## GUIslice 0.8.6

Generated by Doxygen 1.8.8

Fri Mar 17 2017 23:41:01

# **Contents**

1	KEA	DME												1
2	Clas	s Index	<b>C</b>											3
	2.1	Class	List				 	 	 	 		 		 3
3	File	Index												5
	3.1	File Lis	st				 	 	 	 		 		 5
4	Clas	s Docu	mentation	n										7
	4.1	gslc_ts	sCollect St	truct Refere	ence		 	 	 	 	 	 		 7
		4.1.1	Detailed	Descriptio	n		 	 	 	 	 	 		 8
		4.1.2	Member	Data Docu	ımentatio	n	 	 	 	 	 	 		 8
			4.1.2.1	asElem			 	 	 	 	 	 		 8
			4.1.2.2	asElemF	Ref		 	 	 	 	 	 		 8
			4.1.2.3	nElemAu	itoldNext		 	 	 	 	 	 		 8
			4.1.2.4	nElemCr	nt		 	 	 	 	 	 		 8
			4.1.2.5	nElemMa	ax		 	 	 	 	 	 		 8
			4.1.2.6	nElemRe	efCnt		 	 	 	 	 	 		 8
			4.1.2.7	nElemRe	efMax .		 	 	 	 	 	 		 8
			4.1.2.8	pElemTra	acked .		 	 	 	 	 	 		 8
			4.1.2.9	pfuncXE	vent		 	 	 	 	 	 		 9
	4.2	gslc_ts	sColor Stru	uct Referer	ice		 	 	 	 	 	 		 9
		4.2.1	Detailed	Descriptio	n		 	 	 	 	 	 		 9
		4.2.2	Member	Data Docu	ımentatio	n	 	 	 	 	 	 		 9
			4.2.2.1	b			 	 	 	 	 	 		 9
			4.2.2.2	g			 	 	 	 	 	 		 9
			4.2.2.3	r			 	 	 	 	 	 		 9
			4.2.2.4	unused			 	 	 	 	 	 		 9
	4.3	gslc_ts	sDriver Str	uct Refere	nce		 	 	 	 	 	 		 10
		4.3.1	Member	Data Docu	ımentatio	n	 	 	 	 	 	 		 10
			4.3.1.1	nColRaw	Bkgnd		 	 	 	 	 	 		 10
			4312	nSurfScr	een									10

iv CONTENTS

		4.3.1.3	pTsDev	10
		4.3.1.4	rClipRect	10
4.4	gslc_ts	Elem Stru	uct Reference	11
	4.4.1	Detailed	Description	12
	4.4.2	Member	Data Documentation	12
		4.4.2.1	bClickEn	12
		4.4.2.2	bFillEn	13
		4.4.2.3	bFrameEn	13
		4.4.2.4	bGlowEn	13
		4.4.2.5	bGlowing	13
		4.4.2.6	bNeedRedraw	13
		4.4.2.7	bValid	13
		4.4.2.8	colElemFill	13
		4.4.2.9	colElemFillGlow	13
		4.4.2.10	colElemFrame	13
		4.4.2.11	colElemFrameGlow	13
		4.4.2.12	colElemText	13
		4.4.2.13	colElemTextGlow	13
		4.4.2.14	eTxtAlign	14
		4.4.2.15	eTxtFlags	14
		4.4.2.16	nGroup	14
		4.4.2.17	nld	14
		4.4.2.18	nStrBufMax	14
		4.4.2.19	nTxtMargin	14
		4.4.2.20	nType	14
		4.4.2.21	pElemParent	14
		4.4.2.22	pfuncXDraw	14
		4.4.2.23	pfuncXEvent	14
		4.4.2.24	pfuncXTick	14
		4.4.2.25	pfuncXTouch	14
		4.4.2.26	pStrBuf	15
		4.4.2.27	pTxtFont	15
		4.4.2.28	pXData	15
		4.4.2.29	rElem	15
		4.4.2.30	sImgRefGlow	15
		4.4.2.31	sImgRefNorm	15
4.5	gslc_ts	ElemRef	Struct Reference	15
	4.5.1	Detailed	Description	16
	4.5.2	Member	Data Documentation	16
		4.5.2.1	eElemFlags	16

CONTENTS

		4.5.2.2	pElem	16
4.6	gslc_ts	Event Stru	uct Reference	16
	4.6.1	Detailed	Description	16
	4.6.2	Member	Data Documentation	16
		4.6.2.1	eType	16
		4.6.2.2	nSubType	17
		4.6.2.3	pvData	17
		4.6.2.4	pvScope	17
4.7	gslc_ts	EventTouc	ch Struct Reference	17
	4.7.1	Detailed	Description	17
	4.7.2	Member	Data Documentation	17
		4.7.2.1	eTouch	17
		4.7.2.2	nX	17
		4.7.2.3	nY	18
4.8	gslc_ts	Font Struc	ct Reference	18
	4.8.1	Detailed	Description	18
	4.8.2	Member	Data Documentation	18
		4.8.2.1	nld	18
		4.8.2.2	nSize	18
		4.8.2.3	pvFont	18
4.9	gslc_ts	Gui Struct	t Reference	18
	4.9.1	Detailed	Description	20
	4.9.2	Member	Data Documentation	20
		4.9.2.1	asFont	20
		4.9.2.2	asPage	20
		4.9.2.3	bRedrawPartialEn	20
		4.9.2.4	nDispDepth	20
		4.9.2.5	nDispH	20
		4.9.2.6	nDispW	20
		4.9.2.7	nFontCnt	21
		4.9.2.8	nFontMax	21
		4.9.2.9	nFrameRateCnt	21
		4.9.2.10	nFrameRateStart	21
		4.9.2.11	nPageCnt	21
		4.9.2.12	nPageMax	21
		4.9.2.13	nTouchLastPress	21
		4.9.2.14	nTouchLastX	21
		4.9.2.15	nTouchLastY	21
		4.9.2.16	pCurPage	21
		4.9.2.17	pCurPageCollect	21

vi CONTENTS

		4.9.2.18 pfuncXEvent	21
		4.9.2.19 pvDriver	22
		4.9.2.20 sElemTmp	22
		4.9.2.21 slmgRefBkgnd	22
4.10	gslc_ts	ImgRef Struct Reference	22
	4.10.1	Detailed Description	22
	4.10.2	Member Data Documentation	22
		4.10.2.1 elmgFlags	22
		4.10.2.2 pFname	22
		4.10.2.3 plmgBuf	22
		4.10.2.4 pvlmgRaw	23
4.11	gslc_ts	Page Struct Reference	23
	4.11.1	Detailed Description	23
	4.11.2	Member Data Documentation	24
		4.11.2.1 bPageNeedFlip	24
		4.11.2.2 bPageNeedRedraw	24
		4.11.2.3 nPageId	24
		4.11.2.4 pfuncXEvent	24
		4.11.2.5 sCollect	24
4.12	gslc_ts	Pt Struct Reference	24
	4.12.1	Detailed Description	24
	4.12.2	Member Data Documentation	25
		4.12.2.1 x	25
		4.12.2.2 y	25
4.13	gslc_ts	Rect Struct Reference	25
	4.13.1	Detailed Description	25
	4.13.2	Member Data Documentation	25
		4.13.2.1 h	25
		4.13.2.2 w	25
		4.13.2.3 x	25
		4.13.2.4 y	26
4.14	gslc_ts	XCheckbox Struct Reference	26
	4.14.1	Detailed Description	26
	4.14.2	Member Data Documentation	27
		4.14.2.1 bChecked	27
		4.14.2.2 bRadio	27
		4.14.2.3 colCheck	27
		4.14.2.4 nStyle	27
		4.14.2.5 pGui	27
4.15	aslc ts	XGauge Struct Reference	27

