maximum number of points in buffer
in	*colGraph*	Color of the graph

Returns

Pointer to Element reference or NULL if failure

9.7.3.3 bool gslc_ElemXGraphDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a Graph element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.7.3.4 void gslc_ElemXGraphScrollSet (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* uint8_t *nScrollPos,* uint8_t *nScrollMax*)

Set the graph scroll position (nScrollPos) as a fraction of nScrollMax.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nScrollPos*	New scroll position
in	*nScrollMax*	Maximum scroll position

Returns

none

9.7.3.5 void gslc_ElemXGraphSetRange (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int16_t *nYMin*, int16_t *nYMax*)

Set the graph's drawing range.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nYMin*	Minimum Y value to draw
in	*nYMax*	Maximum Y value to draw

Returns

none

9.7.3.6 void gslc_ElemXGraphSetStyle (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_teXGraphStyle *eStyle*, uint8_t *nMargin*)

Set the graph's additional drawing characteristics.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*eStyle*	Drawing style for the graph
in	*nMargin*	Margin to provide around graph area inside frame

Returns

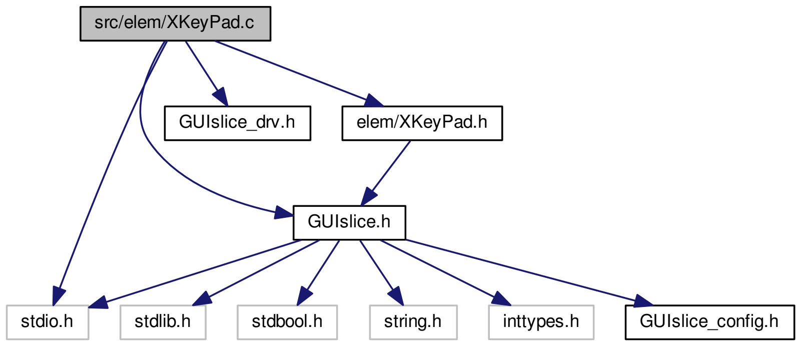
none

9.8 src/elem/XKeyPad.c File Reference

```
#include "GUIslice.h"
```

```
#include "GUIslice_drv.h"
#include "elem/XKeyPad.h"
#include <stdio.h>
```

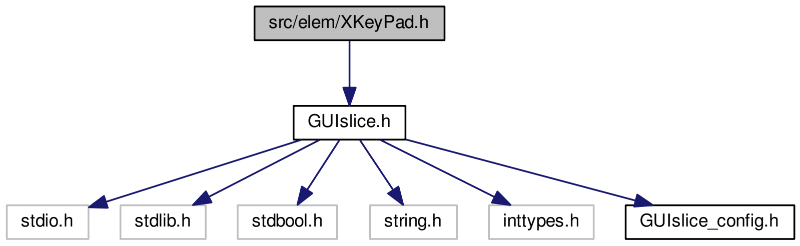
Include dependency graph for XKeyPad.c:



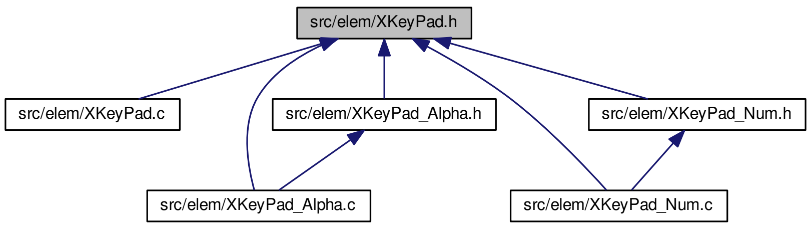
9.9 src/elem/XKeyPad.h File Reference

```
#include "GUIslice.h"
```

Include dependency graph for XKeyPad.h:



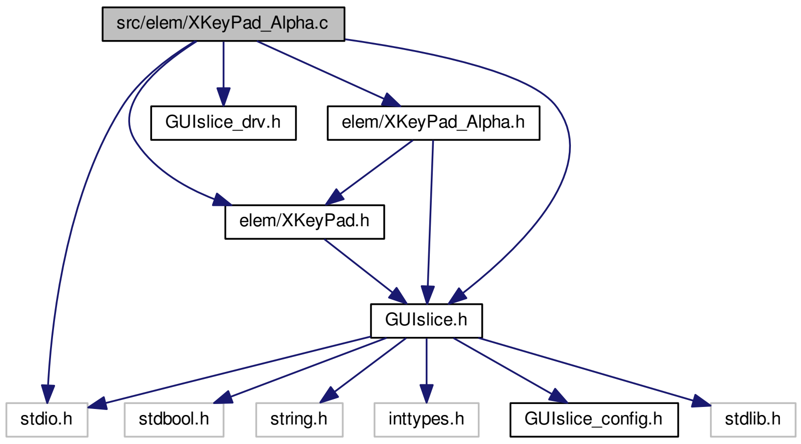
This graph shows which files directly or indirectly include this file:



9.10 src/elem/XKeyPad_Alpha.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XKeyPad.h"
#include "elem/XKeyPad_Alpha.h"
#include <stdio.h>
```

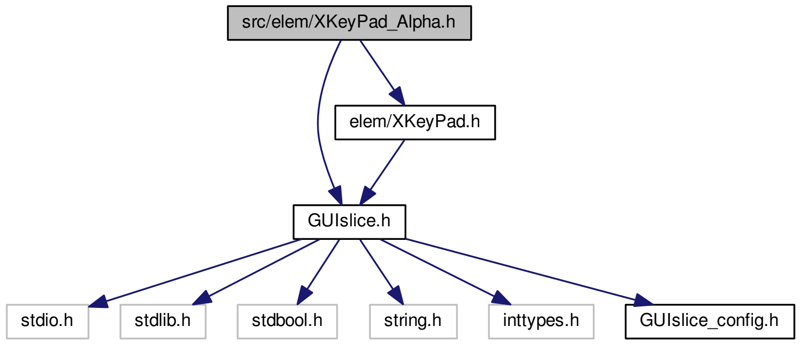
Include dependency graph for XKeyPad_Alpha.c:



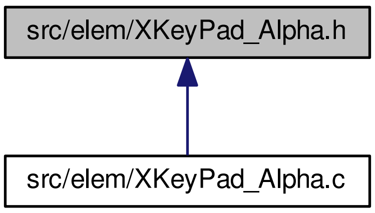
9.11 src/elem/XKeyPad_Alpha.h File Reference

```
#include "GUIslice.h"
#include "elem/XKeyPad.h"
```

Include dependency graph for XKeyPad_Alpha.h:



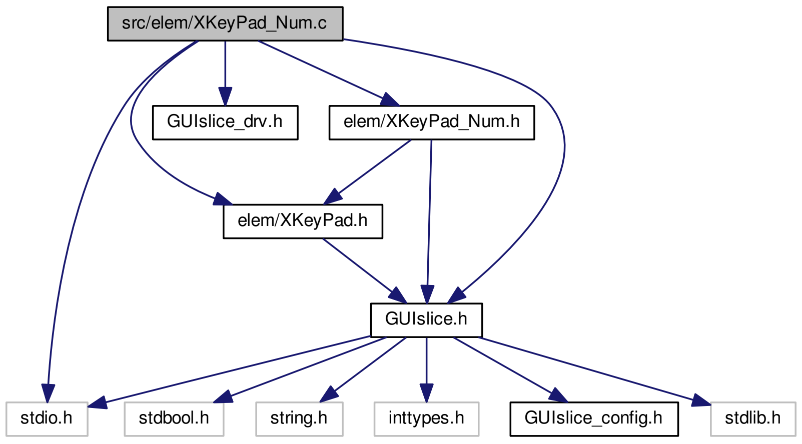
This graph shows which files directly or indirectly include this file:



9.12 src/elem/XKeyPad_Num.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XKeyPad.h"
#include "elem/XKeyPad_Num.h"
#include <stdio.h>
```

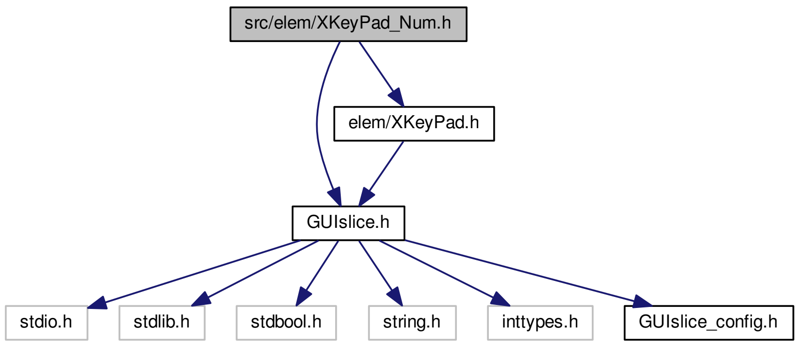
Include dependency graph for XKeyPad_Num.c:



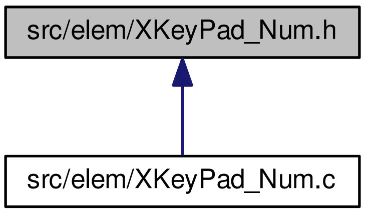
9.13 src/elem/XKeyPad_Num.h File Reference

```
#include "GUIslice.h"
#include "elem/XKeyPad.h"
```

Include dependency graph for XKeyPad_Num.h:



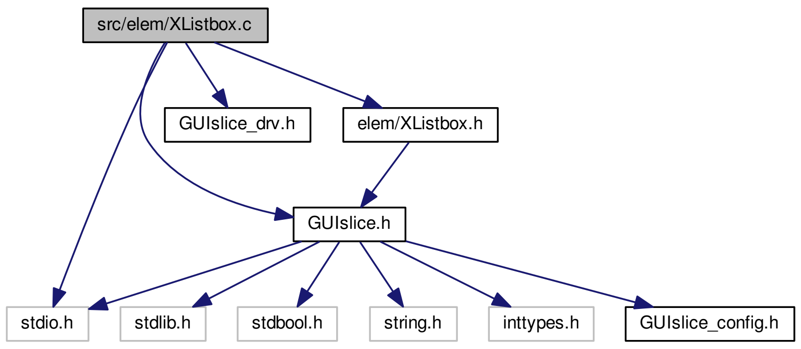
This graph shows which files directly or indirectly include this file:



9.14 src/elem/XListbox.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XListbox.h"
#include <stdio.h>
```

Include dependency graph for XListbox.c:



Macros

- #define XLISTBOX_MAX_STR

Functions

- bool gslc_ElemXListboxRecalcSize (gslc_tsXListbox ∗pListbox, gslc_tsRect rElem)
- void gslc_ElemXListboxSetSize (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nRows, int8_t nCols)

  Configure the number of rows & columns to display in the listbox.
- void gslc_ElemXListboxSetMargin (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nMarginW, int8_t nMarginH)

  Configure the margin inside the listbox.
- void gslc_ElemXListboxItemsSetTxtMargin (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nMarginW, int8_t nMarginH)

  Configure the text margin inside the listbox items.
- void gslc_ElemXListboxItemsSetSize (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nItemW, int16↩_t nItemH)

  Configure the size of the listbox items.
- void gslc_ElemXListboxItemsSetGap (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nGap, gslc_ts↩Color colGap)

  Configure the gap between listbox items.
- void gslc_ElemXListboxReset (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Empty the listbox of all items.
- bool gslc_ElemXListboxAddItem (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, const char ∗pStrItem)

  Add an item to the listbox.
- bool gslc_ElemXListboxGetItem (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nItemCurSel, char ∗pStrItem, uint8_t nStrItemLen)

  Get the indexed listbox item.
- int16_t gslc_ElemXListboxGetItemCnt (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the number of items in the listbox.
- gslc_tsElemRef ∗ gslc_ElemXListboxCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_ts↩XListbox ∗pXData, gslc_tsRect rElem, int16_t nFontId, uint8_t ∗pBufItems, uint16_t nBufItemsMax, int16_t nItemDefault)

  Create a Listbox Element.

- bool gslc_ElemXListboxDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

    Draw a Listbox element on the screen.
- bool gslc_ElemXListboxTouch (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)

    Handle touch events to Listbox element.
- int16_t gslc_ElemXListboxGetSel (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

    Get a Listbox element's current selection.
- bool gslc_ElemXListboxSetSel (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nItemCurSel)

    Set a Listbox element's current selection.
- bool gslc_ElemXListboxSetScrollPos (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint16_t nScrollPos)

    Set the Listbox scroll position.
- void gslc_ElemXListboxSetSelFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_XLISTBO←X_SEL funcCb)

    Assign the selection callback function for a Listbox.

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.14.1 Macro Definition Documentation

9.14.1.1 #define XLISTBOX_MAX_STR

9.14.2 Function Documentation

9.14.2.1 bool gslc_ElemXListboxAddItem (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* const char ∗ *pStrItem*)

Add an item to the listbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*pStrItem*	String to use when creating the listbox item

Returns

true if OK, false if fail (eg. insufficient buffer storage)

9.14.2.2 gslc_tsElemRef∗ gslc_ElemXListboxCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXListbox ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nFontId,* uint8_t ∗ *pBufItems,* uint16_t *nBufItemsMax,* int16_t *nSelDefault*)

Create a Listbox Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*nFontId*	Font ID for item display
in	*pBufItems*	Pointer to buffer that will contain list of items
in	*nBufItemsMax*	Max size of buffer for list of items (pBufItems)
in	*nSelDefault*	Default item to select

Returns

Pointer to Element reference or NULL if failure

9.14.2.3 bool gslc_ElemXListboxDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a Listbox element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.14.2.4 bool gslc_ElemXListboxGetItem (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nItemCurSel,* char ∗ *pStrItem,* uint8_t *nStrItemLen*)

Get the indexed listbox item.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nItemCurSel*	Item index to fetch
out	*pStrItem*	Ptr to the string buffer to receive the item
in	*nStrItemLen*	Maximum buffer length of pStrItem

Returns

true if success, false if fail (eg. can't locate item)

9.14.2.5 int16_t gslc_ElemXListboxGetItemCnt (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*)

Get the number of items in the listbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update

Returns

Number of items

9.14.2.6 int16_t gslc_ElemXListboxGetSel (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*)

Get a Listbox element's current selection.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

Current Listbox selection (or -1 if none)

9.14.2.7 void gslc_ElemXListboxItemsSetGap (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int8_t *nGap*, gslc_tsColor *colGap*)

Configure the gap between listbox items.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nGap*	Set the gap between listbox items (0 for none)
in	*colGap*	Set the color of the gap between listbox items

Returns

none

9.14.2.8 void gslc_ElemXListboxItemsSetSize (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nItemW,* int16_t *nItemH*)

Configure the size of the listbox items.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nItemW*	Set the width of a listbox item (or -1 to auto-size)
in	*nItemH*	Set the height of a listbox item

Returns

none

9.14.2.9 void gslc_ElemXListboxItemsSetTxtMargin (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int8_t *nMarginW,* int8_t *nMarginH*)

Configure the text margin inside the listbox items.

- Defines the region bewteen the listbox item and the text labels

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nMarginW*	Set the margin (horizontal) inside the item (0 for none)
in	*nMarginH*	Set the margin (horizontal) inside the item (0 for none)

Returns

none

9.14.2.10 bool gslc_ElemXListboxRecalcSize (gslc_tsXListbox ∗ *pListbox,* gslc_tsRect *rElem*)

9.14.2.11 void gslc_ElemXListboxReset (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Empty the listbox of all items.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update

Returns

none

9.14.2.12 void gslc_ElemXListboxSetMargin (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int8_t *nMarginW*, int8_t *nMarginH*)

Configure the margin inside the listbox.

- Defines the region bewteen the element rect and the inner listbox items

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nMarginW*	Set the margin (horizontal) inside the listbox (0 for none)
in	*nMarginH*	Set the margin (horizontal) inside the listbox (0 for none)

Returns

none

9.14.2.13 bool gslc_ElemXListboxSetScrollPos (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, uint16_t *nScrollPos*)

Set the Listbox scroll position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nScrollPos*	Scroll the listbox so that the nScrollPos item is at the top (0 default)

Returns

true if success, false if fail

9.14.2.14 bool gslc_ElemXListboxSetSel (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int16_t *nItemCurSel*)

Set a Listbox element's current selection.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nItemCurSel*	Listbox item to select (or -1 for none)

Returns

true if success, false if fail

9.14.2.15 void gslc_ElemXListboxSetSelFunc (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, GSLC_CB_XLISTBOX_SEL *funcCb*)

Assign the selection callback function for a Listbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*funcCb*	Function pointer to selection routine (or NULL for none)

Returns

none

9.14.2.16 void gslc_ElemXListboxSetSize (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int8_t *nRows*, int8_t *nCols*)

Configure the number of rows & columns to display in the listbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nRows*	Number of rows (>= 1, or XLISTBOX_SIZE_AUTO to base on content)
in	*nCols*	Number of columns (>= 1)

Returns

none

9.14.2.17 bool gslc_ElemXListboxTouch (void ∗ *pvGui*, void ∗ *pvElemRef*, gslc_teTouch *eTouch*, int16_t *nRelX*, int16_t *nRelY*)

Handle touch events to Listbox element.

- Called from gslc_ElemSendEventTouch()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element ref (typecast to gslc_tsElemRef∗)
in	*eTouch*	Touch event type
in	*nRelX*	Touch X coord relative to element
in	*nRelY*	Touch Y coord relative to element

Returns

true if success, false otherwise

9.14.3 Variable Documentation

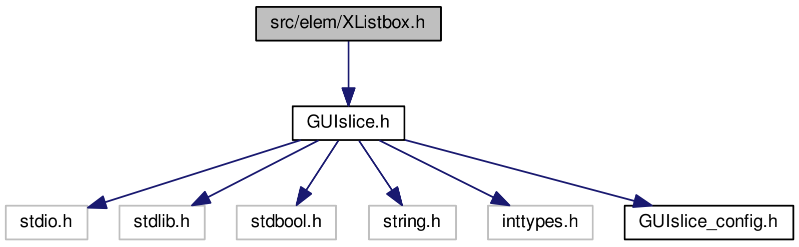
9.14.3.1 const char GSLC_PMEM ERRSTR_NULL[ ]

9.14.3.2 const char GSLC_PMEM ERRSTR_PXD_NULL[ ]

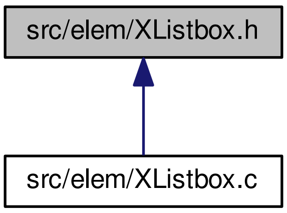
9.15 src/elem/XListbox.h File Reference

```
#include "GUIslice.h"
```

Include dependency graph for XListbox.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsXListbox

  Extended data for Listbox element.

Macros

- #define GSLC_TYPEX_LISTBOX
- #define XLISTBOX_SEL_NONE
- #define XLISTBOX_SIZE_AUTO
- #define XLISTBOX_BUF_OH_R

Typedefs

- typedef bool(∗ GSLC_CB_XLISTBOX_SEL) (void ∗pvGui, void ∗pvElem, int16_t nSel)

  Callback function for Listbox feedback.

Functions

- gslc_tsElemRef ∗ gslc_ElemXListboxCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_ts←XListbox ∗pXData, gslc_tsRect rElem, int16_t nFontId, uint8_t ∗pBufItems, uint16_t nBufItemsMax, int16_t nSelDefault)

  Create a Listbox Element.
- void gslc_ElemXListboxSetSize (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nRows, int8_t nCols)

  Configure the number of rows & columns to display in the listbox.
- void gslc_ElemXListboxSetMargin (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nMarginW, int8_t nMarginH)

  Configure the margin inside the listbox.
- void gslc_ElemXListboxItemsSetTxtMargin (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nMarginW, int8_t nMarginH)

  Configure the text margin inside the listbox items.
- void gslc_ElemXListboxItemsSetSize (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nItemW, int16←_t nItemH)

  Configure the size of the listbox items.
- void gslc_ElemXListboxItemsSetGap (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nGap, gslc_ts←Color colGap)

  Configure the gap between listbox items.
- void gslc_ElemXListboxReset (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Empty the listbox of all items.
- bool gslc_ElemXListboxAddItem (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, const char ∗pStrItem)

  Add an item to the listbox.
- bool gslc_ElemXListboxGetItem (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nItemCurSel, char ∗pStrItem, uint8_t nStrItemLen)

  Get the indexed listbox item.
- int16_t gslc_ElemXListboxGetItemCnt (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the number of items in the listbox.
- bool gslc_ElemXListboxDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a Listbox element on the screen.
- bool gslc_ElemXListboxTouch (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)

  Handle touch events to Listbox element.
- int16_t gslc_ElemXListboxGetSel (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get a Listbox element's current selection.
- bool gslc_ElemXListboxSetSel (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nItemCurSel)

  Set a Listbox element's current selection.
- bool gslc_ElemXListboxSetScrollPos (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint16_t nScrollPos)

  Set the Listbox scroll position.
- void gslc_ElemXListboxSetSelFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_XLISTBO←X_SEL funcCb)

  Assign the selection callback function for a Listbox.

9.15.1 Macro Definition Documentation

9.15.1.1 #define GSLC_TYPEX_LISTBOX

9.15.1.2 #define XLISTBOX_BUF_OH_R

9.15.1.3 #define XLISTBOX_SEL_NONE

9.15.1.4 #define XLISTBOX_SIZE_AUTO

9.15.2 Typedef Documentation

9.15.2.1 typedef bool(∗ GSLC_CB_XLISTBOX_SEL) (void ∗pvGui, void ∗pvElem, int16_t nSel)

Callback function for Listbox feedback.

9.15.3 Function Documentation

9.15.3.1 bool gslc_ElemXListboxAddItem (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* const char ∗ *pStrItem*)

Add an item to the listbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*pStrItem*	String to use when creating the listbox item

Returns

true if OK, false if fail (eg. insufficient buffer storage)

9.15.3.2 gslc_tsElemRef∗ gslc_ElemXListboxCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXListbox ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nFontId,* uint8_t ∗ *pBufItems,* uint16_t *nBufItemsMax,* int16_t *nSelDefault*)

Create a Listbox Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*nFontId*	Font ID for item display

Parameters

in	*pBufItems*	Pointer to buffer that will contain list of items
in	*nBufItemsMax*	Max size of buffer for list of items (pBufItems)
in	*nSelDefault*	Default item to select

Returns

Pointer to Element reference or NULL if failure

9.15.3.3 bool gslc_ElemXListboxDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a Listbox element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.15.3.4 bool gslc_ElemXListboxGetItem (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nItemCurSel,* char ∗ *pStrItem,* uint8_t *nStrItemLen*)

Get the indexed listbox item.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nItemCurSel*	Item index to fetch
out	*pStrItem*	Ptr to the string buffer to receive the item
in	*nStrItemLen*	Maximum buffer length of pStrItem

Returns

true if success, false if fail (eg. can't locate item)

9.15.3.5 int16_t gslc_ElemXListboxGetItemCnt (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get the number of items in the listbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update

Returns

Number of items

9.15.3.6 int16_t gslc_ElemXListboxGetSel (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get a Listbox element's current selection.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

Current Listbox selection (or -1 if none)

9.15.3.7 void gslc_ElemXListboxItemsSetGap (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int8_t *nGap,* gslc_tsColor *colGap*)

Configure the gap between listbox items.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nGap*	Set the gap between listbox items (0 for none)
in	*colGap*	Set the color of the gap between listbox items

Returns

none

9.15.3.8 void gslc_ElemXListboxItemsSetSize (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nItemW,* int16_t *nItemH*)

Configure the size of the listbox items.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nItemW*	Set the width of a listbox item (or -1 to auto-size)
in	*nItemH*	Set the height of a listbox item

Returns

none

9.15.3.9 void gslc_ElemXListboxItemsSetTxtMargin (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int8_t *nMarginW,* int8_t *nMarginH*)

Configure the text margin inside the listbox items.

- Defines the region bewteen the listbox item and the text labels

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nMarginW*	Set the margin (horizontal) inside the item (0 for none)
in	*nMarginH*	Set the margin (horizontal) inside the item (0 for none)

Returns

none

9.15.3.10 void gslc_ElemXListboxReset (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Empty the listbox of all items.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update

Returns

none

9.15.3.11 void gslc_ElemXListboxSetMargin (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int8_t *nMarginW,* int8_t *nMarginH*)

Configure the margin inside the listbox.

- Defines the region bewteen the element rect and the inner listbox items

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nMarginW*	Set the margin (horizontal) inside the listbox (0 for none)
in	*nMarginH*	Set the margin (horizontal) inside the listbox (0 for none)

Returns

none

9.15.3.12 bool gslc_ElemXListboxSetScrollPos (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* uint16_t *nScrollPos*)

Set the Listbox scroll position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nScrollPos*	Scroll the listbox so that the nScrollPos item is at the top (0 default)

Returns

true if success, false if fail

9.15.3.13 bool gslc_ElemXListboxSetSel (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nItemCurSel*)

Set a Listbox element's current selection.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nItemCurSel*	Listbox item to select (or -1 for none)

Returns

true if success, false if fail

9.15.3.14 void gslc_ElemXListboxSetSelFunc (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* GSLC_CB_XLISTBOX_SEL *funcCb*)

Assign the selection callback function for a Listbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*funcCb*	Function pointer to selection routine (or NULL for none)

Returns

none

9.15.3.15 void gslc_ElemXListboxSetSize (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int8_t *nRows*, int8_t *nCols*)

Configure the number of rows & columns to display in the listbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Ptr to Element Reference to update
in	*nRows*	Number of rows (>= 1, or XLISTBOX_SIZE_AUTO to base on content)
in	*nCols*	Number of columns (>= 1)

Returns

none

9.15.3.16 bool gslc_ElemXListboxTouch (void ∗ *pvGui*, void ∗ *pvElemRef*, gslc_teTouch *eTouch*, int16_t *nRelX*, int16_t *nRelY*)

Handle touch events to Listbox element.

- Called from gslc_ElemSendEventTouch()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element ref (typecast to gslc_tsElemRef∗)
in	*eTouch*	Touch event type
in	*nRelX*	Touch X coord relative to element
in	*nRelY*	Touch Y coord relative to element

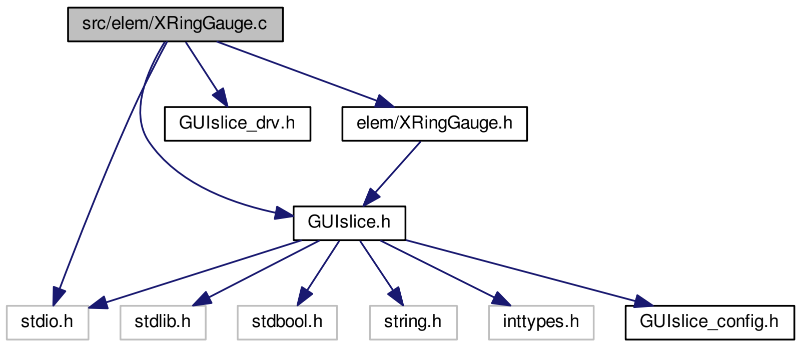
Returns

true if success, false otherwise

9.16 src/elem/XRingGauge.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XRingGauge.h"
#include <stdio.h>
```

Include dependency graph for XRingGauge.c:



Functions

- gslc_tsElemRef ∗ gslc_ElemXRingGaugeCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_←tsXRingGauge ∗pXData, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

  Create an XRingGauge element.
- bool gslc_ElemXRingGaugeDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw the template element on the screen.
- void gslc_ElemXRingGaugeSetPos (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nPos)

  Set an XRingGauge element's current position.
- void gslc_ElemXRingGaugeSetRange (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nPosMin, int16_t nPosMax)
- void gslc_ElemXRingGaugeSetThickness (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nThickness)
- void gslc_ElemXRingGaugeSetRingColorFlat (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colActive)
- void gslc_ElemXRingGaugeSetRingColorGradient (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_ts←Color colStart, gslc_tsColor colEnd)
- void gslc_ElemXRingGaugeSetRingColorInactive (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_ts←Color colInactive)
- void gslc_ElemXRingGaugeSetQuality (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint16_t nSegments)

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.16.1 Function Documentation

9.16.1.1 gslc_tsElemRef∗ gslc_ElemXRingGaugeCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXRingGauge ∗ *pXData,* gslc_tsRect *rElem,* char ∗ *pStrBuf,* uint8_t *nStrBufMax,* int16_t *nFontId*)

Create an XRingGauge element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining element size
in	*pStrBuf*	String buffer to use for gauge inner text
in	*nStrBufMax*	Maximum length of string buffer (pStrBuf)
in	*nFontId*	Font ID to use for text display

Returns

Pointer to Element reference or NULL if failure

9.16.1.2 bool gslc_ElemXRingGaugeDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw the template element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.16.1.3 void gslc_ElemXRingGaugeSetPos (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nPos*)

Set an XRingGauge element's current position.

Parameters

in	*pGui*	Pointer to GUI
in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nPos*	New position value

Returns

none

9.16.1.4 void gslc_ElemXRingGaugeSetQuality (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* uint16_t *nSegments*)

9.16.1.5 void gslc_ElemXRingGaugeSetRange (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nPosMin,* int16_t *nPosMax*)

Todo

9.16.1.6 void gslc_ElemXRingGaugeSetRingColorFlat (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_tsColor *colActive*)

9.16.1.7 void gslc_ElemXRingGaugeSetRingColorGradient (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_tsColor *colStart,* gslc_tsColor *colEnd*)

9.16.1.8 void gslc_ElemXRingGaugeSetRingColorInactive (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_tsColor *colInactive*)

9.16.1.9 void gslc_ElemXRingGaugeSetThickness (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int8_t *nThickness*)

9.16.2 Variable Documentation

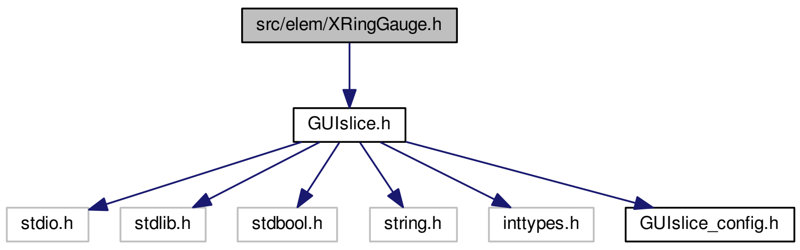
9.16.2.1 const char GSLC_PMEM ERRSTR_NULL[ ]

9.16.2.2 const char GSLC_PMEM ERRSTR_PXD_NULL[ ]

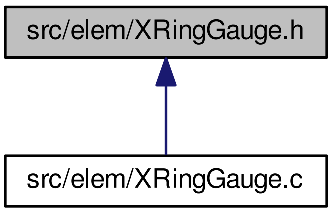
9.17 src/elem/XRingGauge.h File Reference

```
#include "GUIslice.h"
```

Include dependency graph for XRingGauge.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsXRingGauge

    Extended data for XRingGauge element.

