## GUIslice

0.11.2

Generated by Doxygen 1.8.11

## **Contents**

1	GUI	slice lib	rary		1
2	Todo	o List			3
3	Mod	lule Inde	ex		5
	3.1	Module	es		5
4	Hier	archica	l Index		7
	4.1	Class	Hierarchy		7
5	Data	Struct	ure Index		9
	5.1	Data S	Structures		9
6	File	Index			11
	6.1	File Lis	st		11
7	Mod	lule Dod	cumentati	on	13
	7.1	Gener	al Functior	ns	13
		7.1.1	Detailed	Description	14
		7.1.2	Function	Documentation	14
			7.1.2.1	gslc_DebugPrintf(const char *pFmt,)	14
			7.1.2.2	gslc_GetNameDisp(gslc_tsGui *pGui)	14
			7.1.2.3	gslc_GetNameTouch(gslc_tsGui *pGui)	14
			7.1.2.4	gslc_GetVer(gslc_tsGui *pGui)	15
			7.1.2.5	gslc_GuiRotate(gslc_tsGui *pGui, uint8_t nRotation)	15
			7.1.2.6	gslc_Init(gslc_tsGui *pGui, void *pvDriver, gslc_tsPage *asPage, uint8_t nMax↔ Page, gslc_tsFont *asFont, uint8_t nMaxFont)	15

iv CONTENTS

		7.1.2.7	gslc_InitDebug(GSLC_CB_DEBUG_OUT pfunc)	16		
		7.1.2.8	gslc_Quit(gslc_tsGui *pGui)	16		
		7.1.2.9	gslc_SetBkgndColor(gslc_tsGui *pGui, gslc_tsColor nCol)	17		
		7.1.2.10	gslc_SetBkgndImage(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	17		
		7.1.2.11	gslc_SetClipRect(gslc_tsGui *pGui, gslc_tsRect *pRect)	17		
		7.1.2.12	gslc_Update(gslc_tsGui *pGui)	18		
7.2	Graphi	cs Genera	l Functions	19		
	7.2.1	Detailed	Description	19		
	7.2.2	Function	Documentation	20		
		7.2.2.1	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	20		
		7.2.2.2	gslc_ClipPt(gslc_tsRect *pClipRect, int16_t nX, int16_t nY)	20		
		7.2.2.3	gslc_ClipRect(gslc_tsRect *pClipRect, gslc_tsRect *pRect)	20		
		7.2.2.4	gslc_ColorBlend2(gslc_tsColor colStart, gslc_tsColor colEnd, uint16_t nMidAmt, uint16_t nBlendAmt)	21		
		7.2.2.5	gslc_ColorBlend3(gslc_tsColor colStart, gslc_tsColor colMid, gslc_tsColor col← End, uint16_t nMidAmt, uint16_t nBlendAmt)	21		
		7.2.2.6	gslc_ColorEqual(gslc_tsColor a, gslc_tsColor b)	21		
		7.2.2.7	gslc_cosFX(int16_t n64Ang)	22		
		7.2.2.8	gslc_ExpandRect(gslc_tsRect rRect, int16_t nExpandW, int16_t nExpandH)	22		
		7.2.2.9	gslc_GetImageFromFile(const char *pFname, gslc_teImgRefFlags eFmt)	22		
		7.2.2.10	gslc_GetImageFromProg(const unsigned char *pImgBuf, gslc_teImgRefFlags eFmt)	23		
		7.2.2.11	gslc_GetImageFromRam(unsigned char *pImgBuf, gslc_teImgRefFlags eFmt) .	23		
		7.2.2.12	gslc_GetImageFromSD(const char *pFname, gslc_teImgRefFlags eFmt)	23		
		7.2.2.13	gslc_lsInRect(int16_t nSelX, int16_t nSelY, gslc_tsRect rRect)	23		
		7.2.2.14	gslc_lslnWH(int16_t nSelX, int16_t nSelY, uint16_t nWidth, uint16_t nHeight)	24		
		7.2.2.15	gslc_PolarToXY(uint16_t nRad, int16_t n64Ang, int16_t *nDX, int16_t *nDY)	24		
		7.2.2.16	gslc_sinFX(int16_t n64Ang)	25		
7.3	Graphi	cs Primitiv	e Functions	26		
	7.3.1	Detailed	Description	26		
	7.3.2	Function	Function Documentation			

CONTENTS

		7.3.2.1	Radius, gslc_tsColor nCol)	26
		7.3.2.2	gslc_DrawFillQuad(gslc_tsGui *pGui, gslc_tsPt *psPt, gslc_tsColor nCol)	27
		7.3.2.3	gslc_DrawFillRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol)	27
		7.3.2.4	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)	27
		7.3.2.5	gslc_DrawFrameCircle(gslc_tsGui *pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)	28
		7.3.2.6	gslc_DrawFrameQuad(gslc_tsGui *pGui, gslc_tsPt *psPt, gslc_tsColor nCol)	28
		7.3.2.7	gslc_DrawFrameRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol) .	29
		7.3.2.8	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)	29
		7.3.2.9	gslc_DrawLine(gslc_tsGui *pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)	29
		7.3.2.10	gslc_DrawLineH(gslc_tsGui ∗pGui, int16_t nX, int16_t nY, uint16_t nW, gslc_ts⇔ Color nCol)	30
		7.3.2.11	gslc_DrawLinePolar(gslc_tsGui *pGui, int16_t nX, int16_t nY, uint16_t nRadStart, uint16_t nRadEnd, int16_t n64Ang, gslc_tsColor nCol)	30
		7.3.2.12	gslc_DrawLineV(gslc_tsGui *pGui, int16_t nX, int16_t nY, uint16_t nH, gslc_ts⇔ Color nCol)	30
		7.3.2.13	gslc_DrawSetPixel(gslc_tsGui *pGui, int16_t nX, int16_t nY, gslc_tsColor nCol) .	31
7.4	Font F	unctions.		32
	7.4.1	Detailed	Description	32
	7.4.2	Function	Documentation	32
		7.4.2.1	gslc_FontAdd(gslc_tsGui *pGui, int16_t nFontId, gslc_teFontRefType eFontRef↔ Type, const void *pvFontRef, uint16_t nFontSz)	32
		7.4.2.2	gslc_FontGet(gslc_tsGui *pGui, int16_t nFontId)	32
7.5	Page F	unctions		34
	7.5.1	Detailed	Description	34
	7.5.2	Function	Documentation	34
		7.5.2.1	gslc_GetPageCur(gslc_tsGui *pGui)	34
		7.5.2.2	gslc_PageAdd(gslc_tsGui *pGui, int16_t nPageId, gslc_tsElem *psElem, uint16_t nMaxElem, gslc_tsElemRef *psElemRef, uint16_t nMaxElemRef)	35
		7.5.2.3	gslc_PageFindElemByld(gslc_tsGui *pGui, int16_t nPageId, int16_t nElemId)	35
		7.5.2.4	gslc_PageRedrawGet(gslc_tsGui *pGui)	36

vi CONTENTS

		7.5.2.5	gslc_PageRedrawSet(gslc_tsGui *pGui, bool bRedraw)	36
		7.5.2.6	gslc_PopupHide(gslc_tsGui *pGui)	36
		7.5.2.7	gslc_PopupShow(gslc_tsGui *pGui, int16_t nPageId, bool bModal)	36
		7.5.2.8	gslc_SetPageBase(gslc_tsGui *pGui, int16_t nPageId)	37
		7.5.2.9	gslc_SetPageCur(gslc_tsGui *pGui, int16_t nPageId)	37
		7.5.2.10	gslc_SetPageOverlay(gslc_tsGui *pGui, int16_t nPageId)	37
		7.5.2.11	gslc_SetStackPage(gslc_tsGui *pGui, uint8_t nStackPos, int16_t nPageId)	38
		7.5.2.12	gslc_SetStackState(gslc_tsGui *pGui, uint8_t nStackPos, bool bActive, bool b⊷ DoDraw)	38
7.6	Eleme	nt Functior	ns	39
	7.6.1	Detailed	Description	39
7.7	Eleme	nt: Creatio	n Functions	40
	7.7.1	Detailed	Description	40
	7.7.2	Function	Documentation	40
		7.7.2.1	gslc_ElemCreateBox(gslc_tsGui *pGui, int16_t nElemId, int16_t nPage, gslc_← tsRect rElem)	40
		7.7.2.2	gslc_ElemCreateBtnImg(gslc_tsGui *pGui, int16_t nElemId, int16_t nPage, gslc_tsRect rElem, gslc_tsImgRef sImgRef, gslc_tsImgRef sImgRefSel, GSLC← CB_TOUCH cbTouch)	41
		7.7.2.3	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)	41
		7.7.2.4	gslc_ElemCreateImg(gslc_tsGui *pGui, int16_t nElemId, int16_t nPage, gslc_← tsRect rElem, gslc_tsImgRef sImgRef)	42
		7.7.2.5	gslc_ElemCreateLine(gslc_tsGui *pGui, int16_t nElemId, int16_t nPage, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1)	42
		7.7.2.6	gslc_ElemCreateTxt(gslc_tsGui *pGui, int16_t nElemId, int16_t nPage, gslc_ts⇔ Rect rElem, char *pStrBuf, uint8_t nStrBufMax, int16_t nFontId)	43
7.8	Eleme	nt: Genera	al Functions	44
	7.8.1	Detailed	Description	44
	7.8.2	Function	Documentation	44
		7.8.2.1	gslc_ElemGetId(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	44
7.9	Eleme	nt: Update	Functions	45
	7.9.1	Detailed	Description	46

CONTENTS vii

7.9.2	Function	Documentation	46
	7.9.2.1	gslc_ElemGetGlow(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	46
	7.9.2.2	gslc_ElemGetGlowEn(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	46
	7.9.2.3	gslc_ElemGetGroup(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	47
	7.9.2.4	gslc_ElemGetRedraw(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	47
	7.9.2.5	gslc_ElemGetVisible(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	47
	7.9.2.6	gslc_ElemOwnsCoord(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nX, int16_t nY, bool bOnlyClickEn)	48
	7.9.2.7	gslc_ElemSetClickEn(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool b↔ ClickEn)	48
	7.9.2.8	gslc_ElemSetCol(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_tsColor colFrame, gslc_tsColor colFill, gslc_tsColor colFillGlow)	48
	7.9.2.9	gslc_ElemSetDrawFunc(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, GSLC ← _ CB_DRAW funcCb)	49
	7.9.2.10	gslc_ElemSetFillEn(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool bFillEn)	49
	7.9.2.11	gslc_ElemSetFrameEn(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool b← FrameEn)	50
	7.9.2.12	gslc_ElemSetGlow(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool bGlowing)	50
	7.9.2.13	gslc_ElemSetGlowCol(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_ts↔ Color colFrameGlow, gslc_tsColor colFillGlow, gslc_tsColor colTxtGlow)	50
	7.9.2.14	gslc_ElemSetGlowEn(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool b⇔ GlowEn)	51
	7.9.2.15	gslc_ElemSetGroup(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int nGroupId)	51
	7.9.2.16	gslc_ElemSetRedraw(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_te← RedrawType eRedraw)	51
	7.9.2.17	gslc_ElemSetStyleFrom(gslc_tsGui *pGui, gslc_tsElemRef *pElemRefSrc, gslc_tsElemRef *pElemRefDest)	52
	7.9.2.18	gslc_ElemSetTickFunc(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, GSLC_← CB_TICK funcCb)	52
	7.9.2.19	gslc_ElemSetTxtAlign(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, unsigned nAlign)	52
	7.9.2.20	gslc_ElemSetTxtCol(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_tsColor colVal)	53
	7.9.2.21	gslc_ElemSetTxtEnc(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_te← TxtFlags eFlags)	53
	7.9.2.22	gslc_ElemSetTxtMargin(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, unsigned nMargin)	54

viii CONTENTS

		7.9.2.23	gslc_ElemSetTxtMem(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_te← TxtFlags eFlags)	54
		7.9.2.24	gslc_ElemSetTxtStr(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, const char *pStr)	54
		7.9.2.25	gslc_ElemSetVisible(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool bVisible)	55
		7.9.2.26	gslc_ElemUpdateFont(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int nFontId)	55
7.10	Touchs	creen Fun	octions	56
	7.10.1	Detailed	Description	56
	7.10.2	Macro De	efinition Documentation	56
		7.10.2.1	TOUCH_ROTATION_DATA	56
		7.10.2.2	TOUCH_ROTATION_DATA	57
		7.10.2.3	TOUCH_ROTATION_FLIPX	57
		7.10.2.4	TOUCH_ROTATION_FLIPX	57
		7.10.2.5	TOUCH_ROTATION_FLIPY	57
		7.10.2.6	TOUCH_ROTATION_FLIPY	57
		7.10.2.7	TOUCH_ROTATION_SWAPXY	57
		7.10.2.8	TOUCH_ROTATION_SWAPXY	57
	7.10.3	Function	Documentation	57
		7.10.3.1	gslc_GetTouch(gslc_tsGui *pGui, int16_t *pnX, int16_t *pnY, uint16_t *pnPress, gslc_teInputRawEvent *peInputEvent, int16_t *pnInputVal)	57
		7.10.3.2	gslc_InitTouch(gslc_tsGui *pGui, const char *acDev)	57
		7.10.3.3	gslc_SetTouchRemapCal(gslc_tsGui *pGui, uint16_t nXMin, uint16_t nXMax, uint16_t nYMin, uint16_t nYMax)	58
		7.10.3.4	gslc_SetTouchRemapEn(gslc_tsGui *pGui, bool bEn)	58
		7.10.3.5	gslc_SetTouchRemapYX(gslc_tsGui *pGui, bool bSwap)	58
7.11	Input M	lapping Fu	unctions	60
	7.11.1	Detailed	Description	60
	7.11.2	Function	Documentation	60
		7.11.2.1	gslc_InitInputMap(gslc_tsGui *pGui, gslc_tsInputMap *asInputMap, uint8_t n⊷ InputMapMax)	60
		7.11.2.2	gslc_InputMapAdd(gslc_tsGui *pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc_teAction eAction, int16_t nActionVal)	60
		7.11.2.3	gslc_SetPinPollFunc(gslc_tsGui *pGui, GSLC_CB_PIN_POLL pfunc)	60

CONTENTS

7.12	Genera	ll Purpose Macros	61
	7.12.1	Detailed Description	61
	7.12.2	Macro Definition Documentation	61
		7.12.2.1 GSLC_DEBUG_PRINT	61
		7.12.2.2 GSLC_DEBUG_PRINT_CONST	61
7.13	Flash-b	pased Element Macros	62
	7.13.1	Detailed Description	62
	7.13.2	Macro Definition Documentation	62
		7.13.2.1 gslc_ElemCreateBox_P	62
		7.13.2.2 gslc_ElemCreateBtnTxt_P	63
		7.13.2.3 gslc_ElemCreateLine_P	63
		7.13.2.4 gslc_ElemCreateTxt_P	63
		7.13.2.5 gslc_ElemCreateTxt_P_R	64
7.14	Interna	l Functions	65
	7.14.1	Detailed Description	70
	7.14.2	Variable Documentation	70
		7.14.2.1 abPageStackActive	70
		7.14.2.2 abPageStackDoDraw	70
		7.14.2.3 apPageStack	70
		7.14.2.4 asElem	70
		7.14.2.5 asElemRef	70
		7.14.2.6 asFont	70
		7.14.2.7 asInputMap	71
		7.14.2.8 asPage	71
		7.14.2.9 b	71
		7.14.2.10 bRedrawPartialEn	71
		7.14.2.11 bScreenNeedFlip	71
		7.14.2.12 bScreenNeedRedraw	71
		7.14.2.13 bTouchRemapEn	71
		7.14.2.14 bTouchRemapYX	71

CONTENTS

7.14.2.15 colElemFill
7.14.2.16 colElemFillGlow
7.14.2.17 colElemFrame
7.14.2.18 colElemFrameGlow
7.14.2.19 colElemText
7.14.2.20 colElemTextGlow
7.14.2.21 eAction
7.14.2.22 eElemFlags
7.14.2.23 eEvent
7.14.2.24 eFontRefType
7.14.2.25 elmgFlags
7.14.2.26 eInitStatTouch
7.14.2.27 eTouch
7.14.2.28 eTxtAlign
7.14.2.29 eTxtFlags
7.14.2.30 eType
7.14.2.31 g
7.14.2.32 h
7.14.2.33 nActionVal
7.14.2.34 nDisp0H
7.14.2.35 nDisp0W
7.14.2.36 nDispDepth
7.14.2.37 nDispH
7.14.2.38 nDispW
7.14.2.39 nElemAutoldNext
7.14.2.40 nElemCnt
7.14.2.41 nElemIndFocused
7.14.2.42 nElemMax
7.14.2.43 nElemRefCnt
7.14.2.44 nElemRefMax

CONTENTS xi

7.14.2.45 nFeatures
7.14.2.46 nFlipX
7.14.2.47 nFlipY
7.14.2.48 nFontCnt
7.14.2.49 nFontMax
7.14.2.50 nFrameRateCnt
7.14.2.51 nFrameRateStart
7.14.2.52 nGroup
7.14.2.53 nld
7.14.2.54 nld
7.14.2.55 nInputMapCnt
7.14.2.56 nInputMapMax
7.14.2.57 nPageCnt
7.14.2.58 nPageId
7.14.2.59 nPageMax
7.14.2.60 nRotation
7.14.2.61 nSize
7.14.2.62 nStrBufMax
7.14.2.63 nSubType
7.14.2.64 nSwapXY
7.14.2.65 nTouchCalXMax
7.14.2.66 nTouchCalXMin
7.14.2.67 nTouchCalYMax
7.14.2.68 nTouchCalYMin
7.14.2.69 nTouchLastPress
7.14.2.70 nTouchLastX
7.14.2.71 nTouchLastY
7.14.2.72 nTouchRotation
7.14.2.73 nTxtMargin
7.14.2.74 nType

xii CONTENTS

7.14.2.75 nVal
7.14.2.76 nX
7.14.2.77 nY
7.14.2.78 pElem
7.14.2.79 pElemRefParent
7.14.2.80 pElemRefTracked
7.14.2.81 pFname
7.14.2.82 pfuncPinPoll
7.14.2.83 pfuncXDraw
7.14.2.84 pfuncXEvent
7.14.2.85 pfuncXTick
7.14.2.86 pfuncXTouch
7.14.2.87 plmgBuf
7.14.2.88 pStrBuf
7.14.2.89 pTxtFont
7.14.2.90 pvData
7.14.2.91 pvDriver
7.14.2.92 pvFont
7.14.2.93 pvlmgRaw
7.14.2.94 pvScope
7.14.2.95 pXData
7.14.2.96 r
7.14.2.97 rElem
7.14.2.98 sCollect
7.14.2.99 sElemTmpProg
7.14.2.100sImgRefBkgnd
7.14.2.101sImgRefGlow
7.14.2.102sImgRefNorm
7.14.2.103w
7.14.2.104x

CONTENTS xiii

	7.14.2.10	05x	80
	7.14.2.10	06y	80
	7.14.2.10	)7y	80
7.15 Interna	al: Misc Fu	nctions	81
7.15.1	Detailed	Description	81
7.15.2	Function	Documentation	81
	7.15.2.1	gslc_ResetImage()	81
7.16 Interna	al: Element	t Functions	82
7.16.1	Detailed	Description	82
7.16.2	Function	Documentation	82
	7.16.2.1	gslc_ElemAdd(gslc_tsGui *pGui, int16_t nPageId, gslc_tsElem *pElem, gslc_← teElemRefFlags eFlags)	82
	7.16.2.2	$ \begin{split} & gslc\_ElemCreate(gslc\_tsGui*pGui, int16\_t  nElemId, int16\_t  nPageId, int16\_t  n \leftrightarrow \\ & Type,  gslc\_tsRect  rElem,  char*pStrBuf,  uint8\_t  nStrBufMax,  int16\_t  nFontId)  .  . \end{split} $	83
	7.16.2.3	gslc_ElemDraw(gslc_tsGui *pGui, int16_t nPageId, int16_t nElemId)	83
	7.16.2.4	gslc_ElemDrawByRef(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_te⇔ RedrawType eRedraw)	84
	7.16.2.5	gslc_ElemSetImage(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_tsImg↔ Ref sImgRef, gslc_tsImgRef sImgRefSel)	84
	7.16.2.6	gslc_GetElemFromRef(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	84
	7.16.2.7	gslc_GetElemRefFlag(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, uint8_t n↔ FlagMask)	85
	7.16.2.8	gslc_GetXDataFromRef(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_↔ t nType, int16_t nLineNum)	85
	7.16.2.9	gslc_SetElemRefFlag(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, uint8_t n↔ FlagMask, uint8_t nFlagVal)	85
7.17 Interna	al: Page Fu	unctions	87
7.17.1	Detailed	Description	87
7.17.2	Function	Documentation	87
	7.17.2.1	gslc_ElemEvent(void *pvGui, gslc_tsEvent sEvent)	87
	7.17.2.2	gslc_ElemSendEventTouch(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef↔ Tracked, gslc_teTouch eTouch, int16_t nX, int16_t nY)	88
	7.17.2.3	gslc_EventCreate(gslc_tsGui *pGui, gslc_teEventType eType, uint8_t nSubType, void *pvScope, void *pvData)	88

xiv CONTENTS

		7.17.2.4	gslc_PageEvent(void *pvGui, gslc_tsEvent sEvent)	89
		7.17.2.5	gslc_PageFindById(gslc_tsGui *pGui, int16_t nPageId)	89
		7.17.2.6	gslc_PageFlipGet(gslc_tsGui *pGui)	89
		7.17.2.7	gslc_PageFlipGo(gslc_tsGui *pGui)	89
		7.17.2.8	gslc_PageFlipSet(gslc_tsGui *pGui, bool bNeeded)	90
		7.17.2.9	gslc_PageFocusStep(gslc_tsGui *pGui, gslc_tsPage *pPage, bool bNext)	90
		7.17.2.10	gslc_PageRedrawCalc(gslc_tsGui *pGui)	90
		7.17.2.11	gslc_PageRedrawGo(gslc_tsGui *pGui)	91
7.18	Internal	l: Element	Collection Functions	92
	7.18.1	Detailed [	Description	93
	7.18.2	Function	Documentation	93
		7.18.2.1	gslc_CollectElemAdd(gslc_tsGui *pGui, gslc_tsCollect *pCollect, const gslc_ts ← Elem *pElem, gslc_teElemRefFlags eFlags)	93
		7.18.2.2	gslc_CollectFindElemById(gslc_tsGui *pGui, gslc_tsCollect *pCollect, int16_← t nElemId)	93
		7.18.2.3	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	93
		7.18.2.4	gslc_CollectFindFocusStep(gslc_tsGui *pGui, gslc_tsCollect *pCollect, bool b↔ Next, bool *pbWrapped, int16_t *pnElemInd)	94
		7.18.2.5	gslc_CollectGetElemRefTracked(gslc_tsGui *pGui, gslc_tsCollect *pCollect)	94
		7.18.2.6	gslc_CollectGetFocus(gslc_tsGui *pGui, gslc_tsCollect *pCollect)	94
		7.18.2.7	gslc_CollectGetNextId(gslc_tsGui *pGui, gslc_tsCollect *pCollect)	95
		7.18.2.8	gslc_CollectGetRedraw(gslc_tsGui *pGui, gslc_tsCollect *pCollect)	95
		7.18.2.9	$gslc\_CollectReset(gslc\_tsCollect *pCollect, gslc\_tsElem *asElem, uint16\_t n \leftarrow ElemMax, gslc\_tsElemRef *asElemRef, uint16\_t n ElemRefMax)$	95
		7.18.2.10	gslc_CollectSetElemTracked(gslc_tsGui *pGui, gslc_tsCollect *pCollect, gslc_← tsElemRef *pElemRef)	96
		7.18.2.11	gslc_CollectSetFocus(gslc_tsGui *pGui, gslc_tsCollect *pCollect, int16_t n← ElemInd)	96
		7.18.2.12	gslc_CollectSetParent(gslc_tsGui *pGui, gslc_tsCollect *pCollect, gslc_tsElem← Ref *pElemRefParent)	96
7.19	Internal	l: Element	Collection Event Functions	98
	7.19.1	Detailed [	Description	98
	7.19.2	Function	Documentation	98