CONTENTS vii

	4.15.1	Detailed Description
	4.15.2	Member Data Documentation
		4.15.2.1 bGaugeFlip
		4.15.2.2 bGaugeVert
		4.15.2.3 colGauge
		4.15.2.4 nGaugeMax
		4.15.2.5 nGaugeMin
		4.15.2.6 nGaugeVal
4.16	gslc_ts	XSelNum Struct Reference
	4.16.1	Detailed Description
	4.16.2	Member Data Documentation
		4.16.2.1 acElemTxt
		4.16.2.2 asElem
		4.16.2.3 asElemRef
		4.16.2.4 nCounter
		4.16.2.5 sCollect
4.17	gslc_ts	XSlider Struct Reference
	4.17.1	Detailed Description
	4.17.2	Member Data Documentation
		4.17.2.1 bTrim
		4.17.2.2 bVert
		4.17.2.3 colTick
		4.17.2.4 colTrim
		4.17.2.5 nPos
		4.17.2.6 nPosMax
		4.17.2.7 nPosMin
		4.17.2.8 nThumbSz
		4.17.2.9 nTickDiv
		4.17.2.10 nTickLen
		4.17.2.11 pfuncXPos
File	Docume	entation 33
5.1		ME.md File Reference
5.2		Islice.c File Reference
	5.2.1	Macro Definition Documentation
		5.2.1.1 GUISLICE VER
	5.2.2	Function Documentation
		5.2.2.1 gslc_ClipLine
		5.2.2.2 gslc_ClipPt
		5.2.2.3 gslc_ClipRect

5

viii CONTENTS

5.2.2.4	gslc_CollectDestruct	39
5.2.2.5	gslc_CollectElemAdd	39
5.2.2.6	gslc_CollectEvent	40
5.2.2.7	gslc_CollectFindElemByld	40
5.2.2.8	gslc_CollectFindElemFromCoord	40
5.2.2.9	gslc_CollectGetElemTracked	40
5.2.2.10	gslc_CollectGetNextId	41
5.2.2.11	gslc_CollectGetRedraw	41
5.2.2.12	gslc_CollectReset	41
5.2.2.13	gslc_CollectSetElemTracked	41
5.2.2.14	gslc_CollectSetEventFunc	42
5.2.2.15	gslc_CollectSetParent	42
5.2.2.16	gslc_CollectTouch	42
5.2.2.17	gslc_DebugPrintf	43
5.2.2.18	gslc_DrawFillRect	44
5.2.2.19	gslc_DrawFrameCircle	44
5.2.2.20	gslc_DrawFrameRect	44
5.2.2.21	gslc_DrawLine	44
5.2.2.22	gslc_DrawLineH	45
5.2.2.23	gslc_DrawLineV	45
5.2.2.24	gslc_DrawSetPixel	45
5.2.2.25	gslc_ElemAdd	46
5.2.2.26	gslc_ElemCreate	46
5.2.2.27	gslc_ElemCreateBox	47
5.2.2.28	gslc_ElemCreateBtnImg	47
5.2.2.29	gslc_ElemCreateBtnTxt	47
5.2.2.30	gslc_ElemCreateImg	48
5.2.2.31	gslc_ElemCreateLine	48
5.2.2.32	gslc_ElemCreateTxt	49
5.2.2.33	gslc_ElemDestruct	49
5.2.2.34	gslc_ElemDraw	49
5.2.2.35	gslc_ElemDrawByRef	49
5.2.2.36	gslc_ElemEvent	50
5.2.2.37	gslc_ElemGetGlow	50
5.2.2.38	gslc_ElemGetGlowEn	50
5.2.2.39	gslc_ElemGetGroup	50
5.2.2.40	gslc_ElemGetId	51
5.2.2.41	gslc_ElemGetRedraw	51
5.2.2.42	gslc_ElemOwnsCoord	51
5.2.2.43	gslc_ElemSendEventTouch	51

CONTENTS

5.2.2.44	gslc_ElemSetCol	52
5.2.2.45	gslc_ElemSetDrawFunc	52
5.2.2.46	gslc_ElemSetEventFunc	52
5.2.2.47	gslc_ElemSetFillEn	53
5.2.2.48	gslc_ElemSetFrameEn	53
5.2.2.49	gslc_ElemSetGlow	53
5.2.2.50	gslc_ElemSetGlowCol	53
5.2.2.51	gslc_ElemSetGlowEn	54
5.2.2.52	gslc_ElemSetGroup	54
5.2.2.53	gslc_ElemSetImage	54
5.2.2.54	gslc_ElemSetRedraw	54
5.2.2.55	gslc_ElemSetStyleFrom	55
5.2.2.56	gslc_ElemSetTickFunc	55
5.2.2.57	gslc_ElemSetTxtAlign	56
5.2.2.58	gslc_ElemSetTxtCol	56
5.2.2.59	gslc_ElemSetTxtMargin	56
5.2.2.60	gslc_ElemSetTxtMem	57
5.2.2.61	gslc_ElemSetTxtStr	57
5.2.2.62	gslc_ElemUpdateFont	57
5.2.2.63	gslc_EventCreate	57
5.2.2.64	gslc_ExpandRect	58
5.2.2.65	gslc_FontAdd	58
5.2.2.66	gslc_FontGet	58
5.2.2.67	gslc_GetImageFromFile	59
5.2.2.68	gslc_GetImageFromProg	60
5.2.2.69	gslc_GetImageFromRam	60
5.2.2.70	gslc_GetImageFromSD	60
5.2.2.71	gslc_GetPageCur	60
5.2.2.72	gslc_GetTouch	61
5.2.2.73	gslc_GetVer	61
5.2.2.74	gslc_GuiDestruct	61
5.2.2.75	gslc_Init	61
5.2.2.76	gslc_InitDebug	62
5.2.2.77	gslc_InitTouch	62
5.2.2.78	gslc_lsInRect	62
5.2.2.79	gslc_lslnWH	63
5.2.2.80	gslc_OrderCoord	63
5.2.2.81	gslc_PageAdd	63
5.2.2.82	gslc_PageDestruct	64
5.2.2.83	gslc_PageEvent	64