Macros

- #define GSLC_TYPEX_RING
- #define XRING_STR_MAX

Functions

- gslc_tsElemRef ∗ gslc_ElemXRingGaugeCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_↩ tsXRingGauge ∗pXData, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

    Create an XRingGauge element.
- bool gslc_ElemXRingGaugeDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

    Draw the template element on the screen.
- void gslc_ElemXRingGaugeSetPos (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nPos)

    Set an XRingGauge element's current position.
- void gslc_ElemXRingGaugeSetRange (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nPosMin, int16_t nPosMax)
- void gslc_ElemXRingGaugeSetThickness (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int8_t nThickness)
- void gslc_ElemXRingGaugeSetQuality (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint16_t nSegments)
- void gslc_ElemXRingGaugeSetRingColorInactive (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_ts↩ Color colInactive)
- void gslc_ElemXRingGaugeSetRingColorFlat (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colActive)
- void gslc_ElemXRingGaugeSetRingColorGradient (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_ts↩ Color colStart, gslc_tsColor colEnd)

9.17.1 Macro Definition Documentation

9.17.1.1 #define GSLC_TYPEX_RING

9.17.1.2 #define XRING_STR_MAX

9.17.2 Function Documentation

9.17.2.1 gslc_tsElemRef∗ gslc_ElemXRingGaugeCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXRingGauge ∗ *pXData,* gslc_tsRect *rElem,* char ∗ *pStrBuf,* uint8_t *nStrBufMax,* int16_t *nFontId*)

Create an XRingGauge element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining element size
in	*pStrBuf*	String buffer to use for gauge inner text
in	*nStrBufMax*	Maximum length of string buffer (pStrBuf)
in	*nFontId*	Font ID to use for text display

Returns

Pointer to Element reference or NULL if failure

9.17.2.2 bool gslc_ElemXRingGaugeDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw the template element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.17.2.3 void gslc_ElemXRingGaugeSetPos (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int16_t *nPos*)

Set an XRingGauge element's current position.

Parameters

in	*pGui*	Pointer to GUI
in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nPos*	New position value

Returns

none

9.17.2.4 void gslc_ElemXRingGaugeSetQuality (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, uint16_t *nSegments*)

9.17.2.5 void gslc_ElemXRingGaugeSetRange (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int16_t *nPosMin*, int16_t *nPosMax*)

Todo

9.17.2.6 void gslc_ElemXRingGaugeSetRingColorFlat (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_tsColor *colActive*)

9.17.2.7 void gslc_ElemXRingGaugeSetRingColorGradient (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_tsColor *colStart*, gslc_tsColor *colEnd*)

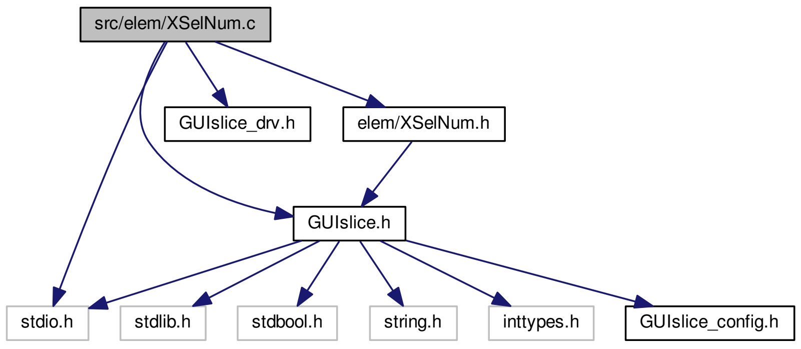
9.17.2.8 void gslc_ElemXRingGaugeSetRingColorInactive (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, gslc_tsColor *colInactive*)

9.17.2.9 void gslc_ElemXRingGaugeSetThickness (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, int8_t *nThickness*)

9.18 src/elem/XSelNum.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XSelNum.h"
#include <stdio.h>
```

Include dependency graph for XSelNum.c:



Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.18.1 Variable Documentation

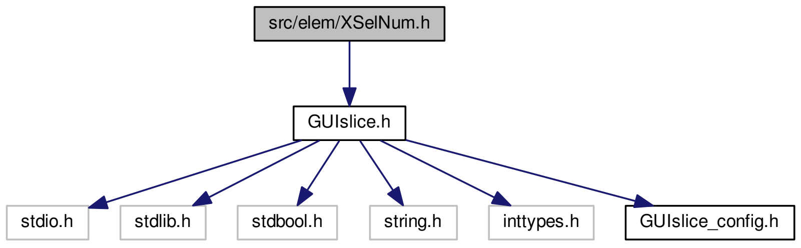
9.18.1.1 const char **GSLC_PMEM** ERRSTR_NULL[ ]

9.18.1.2 const char **GSLC_PMEM** ERRSTR_PXD_NULL[ ]

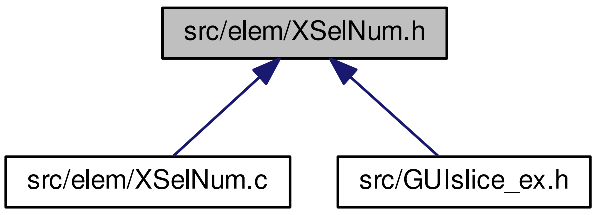
9.19 src/elem/XSelNum.h File Reference

```
#include "GUIslice.h"
```

Include dependency graph for XSelNum.h:

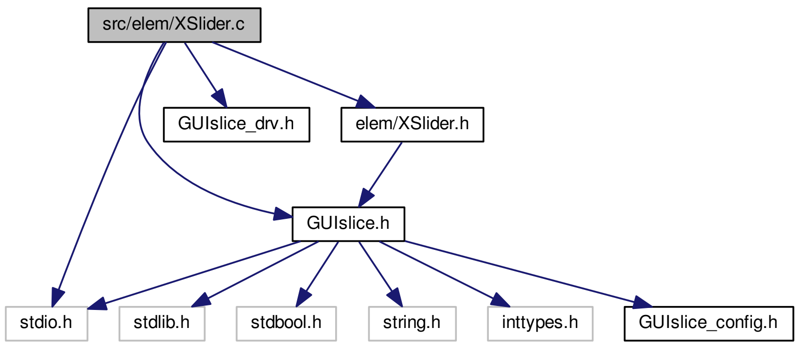


This graph shows which files directly or indirectly include this file:



9.20 src/elem/XSlider.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XSlider.h"
#include <stdio.h>
```
Include dependency graph for XSlider.c:



Functions

- gslc_tsElemRef ∗ gslc_ElemXSliderCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsX↩ Slider ∗pXData, gslc_tsRect rElem, int16_t nPosMin, int16_t nPosMax, int16_t nPos, uint16_t nThumbSz, bool bVert)

  Create a Slider Element.
- void gslc_ElemXSliderSetStyle (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bTrim, gslc_tsColor col↩ Trim, uint16_t nTickDiv, int16_t nTickLen, gslc_tsColor colTick)

  Set a Slider element's current position.
- int gslc_ElemXSliderGetPos (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get a Slider element's current position.

- void gslc_ElemXSliderSetPos (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nPos)

  Set a Slider element's current position.
- void gslc_ElemXSliderSetPosFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_XSLIDER_↩ POS funcCb)

  Assign the position callback function for a slider.
- bool gslc_ElemXSliderDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a Slider element on the screen.
- bool gslc_ElemXSliderTouch (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)

  Handle touch events to Slider element.

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.20.1 Function Documentation

9.20.1.1 gslc_tsElemRef∗ gslc_ElemXSliderCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXSlider ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nPosMin,* int16_t *nPosMax,* int16_t *nPos,* uint16_t *nThumbSz,* bool *bVert*)

Create a Slider Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*nPosMin*	Minimum position value
in	*nPosMax*	Maximum position value
in	*nPos*	Starting position value
in	*nThumbSz*	Size of the thumb control
in	*bVert*	Orientation (true for vertical)

Returns

Pointer to Element reference or NULL if failure

9.20.1.2 bool gslc_ElemXSliderDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a Slider element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.20.1.3 int gslc_ElemXSliderGetPos (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get a Slider element's current position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

Current slider position

9.20.1.4 void gslc_ElemXSliderSetPos (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nPos*)

Set a Slider element's current position.

Parameters

in	*pGui*	Pointer to GUI
in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nPos*	New position value

Returns

none

9.20.1.5 void gslc_ElemXSliderSetPosFunc (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* GSLC_CB_XSLIDER_POS *funcCb*)

Assign the position callback function for a slider.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*funcCb*	Function pointer to position routine (or NULL for none)

Returns

none

9.20.1.6 void gslc_ElemXSliderSetStyle (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* bool *bTrim,* gslc_tsColor *colTrim,* uint16_t *nTickDiv,* int16_t *nTickLen,* gslc_tsColor *colTick*)

Set a Slider element's current position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bTrim*	Show a colored trim?
in	*colTrim*	Color of trim
in	*nTickDiv*	Number of tick divisions to show (0 for none)
in	*nTickLen*	Length of tickmarks
in	*colTick*	Color of ticks

Returns

none

9.20.1.7 bool gslc_ElemXSliderTouch (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teTouch *eTouch,* int16_t *nRelX,* int16_t *nRelY*)

Handle touch events to Slider element.

- Called from gslc_ElemSendEventTouch()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element ref (typecast to gslc_tsElemRef∗)
in	*eTouch*	Touch event type
in	*nRelX*	Touch X coord relative to element
in	*nRelY*	Touch Y coord relative to element

Returns

true if success, false otherwise

9.20.2 Variable Documentation

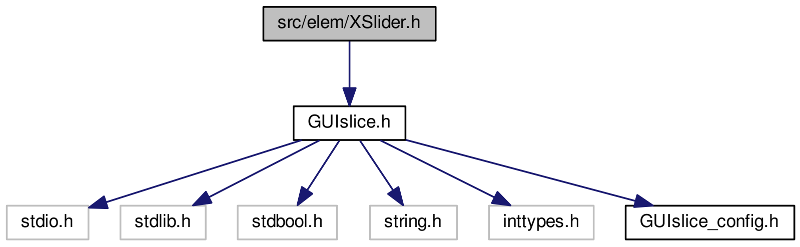
9.20.2.1 const char GSLC_PMEM ERRSTR_NULL[ ]

9.20.2.2 const char GSLC_PMEM ERRSTR_PXD_NULL[ ]

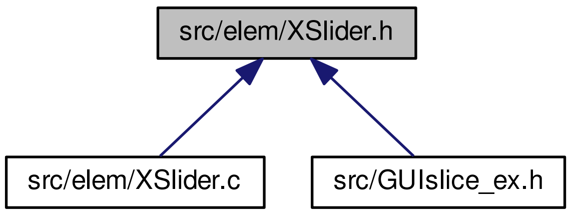
9.21 src/elem/XSlider.h File Reference

```
#include "GUIslice.h"
```

Include dependency graph for XSlider.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsXSlider

  Extended data for Slider element.

Macros

- #define GSLC_TYPEX_SLIDER
- #define gslc_ElemXSliderCreate_P(pGui, nElemId, nPage, nX, nY, nW, nH, nPosMin_, nPosMax_, nPos_, nThumbSz_, bVert_, colFrame_, colFill_)

  Create a Slider Element in Flash.

Typedefs

- typedef bool(∗ GSLC_CB_XSLIDER_POS) (void ∗pvGui, void ∗pvElem, int16_t nPos)

  Callback function for slider feedback.

Functions

- gslc_tsElemRef ∗ gslc_ElemXSliderCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsX←Slider ∗pXData, gslc_tsRect rElem, int16_t nPosMin, int16_t nPosMax, int16_t nPos, uint16_t nThumbSz, bool bVert)

  Create a Slider Element.
- void gslc_ElemXSliderSetStyle (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bTrim, gslc_tsColor col←Trim, uint16_t nTickDiv, int16_t nTickLen, gslc_tsColor colTick)

  Set a Slider element's current position.
- int gslc_ElemXSliderGetPos (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get a Slider element's current position.
- void gslc_ElemXSliderSetPos (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nPos)

  Set a Slider element's current position.
- void gslc_ElemXSliderSetPosFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_XSLIDER_←POS funcCb)

  Assign the position callback function for a slider.
- bool gslc_ElemXSliderDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a Slider element on the screen.
- bool gslc_ElemXSliderTouch (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)

  Handle touch events to Slider element.

9.21.1 Macro Definition Documentation

9.21.1.1 #define gslc_ElemXSliderCreate_P(*pGui, nElemId, nPage, nX, nY, nW, nH, nPosMin_, nPosMax_, nPos_, nThumbSz_, bVert_, colFrame_, colFill_*)

Create a Slider Element in Flash.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Unique element ID to assign
in	*nPage*	Page ID to attach element to
in	*nX*	X coordinate of element
in	*nY*	Y coordinate of element
in	*nW*	Width of element
in	*nH*	Height of element

Parameters

in	*nPosMin_*	Minimum position value
in	*nPosMax_*	Maximum position value
in	*nPos_*	Starting position value
in	*nThumbSz_*	Size of the thumb control
in	*bVert_*	Orientation (true for vertical)
in	*colFrame_*	Color of the element frame
in	*colFill_*	Color of the element fill

Returns

none

9.21.1.2 #define GSLC_TYPEX_SLIDER

9.21.2 Typedef Documentation

9.21.2.1 typedef bool(∗ GSLC_CB_XSLIDER_POS) (void ∗pvGui, void ∗pvElem, int16_t nPos)

Callback function for slider feedback.

9.21.3 Function Documentation

9.21.3.1 gslc_tsElemRef∗ gslc_ElemXSliderCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXSlider ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nPosMin,* int16_t *nPosMax,* int16_t *nPos,* uint16_t *nThumbSz,* bool *bVert*)

Create a Slider Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*nPosMin*	Minimum position value
in	*nPosMax*	Maximum position value
in	*nPos*	Starting position value
in	*nThumbSz*	Size of the thumb control
in	*bVert*	Orientation (true for vertical)

Returns

Pointer to Element reference or NULL if failure

9.21.3.2 bool gslc_ElemXSliderDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a Slider element on the screen.

- Called from gslc_ElemDraw()

Parameters

`in`	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
`in`	*pvElemRef*	Void ptr to Element (typecast to gslc_tsElemRef∗)
`in`	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.21.3.3 int gslc_ElemXSliderGetPos (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Get a Slider element's current position.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference

Returns

Current slider position

9.21.3.4 void gslc_ElemXSliderSetPos (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* int16_t *nPos*)

Set a Slider element's current position.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*nPos*	New position value

Returns

none

9.21.3.5 void gslc_ElemXSliderSetPosFunc (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* GSLC_CB_XSLIDER_POS *funcCb*)

Assign the position callback function for a slider.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*funcCb*	Function pointer to position routine (or NULL for none)

Returns

none

9.21.3.6 void gslc_ElemXSliderSetStyle (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* bool *bTrim,* gslc_tsColor *colTrim,* uint16_t *nTickDiv,* int16_t *nTickLen,* gslc_tsColor *colTick*)

Set a Slider element's current position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bTrim*	Show a colored trim?
in	*colTrim*	Color of trim
in	*nTickDiv*	Number of tick divisions to show (0 for none)
in	*nTickLen*	Length of tickmarks
in	*colTick*	Color of ticks

Returns

none

9.21.3.7 bool gslc_ElemXSliderTouch (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teTouch *eTouch,* int16_t *nRelX,* int16_t *nRelY*)

Handle touch events to Slider element.

- Called from gslc_ElemSendEventTouch()

Parameters

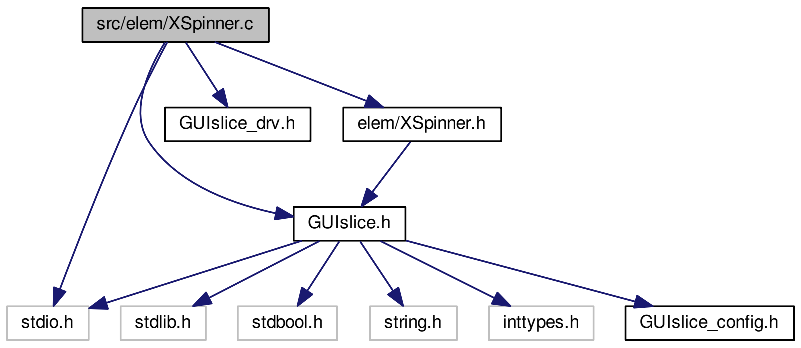
in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element ref (typecast to gslc_tsElemRef∗)
in	*eTouch*	Touch event type
in	*nRelX*	Touch X coord relative to element
in	*nRelY*	Touch Y coord relative to element

Returns

true if success, false otherwise

9.22 src/elem/XSpinner.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XSpinner.h"
#include <stdio.h>
```
Include dependency graph for XSpinner.c:



Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

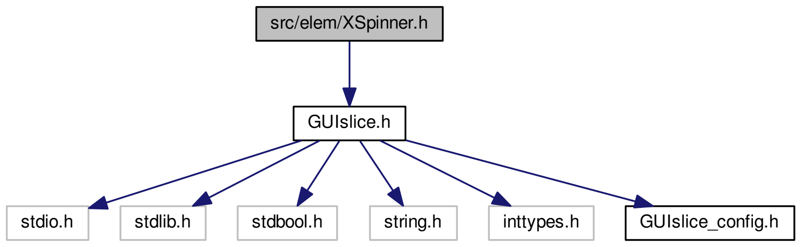
9.22.1 Variable Documentation

9.22.1.1 const char **GSLC_PMEM** ERRSTR_NULL[ ]

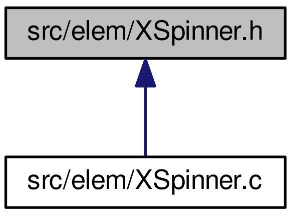
9.22.1.2 const char **GSLC_PMEM** ERRSTR_PXD_NULL[ ]

9.23 src/elem/XSpinner.h File Reference

```
#include "GUIslice.h"
```
Include dependency graph for XSpinner.h:



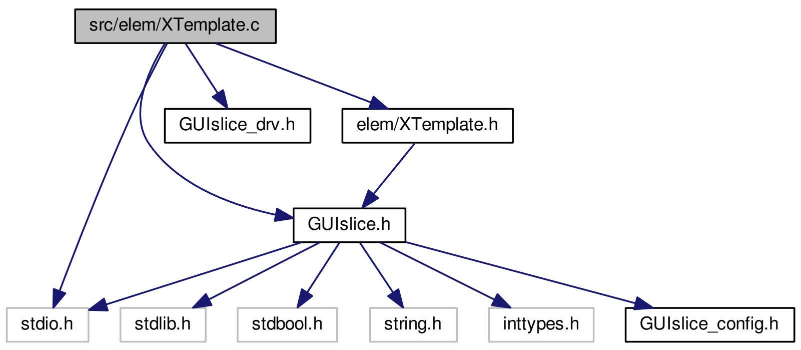
This graph shows which files directly or indirectly include this file:



9.24 src/elem/XTemplate.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XTemplate.h"
#include <stdio.h>
```

Include dependency graph for XTemplate.c:



Functions

- gslc_tsElemRef ∗ gslc_ElemXTemplateCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_ts←XTemplate ∗pXData, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

  Create an Extended Text Field Element.
- bool gslc_ElemXTemplateDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw the template element on the screen.
- bool gslc_ElemXTemplateTouch (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16←_t nRelY)

  Handle touch events to template element.

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.24.1 Function Documentation

9.24.1.1 gslc_tsElemRef∗ gslc_ElemXTemplateCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXTemplate ∗ *pXData,* gslc_tsRect *rElem,* char ∗ *pStrBuf,* uint8_t *nStrBufMax,* int16_t *nFontId*)

Create an Extended Text Field Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*nPosMin*	Minimum position value
in	*nPosMax*	Maximum position value
in	*nPos*	Starting position value
in	*nThumbSz*	Size of the thumb control
in	*bVert*	Orientation (true for vertical)

Returns

Pointer to Element reference or NULL if failure

9.24.1.2 bool gslc_ElemXTemplateDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw the template element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.24.1.3 bool gslc_ElemXTemplateTouch (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teTouch *eTouch,* int16_t *nRelX,* int16_t *nRelY*)

Handle touch events to template element.

- Called from gslc_ElemSendEventTouch()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element ref (typecast to gslc_tsElemRef∗)
in	*eTouch*	Touch event type
in	*nRelX*	Touch X coord relative to element
in	*nRelY*	Touch Y coord relative to element

Returns

true if success, false otherwise

9.24.2 Variable Documentation

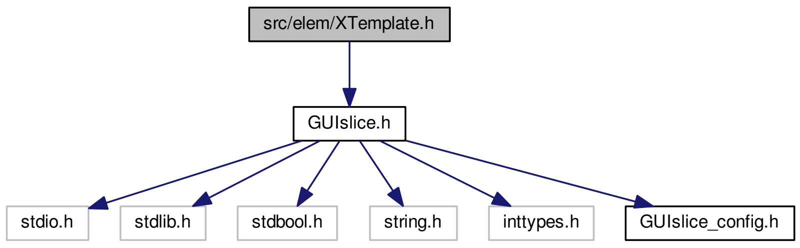
9.24.2.1 const char GSLC_PMEM ERRSTR_NULL[ ]

9.24.2.2 const char GSLC_PMEM ERRSTR_PXD_NULL[ ]

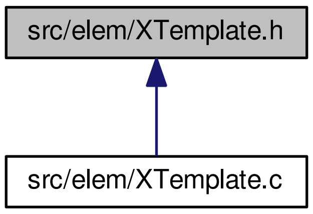
9.25 src/elem/XTemplate.h File Reference

```
#include "GUIslice.h"
```

Include dependency graph for XTemplate.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsXTemplate

  Callback function for slider feedback.

Macros

- #define GSLC_TYPEX_TEMPLATE

Functions

- gslc_tsElemRef ∗ gslc_ElemXTemplateCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_ts←XTemplate ∗pXData, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

  Create an Extended Text Field Element.
- bool gslc_ElemXTemplateDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw the template element on the screen.
- bool gslc_ElemXTemplateTouch (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16←_t nRelY)

  Handle touch events to template element.

9.25.1 Macro Definition Documentation

9.25.1.1 #define GSLC_TYPEX_TEMPLATE

9.25.2 Function Documentation

9.25.2.1 gslc_tsElemRef∗ gslc_ElemXTemplateCreate (gslc_tsGui ∗ *pGui*, int16_t *nElemId*, int16_t *nPage*, gslc_tsXTemplate ∗ *pXData*, gslc_tsRect *rElem*, char ∗ *pStrBuf*, uint8_t *nStrBufMax*, int16_t *nFontId*)

Create an Extended Text Field Element.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining checkbox size
in	*nPosMin*	Minimum position value
in	*nPosMax*	Maximum position value
in	*nPos*	Starting position value
in	*nThumbSz*	Size of the thumb control
in	*bVert*	Orientation (true for vertical)

Returns

Pointer to Element reference or NULL if failure

9.25.2.2 bool gslc_ElemXTemplateDraw (void ∗ *pvGui*, void ∗ *pvElemRef*, gslc_teRedrawType *eRedraw*)

Draw the template element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.25.2.3 bool gslc_ElemXTemplateTouch (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teTouch *eTouch,* int16_t *nRelX,* int16_t *nRelY*)

Handle touch events to template element.

- Called from gslc_ElemSendEventTouch()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element ref (typecast to gslc_tsElemRef∗)
in	*eTouch*	Touch event type
in	*nRelX*	Touch X coord relative to element
in	*nRelY*	Touch Y coord relative to element

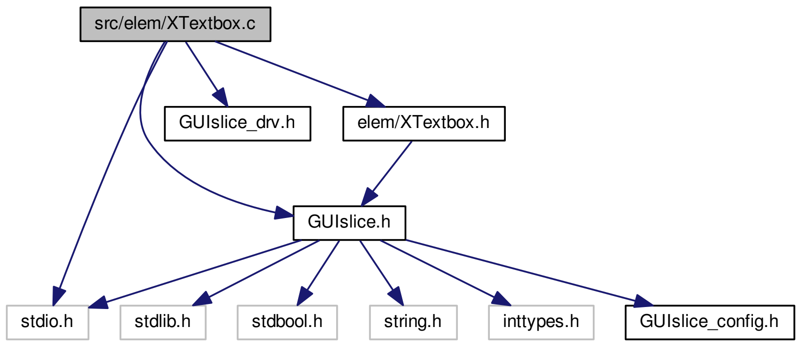
Returns

true if success, false otherwise

9.26 src/elem/XTextbox.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include "elem/XTextbox.h"
#include <stdio.h>
```

Include dependency graph for XTextbox.c:



Functions

- gslc_tsElemRef ∗ gslc_ElemXTextboxCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsX←Textbox ∗pXData, gslc_tsRect rElem, int16_t nFontId, char ∗pBuf, uint16_t nBufRows, uint16_t nBufCols)

  Create a Textbox Element.
- void gslc_ElemXTextboxReset (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

Reset the contents of the textbox.

- void gslc_ElemXTextboxLineWrAdv (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)
- void gslc_ElemXTextboxScrollSet (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nScrollPos, uint8←_t nScrollMax)

  Set the textbox scroll position (nScrollPos) as a fraction of nScrollMax.
- void gslc_ElemXTextboxBufAdd (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, unsigned char chNew, bool bAdvance)
- void gslc_ElemXTextboxColSet (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor nCol)

  Insert a color set code into the current buffer position.
- void gslc_ElemXTextboxColReset (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Insert a color reset code into the current buffer position.
- void gslc_ElemXTextboxWrapSet (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bWrapEn)

  Enable or disable line wrap within textbox.
- void gslc_ElemXTextboxAdd (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, char ∗pTxt)

  Add a text string to the textbox.
- bool gslc_ElemXTextboxDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a Textbox element on the screen.

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.26.1 Function Documentation

9.26.1.1 void gslc_ElemXTextboxAdd (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* char ∗ *pTxt*)

Add a text string to the textbox.

- If it includes a newline then the buffer will advance to the next row
- If wrap has been enabled, then a newline will be forced

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*pTxt*	Pointer to text string (null-terminated)

Returns

none

9.26.1.2 void gslc_ElemXTextboxBufAdd (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* unsigned char *chNew,* bool *bAdvance*)

9.26.1.3 void gslc_ElemXTextboxColReset (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Insert a color reset code into the current buffer position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

none

9.26.1.4 void gslc_ElemXTextboxColSet (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_tsColor *nCol*)

Insert a color set code into the current buffer position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nCol*	Color to assign for next text written to textbox

Returns

none

9.26.1.5 gslc_tsElemRef∗ gslc_ElemXTextboxCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXTextbox ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nFontId,* char ∗ *pBuf,* uint16_t *nBufRows,* uint16_t *nBufCols*)

Create a Textbox Element.

- The textbox is a scrolling window designed for displaying multi-line text using a monospaced font. A character buffer is defined by nBufRows∗nBufCols to capture the added text. If the allocation buffer is larger than the display size (defined by rElem), then a scrollbar will be shown.
- Support for changing color within a row can be enabled with GSLC_FEATURE_XTEXTBOX_EMBED 1
- Note that each color change command will consume 4 of the available "column" bytes.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining textbox size
in	*nFontId*	Font ID to use for text area
in	*pBuf*	Ptr to text buffer (already allocated) with size (nBufRows∗nBufCols) chars
in	*nBufRows*	Number of rows in buffer
in	*nBufCols*	Number of columns in buffer (incl special codes)

Returns

Pointer to Element reference or NULL if failure

9.26.1.6 bool gslc_ElemXTextboxDraw (void ∗ *pvGui*, void ∗ *pvElemRef*, gslc_teRedrawType *eRedraw*)

Draw a Textbox element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.26.1.7 void gslc_ElemXTextboxLineWrAdv (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*)

9.26.1.8 void gslc_ElemXTextboxReset (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*)

Reset the contents of the textbox.

- Clears the buffer and resets the position

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

none

9.26.1.9 void gslc_ElemXTextboxScrollSet (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, uint8_t *nScrollPos*, uint8_t *nScrollMax*)

Set the textbox scroll position (nScrollPos) as a fraction of nScrollMax.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nScrollPos*	New scroll position
in	*nScrollMax*	Maximum scroll position

Returns

none

9.26.1.10 void gslc_ElemXTextboxWrapSet (gslc_tsGui ∗ *pGui*, gslc_tsElemRef ∗ *pElemRef*, bool *bWrapEn*)

Enable or disable line wrap within textbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bWrapEn*	Enable line wrap if true

Returns

none

9.26.2 Variable Documentation

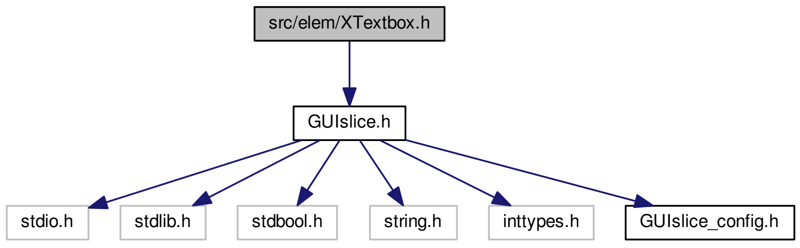
9.26.2.1 const char GSLC_PMEM ERRSTR_NULL[ ]

9.26.2.2 const char GSLC_PMEM ERRSTR_PXD_NULL[ ]

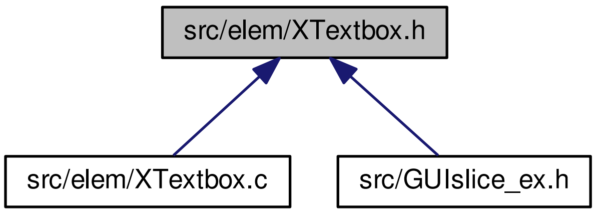
9.27 src/elem/XTextbox.h File Reference

`#include "GUIslice.h"`

Include dependency graph for XTextbox.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsXTextbox

  Extended data for Textbox element.

Macros

- #define GSLC_TYPEX_TEXTBOX
- #define GSLC_XTEXTBOX_CODE_COL_SET

  Definitions for textbox special inline codes.
- #define GSLC_XTEXTBOX_CODE_COL_RESET
- #define XTEXTBOX_REDRAW_NONE
- #define XTEXTBOX_REDRAW_ALL

Functions

- gslc_tsElemRef ∗ gslc_ElemXTextboxCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsX←Textbox ∗pXData, gslc_tsRect rElem, int16_t nFontId, char ∗pBuf, uint16_t nBufRows, uint16_t nBufCols)

  Create a Textbox Element.
- void gslc_ElemXTextboxReset (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Reset the contents of the textbox.
- bool gslc_ElemXTextboxDraw (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Draw a Textbox element on the screen.
- void gslc_ElemXTextboxAdd (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, char ∗pTxt)

  Add a text string to the textbox.
- void gslc_ElemXTextboxColSet (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor nCol)

  Insert a color set code into the current buffer position.
- void gslc_ElemXTextboxColReset (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Insert a color reset code into the current buffer position.
- void gslc_ElemXTextboxWrapSet (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bWrapEn)

  Enable or disable line wrap within textbox.
- void gslc_ElemXTextboxScrollSet (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nScrollPos, uint8←_t nScrollMax)

  Set the textbox scroll position (nScrollPos) as a fraction of nScrollMax.

9.27.1 Macro Definition Documentation

9.27.1.1 #define GSLC_TYPEX_TEXTBOX

9.27.1.2 #define GSLC_XTEXTBOX_CODE_COL_RESET

9.27.1.3 #define GSLC_XTEXTBOX_CODE_COL_SET

Definitions for textbox special inline codes.

9.27.1.4 #define XTEXTBOX_REDRAW_ALL

9.27.1.5 #define XTEXTBOX_REDRAW_NONE

9.27.2 Function Documentation

9.27.2.1 void gslc_ElemXTextboxAdd (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* char ∗ *pTxt*)

Add a text string to the textbox.

- If it includes a newline then the buffer will advance to the next row
- If wrap has been enabled, then a newline will be forced

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference
`in`	*pTxt*	Pointer to text string (null-terminated)

Returns

none

9.27.2.2 void gslc_ElemXTextboxColReset (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Insert a color reset code into the current buffer position.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElemRef*	Pointer to Element reference

Returns

none

9.27.2.3 void gslc_ElemXTextboxColSet (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* gslc_tsColor *nCol*)

Insert a color set code into the current buffer position.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nCol*	Color to assign for next text written to textbox

Returns

none

9.27.2.4 gslc_tsElemRef∗ gslc_ElemXTextboxCreate (gslc_tsGui ∗ *pGui,* int16_t *nElemId,* int16_t *nPage,* gslc_tsXTextbox ∗ *pXData,* gslc_tsRect *rElem,* int16_t *nFontId,* char ∗ *pBuf,* uint16_t *nBufRows,* uint16_t *nBufCols*)

Create a Textbox Element.

- The textbox is a scrolling window designed for displaying multi-line text using a monospaced font. A character buffer is defined by nBufRows∗nBufCols to capture the added text. If the allocation buffer is larger than the display size (defined by rElem), then a scrollbar will be shown.
- Support for changing color within a row can be enabled with GSLC_FEATURE_XTEXTBOX_EMBED 1
- Note that each color change command will consume 4 of the available "column" bytes.

Parameters

in	*pGui*	Pointer to GUI
in	*nElemId*	Element ID to assign (0..16383 or GSLC_ID_AUTO to autogen)
in	*nPage*	Page ID to attach element to
in	*pXData*	Ptr to extended element data structure
in	*rElem*	Rectangle coordinates defining textbox size
in	*nFontId*	Font ID to use for text area
in	*pBuf*	Ptr to text buffer (already allocated) with size (nBufRows∗nBufCols) chars
in	*nBufRows*	Number of rows in buffer
in	*nBufCols*	Number of columns in buffer (incl special codes)

Returns

Pointer to Element reference or NULL if failure

9.27.2.5 bool gslc_ElemXTextboxDraw (void ∗ *pvGui,* void ∗ *pvElemRef,* gslc_teRedrawType *eRedraw*)

Draw a Textbox element on the screen.

- Called from gslc_ElemDraw()

Parameters

in	*pvGui*	Void ptr to GUI (typecast to gslc_tsGui∗)
in	*pvElemRef*	Void ptr to Element reference (typecast to gslc_tsElemRef∗)
in	*eRedraw*	Redraw mode

Returns

true if success, false otherwise

9.27.2.6 void gslc_ElemXTextboxReset (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef*)

Reset the contents of the textbox.

- Clears the buffer and resets the position

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference

Returns

none

9.27.2.7 void gslc_ElemXTextboxScrollSet (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* uint8_t *nScrollPos,* uint8_t *nScrollMax*)

Set the textbox scroll position (nScrollPos) as a fraction of nScrollMax.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*nScrollPos*	New scroll position
in	*nScrollMax*	Maximum scroll position

Returns

none

9.27.2.8 void gslc_ElemXTextboxWrapSet (gslc_tsGui ∗ *pGui,* gslc_tsElemRef ∗ *pElemRef,* bool *bWrapEn*)

Enable or disable line wrap within textbox.

Parameters

in	*pGui*	Pointer to GUI
in	*pElemRef*	Pointer to Element reference
in	*bWrapEn*	Enable line wrap if true

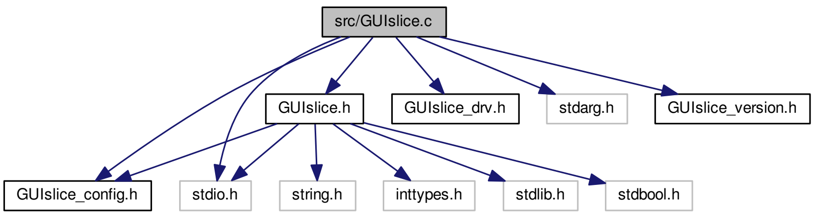
Returns

none

9.28 src/GUIslice.c File Reference