CONTENTS xv

. 98	2.1 gslc_CollectEvent(void *pvGui, gslc_tsEvent sEvent)	7.19.2.1	
	gslc_CollectInput(gslc_tsGui *pGui, gslc_tsCollect *pCollect, gslc_tsEventTouch *pEventTouch)	7.19.2.2	
	2.3 gslc_CollectTouch(gslc_tsGui *pGui, gslc_tsCollect *pCollect, gslc_tsEvent ← Touch *pEventTouch)	7.19.2.3	
. 100	ring Functions	iternal: Trackin	7.20 I
. 100	ed Description	.20.1 Detailed	7
. 100	ion Documentation	.20.2 Function	7
	2.1 gslc_InputMapLookup(gslc_tsGui *pGui, gslc_teInputRawEvent eInputEvent, int16_t nInputVal, gslc_teAction *peAction, int16_t *pnActionVal)	7.20.2.1	
	2.2 gslc_TrackInput(gslc_tsGui *pGui, gslc_tsPage *pPage, gslc_teInputRawEvent eInputEvent, int16_t nInputVal)	7.20.2.2	
	2.3 gslc_TrackTouch(gslc_tsGui *pGui, gslc_tsPage *pPage, int16_t nX, int16_t nY, uint16_t nPress)	7.20.2.3	
. 102	nup Functions	iternal: Cleanu	7.21 I
. 102	ed Description	.21.1 Detailed	7
. 102	ion Documentation	.21.2 Function	7
. 102	2.1 gslc_CollectDestruct(gslc_tsGui *pGui, gslc_tsCollect *pCollect)	7.21.2.1	
. 103	2.2 gslc_ElemDestruct(gslc_tsElem *pElem)	7.21.2.2	
. 104	2.3 gslc_GuiDestruct(gslc_tsGui *pGui)	7.21.2.3	
. 104	2.4 gslc_PageDestruct(gslc_tsGui *pGui, gslc_tsPage *pPage)	7.21.2.4	
. 104	2.5 gslc_ResetElem(gslc_tsElem *pElem)	7.21.2.5	
. 105	2.6 gslc_ResetFont(gslc_tsFont *pFont)	7.21.2.6	

xvi CONTENTS

8	Data	Structure Documentation	107
	8.1	gslc_tsCollect Struct Reference	107
		8.1.1 Detailed Description	108
	8.2	gslc_tsColor Struct Reference	108
		8.2.1 Detailed Description	108
	8.3	gslc_tsDriver Struct Reference	109
		8.3.1 Field Documentation	109
		8.3.1.1 nColBkgnd	109
		8.3.1.2 rClipRect	109
	8.4	gslc_tsElem Struct Reference	110
		8.4.1 Detailed Description	111
	8.5	gslc_tsElemRef Struct Reference	112
		8.5.1 Detailed Description	112
	8.6	gslc_tsEvent Struct Reference	112
		8.6.1 Detailed Description	113
	8.7	gslc_tsEventTouch Struct Reference	113
		8.7.1 Detailed Description	113
	8.8	gslc_tsFont Struct Reference	113
		8.8.1 Detailed Description	114
	8.9	gslc_tsGui Struct Reference	114
		8.9.1 Detailed Description	116
	8.10	gslc_tsImgRef Struct Reference	116
		8.10.1 Detailed Description	117
	8.11	gslc_tsInputMap Struct Reference	117
		8.11.1 Detailed Description	117
	8.12	gslc_tsPage Struct Reference	118
		8.12.1 Detailed Description	118
	8.13	gslc_tsPt Struct Reference	118
		8.13.1 Detailed Description	119
	8.14	gslc_tsRect Struct Reference	119

CONTENTS xvii

	0444	Detailed Description	110
0.45		Detailed Description	
8.15	_	Checkbox Struct Reference	
		Detailed Description	
	8.15.2	Field Documentation	120
		8.15.2.1 bChecked	120
		8.15.2.2 bRadio	120
		8.15.2.3 colCheck	121
		8.15.2.4 nStyle	121
		8.15.2.5 pfuncXToggle	121
8.16	gslc_ts	Gauge Struct Reference	121
	8.16.1	Detailed Description	122
	8.16.2	Field Documentation	122
		8.16.2.1 bFlip	122
		8.16.2.2 blndicFill	122
		8.16.2.3 bValLastValid	123
		8.16.2.4 bVert	123
		8.16.2.5 colGauge	123
		8.16.2.6 colTick	123
		8.16.2.7 nIndicLen	123
		8.16.2.8 nIndicTip	123
		8.16.2.9 nMax	123
		8.16.2.10 nMin	123
		8.16.2.11 nStyle	
		8.16.2.12 nTickCnt	
		8.16.2.13 nTickLen	
		8.16.2.14 nVal	
		8.16.2.15 nValLast	
8.17	_	Graph Struct Reference	
	8.17.1	Detailed Description	125
	8.17.2	Field Documentation	125

xviii CONTENTS

8.17.2.1 bScrollEn
8.17.2.2 colGraph
8.17.2.3 eStyle
8.17.2.4 nBufCnt
8.17.2.5 nBufMax
8.17.2.6 nMargin
8.17.2.7 nPlotIndMax
8.17.2.8 nPlotIndStart
8.17.2.9 nPlotValMax
8.17.2.10 nPlotValMin
8.17.2.11 nScrollPos
8.17.2.12 nWndHeight
8.17.2.13 nWndWidth
8.17.2.14 pBuf
sXListbox Struct Reference
Detailed Description
2 Field Documentation
8.18.2.1 bltemAutoSizeH
8.18.2.2 bltemAutoSizeW
8.18.2.3 bNeedRecalc
8.18.2.4 colGap
8.18.2.5 nBufltemsMax
8.18.2.6 nBufltemsPos
8.18.2.7 nCols
8.18.2.8 nltemCnt
8.18.2.9 nltemCurSel
8.18.2.10 nltemCurSelLast
8.18.2.11 nltemGap
8.18.2.12 nltemH
8.18.2.13 nltemMarginH

CONTENTS xix

	8.18.2.14 nltemMarginW	30
	8.18.2.15 nltemSavedSel	30
	8.18.2.16 nltemTop	30
	8.18.2.17 nltemW	30
	8.18.2.18 nMarginH	30
	8.18.2.19 nMarginW	30
	8.18.2.20 nRows	30
	8.18.2.21 pBufltems	30
	8.18.2.22 pfuncXSel	30
8.19 gslc_ts	sXSlider Struct Reference	31
8.19.1	Detailed Description	31
8.19.2	Prield Documentation	32
	8.19.2.1 bTrim	32
	8.19.2.2 bVert	32
	8.19.2.3 colTick	32
	8.19.2.4 colTrim	32
	8.19.2.5 nPos	32
	8.19.2.6 nPosMax	32
	8.19.2.7 nPosMin	32
	8.19.2.8 nThumbSz	32
	8.19.2.9 nTickDiv	32
	8.19.2.10 nTickLen	32
	8.19.2.11 pfuncXPos	33
8.20 gslc_ts	sXTextbox Struct Reference	33
8.20.1	Detailed Description	34
8.20.2	Prield Documentation	34
	8.20.2.1 bScrollEn	34
	8.20.2.2 bWrapEn	34
	8.20.2.3 nBufCols	34
	8.20.2.4 nBufPosX	34

CONTENTS

		8.20.2.5 nBufPosY	4
		8.20.2.6 nBufRows	4
		8.20.2.7 nChSizeX	4
		8.20.2.8 nChSizeY	4
		8.20.2.9 nCurPosX	4
		8.20.2.10 nCurPosY	5
		8.20.2.11 nMarginX	5
		8.20.2.12 nMarginY	5
		8.20.2.13 nRedrawRow	5
		8.20.2.14 nScrollPos	5
		8.20.2.15 nWndCols	5
		8.20.2.16 nWndRows	5
		8.20.2.17 nWndRowStart	5
		8.20.2.18 pBuf	5
8.21	THPoin	t Class Reference	6
	8.21.1	Constructor & Destructor Documentation	6
		8.21.1.1 THPoint(void)	6
		8.21.1.2 THPoint(int16_t x, int16_t y, int16_t z)	6
	8.21.2	Member Function Documentation	6
		8.21.2.1 operator"!=(THPoint)	6
		8.21.2.2 operator==(THPoint)	6
	8.21.3	Field Documentation	6
		8.21.3.1 x	6
		8.21.3.2 y	6
		8.21.3.3 z	6
8.22	Touch	landler Class Reference	7
	8.22.1	Constructor & Destructor Documentation	7
		8.22.1.1 TouchHandler()	7
	8.22.2	Member Function Documentation	7
		8.22.2.1 begin(void)	7

CONTENTS xxi

			8.22.2.2	getPoint(void)	137
			8.22.2.3	scale(THPoint pln)	138
			8.22.2.4	setCalibration(uint16_t ts_xMin, uint16_t ts_xMax, uint16_t ts_yMin, uint16_t ts _yMax)	138
			8.22.2.5	setSize(uint16_t_disp_xSize, uint16_t_disp_ySize)	138
			8.22.2.6	setSwapFlip(bool _swapXY, bool _flipX, bool _flipY)	138
	8.23	Touch	landler_XI	PT2046 Class Reference	138
		8.23.1	Construc	tor & Destructor Documentation	139
			8.23.1.1	TouchHandler_XPT2046(SPIClass &spi, uint8_t spi_cs_pin)	139
		8.23.2	Member	Function Documentation	139
			8.23.2.1	begin(void)	139
			8.23.2.2	getPoint(void)	139
		8.23.3	Field Doo	cumentation	139
			8.23.3.1	spi	139
			8.23.3.2	touchDriver	139
9	File	Docume	entation		141
9	<b>File</b> 9.1			Reference	
9		READI	ME.md File	e Reference	141
9	9.1	READI	ME.md File		141 141
9	9.1	READN src/ele	ME.md File	box.c File Reference	141 141 142
9	9.1	READN src/ele	ME.md File m/XCheck Function	box.c File Reference  Documentation  gslc_ElemXCheckboxCreate(gslc_tsGui *pGui, int16_t nElemId, int16_t n← Page, gslc_tsXCheckbox *pXData, gslc_tsRect rElem, bool bRadio, gslc_teX←	141 141 142
9	9.1	READN src/ele	ME.md File m/XCheck Function 9.2.1.1	box.c File Reference  Documentation  gslc_ElemXCheckboxCreate(gslc_tsGui *pGui, int16_t nElemId, int16_t n↔ Page, gslc_tsXCheckbox *pXData, gslc_tsRect rElem, bool bRadio, gslc_teX↔ CheckboxStyle nStyle, gslc_tsColor colCheck, bool bChecked)  gslc_ElemXCheckboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType)	141 141 142 142
9	9.1	READN src/ele	ME.md File m/XCheck Function 9.2.1.1	Documentation  gslc_ElemXCheckboxCreate(gslc_tsGui *pGui, int16_t nElemId, int16_t n↔ Page, gslc_tsXCheckbox *pXData, gslc_tsRect rElem, bool bRadio, gslc_teX↔ CheckboxStyle nStyle, gslc_tsColor colCheck, bool bChecked)  gslc_ElemXCheckboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType eRedraw)	141 141 142 142 143
9	9.1	READN src/ele	ME.md File m/XCheck Function 9.2.1.1 9.2.1.2	Documentation  gslc_ElemXCheckboxCreate(gslc_tsGui *pGui, int16_t nElemId, int16_t n↔ Page, gslc_tsXCheckbox *pXData, gslc_tsRect rElem, bool bRadio, gslc_teX↔ CheckboxStyle nStyle, gslc_tsColor colCheck, bool bChecked)	141 141 142 142 143
9	9.1	READN src/ele	ME.md File m/XCheck Function 9.2.1.1 9.2.1.2 9.2.1.3 9.2.1.4	Documentation  gslc_ElemXCheckboxCreate(gslc_tsGui *pGui, int16_t nElemId, int16_t n← Page, gslc_tsXCheckbox *pXData, gslc_tsRect rElem, bool bRadio, gslc_teX← CheckboxStyle nStyle, gslc_tsColor colCheck, bool bChecked)  gslc_ElemXCheckboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType eRedraw)  gslc_ElemXCheckboxFindChecked(gslc_tsGui *pGui, int16_t nGroupId)  gslc_ElemXCheckboxGetState(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_ElemRef, gslc_ElemRef, gslc_tsElemRef, gslc_tsElemRef, gslc_tsElemRef, gslc_tsElemRef, gslc_tsElemRef, pElemRef, pElemRef, gslc_tsElemRef, pElemRef, pElemRe	141 141 142 142 143 143
9	9.1	READN src/ele	ME.md File m/XCheck Function 9.2.1.1 9.2.1.2 9.2.1.3 9.2.1.4 9.2.1.5	Documentation  gslc_ElemXCheckboxCreate(gslc_tsGui *pGui, int16_t nElemId, int16_t n↔ Page, gslc_tsXCheckbox *pXData, gslc_tsRect rElem, bool bRadio, gslc_teX↔ CheckboxStyle nStyle, gslc_tsColor colCheck, bool bChecked)  gslc_ElemXCheckboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType eRedraw)  gslc_ElemXCheckboxFindChecked(gslc_tsGui *pGui, int16_t nGroupId)  gslc_ElemXCheckboxGetState(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool bChecked)  gslc_ElemXCheckboxSetStateFunc(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bcol bChecked)  gslc_ElemXCheckboxSetStateFunc(gslc_tsGui *pGui, gslc_tsElemRef *pElemAper Ref, GSLC_CB_XCHECKBOX pfuncCb)  gslc_ElemXCheckboxSetStateHelp(gslc_tsGui *pGui, gslc_tsElemRef *pElemAper Ref, GSLC_CB_XCHECKBOX pfuncCb)	141 141 142 142 143 143

xxii CONTENTS

		9.2.1.9	gslc_ElemXCheckboxTouch(void *pvGui, void *pvElemRef, gslc_teTouch e ← Touch, int16_t nRelX, int16_t nRelY)	144
	9.2.2	Variable	Documentation	145
		9.2.2.1	ERRSTR_NULL	145
		9.2.2.2	ERRSTR PXD NULL	145
9.3	src/ele		box.h File Reference	
0.0	9.3.1		efinition Documentation	
	0.0.1	9.3.1.1	gslc_ElemXCheckboxCreate_P	
			GSLC_TYPEX_CHECKBOX	
	0.00	9.3.1.2		
	9.3.2		Documentation	
		9.3.2.1	GSLC_CB_XCHECKBOX	
	9.3.3	Enumera	ttion Type Documentation	
		9.3.3.1	gslc_teXCheckboxStyle	147
	9.3.4	Function	Documentation	148
		9.3.4.1	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	148
		9.3.4.2	gslc_ElemXCheckboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType eRedraw)	149
		9.3.4.3	gslc_ElemXCheckboxFindChecked(gslc_tsGui *pGui, int16_t nGroupId)	149
		9.3.4.4	gslc_ElemXCheckboxGetState(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef) .	150
		9.3.4.5	gslc_ElemXCheckboxSetState(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool bChecked)	150
		9.3.4.6	gslc_ElemXCheckboxSetStateFunc(gslc_tsGui *pGui, gslc_tsElemRef *pElem← Ref, GSLC_CB_XCHECKBOX pfuncCb)	150
		9.3.4.7	gslc_ElemXCheckboxToggleState(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	150
		9.3.4.8	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	151
9.4	src/ele	m/XGauge	e.c File Reference	151
	9.4.1	Function	Documentation	153
		9.4.1.1	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)	153
		9.4.1.2	gslc_ElemXGaugeDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	153

CONTENTS xxiii

		9.4.1.3	gslc_ElemXGaugeDrawProgressBar(gslc_tsGui *pGui, gslc_tsElemRef *p← ElemRef, gslc_teRedrawType eRedraw)	154
		9.4.1.4	gslc_ElemXGaugeSetFlip(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool bFlip)	
		9.4.1.5	gslc_ElemXGaugeSetIndicator(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_tsColor colGauge, uint16_t nIndicLen, uint16_t nIndicTip, bool bIndicFill)	154
		9.4.1.6	gslc_ElemXGaugeSetStyle(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_teXGaugeStyle nStyle)	155
		9.4.1.7	gslc_ElemXGaugeSetTicks(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_tsColor colTick, uint16_t nTickCnt, uint16_t nTickLen)	155
		9.4.1.8	gslc_ElemXGaugeUpdate(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16← _t nVal)	156
	9.4.2	Variable	Documentation	156
		9.4.2.1	ERRSTR_NULL	156
		9.4.2.2	ERRSTR_PXD_NULL	156
9.5	src/ele	m/XGauge	e.h File Reference	156
	9.5.1	Macro D	efinition Documentation	158
		9.5.1.1	gslc_ElemXGaugeCreate_P	158
		9.5.1.2	GSLC_TYPEX_GAUGE	158
	9.5.2	Enumera	ation Type Documentation	158
		9.5.2.1	gslc_teXGaugeStyle	158
	9.5.3	Function	Documentation	159
		9.5.3.1	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)	159
		9.5.3.2	gslc_ElemXGaugeDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	159
		9.5.3.3	gslc_ElemXGaugeDrawProgressBar(gslc_tsGui *pGui, gslc_tsElemRef *p↔ ElemRef, gslc_teRedrawType eRedraw)	160
		9.5.3.4	gslc_ElemXGaugeSetFlip(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool bFlip)	160
		9.5.3.5	gslc_ElemXGaugeSetIndicator(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_tsColor colGauge, uint16_t nIndicLen, uint16_t nIndicTip, bool bIndicFill)	160
		9.5.3.6	gslc_ElemXGaugeSetStyle(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_teXGaugeStyle nType)	161
		9.5.3.7	gslc_ElemXGaugeSetTicks(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc_tsColor colTick, uint16_t nTickCnt, uint16_t nTickLen)	161

xxiv CONTENTS

		9.5.3.8	gsic_ElemXGaugeUpdate(gsic_tsGui *pGui, gsic_tsElemRef *pElemRef, int16↔ _t nVal)	162
9.6	src/ele	m/XGraph	.c File Reference	162
	9.6.1	Function	Documentation	163
		9.6.1.1	gslc_ElemXGraphAdd(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nVal	1)163
		9.6.1.2	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 nBufMax, gslc_tsColor colGraph)	163
		9.6.1.3	gslc_ElemXGraphDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	164
		9.6.1.4	gslc_ElemXGraphScrollSet(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, uint8_t nScrollPos, uint8_t nScrollMax)	164
		9.6.1.5	gslc_ElemXGraphSetRange(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nYMin, int16_t nYMax)	165
		9.6.1.6	gslc_ElemXGraphSetStyle(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc _teXGraphStyle eStyle, uint8_t nMargin)	165
	9.6.2	Variable	Documentation	165
		9.6.2.1	ERRSTR_NULL	165
		9.6.2.2	ERRSTR_PXD_NULL	165
9.7	src/ele	m/XGraph	.h File Reference	165
	9.7.1	Macro D	efinition Documentation	167
		9.7.1.1	GSLC_TYPEX_GRAPH	167
	9.7.2	Enumera	tion Type Documentation	167
		9.7.2.1	gslc_teXGraphStyle	167
	9.7.3	Function	Documentation	167
		9.7.3.1	gslc_ElemXGraphAdd(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nVal	)167
		9.7.3.2	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)	168
		9.7.3.3	gslc_ElemXGraphDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	168
		9.7.3.4	gslc_ElemXGraphScrollSet(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, uint8_t nScrollPos, uint8_t nScrollMax)	168
		9.7.3.5	gslc_ElemXGraphSetRange(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nYMin, int16_t nYMax)	169

CONTENTS xxv

		9.7.3.6	gslc_ElemXGraphSetStyle(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc _teXGraphStyle eStyle, uint8_t nMargin)	169
9.8	src/ele	m/XListbo	x.c File Reference	169
	9.8.1	Macro De	efinition Documentation	171
		9.8.1.1	XLISTBOX_MAX_STR	171
	9.8.2	Function	Documentation	171
		9.8.2.1	gslc_ElemXListboxAddItem(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, const char *pStrItem)	171
		9.8.2.2	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	171
		9.8.2.3	gslc_ElemXListboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	172
		9.8.2.4	gslc_ElemXListboxGetItem(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nItemCurSel, char *pStrItem, uint8_t nStrItemLen)	172
		9.8.2.5	gslc_ElemXListboxGetItemCnt(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef) .	173
		9.8.2.6	gslc_ElemXListboxGetSel(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	173
		9.8.2.7	gslc_ElemXListboxItemsSetGap(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int8_t nGap, gslc_tsColor colGap)	173
		9.8.2.8	gslc_ElemXListboxItemsSetSize(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nItemW, int16_t nItemH)	174
		9.8.2.9	gslc_ElemXListboxItemsSetTxtMargin(gslc_tsGui *pGui, gslc_tsElemRef *p↔ ElemRef, int8_t nMarginW, int8_t nMarginH)	174
		9.8.2.10	gslc_ElemXListboxRecalcSize(gslc_tsXListbox *pListbox, gslc_tsRect rElem)	174
		9.8.2.11	gslc_ElemXListboxReset(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	174
		9.8.2.12	gslc_ElemXListboxSetMargin(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int8_t nMarginW, int8_t nMarginH)	175
		9.8.2.13	gslc_ElemXListboxSetScrollPos(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, uint16_t nScrollPos)	175
		9.8.2.14	gslc_ElemXListboxSetSel(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16↔ _t nltemCurSel)	175
		9.8.2.15	gslc_ElemXListboxSetSelFunc(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, GSLC_CB_XLISTBOX_SEL funcCb)	176
		9.8.2.16	gslc_ElemXListboxSetSize(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int8↔ _t nRows, int8_t nCols)	176
		9.8.2.17	gslc_ElemXListboxTouch(void *pvGui, void *pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)	176

xxvi CONTENTS

	9.8.3	Variable I	Documentation	177
		9.8.3.1	ERRSTR_NULL	177
		9.8.3.2	ERRSTR_PXD_NULL	177
9.9	src/ele	m/XListbox	x.h File Reference	177
	9.9.1	Macro De	efinition Documentation	179
		9.9.1.1	GSLC_TYPEX_LISTBOX	179
		9.9.1.2	XLISTBOX_BUF_OH_R	179
		9.9.1.3	XLISTBOX_SEL_NONE	179
		9.9.1.4	XLISTBOX_SIZE_AUTO	179
	9.9.2	Typedef I	Documentation	179
		9.9.2.1	GSLC_CB_XLISTBOX_SEL	179
	9.9.3	Function	Documentation	179
		9.9.3.1	gslc_ElemXListboxAddItem(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, const char *pStrItem)	179
		9.9.3.2	gslc_ElemXListboxCreate(gslc_tsGui *pGui, int16_t nElemId, int16_t nPage, gslc_tsXListbox *pXData, gslc_tsRect rElem, int16_t nFontId, uint8_t *pBuf↔ ltems, uint16_t nBufltemsMax, int16_t nSelDefault)	179
		9.9.3.3	gslc_ElemXListboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	180
		9.9.3.4	gslc_ElemXListboxGetItem(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nItemCurSel, char *pStrItem, uint8_t nStrItemLen)	180
		9.9.3.5	gslc_ElemXListboxGetItemCnt(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef) .	181
		9.9.3.6	gslc_ElemXListboxGetSel(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	182
		9.9.3.7	gslc_ElemXListboxItemsSetGap(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int8_t nGap, gslc_tsColor colGap)	182
		9.9.3.8	gslc_ElemXListboxItemsSetSize(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nItemW, int16_t nItemH)	182
		9.9.3.9	$gslc\_ElemXListboxItemsSetTxtMargin(gslc\_tsGui *pGui, gslc\_tsElemRef *p \leftarrow ElemRef, int8\_t nMarginW, int8\_t nMarginH) \\ \dots $	183
		9.9.3.10	gslc_ElemXListboxReset(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	183
		9.9.3.11	gslc_ElemXListboxSetMargin(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int8_t nMarginW, int8_t nMarginH)	183
		9.9.3.12	gslc_ElemXListboxSetScrollPos(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, uint16_t nScrollPos)	184