X CONTENTS

	5.2.2.84	gslc_PageFindByld	64
	5.2.2.85	gslc_PageFindElemByld	64
	5.2.2.86	gslc_PageFlipGet	65
	5.2.2.87	gslc_PageFlipGo	65
	5.2.2.88	gslc_PageFlipSet	65
	5.2.2.89	gslc_PageRedrawCalc	65
	5.2.2.90	gslc_PageRedrawGet	66
	5.2.2.91	gslc_PageRedrawGo	66
	5.2.2.92	gslc_PageRedrawSet	66
	5.2.2.93	gslc_PageSetEventFunc	66
	5.2.2.94	gslc_Quit	67
	5.2.2.95	gslc_ResetElem	67
	5.2.2.96	gslc_ResetFont	67
	5.2.2.97	gslc_ResetImage	67
	5.2.2.98	gslc_SetBkgndColor	68
	5.2.2.99	gslc_SetBkgndImage	68
	5.2.2.100	gslc_SetClipRect	68
	5.2.2.101	gslc_SetPageCur	68
	5.2.2.102	gslc_TrackTouch	69
	5.2.2.103	gslc_Update	69
5.2.3	Variable I	Documentation	69
	5.2.3.1	g_pfDebugOut	69
src/GU	Ilslice.h Fil	e Reference	69
5.3.1	Macro De		79
	5.3.1.1	GSLC_ALIGN_BOT_LEFT	79
	5.3.1.2		79
	5.3.1.3		79
			79
			79
	5.3.1.6		80
			80
			80
			80
	5.3.1.10		80
	5.3.1.11		80
	5.3.1.12		80
	5.3.1.13	GSLC_ALIGNV_BOT	80
	5.3.1.13 5.3.1.14	GSLC_ALIGNV_BOT	80 80
	5.3.1.13 5.3.1.14 5.3.1.15	GSLC_ALIGNV_BOT	80
	src/GU	5.2.2.86 5.2.2.87 5.2.2.88 5.2.2.89 5.2.2.90 5.2.2.91 5.2.2.92 5.2.2.93 5.2.2.94 5.2.2.95 5.2.2.96 5.2.2.97 5.2.2.98 5.2.2.100 5.2.2.101 5.2.2.102 5.2.2.103 5.2.3.1 src/GUIslice.h Fill 5.3.1 Macro De 5.3.1.1 5.3.1.2 5.3.1.3 5.3.1.4 5.3.1.5 5.3.1.6 5.3.1.7 5.3.1.8 5.3.1.9	5.2.2.86       gslc_PageFlipGet         5.2.2.87       gslc_PageFlipGo         5.2.2.88       gslc_PageFlipSet         5.2.2.90       gslc_PageRedrawGet         5.2.2.91       gslc_PageRedrawGe         5.2.2.92       gslc_PageRedrawGet         5.2.2.93       gslc_PageRedrawSet         5.2.2.94       gslc_Quit         5.2.2.95       gslc_ResetEventFunc         5.2.2.96       gslc_ResetImage         5.2.2.97       gslc_ResetImage         5.2.2.98       gslc_SetBkgndColor         5.2.2.99       gslc_SetBkgndImage         5.2.2.100       gslc_SetClipRect         5.2.2.101       gslc_SetPageCur         5.2.2.102       gslc_TrackTouch         5.2.2.103       gslc_Update         5.2.3       Variable Documentation         5.2.3.1       g_pfDebugOut         src/GUIslice.h File Reference         5.3.1.1       GSLC_ALIGN_BOT_LEFT         5.3.1.2       GSLC_ALIGN_BOT_MID         5.3.1.3       GSLC_ALIGN_MID_LEFT         5.3.1.4       GSLC_ALIGN_MID_RIGHT         5.3.1.6       GSLC_ALIGN_MID_RIGHT         5.3.1.8       GSLC_ALIGN_TOP_LEFT         5.3.1.9       GSLC_ALIGN_TOP_RIGHT

CONTENTS xi

5.3.1.17	GSLC_COL_BLUE	80
5.3.1.18	GSLC_COL_BLUE_DK1	81
5.3.1.19	GSLC_COL_BLUE_DK2	81
5.3.1.20	GSLC_COL_BLUE_DK3	81
5.3.1.21	GSLC_COL_BLUE_DK4	81
5.3.1.22	GSLC_COL_BLUE_LT1	81
5.3.1.23	GSLC_COL_BLUE_LT2	81
5.3.1.24	GSLC_COL_BLUE_LT3	81
5.3.1.25	GSLC_COL_BLUE_LT4	81
5.3.1.26	GSLC_COL_BROWN	81
5.3.1.27	GSLC_COL_CYAN	81
5.3.1.28	GSLC_COL_GRAY	81
5.3.1.29	GSLC_COL_GRAY_DK1	81
5.3.1.30	GSLC_COL_GRAY_DK2	82
5.3.1.31	GSLC_COL_GRAY_DK3	82
5.3.1.32	GSLC_COL_GRAY_LT1	82
5.3.1.33	GSLC_COL_GRAY_LT2	82
5.3.1.34	GSLC_COL_GRAY_LT3	82
5.3.1.35	GSLC_COL_GREEN	82
5.3.1.36	GSLC_COL_GREEN_DK1	82
5.3.1.37	GSLC_COL_GREEN_DK2	82
5.3.1.38	GSLC_COL_GREEN_DK3	82
5.3.1.39	GSLC_COL_GREEN_DK4	82
5.3.1.40	GSLC_COL_GREEN_LT1	82
5.3.1.41	GSLC_COL_GREEN_LT2	82
5.3.1.42	GSLC_COL_GREEN_LT3	83
5.3.1.43	GSLC_COL_GREEN_LT4	83
5.3.1.44	GSLC_COL_MAGENTA	83
5.3.1.45	GSLC_COL_ORANGE	83
5.3.1.46	GSLC_COL_PURPLE	83
5.3.1.47	GSLC_COL_RED	83
5.3.1.48	GSLC_COL_RED_DK1	83
5.3.1.49	GSLC_COL_RED_DK2	83
5.3.1.50	GSLC_COL_RED_DK3	83
5.3.1.51	GSLC_COL_RED_DK4	83
5.3.1.52	GSLC_COL_RED_LT1	83
5.3.1.53	GSLC_COL_RED_LT2	83
5.3.1.54	GSLC_COL_RED_LT3	84
5.3.1.55	GSLC_COL_RED_LT4	84
5.3.1.56	GSLC_COL_TEAL	84

xii CONTENTS

	5.3.1.57	GSLC_COL_WHITE	84
	5.3.1.58	GSLC_COL_YELLOW	84
	5.3.1.59	GSLC_COL_YELLOW_DK	84
	5.3.1.60	GSLC_DEBUG_PRINT	84
	5.3.1.61	gslc_ElemCreateBox_P	84
	5.3.1.62	gslc_ElemCreateTxt_P	85
5.3.2	Typedef [	Documentation	86
	5.3.2.1	GSLC_CB_DEBUG_OUT	86
	5.3.2.2	GSLC_CB_DRAW	86
	5.3.2.3	GSLC_CB_EVENT	86
	5.3.2.4	GSLC_CB_TICK	86
	5.3.2.5	GSLC_CB_TOUCH	86
	5.3.2.6	gslc_tsColor	86
	5.3.2.7	gslc_tsElem	87
	5.3.2.8	gslc_tsEvent	87
	5.3.2.9	gslc_tsEventTouch	87
	5.3.2.10	gslc_tsPt	87
	5.3.2.11	gslc_tsRect	87
5.3.3	Enumera	tion Type Documentation	87
	5.3.3.1	gslc_teDebugPrintState	87
	5.3.3.2	gslc_teElemId	87
	5.3.3.3	gslc_teElemInd	88
	5.3.3.4	gslc_teElemRefFlags	88
	5.3.3.5	gslc_teEventSubType	88
	5.3.3.6	gslc_teEventType	89
	5.3.3.7	gslc_teFontId	89
	5.3.3.8	gslc_teGroupId	89
	5.3.3.9	gslc_teImgRefFlags	89
	5.3.3.10	gslc_tePageId	90
	5.3.3.11	gslc_teTouch	90
	5.3.3.12	gslc_teTxtFlags	90
	5.3.3.13	gslc_teTypeCore	91
5.3.4	Function	Documentation	91
	5.3.4.1	gslc_ClipLine	91
	5.3.4.2	gslc_ClipPt	91
	5.3.4.3	gslc_ClipRect	92
	5.3.4.4	gslc_CollectDestruct	92
	5.3.4.5	gslc_CollectElemAdd	92
	5.3.4.6	gslc_CollectEvent	92
	5.3.4.7	gslc_CollectFindElemByld	93