```
#include "GUIslice_config.h"
#include "GUIslice.h"
#include "GUIslice_drv.h"
#include <stdio.h>
#include <stdarg.h>
#include "GUIslice_version.h"
```

Include dependency graph for GUIslice.c:



Enumerations

- enum gslc_teDebugPrintState {
  GSLC_DEBUG2_PRINT_NORM, GSLC_DEBUG2_PRINT_TOKEN, GSLC_DEBUG2_PRINT_UINT16, G←
  SLC_DEBUG2_PRINT_CHAR,
  GSLC_DEBUG2_PRINT_STR, GSLC_DEBUG2_PRINT_STR_P }

Functions

- char ∗ gslc_GetVer (gslc_tsGui ∗pGui)

    Get the GUIslice version number.
- const char ∗ gslc_GetNameDisp (gslc_tsGui ∗pGui)

    Get the GUIslice display driver name.
- const char ∗ gslc_GetNameTouch (gslc_tsGui ∗pGui)

    Get the GUIslice touch driver name.
- bool gslc_Init (gslc_tsGui ∗pGui, void ∗pvDriver, gslc_tsPage ∗asPage, uint8_t nMaxPage, gslc_tsFont ∗as←Font, uint8_t nMaxFont)

    Initialize the GUIslice library.
- void gslc_SetPinPollFunc (gslc_tsGui ∗pGui, GSLC_CB_PIN_POLL pfunc)
- void gslc_InitInputMap (gslc_tsGui ∗pGui, gslc_tsInputMap ∗asInputMap, uint8_t nInputMapMax)
- void gslc_InputMapAdd (gslc_tsGui ∗pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc_te←Action eAction, int16_t nActionVal)
- bool gslc_InputMapLookup (gslc_tsGui ∗pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc←_teAction ∗peAction, int16_t ∗pnActionVal)
- void gslc_InitDebug (GSLC_CB_DEBUG_OUT pfunc)

    Initialize debug output.
- void gslc_DebugPrintf (const char ∗pFmt,...)

    Optimized printf routine for GUIslice debug/error output.
- void gslc_Quit (gslc_tsGui ∗pGui)

    Exit the GUIslice environment.
- void gslc_Update (gslc_tsGui ∗pGui)

    Perform main GUIslice handling functions.
- gslc_tsEvent gslc_EventCreate (gslc_tsGui ∗pGui, gslc_teEventType eType, uint8_t nSubType, void ∗pv←Scope, void ∗pvData)

    Create an event structure.
- bool gslc_IsInRect (int16_t nSelX, int16_t nSelY, gslc_tsRect rRect)

    Determine if a coordinate is inside of a rectangular region.
- bool gslc_IsInWH (int16_t nSelX, int16_t nSelY, uint16_t nWidth, uint16_t nHeight)

    Determine if a coordinate is inside of a width x height region.
- void gslc_OrderCoord (int16_t ∗pnX0, int16_t ∗pnY0, int16_t ∗pnX1, int16_t ∗pnY1)
- bool gslc_ClipPt (gslc_tsRect ∗pClipRect, int16_t nX, int16_t nY)

    Perform basic clipping of a single point to a clipping region.
- bool gslc_ClipLine (gslc_tsRect ∗pClipRect, int16_t ∗pnX0, int16_t ∗pnY0, int16_t ∗pnX1, int16_t ∗pnY1)

    Perform basic clipping of a line to a clipping region.
- bool gslc_ClipRect (gslc_tsRect ∗pClipRect, gslc_tsRect ∗pRect)

    Perform basic clipping of a rectangle to a clipping region.
- gslc_tsImgRef gslc_ResetImage ()

    Create a blank image reference structure.
- gslc_tsImgRef gslc_GetImageFromFile (const char ∗pFname, gslc_teImgRefFlags eFmt)

    Create an image reference to a bitmap file in LINUX filesystem.
- gslc_tsImgRef gslc_GetImageFromSD (const char ∗pFname, gslc_teImgRefFlags eFmt)

    Create an image reference to a bitmap file in SD card.
- gslc_tsImgRef gslc_GetImageFromRam (unsigned char ∗pImgBuf, gslc_teImgRefFlags eFmt)

    Create an image reference to a bitmap in SRAM.
- gslc_tsImgRef gslc_GetImageFromProg (const unsigned char ∗pImgBuf, gslc_teImgRefFlags eFmt)

    Create an image reference to a bitmap in program memory (PROGMEM)
- int16_t gslc_sinFX (int16_t n64Ang)

    Calculate fixed-point sine function from fractional degrees.
- int16_t gslc_cosFX (int16_t n64Ang)

Calculate fixed-point cosine function from fractional degrees.

- void gslc_PolarToXY (uint16_t nRad, int16_t n64Ang, int16_t ∗nDX, int16_t ∗nDY)

  Convert polar coordinate to cartesian.
- gslc_tsColor gslc_ColorBlend2 (gslc_tsColor colStart, gslc_tsColor colEnd, uint16_t nMidAmt, uint16_t n←BlendAmt)

  Create a color based on a blend between two colors.
- gslc_tsColor gslc_ColorBlend3 (gslc_tsColor colStart, gslc_tsColor colMid, gslc_tsColor colEnd, uint16_t n←MidAmt, uint16_t nBlendAmt)

  Create a color based on a blend between three colors.
- bool gslc_ColorEqual (gslc_tsColor a, gslc_tsColor b)

  Check whether two colors are equal.
- void gslc_DrawSetPixel (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, gslc_tsColor nCol)

  Set a pixel on the active screen to the given color with lock.
- void gslc_DrawLine (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)

  Draw an arbitrary line using Bresenham's algorithm.
- void gslc_DrawLineH (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nW, gslc_tsColor nCol)

  Draw a horizontal line.
- void gslc_DrawLineV (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nH, gslc_tsColor nCol)

  Draw a vertical line.
- void gslc_DrawLinePolar (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nRadStart, uint16_t nRadEnd, int16_t n64Ang, gslc_tsColor nCol)

  Draw a polar ray segment.
- void gslc_DrawFrameRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a framed rectangle.
- void gslc_DrawFrameRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

  Draw a framed rounded rectangle.
- void gslc_DrawFillRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a filled rectangle.
- void gslc_DrawFillRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

  Draw a filled rounded rectangle.
- gslc_tsRect gslc_ExpandRect (gslc_tsRect rRect, int16_t nExpandW, int16_t nExpandH)

  Expand or contract a rectangle in width and/or height (equal amounts on both side), based on the centerpoint of the rectangle.
- void gslc_UnionRect (gslc_tsRect ∗pRect, gslc_tsRect rAddRect)

  Expand a rect to include another rect.
- void gslc_InvalidateRgnReset (gslc_tsGui ∗pGui)

  Reset the invalidation region.
- void gslc_InvalidateRgnScreen (gslc_tsGui ∗pGui)

  Mark the entire screen as invalidated.
- void gslc_InvalidateRgnPage (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage)

  Include an entire page (eg.
- void gslc_InvalidateRgnAdd (gslc_tsGui ∗pGui, gslc_tsRect rAddRect)

  Add a rectangular region to the invalidation region.
- void gslc_DrawFrameCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)

  Draw a framed circle.
- void gslc_DrawFillCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor n←Col)

  Draw a filled circle.
- void gslc_DrawFrameTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

  Draw a framed triangle.

- void gslc_SwapCoords (int16_t ∗pnXa, int16_t ∗pnYa, int16_t ∗pnXb, int16_t ∗pnYb)
- void gslc_DrawFillTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

  Draw a filled triangle.
- void gslc_DrawFrameQuad (gslc_tsGui ∗pGui, gslc_tsPt ∗psPt, gslc_tsColor nCol)

  Draw a framed quadrilateral.
- void gslc_DrawFillQuad (gslc_tsGui ∗pGui, gslc_tsPt ∗psPt, gslc_tsColor nCol)

  Draw a filled quadrilateral.
- bool gslc_FontSetBase (gslc_tsGui ∗pGui, uint8_t nFontInd, int16_t nFontId, gslc_teFontRefType eFontRef↩ Type, const void ∗pvFontRef, uint16_t nFontSz)
- bool gslc_FontSet (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefType eFontRefType, const void ∗pv↩ FontRef, uint16_t nFontSz)

  Load a font into the local font cache and store as font ID (nFontId)
- bool gslc_FontAdd (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefType eFontRefType, const void ∗pv↩ FontRef, uint16_t nFontSz)

  Load a font into the local font cache and assign font ID (nFontId).
- gslc_tsFont ∗ gslc_FontGet (gslc_tsGui ∗pGui, int16_t nFontId)

  Fetch a font from its ID value.
- bool gslc_FontSetMode (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefMode eFontMode)

  Set a font's mode.
- bool gslc_PageEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for a page.
- void gslc_PageAdd (gslc_tsGui ∗pGui, int16_t nPageId, gslc_tsElem ∗psElem, uint16_t nMaxElem, gslc_↩ tsElemRef ∗psElemRef, uint16_t nMaxElemRef)

  Add a page to the GUI.
- int gslc_GetPageCur (gslc_tsGui ∗pGui)

  Fetch the current page ID.
- void gslc_SetStackPage (gslc_tsGui ∗pGui, uint8_t nStackPos, int16_t nPageId)

  Assign a page to the page stack.
- void gslc_SetStackState (gslc_tsGui ∗pGui, uint8_t nStackPos, bool bActive, bool bDoDraw)

  Change the status of a page in a page stack.
- void gslc_SetPageBase (gslc_tsGui ∗pGui, int16_t nPageId)

  Assigns a page for the base layer in the page stack.
- void gslc_SetPageCur (gslc_tsGui ∗pGui, int16_t nPageId)

  Select a page for the current layer in the page stack.
- void gslc_SetPageOverlay (gslc_tsGui ∗pGui, int16_t nPageId)

  Select a page for the overlay layer in the page stack.
- void gslc_PopupShow (gslc_tsGui ∗pGui, int16_t nPageId, bool bModal)

  Show a popup dialog.
- void gslc_PopupHide (gslc_tsGui ∗pGui)

  Hides the currently active popup dialog.
- void gslc_PageRedrawSet (gslc_tsGui ∗pGui, bool bRedraw)

  Update the need-redraw status for the current page.
- bool gslc_PageRedrawGet (gslc_tsGui ∗pGui)

  Get the need-redraw status for the current page.
- void gslc_PageRedrawCalc (gslc_tsGui ∗pGui)

  Perform a redraw calculation on the page to determine if additional elements should also be redrawn.
- void gslc_PageRedrawGo (gslc_tsGui ∗pGui)

  Redraw all elements on the active page.
- void gslc_PageFlipSet (gslc_tsGui ∗pGui, bool bNeeded)

  Indicate whether the screen requires page flip.

- bool gslc_PageFlipGet (gslc_tsGui ∗pGui)

  Get state of pending page flip state.
- void gslc_PageFlipGo (gslc_tsGui ∗pGui)

  Update the visible screen if page has been marked for flipping.
- gslc_tsPage ∗ gslc_PageFindById (gslc_tsGui ∗pGui, int16_t nPageId)

  Find a page in the GUI by its ID.
- gslc_tsElemRef ∗ gslc_PageFindElemById (gslc_tsGui ∗pGui, int16_t nPageId, int16_t nElemId)

  Find an element in the GUI by its Page ID and Element ID.
- int16_t gslc_PageFocusStep (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, bool bNext)
- int gslc_ElemGetId (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get an Element ID from an element structure.
- uint8_t gslc_GetElemRefFlag (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nFlagMask)

  Get the flags associated with an element reference.
- void gslc_SetElemRefFlag (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nFlagMask, uint8_t n←
  FlagVal)

  Set the flags associated with an element reference.
- gslc_tsElem ∗ gslc_GetElemFromRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Returns a pointer to an element from an element reference, copying from FLASH to RAM if element is stored in PROGMEM.
- gslc_tsElem ∗ gslc_GetElemFromRefD (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nLineNum)

  Returns a pointer to an element from an element reference.
- void ∗ gslc_GetXDataFromRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nType, int16_t nLine←
  Num)

  Returns a pointer to the data structure associated with an extended element.
- void gslc_SetRoundRadius (gslc_tsGui ∗pGui, uint8_t nRadius)

  Set the global rounded radius.
- gslc_tsElemRef ∗ gslc_ElemCreateTxt (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

  Create a Text Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBtnTxt (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId, GSLC_CB_TOUCH cbTouch)

  Create a textual Button Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBtnImg (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, gslc_tsImgRef sImgRef, gslc_tsImgRef sImgRefSel, GSLC_CB_TOUCH cbTouch)

  Create a graphical Button Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBox (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect r←
  Elem)

  Create a Box Element.
- gslc_tsElemRef ∗ gslc_ElemCreateLine (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1)

  Create a Line Element.
- gslc_tsElemRef ∗ gslc_ElemCreateImg (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect r←
  Elem, gslc_tsImgRef sImgRef)

  Create an image Element.
- bool gslc_ElemEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for an element.
- void gslc_ElemDraw (gslc_tsGui ∗pGui, int16_t nPageId, int16_t nElemId)

  Draw an element to the active display.
- void gslc_DrawTxtBase (gslc_tsGui ∗pGui, char ∗pStrBuf, gslc_tsRect rTxt, gslc_tsFont ∗pTxtFont, gslc←
  _teTxtFlags eTxtFlags, int8_t eTxtAlign, gslc_tsColor colTxt, gslc_tsColor colBg, int16_t nMarginW, int16_t nMarginH)

  Draw text with full text justification.

- bool gslc_ElemDrawByRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teRedrawType eRedraw)

  Draw an element to the active display.
- void gslc_ElemSetFillEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bFillEn)

  Set the fill state for an Element.
- void gslc_ElemSetFrameEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bFrameEn)

  Set the frame state for an Element.
- void gslc_ElemSetRoundEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bRoundEn)

  Set the rounded frame/fill state for an Element.
- void gslc_ElemSetCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colFrame, gslc_tsColor colFill, gslc_tsColor colFillGlow)

  Update the common color selection for an Element.
- void gslc_ElemSetGlowCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colFrameGlow, gslc_tsColor colFillGlow, gslc_tsColor colTxtGlow)

  Update the common color selection for glowing state of an Element.
- void gslc_ElemSetGroup (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int nGroupId)

  Set the group ID for an element.
- int gslc_ElemGetGroup (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the group ID for an element.
- void gslc_ElemSetTxtAlign (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, unsigned nAlign)

  Set the alignment of a textual element (horizontal and vertical)
- void gslc_ElemSetTxtMargin (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, unsigned nMargin)

  Set the margin around of a textual element.
- void gslc_ElemSetTxtStr (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, const char ∗pStr)

  Update the text string associated with an Element.
- char ∗ gslc_ElemGetTxtStr (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Fetch the current text string associated with an Element.
- void gslc_ElemSetTxtCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colVal)

  Update the text string color associated with an Element ID.
- void gslc_ElemSetTxtMem (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teTxtFlags eFlags)

  Update the text string location in memory.
- void gslc_ElemSetTxtEnc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teTxtFlags eFlags)

  Update the text string encoding mode.
- void gslc_ElemUpdateFont (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int nFontId)

  Update the Font selected for an Element's text.
- void gslc_ElemSetRedraw (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teRedrawType eRedraw)

  Update the need-redraw status for an element.
- gslc_teRedrawType gslc_ElemGetRedraw (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the need-redraw status for an element.
- void gslc_ElemSetGlow (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bGlowing)

  Update the glowing indicator for an element.
- bool gslc_ElemGetGlow (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the glowing indicator for an element.
- void gslc_ElemSetVisible (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bVisible)

  Update the visibility status for an element.
- bool gslc_ElemGetVisible (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the visibility status for an element.
- void gslc_ElemSetGlowEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bGlowEn)

  Update the glowing enable for an element.
- bool gslc_ElemGetGlowEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the glowing enable for an element.
- void gslc_ElemSetClickEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bClickEn)

Update the click enable for an element.

- void gslc_ElemSetTouchFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_TOUCH funcCb)

  Update the touch function callback for an element.
- void gslc_ElemSetStyleFrom (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRefSrc, gslc_tsElemRef ∗pElem↩ RefDest)

  Copy style settings from one element to another.
- void gslc_ElemSetDrawFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_DRAW funcCb)

  Assign the drawing callback function for an element.
- void gslc_ElemSetTickFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_TICK funcCb)

  Assign the tick callback function for an element.
- bool gslc_ElemOwnsCoord (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nX, int16_t nY, bool b↩ OnlyClickEn)

  Determine if a coordinate is inside of an element.
- void gslc_CollectInput (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsEventTouch ∗pEventTouch)

  Handle direct input events within the element collection.
- void gslc_CollectTouch (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsEventTouch ∗pEventTouch)

  Handle touch events within the element collection.
- void gslc_TrackInput (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, gslc_teInputRawEvent eInputEvent, int16_↩ t nInputVal)

  Handles a direct input event and performs the necessary tracking, glowing and selection actions depending on the state.
- void gslc_TrackTouch (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, int16_t nX, int16_t nY, uint16_t nPress)

  Handles a touch event and performs the necessary tracking, glowing and selection actions depending on the press state.
- bool gslc_InitTouch (gslc_tsGui ∗pGui, const char ∗acDev)

  Initialize the touchscreen device driver.
- bool gslc_GetTouch (gslc_tsGui ∗pGui, int16_t ∗pnX, int16_t ∗pnY, uint16_t ∗pnPress, gslc_teInputRawEvent ∗peInputEvent, int16_t ∗pnInputVal)

  Initialize the touchscreen device driver.
- void gslc_SetTouchRemapEn (gslc_tsGui ∗pGui, bool bEn)

  Configure touchscreen remapping.
- void gslc_SetTouchRemapCal (gslc_tsGui ∗pGui, uint16_t nXMin, uint16_t nXMax, uint16_t nYMin, uint16_t nYMax)

  Configure touchscreen calibration values.
- void gslc_SetTouchRemapYX (gslc_tsGui ∗pGui, bool bSwap)

  Configure touchscreen XY swap.
- gslc_tsElem gslc_ElemCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPageId, int16_t nType, gslc_ts↩ Rect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

  Create a new element with default styling.
- bool gslc_CollectEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for an element collection.
- gslc_tsElemRef ∗ gslc_CollectElemAdd (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, const gslc_tsElem ∗p↩ Elem, gslc_teElemRefFlags eFlags)

  Add an element to a collection.
- bool gslc_CollectGetRedraw (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Determine if any elements in a collection need redraw.
- gslc_tsElemRef ∗ gslc_ElemAdd (gslc_tsGui ∗pGui, int16_t nPageId, gslc_tsElem ∗pElem, gslc_teElem↩ RefFlags eFlags)

  Add the Element to the list of generated elements in the GUI environment.
- bool gslc_SetClipRect (gslc_tsGui ∗pGui, gslc_tsRect ∗pRect)

  Set the clipping rectangle for further drawing.

- void gslc_ElemSetImage (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsImgRef sImgRef, gslc_ts←ImgRef sImgRefSel)

  Set an element to use a bitmap image.
- bool gslc_SetBkgndImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Configure the background to use a bitmap image.
- bool gslc_SetBkgndColor (gslc_tsGui ∗pGui, gslc_tsColor nCol)

  Configure the background to use a solid color.
- bool gslc_GuiRotate (gslc_tsGui ∗pGui, uint8_t nRotation)

  Dynamically change rotation, automatically adapt touchscreen axes swap/flip.
- bool gslc_ElemSendEventTouch (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRefTracked, gslc_teTouch e←Touch, int16_t nX, int16_t nY)

  Trigger an element's touch event.
- void gslc_ResetElem (gslc_tsElem ∗pElem)

  Initialize an Element struct.
- void gslc_ResetFont (gslc_tsFont ∗pFont)

  Initialize a Font struct.
- void gslc_ElemDestruct (gslc_tsElem ∗pElem)

  Free up any members associated with an element.
- void gslc_CollectDestruct (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Free up any members associated with an element collection.
- void gslc_PageDestruct (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage)

  Free up any members associated with a page.
- void gslc_GuiDestruct (gslc_tsGui ∗pGui)

  Free up any surfaces associated with the GUI, pages, collections and elements.
- void gslc_CollectReset (gslc_tsCollect ∗pCollect, gslc_tsElem ∗asElem, uint16_t nElemMax, gslc_tsElemRef ∗asElemRef, uint16_t nElemRefMax)

  Reset the members of an element collection.
- bool gslc_CollectFindFocusStep (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, bool bNext, bool ∗pbWrapped, int16_t ∗pnElemInd)
- gslc_tsElemRef ∗ gslc_CollectFindElemById (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nElemId)

  Find an element in a collection by its Element ID.
- int gslc_CollectGetNextId (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Allocate the next available Element ID in a collection.
- gslc_tsElemRef ∗ gslc_CollectGetElemRefTracked (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Get the element within a collection that is currently being tracked.
- void gslc_CollectSetElemTracked (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsElemRef ∗pElemRef)

  Set the element within a collection that is currently being tracked.
- gslc_tsElemRef ∗ gslc_CollectFindElemFromCoord (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nX, int16_t nY)

  Find an element in a collection by a coordinate coordinate.
- int16_t gslc_CollectGetFocus (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Get the element index within a collection that is currently in focus.
- void gslc_CollectSetFocus (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nElemInd)

  Set the element index within a collection that is currently in focus.

Variables

- GSLC_CB_DEBUG_OUT g_pfDebugOut

  Global debug output function.
- uint16_t m_nLUTSinF0X16 [257]
- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.28.1 Enumeration Type Documentation

9.28.1.1 enum gslc_teDebugPrintState

Enumerator

GSLC_DEBUG2_PRINT_NORM

GSLC_DEBUG2_PRINT_TOKEN

GSLC_DEBUG2_PRINT_UINT16

GSLC_DEBUG2_PRINT_CHAR

GSLC_DEBUG2_PRINT_STR

GSLC_DEBUG2_PRINT_STR_P

9.28.2 Function Documentation

9.28.2.1 bool gslc_FontSetBase (gslc_tsGui ∗ *pGui,* uint8_t *nFontInd,* int16_t *nFontId,* gslc_teFontRefType *eFontRefType,* const void ∗ *pvFontRef,* uint16_t *nFontSz*)

9.28.2.2 void gslc_OrderCoord (int16_t ∗ *pnX0,* int16_t ∗ *pnY0,* int16_t ∗ *pnX1,* int16_t ∗ *pnY1*)

9.28.2.3 void gslc_SwapCoords (int16_t ∗ *pnXa,* int16_t ∗ *pnYa,* int16_t ∗ *pnXb,* int16_t ∗ *pnYb*)

9.28.3 Variable Documentation

9.28.3.1 const char ERRSTR_NULL[ ]

9.28.3.2 const char GSLC_PMEM ERRSTR_PXD_NULL[ ]

9.28.3.3 GSLC_CB_DEBUG_OUT g_pfDebugOut

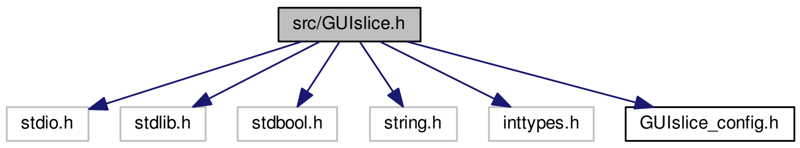
Global debug output function.

- The user assigns this function via gslc_InitDebug()

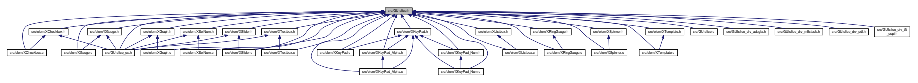
9.28.3.4 uint16_t m_nLUTSinF0X16

9.29 src/GUIslice.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <inttypes.h>
#include "GUIslice_config.h"
```
Include dependency graph for GUIslice.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gslc_tsRect

    Rectangular region. Defines X,Y corner coordinates plus dimensions.
- struct gslc_tsPt

    Define point coordinates.
- struct gslc_tsColor

    Color structure. Defines RGB triplet.
- struct gslc_tsEvent

    Event structure.
- struct gslc_tsEventTouch

    Structure used to pass touch data through event.
- struct gslc_tsFont

    Font reference structure.
- struct gslc_tsImgRef

    Image reference structure.
- struct gslc_tsElemRef

    Element reference structure.
- struct gslc_tsElem

    Element Struct.
- struct gslc_tsCollect

Element collection struct.
- struct gslc_tsPage

    Page structure.
- struct gslc_tsInputMap

    Input mapping.
- struct gslc_tsGui

    GUI structure.