CONTENTS xxvii

		9.9.3.13	gslc_ElemXListboxSetSel(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16 ← _t nltemCurSel)	184
		9.9.3.14	gslc_ElemXListboxSetSelFunc(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, GSLC_CB_XLISTBOX_SEL funcCb)	184
		9.9.3.15	gslc_ElemXListboxSetSize(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int8← _t nRows, int8_t nCols)	185
		9.9.3.16	gslc_ElemXListboxTouch(void *pvGui, void *pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)	185
9.10	src/elei	m/XSelNu	m.c File Reference	186
	9.10.1	Variable I	Documentation	186
		9.10.1.1	ERRSTR_NULL	186
		9.10.1.2	ERRSTR_PXD_NULL	186
9.11	src/elei	m/XSelNu	m.h File Reference	186
9.12	src/elei	m/XSlider.	c File Reference	187
	9.12.1	Function	Documentation	188
		9.12.1.1	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)	188
		9.12.1.2	gslc_ElemXSliderDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	188
		9.12.1.3	gslc_ElemXSliderGetPos(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	189
		9.12.1.4	gslc_ElemXSliderSetPos(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nPos)	189
		9.12.1.5	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	189
		9.12.1.6	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	190
		9.12.1.7	gslc_ElemXSliderTouch(void *pvGui, void *pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)	190
	9.12.2	Variable I	Documentation	191
		9.12.2.1	ERRSTR_NULL	191
		9.12.2.2	ERRSTR_PXD_NULL	191
9.13	src/elei	m/XSlider.	h File Reference	191
	9.13.1	Macro De	efinition Documentation	192
		9.13.1.1	gslc_ElemXSliderCreate_P	192

xxviii CONTENTS

	9.13.1.2	GSLC_TYPEX_SLIDER	193
9.13.2	Typedef I	Documentation	193
	9.13.2.1	GSLC_CB_XSLIDER_POS	193
9.13.3	Function	Documentation	193
	9.13.3.1	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)	193
	9.13.3.2	gslc_ElemXSliderDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	194
	9.13.3.3	gslc_ElemXSliderGetPos(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	194
	9.13.3.4	gslc_ElemXSliderSetPos(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, int16_t nPos)	194
	9.13.3.5	gslc_ElemXSliderSetPosFunc(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, GSLC_CB_XSLIDER_POS funcCb)	195
	9.13.3.6	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	195
	9.13.3.7	gslc_ElemXSliderTouch(void *pvGui, void *pvElemRef, gslc_teTouch eTouch, int16_t nRelX, int16_t nRelY)	195
9.14 src/ele	m/XTextbo	ox.c File Reference	196
9.14.1	Function	Documentation	197
	9.14.1.1	gslc_ElemXTextboxAdd(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, char *pTxf	1)197
	9.14.1.2	gslc_ElemXTextboxBufAdd(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, unsigned char chNew, bool bAdvance)	
	9.14.1.3	gslc_ElemXTextboxColReset(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	197
	9.14.1.4	gslc_ElemXTextboxColSet(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc⇔ _tsColor nCol)	198
	9.14.1.5	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)	198
	9.14.1.6	gslc_ElemXTextboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	199
	9.14.1.7	gslc_ElemXTextboxLineWrAdv(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef) .	199
	9.14.1.8	gslc_ElemXTextboxReset(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	199
	9.14.1.8 9.14.1.9	gslc_ElemXTextboxReset(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)  gslc_ElemXTextboxScrollSet(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, uint8_t nScrollPos, uint8_t nScrollMax)	

CONTENTS xxix

		9.14.1.10	gslc_ElemXTextboxWrapSet(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool	
			bWrapEn)	200
	9.14.2	Variable I	Documentation	200
		9.14.2.1	ERRSTR_NULL	200
		9.14.2.2	ERRSTR_PXD_NULL	200
9.15	src/elei	m/XTextbo	x.h File Reference	200
	9.15.1	Macro De	efinition Documentation	202
		9.15.1.1	GSLC_TYPEX_TEXTBOX	202
		9.15.1.2	GSLC_XTEXTBOX_CODE_COL_RESET	202
		9.15.1.3	GSLC_XTEXTBOX_CODE_COL_SET	202
		9.15.1.4	XTEXTBOX_REDRAW_ALL	202
			XTEXTBOX_REDRAW_NONE	
	9.15.2	Function	Documentation	202
		9.15.2.1	gslc_ElemXTextboxAdd(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, char *pTx	t)202
		9.15.2.2	gslc_ElemXTextboxColReset(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	202
		9.15.2.3	gslc_ElemXTextboxColSet(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, gslc⇔ _tsColor nCol)	203
		9.15.2.4	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)	203
		9.15.2.5	gslc_ElemXTextboxDraw(void *pvGui, void *pvElemRef, gslc_teRedrawType e↔ Redraw)	204
		9.15.2.6	gslc_ElemXTextboxReset(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef)	204
		9.15.2.7	gslc_ElemXTextboxScrollSet(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, uint8_t nScrollPos, uint8_t nScrollMax)	204
		9.15.2.8	gslc_ElemXTextboxWrapSet(gslc_tsGui *pGui, gslc_tsElemRef *pElemRef, bool bWrapEn)	205
9.16	src/GU	Islice.c File	e Reference	205
	9.16.1	Enumera	tion Type Documentation	212
		9.16.1.1	gslc_teDebugPrintState	212
	9.16.2	Function	Documentation	212
		9.16.2.1	$gslc\_OrderCoord(int16\_t *pnX0, int16\_t *pnY0, int16\_t *pnX1, int16\_t *pnY1)  .$	212
		9.16.2.2	gslc_SwapCoords(int16_t *pnXa, int16_t *pnYa, int16_t *pnXb, int16_t *pnYb) .	212

CONTENTS

9.16.	3 Variable Documentation
	9.16.3.1 ERRSTR_NULL
	9.16.3.2 ERRSTR_PXD_NULL
	9.16.3.3 g_pfDebugOut
	9.16.3.4 m_nLUTSinF0X16
9.17 src/G	Ulslice.h File Reference
9.17.	1 Macro Definition Documentation
	9.17.1.1 GSLC_2PI
	9.17.1.2 GSLC_ALIGN_BOT_LEFT
	9.17.1.3 GSLC_ALIGN_BOT_MID
	9.17.1.4 GSLC_ALIGN_BOT_RIGHT
	9.17.1.5 GSLC_ALIGN_MID_LEFT
	9.17.1.6 GSLC_ALIGN_MID_MID
	9.17.1.7 GSLC_ALIGN_MID_RIGHT
	9.17.1.8 GSLC_ALIGN_TOP_LEFT
	9.17.1.9 GSLC_ALIGN_TOP_MID
	9.17.1.10 GSLC_ALIGN_TOP_RIGHT
	9.17.1.11 GSLC_ALIGNH_LEFT
	9.17.1.12 GSLC_ALIGNH_MID
	9.17.1.13 GSLC_ALIGNH_RIGHT
	9.17.1.14 GSLC_ALIGNV_BOT
	9.17.1.15 GSLC_ALIGNV_MID
	9.17.1.16 GSLC_ALIGNV_TOP
	9.17.1.17 GSLC_COL_BLACK
	9.17.1.18 GSLC_COL_BLUE
	9.17.1.19 GSLC_COL_BLUE_DK1
	9.17.1.20 GSLC_COL_BLUE_DK2
	9.17.1.21 GSLC_COL_BLUE_DK3
	9.17.1.22 GSLC_COL_BLUE_DK4
	9.17.1.23 GSLC_COL_BLUE_LT1

CONTENTS xxxi

9.17.1.24 GSLC_COL_BLUE_LT2	:28
9.17.1.25 GSLC_COL_BLUE_LT3	28
9.17.1.26 GSLC_COL_BLUE_LT4	28
9.17.1.27 GSLC_COL_BROWN	28
9.17.1.28 GSLC_COL_CYAN	28
9.17.1.29 GSLC_COL_GRAY	28
9.17.1.30 GSLC_COL_GRAY_DK1	28
9.17.1.31 GSLC_COL_GRAY_DK2	28
9.17.1.32 GSLC_COL_GRAY_DK3	28
9.17.1.33 GSLC_COL_GRAY_LT1	29
9.17.1.34 GSLC_COL_GRAY_LT2	29
9.17.1.35 GSLC_COL_GRAY_LT3	29
9.17.1.36 GSLC_COL_GREEN	29
9.17.1.37 GSLC_COL_GREEN_DK1	29
9.17.1.38 GSLC_COL_GREEN_DK2	29
9.17.1.39 GSLC_COL_GREEN_DK3	29
9.17.1.40 GSLC_COL_GREEN_DK4	29
9.17.1.41 GSLC_COL_GREEN_LT1	29
9.17.1.42 GSLC_COL_GREEN_LT2	29
9.17.1.43 GSLC_COL_GREEN_LT3	:30
9.17.1.44 GSLC_COL_GREEN_LT4	:30
9.17.1.45 GSLC_COL_MAGENTA	:30
9.17.1.46 GSLC_COL_ORANGE	:30
9.17.1.47 GSLC_COL_PURPLE	:30
9.17.1.48 GSLC_COL_RED	:30
9.17.1.49 GSLC_COL_RED_DK1	:30
9.17.1.50 GSLC_COL_RED_DK2	:30
9.17.1.51 GSLC_COL_RED_DK3	:30
9.17.1.52 GSLC_COL_RED_DK4	:30
9.17.1.53 GSLC_COL_RED_LT1	231

xxxii CONTENTS

	9.17.1.54	4 (	GS	LC	_C	OL	_R	EC	ו_כ	LT.	2											٠			231
	9.17.1.55	5 (	GS	LC	:_C	OL	_R	ΕC	D_I	LT	3														231
	9.17.1.56	6 (	GS	LC	_C	OL	_R	ΕC	ו_כ	LΤ	4														231
	9.17.1.57	7 (	GS	LC	_C	OL	_TI	ΕA	٨L																231
	9.17.1.58	8 (	GS	LC	:_C	OL	_W	/HI	ITE	Ε.															231
	9.17.1.59	9 (	GS	LC	:_C	OL	_YI	EL	_L(	NC	<b>/</b> .														231
	9.17.1.60	0 0	GS	LC	:_C	OL	_YI	EL	_L(	NC	<b>/</b> _[	DΚ													231
	9.17.1.61	1 (	GS	LC	_C	OL	MC	N	IO_	_BI	LA	Ck	٠.												231
	9.17.1.62	2 (	GS	LC	_C	OL	MC	N	IO_	_W	/HI	ITE													231
	9.17.1.63	3 (	GS	LC	<u>_</u> E	LEI	M_I	FE	Ξ <b>Α</b> _	_C	LIC	CK	_E	N											232
	9.17.1.64	4 (	GS	LC	<u>_</u> E	LEI	M_I	FE	Ξ <b>Α</b> _	_F	ILL	E	ΞN												232
	9.17.1.65	5 (	GS	LC	<u>_</u> E	LEI	M_I	FE	Ξ <b>Α</b> _	_FI	RA	M	E_	ΕN											232
	9.17.1.66	6 (	GS	LC	<u>_</u> E	LEI	M_I	FE	Ξ <b>Α</b> _	_G	LC	)W	<u>_</u> E	N											232
	9.17.1.67	7 (	GS	LC	<u>_</u> E	LEI	M_I	FE	Ξ <b>Α</b> _	_N	10	٧E													232
	9.17.1.68	8 (	GS	LC	<u>_</u> E	LEI	M_I	FE	Ξ <b>Α</b> _	_V	AL	.ID													232
	9.17.1.69	9 (	GS	LC	<u>_</u> E	LEI	ИR	EF	F_I	DE	FÆ	٩U	LT												232
	9.17.1.70	0 0	GS	LC	<u>_</u> P	ME	М																		232
9.17.2	Typedef [	Do	ocu	me	enta	atio	n .																		232
	9.17.2.1	C	GS	LC	:_C	B_I	DE	Βl	JG	i_C	DU.	Т													232
	9.17.2.2	C	GS	LC	:_C	B_I	DR	A۷	N																232
	9.17.2.3	C	GS	LC	_C	B_I	ΕVI	ΕN	٧T																233
	9.17.2.4	C	GS	LC	:_C	B_I	PIN	<b>\_</b> F	PC	)LL															233
	9.17.2.5	C	GS	LC	:_C	B_	TIC	K																	233
	9.17.2.6	C	GS	LC	_C	B_	ТО	UC	СН	١.															233
	9.17.2.7	g	gslo	o_t	sC	olor	٠.																		233
	9.17.2.8	g	gslo	o_t	sEl	lem																			233
	9.17.2.9	g	gslo	o_t	sEv	ven	t.																		233
	9.17.2.10	0 g	gslo	o_t	sEv	ven	tTo	uc	h																233
	9.17.2.11	1 g	gslo	o_t	sPt	t.																			234
	9.17.2.12	2 g	gslo	o_t:	sR	ect																			234

CONTENTS xxxiii

9.17.	3 Enumeration Type Documentation
	9.17.3.1 gslc_teAction
	9.17.3.2 gslc_teElemId
	9.17.3.3 gslc_teElemInd
	9.17.3.4 gslc_teElemRefFlags
	9.17.3.5 gslc_teEventSubType
	9.17.3.6 gslc_teEventType
	9.17.3.7 gslc_teFontId
	9.17.3.8 gslc_teFontRefType
	9.17.3.9 gslc_teGroupId
	9.17.3.10 gslc_teImgRefFlags
	9.17.3.11 gslc_telnitStat
	9.17.3.12 gslc_teInputRawEvent
	9.17.3.13 gslc_tePageId
	9.17.3.14 gslc_tePin
	9.17.3.15 gslc_teRedrawType
	9.17.3.16 gslc_teStackPage
	9.17.3.17 gslc_teTouch
	9.17.3.18 gslc_teTxtFlags
	9.17.3.19 gslc_teTypeCore
9.17.	4 Variable Documentation
	9.17.4.1 g_pfDebugOut
9.18 src/G	Ulslice_config.h File Reference
9.19 src/G	Ulslice_config_ard.h File Reference
9.19.	1 Macro Definition Documentation
	9.19.1.1 ADAGFX_PIN_CLK
	9.19.1.2 ADAGFX_PIN_CS
	9.19.1.3 ADAGFX_PIN_DC
	9.19.1.4 ADAGFX_PIN_MISO
	9.19.1.5 ADAGFX_PIN_MOSI

CONTENTS

9.19.1.6 ADAGFX_PIN_RD
9.19.1.7 ADAGFX_PIN_RST
9.19.1.8 ADAGFX_PIN_SDCS
9.19.1.9 ADAGFX_PIN_WR
9.19.1.10 ADAGFX_SPI_HW
9.19.1.11 ADATOUCH_FLIP_X
9.19.1.12 ADATOUCH_FLIP_Y
9.19.1.13 ADATOUCH_I2C_ADDR
9.19.1.14 ADATOUCH_I2C_HW
9.19.1.15 ADATOUCH_PIN_CS
9.19.1.16 ADATOUCH_SPI_HW
9.19.1.17 ADATOUCH_SPI_SW
9.19.1.18 ADATOUCH_SWAP_XY
9.19.1.19 ADATOUCH_X_MAX
9.19.1.20 ADATOUCH_X_MIN
9.19.1.21 ADATOUCH_Y_MAX
9.19.1.22 ADATOUCH_Y_MIN
9.19.1.23 DEBUG_ERR
9.19.1.24 DRV_DISP_ADAGFX
9.19.1.25 DRV_DISP_ADAGFX_ILI9341
9.19.1.26 DRV_TOUCH_ADA_STMPE610
9.19.1.27 GSLC_BMP_TRANS_EN
9.19.1.28 GSLC_BMP_TRANS_RGB
9.19.1.29 GSLC_CLIP_EN
9.19.1.30 GSLC_DEV_TOUCH
9.19.1.31 GSLC_FEATURE_COMPOUND
9.19.1.32 GSLC_FEATURE_INPUT
9.19.1.33 GSLC_FEATURE_XGAUGE_RADIAL
9.19.1.34 GSLC_FEATURE_XGAUGE_RAMP
9.19.1.35 GSLC_FEATURE_XTEXTBOX_EMBED

CONTENTS XXXV

		9.19.1.36	G GSLC_LOCAL_STR	 	243
		9.19.1.37	7 GSLC_LOCAL_STR_LEN	 	243
		9.19.1.38	B GSLC_ROTATE	 	243
		9.19.1.39	GSLC_SD_BUFFPIXEL	 	243
		9.19.1.40	O GSLC_SD_EN	 	243
		9.19.1.41	1 GSLC_TOUCH_MAX_EVT	 	244
		9.19.1.42	2 GSLC_USE_FLOAT	 	244
		9.19.1.43	3 GSLC_USE_PROGMEM	 	244
		9.19.1.44	4 TOUCH_ROTATION_DATA	 	244
		9.19.1.45	5 TOUCH_ROTATION_FLIPX	 	244
		9.19.1.46	TOUCH_ROTATION_FLIPY	 	244
		9.19.1.47	7 TOUCH_ROTATION_SWAPXY	 	244
9.20	o src/GU	Islice_con	nfig_linux.h File Reference	 	244
	9.20.1	Macro De	efinition Documentation	 	245
		9.20.1.1	ADATOUCH_FLIP_X	 	245
		9.20.1.2	ADATOUCH_FLIP_Y	 	245
		9.20.1.3	ADATOUCH_SWAP_XY	 	245
		9.20.1.4	DEBUG_ERR	 	245
		9.20.1.5	DRV_DISP_SDL1	 	245
		9.20.1.6	DRV_SDL_FIX_START	 	245
		9.20.1.7	DRV_SDL_MOUSE_SHOW	 	245
		9.20.1.8	DRV_TOUCH_TSLIB	 	245
		9.20.1.9	GSLC_BMP_TRANS_EN	 	245
		9.20.1.10	GSLC_BMP_TRANS_RGB	 	245
		9.20.1.11	1 GSLC_DEV_FB	 	245
		9.20.1.12	2 GSLC_DEV_TOUCH	 	245
		9.20.1.13	3 GSLC_DEV_VID_DRV	 	245
		9.20.1.14	4 GSLC_FEATURE_COMPOUND	 	245
		9.20.1.15	GSLC_FEATURE_INPUT	 	245
		9.20.1.16	GSLC_FEATURE_XGAUGE_RADIAL	 	245

xxxvi CONTENTS

		9.20.1.17	GSLC_FEATURE_XGAUGE_RAMP	245
		9.20.1.18	GSLC_FEATURE_XTEXTBOX_EMBED	245
		9.20.1.19	GSLC_LOCAL_STR	245
		9.20.1.20	GSLC_LOCAL_STR_LEN	245
		9.20.1.21	GSLC_TOUCH_MAX_EVT	245
		9.20.1.22	GSLC_USE_FLOAT	245
		9.20.1.23	GSLC_USE_PROGMEM	245
9.21 s	rc/GUI	slice_drv.	h File Reference	245
9.22 s	rc/GUI	slice_drv_	_adagfx.cpp File Reference	246
9.23 s	rc/GUI	slice_drv_	_adagfx.h File Reference	246
9	.23.1	Detailed I	Description	249
9	.23.2	Macro De	efinition Documentation	249
		9.23.2.1	DRV_HAS_DRAW_CIRCLE_FILL	249
		9.23.2.2	DRV_HAS_DRAW_CIRCLE_FRAME	249
		9.23.2.3	DRV_HAS_DRAW_LINE	249
		9.23.2.4	DRV_HAS_DRAW_POINT	249
		9.23.2.5	DRV_HAS_DRAW_POINTS	249
		9.23.2.6	DRV_HAS_DRAW_RECT_FILL	249
		9.23.2.7	DRV_HAS_DRAW_RECT_FRAME	249
		9.23.2.8	DRV_HAS_DRAW_TEXT	249
		9.23.2.9	DRV_HAS_DRAW_TRI_FILL	249
		9.23.2.10	DRV_HAS_DRAW_TRI_FRAME	250
		9.23.2.11	DRV_OVERRIDE_TXT_ALIGN	250
9	.23.3	Function	Documentation	250
		9.23.3.1	gslc_DrvAdaptColorToRaw(gslc_tsColor nCol)	250
		9.23.3.2	gslc_DrvDestruct(gslc_tsGui *pGui)	250
		9.23.3.3	gslc_DrvDrawBkgnd(gslc_tsGui *pGui)	250
		9.23.3.4	gslc_DrvDrawBmp24FromMem(gslc_tsGui *pGui, int16_t nDstX, int16_t nDstY, const unsigned char *pBitmap, bool bProgMem)	250
		9.23.3.5	gslc_DrvDrawFillCircle(gslc_tsGui *pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)	251

CONTENTS xxxvii

9.23.3.6	gslc_DrvDrawFillRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol) .	251
9.23.3.7	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)	252
9.23.3.8	gslc_DrvDrawFrameCircle(gslc_tsGui *pGui, int16_t nMidX, int16_t nMid↔ Y, uint16_t nRadius, gslc_tsColor nCol)	252
9.23.3.9	gslc_DrvDrawFrameRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol)	252
9.23.3.10	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)	253
9.23.3.11	gslc_DrvDrawImage(gslc_tsGui *pGui, int16_t nDstX, int16_t nDstY, gslc_ts⇔ ImgRef sImgRef)	253
9.23.3.12	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	254
9.23.3.13	gslc_DrvDrawMonoFromMem(gslc_tsGui *pGui, int16_t nDstX, int16_t nDst↔ Y, const unsigned char *pBitmap, bool bProgMem)	254
9.23.3.14	gslc_DrvDrawPoint(gslc_tsGui *pGui, int16_t nX, int16_t nY, gslc_tsColor nCol) .	254
9.23.3.15	gslc_DrvDrawPoints(gslc_tsGui *pGui, gslc_tsPt *asPt, uint16_t nNumPt, gslc↔ _tsColor nCol)	255
9.23.3.16	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)	255
9.23.3.17	gslc_DrvFontAdd(gslc_teFontRefType eFontRefType, const void *pvFontRef, uint16_t nFontSz)	255
9.23.3.18	gslc_DrvFontsDestruct(gslc_tsGui *pGui)	256
9.23.3.19	gslc_DrvGetNameDisp(gslc_tsGui *pGui)	256
9.23.3.20	gslc_DrvGetNameTouch(gslc_tsGui *pGui)	256
9.23.3.21	gslc_DrvGetTouch(gslc_tsGui *pGui, int16_t *pnX, int16_t *pnY, uint16_t *pn↔ Press, gslc_teInputRawEvent *peInputEvent, int16_t *pnInputVal)	256
9.23.3.22	gslc_DrvGetTxtSize(gslc_tsGui *pGui, gslc_tsFont *pFont, const char *pStr, gslc_teTxtFlags eTxtFlags, int16_t *pnTxtX, int16_t *pnTxtY, uint16_t *pnTxt← SzW, uint16_t *pnTxtSzH)	257
9.23.3.23	gslc_DrvImageDestruct(void *pvImg)	257
9.23.3.24	gslc_DrvInit(gslc_tsGui *pGui)	258
9.23.3.25	gslc_DrvInitTouch(gslc_tsGui *pGui, const char *acDev)	258
9.23.3.26	gslc_DrvInitTs(gslc_tsGui *pGui, const char *acDev)	258
9.23.3.27	gslc_DrvLoadImage(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	259
9.23.3.28	gslc_DrvPageFlipNow(gslc_tsGui *pGui)	259

xxxviii CONTENTS

9.23.3.29 gslc_DrvRotate(gslc_tsGui *pGui, uint8_t nRotation)	259
9.23.3.30 gslc_DrvSetBkgndColor(gslc_tsGui *pGui, gslc_tsColor nCol)	260
9.23.3.31 gslc_DrvSetBkgndImage(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	260
9.23.3.32 gslc_DrvSetClipRect(gslc_tsGui *pGui, gslc_tsRect *pRect)	260
9.23.3.33 gslc_DrvSetElemImageGlow(gslc_tsGui *pGui, gslc_tsElem *pElem, gslc_ts↔ ImgRef sImgRef)	261
9.23.3.34 gslc_DrvSetElemImageNorm(gslc_tsGui *pGui, gslc_tsElem *pElem, gslc_ts↔ ImgRef sImgRef)	261
9.24 src/GUIslice_drv_m5stack.cpp File Reference	261
9.25 src/GUIslice_drv_m5stack.h File Reference	262
9.25.1 Detailed Description	264
9.25.2 Macro Definition Documentation	264
9.25.2.1 DRV_HAS_DRAW_CIRCLE_FILL	264
9.25.2.2 DRV_HAS_DRAW_CIRCLE_FRAME	264
9.25.2.3 DRV_HAS_DRAW_LINE	265
9.25.2.4 DRV_HAS_DRAW_POINT	265
9.25.2.5 DRV_HAS_DRAW_POINTS	265
9.25.2.6 DRV_HAS_DRAW_RECT_FILL	265
9.25.2.7 DRV_HAS_DRAW_RECT_FRAME	265
9.25.2.8 DRV_HAS_DRAW_TEXT	265
9.25.2.9 DRV_HAS_DRAW_TRI_FILL	265
9.25.2.10 DRV_HAS_DRAW_TRI_FRAME	265
9.25.2.11 DRV_OVERRIDE_TXT_ALIGN	265
9.25.3 Function Documentation	265
9.25.3.1 gslc_DrvAdaptColorToRaw(gslc_tsColor nCol)	265
9.25.3.2 gslc_DrvDestruct(gslc_tsGui *pGui)	265
9.25.3.3 gslc_DrvDrawBkgnd(gslc_tsGui *pGui)	266
9.25.3.4 gslc_DrvDrawBmp24FromMem(gslc_tsGui *pGui, int16_t nDstX, int16_t nDstY, const unsigned char *pBitmap, bool bProgMem)	266
9.25.3.5 gslc_DrvDrawFillCircle(gslc_tsGui *pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)	266
9.25.3.6 gslc_DrvDrawFillRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol) .	267