CONTENTS xiii

5.3.4.8	gslc_CollectFindElemFromCoord
5.3.4.9	gslc_CollectGetElemTracked
5.3.4.10	gslc_CollectGetNextId
5.3.4.11	gslc_CollectGetRedraw
5.3.4.12	gslc_CollectReset
5.3.4.13	gslc_CollectSetElemTracked
5.3.4.14	gslc_CollectSetEventFunc
5.3.4.15	gslc_CollectSetParent
5.3.4.16	gslc_CollectTouch
5.3.4.17	gslc_DebugPrintf
5.3.4.18	gslc_DrawFillRect
5.3.4.19	gslc_DrawFrameCircle
5.3.4.20	gslc_DrawFrameRect
5.3.4.21	gslc_DrawLine
5.3.4.22	gslc_DrawLineH
5.3.4.23	gslc_DrawLineV
5.3.4.24	gslc_DrawSetPixel
5.3.4.25	gslc_ElemAdd
5.3.4.26	gslc_ElemCreate
5.3.4.27	gslc_ElemCreateBox
5.3.4.28	gslc_ElemCreateBtnImg
5.3.4.29	gslc_ElemCreateBtnTxt
5.3.4.30	gslc_ElemCreateImg
5.3.4.31	gslc_ElemCreateLine
5.3.4.32	gslc_ElemCreateTxt
5.3.4.33	gslc_ElemDestruct
5.3.4.34	gslc_ElemDraw
5.3.4.35	gslc_ElemDrawByRef
5.3.4.36	gslc_ElemEvent
5.3.4.37	gslc_ElemGetGlow
5.3.4.38	gslc_ElemGetGlowEn
5.3.4.39	gslc_ElemGetGroup
5.3.4.40	gslc_ElemGetId
5.3.4.41	gslc_ElemGetRedraw
5.3.4.42	gslc_ElemOwnsCoord
5.3.4.43	gslc_ElemSendEventTouch
5.3.4.44	gslc_ElemSetCol
5.3.4.45	gslc_ElemSetDrawFunc
5.3.4.46	gslc_ElemSetEventFunc
5.3.4.47	gslc_ElemSetFillEn

XIV

5.3.4.48	gslc_ElemSetFrameEn
5.3.4.49	gslc_ElemSetGlow
5.3.4.50	gslc_ElemSetGlowCol
5.3.4.51	gslc_ElemSetGlowEn
5.3.4.52	gslc_ElemSetGroup
5.3.4.53	gslc_ElemSetImage
5.3.4.54	gslc_ElemSetRedraw
5.3.4.55	gslc_ElemSetStyleFrom
5.3.4.56	gslc_ElemSetTickFunc
5.3.4.57	gslc_ElemSetTxtAlign
5.3.4.58	gslc_ElemSetTxtCol
5.3.4.59	gslc_ElemSetTxtMargin
5.3.4.60	gslc_ElemSetTxtMem
5.3.4.61	gslc_ElemSetTxtStr
5.3.4.62	gslc_ElemUpdateFont
5.3.4.63	gslc_EventCreate
5.3.4.64	gslc_ExpandRect
5.3.4.65	gslc_FontAdd
5.3.4.66	gslc_FontGet
5.3.4.67	gslc_GetImageFromFile
5.3.4.68	gslc_GetImageFromProg
5.3.4.69	gslc_GetImageFromRam
5.3.4.70	gslc_GetImageFromSD
5.3.4.71	gslc_GetPageCur
5.3.4.72	gslc_GetTouch
5.3.4.73	gslc_GetVer
5.3.4.74	gslc_GuiDestruct
5.3.4.75	gslc_Init
5.3.4.76	gslc_InitDebug
5.3.4.77	gslc_InitTouch
5.3.4.78	gslc_lslnRect
5.3.4.79	gslc_lslnWH
5.3.4.80	gslc_PageAdd
5.3.4.81	gslc_PageDestruct
5.3.4.82	gslc_PageEvent
5.3.4.83	gslc_PageFindByld
5.3.4.84	gslc_PageFindElemByld
5.3.4.85	gslc_PageFlipGet
5.3.4.86	gslc_PageFlipGo
5.3.4.87	gslc_PageFlipSet

CONTENTS xv

		5.3.4.88	gslc_PageRedrawCalc
		5.3.4.89	gslc_PageRedrawGet
		5.3.4.90	gslc_PageRedrawGo
		5.3.4.91	gslc_PageRedrawSet
		5.3.4.92	gslc_PageSetEventFunc
		5.3.4.93	gslc_Quit
		5.3.4.94	gslc_ResetElem
		5.3.4.95	gslc_ResetFont
		5.3.4.96	gslc_ResetImage
		5.3.4.97	gslc_SetBkgndColor
		5.3.4.98	gslc_SetBkgndImage
		5.3.4.99	gslc_SetClipRect
		5.3.4.100	gslc_SetPageCur
		5.3.4.101	gslc_TrackTouch
		5.3.4.102	gslc_Update
	5.3.5	Variable I	Documentation
		5.3.5.1	g_pfDebugOut
5.4	src/GU	Islice_con	fig.h File Reference
	5.4.1	Macro De	efinition Documentation
		5.4.1.1	ADATOUCH_FLIP_X
		5.4.1.2	ADATOUCH_FLIP_Y
		5.4.1.3	ADATOUCH_SWAP_XY
		5.4.1.4	DEBUG_ERR
		5.4.1.5	DRV_DISP_SDL1
		5.4.1.6	DRV_SDL_FIX_START
		5.4.1.7	DRV_SDL_MOUSE_SHOW
		5.4.1.8	DRV_TOUCH_TSLIB
		5.4.1.9	GSLC_BMP_TRANS_EN
		5.4.1.10	GSLC_BMP_TRANS_RGB
		5.4.1.11	GSLC_DEV_FB
		5.4.1.12	GSLC_DEV_TOUCH
		5.4.1.13	GSLC_DEV_VID_DRV
		5.4.1.14	GSLC_LOCAL_STR
		5.4.1.15	GSLC_LOCAL_STR_LEN
		5.4.1.16	GSLC_USE_PROGMEM
5.5	src/GU	Islice_drv.	h File Reference
5.6	src/GU	Islice_drv_	_adagfx.cpp File Reference
	5.6.1	Function	Documentation
		5.6.1.1	gslc_DrvAdaptColorToRaw
		5.6.1.2	gslc_DrvDestruct