Macros

- #define GSLC_PMEM
- #define GSLC_2PI
- #define GSLC_ELEM_FEA_VALID

    Element features type.
- #define GSLC_ELEM_FEA_ROUND_EN

    Element is drawn with a rounded profile.
- #define GSLC_ELEM_FEA_CLICK_EN

    Element accepts touch presses.
- #define GSLC_ELEM_FEA_GLOW_EN

    Element supports glowing state.
- #define GSLC_ELEM_FEA_FRAME_EN

    Element is drawn with a frame.
- #define GSLC_ELEM_FEA_FILL_EN

    Element is drawn with a fill.
- #define GSLC_ELEM_FEA_NONE

    Element default (no features set))
- #define GSLC_ALIGNV_TOP

    Element text alignment.
- #define GSLC_ALIGNV_MID

    Vertical align to middle.
- #define GSLC_ALIGNV_BOT

    Vertical align to bottom.
- #define GSLC_ALIGNH_LEFT

    Horizontal align to left.
- #define GSLC_ALIGNH_MID

    Horizontal align to middle.
- #define GSLC_ALIGNH_RIGHT

    Horizontal align to right.
- #define GSLC_ALIGN_TOP_LEFT

    Align to top-left.
- #define GSLC_ALIGN_TOP_MID

    Align to middle of top.
- #define GSLC_ALIGN_TOP_RIGHT

    Align to top-right.
- #define GSLC_ALIGN_MID_LEFT

    Align to middle of left side.
- #define GSLC_ALIGN_MID_MID

    Align to center.
- #define GSLC_ALIGN_MID_RIGHT

    Align to middle of right side.

- #define GSLC_ALIGN_BOT_LEFT

  Align to bottom-left.
- #define GSLC_ALIGN_BOT_MID

  Align to middle of bottom.
- #define GSLC_ALIGN_BOT_RIGHT

  Align to bottom-right.
- #define GSLC_COL_RED_DK4

  Basic color definition.
- #define GSLC_COL_RED_DK3

  Red (dark3)
- #define GSLC_COL_RED_DK2

  Red (dark2)
- #define GSLC_COL_RED_DK1

  Red (dark1)
- #define GSLC_COL_RED

  Red.
- #define GSLC_COL_RED_LT1

  Red (light1)
- #define GSLC_COL_RED_LT2

  Red (light2)
- #define GSLC_COL_RED_LT3

  Red (light3)
- #define GSLC_COL_RED_LT4

  Red (light4)
- #define GSLC_COL_GREEN_DK4

  Green (dark4)
- #define GSLC_COL_GREEN_DK3

  Green (dark3)
- #define GSLC_COL_GREEN_DK2

  Green (dark2)
- #define GSLC_COL_GREEN_DK1

  Green (dark1)
- #define GSLC_COL_GREEN

  Green.
- #define GSLC_COL_GREEN_LT1

  Green (light1)
- #define GSLC_COL_GREEN_LT2

  Green (light2)
- #define GSLC_COL_GREEN_LT3

  Green (light3)
- #define GSLC_COL_GREEN_LT4

  Green (light4)
- #define GSLC_COL_BLUE_DK4

  Blue (dark4)
- #define GSLC_COL_BLUE_DK3

  Blue (dark3)
- #define GSLC_COL_BLUE_DK2

  Blue (dark2)
- #define GSLC_COL_BLUE_DK1

  Blue (dark1)
- #define GSLC_COL_BLUE

Blue.
- #define GSLC_COL_BLUE_LT1

  Blue (light1)
- #define GSLC_COL_BLUE_LT2

  Blue (light2)
- #define GSLC_COL_BLUE_LT3

  Blue (light3)
- #define GSLC_COL_BLUE_LT4

  Blue (light4)
- #define GSLC_COL_BLACK

  Black.
- #define GSLC_COL_GRAY_DK3

  Gray (dark)
- #define GSLC_COL_GRAY_DK2

  Gray (dark)
- #define GSLC_COL_GRAY_DK1

  Gray (dark)
- #define GSLC_COL_GRAY

  Gray.
- #define GSLC_COL_GRAY_LT1

  Gray (light1)
- #define GSLC_COL_GRAY_LT2

  Gray (light2)
- #define GSLC_COL_GRAY_LT3

  Gray (light3)
- #define GSLC_COL_WHITE

  White.
- #define GSLC_COL_YELLOW

  Yellow.
- #define GSLC_COL_YELLOW_DK

  Yellow (dark)
- #define GSLC_COL_PURPLE

  Purple.
- #define GSLC_COL_CYAN

  Cyan.
- #define GSLC_COL_MAGENTA

  Magenta.
- #define GSLC_COL_TEAL

  Teal.
- #define GSLC_COL_ORANGE

  Orange.
- #define GSLC_COL_BROWN

  Brown.
- #define GSLC_COLMONO_BLACK

  Black.
- #define GSLC_COLMONO_WHITE

  White.
- #define TOUCH_ROTATION_DATA

  Additional definitions for Touch Handling These macros define the transforms used in remapping the touchscreen inputs on the basis of the GUI nRotation setting.
- #define TOUCH_ROTATION_SWAPXY(rotation)

- #define TOUCH_ROTATION_FLIPX(rotation)
- #define TOUCH_ROTATION_FLIPY(rotation)
- #define GSLC_ELEMREF_DEFAULT

  Define the default element reference flags for new elements.
- #define TOUCH_ROTATION_DATA

  Additional definitions for Touch Handling These macros define the transforms used in remapping the touchscreen inputs on the basis of the GUI nRotation setting.
- #define TOUCH_ROTATION_SWAPXY(rotation)
- #define TOUCH_ROTATION_FLIPX(rotation)
- #define TOUCH_ROTATION_FLIPY(rotation)
- #define GSLC_DEBUG_PRINT(sFmt, ...)

  Macro to enable optional debug output.
- #define GSLC_DEBUG2_PRINT(sFmt, ...)
- #define GSLC_DEBUG_PRINT_CONST(sFmt, ...)
- #define GSLC_DEBUG2_PRINT_CONST(sFmt, ...)
- #define gslc_ElemCreateTxt_P(pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, pFont, colTxt, colFrame, col←Fill, nAlignTxt, bFrameEn, bFillEn)

  Create a read-only text element.
- #define gslc_ElemCreateTxt_P_R(pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, strLength, pFont, colTxt, colFrame, colFill, nAlignTxt, bFrameEn, bFillEn)

  Create a read-write text element (element in Flash, string in RAM)
- #define gslc_ElemCreateBox_P(pGui, nElemId, nPage, nX, nY, nW, nH, colFrame, colFill, bFrameEn, bFillEn, pfuncXDraw, pfuncXTick)

  Create a read-only box element.
- #define gslc_ElemCreateLine_P(pGui, nElemId, nPage, nX0, nY0, nX1, nY1, colFill)

  Create a read-only line element.
- #define gslc_ElemCreateBtnTxt_P(pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, pFont, colTxt, colFrame, colFill, colFrameGlow, colFillGlow, nAlignTxt, bFrameEn, bFillEn, callFunc, extraData)

  Create a text button element.

Typedefs

- typedef int16_t(∗ GSLC_CB_DEBUG_OUT) (char ch)
- typedef struct gslc_tsElem gslc_tsElem

  Element Struct.
- typedef struct gslc_tsEvent gslc_tsEvent

  Event structure.
- typedef bool(∗ GSLC_CB_EVENT) (void ∗pvGui, gslc_tsEvent sEvent)

  Callback function for element drawing.
- typedef bool(∗ GSLC_CB_DRAW) (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

  Callback function for element drawing.
- typedef bool(∗ GSLC_CB_TOUCH) (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nX, int16←_t nY)

  Callback function for element touch tracking.
- typedef bool(∗ GSLC_CB_TICK) (void ∗pvGui, void ∗pvElemRef)

  Callback function for element tick.
- typedef bool(∗ GSLC_CB_PIN_POLL) (void ∗pvGui, int16_t ∗pnPinInd, int16_t ∗pnPinVal)

  Callback function for pin polling.
- typedef bool(∗ GSLC_CB_INPUT) (void ∗pvGui, void ∗pvElemRef, int16_t nStatus, void ∗pvData)

  Callback function for element input ready.
- typedef struct gslc_tsRect gslc_tsRect

Rectangular region. Defines X,Y corner coordinates plus dimensions.
- typedef struct gslc_tsPt gslc_tsPt

  Define point coordinates.
- typedef struct gslc_tsColor gslc_tsColor

  Color structure. Defines RGB triplet.
- typedef struct gslc_tsEventTouch gslc_tsEventTouch

  Structure used to pass touch data through event.

Enumerations

- enum gslc_teElemId {
  GSLC_ID_USER_BASE, GSLC_ID_NONE, GSLC_ID_AUTO, GSLC_ID_TEMP,
  GSLC_ID_AUTO_BASE }

  Element ID enumerations.
- enum gslc_tePageId { GSLC_PAGE_USER_BASE, GSLC_PAGE_NONE }

  Page ID enumerations.
- enum gslc_teStackPage { GSLC_STACK_BASE, GSLC_STACK_CUR, GSLC_STACK_OVERLAY, GSLC↩
  _STACK__MAX }

  Define page stack.
- enum gslc_teGroupId { GSLC_GROUP_ID_USER_BASE, GSLC_GROUP_ID_NONE }

  Group ID enumerations.
- enum gslc_teFontId { GSLC_FONT_USER_BASE, GSLC_FONT_NONE }

  Font ID enumerations.
- enum gslc_teElemInd { GSLC_IND_NONE, GSLC_IND_FIRST }

  Element Index enumerations.
- enum gslc_teTypeCore {
  GSLC_TYPE_NONE, GSLC_TYPE_BKGND, GSLC_TYPE_BTN, GSLC_TYPE_TXT,
  GSLC_TYPE_BOX, GSLC_TYPE_LINE, GSLC_TYPE_BASE_EXTEND }

  Element type.
- enum gslc_teInputRawEvent {
  GSLC_INPUT_NONE, GSLC_INPUT_TOUCH, GSLC_INPUT_KEY_DOWN, GSLC_INPUT_KEY_UP,
  GSLC_INPUT_PIN_ASSERT, GSLC_INPUT_PIN_DEASSERT }

  Raw input event types: touch, key, GPIOs.
- enum gslc_teAction {
  GSLC_ACTION_UNDEF, GSLC_ACTION_NONE, GSLC_ACTION_FOCUS_PREV, GSLC_ACTION_FO↩
  CUS_NEXT,
  GSLC_ACTION_SELECT, GSLC_ACTION_SET_REL, GSLC_ACTION_SET_ABS, GSLC_ACTION_DE↩
  BUG }

  GUI Action Requested These actions are usually the result of an InputMap lookup.
- enum gslc_tePin {
  GSLC_PIN_BTN_A, GSLC_PIN_BTN_A_LONG, GSLC_PIN_BTN_B, GSLC_PIN_BTN_B_LONG,
  GSLC_PIN_BTN_C, GSLC_PIN_BTN_C_LONG, GSLC_PIN_BTN_D, GSLC_PIN_BTN_D_LONG,
  GSLC_PIN_BTN_E, GSLC_PIN_BTN_E_LONG, GSLC_PIN_BTN_UP, GSLC_PIN_BTN_DOWN,
  GSLC_PIN_BTN_LEFT, GSLC_PIN_BTN_RIGHT, GSLC_PIN_BTN_SEL }

  General purpose pin/button constants.
- enum gslc_teTouch {
  GSLC_TOUCH_NONE, GSLC_TOUCH_TYPE_MASK, GSLC_TOUCH_COORD, GSLC_TOUCH_DIRECT,
  GSLC_TOUCH_SUBTYPE_MASK, GSLC_TOUCH_DOWN, GSLC_TOUCH_DOWN_IN, GSLC_TOUCH↩
  _DOWN_OUT,
  GSLC_TOUCH_UP, GSLC_TOUCH_UP_IN, GSLC_TOUCH_UP_OUT, GSLC_TOUCH_MOVE,
  GSLC_TOUCH_MOVE_IN, GSLC_TOUCH_MOVE_OUT, GSLC_TOUCH_FOCUS_ON, GSLC_TOUCH_↩
  FOCUS_OFF,
  GSLC_TOUCH_FOCUS_SELECT, GSLC_TOUCH_SET_REL, GSLC_TOUCH_SET_ABS }

Processed event from input raw events and actions.

- enum gslc_teInitStat { GSLC_INITSTAT_UNDEF, GSLC_INITSTAT_INACTIVE, GSLC_INITSTAT_FAIL, GSLC_INITSTAT_ACTIVE }

    Status of a module's initialization.
- enum gslc_teEventType {
GSLC_EVT_NONE, GSLC_EVT_DRAW, GSLC_EVT_TOUCH, GSLC_EVT_TICK,
GSLV_EVT_CUSTOM }

    Event types.
- enum gslc_teEventSubType { GSLC_EVTSUB_NONE, GSLC_EVTSUB_DRAW_NEEDED, GSLC_EVTS↩
UB_DRAW_FORCE }

    Event sub-types.
- enum gslc_teRedrawType { GSLC_REDRAW_NONE, GSLC_REDRAW_FULL, GSLC_REDRAW_INC }

    Redraw types.
- enum gslc_teFontRefType { GSLC_FONTREF_FNAME, GSLC_FONTREF_PTR }

    Font Reference types.
- enum gslc_teFontRefMode { GSLC_FONTREF_MODE_DEFAULT, GSLC_FONTREF_MODE_1, GSLC_↩
FONTREF_MODE_2, GSLC_FONTREF_MODE_3 }

    Font Reference modes.
- enum gslc_teElemRefFlags {
GSLC_ELEMREF_NONE, GSLC_ELEMREF_SRC_RAM, GSLC_ELEMREF_SRC_PROG, GSLC_ELEM↩
REF_SRC_CONST,
GSLC_ELEMREF_REDRAW_NONE, GSLC_ELEMREF_REDRAW_FULL, GSLC_ELEMREF_REDRAW↩
_INC, GSLC_ELEMREF_GLOWING,
GSLC_ELEMREF_VISIBLE, GSLC_ELEMREF_SRC, GSLC_ELEMREF_REDRAW_MASK }

    Element reference flags: Describes characteristics of an element.
- enum gslc_teImgRefFlags {
GSLC_IMGREF_NONE, GSLC_IMGREF_SRC_FILE, GSLC_IMGREF_SRC_SD, GSLC_IMGREF_SRC_↩
RAM,
GSLC_IMGREF_SRC_PROG, GSLC_IMGREF_FMT_BMP24, GSLC_IMGREF_FMT_BMP16, GSLC_IM↩
GREF_FMT_RAW1,
GSLC_IMGREF_SRC, GSLC_IMGREF_FMT }

    Image reference flags: Describes characteristics of an image reference.
- enum gslc_teTxtFlags {
GSLC_TXT_MEM_RAM, GSLC_TXT_MEM_PROG, GSLC_TXT_ALLOC_NONE, GSLC_TXT_ALLOC_INT,
GSLC_TXT_ALLOC_EXT, GSLC_TXT_ENC_PLAIN, GSLC_TXT_ENC_UTF8, GSLC_TXT_MEM,
GSLC_TXT_ALLOC, GSLC_TXT_ENC, GSLC_TXT_DEFAULT }

    Text reference flags: Describes the characteristics of a text string (ie.

Functions

- char ∗ gslc_GetVer (gslc_tsGui ∗pGui)

    Get the GUIslice version number.
- const char ∗ gslc_GetNameDisp (gslc_tsGui ∗pGui)

    Get the GUIslice display driver name.
- const char ∗ gslc_GetNameTouch (gslc_tsGui ∗pGui)

    Get the GUIslice touch driver name.
- bool gslc_Init (gslc_tsGui ∗pGui, void ∗pvDriver, gslc_tsPage ∗asPage, uint8_t nMaxPage, gslc_tsFont ∗as↩
Font, uint8_t nMaxFont)

    Initialize the GUIslice library.
- void gslc_InitDebug (GSLC_CB_DEBUG_OUT pfunc)

    Initialize debug output.
- void gslc_DebugPrintf (const char ∗pFmt,...)

Optimized printf routine for GUIslice debug/error output.
- bool gslc_GuiRotate (gslc_tsGui ∗pGui, uint8_t nRotation)

  Dynamically change rotation, automatically adapt touchscreen axes swap/flip.
- void gslc_Quit (gslc_tsGui ∗pGui)

  Exit the GUIslice environment.
- void gslc_Update (gslc_tsGui ∗pGui)

  Perform main GUIslice handling functions.
- bool gslc_SetBkgndImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Configure the background to use a bitmap image.
- bool gslc_SetBkgndColor (gslc_tsGui ∗pGui, gslc_tsColor nCol)

  Configure the background to use a solid color.
- bool gslc_SetClipRect (gslc_tsGui ∗pGui, gslc_tsRect ∗pRect)

  Set the clipping rectangle for further drawing.
- bool gslc_IsInRect (int16_t nSelX, int16_t nSelY, gslc_tsRect rRect)

  Determine if a coordinate is inside of a rectangular region.
- gslc_tsRect gslc_ExpandRect (gslc_tsRect rRect, int16_t nExpandW, int16_t nExpandH)

  Expand or contract a rectangle in width and/or height (equal amounts on both side), based on the centerpoint of the rectangle.
- bool gslc_IsInWH (int16_t nSelX, int16_t nSelY, uint16_t nWidth, uint16_t nHeight)

  Determine if a coordinate is inside of a width x height region.
- void gslc_UnionRect (gslc_tsRect ∗pRect, gslc_tsRect rAddRect)

  Expand a rect to include another rect.
- void gslc_InvalidateRgnReset (gslc_tsGui ∗pGui)

  Reset the invalidation region.
- void gslc_InvalidateRgnPage (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage)

  Include an entire page (eg.
- void gslc_InvalidateRgnScreen (gslc_tsGui ∗pGui)

  Mark the entire screen as invalidated.
- void gslc_InvalidateRgnAdd (gslc_tsGui ∗pGui, gslc_tsRect rAddRect)

  Add a rectangular region to the invalidation region.
- bool gslc_ClipPt (gslc_tsRect ∗pClipRect, int16_t nX, int16_t nY)

  Perform basic clipping of a single point to a clipping region.
- bool gslc_ClipLine (gslc_tsRect ∗pClipRect, int16_t ∗pnX0, int16_t ∗pnY0, int16_t ∗pnX1, int16_t ∗pnY1)

  Perform basic clipping of a line to a clipping region.
- bool gslc_ClipRect (gslc_tsRect ∗pClipRect, gslc_tsRect ∗pRect)

  Perform basic clipping of a rectangle to a clipping region.
- gslc_tsImgRef gslc_GetImageFromFile (const char ∗pFname, gslc_teImgRefFlags eFmt)

  Create an image reference to a bitmap file in LINUX filesystem.
- gslc_tsImgRef gslc_GetImageFromSD (const char ∗pFname, gslc_teImgRefFlags eFmt)

  Create an image reference to a bitmap file in SD card.
- gslc_tsImgRef gslc_GetImageFromRam (unsigned char ∗pImgBuf, gslc_teImgRefFlags eFmt)

  Create an image reference to a bitmap in SRAM.
- gslc_tsImgRef gslc_GetImageFromProg (const unsigned char ∗pImgBuf, gslc_teImgRefFlags eFmt)

  Create an image reference to a bitmap in program memory (PROGMEM)
- void gslc_PolarToXY (uint16_t nRad, int16_t n64Ang, int16_t ∗nDX, int16_t ∗nDY)

  Convert polar coordinate to cartesian.
- int16_t gslc_sinFX (int16_t n64Ang)

  Calculate fixed-point sine function from fractional degrees.
- int16_t gslc_cosFX (int16_t n64Ang)

  Calculate fixed-point cosine function from fractional degrees.

- gslc_tsColor gslc_ColorBlend2 (gslc_tsColor colStart, gslc_tsColor colEnd, uint16_t nMidAmt, uint16_t n←-BlendAmt)

  Create a color based on a blend between two colors.
- gslc_tsColor gslc_ColorBlend3 (gslc_tsColor colStart, gslc_tsColor colMid, gslc_tsColor colEnd, uint16_t n←-MidAmt, uint16_t nBlendAmt)

  Create a color based on a blend between three colors.
- bool gslc_ColorEqual (gslc_tsColor a, gslc_tsColor b)

  Check whether two colors are equal.
- void gslc_DrawSetPixel (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, gslc_tsColor nCol)

  Set a pixel on the active screen to the given color with lock.
- void gslc_DrawLine (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)

  Draw an arbitrary line using Bresenham's algorithm.
- void gslc_DrawLineH (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nW, gslc_tsColor nCol)

  Draw a horizontal line.
- void gslc_DrawLineV (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nH, gslc_tsColor nCol)

  Draw a vertical line.
- void gslc_DrawLinePolar (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nRadStart, uint16_t nRadEnd, int16_t n64Ang, gslc_tsColor nCol)

  Draw a polar ray segment.
- void gslc_DrawFrameRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a framed rectangle.
- void gslc_DrawFrameRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

  Draw a framed rounded rectangle.
- void gslc_DrawFillRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a filled rectangle.
- void gslc_DrawFillRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

  Draw a filled rounded rectangle.
- void gslc_DrawFrameCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)

  Draw a framed circle.
- void gslc_DrawFillCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor n←-Col)

  Draw a filled circle.
- void gslc_DrawFrameTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

  Draw a framed triangle.
- void gslc_DrawFillTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

  Draw a filled triangle.
- void gslc_DrawFrameQuad (gslc_tsGui ∗pGui, gslc_tsPt ∗psPt, gslc_tsColor nCol)

  Draw a framed quadrilateral.
- void gslc_DrawFillQuad (gslc_tsGui ∗pGui, gslc_tsPt ∗psPt, gslc_tsColor nCol)

  Draw a filled quadrilateral.
- bool gslc_FontAdd (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefType eFontRefType, const void ∗pv←-FontRef, uint16_t nFontSz)

  Load a font into the local font cache and assign font ID (nFontId).
- bool gslc_FontSet (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefType eFontRefType, const void ∗pv←-FontRef, uint16_t nFontSz)

  Load a font into the local font cache and store as font ID (nFontId)
- gslc_tsFont ∗ gslc_FontGet (gslc_tsGui ∗pGui, int16_t nFontId)

  Fetch a font from its ID value.
- bool gslc_FontSetMode (gslc_tsGui ∗pGui, int16_t nFontId, gslc_teFontRefMode eFontMode)

Set a font's mode.
- int gslc_GetPageCur (gslc_tsGui ∗pGui)

  Fetch the current page ID.
- void gslc_SetStackPage (gslc_tsGui ∗pGui, uint8_t nStackPos, int16_t nPageId)

  Assign a page to the page stack.
- void gslc_SetStackState (gslc_tsGui ∗pGui, uint8_t nStackPos, bool bActive, bool bDoDraw)

  Change the status of a page in a page stack.
- void gslc_SetPageBase (gslc_tsGui ∗pGui, int16_t nPageId)

  Assigns a page for the base layer in the page stack.
- void gslc_SetPageCur (gslc_tsGui ∗pGui, int16_t nPageId)

  Select a page for the current layer in the page stack.
- void gslc_SetPageOverlay (gslc_tsGui ∗pGui, int16_t nPageId)

  Select a page for the overlay layer in the page stack.
- void gslc_PopupShow (gslc_tsGui ∗pGui, int16_t nPageId, bool bModal)

  Show a popup dialog.
- void gslc_PopupHide (gslc_tsGui ∗pGui)

  Hides the currently active popup dialog.
- void gslc_PageRedrawSet (gslc_tsGui ∗pGui, bool bRedraw)

  Update the need-redraw status for the current page.
- bool gslc_PageRedrawGet (gslc_tsGui ∗pGui)

  Get the need-redraw status for the current page.
- void gslc_PageAdd (gslc_tsGui ∗pGui, int16_t nPageId, gslc_tsElem ∗psElem, uint16_t nMaxElem, gslc_↩
tsElemRef ∗psElemRef, uint16_t nMaxElemRef)

  Add a page to the GUI.
- gslc_tsElemRef ∗ gslc_PageFindElemById (gslc_tsGui ∗pGui, int16_t nPageId, int16_t nElemId)

  Find an element in the GUI by its Page ID and Element ID.
- gslc_tsElemRef ∗ gslc_ElemCreateTxt (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

  Create a Text Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBtnTxt (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId, GSLC_CB_TOUCH cbTouch)

  Create a textual Button Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBtnImg (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, gslc_tsImgRef sImgRef, gslc_tsImgRef sImgRefSel, GSLC_CB_TOUCH cbTouch)

  Create a graphical Button Element.
- gslc_tsElemRef ∗ gslc_ElemCreateBox (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect r↩
Elem)

  Create a Box Element.
- gslc_tsElemRef ∗ gslc_ElemCreateLine (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1)

  Create a Line Element.
- gslc_tsElemRef ∗ gslc_ElemCreateImg (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPage, gslc_tsRect r↩
Elem, gslc_tsImgRef sImgRef)

  Create an image Element.
- int gslc_ElemGetId (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get an Element ID from an element structure.
- void gslc_ElemSetFillEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bFillEn)

  Set the fill state for an Element.
- void gslc_ElemSetFrameEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bFrameEn)

  Set the frame state for an Element.
- void gslc_ElemSetRoundEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bRoundEn)

  Set the rounded frame/fill state for an Element.

- void gslc_ElemSetCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colFrame, gslc_tsColor colFill, gslc_tsColor colFillGlow)

  Update the common color selection for an Element.
- void gslc_ElemSetGlowCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colFrameGlow, gslc_tsColor colFillGlow, gslc_tsColor colTxtGlow)

  Update the common color selection for glowing state of an Element.
- void gslc_ElemSetGroup (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int nGroupId)

  Set the group ID for an element.
- int gslc_ElemGetGroup (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the group ID for an element.
- void gslc_ElemSetTxtAlign (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, unsigned nAlign)

  Set the alignment of a textual element (horizontal and vertical)
- void gslc_ElemSetTxtMargin (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, unsigned nMargin)

  Set the margin around of a textual element.
- void gslc_ElemSetTxtStr (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, const char ∗pStr)

  Update the text string associated with an Element.
- char ∗ gslc_ElemGetTxtStr (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Fetch the current text string associated with an Element.
- void gslc_ElemSetTxtCol (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsColor colVal)

  Update the text string color associated with an Element ID.
- void gslc_ElemSetTxtMem (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teTxtFlags eFlags)

  Update the text string location in memory.
- void gslc_ElemSetTxtEnc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teTxtFlags eFlags)

  Update the text string encoding mode.
- void gslc_ElemUpdateFont (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int nFontId)

  Update the Font selected for an Element's text.
- void gslc_ElemSetRedraw (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teRedrawType eRedraw)

  Update the need-redraw status for an element.
- gslc_teRedrawType gslc_ElemGetRedraw (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the need-redraw status for an element.
- void gslc_ElemSetGlowEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bGlowEn)

  Update the glowing enable for an element.
- void gslc_ElemSetClickEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bClickEn)

  Update the click enable for an element.
- void gslc_ElemSetTouchFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_TOUCH funcCb)

  Update the touch function callback for an element.
- void gslc_ElemSetStyleFrom (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRefSrc, gslc_tsElemRef ∗pElem←RefDest)

  Copy style settings from one element to another.
- bool gslc_ElemGetGlowEn (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the glowing enable for an element.
- void gslc_ElemSetGlow (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bGlowing)

  Update the glowing indicator for an element.
- bool gslc_ElemGetGlow (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the glowing indicator for an element.
- void gslc_ElemSetVisible (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, bool bVisible)

  Update the visibility status for an element.
- bool gslc_ElemGetVisible (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

  Get the visibility status for an element.
- void gslc_ElemSetDrawFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_DRAW funcCb)

  Assign the drawing callback function for an element.

- void gslc_ElemSetTickFunc (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, GSLC_CB_TICK funcCb)

    Assign the tick callback function for an element.
- bool gslc_ElemOwnsCoord (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nX, int16_t nY, bool b←OnlyClickEn)

    Determine if a coordinate is inside of an element.
- bool gslc_InitTouch (gslc_tsGui ∗pGui, const char ∗acDev)

    Initialize the touchscreen device driver.
- bool gslc_GetTouch (gslc_tsGui ∗pGui, int16_t ∗pnX, int16_t ∗pnY, uint16_t ∗pnPress, gslc_teInputRawEvent ∗peInputEvent, int16_t ∗pnInputVal)

    Initialize the touchscreen device driver.
- void gslc_SetTouchRemapEn (gslc_tsGui ∗pGui, bool bEn)

    Configure touchscreen remapping.
- void gslc_SetTouchRemapCal (gslc_tsGui ∗pGui, uint16_t nXMin, uint16_t nXMax, uint16_t nYMin, uint16_t nYMax)

    Configure touchscreen calibration values.
- void gslc_SetTouchRemapYX (gslc_tsGui ∗pGui, bool bSwap)

    Configure touchscreen XY swap.
- void gslc_SetPinPollFunc (gslc_tsGui ∗pGui, GSLC_CB_PIN_POLL pfunc)
- void gslc_InitInputMap (gslc_tsGui ∗pGui, gslc_tsInputMap ∗asInputMap, uint8_t nInputMapMax)
- void gslc_InputMapAdd (gslc_tsGui ∗pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc_te←Action eAction, int16_t nActionVal)
- gslc_tsImgRef gslc_ResetImage ()

    Create a blank image reference structure.
- gslc_tsElem gslc_ElemCreate (gslc_tsGui ∗pGui, int16_t nElemId, int16_t nPageId, int16_t nType, gslc_ts←Rect rElem, char ∗pStrBuf, uint8_t nStrBufMax, int16_t nFontId)

    Create a new element with default styling.
- gslc_tsElemRef ∗ gslc_ElemAdd (gslc_tsGui ∗pGui, int16_t nPageId, gslc_tsElem ∗pElem, gslc_teElem←RefFlags eFlags)

    Add the Element to the list of generated elements in the GUI environment.
- uint8_t gslc_GetElemRefFlag (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nFlagMask)

    Get the flags associated with an element reference.
- void gslc_SetElemRefFlag (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, uint8_t nFlagMask, uint8_t n←FlagVal)

    Set the flags associated with an element reference.
- gslc_tsElem ∗ gslc_GetElemFromRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef)

    Returns a pointer to an element from an element reference, copying from FLASH to RAM if element is stored in PROGMEM.
- gslc_tsElem ∗ gslc_GetElemFromRefD (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nLineNum)

    Returns a pointer to an element from an element reference.
- void ∗ gslc_GetXDataFromRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, int16_t nType, int16_t nLine←Num)

    Returns a pointer to the data structure associated with an extended element.
- void gslc_ElemSetImage (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_tsImgRef sImgRef, gslc_ts←ImgRef sImgRefSel)

    Set an element to use a bitmap image.
- bool gslc_ElemDrawByRef (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRef, gslc_teRedrawType eRedraw)

    Draw an element to the active display.
- void gslc_ElemDraw (gslc_tsGui ∗pGui, int16_t nPageId, int16_t nElemId)

    Draw an element to the active display.
- void gslc_DrawTxtBase (gslc_tsGui ∗pGui, char ∗pStrBuf, gslc_tsRect rTxt, gslc_tsFont ∗pTxtFont, gslc←_teTxtFlags eTxtFlags, int8_t eTxtAlign, gslc_tsColor colTxt, gslc_tsColor colBg, int16_t nMarginW, int16_t nMarginH)

Draw text with full text justification.
- void gslc_SetRoundRadius (gslc_tsGui ∗pGui, uint8_t nRadius)

  Set the global rounded radius.
- bool gslc_PageEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for a page.
- void gslc_PageRedrawGo (gslc_tsGui ∗pGui)

  Redraw all elements on the active page.
- void gslc_PageFlipSet (gslc_tsGui ∗pGui, bool bNeeded)

  Indicate whether the screen requires page flip.
- bool gslc_PageFlipGet (gslc_tsGui ∗pGui)

  Get state of pending page flip state.
- void gslc_PageFlipGo (gslc_tsGui ∗pGui)

  Update the visible screen if page has been marked for flipping.
- gslc_tsPage ∗ gslc_PageFindById (gslc_tsGui ∗pGui, int16_t nPageId)

  Find a page in the GUI by its ID.
- void gslc_PageRedrawCalc (gslc_tsGui ∗pGui)

  Perform a redraw calculation on the page to determine if additional elements should also be redrawn.
- int16_t gslc_PageFocusStep (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, bool bNext)
- gslc_tsEvent gslc_EventCreate (gslc_tsGui ∗pGui, gslc_teEventType eType, uint8_t nSubType, void ∗pv←Scope, void ∗pvData)

  Create an event structure.
- bool gslc_ElemEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for an element.
- bool gslc_ElemSendEventTouch (gslc_tsGui ∗pGui, gslc_tsElemRef ∗pElemRefTracked, gslc_teTouch e←Touch, int16_t nX, int16_t nY)

  Trigger an element's touch event.
- void gslc_CollectReset (gslc_tsCollect ∗pCollect, gslc_tsElem ∗asElem, uint16_t nElemMax, gslc_tsElemRef ∗asElemRef, uint16_t nElemRefMax)

  Reset the members of an element collection.
- gslc_tsElemRef ∗ gslc_CollectElemAdd (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, const gslc_tsElem ∗p←Elem, gslc_teElemRefFlags eFlags)

  Add an element to a collection.
- bool gslc_CollectGetRedraw (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Determine if any elements in a collection need redraw.
- gslc_tsElemRef ∗ gslc_CollectFindElemById (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nElemId)

  Find an element in a collection by its Element ID.
- gslc_tsElemRef ∗ gslc_CollectFindElemFromCoord (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nX, int16_t nY)

  Find an element in a collection by a coordinate coordinate.
- int gslc_CollectGetNextId (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Allocate the next available Element ID in a collection.
- gslc_tsElemRef ∗ gslc_CollectGetElemRefTracked (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Get the element within a collection that is currently being tracked.
- void gslc_CollectSetElemTracked (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsElemRef ∗pElemRef)

  Set the element within a collection that is currently being tracked.
- int16_t gslc_CollectGetFocus (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Get the element index within a collection that is currently in focus.
- void gslc_CollectSetFocus (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, int16_t nElemInd)

  Set the element index within a collection that is currently in focus.
- bool gslc_CollectFindFocusStep (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, bool bNext, bool ∗pbWrapped, int16_t ∗pnElemInd)
- void gslc_CollectSetParent (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsElemRef ∗pElemRefParent)

Assign the parent element reference to all elements within a collection.