CONTENTS xxxix

9.25.3.7	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)	267
9.25.3.8	gslc_DrvDrawFrameCircle(gslc_tsGui *pGui, int16_t nMidX, int16_t nMid↔ Y, uint16_t nRadius, gslc_tsColor nCol)	268
9.25.3.9	gslc_DrvDrawFrameRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol)	268
9.25.3.10	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)	268
9.25.3.11	gslc_DrvDrawImage(gslc_tsGui *pGui, int16_t nDstX, int16_t nDstY, gslc_ts⇔ ImgRef sImgRef)	269
9.25.3.12	gslc_DrvDrawLine(gslc_tsGui *pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)	269
9.25.3.13	gslc_DrvDrawMonoFromMem(gslc_tsGui *pGui, int16_t nDstX, int16_t nDst↔ Y, const unsigned char *pBitmap, bool bProgMem)	269
9.25.3.14	gslc_DrvDrawPoint(gslc_tsGui *pGui, int16_t nX, int16_t nY, gslc_tsColor nCol) .	270
9.25.3.15	gslc_DrvDrawPoints(gslc_tsGui *pGui, gslc_tsPt *asPt, uint16_t nNumPt, gslc _tsColor nCol)	270
9.25.3.16	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)	270
9.25.3.17	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_teTxt← Flags eTxtFlags, gslc_tsColor colTxt, gslc_tsColor colBg)	271
9.25.3.18	gslc_DrvFontAdd(gslc_teFontRefType eFontRefType, const void *pvFontRef, uint16_t nFontSz)	271
9.25.3.19	gslc_DrvFontsDestruct(gslc_tsGui *pGui)	272
9.25.3.20	gslc_DrvGetNameDisp(gslc_tsGui *pGui)	272
9.25.3.21	gslc_DrvGetNameTouch(gslc_tsGui *pGui)	272
9.25.3.22	gslc_DrvGetTxtSize(gslc_tsGui *pGui, gslc_tsFont *pFont, const char *pStr, gslc_teTxtFlags eTxtFlags, int16_t *pnTxtX, int16_t *pnTxtY, uint16_t *pnTxt← SzW, uint16_t *pnTxtSzH)	272
9.25.3.23	gslc_DrvImageDestruct(void *pvImg)	273
9.25.3.24	gslc_DrvInit(gslc_tsGui *pGui)	273
9.25.3.25	gslc_DrvInitTs(gslc_tsGui *pGui, const char *acDev)	274
9.25.3.26	gslc_DrvLoadImage(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	274
9.25.3.27	gslc_DrvPageFlipNow(gslc_tsGui *pGui)	274
9.25.3.28	gslc_DrvRotate(gslc_tsGui *pGui, uint8_t nRotation)	275
9.25.3.29	gslc_DrvSetBkgndColor(gslc_tsGui *pGui, gslc_tsColor nCol)	275

xI CONTENTS

		9.25.3.30	gslc_DrvSetBkgndImage(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	275
		9.25.3.31	gslc_DrvSetClipRect(gslc_tsGui *pGui, gslc_tsRect *pRect)	276
		9.25.3.32	gslc_DrvSetElemImageGlow(gslc_tsGui *pGui, gslc_tsElem *pElem, gslc_ts↔ ImgRef sImgRef)	276
		9.25.3.33	$gslc\_DrvSetElemImageNorm(gslc\_tsGui *pGui, gslc\_tsElem *pElem, gslc\_ts \leftrightarrow ImgRef sImgRef)$	276
	9.25.4	Variable [	Documentation	276
		9.25.4.1	ERRSTR_NULL	276
		9.25.4.2	ERRSTR_PXD_NULL	277
9.26	src/GU	Islice_drv_	sdl.c File Reference	277
9.27	src/GU	Islice_drv_	sdl.h File Reference	277
	9.27.1	Detailed [	Description	279
	9.27.2	Macro De	finition Documentation	279
		9.27.2.1	DRV_HAS_DRAW_POINT	279
		9.27.2.2	DRV_OVERRIDE_TXT_ALIGN	279
	9.27.3	Function I	Documentation	279
		9.27.3.1	gslc_DrvAdaptColor(gslc_tsColor sCol)	279
		9.27.3.2	gslc_DrvAdaptRect(gslc_tsRect rRect)	280
		9.27.3.3	gslc_DrvCleanStart(const char *sTTY)	280
		9.27.3.4	gslc_DrvDestruct(gslc_tsGui *pGui)	280
		9.27.3.5	gslc_DrvDrawBkgnd(gslc_tsGui *pGui)	281
		9.27.3.6	gslc_DrvDrawFillRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol) .	281
		9.27.3.7	gslc_DrvDrawFrameRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol)	281
		9.27.3.8	gslc_DrvDrawImage(gslc_tsGui *pGui, int16_t nDstX, int16_t nDstY, gslc_ts⇔ ImgRef sImgRef)	281
		9.27.3.9	gslc_DrvDrawLine(gslc_tsGui *pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)	282
		9.27.3.10	gslc_DrvDrawPoint(gslc_tsGui *pGui, int16_t nX, int16_t nY, gslc_tsColor nCol) .	282
		9.27.3.11	gslc_DrvDrawPoints(gslc_tsGui *pGui, gslc_tsPt *asPt, uint16_t nNumPt, gslc ← _tsColor nCol)	283
		9.27.3.12	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)	284

CONTENTS xli

· · · · · · · · · · · · · · · · · · ·	gslc_teFontRefType eFontRefType, const void *pvFontRef,	284
9.27.3.14 gslc_DrvFontsDes	struct(gslc_tsGui *pGui)	285
9.27.3.15 gslc_DrvGetName	eDisp(gslc_tsGui *pGui)	285
9.27.3.16 gslc_DrvGetName	eTouch(gslc_tsGui *pGui)	285
<del>-</del> -	n(gslc_tsGui *pGui, int16_t *pnX, int16_t *pnY, uint16_t *pn↔ utRawEvent *peInputEvent, int16_t *pnInputVal)	285
gslc_teTxtFlags e	ze(gslc_tsGui *pGui, gslc_tsFont *pFont, const char *pStr, TxtFlags, int16_t *pnTxtX, int16_t *pnTxtY, uint16_t *pnTxt \cdot TxtSzH)	286
9.27.3.19 gslc_DrvImageDe	struct(void *pvImg)	286
9.27.3.20 gslc_DrvInit(gslc_	tsGui *pGui)	286
9.27.3.21 gslc_DrvInitTouch	(gslc_tsGui *pGui, const char *acDev)	287
9.27.3.22 gslc_DrvLoadIma	ge(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	287
9.27.3.23 gslc_DrvPageFlipI	Now(gslc_tsGui *pGui)	288
9.27.3.24 gslc_DrvReportInf	ioPost()	288
9.27.3.25 gslc_DrvReportInf	oPre()	288
9.27.3.26 gslc_DrvRotate(gs	slc_tsGui *pGui, uint8_t nRotation)	288
9.27.3.27 gslc_DrvSetBkgno	dColor(gslc_tsGui *pGui, gslc_tsColor nCol)	288
9.27.3.28 gslc_DrvSetBkgnc	dImage(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	289
9.27.3.29 gslc_DrvSetClipRo	ect(gslc_tsGui *pGui, gslc_tsRect *pRect)	289
<del>-</del> -	mageGlow(gslc_tsGui *pGui, gslc_tsElem *pElem, gslc_ts↔	289
<u> </u>	mageNorm(gslc_tsGui *pGui, gslc_tsElem *pElem, gslc_ts↔	290
9.28 src/GUIslice_drv_tft_espi.cpp File Re	eference	290
9.29 src/GUIslice_drv_tft_espi.h File Refe	erence	290
9.29.1 Detailed Description		293
9.29.2 Macro Definition Documenta	tion	293
9.29.2.1 DRV_HAS_DRAW	V_CIRCLE_FILL	293
9.29.2.2 DRV_HAS_DRAW	V_CIRCLE_FRAME	293
9.29.2.3 DRV_HAS_DRAW	V_LINE	293
9.29.2.4 DRV_HAS_DRAW	V_POINT	294

XIII CONTENTS

	9.29.2.5	DRV_HAS_DRAW_POINTS	294
	9.29.2.6	DRV_HAS_DRAW_RECT_FILL	294
	9.29.2.7	DRV_HAS_DRAW_RECT_FRAME	294
	9.29.2.8	DRV_HAS_DRAW_TEXT	294
	9.29.2.9	DRV_HAS_DRAW_TRI_FILL	294
	9.29.2.10	DRV_HAS_DRAW_TRI_FRAME	294
	9.29.2.11	DRV_OVERRIDE_TXT_ALIGN	294
9.29.3	Function	Documentation	294
	9.29.3.1	gslc_DrvAdaptColorToRaw(gslc_tsColor nCol)	294
	9.29.3.2	gslc_DrvDestruct(gslc_tsGui *pGui)	294
	9.29.3.3	gslc_DrvDrawBkgnd(gslc_tsGui *pGui)	295
	9.29.3.4	gslc_DrvDrawBmp24FromMem(gslc_tsGui *pGui, int16_t nDstX, int16_t nDstY, const unsigned char *pBitmap, bool bProgMem)	295
	9.29.3.5	gslc_DrvDrawFillCircle(gslc_tsGui *pGui, int16_t nMidX, int16_t nMidY, uint16_t nRadius, gslc_tsColor nCol)	295
	9.29.3.6	gslc_DrvDrawFillRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol) .	296
	9.29.3.7	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)	296
	9.29.3.8	gslc_DrvDrawFrameCircle(gslc_tsGui *pGui, int16_t nMidX, int16_t nMid↔ Y, uint16_t nRadius, gslc_tsColor nCol)	297
	9.29.3.9	gslc_DrvDrawFrameRect(gslc_tsGui *pGui, gslc_tsRect rRect, gslc_tsColor nCol)	297
	9.29.3.10	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)	297
	9.29.3.11	gslc_DrvDrawImage(gslc_tsGui *pGui, int16_t nDstX, int16_t nDstY, gslc_ts⇔ ImgRef sImgRef)	298
	9.29.3.12	gslc_DrvDrawLine(gslc_tsGui *pGui, int16_t nX0, int16_t nY0, int16_t nX1, int16_t nY1, gslc_tsColor nCol)	298
	9.29.3.13	gslc_DrvDrawMonoFromMem(gslc_tsGui *pGui, int16_t nDstX, int16_t nDst↔ Y, const unsigned char *pBitmap, bool bProgMem)	298
	9.29.3.14	gslc_DrvDrawPoint(gslc_tsGui *pGui, int16_t nX, int16_t nY, gslc_tsColor nCol) .	299
	9.29.3.15	gslc_DrvDrawPoints(gslc_tsGui *pGui, gslc_tsPt *asPt, uint16_t nNumPt, gslc _tsColor nCol)	299
	9.29.3.16	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)	299

CONTENTS xliii

		9.29.3.17	$ \begin{split} & 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\_teTxt \\ & Flags eTxtFlags, gslc\_tsColor colTxt, gslc\_tsColor colBg) \\ & \dots \dots \dots . \end{split} $	300
		9.29.3.18	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	300
		9.29.3.19	gslc_DrvFontsDestruct(gslc_tsGui *pGui)	301
		9.29.3.20	gslc_DrvGetNameDisp(gslc_tsGui *pGui)	301
		9.29.3.21	gslc_DrvGetNameTouch(gslc_tsGui *pGui)	301
		9.29.3.22	$ \begin{split} & gslc\_DrvGetTxtSize(gslc\_tsGui *pGui, ~gslc\_tsFont *pFont, ~const ~char *pStr, \\ & gslc\_teTxtFlags ~eTxtFlags, ~int16\_t *pnTxtX, ~int16\_t *pnTxtY, ~uint16\_t *pnTxtC \\ & SzW, ~uint16\_t *pnTxtSzH) ~. & . & . & . & . & . & . & . & . \\ \end{split} $	302
		9.29.3.23	gslc_DrvImageDestruct(void *pvImg)	302
		9.29.3.24	gslc_DrvInit(gslc_tsGui *pGui)	302
		9.29.3.25	gslc_DrvInitTs(gslc_tsGui *pGui, const char *acDev)	303
		9.29.3.26	gslc_DrvLoadImage(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	303
		9.29.3.27	gslc_DrvPageFlipNow(gslc_tsGui *pGui)	303
		9.29.3.28	gslc_DrvRotate(gslc_tsGui *pGui, uint8_t nRotation)	304
		9.29.3.29	gslc_DrvSetBkgndColor(gslc_tsGui *pGui, gslc_tsColor nCol)	304
		9.29.3.30	gslc_DrvSetBkgndImage(gslc_tsGui *pGui, gslc_tsImgRef sImgRef)	304
		9.29.3.31	gslc_DrvSetClipRect(gslc_tsGui *pGui, gslc_tsRect *pRect)	305
		9.29.3.32	$gslc\_DrvSetElemImageGlow(gslc\_tsGui *pGui, gslc\_tsElem *pElem, gslc\_ts \hookleftarrow ImgRef sImgRef) \dots \dots$	305
		9.29.3.33	$gslc\_DrvSetElemImageNorm(gslc\_tsGui *pGui, gslc\_tsElem *pElem, gslc\_ts \hookleftarrow ImgRef sImgRef)$	305
9.30	src/GU	Islice_ex.h	File Reference	306
9.31	src/GU	Islice_th.c	pp File Reference	306
	9.31.1	Function	Documentation	307
		9.31.1.1	gslc_getTouchHandler(void)	307
		9.31.1.2	gslc_InitTouchHandler(TouchHandler *pTH)	307
	9.31.2	Variable [	Documentation	307
		9.31.2.1	pTouchHandler	307
9.32	src/GU	Islice_th.h	File Reference	307
	9.32.1	Function	Documentation	308
		9.32.1.1	gslc_getTouchHandler(void)	308
		9.32.1.2	gslc_InitTouchHandler(TouchHandler *pTHO)	308
9.33	src/GU	Islice_th_>	(PT2046.h File Reference	308
9.34	src/GU	Islice_vers	ion.h File Reference	309
	9.34.1	Macro De	finition Documentation	309
		9.34.1.1	GUISLICE_VER	309

# **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, 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, 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, Waveshare, 4D Cape

2 GUIslice library

- Display drivers include:
  - ILI9341, ST7735, SSD1306, HX8347D, HX8357, PCD8544
- Touchscreen control including:
  - STMPE610, FT6206, XPT2046, 4-wire, tslib
- Foreign characters / UTF-8 encoding (in SDL mode), anti-aliased fonts (in TFT\_eSPI mode)
- · Dynamic display rotation
- GPIO / pin / keyboard 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

# **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 nPageld, int16\_t nElemId)
Unused?

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

4 Todo List

# **Module Index**

# 3.1 Modules

# Here is a list of all modules:

eneral Functions	13
raphics General Functions	19
raphics Primitive Functions	26
ont Functions	32
age Functions	34
ement Functions	39
Element: Creation Functions	40
Element: General Functions	44
Element: Update Functions	45
ouchscreen Functions	56
put Mapping Functions	60
eneral Purpose Macros	61
ash-based Element Macros	62
ternal Functions	65
Internal: Misc Functions	81
Internal: Element Functions	82
Internal: Page Functions	87
Internal: Element Collection Functions	92
Internal: Element Collection Event Functions	98
Internal: Tracking Functions	00
Internal: Cleanup Functions	02

6 Module Index

# **Hierarchical Index**

# 4.1 Class Hierarchy

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

slc_tsCollect	07
slc_tsColor	80
slc_tsDriver	09
slc_tsElem	10
slc_tsElemRef	12
slc_tsEvent	12
slc_tsEventTouch	13
slc_tsFont	13
slc_tsGui	14
slc_tsImgRef	16
slc_tsInputMap	
slc_tsPage	18
slc_tsPt	18
slc_tsRect	
slc_tsXCheckbox	20
slc_tsXGauge	
slc_tsXGraph	
slc_tsXListbox	
slc_tsXSlider	
slc_tsXTextbox	
HPoint	
ouchHandler	37
TouchHandler XPT2046	38

8 Hierarchical Index

# **Data Structure Index**

# 5.1 Data Structures

Here are the data structures with brief descriptions:

gslc_tsCollect
Element collection struct
gslc_tsColor
Color structure. Defines RGB triplet
gslc_tsDriver
gslc_tsElem
Element Struct
gslc_tsElemRef
Element reference structure
gslc_tsEvent
Event structure
gslc_tsEventTouch
Structure used to pass touch data through event
gslc_tsFont  Font reference structure
gslc_tsGui
GUI structure
gslc_tsImgRef
Image reference structure
gslc_tsInputMap
Input mapping
gslc tsPage
Page structure
gslc_tsPt
Define point coordinates
gslc_tsRect
Rectangular region. Defines X,Y corner coordinates plus dimensions
gslc_tsXCheckbox
Extended data for Checkbox element
gslc_tsXGauge
Extended data for Gauge element
gslc_tsXGraph
Extended data for Graph element
gslc_tsXListbox
Extended data for Liethov element

10 Data Structure Index

gslc_tsXSlider	
Extended data for Slider element	
gslc_tsXTextbox	
Extended data for Textbox element	
THPoint	
TouchHandler	
TouchHandler XPT2046	

# File Index

# 6.1 File List

Here is a list of all files with brief descriptions:

src/GUIslice.c
src/GUIslice.h
src/GUIslice_config.h
src/GUIslice_config_ard.h
src/GUIslice_config_linux.h
src/GUIslice_drv.h
src/GUIslice_drv_adagfx.cpp
src/GUIslice_drv_adagfx.h
GUIslice library (driver layer for Adafruit-GFX)
src/GUIslice_drv_m5stack.cpp
src/GUIslice_drv_m5stack.h
GUIslice library (driver layer for M5stack)
src/GUIslice_drv_sdl.c
src/GUIslice_drv_sdl.h
GUIslice library (driver layer for LINUX / SDL)
src/GUIslice_drv_tft_espi.cpp
src/GUIslice_drv_tft_espi.h
GUIslice library (driver layer for TFT-eSPI)
src/GUIslice_ex.h
src/GUIslice_th.cpp
src/GUIslice_th.h
src/GUIslice_th_XPT2046.h
src/GUIslice_version.h
src/elem/XCheckbox.c
src/elem/XCheckbox.h
src/elem/XGauge.c
src/elem/XGauge.h
src/elem/XGraph.c
src/elem/XGraph.h
src/elem/XListbox.c
src/elem/XListbox.h
src/elem/XSelNum.c
src/elem/XSelNum.h
src/elem/XSlider.c
src/elem/XSlider.h
src/elem/XTextbox.c
src/elem/XTextbox.h

12 File Index

# **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

7.1 General Functions 15

#### 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 General Functions 17

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
ſ	in	nCol	RGB Color to use

# Returns

true if success, false if fail

7.1.2.10 bool gslc\_SetBkgndlmage ( gslc\_tsGui \* pGui, gslc\_tslmgRef slmgRef )

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\_lslnRect (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\_lslnWH (int16\_t nSelX, int16\_t nSelY, uint16\_t nWidth, uint16\_t nHeight)

Determine if a coordinate is inside of a width x height 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\_tslmgRef gslc\_GetImageFromSD (const char \*pFname, gslc\_teImgRefFlags eFmt)

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

• gslc\_tslmgRef gslc\_GetImageFromRam (unsigned char \*pImgBuf, gslc\_teImgRefFlags eFmt)

Create an image reference to a bitmap in SRAM.

• gslc\_tslmgRef 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 \* pnX1, int16\_t \* pnX1)

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	nΧ	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	nRect	Ptr to rectangle
111,000 prioot		i ii to rootangio

#### 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 (01000) between start and end color at which the midpoint between colors should appear. Normally set to 500 (half-way).	
in	nBlendAmt	The position (01000) 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 (01000) between start and end color at which the intermediate color should	
		appear.	
in	nBlendAmt	The position (01000) 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	а	First color
in	b	Second color

Generated by Doxygen

#### 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\_tslmgRef gslc\_GetlmageFromFile ( const char \* pFname, gslc\_telmgRefFlags 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\_tslmgRef gslc\_GetlmageFromProg ( const unsigned char \* plmgBuf, gslc\_telmgRefFlags eFmt )

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

#### **Parameters**

in	plmgBuf	Pointer to image buffer in memory
in	eFmt	Image format

#### Returns

Loaded image reference

7.2.2.11  $gslc\_tslmgRef gslc\_GetlmageFromRam ( unsigned char * plmgBuf, gslc\_telmgRefFlags eFmt )$ 

Create an image reference to a bitmap in SRAM.

#### **Parameters**

in	plmgBuf	Pointer to image buffer in memory
in	eFmt	Image format

# Returns

Loaded image reference

7.2.2.12  $gslc\_tslmgRef gslc\_GetlmageFromSD ( const char * pFname, gslc\_telmgRefFlags eFmt )$ 

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 bool gslc\_lslnRect ( 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**

ir	า	nSelX	X coordinate to test
ir	n.	nSelY	X coordinate to test
ir	n	rRect	Rectangular region to compare against

# Returns

true if inside region, false otherwise

7.2.2.14 bool gslc\_lslnWH ( 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

7.2.2.15 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

7.2.2.16 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.

# 7.3 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\_DrawFillRect (gslc\_tsGui \*pGui, gslc\_tsRect rRect, gslc\_tsColor nCol)

Draw a filled 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\_DrawFillTriangle ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nY0, int16\_t nX1, int16\_t nY1, int16\_t nX1, int16\_t

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

7.3.2.5 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

7.3.2.6 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

7.3.2.7 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

7.3.2.8 void gslc\_DrawFrameTriangle ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nY0, int16\_t nX1, int16\_

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

7.3.2.9 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.

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

7.3.2.10 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	nΧ	X coordinate of line startpoint
in	nΥ	Y coordinate of line startpoint
in	nW	Width of line (in +X direction)
in	nCol	Color RGB value for the line

#### Returns

none

7.3.2.11 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	nΧ	X coordinate of line startpoint
in	nΥ	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

7.3.2.12 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	nΧ	X coordinate of line startpoint
in	nΥ	Y coordinate of line startpoint
in	nΗ	Height of line (in +Y direction)
in	nCol	Color RGB value for the line

#### Returns

none

7.3.2.13 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	nΧ	Pixel X coordinate to set
in	nΥ	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).

gslc\_tsFont \* gslc\_FontGet (gslc\_tsGui \*pGui, int16\_t nFontId)

Fetch a font from its ID value.

## 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).

### **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.

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

7.4 Font Functions 33

A pointer to the font structure or NULL if error

# 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\_PageFindElemByld (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.

in	pGui	Pointer to GUI

7.5 Page Functions 35

#### 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

#### **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\_PageFindElemByld ( 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⊷	Page ID to search
	Pageld	
in	n⊷	Element ID to search
	ElemId	

## 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 nPageld, bool bModal )

Show a popup dialog.

• Popup dialogs use the overlay layer in the page stack

7.5 Page Functions 37

## **Parameters**

in	pGui	Pointer to GUI
in	n⊷	Page ID to use as the popup dialog
	Pageld	
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⊷	Page ID to select (or GSLC_PAGE_NONE to disable)
	Pageld	

## 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⊷	Page ID to select
	Pageld	

### 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⊷	Page ID to select (or GSLC_PAGE_NONE to disable)
	Pageld	

## Returns

none

7.5.2.11 void gslc\_SetStackPage ( gslc\_tsGui \* pGui, uint8\_t nStackPos, int16\_t nPageld )

Assign a page to the page stack.