xvi CONTENTS

		5.6.1.3	gslc_DrvDrawBkgnd	26
		5.6.1.4	gslc_DrvDrawFillCircle	26
		5.6.1.5	gslc_DrvDrawFillRect	26
		5.6.1.6	gslc_DrvDrawFrameCircle	27
		5.6.1.7	gslc_DrvDrawFrameRect	27
		5.6.1.8	gslc_DrvDrawImage	27
		5.6.1.9	gslc_DrvDrawLine	28
		5.6.1.10	gslc_DrvDrawMonoFromMem	29
		5.6.1.11	gslc_DrvDrawPoint	29
		5.6.1.12	gslc_DrvDrawPoints	29
		5.6.1.13	gslc_DrvDrawTxt	29
		5.6.1.14	gslc_DrvFontAdd	30
		5.6.1.15	gslc_DrvFontsDestruct	30
		5.6.1.16	gslc_DrvGetTouch	30
		5.6.1.17	gslc_DrvGetTxtSize	31
		5.6.1.18	gslc_DrvImageDestruct	32
		5.6.1.19	gslc_DrvInit	32
		5.6.1.20	gslc_DrvInitTouch	32
		5.6.1.21	gslc_DrvLoadImage	33
		5.6.1.22	gslc_DrvPageFlipNow	33
		5.6.1.23	gslc_DrvSetBkgndColor	33
		5.6.1.24	gslc_DrvSetBkgndImage	33
		5.6.1.25	gslc_DrvSetClipRect	34
		5.6.1.26	gslc_DrvSetElemImageGlow	34
		5.6.1.27	gslc_DrvSetElemImageNorm	34
5.7	src/GU	Islice_drv_	_adagfx.h File Reference	34
	5.7.1	Macro De	efinition Documentation	37
		5.7.1.1	DRV_HAS_DRAW_CIRCLE_FILL	37
		5.7.1.2	DRV_HAS_DRAW_CIRCLE_FRAME	37
		5.7.1.3	DRV_HAS_DRAW_LINE	37
		5.7.1.4	DRV_HAS_DRAW_POINT	37
		5.7.1.5	DRV_HAS_DRAW_POINTS	37
		5.7.1.6	DRV_HAS_DRAW_RECT_FILL	37
		5.7.1.7	DRV_HAS_DRAW_RECT_FRAME	37
		5.7.1.8	DRV_HAS_DRAW_TEXT	37
	5.7.2	Function	Documentation	37
		5.7.2.1	gslc_DrvAdaptColorToRaw	37
		5.7.2.2	gslc_DrvDestruct	37
		5.7.2.3	gslc_DrvDrawBkgnd	38
		5.7.2.4	gslc_DrvDrawFillCircle	38

CONTENTS xvii

		5.7.2.5	gslc_DrvDrawFillRect
		5.7.2.6	gslc_DrvDrawFrameCircle
		5.7.2.7	gslc_DrvDrawFrameRect
		5.7.2.8	gslc_DrvDrawImage
		5.7.2.9	gslc_DrvDrawLine
		5.7.2.10	gslc_DrvDrawPoint
		5.7.2.11	gslc_DrvDrawPoints
		5.7.2.12	gslc_DrvDrawTxt
		5.7.2.13	gslc_DrvFontAdd
		5.7.2.14	gslc_DrvFontsDestruct
		5.7.2.15	gslc_DrvGetTouch
		5.7.2.16	gslc_DrvGetTxtSize
		5.7.2.17	gslc_DrvImageDestruct
		5.7.2.18	gslc_DrvInit
		5.7.2.19	gslc_DrvInitTouch
		5.7.2.20	gslc_DrvInitTs
		5.7.2.21	gslc_DrvLoadImage
		5.7.2.22	gslc_DrvPageFlipNow
		5.7.2.23	gslc_DrvSetBkgndColor
		5.7.2.24	gslc_DrvSetBkgndImage
		5.7.2.25	gslc_DrvSetClipRect
		5.7.2.26	gslc_DrvSetElemImageGlow
		5.7.2.27	gslc_DrvSetElemImageNorm
5.8	src/GU	Islice_drv_	sdl.c File Reference
	5.8.1	Macro De	efinition Documentation
		5.8.1.1	DRV_SDL_FIX_TTY
	5.8.2	Function	Documentation
		5.8.2.1	gslc_DrvAdaptColor
		5.8.2.2	gslc_DrvAdaptColorRaw
		5.8.2.3	gslc_DrvAdaptRect
		5.8.2.4	gslc_DrvCleanStart
		5.8.2.5	gslc_DrvDestruct
		5.8.2.6	gslc_DrvDrawBkgnd
		5.8.2.7	gslc_DrvDrawFillRect
		5.8.2.8	gslc_DrvDrawFrameRect
		5.8.2.9	gslc_DrvDrawGetPixelRaw
		5.8.2.10	gslc_DrvDrawImage
		5.8.2.11	gslc_DrvDrawLine
		5.8.2.12	gslc_DrvDrawPoint
		5.8.2.13	gslc_DrvDrawPoints

xviii CONTENTS

		5.8.2.14	gslc_DrvDrawSetPixelRaw
		5.8.2.15	gslc_DrvDrawTxt
		5.8.2.16	gslc_DrvFontAdd
		5.8.2.17	gslc_DrvFontsDestruct
		5.8.2.18	gslc_DrvGetTouch
		5.8.2.19	gslc_DrvGetTxtSize
		5.8.2.20	gslc_DrvImageDestruct
		5.8.2.21	gslc_DrvInit
		5.8.2.22	gslc_DrvInitTouch
		5.8.2.23	gslc_DrvLoadImage
		5.8.2.24	gslc_DrvPageFlipNow
		5.8.2.25	gslc_DrvPasteSurface
		5.8.2.26	gslc_DrvScreenLock
		5.8.2.27	gslc_DrvScreenUnlock
		5.8.2.28	gslc_DrvSetBkgndColor
		5.8.2.29	gslc_DrvSetBkgndImage
		5.8.2.30	gslc_DrvSetClipRect
		5.8.2.31	gslc_DrvSetElemImageGlow
		5.8.2.32	gslc_DrvSetElemImageNorm
		5.8.2.33	gslc_TDrvGetTouch
		5.8.2.34	gslc_TDrvInitTouch
5.9	src/GU	Ilslice_drv_	_sdl.h File Reference
	5.9.1	Macro De	efinition Documentation
		5.9.1.1	DRV_HAS_DRAW_CIRCLE_FILL
		5.9.1.2	DRV_HAS_DRAW_CIRCLE_FRAME
		5.9.1.3	DRV_HAS_DRAW_LINE
		5.9.1.4	DRV_HAS_DRAW_POINT 160
		5.9.1.5	DRV_HAS_DRAW_POINTS
		5.9.1.6	DRV_HAS_DRAW_RECT_FILL
		5.9.1.7	DRV_HAS_DRAW_RECT_FRAME
		5.9.1.8	DRV_HAS_DRAW_TEXT
	5.9.2	Function	Documentation
		5.9.2.1	gslc_DrvAdaptColor
		5.9.2.2	gslc_DrvAdaptColorRaw
		5.9.2.3	gslc_DrvAdaptRect 161
		5.9.2.4	gslc_DrvCleanStart
		5.9.2.5	gslc_DrvDestruct
		5.9.2.6	gslc_DrvDrawBkgnd
		5.9.2.7	gslc_DrvDrawFillRect
		5.9.2.8	gslc_DrvDrawFrameRect