- bool gslc_CollectEvent (void ∗pvGui, gslc_tsEvent sEvent)

  Common event handler function for an element collection.
- void gslc_CollectTouch (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsEventTouch ∗pEventTouch)

  Handle touch events within the element collection.
- bool gslc_CollectTouchCompound (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY, gslc_tsCollect ∗pCollect)

  Handle dispatch of touch (up,down,move) events to compound elements sub elements.
- void gslc_CollectInput (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect, gslc_tsEventTouch ∗pEventTouch)

  Handle direct input events within the element collection.
- void gslc_TrackTouch (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, int16_t nX, int16_t nY, uint16_t nPress)

  Handles a touch event and performs the necessary tracking, glowing and selection actions depending on the press state.
- void gslc_TrackInput (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage, gslc_teInputRawEvent eInputEvent, int16_↩ t nInputVal)

  Handles a direct input event and performs the necessary tracking, glowing and selection actions depending on the state.
- bool gslc_InputMapLookup (gslc_tsGui ∗pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc↩ _teAction ∗peAction, int16_t ∗pnActionVal)
- void gslc_GuiDestruct (gslc_tsGui ∗pGui)

  Free up any surfaces associated with the GUI, pages, collections and elements.
- void gslc_PageDestruct (gslc_tsGui ∗pGui, gslc_tsPage ∗pPage)

  Free up any members associated with a page.
- void gslc_CollectDestruct (gslc_tsGui ∗pGui, gslc_tsCollect ∗pCollect)

  Free up any members associated with an element collection.
- void gslc_ElemDestruct (gslc_tsElem ∗pElem)

  Free up any members associated with an element.
- void gslc_ResetFont (gslc_tsFont ∗pFont)

  Initialize a Font struct.
- void gslc_ResetElem (gslc_tsElem ∗pElem)

  Initialize an Element struct.

Variables

- GSLC_CB_DEBUG_OUT g_pfDebugOut

  Global debug output function.

9.29.1 Macro Definition Documentation

9.29.1.1 #define GSLC_2PI

9.29.1.2 #define GSLC_ALIGN_BOT_LEFT

Align to bottom-left.

9.29.1.3 #define GSLC_ALIGN_BOT_MID

Align to middle of bottom.

9.29.1.4 #define GSLC_ALIGN_BOT_RIGHT

Align to bottom-right.

9.29.1.5 #define GSLC_ALIGN_MID_LEFT

Align to middle of left side.

9.29.1.6 #define GSLC_ALIGN_MID_MID

Align to center.

9.29.1.7 #define GSLC_ALIGN_MID_RIGHT

Align to middle of right side.

9.29.1.8 #define GSLC_ALIGN_TOP_LEFT

Align to top-left.

9.29.1.9 #define GSLC_ALIGN_TOP_MID

Align to middle of top.

9.29.1.10 #define GSLC_ALIGN_TOP_RIGHT

Align to top-right.

9.29.1.11 #define GSLC_ALIGNH_LEFT

Horizontal align to left.

9.29.1.12 #define GSLC_ALIGNH_MID

Horizontal align to middle.

9.29.1.13 #define GSLC_ALIGNH_RIGHT

Horizontal align to right.

9.29.1.14 #define GSLC_ALIGNV_BOT

Vertical align to bottom.

9.29.1.15 #define GSLC_ALIGNV_MID

Vertical align to middle.

9.29.1.16 #define GSLC_ALIGNV_TOP

Element text alignment.

Vertical align to top

9.29.1.17 #define GSLC_COL_BLACK

Black.

9.29.1.18 #define GSLC_COL_BLUE

Blue.

9.29.1.19 #define GSLC_COL_BLUE_DK1

Blue (dark1)

9.29.1.20 #define GSLC_COL_BLUE_DK2

Blue (dark2)

9.29.1.21 #define GSLC_COL_BLUE_DK3

Blue (dark3)

9.29.1.22 #define GSLC_COL_BLUE_DK4

Blue (dark4)

9.29.1.23 #define GSLC_COL_BLUE_LT1

Blue (light1)

9.29.1.24 #define GSLC_COL_BLUE_LT2

Blue (light2)

9.29.1.25 #define GSLC_COL_BLUE_LT3

Blue (light3)

9.29.1.26 #define GSLC_COL_BLUE_LT4

Blue (light4)

9.29.1.27 #define GSLC_COL_BROWN

Brown.

9.29.1.28 #define GSLC_COL_CYAN

Cyan.

9.29.1.29 #define GSLC_COL_GRAY

Gray.

9.29.1.30 #define GSLC_COL_GRAY_DK1

Gray (dark)

9.29.1.31 #define GSLC_COL_GRAY_DK2

Gray (dark)

9.29.1.32 #define GSLC_COL_GRAY_DK3

Gray (dark)

9.29.1.33 #define GSLC_COL_GRAY_LT1

Gray (light1)

9.29.1.34 #define GSLC_COL_GRAY_LT2

Gray (light2)

9.29.1.35 #define GSLC_COL_GRAY_LT3

Gray (light3)

9.29.1.36 #define GSLC_COL_GREEN

Green.

9.29.1.37 #define GSLC_COL_GREEN_DK1

Green (dark1)

9.29.1.38 #define GSLC_COL_GREEN_DK2

Green (dark2)

9.29.1.39 #define GSLC_COL_GREEN_DK3

Green (dark3)

9.29.1.40 #define GSLC_COL_GREEN_DK4

Green (dark4)

9.29.1.41 #define GSLC_COL_GREEN_LT1

Green (light1)

9.29.1.42 #define GSLC_COL_GREEN_LT2

Green (light2)

9.29.1.43 #define GSLC_COL_GREEN_LT3

Green (light3)

9.29.1.44 #define GSLC_COL_GREEN_LT4

Green (light4)

9.29.1.45 #define GSLC_COL_MAGENTA

Magenta.

9.29.1.46 #define GSLC_COL_ORANGE

Orange.

9.29.1.47 #define GSLC_COL_PURPLE

Purple.

9.29.1.48 #define GSLC_COL_RED

Red.

9.29.1.49 #define GSLC_COL_RED_DK1

Red (dark1)

9.29.1.50 #define GSLC_COL_RED_DK2

Red (dark2)

9.29.1.51 #define GSLC_COL_RED_DK3

Red (dark3)

9.29.1.52 #define GSLC_COL_RED_DK4

Basic color definition.

Red (dark4)

9.29.1.53 #define GSLC_COL_RED_LT1

Red (light1)

9.29.1.54 #define GSLC_COL_RED_LT2

Red (light2)

9.29.1.55 #define GSLC_COL_RED_LT3

Red (light3)

9.29.1.56 #define GSLC_COL_RED_LT4

Red (light4)

9.29.1.57 #define GSLC_COL_TEAL

Teal.

9.29.1.58 #define GSLC_COL_WHITE

White.

9.29.1.59 #define GSLC_COL_YELLOW

Yellow.

9.29.1.60 #define GSLC_COL_YELLOW_DK

Yellow (dark)

9.29.1.61 #define GSLC_COLMONO_BLACK

Black.

9.29.1.62 #define GSLC_COLMONO_WHITE

White.

9.29.1.63 #define GSLC_ELEM_FEA_CLICK_EN

Element accepts touch presses.

9.29.1.64 #define GSLC_ELEM_FEA_FILL_EN

Element is drawn with a fill.

9.29.1.65 #define GSLC_ELEM_FEA_FRAME_EN

Element is drawn with a frame.

9.29.1.66 #define GSLC_ELEM_FEA_GLOW_EN

Element supports glowing state.

9.29.1.67 #define GSLC_ELEM_FEA_NONE

Element default (no features set))

9.29.1.68 #define GSLC_ELEM_FEA_ROUND_EN

Element is drawn with a rounded profile.

9.29.1.69 #define GSLC_ELEM_FEA_VALID

Element features type.

Element record is valid

9.29.1.70 #define GSLC_ELEMREF_DEFAULT

Define the default element reference flags for new elements.

9.29.1.71 #define GSLC_PMEM

9.29.2 Typedef Documentation

9.29.2.1 typedef int16_t(∗ GSLC_CB_DEBUG_OUT) (char ch)

9.29.2.2 typedef bool(∗ GSLC_CB_DRAW) (void ∗pvGui, void ∗pvElemRef, gslc_teRedrawType eRedraw)

Callback function for element drawing.

9.29.2.3 typedef bool(∗ GSLC_CB_EVENT) (void ∗pvGui, gslc_tsEvent sEvent)

Callback function for element drawing.

9.29.2.4 typedef bool(∗ GSLC_CB_INPUT) (void ∗pvGui, void ∗pvElemRef, int16_t nStatus, void ∗pvData)

Callback function for element input ready.

9.29.2.5 typedef bool(∗ GSLC_CB_PIN_POLL) (void ∗pvGui, int16_t ∗pnPinInd, int16_t ∗pnPinVal)

Callback function for pin polling.

9.29.2.6 typedef bool(∗ GSLC_CB_TICK) (void ∗pvGui, void ∗pvElemRef)

Callback function for element tick.

9.29.2.7 typedef bool(∗ GSLC_CB_TOUCH) (void ∗pvGui, void ∗pvElemRef, gslc_teTouch eTouch, int16_t nX, int16_t nY)

Callback function for element touch tracking.

9.29.2.8 typedef struct gslc_tsColor gslc_tsColor

Color structure. Defines RGB triplet.

9.29.2.9 typedef struct gslc_tsElem gslc_tsElem

Element Struct.

- Represents a single graphic element in the GUIslice environment
- A page is made up of a number of elements
- Each element is created with a user-specified ID for further accesses (or GSLC_ID_AUTO for it to be auto-generated)
- Display order of elements in a page is based upon the creation order
- Extensions to the core element types is provided through the pXData reference and pfuncX∗ callback functions.

9.29.2.10 typedef struct gslc_tsEvent gslc_tsEvent

Event structure.

9.29.2.11 typedef struct gslc_tsEventTouch gslc_tsEventTouch

Structure used to pass touch data through event.

9.29.2.12 typedef struct gslc_tsPt gslc_tsPt

Define point coordinates.

9.29.2.13 typedef struct gslc_tsRect gslc_tsRect

Rectangular region. Defines X,Y corner coordinates plus dimensions.

9.29.3 Enumeration Type Documentation

9.29.3.1 enum gslc_teAction

GUI Action Requested These actions are usually the result of an InputMap lookup.

Enumerator

GSLC_ACTION_UNDEF	Invalid action.
GSLC_ACTION_NONE	No action to perform.
GSLC_ACTION_FOCUS_PREV	Advance focus to the previous GUI element.
GSLC_ACTION_FOCUS_NEXT	Advance focus to the next GUI element.
GSLC_ACTION_SELECT	Select the currently focused GUI element.
GSLC_ACTION_SET_REL	Adjust value (relative) of focused element.
GSLC_ACTION_SET_ABS	Adjust value (absolute) of focused element.
GSLC_ACTION_DEBUG	Internal debug action.

9.29.3.2 enum gslc_teElemId

Element ID enumerations.

- The Element ID is the primary means for user code to reference a graphic element.
- Application code can assign arbitrary Element ID values in the range of 0...16383
- Specifying GSLC_ID_AUTO to ElemCreate() requests that GUIslice auto-assign an ID value for the Element. These auto-assigned values will begin at GSLC_ID_AUTO_BASE.
- Negative Element ID values are reserved

Enumerator

GSLC_ID_USER_BASE	Starting Element ID for user assignments.
GSLC_ID_NONE	No Element ID has been assigned.
GSLC_ID_AUTO	Auto-assigned Element ID requested.
GSLC_ID_TEMP	ID for Temporary Element.
GSLC_ID_AUTO_BASE	Starting Element ID to start auto-assignment (when GSLC_ID_AUTO is specified)

9.29.3.3 enum gslc_teElemInd

Element Index enumerations.

- The Element Index is used for internal purposes as an offset

Enumerator

GSLC_IND_NONE No Element Index is available.

GSLC_IND_FIRST User elements start at index 0.

9.29.3.4 enum gslc_teElemRefFlags

Element reference flags: Describes characteristics of an element.

- Primarily used to support relocation of elements to Flash memory (PROGMEM)

Enumerator

GSLC_ELEMREF_NONE No element defined.

GSLC_ELEMREF_SRC_RAM Element is read/write Stored in RAM (internal element array)) Access directly.

GSLC_ELEMREF_SRC_PROG Element is read-only / const Stored in FLASH (external to element array) Access via PROGMEM.

GSLC_ELEMREF_SRC_CONST Element is read-only / const Stored in FLASH (external to element array) Access directly.

GSLC_ELEMREF_REDRAW_NONE No redraw requested.

GSLC_ELEMREF_REDRAW_FULL Full redraw of element requested.

GSLC_ELEMREF_REDRAW_INC Incremental redraw of element requested.

GSLC_ELEMREF_GLOWING Element state is glowing.

GSLC_ELEMREF_VISIBLE Element is currently shown (ie. visible)

GSLC_ELEMREF_SRC Mask for Source flags.

GSLC_ELEMREF_REDRAW_MASK Mask for Redraw flags.

9.29.3.5 enum gslc_teEventSubType

Event sub-types.

Enumerator

GSLC_EVTSUB_NONE

GSLC_EVTSUB_DRAW_NEEDED Incremental redraw (as needed)

GSLC_EVTSUB_DRAW_FORCE Force a full redraw.

9.29.3.6 enum gslc_teEventType

Event types.

Enumerator

GSLC_EVT_NONE No event; ignore.

GSLC_EVT_DRAW Perform redraw.

GSLC_EVT_TOUCH Track touch event.

GSLC_EVT_TICK Perform background tick handling.

GSLV_EVT_CUSTOM Custom event.

9.29.3.7 enum gslc_teFontId

Font ID enumerations.

- The Font ID is the primary means for user code to reference a specific font.
- Application code can assign arbitrary Font ID values in the range of 0...16383
- Negative Font ID values are reserved

Enumerator

GSLC_FONT_USER_BASE Starting Font ID for user assignments.

GSLC_FONT_NONE No Font ID has been assigned.

9.29.3.8 enum gslc_teFontRefMode

Font Reference modes.

- The Font Reference mode defines the source for the selected font. For graphics libraries that offer multiple types of fonts, this can be used to differentiate between a default font, hardware fonts, software fonts, etc.
- The encoding between the different modes is driver-specific.

Enumerator

GSLC_FONTREF_MODE_DEFAULT Default font mode.

GSLC_FONTREF_MODE_1 Font mode 1.

GSLC_FONTREF_MODE_2 Font mode 2.

GSLC_FONTREF_MODE_3 Font mode 3.

9.29.3.9 enum gslc_teFontRefType

Font Reference types.

- The Font Reference type defines the way in which a font is selected. In some device targets (such as LINUX SDL) a filename to a font file is provided. In others (such as Arduino, ESP8266), a pointer is given to a font structure (or NULL for default).

Enumerator

GSLC_FONTREF_FNAME	Font reference is a filename (full path)
GSLC_FONTREF_PTR	Font reference is a pointer to a font structure.

9.29.3.10 enum gslc_teGroupId

Group ID enumerations.

Enumerator

GSLC_GROUP_ID_USER_BASE	Starting Group ID for user assignments.
GSLC_GROUP_ID_NONE	No Group ID has been assigned.

9.29.3.11 enum gslc_teImgRefFlags

Image reference flags: Describes characteristics of an image reference.

Enumerator

GSLC_IMGREF_NONE	No image defined.
GSLC_IMGREF_SRC_FILE	Image is stored in file system.
GSLC_IMGREF_SRC_SD	Image is stored on SD card.
GSLC_IMGREF_SRC_RAM	Image is stored in RAM.
GSLC_IMGREF_SRC_PROG	Image is stored in program memory (PROGMEM)
GSLC_IMGREF_FMT_BMP24	Image format is BMP (24-bit)
GSLC_IMGREF_FMT_BMP16	Image format is BMP (16-bit RGB565)
GSLC_IMGREF_FMT_RAW1	Image format is raw monochrome (1-bit)
GSLC_IMGREF_SRC	Mask for Source flags.
GSLC_IMGREF_FMT	Mask for Format flags.

9.29.3.12 enum gslc_teInitStat

Status of a module's initialization.

Enumerator

GSLC_INITSTAT_UNDEF	Module status has not been defined yet.
GSLC_INITSTAT_INACTIVE	Module is not enabled.
GSLC_INITSTAT_FAIL	Module is enabled but failed to init.
GSLC_INITSTAT_ACTIVE	Module is enabled and initalized OK.

9.29.3.13 enum gslc_teInputRawEvent

Raw input event types: touch, key, GPIOs.

Enumerator

GSLC_INPUT_NONE	No input event.
GSLC_INPUT_TOUCH	Touch / mouse event.
GSLC_INPUT_KEY_DOWN	Key press down / pin input asserted.
GSLC_INPUT_KEY_UP	Key press up (released)
GSLC_INPUT_PIN_ASSERT	GPIO pin input asserted (eg. set to 1 / High)
GSLC_INPUT_PIN_DEASSERT	GPIO pin input deasserted (eg. set to 0 / Low)

9.29.3.14 enum gslc_tePageId

Page ID enumerations.

- The Page ID is the primary means for user code to reference a specific page of elements.
- Application code can assign arbitrary Page ID values in the range of 0...16383
- Negative Page ID values are reserved

Enumerator

GSLC_PAGE_USER_BASE	Starting Page ID for user assignments.
GSLC_PAGE_NONE	No Page ID has been assigned.

9.29.3.15 enum gslc_tePin

General purpose pin/button constants.

Enumerator

GSLC_PIN_BTN_A	Button A (short press)
GSLC_PIN_BTN_A_LONG	Button A (long press)
GSLC_PIN_BTN_B	Button B (short press)
GSLC_PIN_BTN_B_LONG	Button B (long press)
GSLC_PIN_BTN_C	Button C (short press)
GSLC_PIN_BTN_C_LONG	Button C (long press)
GSLC_PIN_BTN_D	Button D (short press)
GSLC_PIN_BTN_D_LONG	Button D (long press)
GSLC_PIN_BTN_E	Button E (short press)
GSLC_PIN_BTN_E_LONG	Button E (long press)
GSLC_PIN_BTN_UP	Button Up (short press)
GSLC_PIN_BTN_DOWN	Button Down (short press)
GSLC_PIN_BTN_LEFT	Button Left (short press)
GSLC_PIN_BTN_RIGHT	Button Right (short press)
GSLC_PIN_BTN_SEL	Button Select (short press)

9.29.3.16 enum gslc_teRedrawType

Redraw types.

Enumerator

GSLC_REDRAW_NONE No redraw requested.

GSLC_REDRAW_FULL Full redraw of element requested.

GSLC_REDRAW_INC Incremental redraw of element requested.

9.29.3.17 enum gslc_teStackPage

Define page stack.

Enumerator

GSLC_STACK_BASE Base page.

GSLC_STACK_CUR Current page.

GSLC_STACK_OVERLAY Overlay page (eg. popups)

GSLC_STACK__MAX Defines maximum number of pages in stack.

9.29.3.18 enum gslc_teTouch

Processed event from input raw events and actions.

Enumerator

GSLC_TOUCH_NONE No touch event active.

GSLC_TOUCH_TYPE_MASK Mask for type: coord/direct mode.

GSLC_TOUCH_COORD Event based on touch coordinate.

GSLC_TOUCH_DIRECT Event based on specific element index (keyboard/GPIO action)

GSLC_TOUCH_SUBTYPE_MASK Mask for subtype.

GSLC_TOUCH_DOWN Touch event (down)

GSLC_TOUCH_DOWN_IN Touch event (down inside tracked element)

GSLC_TOUCH_DOWN_OUT Touch event (down outside tracked element)

GSLC_TOUCH_UP Touch event (up)

GSLC_TOUCH_UP_IN Touch event (up inside tracked element)

GSLC_TOUCH_UP_OUT Touch event (up inside tracked element)

GSLC_TOUCH_MOVE Touch event (move)

GSLC_TOUCH_MOVE_IN Touch event (move inside tracked element)

GSLC_TOUCH_MOVE_OUT Touch event (move outside tracked element)

GSLC_TOUCH_FOCUS_ON Direct event focus on element.

GSLC_TOUCH_FOCUS_OFF Direct event focus away from focused element.

GSLC_TOUCH_FOCUS_SELECT Direct event select focus element.

GSLC_TOUCH_SET_REL Direct event set value (relative) on focus element.

GSLC_TOUCH_SET_ABS Direct event set value (absolute) on focus element.

9.29.3.19 enum gslc_teTxtFlags

Text reference flags: Describes the characteristics of a text string (ie.

whether internal to element or external and RAM vs Flash).)

Supported flag combinations are:

- ALLOC_NONE
- ALLOC_INT | MEM_RAM
- ALLOC_EXT | MEM_RAM
- ALLOC_EXT | MEM_PROG

Enumerator

GSLC_TXT_MEM_RAM	Text string is in SRAM (read-write)
GSLC_TXT_MEM_PROG	Text string is in PROGMEM (read-only)
GSLC_TXT_ALLOC_NONE	No text string present.
GSLC_TXT_ALLOC_INT	Text string allocated in internal element memory (GSLC_STR_LOCAL=1)
GSLC_TXT_ALLOC_EXT	Text string allocated in external memory (GSLC_STR_LOCAL=0), ie. user code.
GSLC_TXT_ENC_PLAIN	Encoding is plain text (LATIN1))
GSLC_TXT_ENC_UTF8	Encoding is UTF-8.
GSLC_TXT_MEM	Mask for updating text memory type.
GSLC_TXT_ALLOC	Mask for updating location of text string buffer allocation.
GSLC_TXT_ENC	Mask for updating text encoding.
GSLC_TXT_DEFAULT	

9.29.3.20 enum gslc_teTypeCore

Element type.

Enumerator

GSLC_TYPE_NONE	No element type specified.
GSLC_TYPE_BKGND	Background element type.
GSLC_TYPE_BTN	Button element type.
GSLC_TYPE_TXT	Text label element type.
GSLC_TYPE_BOX	Box / frame element type.
GSLC_TYPE_LINE	Line element type.
GSLC_TYPE_BASE_EXTEND	Base value for extended type enumerations.

9.29.4 Variable Documentation

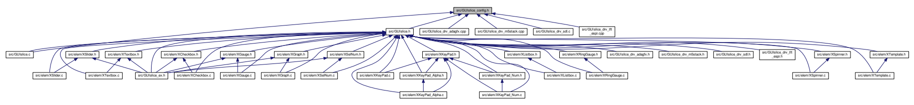
9.29.4.1 GSLC_CB_DEBUG_OUT g_pfDebugOut

Global debug output function.

- The user assigns this function via gslc_InitDebug()

9.30 src/GUIslice_config.h File Reference

This graph shows which files directly or indirectly include this file:



9.31 src/GUIslice_config_ard.h File Reference

Macros

- #define DRV_DISP_ADAGFX
- #define DRV_TOUCH_NONE
- #define DRV_DISP_ADAGFX_ILI9341
- #define ADAGFX_PIN_CS
- #define ADAGFX_PIN_DC
- #define ADAGFX_PIN_RST
- #define ADAGFX_PIN_SDCS
- #define ADAGFX_PIN_WR
- #define ADAGFX_PIN_RD
- #define ADAGFX_SPI_HW
- #define ADAGFX_PIN_MOSI
- #define ADAGFX_PIN_MISO
- #define ADAGFX_PIN_CLK
- #define GSLC_ROTATE
- #define TOUCH_ROTATION_DATA
- #define TOUCH_ROTATION_SWAPXY(rotation)
- #define TOUCH_ROTATION_FLIPX(rotation)
- #define TOUCH_ROTATION_FLIPY(rotation)
- #define ADATOUCH_SWAP_XY
- #define ADATOUCH_FLIP_X
- #define ADATOUCH_FLIP_Y
- #define GSLC_TOUCH_MAX_EVT
- #define DEBUG_ERR
- #define GSLC_FEATURE_COMPOUND
- #define GSLC_FEATURE_XGAUGE_RADIAL
- #define GSLC_FEATURE_XGAUGE_RAMP
- #define GSLC_FEATURE_XTEXTBOX_EMBED
- #define GSLC_FEATURE_INPUT
- #define GSLC_SD_EN
- #define GSLC_SD_BUFFPIXEL
- #define GSLC_CLIP_EN
- #define GSLC_BMP_TRANS_EN
- #define GSLC_BMP_TRANS_RGB
- #define GSLC_LOCAL_STR
- #define GSLC_LOCAL_STR_LEN
- #define GSLC_USE_FLOAT
- #define GSLC_DEV_TOUCH
- #define GSLC_USE_PROGMEM

9.31.1 Macro Definition Documentation

9.31.1.1 #define ADAGFX_PIN_CLK

9.31.1.2 #define ADAGFX_PIN_CS

9.31.1.3 #define ADAGFX_PIN_DC

9.31.1.4 #define ADAGFX_PIN_MISO

9.31.1.5 #define ADAGFX_PIN_MOSI

9.31.1.6 #define ADAGFX_PIN_RD

9.31.1.7 #define ADAGFX_PIN_RST

9.31.1.8 #define ADAGFX_PIN_SDCS

9.31.1.9 #define ADAGFX_PIN_WR

9.31.1.10 #define ADAGFX_SPI_HW

9.31.1.11 #define ADATOUCH_FLIP_X

9.31.1.12 #define ADATOUCH_FLIP_Y

9.31.1.13 #define ADATOUCH_SWAP_XY

9.31.1.14 #define DEBUG_ERR

9.31.1.15 #define DRV_DISP_ADAGFX

9.31.1.16 #define DRV_DISP_ADAGFX_ILI9341

9.31.1.17 #define DRV_TOUCH_NONE

9.31.1.18 #define GSLC_BMP_TRANS_EN

9.31.1.19 #define GSLC_BMP_TRANS_RGB

9.31.1.20 #define GSLC_CLIP_EN

9.31.1.21 #define GSLC_DEV_TOUCH

9.31.1.22 #define GSLC_FEATURE_COMPOUND

9.31.1.23 #define GSLC_FEATURE_INPUT

9.31.1.24 #define GSLC_FEATURE_XGAUGE_RADIAL

9.31.1.25 #define GSLC_FEATURE_XGAUGE_RAMP

9.31.1.26 #define GSLC_FEATURE_XTEXTBOX_EMBED

9.31.1.27 #define GSLC_LOCAL_STR

9.31.1.28 #define GSLC_LOCAL_STR_LEN

9.31.1.29 #define GSLC_ROTATE

9.31.1.30 #define GSLC_SD_BUFFPIXEL

9.31.1.31 #define GSLC_SD_EN

9.31.1.32 #define GSLC_TOUCH_MAX_EVT

9.31.1.33 #define GSLC_USE_FLOAT

9.31.1.34 #define GSLC_USE_PROGMEM

9.31.1.35 #define TOUCH_ROTATION_DATA

9.31.1.36 #define TOUCH_ROTATION_FLIPX(*rotation*)

9.31.1.37 #define TOUCH_ROTATION_FLIPY(*rotation*)

9.31.1.38 #define TOUCH_ROTATION_SWAPXY(*rotation*)