#### **Parameters**

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

## Returns

none

 $7.5.2.12 \quad \text{void gslc\_SetStackState (} \quad \text{gslc\_tsGui} * \textit{pGui}, \text{ uint8\_t } \textit{nStackPos}, \text{ bool } \textit{bActive}, \text{ bool } \textit{bDoDraw} \text{ )}$ 

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 (0GSLC_STACKMAX-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

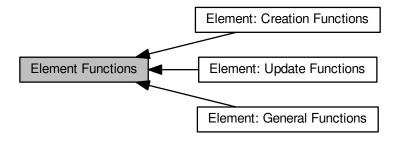
none

7.6 Element Functions 39

# 7.6 Element Functions

Functions that are used to create and manipulate elements.

Collaboration diagram for Element 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⊷	Element ID to assign (016383 or GSLC_ID_AUTO to autogen)
	ElemId	
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\_ElemCreateBtnlmg ( 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 (016383 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

in	pGui	Pointer to GUI	
in nElemId Element ID to assign (016383 or GSLC_ID_AUTO to autogen)		Element ID to assign (016383 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 ← Element ID to assign (016383 or GSLC_ID_AUTO to		Element ID to assign (016383 or GSLC_ID_AUTO to autogen)	
	ElemId		
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

in	pGui	Pointer to GUI	
in	n⊷	Element ID to assign (016383 or GSLC_ID_AUTO to autogen)	
	ElemId		

## **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 (016383 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\_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 ID.

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\_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  $gslc_teRedrawType gslc_ElemGetRedraw ( <math>gslc_tsGui*pGui, gslc_tsElemRef*pElemRef)$ 

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 bool gslc\_ElemGetVisible (  $gslc_tsGui*pGui, gslc_tsElemRef*pElemRef$  )

Get the visibility status for an element.

in	pGui	Pointer to GUI
in	pElemRef	Pointer to Element reference

#### Returns

True if element is shown, false if hidden

7.9.2.6 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	nΥ	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

7.9.2.7 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

7.9.2.8 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	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

7.9.2.9 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

7.9.2.10 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()

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

#### Returns

none

7.9.2.11 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

7.9.2.12 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

7.9.2.13 void gslc\_ElemSetGlowCol ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, gslc\_tsColor colFrameGlow, gslc\_tsColor colTxtGlow )

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

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

7.9.2.14 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

7.9.2.15 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

7.9.2.16 void gslc\_ElemSetRedraw (  $gslc_tsGui*pGui, gslc_tsElemRef*pElemRef, gslc_teRedrawType eRedraw$  )

Update the need-redraw status for an element.

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

#### Returns

none

7.9.2.17 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

7.9.2.18 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

7.9.2.19 void gslc\_ElemSetTxtAlign (  $gslc_tsGui*pGui, gslc_tsElemRef*pElemRef, unsigned nAlign$  )

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

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

7.9.2.20 void gslc\_ElemSetTxtCol ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, gslc\_tsColor colVal )

Update the text string color associated with an Element ID.

#### **Parameters**

iı	n	pGui	Pointer to GUI
iı	n	pElemRef	Pointer to Element reference
i	n	colVal	RGB color to change to

## Returns

none

 $7.9.2.21 \quad \text{void gslc\_ElemSetTxtEnc ( } \textbf{gslc\_tsGui} * \textbf{pGui, } \textbf{gslc\_tsElemRef} * \textbf{pElemRef, } \textbf{gslc\_teTxtFlags } \textbf{eFlags )}$ 

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

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

#### Returns

none

7.9.2.22 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

7.9.2.23 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

7.9.2.24 void gslc\_ElemSetTxtStr (  $gslc\_tsGui*pGui, gslc\_tsElemRef*pElemRef, const char*pStr$  )

Update the text string associated with an Element ID.

# Parameters

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

#### Returns

none

7.9.2.25 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

7.9.2.26 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\_lnitTouch (  $gslc_tsGui * pGui$ , const char \* acDev )

Initialize the touchscreen device driver.

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 va		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
ĺ	in	bEn	Enable touchscreen remapping?

## Returns

none

7.10.3.5 void gslc\_SetTouchRemapYX ( gslc\_tsGui \* pGui, bool bSwap )

Configure touchscreen XY swap.

i	n	pGui	Pointer to GUI
i	n	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)

## 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_DEBUG_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_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
```

7.12.2.2 #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.

- $\bullet \ \ \text{\#define } \underline{\text{gslc\_ElemCreateLine\_P(pGui, nElemId, nPage, nX0, nY0, nX1, nY1, colFill)}\\$ 
  - 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.

in	pGui	Pointer to GUI
in	nElemId	Unique element ID to assign
in <i>nPage</i>		Page ID to attach element to
in	nΧ	X coordinate of element
in	nΥ	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**

	- ·	B
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	nΥ	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⇔	Unique element ID to assign
	ElemId	
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	nΧ	X coordinate of element
in	nΥ	Y coordinate of element
in	nW	Width of element
in	nΗ	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 )

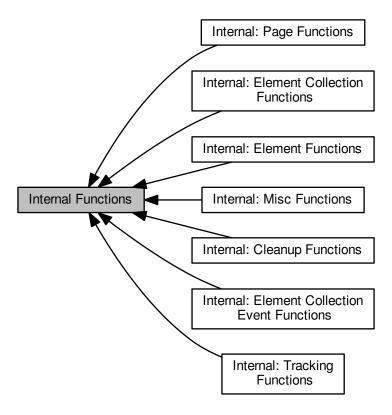
Create a read-write text element (element in Flash, string in RAM)

in	pGui	Pointer to GUI
in	nElemId	Unique element ID to assign
in	nPage	Page ID to attach element to
in	nΧ	X coordinate of element
in	nΥ	Y coordinate of element
in	nW	Width of element
in	nΗ	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:



# **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::nld

Font ID specified by user.

gslc\_teFontRefType gslc\_tsFont::eFontRefType

Font reference type.

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\_tslmgRef::plmgBuf

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\_tslmgRef::pvlmgRaw

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::nld

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\_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.

gslc\_tsElem gslc\_tsGui::sElemTmpProg

Temporary element for Flash compatibility.

gslc\_telnitStat gslc\_tsGui::elnitStatTouch

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.

• 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.

 $7.14.2.5 \quad gslc\_tsElemRef* \\ gslc\_tsCollect::asElemRef$ 

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::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.

7.14.2.11 bool gslc\_tsGui::bScreenNeedFlip

Screen requires a page flip.

7.14.2.12 bool gslc\_tsGui::bScreenNeedRedraw

Screen requires a redraw.

7.14.2.13 bool gslc\_tsGui::bTouchRemapEn

Enable touch remapping?

7.14.2.14 bool gslc\_tsGui::bTouchRemapYX

Enable touch controller swapping of X & Y.

7.14.2.15 gslc\_tsColor gslc\_tsElem::colElemFill

Color for background fill.

7.14.2.16 gslc\_tsColor gslc\_tsElem::colElemFillGlow

Color to use for fill when glowing.

7.14.2.17 gslc\_tsColor gslc\_tsElem::colElemFrame Color for frame. 7.14.2.18 gslc\_tsColor gslc\_tsElem::colElemFrameGlow Color to use for frame when glowing. 7.14.2.19 gslc\_tsColor gslc\_tsElem::colElemText Color of overlay text. 7.14.2.20 gslc\_tsColor gslc\_tsElem::colElemTextGlow Color of overlay text when glowing. 7.14.2.21 gslc\_teAction gslc\_tslnputMap::eAction Resulting action. 7.14.2.22 gslc\_teElemRefFlags gslc\_tsElemRef::eElemFlags Element reference flags. 7.14.2.23 gslc\_teInputRawEvent gslc\_tsInputMap::eEvent The input event. 7.14.2.24 gslc\_teFontRefType gslc\_tsFont::eFontRefType Font reference type. 7.14.2.25 gslc\_telmgRefFlags gslc\_tslmgRef::elmgFlags Image reference flags. 7.14.2.26 gslc\_telnitStat gslc\_tsGui::elnitStatTouch Status of touch initialization.

7.14.2.27 gslc\_teTouch gslc\_tsEventTouch::eTouch Touch state. 7.14.2.28 int8\_t gslc\_tsElem::eTxtAlign Alignment of overlay text. 7.14.2.29 gslc\_teTxtFlags gslc\_tsElem::eTxtFlags Flags associated with text buffer. 7.14.2.30 gslc\_teEventType gslc\_tsEvent::eType Event type. 7.14.2.31 uint8\_t gslc\_tsColor::g RGB green value. 7.14.2.32 uint16\_t gslc\_tsRect::h Height of region. 7.14.2.33 int16\_t gslc\_tsInputMap::nActionVal The value for the output action. 7.14.2.34 uint16\_t gslc\_tsGui::nDisp0H Height of the display (pixels) in native orientation. 7.14.2.35 uint16\_t gslc\_tsGui::nDisp0W Width of the display (pixels) in native orientation. 7.14.2.36 uint8\_t gslc\_tsGui::nDispDepth Bit depth of display (bits per pixel)

7.14.2.37 uint16\_t gslc\_tsGui::nDispH Height of the display (pixels) 7.14.2.38 uint16\_t gslc\_tsGui::nDispW Width of the display (pixels) 7.14.2.39 int16\_t gslc\_tsCollect::nElemAutoIdNext Next Element ID for auto-assignment. 7.14.2.40 uint16\_t gslc\_tsCollect::nElemCnt Number of elements allocated. 7.14.2.41 int16\_t gslc\_tsCollect::nElemIndFocused Element index currently in focus (eg. by keyboard/pin control), GSLC\_IND\_NONE for none. 7.14.2.42 uint16\_t gslc\_tsCollect::nElemMax Maximum number of elements to allocate (in RAM) 7.14.2.43 uint16\_t gslc\_tsCollect::nElemRefCnt Number of element references allocated. 7.14.2.44 uint16\_t gslc\_tsCollect::nElemRefMax Maximum number of element references to allocate. 7.14.2.45 uint8\_t gslc\_tsElem::nFeatures Element feature vector (appearance/behavior)) 7.14.2.46 uint8\_t gslc\_tsGui::nFlipX

Adafruit GFX Touch Flip x axis.

7.14.2.47 uint8\_t gslc\_tsGui::nFlipY Adafruit GFX Touch Flip x axis. 7.14.2.48 uint8\_t gslc\_tsGui::nFontCnt Number of fonts allocated. 7.14.2.49 uint8\_t gslc\_tsGui::nFontMax Maximum number of fonts to allocate. 7.14.2.50 uint8\_t gslc\_tsGui::nFrameRateCnt Diagnostic frame rate count. 7.14.2.51 uint8\_t gslc\_tsGui::nFrameRateStart Diagnostic frame rate timestamp. 7.14.2.52 int16\_t gslc\_tsElem::nGroup Group ID that the element belongs to. 7.14.2.53 int16\_t gslc\_tsFont::nld Font ID specified by user. 7.14.2.54 int16\_t gslc\_tsElem::nld Element ID specified by user. 7.14.2.55 uint8\_t gslc\_tsGui::nInputMapCnt Current number of input maps. 7.14.2.56 uint8\_t gslc\_tsGui::nlnputMapMax

Maximum number of input maps.

7.14.2.57 uint8\_t gslc\_tsGui::nPageCnt Current number of pages defined. 7.14.2.58 int16\_t gslc\_tsPage::nPageId Page identifier. 7.14.2.59 uint8\_t gslc\_tsGui::nPageMax Maximum number of pages that can be defined. 7.14.2.60 uint8\_t gslc\_tsGui::nRotation Adafruit GFX Rotation of display. 7.14.2.61 uint16\_t gslc\_tsFont::nSize Font size. 7.14.2.62 uint8\_t gslc\_tsElem::nStrBufMax Size of string buffer. 7.14.2.63 uint8\_t gslc\_tsEvent::nSubType Event sub-type. 7.14.2.64 uint8\_t gslc\_tsGui::nSwapXY Adafruit GFX Touch Swap x and y axes. 7.14.2.65 uint16\_t gslc\_tsGui::nTouchCalXMax Calibration X maximum reading. 7.14.2.66 uint16\_t gslc\_tsGui::nTouchCalXMin

Calibration X minimum reading.

7.14.2.67 uint16\_t gslc\_tsGui::nTouchCalYMax

Calibration Y maximum reading.

7.14.2.68 uint16\_t gslc\_tsGui::nTouchCalYMin

Calibration Y minimum reading.

7.14.2.69 uint16\_t gslc\_tsGui::nTouchLastPress

Last touch event pressure (0=none))

7.14.2.70 int16\_t gslc\_tsGui::nTouchLastX

Last touch event X coord.

7.14.2.71 int16\_t gslc\_tsGui::nTouchLastY

Last touch event Y coord.

7.14.2.72 uint8\_t gslc\_tsGui::nTouchRotation

Touchscreen rotation offset vs display.

7.14.2.73 uint8\_t gslc\_tsElem::nTxtMargin

Margin of overlay text within rect region.

7.14.2.74 int16\_t gslc\_tsElem::nType

Element type enumeration.

7.14.2.75 int16\_t gslc\_tslnputMap::nVal

The value associated with the input event.

7.14.2.76 int16\_t gslc\_tsEventTouch::nX

Touch X coordinate (or param1)

7.14.2.77 int16\_t gslc\_tsEventTouch::nY

Touch Y coordinate (or param2)

7.14.2.78 gslc\_tsElem\* gslc\_tsElemRef::pElem

Pointer to element in memory [RAM,FLASH].

7.14.2.79 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.

7.14.2.80 gslc\_tsElemRef\* gslc\_tsCollect::pElemRefTracked

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

7.14.2.81 const char\* gslc\_tslmgRef::pFname

Pathname to input image file [FILE,SD].

7.14.2.82 GSLC\_CB\_PIN\_POLL gslc\_tsGui::pfuncPinPoll

Callback func ptr for pin polling.

7.14.2.83 GSLC\_CB\_DRAW gslc\_tsElem::pfuncXDraw

Callback func ptr for custom drawing.

7.14.2.84 GSLC\_CB\_EVENT gslc\_tsElem::pfuncXEvent

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

7.14.2.85 GSLC\_CB\_TICK gslc\_tsElem::pfuncXTick

Callback func ptr for timer/main loop tick.

7.14.2.86 GSLC\_CB\_TOUCH gslc\_tsElem::pfuncXTouch

Callback func ptr for touch.

7.14.2.87 const unsigned char\* gslc\_tslmgRef::plmgBuf

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

7.14.2.88 char\* gslc\_tsElem::pStrBuf

Ptr to text string buffer to overlay.

7.14.2.89 gslc\_tsFont\* gslc\_tsElem::pTxtFont

Ptr to Font for overlay text.

7.14.2.90 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.

7.14.2.91 void\* gslc\_tsGui::pvDriver

Driver-specific members (gslc\_tsDriver\*)

7.14.2.92 const void\* gslc\_tsFont::pvFont

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

7.14.2.93 void\* gslc\_tslmgRef::pvlmgRaw

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

7.14.2.94 void\* gslc\_tsEvent::pvScope

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

7.14.2.95 void\* gslc\_tsElem::pXData

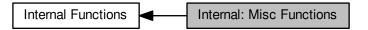
Ptr to extended data structure.

7.14.2.96 uint8\_t gslc\_tsColor::r RGB red value. 7.14.2.97 gslc\_tsRect gslc\_tsElem::rElem Rect region containing element. 7.14.2.98 gslc\_tsCollect gslc\_tsPage::sCollect Collection of elements on page. 7.14.2.99 gslc\_tsElem gslc\_tsGui::sElemTmpProg Temporary element for Flash compatibility. 7.14.2.100 gslc\_tslmgRef gslc\_tsGui::slmgRefBkgnd Image reference for background. 7.14.2.101 gslc\_tslmgRef gslc\_tsElem::slmgRefGlow Image reference to draw (glowing) 7.14.2.102 gslc\_tslmgRef gslc\_tsElem::slmgRefNorm Image reference to draw (normal) 7.14.2.103 uint16\_t gslc\_tsRect::w Width of region. 7.14.2.104 int16\_t gslc\_tsRect::x X coordinate of corner. 7.14.2.105 int16\_t gslc\_tsPt::x X coordinate. 7.14.2.106 int16\_t gslc\_tsRect::y Y coordinate of corner. 7.14.2.107 int16\_t gslc\_tsPt::y

Y coordinate.

# 7.15 Internal: Misc Functions

Collaboration diagram for Internal: Misc Functions:



# **Functions**

• gslc\_tslmgRef gslc\_ResetImage ()

Create a blank image reference structure.

# 7.15.1 Detailed Description

## 7.15.2 Function Documentation

7.15.2.1 gslc\_tslmgRef gslc\_ResetImage ( )

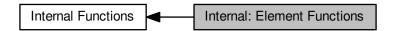
Create a blank image reference structure.

## 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.
- 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.

## 7.16.1 Detailed Description

## 7.16.2 Function Documentation

7.16.2.1 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⊷	Page ID to add element to (GSLC_PAGE_NONE to skip in case of temporary creation for
	Pageld	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

7.16.2.2 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	пТуре	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	nFontld	Font ID for textual elements

### Returns

Initialized structure

7.16.2.3 void gslc\_ElemDraw ( gslc\_tsGui \* pGui, int16\_t nPageld, 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⊷ Pageld	ID of page containing element
in	n⊷	ID of element
	Fiemid	

Generated by Doxygen

#### Returns

none

Todo Unused?

7.16.2.4 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

7.16.2.5 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

7.16.2.6  $gslc_tsElem* gslc_GetElemFromRef ( <math>gslc_tsGui* pGui, gslc_tsElemRef* pElemRef*)$ 

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

7.16.2.7 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)

7.16.2.8 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	пТуре	Expected type indicator (ie. GSLC_TYPEX_*)
in	nLineNum	Line number from calling function (ie. <b>LINE</b> )

## Returns

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

7.16.2.9 void gslc\_SetElemRefFlag ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, uint8\_t nFlagNask, uint8\_t nFlagNask)

Set the flags associated with an element reference.

# **Parameters**

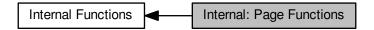
in	pGui	Pointer to GUI
in	pElemRef	Element reference pointer
in	nFlagMask	Flags to read
in	nFlagVal	Values to assign to masked flags

# Returns

none

# 7.17 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\_PageFindByld (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.

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	nΧ	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	еТуре	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\_PageFindByld ( gslc\_tsGui \* pGui, int16\_t nPageld )

Find a page in the GUI by its ID.

### **Parameters**

in	pGui	Pointer to GUI
in	n⊷	Page ID to search
	Pageld	

#### 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	рGui	Pointer to GUI
T11	paul	i diriter to dor

## 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**

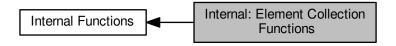
│ in │ <i>pGui</i> │ Pointer to GUI
-------------------------------------

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\_CollectFindElemByld (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

 $7.18.2.2 \quad \textbf{gslc\_tsElemRef* gslc\_CollectFindElemByld (} \quad \textbf{gslc\_tsGui} * \textit{pGui}, \\ \quad \textbf{gslc\_tsCollect} * \textit{pCollect}, \\ \quad \textbf{int16\_t} \; \textit{nElemId} \; \textbf{)}$ 

Find an element in a collection by its Element ID.

#### **Parameters**

in	pGui	Pointer to GUI
in	pCollect	Pointer to the collection
in	n⊷	Element ID to search for
	ElemId	

### 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	nΧ	Absolute X coordinate to use for search
in	nΥ	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  $gslc_tsElemRef* gslc_CollectGetElemRefTracked ( <math>gslc_tsGui* pGui, gslc_tsCollect* pCollect )$ 

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**

j	in	pGui	Pointer to GUI
j	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.
- 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**

	Pointer to the GUI	pGui	in
Generated by Do	Ptr to the element collection	pCollect	in
deficiated by Bo	Ptr to the touch event structure	pEventTouch	in

# Returns

none

 $7.19.2.3 \quad \text{void gslc\_CollectTouch ( } \textbf{gslc\_tsGui} * \textbf{pGui}, \textbf{gslc\_tsCollect} * \textbf{pCollect}, \textbf{gslc\_tsEventTouch} * \textbf{pEventTouch} )$ 

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

# 7.20 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	nΧ	X coordinate of touch event
in	nΥ	Y coordinate of touch event
in	nPress	Pressure level of touch event (0 for none, else touch)

#### Returns

none

102 Module Documentation

## 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.

104 Module Documentation

#### **Parameters**

#### 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

106 Module Documentation

# **Chapter 8**

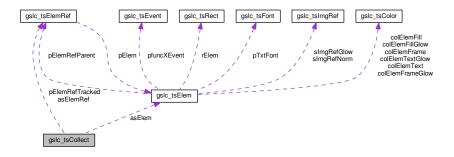
## **Data Structure Documentation**

## 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 tb

RGB blue value.

#### 8.2.1 Detailed Description

Color structure. Defines RGB triplet.

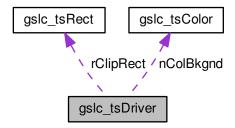
The documentation for this struct was generated from the following file:

• 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:

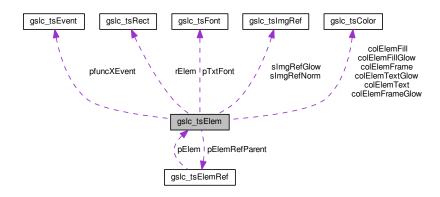
- src/GUIslice\_drv\_adagfx.h
- src/GUIslice\_drv\_m5stack.h
- src/GUIslice\_drv\_tft\_espi.h

## 8.4 gslc\_tsElem Struct Reference

#### Element Struct.

#include <GUIslice.h>

Collaboration diagram for gslc\_tsElem:



#### **Data Fields**

• int16\_t nld

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\_tslmgRef slmgRefNorm

Image reference to draw (normal)

gslc\_tslmgRef slmgRefGlow

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 autogenerated)
- · 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:

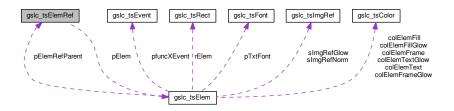
· src/GUIslice.h

## 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 nld

Font ID specified by user.

• gslc\_teFontRefType eFontRefType

Font reference type.

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:

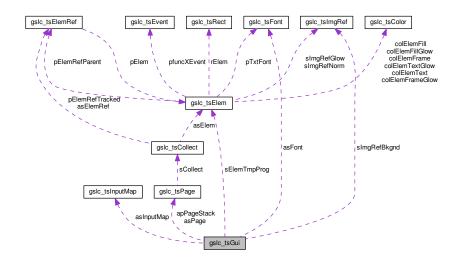
· src/GUIslice.h

## 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.

gslc\_tsElem sElemTmpProg

Temporary element for Flash compatibility.

gslc\_telnitStat elnitStatTouch

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\_tslmgRef slmgRefBkgnd

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.

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\_tslmgRef Struct Reference

Image reference structure.

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

#### **Data Fields**

· const unsigned char \* plmgBuf

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:

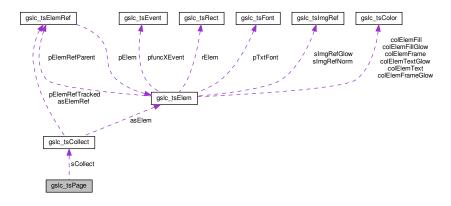
• src/GUIslice.h

## 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.

#### 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:

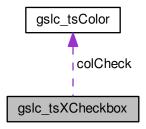
• src/GUIslice.h

## 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:

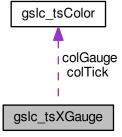
• src/elem/XCheckbox.h

## 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 blndicFill

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::blndicFill

Fill the indicator if true.

Generated by Doxygen

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:

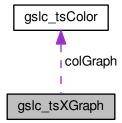
• src/elem/XGauge.h

## 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.

Visible window height.

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

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:

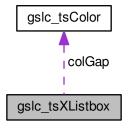
• src/elem/XGraph.h

## 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 nBufltemsMax

Max size of buffer containing items.

• uint16\_t nBufltemsPos

Current buffer position.

int16\_t nltemCnt

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 nltemW

Width of listbox item.

• int16\_t nltemH

Height of listbox item.

· int8\_t nltemGap

Gap between listbox items.

gslc\_tsColor colGap

Gap color.

int8\_t nltemMarginW

Text margin inside listbox items (X offset)

· int8\_t nltemMarginH

Text margin inside listbox items (Y offset)

bool bltemAutoSizeW

Enable auto-sizing of items (in width)

bool bltemAutoSizeH

Enable auto-sizing of items (in height)

• int16\_t nltemCurSel

Currently selected item (XLISTBOX\_SEL\_NONE for none)

int16\_t nltemCurSelLast

Old selected item to redraw (XLISTBOX\_SEL\_NONE for none)

int16\_t nltemSavedSel

Persistent selected item (ie. saved selection)

int16\_t nltemTop

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::bltemAutoSizeH

Enable auto-sizing of items (in height)

8.18.2.2 bool gslc\_tsXListbox::bltemAutoSizeW

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::nBufltemsMax Max size of buffer containing items. 8.18.2.6 uint16\_t gslc\_tsXListbox::nBufltemsPos Current buffer position. 8.18.2.7 int8\_t gslc\_tsXListbox::nCols Number of columns. 8.18.2.8 int16\_t gslc\_tsXListbox::nltemCnt Number of items in the list. 8.18.2.9 int16\_t gslc\_tsXListbox::nltemCurSel Currently selected item (XLISTBOX\_SEL\_NONE for none) 8.18.2.10 int16\_t gslc\_tsXListbox::nltemCurSelLast Old selected item to redraw (XLISTBOX\_SEL\_NONE for none) 8.18.2.11 int8\_t gslc\_tsXListbox::nltemGap Gap between listbox items. 8.18.2.12 int16\_t gslc\_tsXListbox::nltemH

Height of listbox item.

8.18.2.13 int8\_t gslc\_tsXListbox::nltemMarginH Text margin inside listbox items (Y offset) 8.18.2.14 int8\_t gslc\_tsXListbox::nltemMarginW Text margin inside listbox items (X offset) 8.18.2.15 int16\_t gslc\_tsXListbox::nltemSavedSel Persistent selected item (ie. saved selection) 8.18.2.16 int16\_t gslc\_tsXListbox::nltemTop Item to show at top of list after scrolling (0 is default) 8.18.2.17 int16\_t gslc\_tsXListbox::nltemW 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::pBufltems 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:

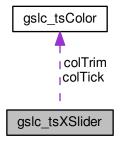
• src/elem/XListbox.h

## 8.19 gslc\_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.

#### 8.19.1 Detailed Description

Extended data for Slider element.

#### 8.19.2 Field Documentation

8.19.2.1 bool gslc\_tsXSlider::bTrim

Style: show a trim color.

8.19.2.2 bool gslc\_tsXSlider::bVert

Orientation: true if vertical, else horizontal.

8.19.2.3 gslc\_tsColor gslc\_tsXSlider::colTick

Style: color of ticks.

8.19.2.4 gslc\_tsColor gslc\_tsXSlider::colTrim

Style: color of trim.

8.19.2.5 int16\_t gslc\_tsXSlider::nPos

Current position value of the slider.

8.19.2.6 int16\_t gslc\_tsXSlider::nPosMax

Maximum position value of the slider.

8.19.2.7 int16\_t gslc\_tsXSlider::nPosMin

Minimum position value of the slider.

8.19.2.8 int16\_t gslc\_tsXSlider::nThumbSz

Size of the thumb control.

8.19.2.9 uint16\_t gslc\_tsXSlider::nTickDiv

Style: number of tickmark divisions (0 for none)

8.19.2.10 int16\_t gslc\_tsXSlider::nTickLen

Style: length of tickmarks.

#### 8.19.2.11 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

## 8.20 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)

# 8.20.1 Detailed Description Extended data for Textbox element.

#### 8.20.2 Field Documentation

8.20.2.1 bool gslc\_tsXTextbox::bScrollEn

Enable for scrollbar.

8.20.2.2 bool gslc\_tsXTextbox::bWrapEn

Enable for line wrapping.

8.20.2.3 uint16\_t gslc\_tsXTextbox::nBufCols

Number of columns in buffer.

8.20.2.4 uint8\_t gslc\_tsXTextbox::nBufPosX

Buffer X position.

8.20.2.5 uint8\_t gslc\_tsXTextbox::nBufPosY

Buffer Y position.

8.20.2.6 uint16\_t gslc\_tsXTextbox::nBufRows

Number of rows in buffer.

8.20.2.7 uint8\_t gslc\_tsXTextbox::nChSizeX

Width of characters (pixels)

8.20.2.8 uint8\_t gslc\_tsXTextbox::nChSizeY

Height of characters (pixels)

8.20.2.9 uint8\_t gslc\_tsXTextbox::nCurPosX

Cursor X position.