CONTENTS xix

	5.9.2.9	gslc_DrvDrawGetPixelRaw	34
	5.9.2.10	gslc_DrvDrawImage	34
	5.9.2.11	gslc_DrvDrawLine	34
	5.9.2.12	gslc_DrvDrawPoint	35
	5.9.2.13	gslc_DrvDrawPoints	35
	5.9.2.14	gslc_DrvDrawSetPixelRaw	35
	5.9.2.15	gslc_DrvDrawTxt	6
	5.9.2.16	gslc_DrvFontAdd	6
	5.9.2.17	gslc_DrvFontsDestruct	6
	5.9.2.18	gslc_DrvGetTouch	37
	5.9.2.19	gslc_DrvGetTxtSize	8
	5.9.2.20	gslc_DrvImageDestruct	8
	5.9.2.21	gslc_DrvInit	8
	5.9.2.22	gslc_DrvInitTouch	9
	5.9.2.23	gslc_DrvLoadImage	9
	5.9.2.24	gslc_DrvPageFlipNow	'0
	5.9.2.25	gslc_DrvPasteSurface	'0
	5.9.2.26	gslc_DrvScreenLock	'0
	5.9.2.27	gslc_DrvScreenUnlock	′1
	5.9.2.28	gslc_DrvSetBkgndColor	1
	5.9.2.29	gslc_DrvSetBkgndImage	1
	5.9.2.30	gslc_DrvSetClipRect	'2
	5.9.2.31	gslc_DrvSetElemImageGlow	'2
	5.9.2.32	gslc_DrvSetElemImageNorm	'2
	5.9.2.33	gslc_TDrvGetTouch	'3
	5.9.2.34	gslc_TDrvInitTouch	<b>'</b> 4
5.10 src/GU	Islice_ex.c	File Reference	′4
5.10.1	Function	Documentation	′6
	5.10.1.1	gslc_ElemXCheckboxCreate	′6
	5.10.1.2	gslc_ElemXCheckboxDraw	′6
	5.10.1.3	gslc_ElemXCheckboxFindChecked	'6
	5.10.1.4	gslc_ElemXCheckboxGetState	7
	5.10.1.5	gslc_ElemXCheckboxSetState	'8
	5.10.1.6	gslc_ElemXCheckboxToggleState	'8
	5.10.1.7	gslc_ElemXCheckboxTouch	'8
	5.10.1.8	gslc_ElemXGaugeCreate	'9
	5.10.1.9	gslc_ElemXGaugeDraw	'9
	5.10.1.10	gslc_ElemXGaugeSetFlip	'9
	5.10.1.11	gslc_ElemXGaugeUpdate	30
	5.10.1.12	gslc_ElemXSelNumClick	30

CONTENTS

	5.10.1.13 gslc_ElemXSelNumCreate
	5.10.1.14 gslc_ElemXSelNumDraw
	5.10.1.15 gslc_ElemXSelNumGetCounter
	5.10.1.16 gslc_ElemXSelNumSetCounter
	5.10.1.17 gslc_ElemXSelNumTouch
	5.10.1.18 gslc_ElemXSliderCreate
	5.10.1.19 gslc_ElemXSliderDraw
	5.10.1.20 gslc_ElemXSliderGetPos
	5.10.1.21 gslc_ElemXSliderSetPos
	5.10.1.22 gslc_ElemXSliderSetPosFunc
	5.10.1.23 gslc_ElemXSliderSetStyle
	5.10.1.24 gslc_ElemXSliderTouch
5.10	.2 Variable Documentation
	5.10.2.1 SELNUM_ID_BTN_DEC
	5.10.2.2 SELNUM_ID_BTN_INC
	5.10.2.3 SELNUM_ID_TXT
5.11 src/0	GUIslice_ex.h File Reference
5.11	.1 Macro Definition Documentation
	5.11.1.1 SELNUM_STR_LEN
5.11	.2 Typedef Documentation
	5.11.2.1 GSLC_CB_XSLIDER_POS
5.11	.3 Enumeration Type Documentation
	5.11.3.1 gslc_teTypeExtend
	5.11.3.2 gslc_teXCheckboxStyle
5.11	.4 Function Documentation
	5.11.4.1 gslc_ElemXCheckboxCreate
	5.11.4.2 gslc_ElemXCheckboxDraw
	5.11.4.3 gslc_ElemXCheckboxFindChecked
	5.11.4.4 gslc_ElemXCheckboxGetState
	5.11.4.5 gslc_ElemXCheckboxSetState
	5.11.4.6 gslc_ElemXCheckboxToggleState
	5.11.4.7 gslc_ElemXCheckboxTouch
	5.11.4.8 gslc_ElemXGaugeCreate
	5.11.4.9 gslc_ElemXGaugeDraw
	5.11.4.10 gslc_ElemXGaugeSetFlip
	5.11.4.11 gslc_ElemXGaugeUpdate
	5.11.4.12 gslc_ElemXSelNumClick
	5.11.4.13 gslc_ElemXSelNumCreate
	5.11.4.14 gslc_ElemXSelNumDraw
	5.11.4.15 gslc_ElemXSelNumGetCounter

CONTENTS xxi

5.11.4.16 gslc_ElemXSelNumSetCounter	91
5.11.4.17 gslc_ElemXSelNumTouch	91
5.11.4.18 gslc_ElemXSliderCreate	192
5.11.4.19 gslc_ElemXSliderDraw	192
5.11.4.20 gslc_ElemXSliderGetPos	92
5.11.4.21 gslc_ElemXSliderSetPos	193
5.11.4.22 gslc_ElemXSliderSetPosFunc	193
5.11.4.23 gslc_ElemXSliderSetStyle	193
5 11 4 24 aslc. FlemXSliderTouch	19:

## **Chapter 1**

## **README**

#### **GUIslice library**

A lightweight GUI framework suitable for embedded displays

- Website (www.impulseadventure.com)
- Documentation wiki (github)
- Release notes
- Pure C library, no dynamic memory allocation
- Widgets: text, images, buttons, checkboxes, radio buttons, sliders, etc. plus extensions and multiple pages.
- Platform-independent GUI core currently supports: SDL1.2, SDL2.0, Adafruit-GFX
- Typical target: Raspberry Pi, Arduino, Cortex M0 (Feather M0), LINUX, Beaglebone Black
- Typical displays: PiTFT, Waveshare, Adafruit TFT 2.2" / 2.8" / 1.44', OLED 0.96"
- · Supports touchscreen control
- No GUIslice installation just add include files and go!
- · LINUX Dependencies: sdl, sdl-ttf, optional: tslib
- · Arduino Dependencies: Adafruit-GFX plus display (eg. ILI9341) / touch driver library (eg. STMPE610)