9.32 src/GUIslice_config_linux.h File Reference

Macros

- #define DRV_DISP_SDL1
- #define DRV_TOUCH_TSLIB
- #define GSLC_FEATURE_COMPOUND
- #define GSLC_FEATURE_XGAUGE_RADIAL
- #define GSLC_FEATURE_XGAUGE_RAMP
- #define GSLC_FEATURE_XTEXTBOX_EMBED
- #define GSLC_FEATURE_INPUT
- #define DEBUG_ERR
- #define GSLC_DEV_FB
- #define GSLC_DEV_TOUCH
- #define GSLC_DEV_VID_DRV
- #define DRV_SDL_FIX_START
- #define DRV_SDL_MOUSE_SHOW
- #define GSLC_LOCAL_STR
- #define GSLC_USE_FLOAT
- #define ADATOUCH_SWAP_XY
- #define ADATOUCH_FLIP_X
- #define ADATOUCH_FLIP_Y
- #define GSLC_TOUCH_MAX_EVT
- #define GSLC_LOCAL_STR_LEN
- #define GSLC_BMP_TRANS_EN
- #define GSLC_BMP_TRANS_RGB
- #define GSLC_USE_PROGMEM

9.32.1 Macro Definition Documentation

9.32.1.1 #define ADATOUCH_FLIP_X

9.32.1.2 #define ADATOUCH_FLIP_Y

9.32.1.3 #define ADATOUCH_SWAP_XY

9.32.1.4 #define DEBUG_ERR

9.32.1.5 #define DRV_DISP_SDL1

9.32.1.6 #define DRV_SDL_FIX_START

9.32.1.7 #define DRV_SDL_MOUSE_SHOW

9.32.1.8 #define DRV_TOUCH_TSLIB

9.32.1.9 #define GSLC_BMP_TRANS_EN

9.32.1.10 #define GSLC_BMP_TRANS_RGB

9.32.1.11 #define GSLC_DEV_FB

9.32.1.12 #define GSLC_DEV_TOUCH

9.32.1.13 #define GSLC_DEV_VID_DRV

9.32.1.14 #define GSLC_FEATURE_COMPOUND

9.32.1.15 #define GSLC_FEATURE_INPUT

9.32.1.16 #define GSLC_FEATURE_XGAUGE_RADIAL

9.32.1.17 #define GSLC_FEATURE_XGAUGE_RAMP

9.32.1.18 #define GSLC_FEATURE_XTEXTBOX_EMBED

9.32.1.19 #define GSLC_LOCAL_STR

9.32.1.20 #define GSLC_LOCAL_STR_LEN

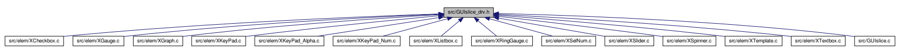
9.32.1.21 #define GSLC_TOUCH_MAX_EVT

9.32.1.22 #define GSLC_USE_FLOAT

9.32.1.23 #define GSLC_USE_PROGMEM

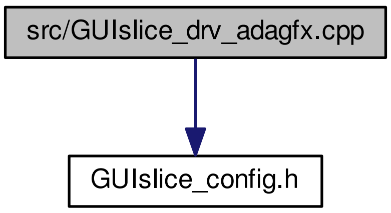
9.33 src/GUIslice_drv.h File Reference

This graph shows which files directly or indirectly include this file:



9.34 src/GUIslice_drv_adagfx.cpp File Reference

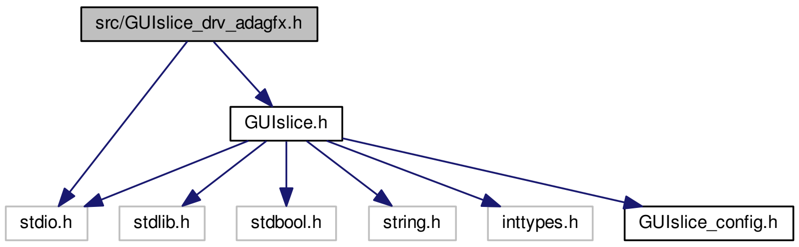
```
#include "GUIslice_config.h"
```
Include dependency graph for GUIslice_drv_adagfx.cpp:



9.35 src/GUIslice_drv_adagfx.h File Reference

GUIslice library (driver layer for Adafruit-GFX)

```
#include "GUIslice.h"
#include <stdio.h>
```
Include dependency graph for GUIslice_drv_adagfx.h:



Data Structures

- struct gslc_tsDriver

Macros

- #define DRV_HAS_DRAW_POINT

  Support gslc_DrvDrawPoint()
- #define DRV_HAS_DRAW_POINTS

  Support gslc_DrvDrawPoints()

- #define DRV_HAS_DRAW_LINE

  Support gslc_DrvDrawLine()
- #define DRV_HAS_DRAW_RECT_FRAME

  Support gslc_DrvDrawFrameRect()
- #define DRV_HAS_DRAW_RECT_FILL

  Support gslc_DrvDrawFillRect()
- #define DRV_HAS_DRAW_RECT_ROUND_FRAME

  Support gslc_DrvDrawFrameRoundRect()
- #define DRV_HAS_DRAW_RECT_ROUND_FILL

  Support gslc_DrvDrawFillRoundRect()
- #define DRV_HAS_DRAW_CIRCLE_FRAME

  Support gslc_DrvDrawFrameCircle()
- #define DRV_HAS_DRAW_CIRCLE_FILL

  Support gslc_DrvDrawFillCircle()
- #define DRV_HAS_DRAW_TRI_FRAME

  Support gslc_DrvDrawFrameTriangle()
- #define DRV_HAS_DRAW_TRI_FILL

  Support gslc_DrvDrawFillTriangle()
- #define DRV_HAS_DRAW_TEXT

  Support gslc_DrvDrawTxt()
- #define DRV_OVERRIDE_TXT_ALIGN

  Driver provides text alignment.

Functions

- bool gslc_DrvInit (gslc_tsGui ∗pGui)

  Initialize the SDL library.
- bool gslc_DrvInitTs (gslc_tsGui ∗pGui, const char ∗acDev)

  Perform any touchscreen-specific initialization.
- void gslc_DrvDestruct (gslc_tsGui ∗pGui)

  Free up any members associated with the driver.
- const char ∗ gslc_DrvGetNameDisp (gslc_tsGui ∗pGui)

  Get the display driver name.
- const char ∗ gslc_DrvGetNameTouch (gslc_tsGui ∗pGui)

  Get the touch driver name.
- void ∗ gslc_DrvLoadImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Load a bitmap (∗.bmp) and create a new image resource.
- bool gslc_DrvSetBkgndImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Configure the background to use a bitmap image.
- bool gslc_DrvSetBkgndColor (gslc_tsGui ∗pGui, gslc_tsColor nCol)

  Configure the background to use a solid color.
- bool gslc_DrvSetElemImageNorm (gslc_tsGui ∗pGui, gslc_tsElem ∗pElem, gslc_tsImgRef sImgRef)

  Set an element's normal-state image.
- bool gslc_DrvSetElemImageGlow (gslc_tsGui ∗pGui, gslc_tsElem ∗pElem, gslc_tsImgRef sImgRef)

  Set an element's glow-state image.
- void gslc_DrvImageDestruct (void ∗pvImg)

  Release an image surface.
- bool gslc_DrvSetClipRect (gslc_tsGui ∗pGui, gslc_tsRect ∗pRect)

  Set the clipping rectangle for future drawing updates.
- const void ∗ gslc_DrvFontAdd (gslc_teFontRefType eFontRefType, const void ∗pvFontRef, uint16_t nFontSz)

Load a font from a resource and return pointer to it.

- void gslc_DrvFontsDestruct (gslc_tsGui ∗pGui)

  Release all fonts defined in the GUI.

- bool gslc_DrvGetTxtSize (gslc_tsGui ∗pGui, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxt←Flags, int16_t ∗pnTxtX, int16_t ∗pnTxtY, uint16_t ∗pnTxtSzW, uint16_t ∗pnTxtSzH)

  Get the extent (width and height) of a text string.

- bool gslc_DrvDrawTxt (gslc_tsGui ∗pGui, int16_t nTxtX, int16_t nTxtY, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxtFlags, gslc_tsColor colTxt, gslc_tsColor colBg)

  Draw a text string at the given coordinate.

- void gslc_DrvPageFlipNow (gslc_tsGui ∗pGui)

  Force a page flip to occur.

- bool gslc_DrvDrawPoint (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, gslc_tsColor nCol)

  Draw a point.

- bool gslc_DrvDrawPoints (gslc_tsGui ∗pGui, gslc_tsPt ∗asPt, uint16_t nNumPt, gslc_tsColor nCol)

  Draw a point.

- bool gslc_DrvDrawFrameRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a framed rectangle.

- bool gslc_DrvDrawFillRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a filled rectangle.

- bool gslc_DrvDrawFrameRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor n←Col)

  Draw a framed rounded rectangle.

- bool gslc_DrvDrawFillRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

  Draw a filled rounded rectangle.

- bool gslc_DrvDrawLine (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)

  Draw a line.

- bool gslc_DrvDrawFrameCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_ts←Color nCol)

  Draw a framed circle.

- bool gslc_DrvDrawFillCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)

  Draw a filled circle.

- bool gslc_DrvDrawFrameTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

  Draw a framed triangle.

- bool gslc_DrvDrawFillTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

  Draw a filled triangle.

- bool gslc_DrvDrawImage (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, gslc_tsImgRef sImgRef)

  Copy all of source image to destination screen at specified coordinate.

- void gslc_DrvDrawMonoFromMem (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, const unsigned char ∗p←Bitmap, bool bProgMem)

  Draw a monochrome bitmap from a memory array.

- void gslc_DrvDrawBmp24FromMem (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, const unsigned char ∗pBitmap, bool bProgMem)

  Draw a color 24-bit depth bitmap from a memory array.

- void gslc_DrvDrawBkgnd (gslc_tsGui ∗pGui)

  Copy the background image to destination screen.

- bool gslc_DrvInitTouch (gslc_tsGui ∗pGui, const char ∗acDev)

  Perform any touchscreen-specific initialization.

- bool gslc_DrvGetTouch (gslc_tsGui ∗pGui, int16_t ∗pnX, int16_t ∗pnY, uint16_t ∗pnPress, gslc_teInputRaw←Event ∗peInputEvent, int16_t ∗pnInputVal)

  Get the last touch event from the internal touch handler.
- bool gslc_DrvRotate (gslc_tsGui ∗pGui, uint8_t nRotation)

  Change rotation, automatically adapt touchscreen axes swap/flip.
- uint16_t gslc_DrvAdaptColorToRaw (gslc_tsColor nCol)

9.35.1 Detailed Description

GUIslice library (driver layer for Adafruit-GFX)

9.35.2 Macro Definition Documentation

9.35.2.1 #define DRV_HAS_DRAW_CIRCLE_FILL

Support gslc_DrvDrawFillCircle()

9.35.2.2 #define DRV_HAS_DRAW_CIRCLE_FRAME

Support gslc_DrvDrawFrameCircle()

9.35.2.3 #define DRV_HAS_DRAW_LINE

Support gslc_DrvDrawLine()

9.35.2.4 #define DRV_HAS_DRAW_POINT

Support gslc_DrvDrawPoint()

9.35.2.5 #define DRV_HAS_DRAW_POINTS

Support gslc_DrvDrawPoints()

9.35.2.6 #define DRV_HAS_DRAW_RECT_FILL

Support gslc_DrvDrawFillRect()

9.35.2.7 #define DRV_HAS_DRAW_RECT_FRAME

Support gslc_DrvDrawFrameRect()

9.35.2.8 #define DRV_HAS_DRAW_RECT_ROUND_FILL

Support gslc_DrvDrawFillRoundRect()

9.35.2.9 #define DRV_HAS_DRAW_RECT_ROUND_FRAME

Support gslc_DrvDrawFrameRoundRect()

9.35.2.10 #define DRV_HAS_DRAW_TEXT

Support gslc_DrvDrawTxt()

9.35.2.11 #define DRV_HAS_DRAW_TRI_FILL

Support gslc_DrvDrawFillTriangle()

9.35.2.12 #define DRV_HAS_DRAW_TRI_FRAME

Support gslc_DrvDrawFrameTriangle()

9.35.2.13 #define DRV_OVERRIDE_TXT_ALIGN

Driver provides text alignment.

9.35.3 Function Documentation

9.35.3.1 uint16_t gslc_DrvAdaptColorToRaw (gslc_tsColor *nCol*)

9.35.3.2 void gslc_DrvDestruct (gslc_tsGui $*$ *pGui*)

Free up any members associated with the driver.

- Eg. renderers, windows, background surfaces, etc.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.35.3.3 void gslc_DrvDrawBkgnd (gslc_tsGui ∗ *pGui*)

Copy the background image to destination screen.

Parameters

in	*pGui*	Pointer to GUI

Returns

true if success, false if fail

9.35.3.4 void gslc_DrvDrawBmp24FromMem (gslc_tsGui ∗ *pGui,* int16_t *nDstX,* int16_t *nDstY,* const unsigned char ∗ *pBitmap,* bool *bProgMem*)

Draw a color 24-bit depth bitmap from a memory array.

- Note that users must convert images from their native format (eg. BMP, PNG, etc.) into a C array. Please refer to the following guide for details: https://github.com/ImpulseAdventure/GU←Islice/wiki/Display-Images-from-FLASH
- The converted file (c array) can then be included in the sketch.

Parameters

in	*pGui*	Pointer to GUI
in	*nDstX*	X coord for copy
in	*nDstY*	Y coord for copy
in	*pBitmap*	Pointer to bitmap buffer
in	*bProgMem*	Bitmap is stored in Flash if true, RAM otherwise

Returns

none

9.35.3.5 bool gslc_DrvDrawFillCircle (gslc_tsGui ∗ *pGui,* int16_t *nMidX,* int16_t *nMidY,* uint16_t *nRadius,* gslc_tsColor *nCol*)

Draw a filled circle.

Parameters

in	*pGui*	Pointer to GUI
in	*nMidX*	Center of circle (X coordinate)
in	*nMidY*	Center of circle (Y coordinate)
in	*nRadius*	Radius of circle
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.35.3.6 bool gslc_DrvDrawFillRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* gslc_tsColor *nCol*)

Draw a filled rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.35.3.7 bool gslc_DrvDrawFillRoundRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* int16_t *nRadius,* gslc_tsColor *nCol*)

Draw a filled rounded rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nRadius*	Radius for rounded corners
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.35.3.8 bool gslc_DrvDrawFillTriangle (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* int16_t *nX2,* int16_t *nY2,* gslc_tsColor *nCol*)

Draw a filled triangle.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X Coordinate #1
in	*nY0*	Y Coordinate #1
in	*nX1*	X Coordinate #2
in	*nY1*	Y Coordinate #2
in	*nX2*	X Coordinate #3
in	*nY2*	Y Coordinate #3
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.35.3.9 bool gslc_DrvDrawFrameCircle (gslc_tsGui ∗ *pGui*, int16_t *nMidX*, int16_t *nMidY*, uint16_t *nRadius*, gslc_tsColor *nCol*)

Draw a framed circle.

Parameters

in	*pGui*	Pointer to GUI
in	*nMidX*	Center of circle (X coordinate)
in	*nMidY*	Center of circle (Y coordinate)
in	*nRadius*	Radius of circle
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.35.3.10 bool gslc_DrvDrawFrameRect (gslc_tsGui ∗ *pGui*, gslc_tsRect *rRect*, gslc_tsColor *nCol*)

Draw a framed rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.35.3.11 bool gslc_DrvDrawFrameRoundRect (gslc_tsGui ∗ *pGui*, gslc_tsRect *rRect*, int16_t *nRadius*, gslc_tsColor *nCol*)

Draw a framed rounded rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nRadius*	Radius for rounded corners
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.35.3.12 bool gslc_DrvDrawFrameTriangle (gslc_tsGui * *pGui*, int16_t *nX0*, int16_t *nY0*, int16_t *nX1*, int16_t *nY1*, int16_t *nX2*, int16_t *nY2*, gslc_tsColor *nCol*)

Draw a framed triangle.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nX0*	X Coordinate #1
`in`	*nY0*	Y Coordinate #1
`in`	*nX1*	X Coordinate #2
`in`	*nY1*	Y Coordinate #2
`in`	*nX2*	X Coordinate #3
`in`	*nY2*	Y Coordinate #3
`in`	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.35.3.13 bool gslc_DrvDrawImage (gslc_tsGui * *pGui*, int16_t *nDstX*, int16_t *nDstY*, gslc_tsImgRef *sImgRef*)

Copy all of source image to destination screen at specified coordinate.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nDstX*	Destination X coord for copy
`in`	*nDstY*	Destination Y coord for copy
`in`	*sImgRef*	Image reference

Returns

true if success, false if fail

9.35.3.14 bool gslc_DrvDrawLine (gslc_tsGui * *pGui*, int16_t *nX0*, int16_t *nY0*, int16_t *nX1*, int16_t *nY1*, gslc_tsColor *nCol*)

Draw a line.

Parameters

`in`	*pGui*	Pointer to GUI

Parameters

`in`	*nX0*	Line start (X coordinate)
`in`	*nY0*	Line start (Y coordinate)
`in`	*nX1*	Line finish (X coordinate)
`in`	*nY1*	Line finish (Y coordinate)
`in`	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.35.3.15 void gslc_DrvDrawMonoFromMem (gslc_tsGui ∗ *pGui,* int16_t *nDstX,* int16_t *nDstY,* const unsigned char ∗ *pBitmap,* bool *bProgMem*)

Draw a monochrome bitmap from a memory array.

- Draw from the bitmap buffer using the foreground color defined in the header (unset bits are transparent)

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nDstX*	Destination X coord for copy
`in`	*nDstY*	Destination Y coord for copy
`in`	*pBitmap*	Pointer to bitmap buffer
`in`	*bProgMem*	Bitmap is stored in Flash if true, RAM otherwise

Returns

none

9.35.3.16 bool gslc_DrvDrawPoint (gslc_tsGui ∗ *pGui,* int16_t *nX,* int16_t *nY,* gslc_tsColor *nCol*)

Draw a point.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nX*	X coordinate of point
`in`	*nY*	Y coordinate of point
`in`	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.35.3.17 bool gslc_DrvDrawPoints (gslc_tsGui ∗ *pGui,* gslc_tsPt ∗ *asPt,* uint16_t *nNumPt,* gslc_tsColor *nCol*)

Draw a point.

Parameters

in	*pGui*	Pointer to GUI
in	*asPt*	Array of points to draw
in	*n←↩ NumPt*	Number of points in array
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.35.3.18 bool gslc_DrvDrawTxt (gslc_tsGui ∗ *pGui,* int16_t *nTxtX,* int16_t *nTxtY,* gslc_tsFont ∗ *pFont,* const char ∗ *pStr,* gslc_teTxtFlags *eTxtFlags,* gslc_tsColor *colTxt,* gslc_tsColor *colBg*)

Draw a text string at the given coordinate.

Parameters

in	*pGui*	Pointer to GUI
in	*nTxtX*	X coordinate of top-left text string
in	*nTxtY*	Y coordinate of top-left text string
in	*pFont*	Ptr to Font
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
in	*colTxt*	Color to draw text
in	*colBg*	unused in ADAGFX, defaults to black

Returns

true if success, false if failure

9.35.3.19 const void∗ gslc_DrvFontAdd (gslc_teFontRefType *eFontRefType,* const void ∗ *pvFontRef,* uint16_t *nFontSz*)

Load a font from a resource and return pointer to it.

Parameters

in	*eFontRefType*	Font reference type (GSLC_FONTREF_PTR for Arduino)
in	*pvFontRef*	Font reference pointer (Pointer to the GFXFont array)
in	*nFontSz*	Typeface size to use

Returns

Void ptr to driver-specific font if load was successful, NULL otherwise

9.35.3.20 void gslc_DrvFontsDestruct (gslc_tsGui ∗ *pGui*)

Release all fonts defined in the GUI.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.35.3.21 const char∗ gslc_DrvGetNameDisp (gslc_tsGui ∗ *pGui*)

Get the display driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

9.35.3.22 const char∗ gslc_DrvGetNameTouch (gslc_tsGui ∗ *pGui*)

Get the touch driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

9.35.3.23 bool gslc_DrvGetTouch (gslc_tsGui ∗ *pGui*, int16_t ∗ *pnX*, int16_t ∗ *pnY*, uint16_t ∗ *pnPress*, gslc_teInputRawEvent ∗ *peInputEvent*, int16_t ∗ *pnInputVal*)

Get the last touch event from the internal touch handler.

Parameters

in	*pGui*	Pointer to GUI
out	*pnX*	Ptr to X coordinate of last touch event
out	*pnY*	Ptr to Y coordinate of last touch event
out	*pnPress*	Ptr to Pressure level of last touch event (0 for none, 1 for touch)
out	*peInputEvent*	Indication of event type
out	*pnInputVal*	Additional data for event type

Returns

true if an event was detected or false otherwise

9.35.3.24 bool gslc_DrvGetTxtSize (gslc_tsGui ∗ *pGui,* gslc_tsFont ∗ *pFont,* const char ∗ *pStr,* gslc_teTxtFlags *eTxtFlags,* int16_t ∗ *pnTxtX,* int16_t ∗ *pnTxtY,* uint16_t ∗ *pnTxtSzW,* uint16_t ∗ *pnTxtSzH*)

Get the extent (width and height) of a text string.

Parameters

in	*pGui*	Pointer to GUI
in	*pFont*	Ptr to Font structure
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
out	*pnTxtX*	Ptr to offset X of text
out	*pnTxtY*	Ptr to offset Y of text
out	*pnTxtSzW*	Ptr to width of text
out	*pnTxtSzH*	Ptr to height of text

Returns

true if success, false if failure

9.35.3.25 void gslc_DrvImageDestruct (void ∗ *pvImg*)

Release an image surface.

Parameters

in	*pvImg*	Void ptr to image

Returns

none

9.35.3.26 bool gslc_DrvInit (gslc_tsGui * *pGui*)

Initialize the SDL library.

- Performs clean startup workaround (if enabled)
- Configures video mode
- Initializes font support

PRE:

- The environment variables should be configured before calling gslc_DrvInit(). This can be done with gslc_↩ DrvInitEnv() or manually in user function.

Parameters

in	*pGui*	Pointer to GUI

Returns

true if success, false if fail

9.35.3.27 bool gslc_DrvInitTouch (gslc_tsGui * *pGui*, const char * *acDev*)

Perform any touchscreen-specific initialization.

Parameters

in	*pGui*	Pointer to GUI
in	*acDev*	Device path to touchscreen eg. "/dev/input/touchscreen"

Returns

true if successful

9.35.3.28 bool gslc_DrvInitTs (gslc_tsGui * *pGui*, const char * *acDev*)

Perform any touchscreen-specific initialization.

Parameters

in	*pGui*	Pointer to GUI
in	*acDev*	Device path to touchscreen eg. "/dev/input/touchscreen"

Returns

true if successful

9.35.3.29 void∗ gslc_DrvLoadImage (gslc_tsGui ∗ *pGui,* gslc_tsImgRef *sImgRef*)

Load a bitmap (∗.bmp) and create a new image resource.

Transparency is enabled by GSLC_BMP_TRANS_EN through use of color (GSLC_BMP_TRANS_RGB).

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*sImgRef*	Image reference

Returns

Image pointer (surface/texture) or NULL if error

9.35.3.30 void gslc_DrvPageFlipNow (gslc_tsGui ∗ *pGui*)

Force a page flip to occur.

This generally copies active screen surface to the display.

Parameters

`in`	*pGui*	Pointer to GUI

Returns

none

9.35.3.31 bool gslc_DrvRotate (gslc_tsGui ∗ *pGui,* uint8_t *nRotation*)

Change rotation, automatically adapt touchscreen axes swap/flip.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*nRotation*	Screen Rotation value (0, 1, 2 or 3)

Returns

true if successful

9.35.3.32 bool gslc_DrvSetBkgndColor (gslc_tsGui ∗ *pGui,* gslc_tsColor *nCol*)

Configure the background to use a solid color.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*nCol*	RGB Color to use

Returns

true if success, false if fail

9.35.3.33 bool gslc_DrvSetBkgndImage (gslc_tsGui ∗ *pGui,* gslc_tsImgRef *sImgRef*)

Configure the background to use a bitmap image.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*sImgRef*	Image reference

Returns

true if success, false if fail

9.35.3.34 bool gslc_DrvSetClipRect (gslc_tsGui ∗ *pGui,* gslc_tsRect ∗ *pRect*)

Set the clipping rectangle for future drawing updates.

Parameters

in	*pGui*	Pointer to GUI
in	*pRect*	Rectangular region to constrain edits

Returns

true if success, false if error

9.35.3.35 bool gslc_DrvSetElemImageGlow (gslc_tsGui ∗ *pGui,* gslc_tsElem ∗ *pElem,* gslc_tsImgRef *sImgRef*)

Set an element's glow-state image.

Parameters

in	*pGui*	Pointer to GUI
in	*pElem*	Pointer to Element to update
in	*sImgRef*	Image reference

Returns

true if success, false if error

9.35.3.36 bool gslc_DrvSetElemImageNorm (gslc_tsGui ∗ *pGui,* gslc_tsElem ∗ *pElem,* gslc_tsImgRef *sImgRef*)

Set an element's normal-state image.

Parameters

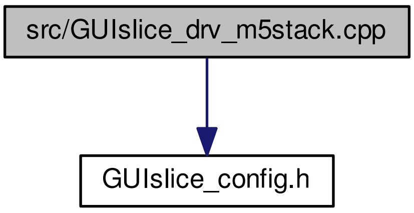
in	*pGui*	Pointer to GUI
in	*pElem*	Pointer to Element to update
in	*sImgRef*	Image reference

Returns

true if success, false if error

9.36 src/GUIslice_drv_m5stack.cpp File Reference

```
#include "GUIslice_config.h"
```
Include dependency graph for GUIslice_drv_m5stack.cpp:

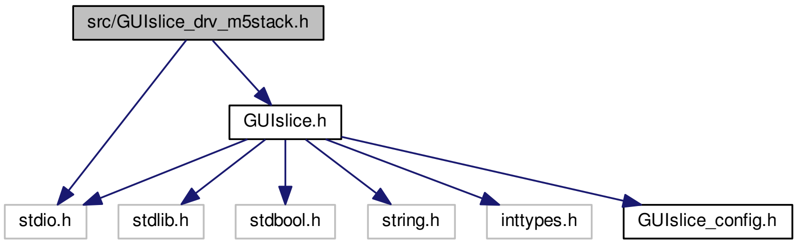


9.37 src/GUIslice_drv_m5stack.h File Reference

GUIslice library (driver layer for M5stack)

```
#include "GUIslice.h"
#include <stdio.h>
```

Include dependency graph for GUIslice_drv_m5stack.h:



Data Structures

- struct gslc_tsDriver

Macros

- #define DRV_HAS_DRAW_POINT

    Support gslc_DrvDrawPoint()
- #define DRV_HAS_DRAW_POINTS

    Support gslc_DrvDrawPoints()
- #define DRV_HAS_DRAW_LINE

    Support gslc_DrvDrawLine()
- #define DRV_HAS_DRAW_RECT_FRAME

    Support gslc_DrvDrawFrameRect()
- #define DRV_HAS_DRAW_RECT_FILL

    Support gslc_DrvDrawFillRect()
- #define DRV_HAS_DRAW_RECT_ROUND_FRAME

    Support gslc_DrvDrawFrameRoundRect()
- #define DRV_HAS_DRAW_RECT_ROUND_FILL

    Support gslc_DrvDrawFillRoundRect()
- #define DRV_HAS_DRAW_CIRCLE_FRAME

    Support gslc_DrvDrawFrameCircle()
- #define DRV_HAS_DRAW_CIRCLE_FILL

    Support gslc_DrvDrawFillCircle()
- #define DRV_HAS_DRAW_TRI_FRAME

    Support gslc_DrvDrawFrameTriangle()
- #define DRV_HAS_DRAW_TRI_FILL

    Support gslc_DrvDrawFillTriangle()
- #define DRV_HAS_DRAW_TEXT

    Support gslc_DrvDrawTxt()
- #define DRV_OVERRIDE_TXT_ALIGN

    Driver provides text alignment.