```
8.20.2.10 uint8_t gslc_tsXTextbox::nCurPosY
Cursor Y position.
8.20.2.11 int8_t gslc_tsXTextbox::nMarginX
Margin for text area within element rect (X)
8.20.2.12 int8_t gslc_tsXTextbox::nMarginY
Margin for text area within element rect (Y)
8.20.2.13 int16_t gslc_tsXTextbox::nRedrawRow
Specific row to update in redraw (if not -1)
8.20.2.14 uint16_t gslc_tsXTextbox::nScrollPos
Current scrollbar position.
8.20.2.15 uint8_t gslc_tsXTextbox::nWndCols
Window X size.
8.20.2.16 uint8_t gslc_tsXTextbox::nWndRows
Window Y size.
8.20.2.17 uint8_t gslc_tsXTextbox::nWndRowStart
First row of current window.
8.20.2.18 char* gslc_tsXTextbox::pBuf
Ptr to the text buffer (circular buffer))
The documentation for this struct was generated from the following file:
```

Generated by Doxygen

• src/elem/XTextbox.h

#### 8.21 THPoint Class Reference

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

#### **Public Member Functions**

- THPoint (void)
- THPoint (int16\_t x, int16\_t y, int16\_t z)
- bool operator== (THPoint)
- bool operator!= (THPoint)

#### **Data Fields**

- int16\_t x
- int16\_t y
- int16\_t z

#### 8.21.1 Constructor & Destructor Documentation

```
8.21.1.1 THPoint::THPoint (void)
```

8.21.1.2 THPoint::THPoint ( int16\_t x, int16\_t y, int16\_t z )

#### 8.21.2 Member Function Documentation

```
8.21.2.1 bool THPoint::operator!= ( THPoint p1 )
```

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

#### 8.21.3 Field Documentation

8.21.3.1 int16\_t THPoint::x

8.21.3.2 int16\_t THPoint::y

8.21.3.3 int16 t THPoint::z

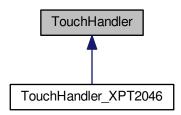
The documentation for this class was generated from the following files:

- src/GUIslice\_th.h
- src/GUIslice\_th.cpp

## 8.22 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 pln)
- virtual void begin (void)
- virtual THPoint getPoint (void)

## 8.22.1 Constructor & Destructor Documentation

**8.22.1.1 TouchHandler::TouchHandler( )** [inline]

## 8.22.2 Member Function Documentation

**8.22.2.1** void TouchHandler::begin (void ) [virtual]

Reimplemented in TouchHandler\_XPT2046.

**8.22.2.2 THPoint TouchHandler::getPoint (void )** [virtual]

Reimplemented in TouchHandler\_XPT2046.

- 8.22.2.3 THPoint TouchHandler::scale (THPoint pln)
- 8.22.2.4 void TouchHandler::setCalibration ( uint16\_t ts\_xMin, uint16\_t ts\_xMax, uint16\_t ts\_yMin, uint16\_t ts\_yMax )
- 8.22.2.5 void TouchHandler::setSize ( uint16\_t \_disp\_xSize, uint16\_t \_disp\_ySize )
- 8.22.2.6 void TouchHandler::setSwapFlip ( bool \_swapXY, bool \_flipX, bool \_flipY )

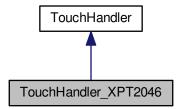
The documentation for this class was generated from the following files:

- · src/GUIslice\_th.h
- src/GUIslice\_th.cpp

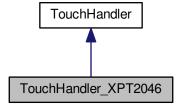
## 8.23 TouchHandler\_XPT2046 Class Reference

```
#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

## 8.23.1 Constructor & Destructor Documentation

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

## 8.23.2 Member Function Documentation

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

Reimplemented from TouchHandler.

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

Reimplemented from TouchHandler.

## 8.23.3 Field Documentation

8.23.3.1 SPIClass TouchHandler\_XPT2046::spi

8.23.3.2 XPT2046\_touch TouchHandler\_XPT2046::touchDriver

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

• src/GUIslice\_th\_XPT2046.h

# **Chapter 9**

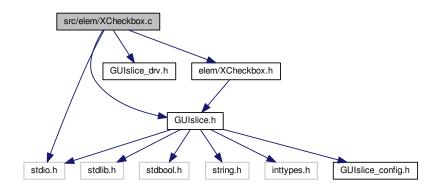
# **File Documentation**

## 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 (016383 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 \* pvElemRef, gslc\_teRedrawType eRedraw )

Draw a Checkbox element on the screen.

• Called from gslc\_ElemDraw()

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)	
in	pvElemRef	Void ptr to Element reference (typecast to gslc_tsElemRef*)	
in	n 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	pGui	Pointer to GUI
in	n⊷	Group ID to search
	GroupId	

## 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.

in	pGui	Pointer to GUI
in	pElemRef	Pointer to Element reference
in <b>bChecked</b>		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**

i	n	pGui	Pointer to GUI
i	n	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()

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

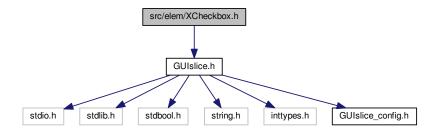
true if success, false otherwise

## 9.2.2 Variable Documentation

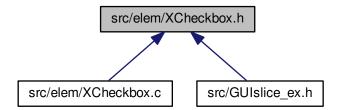
- 9.2.2.1 const char ERRSTR\_NULL
- 9.2.2.2 const char GSLC\_PMEM ERRSTR\_PXD\_NULL[]

## 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 nSelld, bool bChecked)

Callback function for checkbox/radio element state change.

#### **Enumerations**

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.

in	pGui	Pointer to GUI
in	nElemId	Unique element ID to assign
in	nPage	Page ID to attach element to
in	nΧ	X coordinate of element
in	nΥ	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⇔	Color for inner fill when checked
	Check_	
in	b⇔	Default state
	Checked←	
	_	

## 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 nSelld, bool bChecked)

Callback function for checkbox/radio element state change.

- nSelld: 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.

in	pGui	Pointer to GUI	
in	nElemId	Element ID to assign (016383 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⊷	Group ID to search
	GroupId	

## 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.

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

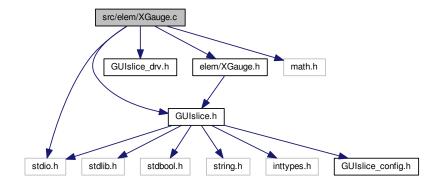
## Returns

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.

- $\bullet \ \ bool\ gslc\_ElemXGaugeDraw\ (void\ *pvGui,\ void\ *pvElemRef,\ gslc\_teRedrawType\ eRedraw)\\$ 
  - 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 <a href="mailto:ssc-ElemXGaugeSetStyle">gslc\_ElemXGaugeSetStyle</a>())</a>

#### **Parameters**

in	pGui	Pointer to GUI	
in	nElemId	nElemId Element ID to assign (016383 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	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()

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.

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 <i>pElemRef</i>		Pointer to Element reference
in	пТуре	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.

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
ĺ	in	nVal	New value to show in gauge

#### Returns

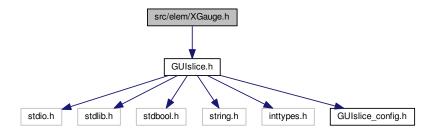
none

## 9.4.2 Variable Documentation

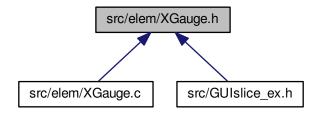
- 9.4.2.1 const char GSLC\_PMEM ERRSTR\_NULL[]
- $9.4.2.2 \quad const \ char \ \textbf{GSLC\_PMEM} \ ERRSTR\_PXD\_NULL[\,]$

# 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.

 $\bullet \ \ bool\ gslc\_ElemXGaugeDraw\ (void\ *pvGui,\ void\ *pvElemRef,\ gslc\_teRedrawType\ eRedraw)\\$ 

Draw a gauge element on the screen.

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	nΧ	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⊷	Color for the gauge frame
	Frame_	
in	colFill_	Color for the gauge background fill
in	col⊷	Color for the gauge indicator
	Gauge_	
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 <a href="mailto:gslc\_ElemXGaugeSetStyle">gslc\_ElemXGaugeSetStyle</a>())</a>

#### **Parameters**

in	pGui	Pointer to GUI
in	nElemId	Element ID to assign (016383 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()

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.

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 <i>pGui</i>		Pointer to GUI
in	pElemRef	Pointer to Element reference
in <i>nType</i>		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.

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

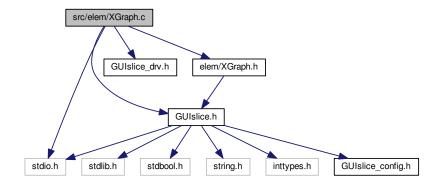
## Returns

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.

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.

in	pGui	Pointer to GUI
in	nElemId	Element ID to assign (016383 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	ontId 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*)
ſ	in	pvElemRef	Void ptr to Element reference (typecast to gslc_tsElemRef*)
ĺ	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
ĺ	in	pElemRef	Pointer to Element reference
Ī	in	nScrollPos	New scroll position
Ī	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

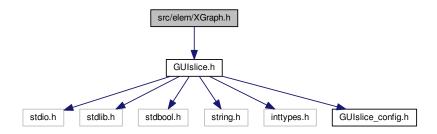
9.6.2.1 const char GSLC\_PMEM ERRSTR\_NULL[]

9.6.2.2 const char GSLC\_PMEM ERRSTR\_PXD\_NULL[]

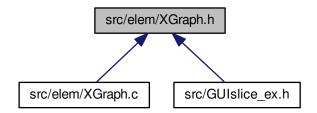
# 9.7 src/elem/XGraph.h File Reference

#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  $\leftrightarrow$  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.

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.

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 (016383 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.

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

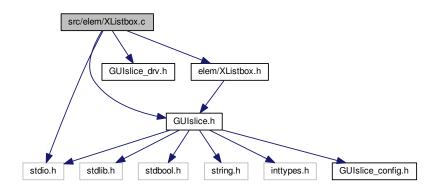
none

## 9.8 src/elem/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 nltemCurSel)

Set a Listbox element's current selection.

- bool gslc\_ElemXListboxSetScrollPos (gslc\_tsGui \*pGui, gslc\_tsElemRef \*pElemRef, uint16\_t nScrollPos) Set the Listbox scroll position.

Assign the selection callback function for a Listbox.

## **Variables**

- const char GSLC\_PMEM ERRSTR\_NULL []
- const char GSLC\_PMEM ERRSTR\_PXD\_NULL []

## 9.8.1 Macro Definition Documentation

9.8.1.1 #define XLISTBOX\_MAX\_STR

## 9.8.2 Function Documentation

9.8.2.1 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)

9.8.2.2 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 (016383 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	nFontld	Font ID for item display
in	pBufItems	Pointer to buffer that will contain list of items
in	nBufltemsMax	Max size of buffer for list of items (pBufItems)
in	nSelDefault	Default item to select

## Returns

Pointer to Element reference or NULL if failure

9.8.2.3 bool gslc\_ElemXListboxDraw ( 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

9.8.2.4 bool gslc\_ElemXListboxGetItem ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, int16\_t nItemCurSel, char \* pStrItem, uint8\_t nStrItemLen )

Get the indexed listbox item.

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)

9.8.2.5 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

9.8.2.6 int16\_t gslc\_ElemXListboxGetSel (  $gslc_tsGui*pGui, gslc_tsElemRef*pElemRef$  )

Get a Listbox element's current selection.

### **Parameters**

in	pGui	Pointer to GUI
in <i>pElemRef</i>		Pointer to Element reference

#### Returns

Current Listbox selection (or -1 if none)

9.8.2.7 void gslc\_ElemXListboxItemsSetGap ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, int8\_t nGap, gslc\_tsColor colGap )

Configure the gap between listbox items.

### **Parameters**

i	n	pGui	Pointer to GUI
i	n	pElemRef	Ptr to Element Reference to update
i	n	nGap	Set the gap between listbox items (0 for none)
i	n	colGap	Set the color of the gap between listbox items

# Returns

none

9.8.2.8 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	nltemW	Set the width of a listbox item (or -1 to auto-size)
in	nltemH	Set the height of a listbox item

# Returns

none

9.8.2.9 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

9.8.2.10 bool gslc\_ElemXListboxRecalcSize ( gslc\_tsXListbox \* pListbox, gslc\_tsRect rElem )

9.8.2.11 void gslc\_ElemXListboxReset ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef )

Empty the listbox of all items.

in <i>pGui</i>		Pointer to GUI	
in	pElemRef	Ptr to Element Reference to update	

Returns

none

9.8.2.12 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

 $9.8.2.13 \quad bool\ gslc\_ElemXListboxSetScrollPos\ (\ gslc\_tsGui*pGui,\ gslc\_tsElemRef*pElemRef,\ uint16\_t\ nScrollPos\ )$ 

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

 $9.8.2.14 \quad bool\ gslc\_ElemXListboxSetSel\ (\ gslc\_tsGui*pGui,\ gslc\_tsElemRef*pElemRef,\ int16\_t\ nltemCurSel\ )$ 

Set a Listbox element's current selection.

in	pGui	Pointer to GUI
in	pElemRef	Pointer to Element reference
in	nltemCurSel	Listbox item to select (or -1 for none)

### Returns

true if success, false if fail

9.8.2.15 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

9.8.2.16 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**

iı	n <i>pGui</i>	Pointer to GUI
iı	pElemRef	Ptr to Element Reference to update
iı	nRows	Number of rows (>= 1, or XLISTBOX_SIZE_AUTO to base on content)
ir	n nCols	Number of columns (>= 1)

#### Returns

none

9.8.2.17 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()

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

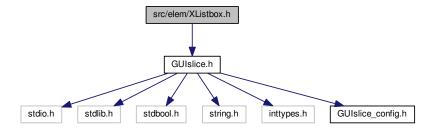
# 9.8.3 Variable Documentation

9.8.3.1 const char GSLC\_PMEM ERRSTR\_NULL[]

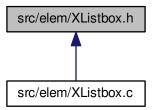
9.8.3.2 const char GSLC\_PMEM ERRSTR\_PXD\_NULL[]

# 9.9 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**

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 nltemCurSel)
  - 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.

# 9.9.1 Macro Definition Documentation

- 9.9.1.1 #define GSLC\_TYPEX\_LISTBOX
- 9.9.1.2 #define XLISTBOX\_BUF\_OH\_R
- 9.9.1.3 #define XLISTBOX\_SEL\_NONE
- 9.9.1.4 #define XLISTBOX\_SIZE\_AUTO

# 9.9.2 Typedef Documentation

9.9.2.1 typedef bool(\* GSLC\_CB\_XLISTBOX\_SEL) (void \*pvGui, void \*pvElem, int16\_t nSel)

Callback function for Listbox feedback.

# 9.9.3 Function Documentation

9.9.3.1 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)

9.9.3.2 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.

in	pGui	Pointer to GUI
in	nElemId	Element ID to assign (016383 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	nFontld	Font ID for item display

### **Parameters**

	in	pBufItems	Pointer to buffer that will contain list of items
	in	nBufltemsMax	Max size of buffer for list of items (pBufItems)
Ī	in	nSelDefault	Default item to select

# Returns

Pointer to Element reference or NULL if failure

9.9.3.3 bool gslc\_ElemXListboxDraw ( 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

9.9.3.4 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	nltemCurSel	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)

9.9.3.5 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

9.9.3.6 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)

9.9.3.7 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

9.9.3.8 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	nltemW	Set the width of a listbox item (or -1 to auto-size)
in	nltemH	Set the height of a listbox item

# Returns

none

9.9.3.9 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

9.9.3.10 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

9.9.3.11 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

9.9.3.12 bool gslc\_ElemXListboxSetScrollPos ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, uint16\_t nScrollPos )

Set the Listbox scroll position.

### **Parameters**

ſ	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

9.9.3.13 bool gslc\_ElemXListboxSetSel (  $gslc\_tsGui*pGui, gslc\_tsElemRef*pElemRef, int16\_t nltemCurSel$  )

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

9.9.3.14 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

9.9.3.15 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

9.9.3.16 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

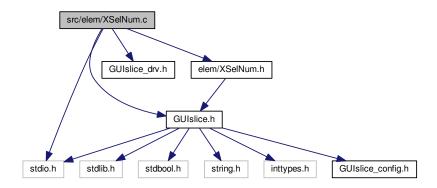
# Returns

true if success, false otherwise

# 9.10 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 []

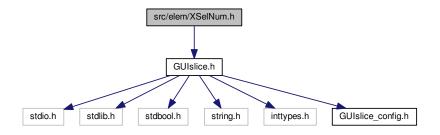
### 9.10.1 Variable Documentation

9.10.1.1 const char GSLC\_PMEM ERRSTR\_NULL[]

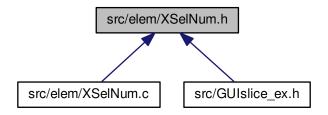
9.10.1.2 const char GSLC\_PMEM ERRSTR\_PXD\_NULL[]

# 9.11 src/elem/XSelNum.h File Reference

#include "GUIslice.h"
Include dependency graph for XSelNum.h:



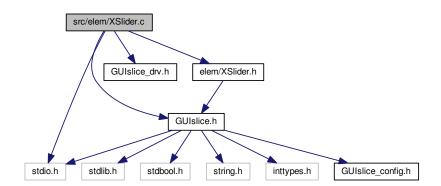
This graph shows which files directly or indirectly include this file:



# 9.12 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 []

# 9.12.1 Function Documentation

9.12.1.1 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 (016383 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

9.12.1.2 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

9.12.1.3 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

9.12.1.4 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

9.12.1.5 void gslc\_ElemXSliderSetPosFunc ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, GSLC\_CB\_XSLIDER\_POS funcCb )

Assign the position callback function for a slider.

# **Parameters**

ir	pGui	Pointer to GUI
ir	pElemRef	Pointer to Element reference
ir	funcCb	Function pointer to position routine (or NULL for none)

### Returns

none

9.12.1.6 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

9.12.1.7 bool gslc\_ElemXSliderTouch ( void \* pvGui, void \* pvElemRef, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t nRelX)

Handle touch events to Slider element.

Called from gslc\_ElemSendEventTouch()

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

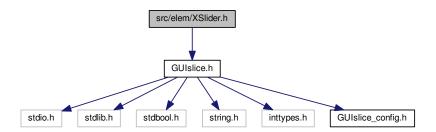
# 9.12.2 Variable Documentation

9.12.2.1 const char GSLC\_PMEM ERRSTR\_NULL[]

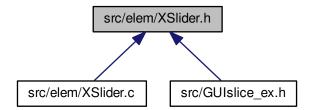
9.12.2.2 const char GSLC\_PMEM ERRSTR\_PXD\_NULL[]

# 9.13 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**

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.

# 9.13.1 Macro Definition Documentation

9.13.1.1 #define gslc\_ElemXSliderCreate\_P( pGui, nElemId, nPage, nX, nY, nW, nH, nPosMin\_, nPosMax\_, nPos\_, nThumbSz\_, bVert\_, colFrame\_, colFill\_)

Create a Slider Element in Flash.

in	pGui	Pointer to GUI
in	nElemId	Unique element ID to assign
in	nPage	Page ID to attach element to
in	nΧ	X coordinate of element
in	nΥ	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	nThumb⇔	Size of the thumb control
	Sz_	
in	bVert_	Orientation (true for vertical)
in	colFrame←	Color of the element frame
	_	
in	colFill_	Color of the element fill

# Returns

none

9.13.1.2 #define GSLC\_TYPEX\_SLIDER

# 9.13.2 Typedef Documentation

9.13.2.1 typedef bool(\* GSLC\_CB\_XSLIDER\_POS) (void \*pvGui, void \*pvElem, int16\_t nPos)

Callback function for slider feedback.