Screenshots

2 README

# **Chapter 2**

## **Class Index**

### 2.1 Class List

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

gslc_tsCollect	
Element collection struct	7
gslc_tsColor	
Color structure. Defines RGB triplet	9
gslc_tsDriver	10
gslc_tsElem	
Element Struct	11
gslc_tsElemRef	
Element reference structure	15
gslc_tsEvent	
Event structure	16
gslc_tsEventTouch	
Structure used to pass touch data through event	17
gslc_tsFont	
Font reference structure	18
gslc_tsGui	
GUI structure	18
gslc_tsImgRef	00
Image reference structure	22
gslc_tsPage	-
Page structure	23
gslc_tsPt	0.4
Define point coordinates	24
gslc_tsRect	0.5
Rectangular region. Defines X,Y corner coordinates plus dimensions	25
gslc_tsXCheckbox	200
Extended data for Checkbox element	26
gslc_tsXGauge	27
Extended data for Gauge element	21
gslc_tsXSelNum  Extended data for SelNum element	28
Extended data for SelNum element	28
gslc_tsXSlider  Extended data for Slider element	30

Class Index

# **Chapter 3**

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

src/GUIslice.c
src/GUIslice.h
src/GUIslice_config.h
src/GUIslice_drv.h
src/GUIslice_drv_adagfx.cpp
src/GUIslice_drv_adagfx.h
src/GUIslice_drv_sdl.c
src/GUIslice_drv_sdl.h
src/GUIslice_ex.c
src/GUIslice_ex.h

6 File Index

## **Chapter 4**

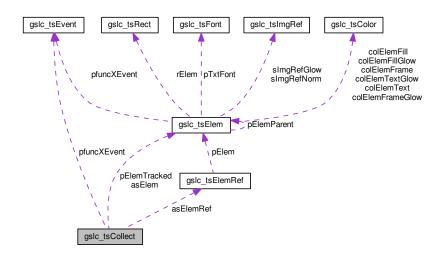
## **Class Documentation**

### 4.1 gslc\_tsCollect Struct Reference

Element collection struct.

#include <GUIslice.h>

Collaboration diagram for gslc\_tsCollect:



#### **Public Attributes**

• 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.

8 Class Documentation

uint16\_t nElemRefMax

Maximum number of element references to allocate.

uint16\_t nElemRefCnt

Number of element references allocated.

gslc\_tsElem \* pElemTracked

Element currently being touch-tracked (NULL for none)

GSLC\_CB\_EVENT pfuncXEvent

Callback func ptr for events.

#### 4.1.1 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

#### 4.1.2 Member Data Documentation

4.1.2.1 gslc\_tsElem\* gslc\_tsCollect::asElem

Array of elements.

4.1.2.2 gslc\_tsElemRef\* gslc\_tsCollect::asElemRef

Array of element references.

4.1.2.3 int16\_t gslc\_tsCollect::nElemAutoIdNext

Next Element ID for auto-assignment.

4.1.2.4 uint16\_t gslc\_tsCollect::nElemCnt

Number of elements allocated.

4.1.2.5 uint16\_t gslc\_tsCollect::nElemMax

Maximum number of elements to allocate (in RAM)

4.1.2.6 uint16\_t gslc\_tsCollect::nElemRefCnt

Number of element references allocated.

4.1.2.7 uint16\_t gslc\_tsCollect::nElemRefMax

Maximum number of element references to allocate.

4.1.2.8 gslc\_tsElem\* gslc\_tsCollect::pElemTracked

Element currently being touch-tracked (NULL for none)

#### 4.1.2.9 GSLC\_CB\_EVENT gslc\_tsCollect::pfuncXEvent

Callback func ptr for events.

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

• src/GUIslice.h

### 4.2 gslc\_tsColor Struct Reference

Color structure. Defines RGB triplet.

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

#### **Public Attributes**

• uint8\_t r

RGB red value.

uint8\_t g

RGB green value.

• uint8 t b

RGB blue value.

· uint8\_t unused

Unused value to pad structure.

#### 4.2.1 Detailed Description

Color structure. Defines RGB triplet.

#### 4.2.2 Member Data Documentation

4.2.2.1 uint8\_t gslc\_tsColor::b

RGB blue value.

4.2.2.2 uint8\_t gslc\_tsColor::g

RGB green value.

4.2.2.3 uint8\_t gslc\_tsColor::r

RGB red value.

4.2.2.4 uint8\_t gslc\_tsColor::unused

Unused value to pad structure.

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

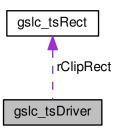
• src/GUIslice.h

10 Class Documentation

### 4.3 gslc\_tsDriver Struct Reference

#include <GUIslice\_drv\_adagfx.h>

Collaboration diagram for gslc\_tsDriver:



#### **Public Attributes**

• uint16\_t nColRawBkgnd

Background color (if not image-based)

gslc\_tsRect rClipRect

Clipping rectangle.

• SDL\_Surface \* pSurfScreen

Surface ptr for screen.

struct tsdev \* pTsDev

Ptr to touchscreen device.

#### 4.3.1 Member Data Documentation

4.3.1.1 uint16\_t gslc\_tsDriver::nColRawBkgnd

Background color (if not image-based)

4.3.1.2 SDL\_Surface\* gslc\_tsDriver::pSurfScreen

Surface ptr for screen.

4.3.1.3 struct tsdev\* gslc\_tsDriver::pTsDev

Ptr to touchscreen device.

4.3.1.4 gslc\_tsRect gslc\_tsDriver::rClipRect

Clipping rectangle.

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

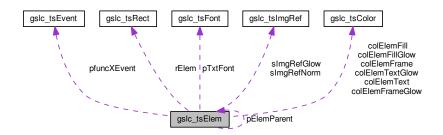
- src/GUIslice\_drv\_adagfx.h
- src/GUIslice\_drv\_sdl.h

### 4.4 gslc\_tsElem Struct Reference

#### Element Struct.

#include <GUIslice.h>

Collaboration diagram for gslc\_tsElem:



#### **Public Attributes**

• int16\_t nld

Element ID specified by user.

bool bValid

Element was created properly.

int16\_t nType

Element type enumeration.

gslc\_tsRect rElem

Rect region containing element.

int16\_t nGroup

Group ID that the element belongs to.

bool bGlowEn

Enable glowing visual state.

bool bClickEn

Element accepts touch events.

bool bFrameEn

Element is drawn with frame.

• bool bFillEn

Element is drawn with inner fill.

• 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)

12 Class Documentation

• gslc\_tsElem \* pElemParent

Parent element reference.

• char pStrBuf [GSLC\_LOCAL\_STR\_LEN]

Text string 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

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.

bool bNeedRedraw

Element needs to be redrawn.

bool bGlowing

Element is currently glowing.

#### 4.4.1 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.

#### 4.4.2 Member Data Documentation

4.4.2.1 bool gslc\_tsElem::bClickEn

Element accepts touch events.

4.4.2.2 bool gslc\_tsElem::bFillEn

Element is drawn with inner fill.

This is also used during redraw to determine if elements underneath are visible and must be redrawn as well.

4.4.2.3 bool gslc\_tsElem::bFrameEn

Element is drawn with frame.

4.4.2.4 bool gslc\_tsElem::bGlowEn

Enable glowing visual state.

4.4.2.5 bool gslc\_tsElem::bGlowing

Element is currently glowing.

4.4.2.6 bool gslc\_tsElem::bNeedRedraw

Element needs to be redrawn.

4.4.2.7 bool gslc\_tsElem::bValid

Element was created properly.