Functions

- bool gslc_DrvInit (gslc_tsGui ∗pGui)

  Initialize the SDL library.
- bool gslc_DrvInitTs (gslc_tsGui ∗pGui, const char ∗acDev)

  Perform any touchscreen-specific initialization.
- void gslc_DrvDestruct (gslc_tsGui ∗pGui)

  Free up any members associated with the driver.
- const char ∗ gslc_DrvGetNameDisp (gslc_tsGui ∗pGui)

  Get the display driver name.
- const char ∗ gslc_DrvGetNameTouch (gslc_tsGui ∗pGui)

  Get the touch driver name.
- void ∗ gslc_DrvLoadImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Load a bitmap (∗.bmp) and create a new image resource.
- bool gslc_DrvSetBkgndImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Configure the background to use a bitmap image.
- bool gslc_DrvSetBkgndColor (gslc_tsGui ∗pGui, gslc_tsColor nCol)

  Configure the background to use a solid color.
- bool gslc_DrvSetElemImageNorm (gslc_tsGui ∗pGui, gslc_tsElem ∗pElem, gslc_tsImgRef sImgRef)

  Set an element's normal-state image.
- bool gslc_DrvSetElemImageGlow (gslc_tsGui ∗pGui, gslc_tsElem ∗pElem, gslc_tsImgRef sImgRef)

  Set an element's glow-state image.
- void gslc_DrvImageDestruct (void ∗pvImg)

  Release an image surface.
- bool gslc_DrvSetClipRect (gslc_tsGui ∗pGui, gslc_tsRect ∗pRect)

  Set the clipping rectangle for future drawing updates.
- const void ∗ gslc_DrvFontAdd (gslc_teFontRefType eFontRefType, const void ∗pvFontRef, uint16_t nFontSz)

  Load a font from a resource and return pointer to it.
- void gslc_DrvFontsDestruct (gslc_tsGui ∗pGui)

  Release all fonts defined in the GUI.
- bool gslc_DrvGetTxtSize (gslc_tsGui ∗pGui, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxt←Flags, int16_t ∗pnTxtX, int16_t ∗pnTxtY, uint16_t ∗pnTxtSzW, uint16_t ∗pnTxtSzH)

  Get the extent (width and height) of a text string.
- bool gslc_DrvDrawTxt (gslc_tsGui ∗pGui, int16_t nTxtX, int16_t nTxtY, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxtFlags, gslc_tsColor colTxt, gslc_tsColor colBg)

  Draw a text string at the given coordinate.
- bool gslc_DrvDrawTxtAlign (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int8_t e←TxtAlign, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxtFlags, gslc_tsColor colTxt, gslc_tsColor colBg)

  Draw a text string in a bounding box using the specified alignment.
- void gslc_DrvPageFlipNow (gslc_tsGui ∗pGui)

  Force a page flip to occur.
- bool gslc_DrvDrawPoint (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, gslc_tsColor nCol)

  Draw a point.
- bool gslc_DrvDrawPoints (gslc_tsGui ∗pGui, gslc_tsPt ∗asPt, uint16_t nNumPt, gslc_tsColor nCol)

  Draw a point.
- bool gslc_DrvDrawFrameRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a framed rectangle.
- bool gslc_DrvDrawFillRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a filled rectangle.
- bool gslc_DrvDrawFrameRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor n←Col)

Draw a framed rounded rectangle.
- bool gslc_DrvDrawFillRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

  Draw a filled rounded rectangle.
- bool gslc_DrvDrawLine (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)

  Draw a line.
- bool gslc_DrvDrawFrameCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_ts←Color nCol)

  Draw a framed circle.
- bool gslc_DrvDrawFillCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)

  Draw a filled circle.
- bool gslc_DrvDrawFrameTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

  Draw a framed triangle.
- bool gslc_DrvDrawFillTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

  Draw a filled triangle.
- bool gslc_DrvDrawImage (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, gslc_tsImgRef sImgRef)

  Copy all of source image to destination screen at specified coordinate.
- void gslc_DrvDrawMonoFromMem (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, const unsigned char ∗p←Bitmap, bool bProgMem)

  Draw a monochrome bitmap from a memory array.
- void gslc_DrvDrawBmp24FromMem (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, const unsigned char ∗pBitmap, bool bProgMem)

  Draw a color 24-bit depth bitmap from a memory array.
- void gslc_DrvDrawBkgnd (gslc_tsGui ∗pGui)

  Copy the background image to destination screen.
- bool gslc_DrvRotate (gslc_tsGui ∗pGui, uint8_t nRotation)

  Change rotation, automatically adapt touchscreen axes swap/flip.
- uint16_t gslc_DrvAdaptColorToRaw (gslc_tsColor nCol)

Variables

- const char GSLC_PMEM ERRSTR_NULL [ ]
- const char GSLC_PMEM ERRSTR_PXD_NULL [ ]

9.37.1 Detailed Description

GUIslice library (driver layer for M5stack)

9.37.2 Macro Definition Documentation

9.37.2.1 #define DRV_HAS_DRAW_CIRCLE_FILL

Support gslc_DrvDrawFillCircle()

9.37.2.2 #define DRV_HAS_DRAW_CIRCLE_FRAME

Support gslc_DrvDrawFrameCircle()

9.37.2.3 #define DRV_HAS_DRAW_LINE

Support gslc_DrvDrawLine()

9.37.2.4 #define DRV_HAS_DRAW_POINT

Support gslc_DrvDrawPoint()

9.37.2.5 #define DRV_HAS_DRAW_POINTS

Support gslc_DrvDrawPoints()

9.37.2.6 #define DRV_HAS_DRAW_RECT_FILL

Support gslc_DrvDrawFillRect()

9.37.2.7 #define DRV_HAS_DRAW_RECT_FRAME

Support gslc_DrvDrawFrameRect()

9.37.2.8 #define DRV_HAS_DRAW_RECT_ROUND_FILL

Support gslc_DrvDrawFillRoundRect()

9.37.2.9 #define DRV_HAS_DRAW_RECT_ROUND_FRAME

Support gslc_DrvDrawFrameRoundRect()

9.37.2.10 #define DRV_HAS_DRAW_TEXT

Support gslc_DrvDrawTxt()

9.37.2.11 #define DRV_HAS_DRAW_TRI_FILL

Support gslc_DrvDrawFillTriangle()

9.37.2.12 #define DRV_HAS_DRAW_TRI_FRAME

Support gslc_DrvDrawFrameTriangle()

9.37.2.13 #define DRV_OVERRIDE_TXT_ALIGN

Driver provides text alignment.

9.37.3 Function Documentation

9.37.3.1 uint16_t gslc_DrvAdaptColorToRaw (gslc_tsColor *nCol*)

9.37.3.2 void gslc_DrvDestruct (gslc_tsGui ∗ *pGui*)

Free up any members associated with the driver.

- Eg. renderers, windows, background surfaces, etc.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.37.3.3 void gslc_DrvDrawBkgnd (gslc_tsGui ∗ *pGui*)

Copy the background image to destination screen.

Parameters

in	*pGui*	Pointer to GUI

Returns

true if success, false if fail

9.37.3.4 void gslc_DrvDrawBmp24FromMem (gslc_tsGui ∗ *pGui,* int16_t *nDstX,* int16_t *nDstY,* const unsigned char ∗ *pBitmap,* bool *bProgMem*)

Draw a color 24-bit depth bitmap from a memory array.

- Note that users must convert images from their native format (eg. BMP, PNG, etc.) into a C array. Please refer to the following guide for details: https://github.com/ImpulseAdventure/GUIslice/wiki/Display-Images-from-FLASH
- The converted file (c array) can then be included in the sketch.

Parameters

in	*pGui*	Pointer to GUI
in	*nDstX*	X coord for copy
in	*nDstY*	Y coord for copy
in	*pBitmap*	Pointer to bitmap buffer
in	*bProgMem*	Bitmap is stored in Flash if true, RAM otherwise

Returns

none

9.37.3.5 bool gslc_DrvDrawFillCircle (gslc_tsGui ∗ *pGui,* int16_t *nMidX,* int16_t *nMidY,* uint16_t *nRadius,* gslc_tsColor *nCol*)

Draw a filled circle.

Parameters

in	*pGui*	Pointer to GUI
in	*nMidX*	Center of circle (X coordinate)
in	*nMidY*	Center of circle (Y coordinate)
in	*nRadius*	Radius of circle
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.37.3.6 bool gslc_DrvDrawFillRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* gslc_tsColor *nCol*)

Draw a filled rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.37.3.7 bool gslc_DrvDrawFillRoundRect (**gslc_tsGui** ∗ *pGui,* **gslc_tsRect** *rRect,* int16_t *nRadius,* **gslc_tsColor** *nCol*)

Draw a filled rounded rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nRadius*	Radius for rounded corners
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.37.3.8 bool gslc_DrvDrawFillTriangle (**gslc_tsGui** ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* int16_t *nX2,* int16_t *nY2,* **gslc_tsColor** *nCol*)

Draw a filled triangle.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X Coordinate #1
in	*nY0*	Y Coordinate #1
in	*nX1*	X Coordinate #2
in	*nY1*	Y Coordinate #2
in	*nX2*	X Coordinate #3
in	*nY2*	Y Coordinate #3
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.37.3.9 bool gslc_DrvDrawFrameCircle (**gslc_tsGui** ∗ *pGui,* int16_t *nMidX,* int16_t *nMidY,* uint16_t *nRadius,* **gslc_tsColor** *nCol*)

Draw a framed circle.

Parameters

in	*pGui*	Pointer to GUI

Parameters

in	*nMidX*	Center of circle (X coordinate)
in	*nMidY*	Center of circle (Y coordinate)
in	*nRadius*	Radius of circle
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.37.3.10 bool gslc_DrvDrawFrameRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* gslc_tsColor *nCol*)

Draw a framed rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.37.3.11 bool gslc_DrvDrawFrameRoundRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* int16_t *nRadius,* gslc_tsColor *nCol*)

Draw a framed rounded rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nRadius*	Radius for rounded corners
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.37.3.12 bool gslc_DrvDrawFrameTriangle (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* int16_t *nX2,* int16_t *nY2,* gslc_tsColor *nCol*)

Draw a framed triangle.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X Coordinate #1
in	*nY0*	Y Coordinate #1
in	*nX1*	X Coordinate #2
in	*nY1*	Y Coordinate #2
in	*nX2*	X Coordinate #3
in	*nY2*	Y Coordinate #3
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.37.3.13 bool gslc_DrvDrawImage (gslc_tsGui ∗ *pGui,* int16_t *nDstX,* int16_t *nDstY,* gslc_tsImgRef *sImgRef*)

Copy all of source image to destination screen at specified coordinate.

Parameters

in	*pGui*	Pointer to GUI
in	*nDstX*	Destination X coord for copy
in	*nDstY*	Destination Y coord for copy
in	*sImgRef*	Image reference

Returns

true if success, false if fail

9.37.3.14 bool gslc_DrvDrawLine (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* gslc_tsColor *nCol*)

Draw a line.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	Line start (X coordinate)
in	*nY0*	Line start (Y coordinate)
in	*nX1*	Line finish (X coordinate)
in	*nY1*	Line finish (Y coordinate)
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.37.3.15 void gslc_DrvDrawMonoFromMem (gslc_tsGui ∗ *pGui,* int16_t *nDstX,* int16_t *nDstY,* const unsigned char ∗ *pBitmap,* bool *bProgMem*)

Draw a monochrome bitmap from a memory array.

- Draw from the bitmap buffer using the foreground color defined in the header (unset bits are transparent)

Parameters

in	*pGui*	Pointer to GUI
in	*nDstX*	Destination X coord for copy
in	*nDstY*	Destination Y coord for copy
in	*pBitmap*	Pointer to bitmap buffer
in	*bProgMem*	Bitmap is stored in Flash if true, RAM otherwise

Returns

none

9.37.3.16 bool gslc_DrvDrawPoint (gslc_tsGui ∗ *pGui,* int16_t *nX,* int16_t *nY,* gslc_tsColor *nCol*)

Draw a point.

Parameters

in	*pGui*	Pointer to GUI
in	*nX*	X coordinate of point
in	*nY*	Y coordinate of point
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.37.3.17 bool gslc_DrvDrawPoints (gslc_tsGui ∗ *pGui,* gslc_tsPt ∗ *asPt,* uint16_t *nNumPt,* gslc_tsColor *nCol*)

Draw a point.

Parameters

in	*pGui*	Pointer to GUI

Parameters

in	*asPt*	Array of points to draw
in	*n↩ NumPt*	Number of points in array
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.37.3.18 bool gslc_DrvDrawTxt (gslc_tsGui ∗ *pGui*, int16_t *nTxtX*, int16_t *nTxtY*, gslc_tsFont ∗ *pFont*, const char ∗ *pStr*, gslc_teTxtFlags *eTxtFlags*, gslc_tsColor *colTxt*, gslc_tsColor *colBg*)

Draw a text string at the given coordinate.

Parameters

in	*pGui*	Pointer to GUI
in	*nTxtX*	X coordinate of top-left text string
in	*nTxtY*	Y coordinate of top-left text string
in	*pFont*	Ptr to Font
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
in	*colTxt*	Color to draw text
in	*colBg*	unused in m5stack, defaults to black

Returns

true if success, false if failure

9.37.3.19 bool gslc_DrvDrawTxtAlign (gslc_tsGui ∗ *pGui*, int16_t *nX0*, int16_t *nY0*, int16_t *nX1*, int16_t *nY1*, int8_t *eTxtAlign*, gslc_tsFont ∗ *pFont*, const char ∗ *pStr*, gslc_teTxtFlags *eTxtFlags*, gslc_tsColor *colTxt*, gslc_tsColor *colBg*)

Draw a text string in a bounding box using the specified alignment.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X coordinate of top-left of bounding box
in	*nY0*	Y coordinate of top-left of bounding box
in	*nX1*	X coordinate of bot-right of bounding box
in	*nY1*	Y coordinate of bot-right of bounding box
in	*eTxtAlign*	Alignment mode]
in	*pFont*	Ptr to Font
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
in	*colTxt*	Color to draw text
in	*colBg*	unused in m5stack, defaults to black

Returns

true if success, false if failure

9.37.3.20 const void∗ gslc_DrvFontAdd (gslc_teFontRefType *eFontRefType,* const void ∗ *pvFontRef,* uint16_t *nFontSz*)

Load a font from a resource and return pointer to it.

Parameters

in	*eFontRefType*	Font reference type (GSLC_FONTREF_PTR for Arduino)
in	*pvFontRef*	Font reference pointer (Pointer to the GFXFont array)
in	*nFontSz*	Typeface size to use

Returns

Void ptr to driver-specific font if load was successful, NULL otherwise

9.37.3.21 void gslc_DrvFontsDestruct (gslc_tsGui ∗ *pGui*)

Release all fonts defined in the GUI.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.37.3.22 const char∗ gslc_DrvGetNameDisp (gslc_tsGui ∗ *pGui*)

Get the display driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

9.37.3.23 const char∗ gslc_DrvGetNameTouch (gslc_tsGui ∗ *pGui*)

Get the touch driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

9.37.3.24 bool gslc_DrvGetTxtSize (gslc_tsGui ∗ *pGui,* gslc_tsFont ∗ *pFont,* const char ∗ *pStr,* gslc_teTxtFlags *eTxtFlags,* int16_t ∗ *pnTxtX,* int16_t ∗ *pnTxtY,* uint16_t ∗ *pnTxtSzW,* uint16_t ∗ *pnTxtSzH*)

Get the extent (width and height) of a text string.

Parameters

in	*pGui*	Pointer to GUI
in	*pFont*	Ptr to Font structure
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
out	*pnTxtX*	Ptr to offset X of text
out	*pnTxtY*	Ptr to offset Y of text
out	*pnTxtSzW*	Ptr to width of text
out	*pnTxtSzH*	Ptr to height of text

Returns

true if success, false if failure

9.37.3.25 void gslc_DrvImageDestruct (void ∗ *pvImg*)

Release an image surface.

Parameters

in	*pvImg*	Void ptr to image

Returns

none

9.37.3.26 bool gslc_DrvInit (gslc_tsGui ∗ *pGui*)

Initialize the SDL library.

- Performs clean startup workaround (if enabled)

- Configures video mode
- Initializes font support

PRE:

- The environment variables should be configured before calling gslc_DrvInit(). This can be done with gslc_↩ DrvInitEnv() or manually in user function.

Parameters

in	*pGui*	Pointer to GUI

Returns

true if success, false if fail

9.37.3.27 bool gslc_DrvInitTs (gslc_tsGui ∗ *pGui,* const char ∗ *acDev*)

Perform any touchscreen-specific initialization.

Parameters

in	*pGui*	Pointer to GUI
in	*acDev*	Device path to touchscreen eg. "/dev/input/touchscreen"

Returns

true if successful

9.37.3.28 void∗ gslc_DrvLoadImage (gslc_tsGui ∗ *pGui,* gslc_tsImgRef *sImgRef*)

Load a bitmap (∗.bmp) and create a new image resource.

Transparency is enabled by GSLC_BMP_TRANS_EN through use of color (GSLC_BMP_TRANS_RGB).

Parameters

in	*pGui*	Pointer to GUI
in	*sImgRef*	Image reference

Returns

Image pointer (surface/texture) or NULL if error

9.37.3.29 void gslc_DrvPageFlipNow (gslc_tsGui ∗ *pGui*)

Force a page flip to occur.

This generally copies active screen surface to the display.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.37.3.30 bool gslc_DrvRotate (gslc_tsGui ∗ *pGui,* uint8_t *nRotation*)

Change rotation, automatically adapt touchscreen axes swap/flip.

Parameters

in	*pGui*	Pointer to GUI
in	*nRotation*	Screen Rotation value (0, 1, 2 or 3)

Returns

true if successful

9.37.3.31 bool gslc_DrvSetBkgndColor (gslc_tsGui ∗ *pGui,* gslc_tsColor *nCol*)

Configure the background to use a solid color.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*nCol*	RGB Color to use

Returns

true if success, false if fail

9.37.3.32 bool gslc_DrvSetBkgndImage (gslc_tsGui ∗ *pGui,* gslc_tsImgRef *sImgRef*)

Configure the background to use a bitmap image.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*sImgRef*	Image reference

Returns

true if success, false if fail

9.37.3.33 bool gslc_DrvSetClipRect (gslc_tsGui ∗ *pGui,* gslc_tsRect ∗ *pRect*)

Set the clipping rectangle for future drawing updates.

Parameters

in	*pGui*	Pointer to GUI
in	*pRect*	Rectangular region to constrain edits

Returns

true if success, false if error

9.37.3.34 bool gslc_DrvSetElemImageGlow (gslc_tsGui ∗ *pGui,* gslc_tsElem ∗ *pElem,* gslc_tsImgRef *sImgRef*)

Set an element's glow-state image.

Parameters

in	*pGui*	Pointer to GUI
in	*pElem*	Pointer to Element to update
in	*sImgRef*	Image reference

Returns

true if success, false if error

9.37.3.35 bool gslc_DrvSetElemImageNorm (gslc_tsGui ∗ *pGui,* gslc_tsElem ∗ *pElem,* gslc_tsImgRef *sImgRef*)

Set an element's normal-state image.

Parameters

in	*pGui*	Pointer to GUI
in	*pElem*	Pointer to Element to update
in	*sImgRef*	Image reference

Returns

true if success, false if error

9.37.4 Variable Documentation

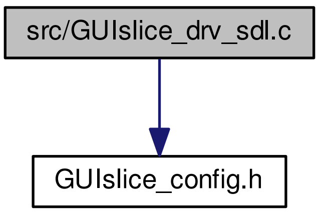
9.37.4.1 const char **GSLC_PMEM** ERRSTR_NULL[ ]

9.37.4.2 const char **GSLC_PMEM** ERRSTR_PXD_NULL[ ]

9.38 src/GUIslice_drv_sdl.c File Reference

```
#include "GUIslice_config.h"
```

Include dependency graph for GUIslice_drv_sdl.c:

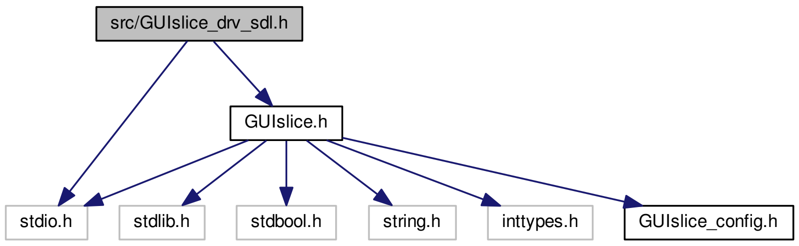


9.39 src/GUIslice_drv_sdl.h File Reference

GUIslice library (driver layer for LINUX / SDL)

```
#include "GUIslice.h"
#include <stdio.h>
```

Include dependency graph for GUIslice_drv_sdl.h:



Data Structures

- struct gslc_tsDriver

Macros

- #define DRV_HAS_DRAW_POINT

  Support gslc_DrvDrawPoint()
- #define DRV_OVERRIDE_TXT_ALIGN

  Driver provides text alignment.

Functions

- bool gslc_DrvInit (gslc_tsGui ∗pGui)

  Initialize the SDL library.
- void gslc_DrvDestruct (gslc_tsGui ∗pGui)

  Free up any members associated with the driver.
- const char ∗ gslc_DrvGetNameDisp (gslc_tsGui ∗pGui)

  Get the display driver name.
- const char ∗ gslc_DrvGetNameTouch (gslc_tsGui ∗pGui)

  Get the touch driver name.
- void ∗ gslc_DrvLoadImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Load a bitmap (∗.bmp) and create a new image resource.
- bool gslc_DrvSetBkgndImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

  Configure the background to use a bitmap image.
- bool gslc_DrvSetBkgndColor (gslc_tsGui ∗pGui, gslc_tsColor nCol)

  Configure the background to use a solid color.
- bool gslc_DrvSetElemImageNorm (gslc_tsGui ∗pGui, gslc_tsElem ∗pElem, gslc_tsImgRef sImgRef)

  Set an element's normal-state image.
- bool gslc_DrvSetElemImageGlow (gslc_tsGui ∗pGui, gslc_tsElem ∗pElem, gslc_tsImgRef sImgRef)

  Set an element's glow-state image.
- void gslc_DrvImageDestruct (void ∗pvImg)

  Release an image surface.
- bool gslc_DrvSetClipRect (gslc_tsGui ∗pGui, gslc_tsRect ∗pRect)

  Set the clipping rectangle for future drawing updates.
- const void ∗ gslc_DrvFontAdd (gslc_teFontRefType eFontRefType, const void ∗pvFontRef, uint16_t nFontSz)

  Load a font from a resource and return pointer to it.
- void gslc_DrvFontsDestruct (gslc_tsGui ∗pGui)

  Release all fonts defined in the GUI.
- bool gslc_DrvGetTxtSize (gslc_tsGui ∗pGui, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxt←Flags, int16_t ∗pnTxtX, int16_t ∗pnTxtY, uint16_t ∗pnTxtSzW, uint16_t ∗pnTxtSzH)

  Get the extent (width and height) of a text string.
- bool gslc_DrvDrawTxt (gslc_tsGui ∗pGui, int16_t nTxtX, int16_t nTxtY, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxtFlags, gslc_tsColor colTxt, gslc_tsColor colBg)

  Draw a text string at the given coordinate.
- void gslc_DrvPageFlipNow (gslc_tsGui ∗pGui)

  Force a page flip to occur.
- bool gslc_DrvDrawPoint (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, gslc_tsColor nCol)

  Draw a point.
- bool gslc_DrvDrawPoints (gslc_tsGui ∗pGui, gslc_tsPt ∗asPt, uint16_t nNumPt, gslc_tsColor nCol)

Draw a point.
- bool gslc_DrvDrawFrameRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a framed rectangle.
- bool gslc_DrvDrawFillRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

  Draw a filled rectangle.
- bool gslc_DrvDrawLine (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)

  Draw a line.
- bool gslc_DrvDrawImage (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, gslc_tsImgRef sImgRef)

  Copy all of source image to destination screen at specified coordinate.
- void gslc_DrvDrawBkgnd (gslc_tsGui ∗pGui)

  Copy the background image to destination screen.
- bool gslc_DrvGetTouch (gslc_tsGui ∗pGui, int16_t ∗pnX, int16_t ∗pnY, uint16_t ∗pnPress, gslc_teInputRaw←Event ∗peInputEvent, int16_t ∗pnInputVal)

  Get the last touch event from the SDL_Event handler.
- bool gslc_DrvRotate (gslc_tsGui ∗pGui, uint8_t nRotation)

  Change rotation, automatically adapt touchscreen axes swap/flip.
- bool gslc_DrvCleanStart (const char ∗sTTY)

  Ensure SDL initializes cleanly to workaround possible issues if previous SDL application failed to close down gracefully.
- void gslc_DrvReportInfoPre ()

  Report driver debug info (before initialization)
- void gslc_DrvReportInfoPost ()

  Report driver debug info (after initialization)
- SDL_Rect gslc_DrvAdaptRect (gslc_tsRect rRect)

  Translate a gslc_tsRect into an SDL_Rect.
- SDL_Color gslc_DrvAdaptColor (gslc_tsColor sCol)

  Translate a gslc_tsColor into an SDL_Color.
- bool gslc_DrvInitTouch (gslc_tsGui ∗pGui, const char ∗acDev)

  Perform any touchscreen-specific initialization.

9.39.1 Detailed Description

GUIslice library (driver layer for LINUX / SDL)

9.39.2 Macro Definition Documentation

9.39.2.1 #define DRV_HAS_DRAW_POINT

Support gslc_DrvDrawPoint()

9.39.2.2 #define DRV_OVERRIDE_TXT_ALIGN

Driver provides text alignment.

9.39.3 Function Documentation

9.39.3.1 SDL_Color gslc_DrvAdaptColor (gslc_tsColor *sCol*)

Translate a gslc_tsColor into an SDL_Color.

Parameters

in	*sCol*	gslc_tsColor

Returns

Converted SDL_Color

9.39.3.2 SDL_Rect gslc_DrvAdaptRect (gslc_tsRect *rRect*)

Translate a gslc_tsRect into an SDL_Rect.

Parameters

in	*rRect*	gslc_tsRect

Returns

Converted SDL_Rect

9.39.3.3 bool gslc_DrvCleanStart (const char ∗ *sTTY*)

Ensure SDL initializes cleanly to workaround possible issues if previous SDL application failed to close down gracefully.

Parameters

in	*sTTY*	Terminal device (eg. "/dev/tty0")

Returns

true if success

9.39.3.4 void gslc_DrvDestruct (gslc_tsGui ∗ *pGui*)

Free up any members associated with the driver.

- Eg. renderers, windows, background surfaces, etc.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.39.3.5 void gslc_DrvDrawBkgnd (gslc_tsGui ∗ *pGui*)

Copy the background image to destination screen.

Parameters

in	*pGui*	Pointer to GUI

Returns

true if success, false if fail

9.39.3.6 bool gslc_DrvDrawFillRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* gslc_tsColor *nCol*)

Draw a filled rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.39.3.7 bool gslc_DrvDrawFrameRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* gslc_tsColor *nCol*)

Draw a framed rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.39.3.8 bool gslc_DrvDrawImage (gslc_tsGui ∗ *pGui,* int16_t *nDstX,* int16_t *nDstY,* gslc_tsImgRef *sImgRef*)

Copy all of source image to destination screen at specified coordinate.

Parameters

in	*pGui*	Pointer to GUI
in	*nDstX*	Destination X coord for copy
in	*nDstY*	Destination Y coord for copy
in	*sImgRef*	Image reference

Returns

true if success, false if fail

9.39.3.9 bool gslc_DrvDrawLine (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* gslc_tsColor *nCol*)

Draw a line.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	Line start (X coordinate)
in	*nY0*	Line start (Y coordinate)
in	*nX1*	Line finish (X coordinate)
in	*nY1*	Line finish (Y coordinate)
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.39.3.10 bool gslc_DrvDrawPoint (gslc_tsGui ∗ *pGui,* int16_t *nX,* int16_t *nY,* gslc_tsColor *nCol*)

Draw a point.

Parameters

in	*pGui*	Pointer to GUI
in	*nX*	X coordinate of point
in	*nY*	Y coordinate of point
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.39.3.11 bool gslc_DrvDrawPoints (gslc_tsGui ∗ *pGui,* gslc_tsPt ∗ *asPt,* uint16_t *nNumPt,* gslc_tsColor *nCol*)

Draw a point.

Parameters

in	*pGui*	Pointer to GUI
in	*asPt*	Array of points to draw
in	*n*↩ *NumPt*	Number of points in array
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.39.3.12 bool gslc_DrvDrawTxt (gslc_tsGui ∗ *pGui,* int16_t *nTxtX,* int16_t *nTxtY,* gslc_tsFont ∗ *pFont,* const char ∗ *pStr,* gslc_teTxtFlags *eTxtFlags,* gslc_tsColor *colTxt,* gslc_tsColor *colBg*)

Draw a text string at the given coordinate.

Parameters

in	*pGui*	Pointer to GUI
in	*nTxtX*	X coordinate of top-left text string
in	*nTxtY*	Y coordinate of top-left text string
in	*pFont*	Ptr to Font
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
in	*colTxt*	Color to draw text
in	*colBg*	unused in SDL, defaults to black

Returns

true if success, false if failure

9.39.3.13 const void∗ gslc_DrvFontAdd (gslc_teFontRefType *eFontRefType,* const void ∗ *pvFontRef,* uint16_t *nFontSz*)

Load a font from a resource and return pointer to it.

Parameters

in	*eFontRefType*	Font reference type (GSLC_FONTREF_FNAME for SDL)
in	*pvFontRef*	Font reference pointer (Pointer to the font filename)
in	*nFontSz*	Typeface size to use

Returns

Void ptr to driver-specific font if load was successful, NULL otherwise

9.39.3.14 void gslc_DrvFontsDestruct (gslc_tsGui ∗ *pGui*)

Release all fonts defined in the GUI.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.39.3.15 const char∗ gslc_DrvGetNameDisp (gslc_tsGui ∗ *pGui*)

Get the display driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

9.39.3.16 const char∗ gslc_DrvGetNameTouch (gslc_tsGui ∗ *pGui*)

Get the touch driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

9.39.3.17 bool gslc_DrvGetTouch (gslc_tsGui ∗ *pGui,* int16_t ∗ *pnX,* int16_t ∗ *pnY,* uint16_t ∗ *pnPress,* gslc_teInputRawEvent ∗ *peInputEvent,* int16_t ∗ *pnInputVal*)

Get the last touch event from the SDL_Event handler.

Parameters

in	*pGui*	Pointer to GUI
out	*pnX*	Ptr to X coordinate of last touch event
out	*pnY*	Ptr to Y coordinate of last touch event
out	*pnPress*	Ptr to Pressure level of last touch event (0 for none, 1 for touch)
out	*peInputEvent*	Indication of event type
out	*pnInputVal*	Additional data for event type

Returns

true if an event was detected or false otherwise

9.39.3.18 bool gslc_DrvGetTxtSize (gslc_tsGui ∗ *pGui,* gslc_tsFont ∗ *pFont,* const char ∗ *pStr,* gslc_teTxtFlags *eTxtFlags,* int16_t ∗ *pnTxtX,* int16_t ∗ *pnTxtY,* uint16_t ∗ *pnTxtSzW,* uint16_t ∗ *pnTxtSzH*)

Get the extent (width and height) of a text string.

Parameters

in	*pGui*	Pointer to GUI
in	*pFont*	Ptr to Font structure
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
out	*pnTxtX*	Ptr to offset X of text
out	*pnTxtY*	Ptr to offset Y of text
out	*pnTxtSzW*	Ptr to width of text
out	*pnTxtSzH*	Ptr to height of text

Returns

true if success, false if failure

9.39.3.19 void gslc_DrvImageDestruct (void ∗ *pvImg*)

Release an image surface.

Parameters

in	*pvImg*	Void ptr to image

Returns

none

9.39.3.20 bool gslc_DrvInit (gslc_tsGui ∗ *pGui*)

Initialize the SDL library.

- Performs clean startup workaround (if enabled)
- Configures video mode
- Initializes font support

PRE:

- The environment variables should be configured before calling gslc_DrvInit().

Parameters

in	*pGui*	Pointer to GUI

Returns

true if success, false if fail

9.39.3.21 bool gslc_DrvInitTouch (gslc_tsGui ∗ *pGui,* const char ∗ *acDev*)

Perform any touchscreen-specific initialization.

Parameters

in	*pGui*	Pointer to GUI
in	*acDev*	Device path to touchscreen eg. "/dev/input/touchscreen"

Returns

true if successful

9.39.3.22 void∗ gslc_DrvLoadImage (gslc_tsGui ∗ *pGui,* gslc_tsImgRef *sImgRef*)

Load a bitmap (∗.bmp) and create a new image resource.

Transparency is enabled by GSLC_BMP_TRANS_EN through use of color (GSLC_BMP_TRANS_RGB).

Parameters

in	*pGui*	Pointer to GUI
in	*sImgRef*	Image reference

Returns

Image pointer (surface/texture/path) or NULL if error

9.39.3.23 void gslc_DrvPageFlipNow (gslc_tsGui ∗ *pGui*)

Force a page flip to occur.