# 9.13.3 Function Documentation

9.13.3.1 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.

in	pGui	Pointer to GUI
in	nElemId	Element ID to assign (016383 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

9.13.3.2 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

9.13.3.3 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

9.13.3.4 void gslc\_ElemXSliderSetPos (  $gslc_tsGui*pGui, gslc_tsElemRef*pElemRef, int16_t nPos*)$ 

Set a Slider element's current position.

in	pGui	Pointer to GUI
in	pGui	Pointer to GUI
in	pElemRef	Pointer to Element reference
in	nPos	New position value

Returns

none

9.13.3.5 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

9.13.3.6 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

9.13.3.7 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

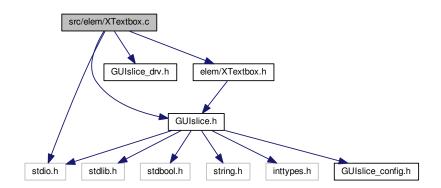
#### Returns

true if success, false otherwise

# 9.14 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 []

### 9.14.1 Function Documentation

9.14.1.1 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
ĺ	in	pElemRef	Pointer to Element reference
Ī	in	pTxt	Pointer to text string (null-terminated)

### Returns

none

- 9.14.1.2 void gslc\_ElemXTextboxBufAdd ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, unsigned char chNew, bool bAdvance )
- 9.14.1.3 void gslc\_ElemXTextboxColReset (  $gslc\_tsGui*pGui, gslc\_tsElemRef*pElemRef$  )

Insert a color reset code into the current buffer position.

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

#### Returns

none

9.14.1.4 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

9.14.1.5 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 (016383 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

9.14.1.6 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

9.14.1.7 void gslc\_ElemXTextboxLineWrAdv ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef )

9.14.1.8 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
ſ	in	pElemRef	Pointer to Element reference

# Returns

none

9.14.1.9 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.

in	pGui	Pointer to GUI
in	pElemRef	Pointer to Element reference
in	nScrollPos	New scroll position
in	nScrollMax	Maximum scroll position

### Returns

none

9.14.1.10 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

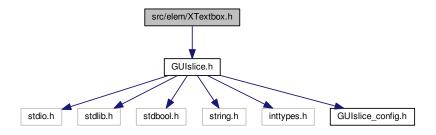
# 9.14.2 Variable Documentation

9.14.2.1 const char GSLC\_PMEM ERRSTR\_NULL[]

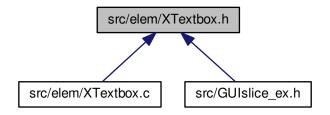
9.14.2.2 const char GSLC\_PMEM ERRSTR\_PXD\_NULL[]

# 9.15 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.

 $\bullet \ \ void \ gslc\_ElemXTextboxAdd \ (gslc\_tsGui \ *pGui, \ gslc\_tsElemRef \ *pElemRef, \ char \ *pTxt)\\$ 

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.

# 9.15.1 Macro Definition Documentation

9.15.1.1 #define GSLC\_TYPEX\_TEXTBOX

9.15.1.2 #define GSLC\_XTEXTBOX\_CODE\_COL\_RESET

9.15.1.3 #define GSLC\_XTEXTBOX\_CODE\_COL\_SET

Definitions for textbox special inline codes.

9.15.1.4 #define XTEXTBOX\_REDRAW\_ALL

9.15.1.5 #define XTEXTBOX\_REDRAW\_NONE

# 9.15.2 Function Documentation

9.15.2.1 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

9.15.2.2 void gslc\_ElemXTextboxColReset ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef )

Insert a color reset code into the current buffer position.

	in	pGui	Pointer to GUI
ſ	in <i>pElemRef</i>		Pointer to Element reference

#### Returns

none

9.15.2.3 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

9.15.2.4 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 (016383 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

9.15.2.5 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

9.15.2.6 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

9.15.2.7 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.

in	pGui	Pointer to GUI
in	pElemRef	Pointer to Element reference
in	nScrollPos	New scroll position
in	nScrollMax	Maximum scroll position

#### Returns

none

9.15.2.8 void gslc\_ElemXTextboxWrapSet ( gslc\_tsGui \* pGui, gslc\_tsElemRef \* pElemRef, bool bWrapEn )

Enable or disable line wrap within textbox.

#### **Parameters**

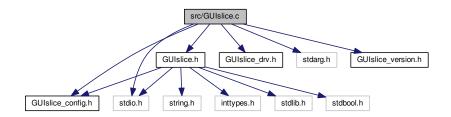
in	pGui	Pointer to GUI
in	pElemRef	Pointer to Element reference
in	bWrapEn	Enable line wrap if true

#### Returns

none

# 9.16 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\_DEBUG\_PRINT\_NORM, GSLC\_DEBUG\_PRINT\_TOKEN, GSLC\_DEBUG\_PRINT\_UINT16, GSL
 C\_DEBUG\_PRINT\_STR,
 GSLC\_DEBUG\_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)
- 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\_lslnRect (int16\_t nSelX, int16\_t nSelY, gslc\_tsRect rRect)

Determine if a coordinate is inside of a rectangular region.

• bool gslc\_lslnWH (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\_tslmgRef gslc\_ResetImage ()

Create a blank image reference structure.

• gslc tslmgRef 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 tslmgRef gslc GetlmageFromRam (unsigned char \*plmgBuf, gslc telmgRefFlags eFmt)

Create an image reference to a bitmap in SRAM.

gslc\_tslmgRef 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\_DrawFillRect (gslc\_tsGui \*pGui, gslc\_tsRect rRect, gslc\_tsColor nCol)

Draw a filled 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\_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\_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\_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.

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.

 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.

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\_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 ID.

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\_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.

 $\bullet \ \ void \ gslc\_ElemSetTickFunc \ (gslc\_tsGui \ *pGui, \ gslc\_tsElemRef \ *pElemRef, \ GSLC\_CB\_TICK \ funcCb)$ 

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 []

## 9.16.1 Enumeration Type Documentation

9.16.1.1 enum gslc\_teDebugPrintState

### Enumerator

```
GSLC_DEBUG_PRINT_NORM
GSLC_DEBUG_PRINT_TOKEN
GSLC_DEBUG_PRINT_UINT16
GSLC_DEBUG_PRINT_STR
GSLC_DEBUG_PRINT_STR_P
```

# 9.16.2 Function Documentation

```
9.16.2.1 void gslc_OrderCoord ( int16_t * pnX0, int16_t * pnY0, int16_t * pnX1, int16_t * pnX1)
```

9.16.2.2 void gslc\_SwapCoords ( int16\_t \* pnXa, int16\_t \* pnYa, int16\_t \* pnXb, int16\_t \* pnYb )

# 9.16.3 Variable Documentation

9.16.3.1 const char ERRSTR\_NULL[]

9.16.3.2 const char GSLC\_PMEM ERRSTR\_PXD\_NULL[]

9.16.3.3 GSLC\_CB\_DEBUG\_OUT g\_pfDebugOut

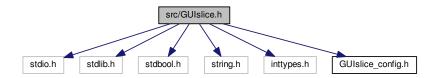
Global debug output function.

The user assigns this function via gslc\_InitDebug()

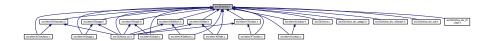
## 9.16.3.4 uint16\_t m\_nLUTSinF0X16

# 9.17 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 tslmgRef

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 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

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\_DATA

• #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\_DEBUG\_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 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 }
    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 telnitStat { 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 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\_telmgRefFlags {

 ${\sf GSLC\_IMGREF\_NONE, GSLC\_IMGREF\_SRC\_FILE, GSLC\_IMGREF\_SRC\_SD, GSLC\_IMGREF\_SRC\_} \leftrightarrow {\sf RAM},$ 

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 {

 $\label{eq:gslc_txt_mem_ram, gslc_txt_mem_prog, gslc_txt_alloc_none, gslc_txt_alloc_int, gslc_txt_alloc_ext, gslc_txt_enc_plain, gslc_txt_enc_utf8, gslc_txt_mem, gslc_txt_alloc, gslc_txt_enc, gslc_txt_default \}$ 

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\_lslnRect (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.

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 tslmgRef gslc GetImageFromFile (const char \*pFname, gslc teImgRefFlags eFmt)

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

gslc\_tslmgRef gslc\_GetImageFromSD (const char \*pFname, gslc\_teImgRefFlags eFmt)

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

gslc\_tslmgRef gslc\_GetImageFromRam (unsigned char \*pImgBuf, gslc\_teImgRefFlags eFmt)

Create an image reference to a bitmap in SRAM.

gslc\_tslmgRef 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.

 $\bullet \ \ void \ gslc\_DrawLine \ (gslc\_tsGui \ *pGui, int16\_t \ nX0, int16\_t \ nY0, int16\_t \ nX1, int16\_t \ nY1, gslc\_tsColor \ nCol)$ 

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\_DrawFillRect (gslc\_tsGui \*pGui, gslc\_tsRect rRect, gslc\_tsColor nCol)

Draw a filled 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).

gslc\_tsFont \* gslc\_FontGet (gslc\_tsGui \*pGui, int16\_t nFontId)

Fetch a font from its ID value.

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\_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 ID.

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\_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)
- gslc\_tslmgRef 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.

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 nPageld, int16 t nElemId)

Draw an element to the active display.

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\_PageFindByld (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.

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.

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

- 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.

## 9.17.1 Macro Definition Documentation

9.17.1.1 #define GSLC\_2PI

9.17.1.2 #define GSLC\_ALIGN\_BOT\_LEFT

Align to bottom-left.

9.17.1.3 #define GSLC\_ALIGN\_BOT\_MID Align to middle of bottom. 9.17.1.4 #define GSLC\_ALIGN\_BOT\_RIGHT Align to bottom-right. 9.17.1.5 #define GSLC\_ALIGN\_MID\_LEFT Align to middle of left side. 9.17.1.6 #define GSLC\_ALIGN\_MID\_MID Align to center. 9.17.1.7 #define GSLC\_ALIGN\_MID\_RIGHT Align to middle of right side. 9.17.1.8 #define GSLC\_ALIGN\_TOP\_LEFT Align to top-left. 9.17.1.9 #define GSLC\_ALIGN\_TOP\_MID Align to middle of top. 9.17.1.10 #define GSLC\_ALIGN\_TOP\_RIGHT Align to top-right. 9.17.1.11 #define GSLC\_ALIGNH\_LEFT Horizontal align to left. 9.17.1.12 #define GSLC\_ALIGNH\_MID

Horizontal align to middle.

9.17.1.13 #define GSLC_ALIGNH_RIGHT		
Horizontal align to right.		
9.17.1.14 #define GSLC_ALIGNV_BOT		
Vertical align to bottom.		
9.17.1.15 #define GSLC_ALIGNV_MID		
Vertical align to middle.		
9.17.1.16 #define GSLC_ALIGNV_TOP		
Element text alignment.		
Vertical align to top		
9.17.1.17 #define GSLC_COL_BLACK		
Black.		
9.17.1.18 #define GSLC_COL_BLUE		
Blue.		
9.17.1.19 #define GSLC_COL_BLUE_DK1		
Blue (dark1)		
9.17.1.20 #define GSLC_COL_BLUE_DK2		
Blue (dark2)		
9.17.1.21 #define GSLC_COL_BLUE_DK3		
Blue (dark3)		
9.17.1.22 #define GSLC_COL_BLUE_DK4		
Blue (dark4)		

```
9.17.1.23 #define GSLC_COL_BLUE_LT1
Blue (light1)
9.17.1.24 #define GSLC_COL_BLUE_LT2
Blue (light2)
9.17.1.25 #define GSLC_COL_BLUE_LT3
Blue (light3)
9.17.1.26 #define GSLC_COL_BLUE_LT4
Blue (light4)
9.17.1.27 #define GSLC_COL_BROWN
Brown.
9.17.1.28 #define GSLC_COL_CYAN
Cyan.
9.17.1.29 #define GSLC_COL_GRAY
Gray.
9.17.1.30 #define GSLC_COL_GRAY_DK1
Gray (dark)
9.17.1.31 #define GSLC_COL_GRAY_DK2
Gray (dark)
9.17.1.32 #define GSLC_COL_GRAY_DK3
Gray (dark)
```

```
9.17.1.33 #define GSLC_COL_GRAY_LT1
Gray (light1)
9.17.1.34 #define GSLC_COL_GRAY_LT2
Gray (light2)
9.17.1.35 #define GSLC_COL_GRAY_LT3
Gray (light3)
9.17.1.36 #define GSLC_COL_GREEN
Green.
9.17.1.37 #define GSLC_COL_GREEN_DK1
Green (dark1)
9.17.1.38 #define GSLC_COL_GREEN_DK2
Green (dark2)
9.17.1.39 #define GSLC_COL_GREEN_DK3
Green (dark3)
9.17.1.40 #define GSLC_COL_GREEN_DK4
Green (dark4)
9.17.1.41 #define GSLC_COL_GREEN_LT1
Green (light1)
9.17.1.42 #define GSLC_COL_GREEN_LT2
Green (light2)
```

```
9.17.1.43 #define GSLC_COL_GREEN_LT3
Green (light3)
9.17.1.44 #define GSLC_COL_GREEN_LT4
Green (light4)
9.17.1.45 #define GSLC_COL_MAGENTA
Magenta.
9.17.1.46 #define GSLC_COL_ORANGE
Orange.
9.17.1.47 #define GSLC_COL_PURPLE
Purple.
9.17.1.48 #define GSLC_COL_RED
Red.
9.17.1.49 #define GSLC_COL_RED_DK1
Red (dark1)
9.17.1.50 #define GSLC_COL_RED_DK2
Red (dark2)
9.17.1.51 #define GSLC_COL_RED_DK3
Red (dark3)
9.17.1.52 #define GSLC_COL_RED_DK4
Basic color definition.
Red (dark4)
```

9.17.1.53	#define GSLC_COL_RED_LT1	
Red (light1)		
9.17.1.54	#define GSLC_COL_RED_LT2	
Red (light2)		
9.17.1.55	#define GSLC_COL_RED_LT3	
Red (light3)		
9.17.1.56	#define GSLC_COL_RED_LT4	
Red (light4)		
9.17.1.57	#define GSLC_COL_TEAL	
Teal.		
9.17.1.58	#define GSLC_COL_WHITE	
White.		
9.17.1.59	#define GSLC_COL_YELLOW	
Yellow.		
9.17.1.60	#define GSLC_COL_YELLOW_DK	
Yellow (dark)		
9.17.1.61	#define GSLC_COLMONO_BLACK	
Black.		
9.17.1.62	#define GSLC_COLMONO_WHITE	
White.		

9.17.1.63 #define GSLC\_ELEM\_FEA\_CLICK\_EN Element accepts touch presses. 9.17.1.64 #define GSLC\_ELEM\_FEA\_FILL\_EN Element is drawn with a fill. 9.17.1.65 #define GSLC\_ELEM\_FEA\_FRAME\_EN Element is drawn with a frame. 9.17.1.66 #define GSLC\_ELEM\_FEA\_GLOW\_EN Element supports glowing state. 9.17.1.67 #define GSLC\_ELEM\_FEA\_NONE Element default (no features set)) 9.17.1.68 #define GSLC\_ELEM\_FEA\_VALID Element features type. Element record is valid 9.17.1.69 #define GSLC\_ELEMREF\_DEFAULT Define the default element reference flags for new elements. 9.17.1.70 #define GSLC\_PMEM 9.17.2 Typedef Documentation 9.17.2.1 typedef int16\_t(\* GSLC\_CB\_DEBUG\_OUT) (char ch) 9.17.2.2 typedef bool(\* GSLC\_CB\_DRAW) (void \*pvGui, void \*pvElemRef, gsIc\_teRedrawType eRedraw)

Callback function for element drawing.

9.17.2.3 typedef bool(\* GSLC\_CB\_EVENT) (void \*pvGui, gslc\_tsEvent sEvent)

Callback function for element drawing.

9.17.2.4 typedef bool(\* GSLC\_CB\_PIN\_POLL) (void \*pvGui, int16\_t \*pnPinInd, int16\_t \*pnPinVal)

Callback function for pin polling.

9.17.2.5 typedef bool(\* GSLC\_CB\_TICK) (void \*pvGui, void \*pvElemRef)

Callback function for element tick.

9.17.2.6 typedef bool(\* GSLC\_CB\_TOUCH) (void \*pvGui, void \*pvElemRef, gslc\_teTouch eTouch, int16\_t nX, int16\_t nY)

Callback function for element touch tracking.

9.17.2.7 typedef struct gslc\_tsColor gslc\_tsColor

Color structure. Defines RGB triplet.

9.17.2.8 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 autogenerated)
- 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.

9.17.2.9 typedef struct gslc\_tsEvent gslc\_tsEvent

Event structure.

9.17.2.10 typedef struct gslc\_tsEventTouch gslc\_tsEventTouch

Structure used to pass touch data through event.

9.17.2.11 typedef struct gslc\_tsPt gslc\_tsPt

Define point coordinates.

9.17.2.12 typedef struct gslc\_tsRect gslc\_tsRect

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

# 9.17.3 Enumeration Type Documentation

9.17.3.1 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.

# 9.17.3.2 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)

9.17.3.3 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.

9.17.3.4 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.

9.17.3.5 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.

## 9.17.3.6 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.
```

## 9.17.3.7 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.
```

## 9.17.3.8 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.
```

## 9.17.3.9 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.
```

## 9.17.3.10 enum gslc\_telmgRefFlags

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.

# 9.17.3.11 enum gslc\_telnitStat

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.

# 9.17.3.12 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)

## 9.17.3.13 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.
```

```
9.17.3.14 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_D_LONG Button D (short press)

GSLC_PIN_BTN_E Button E (short press)
```

**GSLC\_PIN\_BTN\_E\_LONG** Button E (long press)

9.17.3.15 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.
```

9.17.3.16 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.

9.17.3.17 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)

 $\label{eq:GSLC_TOUCH_SUBTYPE_MASK} \textbf{Mask for subtype}.$ 

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.

#### 9.17.3.18 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

# 9.17.3.19 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.

# 9.17.4 Variable Documentation

# 9.17.4.1 GSLC\_CB\_DEBUG\_OUT g\_pfDebugOut

Global debug output function.

The user assigns this function via gslc\_InitDebug()

# 9.18 src/GUIslice\_config.h File Reference

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



# 9.19 src/GUIslice\_config\_ard.h File Reference

### **Macros**

- #define DRV DISP ADAGFX
- #define DRV\_TOUCH\_ADA\_STMPE610
- #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 ADATOUCH I2C HW
- #define ADATOUCH\_SPI\_HW
- #define ADATOUCH\_SPI\_SW
- #define ADATOUCH\_I2C\_ADDR
- #define ADATOUCH PIN CS
- #define ADATOUCH\_X\_MIN
- #define ADATOUCH Y MIN
- #define ADATOUCH\_X\_MAX
- #define ADATOUCH Y MAX
- #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
- 9.19.1 Macro Definition Documentation
- 9.19.1.1 #define ADAGFX\_PIN\_CLK
- 9.19.1.2 #define ADAGFX\_PIN\_CS
- 9.19.1.3 #define ADAGFX\_PIN\_DC
- 9.19.1.4 #define ADAGFX\_PIN\_MISO
- 9.19.1.5 #define ADAGFX\_PIN\_MOSI
- 9.19.1.6 #define ADAGFX\_PIN\_RD
- 9.19.1.7 #define ADAGFX\_PIN\_RST
- 9.19.1.8 #define ADAGFX\_PIN\_SDCS
- 9.19.1.9 #define ADAGFX\_PIN\_WR
- 9.19.1.10 #define ADAGFX\_SPI\_HW
- 9.19.1.11 #define ADATOUCH\_FLIP\_X
- 9.19.1.12 #define ADATOUCH\_FLIP\_Y
- 9.19.1.13 #define ADATOUCH\_I2C\_ADDR
- 9.19.1.14 #define ADATOUCH\_I2C\_HW
- 9.19.1.15 #define ADATOUCH\_PIN\_CS
- 9.19.1.16 #define ADATOUCH\_SPI\_HW
- 9.19.1.17 #define ADATOUCH\_SPI\_SW

9.19.1.18	#define ADATOUCH_SWAP_XY
9.19.1.19	#define ADATOUCH_X_MAX
9.19.1.20	#define ADATOUCH_X_MIN
9.19.1.21	#define ADATOUCH_Y_MAX
9.19.1.22	#define ADATOUCH_Y_MIN
9.19.1.23	#define DEBUG_ERR
9.19.1.24	#define DRV_DISP_ADAGFX
9.19.1.25	#define DRV_DISP_ADAGFX_ILI9341
9.19.1.26	#define DRV_TOUCH_ADA_STMPE610
9.19.1.27	#define GSLC_BMP_TRANS_EN
9.19.1.28	#define GSLC_BMP_TRANS_RGB
9.19.1.29	#define GSLC_CLIP_EN
9.19.1.30	#define GSLC_DEV_TOUCH
9.19.1.31	#define GSLC_FEATURE_COMPOUND
9.19.1.32	#define GSLC_FEATURE_INPUT
9.19.1.33	#define GSLC_FEATURE_XGAUGE_RADIAL
9.19.1.34	#define GSLC_FEATURE_XGAUGE_RAMP
9.19.1.35	#define GSLC_FEATURE_XTEXTBOX_EMBED
9.19.1.36	#define GSLC_LOCAL_STR
9.19.1.37	#define GSLC_LOCAL_STR_LEN
9.19.1.38	#define GSLC_ROTATE
9.19.1.39	#define GSLC_SD_BUFFPIXEL
9.19.1.40	#define GSLC_SD_EN

```
9.19.1.41 #define GSLC_TOUCH_MAX_EVT

9.19.1.42 #define GSLC_USE_FLOAT

9.19.1.43 #define GSLC_USE_PROGMEM

9.19.1.44 #define TOUCH_ROTATION_DATA

9.19.1.45 #define TOUCH_ROTATION_FLIPX( rotation )

9.19.1.46 #define TOUCH_ROTATION_FLIPY( rotation )

9.19.1.47 #define TOUCH_ROTATION_SWAPXY( rotation )
```

# 9.20 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

9.20.1	Macro Definition Documentation
9.20.1.1	#define ADATOUCH_FLIP_X
9.20.1.2	#define ADATOUCH_FLIP_Y
9.20.1.3	#define ADATOUCH_SWAP_XY
9.20.1.4	#define DEBUG_ERR
9.20.1.5	#define DRV_DISP_SDL1
9.20.1.6	#define DRV_SDL_FIX_START
9.20.1.7	#define DRV_SDL_MOUSE_SHOW
9.20.1.8	#define DRV_TOUCH_TSLIB
9.20.1.9	#define GSLC_BMP_TRANS_EN
9.20.1.10	#define GSLC_BMP_TRANS_RGB
9.20.1.11	#define GSLC_DEV_FB
9.20.1.12	#define GSLC_DEV_TOUCH
9.20.1.13	#define GSLC_DEV_VID_DRV
9.20.1.14	#define GSLC_FEATURE_COMPOUND
9.20.1.15	#define GSLC_FEATURE_INPUT
9.20.1.16	#define GSLC_FEATURE_XGAUGE_RADIAL
9.20.1.17	#define GSLC_FEATURE_XGAUGE_RAMP
9.20.1.18	#define GSLC_FEATURE_XTEXTBOX_EMBED
9.20.1.19	#define GSLC_LOCAL_STR
9.20.1.20	#define GSLC_LOCAL_STR_LEN
9.20.1.21	#define GSLC_TOUCH_MAX_EVT
9.20.1.22	#define GSLC_USE_FLOAT
9.20.1.23	#define GSLC_USE_PROGMEM

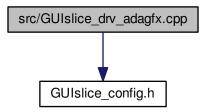
# 9.21 src/GUIslice\_drv.h File Reference

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



# 9.22 src/GUIslice\_drv\_adagfx.cpp File Reference

#include "GUIslice\_config.h"
Include dependency graph for GUIslice\_drv\_adagfx.cpp:

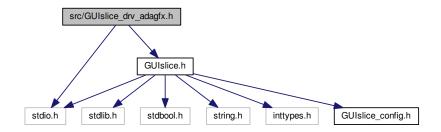


# 9.23 src/GUIslice\_drv\_adagfx.h File Reference

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 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\_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)

```
9.23.1 Detailed Description
GUIslice library (driver layer for Adafruit-GFX)
9.23.2 Macro Definition Documentation
9.23.2.1 #define DRV_HAS_DRAW_CIRCLE_FILL
Support gslc_DrvDrawFillCircle()
9.23.2.2 #define DRV_HAS_DRAW_CIRCLE_FRAME
Support gslc DrvDrawFrameCircle()
9.23.2.3 #define DRV_HAS_DRAW_LINE
Support gslc_DrvDrawLine()
```

9.23.2.4 #define DRV\_HAS\_DRAW\_POINT

Support gslc\_DrvDrawPoint()

9.23.2.5 #define DRV\_HAS\_DRAW\_POINTS

Support gslc\_DrvDrawPoints()

9.23.2.6 #define DRV\_HAS\_DRAW\_RECT\_FILL

Support gslc\_DrvDrawFillRect()

9.23.2.7 #define DRV\_HAS\_DRAW\_RECT\_FRAME

Support gslc\_DrvDrawFrameRect()

9.23.2.8 #define DRV\_HAS\_DRAW\_TEXT

Support gslc\_DrvDrawTxt()

9.23.2.9 #define DRV\_HAS\_DRAW\_TRI\_FILL

Support gslc\_DrvDrawFillTriangle()

9.23.2.10 #define DRV\_HAS\_DRAW\_TRI\_FRAME

Support gslc\_DrvDrawFrameTriangle()

9.23.2.11 #define DRV\_OVERRIDE\_TXT\_ALIGN

Driver provides text alignment.