This generally copies active screen surface to the display.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.39.3.24 void gslc_DrvReportInfoPost ( )

Report driver debug info (after initialization)

Returns

none

9.39.3.25 void gslc_DrvReportInfoPre ( )

Report driver debug info (before initialization)

Returns

none

9.39.3.26 bool gslc_DrvRotate (gslc_tsGui ∗ *pGui,* uint8_t *nRotation*)

Change rotation, automatically adapt touchscreen axes swap/flip.

Parameters

in	*pGui*	Pointer to GUI
in	*nRotation*	Screen Rotation value (0, 1, 2 or 3)

Returns

true if successful

9.39.3.27 bool gslc_DrvSetBkgndColor (gslc_tsGui ∗ *pGui,* gslc_tsColor *nCol*)

Configure the background to use a solid color.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*nCol*	RGB Color to use

Returns

true if success, false if fail

9.39.3.28 bool gslc_DrvSetBkgndImage (gslc_tsGui ∗ *pGui,* gslc_tsImgRef *sImgRef*)

Configure the background to use a bitmap image.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*sImgRef*	Image reference

Returns

true if success, false if fail

9.39.3.29 bool gslc_DrvSetClipRect (gslc_tsGui ∗ *pGui,* gslc_tsRect ∗ *pRect*)

Set the clipping rectangle for future drawing updates.

Parameters

in	*pGui*	Pointer to GUI
in	*pRect*	Rectangular region to constrain edits

Returns

true if success, false if error

9.39.3.30 bool gslc_DrvSetElemImageGlow (gslc_tsGui ∗ *pGui*, gslc_tsElem ∗ *pElem*, gslc_tsImgRef *sImgRef*)

Set an element's glow-state image.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElem*	Pointer to Element to update
`in`	*sImgRef*	Image reference

Returns

true if success, false if error

9.39.3.31 bool gslc_DrvSetElemImageNorm (gslc_tsGui ∗ *pGui*, gslc_tsElem ∗ *pElem*, gslc_tsImgRef *sImgRef*)

Set an element's normal-state image.

Parameters

`in`	*pGui*	Pointer to GUI
`in`	*pElem*	Pointer to Element to update
`in`	*sImgRef*	Image reference

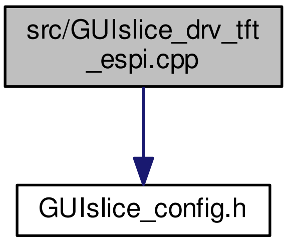
Returns

true if success, false if error

9.40 src/GUIslice_drv_tft_espi.cpp File Reference

```
#include "GUIslice_config.h"
```

Include dependency graph for GUIslice_drv_tft_espi.cpp:

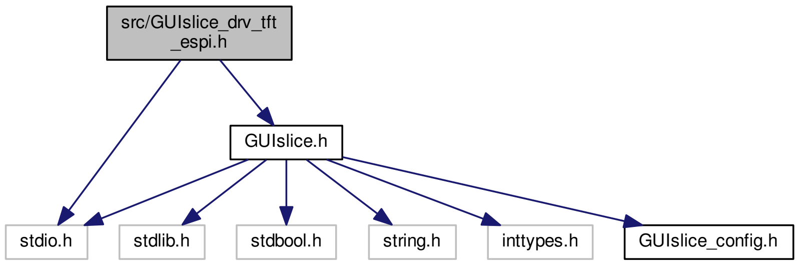


9.41 src/GUIslice_drv_tft_espi.h File Reference

GUIslice library (driver layer for TFT-eSPI)

```
#include "GUIslice.h"
#include <stdio.h>
```

Include dependency graph for GUIslice_drv_tft_espi.h:



Data Structures

- struct gslc_tsDriver

Macros

- #define DRV_HAS_DRAW_POINT

    Support gslc_DrvDrawPoint()
- #define DRV_HAS_DRAW_POINTS

    Support gslc_DrvDrawPoints()
- #define DRV_HAS_DRAW_LINE

    Support gslc_DrvDrawLine()

- #define DRV_HAS_DRAW_RECT_FRAME

    Support gslc_DrvDrawFrameRect()
- #define DRV_HAS_DRAW_RECT_FILL

    Support gslc_DrvDrawFillRect()
- #define DRV_HAS_DRAW_RECT_ROUND_FRAME

    Support gslc_DrvDrawFrameRoundRect()
- #define DRV_HAS_DRAW_RECT_ROUND_FILL

    Support gslc_DrvDrawFillRoundRect()
- #define DRV_HAS_DRAW_CIRCLE_FRAME

    Support gslc_DrvDrawFrameCircle()
- #define DRV_HAS_DRAW_CIRCLE_FILL

    Support gslc_DrvDrawFillCircle()
- #define DRV_HAS_DRAW_TRI_FRAME

    Support gslc_DrvDrawFrameTriangle()
- #define DRV_HAS_DRAW_TRI_FILL

    Support gslc_DrvDrawFillTriangle()
- #define DRV_HAS_DRAW_TEXT

    Support gslc_DrvDrawTxt()
- #define DRV_OVERRIDE_TXT_ALIGN

    Driver provides text alignment.

Functions

- bool gslc_DrvInit (gslc_tsGui ∗pGui)

    Initialize the SDL library.
- bool gslc_DrvInitTs (gslc_tsGui ∗pGui, const char ∗acDev)

    Perform any touchscreen-specific initialization.
- void gslc_DrvDestruct (gslc_tsGui ∗pGui)

    Free up any members associated with the driver.
- const char ∗ gslc_DrvGetNameDisp (gslc_tsGui ∗pGui)

    Get the display driver name.
- const char ∗ gslc_DrvGetNameTouch (gslc_tsGui ∗pGui)

    Get the touch driver name.
- void ∗ gslc_DrvLoadImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

    Load a bitmap (∗.bmp) and create a new image resource.
- bool gslc_DrvSetBkgndImage (gslc_tsGui ∗pGui, gslc_tsImgRef sImgRef)

    Configure the background to use a bitmap image.
- bool gslc_DrvSetBkgndColor (gslc_tsGui ∗pGui, gslc_tsColor nCol)

    Configure the background to use a solid color.
- bool gslc_DrvSetElemImageNorm (gslc_tsGui ∗pGui, gslc_tsElem ∗pElem, gslc_tsImgRef sImgRef)

    Set an element's normal-state image.
- bool gslc_DrvSetElemImageGlow (gslc_tsGui ∗pGui, gslc_tsElem ∗pElem, gslc_tsImgRef sImgRef)

    Set an element's glow-state image.
- void gslc_DrvImageDestruct (void ∗pvImg)

    Release an image surface.
- bool gslc_DrvSetClipRect (gslc_tsGui ∗pGui, gslc_tsRect ∗pRect)

    Set the clipping rectangle for future drawing updates.
- const void ∗ gslc_DrvFontAdd (gslc_teFontRefType eFontRefType, const void ∗pvFontRef, uint16_t nFontSz)

    Load a font from a resource and return pointer to it.
- void gslc_DrvFontsDestruct (gslc_tsGui ∗pGui)

Release all fonts defined in the GUI.
- bool gslc_DrvGetTxtSize (gslc_tsGui ∗pGui, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxt←Flags, int16_t ∗pnTxtX, int16_t ∗pnTxtY, uint16_t ∗pnTxtSzW, uint16_t ∗pnTxtSzH)

    Get the extent (width and height) of a text string.
- bool gslc_DrvDrawTxt (gslc_tsGui ∗pGui, int16_t nTxtX, int16_t nTxtY, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxtFlags, gslc_tsColor colTxt, gslc_tsColor colBg)

    Draw a text string at the given coordinate.
- bool gslc_DrvDrawTxtAlign (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int8_t e←TxtAlign, gslc_tsFont ∗pFont, const char ∗pStr, gslc_teTxtFlags eTxtFlags, gslc_tsColor colTxt, gslc_tsColor colBg)

    Draw a text string in a bounding box using the specified alignment.
- void gslc_DrvPageFlipNow (gslc_tsGui ∗pGui)

    Force a page flip to occur.
- bool gslc_DrvDrawPoint (gslc_tsGui ∗pGui, int16_t nX, int16_t nY, gslc_tsColor nCol)

    Draw a point.
- bool gslc_DrvDrawPoints (gslc_tsGui ∗pGui, gslc_tsPt ∗asPt, uint16_t nNumPt, gslc_tsColor nCol)

    Draw a point.
- bool gslc_DrvDrawFrameRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

    Draw a framed rectangle.
- bool gslc_DrvDrawFillRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, gslc_tsColor nCol)

    Draw a filled rectangle.
- bool gslc_DrvDrawFrameRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor n←Col)

    Draw a framed rounded rectangle.
- bool gslc_DrvDrawFillRoundRect (gslc_tsGui ∗pGui, gslc_tsRect rRect, int16_t nRadius, gslc_tsColor nCol)

    Draw a filled rounded rectangle.
- bool gslc_DrvDrawLine (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)

    Draw a line.
- bool gslc_DrvDrawFrameCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_ts←Color nCol)

    Draw a framed circle.
- bool gslc_DrvDrawFillCircle (gslc_tsGui ∗pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)

    Draw a filled circle.
- bool gslc_DrvDrawFrameTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

    Draw a framed triangle.
- bool gslc_DrvDrawFillTriangle (gslc_tsGui ∗pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, int16_t nX2, int16_t nY2, gslc_tsColor nCol)

    Draw a filled triangle.
- bool gslc_DrvDrawImage (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, gslc_tsImgRef sImgRef)

    Copy all of source image to destination screen at specified coordinate.
- void gslc_DrvDrawMonoFromMem (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, const unsigned char ∗p←Bitmap, bool bProgMem)

    Draw a monochrome bitmap from a memory array.
- void gslc_DrvDrawBmp24FromMem (gslc_tsGui ∗pGui, int16_t nDstX, int16_t nDstY, const unsigned char ∗pBitmap, bool bProgMem)

    Draw a color 24-bit depth bitmap from a memory array.
- void gslc_DrvDrawBkgnd (gslc_tsGui ∗pGui)

    Copy the background image to destination screen.
- bool gslc_DrvRotate (gslc_tsGui ∗pGui, uint8_t nRotation)

    Change rotation, automatically adapt touchscreen axes swap/flip.
- uint16_t gslc_DrvAdaptColorToRaw (gslc_tsColor nCol)

9.41.1 Detailed Description

GUIslice library (driver layer for TFT-eSPI)

9.41.2 Macro Definition Documentation

9.41.2.1 #define DRV_HAS_DRAW_CIRCLE_FILL

Support gslc_DrvDrawFillCircle()

9.41.2.2 #define DRV_HAS_DRAW_CIRCLE_FRAME

Support gslc_DrvDrawFrameCircle()

9.41.2.3 #define DRV_HAS_DRAW_LINE

Support gslc_DrvDrawLine()

9.41.2.4 #define DRV_HAS_DRAW_POINT

Support gslc_DrvDrawPoint()

9.41.2.5 #define DRV_HAS_DRAW_POINTS

Support gslc_DrvDrawPoints()

9.41.2.6 #define DRV_HAS_DRAW_RECT_FILL

Support gslc_DrvDrawFillRect()

9.41.2.7 #define DRV_HAS_DRAW_RECT_FRAME

Support gslc_DrvDrawFrameRect()

9.41.2.8 #define DRV_HAS_DRAW_RECT_ROUND_FILL

Support gslc_DrvDrawFillRoundRect()

9.41.2.9 #define DRV_HAS_DRAW_RECT_ROUND_FRAME

Support gslc_DrvDrawFrameRoundRect()

9.41.2.10 #define DRV_HAS_DRAW_TEXT

Support gslc_DrvDrawTxt()

9.41.2.11 #define DRV_HAS_DRAW_TRI_FILL

Support gslc_DrvDrawFillTriangle()

9.41.2.12 #define DRV_HAS_DRAW_TRI_FRAME

Support gslc_DrvDrawFrameTriangle()

9.41.2.13 #define DRV_OVERRIDE_TXT_ALIGN

Driver provides text alignment.

9.41.3 Function Documentation

9.41.3.1 uint16_t gslc_DrvAdaptColorToRaw (gslc_tsColor *nCol*)

9.41.3.2 void gslc_DrvDestruct (gslc_tsGui ∗ *pGui*)

Free up any members associated with the driver.

- Eg. renderers, windows, background surfaces, etc.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.41.3.3 void gslc_DrvDrawBkgnd (gslc_tsGui ∗ *pGui*)

Copy the background image to destination screen.

Parameters

in	*pGui*	Pointer to GUI

Returns

true if success, false if fail

9.41.3.4 void gslc_DrvDrawBmp24FromMem (gslc_tsGui ∗ *pGui*, int16_t *nDstX*, int16_t *nDstY*, const unsigned char ∗ *pBitmap*, bool *bProgMem*)

Draw a color 24-bit depth bitmap from a memory array.

- Note that users must convert images from their native format (eg. BMP, PNG, etc.) into a C array. Please refer to the following guide for details: https://github.com/ImpulseAdventure/GU↩ Islice/wiki/Display-Images-from-FLASH
- The converted file (c array) can then be included in the sketch.

Parameters

in	*pGui*	Pointer to GUI
in	*nDstX*	X coord for copy
in	*nDstY*	Y coord for copy
in	*pBitmap*	Pointer to bitmap buffer
in	*bProgMem*	Bitmap is stored in Flash if true, RAM otherwise

Returns

none

9.41.3.5 bool gslc_DrvDrawFillCircle (gslc_tsGui ∗ *pGui*, int16_t *nMidX*, int16_t *nMidY*, uint16_t *nRadius*, gslc_tsColor *nCol*)

Draw a filled circle.

Parameters

in	*pGui*	Pointer to GUI
in	*nMidX*	Center of circle (X coordinate)
in	*nMidY*	Center of circle (Y coordinate)
in	*nRadius*	Radius of circle
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.41.3.6 bool gslc_DrvDrawFillRect (gslc_tsGui ∗ *pGui*, gslc_tsRect *rRect*, gslc_tsColor *nCol*)

Draw a filled rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.41.3.7 bool gslc_DrvDrawFillRoundRect (gslc_tsGui ∗ *pGui,* gslc_tsRect *rRect,* int16_t *nRadius,* gslc_tsColor *nCol*)

Draw a filled rounded rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to fill
in	*nRadius*	Radius for rounded corners
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.41.3.8 bool gslc_DrvDrawFillTriangle (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* int16_t *nX2,* int16_t *nY2,* gslc_tsColor *nCol*)

Draw a filled triangle.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X Coordinate #1
in	*nY0*	Y Coordinate #1
in	*nX1*	X Coordinate #2
in	*nY1*	Y Coordinate #2
in	*nX2*	X Coordinate #3
in	*nY2*	Y Coordinate #3
in	*nCol*	Color RGB value to fill

Returns

true if success, false if error

9.41.3.9 bool gslc_DrvDrawFrameCircle (gslc_tsGui ∗ *pGui*, int16_t *nMidX*, int16_t *nMidY*, uint16_t *nRadius*, gslc_tsColor *nCol*)

Draw a framed circle.

Parameters

in	*pGui*	Pointer to GUI
in	*nMidX*	Center of circle (X coordinate)
in	*nMidY*	Center of circle (Y coordinate)
in	*nRadius*	Radius of circle
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.41.3.10 bool gslc_DrvDrawFrameRect (gslc_tsGui ∗ *pGui*, gslc_tsRect *rRect*, gslc_tsColor *nCol*)

Draw a framed rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.41.3.11 bool gslc_DrvDrawFrameRoundRect (gslc_tsGui ∗ *pGui*, gslc_tsRect *rRect*, int16_t *nRadius*, gslc_tsColor *nCol*)

Draw a framed rounded rectangle.

Parameters

in	*pGui*	Pointer to GUI
in	*rRect*	Rectangular region to frame
in	*nRadius*	Radius for rounded corners
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.41.3.12 bool gslc_DrvDrawFrameTriangle (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* int16_t *nX2,* int16_t *nY2,* gslc_tsColor *nCol*)

Draw a framed triangle.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X Coordinate #1
in	*nY0*	Y Coordinate #1
in	*nX1*	X Coordinate #2
in	*nY1*	Y Coordinate #2
in	*nX2*	X Coordinate #3
in	*nY2*	Y Coordinate #3
in	*nCol*	Color RGB value to frame

Returns

true if success, false if error

9.41.3.13 bool gslc_DrvDrawImage (gslc_tsGui ∗ *pGui,* int16_t *nDstX,* int16_t *nDstY,* gslc_tsImgRef *sImgRef*)

Copy all of source image to destination screen at specified coordinate.

Parameters

in	*pGui*	Pointer to GUI
in	*nDstX*	Destination X coord for copy
in	*nDstY*	Destination Y coord for copy
in	*sImgRef*	Image reference

Returns

true if success, false if fail

9.41.3.14 bool gslc_DrvDrawLine (gslc_tsGui ∗ *pGui,* int16_t *nX0,* int16_t *nY0,* int16_t *nX1,* int16_t *nY1,* gslc_tsColor *nCol*)

Draw a line.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	Line start (X coordinate)
in	*nY0*	Line start (Y coordinate)
in	*nX1*	Line finish (X coordinate)
in	*nY1*	Line finish (Y coordinate)
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.41.3.15 void gslc_DrvDrawMonoFromMem (gslc_tsGui ∗ *pGui,* int16_t *nDstX,* int16_t *nDstY,* const unsigned char ∗ *pBitmap,* bool *bProgMem*)

Draw a monochrome bitmap from a memory array.

- Draw from the bitmap buffer using the foreground color defined in the header (unset bits are transparent)

Parameters

in	*pGui*	Pointer to GUI
in	*nDstX*	Destination X coord for copy
in	*nDstY*	Destination Y coord for copy
in	*pBitmap*	Pointer to bitmap buffer
in	*bProgMem*	Bitmap is stored in Flash if true, RAM otherwise

Returns

none

9.41.3.16 bool gslc_DrvDrawPoint (gslc_tsGui ∗ *pGui,* int16_t *nX,* int16_t *nY,* gslc_tsColor *nCol*)

Draw a point.

Parameters

in	*pGui*	Pointer to GUI
in	*nX*	X coordinate of point
in	*nY*	Y coordinate of point
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.41.3.17 bool gslc_DrvDrawPoints (gslc_tsGui ∗ *pGui,* gslc_tsPt ∗ *asPt,* uint16_t *nNumPt,* gslc_tsColor *nCol*)

Draw a point.

Parameters

in	*pGui*	Pointer to GUI

Parameters

in	*asPt*	Array of points to draw
in	*n↩ NumPt*	Number of points in array
in	*nCol*	Color RGB value to draw

Returns

true if success, false if error

9.41.3.18 bool gslc_DrvDrawTxt (gslc_tsGui ∗ *pGui*, int16_t *nTxtX*, int16_t *nTxtY*, gslc_tsFont ∗ *pFont*, const char ∗ *pStr*, gslc_teTxtFlags *eTxtFlags*, gslc_tsColor *colTxt*, gslc_tsColor *colBg*)

Draw a text string at the given coordinate.

Parameters

in	*pGui*	Pointer to GUI
in	*nTxtX*	X coordinate of top-left text string
in	*nTxtY*	Y coordinate of top-left text string
in	*pFont*	Ptr to Font
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
in	*colTxt*	Color to draw text
in	*colBg*	Color of Background for antialias blending

Returns

true if success, false if failure

9.41.3.19 bool gslc_DrvDrawTxtAlign (gslc_tsGui ∗ *pGui*, int16_t *nX0*, int16_t *nY0*, int16_t *nX1*, int16_t *nY1*, int8_t *eTxtAlign*, gslc_tsFont ∗ *pFont*, const char ∗ *pStr*, gslc_teTxtFlags *eTxtFlags*, gslc_tsColor *colTxt*, gslc_tsColor *colBg*)

Draw a text string in a bounding box using the specified alignment.

Parameters

in	*pGui*	Pointer to GUI
in	*nX0*	X coordinate of top-left of bounding box
in	*nY0*	Y coordinate of top-left of bounding box
in	*nX1*	X coordinate of bot-right of bounding box
in	*nY1*	Y coordinate of bot-right of bounding box
in	*eTxtAlign*	Alignment mode]
in	*pFont*	Ptr to Font
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
in	*colTxt*	Color to draw text
in	*colBg*	Color of Background for antialias blending

Returns

true if success, false if failure

9.41.3.20 const void∗ gslc_DrvFontAdd (gslc_teFontRefType *eFontRefType*, const void ∗ *pvFontRef*, uint16_t *nFontSz*)

Load a font from a resource and return pointer to it.

Parameters

in	*eFontRefType*	Font reference type: • GSLC_FONTREF_PTR for Standard TFT_eSPI Fonts • GSLC_FONTREF_FNAME for antialiased Font in SPIFFS
in	*pvFontRef*	Font reference pointer / SPIFFS font filename without ext.
in	*nFontSz*	Typeface size to use, ignored for SPIFFS font

Returns

Void ptr to driver-specific font if load was successful, NULL otherwise

9.41.3.21 void gslc_DrvFontsDestruct (gslc_tsGui ∗ *pGui*)

Release all fonts defined in the GUI.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.41.3.22 const char∗ gslc_DrvGetNameDisp (gslc_tsGui ∗ *pGui*)

Get the display driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

9.41.3.23 const char∗ gslc_DrvGetNameTouch (gslc_tsGui ∗ *pGui*)

Get the touch driver name.

Parameters

in	*pGui*	Pointer to GUI

Returns

String containing driver name

9.41.3.24 bool gslc_DrvGetTxtSize (gslc_tsGui ∗ *pGui,* gslc_tsFont ∗ *pFont,* const char ∗ *pStr,* gslc_teTxtFlags *eTxtFlags,* int16_t ∗ *pnTxtX,* int16_t ∗ *pnTxtY,* uint16_t ∗ *pnTxtSzW,* uint16_t ∗ *pnTxtSzH*)

Get the extent (width and height) of a text string.

Parameters

in	*pGui*	Pointer to GUI
in	*pFont*	Ptr to Font structure
in	*pStr*	String to display
in	*eTxtFlags*	Flags associated with text string
out	*pnTxtX*	Ptr to offset X of text
out	*pnTxtY*	Ptr to offset Y of text
out	*pnTxtSzW*	Ptr to width of text
out	*pnTxtSzH*	Ptr to height of text

Returns

true if success, false if failure

9.41.3.25 void gslc_DrvImageDestruct (void ∗ *pvImg*)

Release an image surface.

Parameters

in	*pvImg*	Void ptr to image

Returns

none

9.41.3.26 bool gslc_DrvInit (gslc_tsGui ∗ *pGui*)

Initialize the SDL library.

- Performs clean startup workaround (if enabled)
- Configures video mode
- Initializes font support

PRE:

- The environment variables should be configured before calling gslc_DrvInit(). This can be done with gslc_↩ DrvInitEnv() or manually in user function.

Parameters

in	*pGui*	Pointer to GUI

Returns

true if success, false if fail

9.41.3.27 bool gslc_DrvInitTs (gslc_tsGui ∗ *pGui*, const char ∗ *acDev*)

Perform any touchscreen-specific initialization.

Parameters

in	*pGui*	Pointer to GUI
in	*acDev*	Device path to touchscreen eg. "/dev/input/touchscreen"

Returns

true if successful

9.41.3.28 void∗ gslc_DrvLoadImage (gslc_tsGui ∗ *pGui*, gslc_tsImgRef *sImgRef*)

Load a bitmap (∗.bmp) and create a new image resource.

Transparency is enabled by GSLC_BMP_TRANS_EN through use of color (GSLC_BMP_TRANS_RGB).

Parameters

in	*pGui*	Pointer to GUI
in	*sImgRef*	Image reference

Returns

Image pointer (surface/texture) or NULL if error

9.41.3.29 void gslc_DrvPageFlipNow (gslc_tsGui ∗ *pGui*)

Force a page flip to occur.

This generally copies active screen surface to the display.

Parameters

in	*pGui*	Pointer to GUI

Returns

none

9.41.3.30 bool gslc_DrvRotate (gslc_tsGui ∗ *pGui,* uint8_t *nRotation*)

Change rotation, automatically adapt touchscreen axes swap/flip.

Parameters

in	*pGui*	Pointer to GUI
in	*nRotation*	Screen Rotation value (0, 1, 2 or 3)

Returns

true if successful

9.41.3.31 bool gslc_DrvSetBkgndColor (gslc_tsGui ∗ *pGui,* gslc_tsColor *nCol*)

Configure the background to use a solid color.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*nCol*	RGB Color to use

Returns

true if success, false if fail

9.41.3.32 bool gslc_DrvSetBkgndImage (gslc_tsGui ∗ *pGui,* gslc_tsImgRef *sImgRef*)

Configure the background to use a bitmap image.

- The background is used when redrawing the entire page

Parameters

in	*pGui*	Pointer to GUI
in	*sImgRef*	Image reference

Returns

true if success, false if fail

9.41.3.33 bool gslc_DrvSetClipRect (gslc_tsGui ∗ *pGui,* gslc_tsRect ∗ *pRect*)

Set the clipping rectangle for future drawing updates.

Parameters

in	*pGui*	Pointer to GUI
in	*pRect*	Rectangular region to constrain edits

Returns

true if success, false if error

9.41.3.34 bool gslc_DrvSetElemImageGlow (gslc_tsGui ∗ *pGui,* gslc_tsElem ∗ *pElem,* gslc_tsImgRef *sImgRef*)

Set an element's glow-state image.

Parameters

in	*pGui*	Pointer to GUI
in	*pElem*	Pointer to Element to update
in	*sImgRef*	Image reference

Returns

true if success, false if error

9.41.3.35 bool gslc_DrvSetElemImageNorm (gslc_tsGui ∗ *pGui,* gslc_tsElem ∗ *pElem,* gslc_tsImgRef *sImgRef*)

Set an element's normal-state image.

Parameters

in	*pGui*	Pointer to GUI
in	*pElem*	Pointer to Element to update
in	*sImgRef*	Image reference

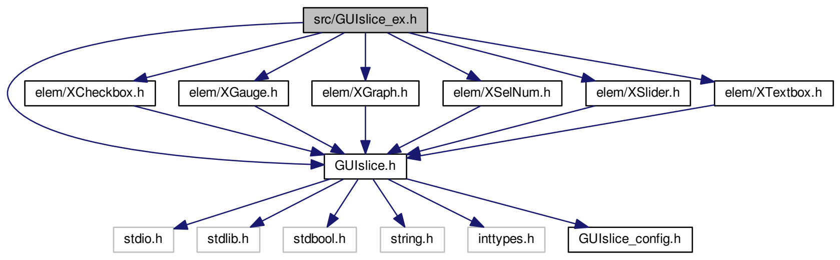
Returns

true if success, false if error

9.42 src/GUIslice_ex.h File Reference

```
#include "GUIslice.h"
#include "elem/XCheckbox.h"
#include "elem/XGauge.h"
#include "elem/XGraph.h"
#include "elem/XSelNum.h"
#include "elem/XSlider.h"
#include "elem/XTextbox.h"
```

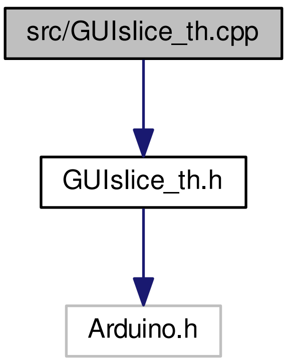
Include dependency graph for GUIslice_ex.h:



9.43 src/GUIslice_th.cpp File Reference

```
#include "GUIslice_th.h"
```

Include dependency graph for GUIslice_th.cpp:



Functions

- void gslc_InitTouchHandler (TouchHandler ∗pTH)
- TouchHandler ∗ gslc_getTouchHandler (void)

Variables

- TouchHandler ∗ pTouchHandler

9.43.1 Function Documentation

9.43.1.1 TouchHandler∗ gslc_getTouchHandler (void)

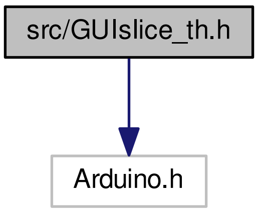
9.43.1.2 void gslc_InitTouchHandler (TouchHandler ∗ *pTH*)

9.43.2 Variable Documentation

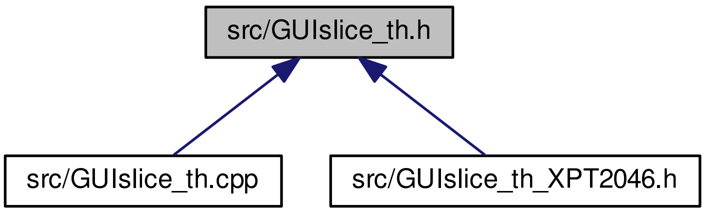
9.43.2.1 TouchHandler∗ pTouchHandler

9.44 src/GUIslice_th.h File Reference

`#include <Arduino.h>`
Include dependency graph for GUIslice_th.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class THPoint
- class TouchHandler

Functions

- void gslc_InitTouchHandler (TouchHandler ∗pTHO)
- TouchHandler ∗ gslc_getTouchHandler (void)

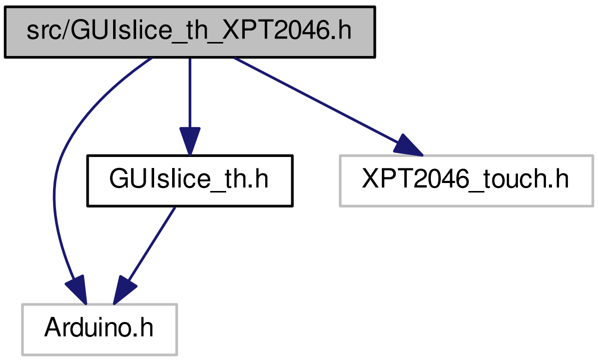
9.44.1 Function Documentation

9.44.1.1 **TouchHandler**∗ gslc_getTouchHandler (void)

9.44.1.2 void gslc_InitTouchHandler (**TouchHandler** ∗ *pTHO*)

9.45 src/GUIslice_th_XPT2046.h File Reference

```
#include <Arduino.h>
#include <GUIslice_th.h>
#include <XPT2046_touch.h>
```
Include dependency graph for GUIslice_th_XPT2046.h:

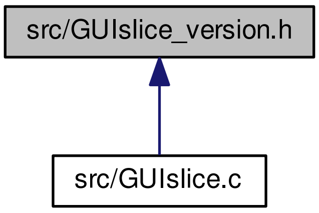


Data Structures

- class TouchHandler_XPT2046

9.46 src/GUIslice_version.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define GUISLICE_VER

9.46.1 Macro Definition Documentation

9.46.1.1 #define GUISLICE_VER