## 9.23.3 Function Documentation

9.23.3.1 uint16\_t gslc\_DrvAdaptColorToRaw ( gslc\_tsColor nCol )

9.23.3.2 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

9.23.3.3 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

9.23.3.4 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

9.23.3.5 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

9.23.3.6 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
ĺ	in	nCol	Color RGB value to fill

#### Returns

true if success, false if error

9.23.3.7 bool gslc\_DrvDrawFillTriangle ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nX0, int16\_t nX1, int

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

9.23.3.8 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

9.23.3.9 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

9.23.3.10 bool gslc\_DrvDrawFrameTriangle ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nY0, int16\_t nX1, int16\_t nX1, 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

9.23.3.11 bool gslc\_DrvDrawlmage ( gslc\_tsGui \* pGui, int16\_t nDstX, int16\_t nDstY, gslc\_tslmgRef slmgRef )

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

9.23.3.12 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

9.23.3.13 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

9.23.3.14 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	nΧ	X coordinate of point
in	nΥ	Y coordinate of point
in	nCol	Color RGB value to draw

#### Returns

true if success, false if error

9.23.3.15 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⊷	Number of points in array
	NumPt	
in	nCol	Color RGB value to draw

#### Returns

true if success, false if error

9.23.3.16 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

 $9.23.3.17 \quad const\ void * \ gslc\_DrvFontAdd\ (\ gslc\_teFontRefType\ \textit{eFontRefType},\ const\ void * \ \textit{pvFontRef},\ uint16\_t\ \textit{nFontSz}\ )$ 

Load a font from a resource and return pointer to it.

## **Parameters**

ir	n <i>eFontRefType</i>	Font reference type (GSLC_FONTREF_PTR for Arduino)
iı	n pvFontRef	Font reference pointer (Pointer to the GFXFont array)
Gene 11	Prated by Doxygen	Typeface size to use

#### Returns

Void ptr to driver-specific font if load was successful, NULL otherwise

9.23.3.18 void gslc\_DrvFontsDestruct ( gslc\_tsGui \* pGui )

Release all fonts defined in the GUI.

#### **Parameters**

in <i>pGui</i>	Pointer to GUI
----------------	----------------

#### Returns

none

9.23.3.19 const char\* gslc\_DrvGetNameDisp ( gslc\_tsGui \* pGui )

Get the display driver name.

#### **Parameters**

in	pGui	Pointer to GUI
----	------	----------------

#### Returns

String containing driver name

9.23.3.20 const char\* gslc\_DrvGetNameTouch (  $gslc_tsGui*pGui$  )

Get the touch driver name.

### **Parameters**

in	pGui	Pointer to GUI

## Returns

String containing driver name

9.23.3.21 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

9.23.3.22 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 \* pnTxtSzW)

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

9.23.3.23 void gslc\_DrvImageDestruct ( void \* pvImg )

Release an image surface.

## **Parameters**

in	pvlmg	Void ptr to image
----	-------	-------------------

## Returns

none

9.23.3.24 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

9.23.3.25 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

9.23.3.26 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

9.23.3.27 void\* gslc\_DrvLoadlmage ( 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

9.23.3.28 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

9.23.3.29 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

9.23.3.30 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

9.23.3.31 bool gslc\_DrvSetBkgndlmage ( gslc\_tsGui \* pGui, gslc\_tsImgRef slmgRef )

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

9.23.3.32 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

 $9.23.3.33 \quad \text{bool gslc\_DrvSetElemImageGlow (} \ \ \text{gslc\_tsGui} * \textit{pGui}, \ \ \text{gslc\_tsElem} * \textit{pElem}, \ \ \text{gslc\_tsImgRef} \ \ \text{slmgRef} \ \ \text{option of the policy of the polic$ 

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

9.23.3.34 bool gslc\_DrvSetElemImageNorm (  $gslc_tsGui*pGui, gslc_tsElem*pElem, gslc_tsImgRef sImgRef$  )

Set an element's normal-state image.

## **Parameters**

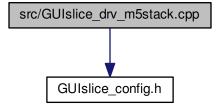
in	pGui	Pointer to GUI
in	pElem	Pointer to Element to update
in	sImgRef	Image reference

### Returns

true if success, false if error

# 9.24 src/GUIslice\_drv\_m5stack.cpp File Reference

#include "GUIslice\_config.h"
Include dependency graph for GUIslice drv m5stack.cpp:

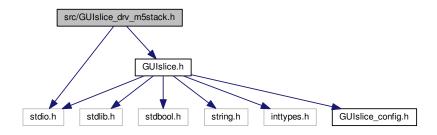


## 9.25 src/GUIslice\_drv\_m5stack.h File Reference

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\_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\_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 []

### 9.25.1 Detailed Description

GUIslice library (driver layer for M5stack)

#### 9.25.2 Macro Definition Documentation

9.25.2.1 #define DRV\_HAS\_DRAW\_CIRCLE\_FILL

Support gslc\_DrvDrawFillCircle()

9.25.2.2 #define DRV\_HAS\_DRAW\_CIRCLE\_FRAME

Support gslc\_DrvDrawFrameCircle()

```
9.25.2.3 #define DRV_HAS_DRAW_LINE
Support gslc_DrvDrawLine()
9.25.2.4 #define DRV_HAS_DRAW_POINT
Support gslc_DrvDrawPoint()
9.25.2.5 #define DRV_HAS_DRAW_POINTS
Support gslc_DrvDrawPoints()
9.25.2.6 #define DRV_HAS_DRAW_RECT_FILL
Support gslc_DrvDrawFillRect()
9.25.2.7 #define DRV_HAS_DRAW_RECT_FRAME
Support gslc_DrvDrawFrameRect()
9.25.2.8 #define DRV_HAS_DRAW_TEXT
Support gslc_DrvDrawTxt()
9.25.2.9 #define DRV_HAS_DRAW_TRI_FILL
Support gslc_DrvDrawFillTriangle()
9.25.2.10 #define DRV_HAS_DRAW_TRI_FRAME
Support gslc_DrvDrawFrameTriangle()
9.25.2.11 #define DRV_OVERRIDE_TXT_ALIGN
Driver provides text alignment.
9.25.3 Function Documentation
9.25.3.1 uint16_t gslc_DrvAdaptColorToRaw ( gslc_tsColor nCol )
9.25.3.2 void gslc_DrvDestruct ( gslc_tsGui * pGui )
Free up any members associated with the driver.
    • Eg. renderers, windows, background surfaces, etc.
```

#### **Parameters**

in <i>pGui</i>	Pointer to GUI
----------------	----------------

#### Returns

none

9.25.3.3 void gslc\_DrvDrawBkgnd ( gslc\_tsGui \* pGui )

Copy the background image to destination screen.

#### **Parameters**

in <i>pGui</i> Pointer to G	GUI
-----------------------------	-----

#### Returns

true if success, false if fail

9.25.3.4 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

9.25.3.5 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

9.25.3.6 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
ĺ	in	nCol	Color RGB value to fill

## Returns

true if success, false if error

9.25.3.7 bool gslc\_DrvDrawFillTriangle ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nX0, int16\_t nX1, int

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

9.25.3.8 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

9.25.3.9 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

9.25.3.10 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

9.25.3.11 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

9.25.3.12 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

9.25.3.13 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

9.25.3.14 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	nΧ	X coordinate of point
in	nΥ	Y coordinate of point
in	nCol	Color RGB value to draw

## Returns

true if success, false if error

9.25.3.15 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⊷	Number of points in array
	NumPt	
in	nCol	Color RGB value to draw

#### Returns

true if success, false if error

9.25.3.16 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			
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	pGui	Pointer to GUI
in pFont Ptr to Font in pStr String to display in eTxtFlags Flags associated with text string in colTxt Color to draw text	in	nTxtX	X coordinate of top-left text string
in pStr String to display in eTxtFlags Flags associated with text string in colTxt Color to draw text	in	nTxtY	Y coordinate of top-left text string
in colTxt Color to draw text	in	pFont	Ptr to Font
in colTxt Color to draw text	in	pStr	String to display
	in	eTxtFlags	Flags associated with text string
in colBg unused in m5stack, defaults to black	in	colTxt	Color to draw text
	in	colBg	unused in m5stack, defaults to black

#### Returns

true if success, false if failure

9.25.3.17 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

 $9.25.3.18 \quad const\ void*\ gslc\_DrvFontAdd\ (\ gslc\_teFontRefType\ \textit{eFontRefType},\ const\ void*\ \textit{pvFontRef},\ uint16\_t\ \textit{nFontSz}\ )$ 

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

Generated by Doxygen

#### Returns

Void ptr to driver-specific font if load was successful, NULL otherwise

9.25.3.19 void gslc\_DrvFontsDestruct ( gslc\_tsGui \* pGui )

Release all fonts defined in the GUI.

#### **Parameters**

in	pGui	Pointer to GUI
----	------	----------------

#### Returns

none

9.25.3.20 const char\* gslc\_DrvGetNameDisp ( gslc\_tsGui \* pGui )

Get the display driver name.

#### **Parameters**

in	pGui	Pointer to GUI
----	------	----------------

#### Returns

String containing driver name

9.25.3.21 const char\* gslc\_DrvGetNameTouch ( gslc\_tsGui \* pGui )

Get the touch driver name.

## Parameters

in	pGui	Pointer to GUI

#### Returns

String containing driver name

9.25.3.22 bool gslc\_DrvGetTxtSize ( gslc\_tsGui \* pGui, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, int16\_t \* pnTxtX, int16\_t \* pnTxtSzW, uint16\_t \* pnTxtSzW, uint16\_t \* pnTxtSzW)

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

9.25.3.23 void gslc\_DrvImageDestruct ( void \* pvImg )

Release an image surface.

#### **Parameters**

in	pvlmg	Void ptr to image
----	-------	-------------------

## Returns

none

9.25.3.24 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

9.25.3.25 bool gslc\_DrvInitTs ( gslc\_tsGui \* pGui, const char \* acDev )

Perform any touchscreen-specific initialization.

#### **Parameters**

	•	Pointer to GUI
in	acDev	Device path to touchscreen eg. "/dev/input/touchscreen"

#### Returns

true if successful

9.25.3.26 void\* gslc\_DrvLoadlmage (  $gslc_tsGui * pGui$ ,  $gslc_tslmgRef slmgRef$  )

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

9.25.3.27 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

9.25.3.28 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

9.25.3.29 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

9.25.3.30 bool gslc\_DrvSetBkgndlmage (  $gslc_tsGui*pGui, gslc_tslmgRef*slmgRef*)$ 

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

9.25.3.31 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

9.25.3.32 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

9.25.3.33 bool gslc\_brvSetElemImageNorm ( 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

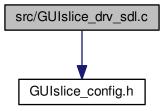
9.25.4 Variable Documentation

9.25.4.1 const char GSLC\_PMEM ERRSTR\_NULL[]

9.25.4.2 const char GSLC\_PMEM ERRSTR\_PXD\_NULL[]

## 9.26 src/GUIslice\_drv\_sdl.c File Reference

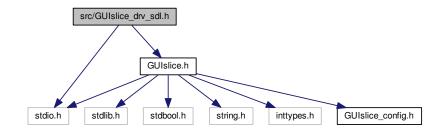
#include "GUIslice\_config.h"
Include dependency graph for GUIslice\_drv\_sdl.c:



# 9.27 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.

## 9.27.1 Detailed Description

GUIslice library (driver layer for LINUX / SDL)

## 9.27.2 Macro Definition Documentation

9.27.2.1 #define DRV\_HAS\_DRAW\_POINT

Support gslc DrvDrawPoint()

9.27.2.2 #define DRV\_OVERRIDE\_TXT\_ALIGN

Driver provides text alignment.

#### 9.27.3 Function Documentation

9.27.3.1 SDL\_Color gslc\_DrvAdaptColor ( gslc\_tsColor sCol )

Translate a gslc\_tsColor into an SDL\_Color.

#### **Parameters**

in | sCol | gslc\_tsColor

#### Returns

Converted SDL\_Color

9.27.3.2 SDL\_Rect gslc\_DrvAdaptRect ( gslc\_tsRect rRect )

Translate a gslc\_tsRect into an SDL\_Rect.

#### **Parameters**

```
in rRect gslc_tsRect
```

## Returns

Converted SDL\_Rect

9.27.3.3 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

9.27.3.4 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

9.27.3.5 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

9.27.3.6 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

9.27.3.7 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

9.27.3.8 bool gslc\_DrvDrawlmage ( gslc\_tsGui \* pGui, int16\_t nDstX, int16\_t nDstY, gslc\_tslmgRef slmgRef )

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

9.27.3.9 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

9.27.3.10 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	nΧ	X coordinate of point
in	nΥ	Y coordinate of point
in	nCol	Color RGB value to draw

## Returns

true if success, false if error

9.27.3.11 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⊷	Number of points in array
	NumPt	
in	nCol	Color RGB value to draw

## Returns

true if success, false if error

9.27.3.12 bool gslc\_DrvDrawTxt ( gslc\_tsGui \* pGui, int16\_t nTxtX, int16\_t nTxtY, gslc\_tsFont \* 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

9.27.3.13 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

9.27.3.14 void gslc\_DrvFontsDestruct ( gslc\_tsGui \* pGui )

Release all fonts defined in the GUI.

**Parameters** 

in	pGui	Pointer to GUI
----	------	----------------

Returns

none

9.27.3.15 const char\* gslc\_DrvGetNameDisp ( gslc\_tsGui \* pGui )

Get the display driver name.

**Parameters** 

in	pGui	Pointer to GUI
----	------	----------------

Returns

String containing driver name

9.27.3.16 const char\* gslc\_DrvGetNameTouch (  $gslc_tsGui*pGui$  )

Get the touch driver name.

**Parameters** 

in	pGui	Pointer to GUI
----	------	----------------

Returns

String containing driver name

9.27.3.17 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

#### **Parameters**

	out	pnPress	Ptr to Pressure level of last touch event (0 for none, 1 for touch)	
Ī	out	peInputEvent	Indication of event type	
ĺ	out	pnInputVal	Additional data for event type	

#### Returns

true if an event was detected or false otherwise

9.27.3.18 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 \* pnTxtSzW)

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

9.27.3.19 void gslc\_DrvImageDestruct ( void \* pvImg )

Release an image surface.

### **Parameters**

in	pvlmg	Void ptr to image

## Returns

none

9.27.3.20 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 <i>pGui</i> Pointer to GL	JI
------------------------------	----

#### Returns

true if success, false if fail

9.27.3.21 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

9.27.3.22 void\* gslc\_DrvLoadlmage ( gslc\_tsGui \* pGui, gslc\_tsImgRef slmgRef )

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

9.27.3.23 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

9.27.3.24 void gslc\_DrvReportInfoPost ( )

Report driver debug info (after initialization)

Returns

none

9.27.3.25 void gslc\_DrvReportInfoPre ( )

Report driver debug info (before initialization)

Returns

none

9.27.3.26 bool gslc\_DrvRotate ( gslc\_tsGui \* pGui, uint8\_t nRotation )

Change rotation, automatically adapt touchscreen axes swap/flip.

## **Parameters**

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

Returns

true if successful

9.27.3.27 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

9.27.3.28 bool gslc\_DrvSetBkgndlmage ( gslc\_tsGui \* pGui, gslc\_tsImgRef slmgRef )

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

9.27.3.29 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

9.27.3.30 bool gslc\_brvSetElemImageGlow ( 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

 $9.27.3.31 \quad bool\ gslc\_brvSetElemImageNorm\ (\ gslc\_tsGui*pGui,\ gslc\_tsElem*pElem,\ gslc\_tsImgRef\ slmgRef\ )$ 

Set an element's normal-state image.

#### **Parameters**

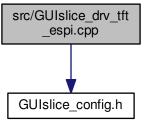
in	pGui	Pointer to GUI
in	pElem	Pointer to Element to update
in	sImgRef	Image reference

#### Returns

true if success, false if error

# 9.28 src/GUIslice\_drv\_tft\_espi.cpp File Reference

#include "GUIslice\_config.h"
Include dependency graph for GUIslice\_drv\_tft\_espi.cpp:

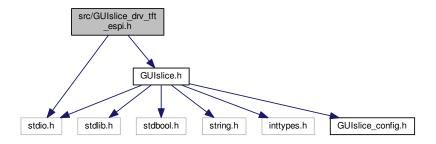


# 9.29 src/GUIslice\_drv\_tft\_espi.h File Reference

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\_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\_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)

## 9.29.1 Detailed Description

GUIslice library (driver layer for TFT-eSPI)

## 9.29.2 Macro Definition Documentation

9.29.2.1 #define DRV\_HAS\_DRAW\_CIRCLE\_FILL

Support gslc\_DrvDrawFillCircle()

9.29.2.2 #define DRV\_HAS\_DRAW\_CIRCLE\_FRAME

Support gslc\_DrvDrawFrameCircle()

9.29.2.3 #define DRV\_HAS\_DRAW\_LINE

Support gslc\_DrvDrawLine()

```
9.29.2.4 #define DRV_HAS_DRAW_POINT
Support gslc_DrvDrawPoint()
9.29.2.5 #define DRV_HAS_DRAW_POINTS
Support gslc_DrvDrawPoints()
9.29.2.6 #define DRV_HAS_DRAW_RECT_FILL
Support gslc_DrvDrawFillRect()
9.29.2.7 #define DRV_HAS_DRAW_RECT_FRAME
Support gslc_DrvDrawFrameRect()
9.29.2.8 #define DRV_HAS_DRAW_TEXT
Support gslc_DrvDrawTxt()
9.29.2.9 #define DRV_HAS_DRAW_TRI_FILL
Support gslc_DrvDrawFillTriangle()
9.29.2.10 #define DRV_HAS_DRAW_TRI_FRAME
Support gslc_DrvDrawFrameTriangle()
9.29.2.11 #define DRV_OVERRIDE_TXT_ALIGN
Driver provides text alignment.
9.29.3 Function Documentation
9.29.3.1 uint16_t gslc_DrvAdaptColorToRaw ( gslc_tsColor nCol )
9.29.3.2 void gslc_DrvDestruct ( gslc_tsGui * pGui )
Free up any members associated with the driver.
```

• Eg. renderers, windows, background surfaces, etc.

#### **Parameters**

in <i>pGui</i> Pointer to GL	JI
------------------------------	----

#### Returns

none

9.29.3.3 void gslc\_DrvDrawBkgnd ( gslc\_tsGui \* pGui )

Copy the background image to destination screen.

#### **Parameters**

in <i>pGui</i> Pointer to G	GUI
-----------------------------	-----

#### Returns

true if success, false if fail

9.29.3.4 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

9.29.3.5 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

9.29.3.6 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

9.29.3.7 bool gslc\_DrvDrawFillTriangle ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nX0, int16\_t nX1, int

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

9.29.3.8 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

9.29.3.9 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

9.29.3.10 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

9.29.3.11 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

9.29.3.12 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

9.29.3.13 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

9.29.3.14 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	nΧ	X coordinate of point
in	nΥ	Y coordinate of point
in	nCol	Color RGB value to draw

## Returns

true if success, false if error

9.29.3.15 bool gslc\_DrvDrawPoints (  $gslc_tsGui*pGui*pGui*, gslc_tsPt*asPt*, uint16_t nNumPt*, <math>gslc_tsColor* nCol$  )

Draw a point.

#### **Parameters**

in	pGui	Pointer to GUI
in	asPt	Array of points to draw
in	n⊷	Number of points in array
	NumPt	
in	nCol	Color RGB value to draw

#### Returns

true if success, false if error

9.29.3.16 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

9.29.3.17 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

 $9.29.3.18 \quad const\ void *\ gslc\_DrvFontAdd\ (\ gslc\_teFontRefType\ \textit{eFontRefType},\ const\ void *\ \textit{pvFontRef},\ uint16\_t\ \textit{nFontSz}\ )$ 

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

9.29.3.19 void gslc\_DrvFontsDestruct ( gslc\_tsGui \* pGui )

Release all fonts defined in the GUI.

#### **Parameters**

in	pGui	Pointer to GUI
----	------	----------------

## Returns

none

9.29.3.20 const char\* gslc\_DrvGetNameDisp ( gslc\_tsGui \* pGui )

Get the display driver name.

### **Parameters**

in	pGui	Pointer to GUI

## Returns

String containing driver name

9.29.3.21 const char\* gslc\_DrvGetNameTouch (  $gslc_tsGui*pGui$  )

Get the touch driver name.

## **Parameters**

in	pGui	Pointer to GUI

#### Returns

String containing driver name

9.29.3.22 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 \* pnTxtSzW)

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

9.29.3.23 void gslc\_DrvImageDestruct ( void \* pvImg )

Release an image surface.

## **Parameters**

in	pvlmg	Void ptr to image
----	-------	-------------------

#### Returns

none

9.29.3.24 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 <i>pGui</i>	Pointer to GUI
----------------	----------------

## Returns

true if success, false if fail

9.29.3.25 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"	1

#### Returns

true if successful

9.29.3.26 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

9.29.3.27 void gslc\_DrvPageFlipNow ( gslc\_tsGui \* pGui )

Force a page flip to occur.

This generally copies active screen surface to the display.

#### **Parameters**

in <i>pGui</i> Pointer to GU
------------------------------

#### Returns

none

9.29.3.28 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

9.29.3.29 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

9.29.3.30 bool gslc\_DrvSetBkgndlmage (  $gslc\_tsGui*pGui, gslc\_tsImgRef*sImgRef*)$ 

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

9.29.3.31 bool gslc\_DrvSetClipRect ( gslc\_tsGui \* pGui, gslc\_tsRect \* pRect )

Set the clipping rectangle for future drawing updates.

#### **Parameters**

i	n	pGui	Pointer to GUI
i	n	pRect	Rectangular region to constrain edits

#### Returns

true if success, false if error

9.29.3.32 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

9.29.3.33 bool gslc\_DrvSetElemImageNorm (  $gslc_tsGui*pGui, gslc_tsElem*pElem, gslc_tsImgRef sImgRef$  )

Set an element's normal-state image.

## **Parameters**

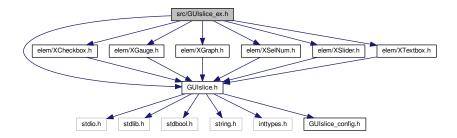
in	pGui	Pointer to GUI
in	pElem	Pointer to Element to update
in	sImgRef	Image reference

## Returns

true if success, false if error

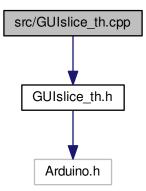
## 9.30 src/GUIslice\_ex.h File Reference

```
#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:
```



# 9.31 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

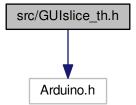
## 9.31.1 Function Documentation

- 9.31.1.1 TouchHandler\* gslc\_getTouchHandler( void )
- 9.31.1.2 void gslc\_InitTouchHandler ( TouchHandler \* pTH )
- 9.31.2 Variable Documentation
- 9.31.2.1 TouchHandler\* pTouchHandler

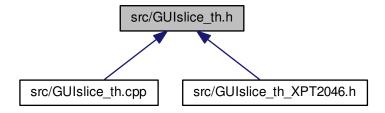
## 9.32 src/GUIslice\_th.h File Reference

#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)

#### 9.32.1 Function Documentation

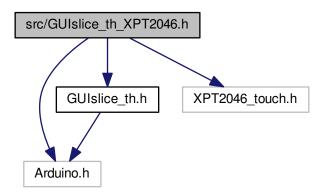
```
9.32.1.1 TouchHandler* gslc_getTouchHandler( void )
```

9.32.1.2 void gslc\_InitTouchHandler ( TouchHandler \* pTHO )

# 9.33 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:

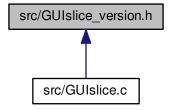


## **Data Structures**

• class TouchHandler\_XPT2046

# 9.34 src/GUIslice\_version.h File Reference

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



## Macros

• #define GUISLICE\_VER

## 9.34.1 Macro Definition Documentation

9.34.1.1 #define GUISLICE\_VER