4.4.2.8 gslc\_tsColor gslc\_tsElem::colElemFill

Color for background fill.

4.4.2.9 gslc\_tsColor gslc\_tsElem::colElemFillGlow

Color to use for fill when glowing.

4.4.2.10 gslc\_tsColor gslc\_tsElem::colElemFrame

Color for frame.

4.4.2.11 gslc\_tsColor gslc\_tsElem::colElemFrameGlow

Color to use for frame when glowing.

4.4.2.12 gslc\_tsColor gslc\_tsElem::colElemText

Color of overlay text.

4.4.2.13 gslc\_tsColor gslc\_tsElem::colElemTextGlow

Color of overlay text when glowing.

14 Class Documentation

4.4.2.14 int8\_t gslc\_tsElem::eTxtAlign

Alignment of overlay text.

4.4.2.15 gslc\_teTxtFlags gslc\_tsElem::eTxtFlags

Flags associated with text buffer.

4.4.2.16 int16\_t gslc\_tsElem::nGroup

Group ID that the element belongs to.

4.4.2.17 int16\_t gslc\_tsElem::nld

Element ID specified by user.

4.4.2.18 uint8\_t gslc\_tsElem::nStrBufMax

Size of string buffer.

4.4.2.19 uint8\_t gslc\_tsElem::nTxtMargin

Margin of overlay text within rect region.

4.4.2.20 int16\_t gslc\_tsElem::nType

Element type enumeration.

4.4.2.21 gslc\_tsElem\* gslc\_tsElem::pElemParent

Parent element reference.

Used during redraw to notify parent elements that they require redraw as well. Primary usage is in compound elements.

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

Callback func ptr for custom drawing.

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

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

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

Callback func ptr for timer/main loop tick.

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

Callback func ptr for touch.

4.4.2.26 char gslc\_tsElem::pStrBuf[GSLC\_LOCAL\_STR\_LEN]

Text string to overlay.

4.4.2.27 gslc\_tsFont\* gslc\_tsElem::pTxtFont

Ptr to Font for overlay text.

4.4.2.28 void\* gslc\_tsElem::pXData

Ptr to extended data structure.

4.4.2.29 gslc\_tsRect gslc\_tsElem::rElem

Rect region containing element.

4.4.2.30 gslc\_tsImgRef gslc\_tsElem::sImgRefGlow

Image reference to draw (glowing)

4.4.2.31 gslc\_tslmgRef gslc\_tsElem::slmgRefNorm

Image reference to draw (normal)

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

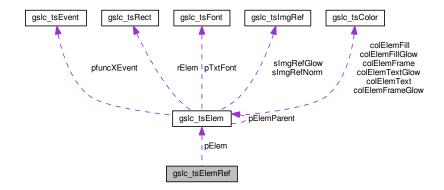
• src/GUIslice.h

# 4.5 gslc\_tsElemRef Struct Reference

Element reference structure.

#include <GUIslice.h>

Collaboration diagram for gslc\_tsElemRef:



# **Public Attributes**

• gslc\_tsElem \* pElem

Pointer to element in memory [RAM,FLASH].

• gslc\_teElemRefFlags eElemFlags

Element reference flags.

# 4.5.1 Detailed Description

Element reference structure.

#### 4.5.2 Member Data Documentation

4.5.2.1 gslc\_teElemRefFlags gslc\_tsElemRef::eElemFlags

Element reference flags.

```
4.5.2.2 gslc_tsElem* gslc_tsElemRef::pElem
```

Pointer to element in memory [RAM,FLASH].

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

· src/GUIslice.h

# 4.6 gslc\_tsEvent Struct Reference

Event structure.

#include <GUIslice.h>

# **Public Attributes**

• 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.

# 4.6.1 Detailed Description

Event structure.

# 4.6.2 Member Data Documentation

4.6.2.1 gslc\_teEventType gslc\_tsEvent::eType

Event type.

4.6.2.2 uint8\_t gslc\_tsEvent::nSubType

Event sub-type.

4.6.2.3 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.

4.6.2.4 void\* gslc\_tsEvent::pvScope

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

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

• src/GUIslice.h

# 4.7 gslc\_tsEventTouch Struct Reference

Structure used to pass touch data through event.

#include <GUIslice.h>

## **Public Attributes**

• gslc\_teTouch eTouch

Touch state.

int16\_t nX

Touch X coordinate (absolute)

int16\_t nY

Touch Y coordinate (absolute)

# 4.7.1 Detailed Description

Structure used to pass touch data through event.

# 4.7.2 Member Data Documentation

4.7.2.1 gslc\_teTouch gslc\_tsEventTouch::eTouch

Touch state.

4.7.2.2 int16\_t gslc\_tsEventTouch::nX

Touch X coordinate (absolute)

## 4.7.2.3 int16\_t gslc\_tsEventTouch::nY

Touch Y coordinate (absolute)

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

• src/GUIslice.h

# 4.8 gslc\_tsFont Struct Reference

```
Font reference structure.
```

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

#### **Public Attributes**

• int16\_t nld

Font ID specified by user.

void \* pvFont

Void ptr to the Font (type defined by driver)

• uint16\_t nSize

Font size.

# 4.8.1 Detailed Description

Font reference structure.

# 4.8.2 Member Data Documentation

```
4.8.2.1 int16_t gslc_tsFont::nld
```

Font ID specified by user.

4.8.2.2 uint16\_t gslc\_tsFont::nSize

Font size.

4.8.2.3 void\* gslc\_tsFont::pvFont

Void ptr to the Font (type defined by driver)

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

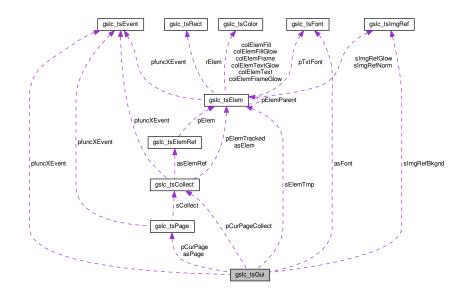
• src/GUIslice.h

# 4.9 gslc\_tsGui Struct Reference

#### GUI structure.

#include <GUIslice.h>

#### Collaboration diagram for gslc\_tsGui:



#### **Public Attributes**

uint16\_t nDispW

Width of the display (pixels)

• uint16\_t nDispH

Height of the display (pixels)

uint8\_t nDispDepth

Bit depth of display (bits per pixel)

• 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 sElemTmp

Temporary element.

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))

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 pages.

uint8\_t nPageMax

Maximum number of pages.

uint8\_t nPageCnt

Current page index.

gslc\_tsPage \* pCurPage

Currently active page.

• gslc\_tsCollect \* pCurPageCollect

Ptr to active page collection.

GSLC\_CB\_EVENT pfuncXEvent

Callback func ptr for events.

## 4.9.1 Detailed Description

GUI structure.

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

#### 4.9.2 Member Data Documentation

4.9.2.1 gslc\_tsFont\* gslc\_tsGui::asFont

Collection of loaded fonts.

4.9.2.2 gslc\_tsPage\* gslc\_tsGui::asPage

Array of pages.

4.9.2.3 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.

4.9.2.4 uint8\_t gslc\_tsGui::nDispDepth

Bit depth of display (bits per pixel)

4.9.2.5 uint16\_t gslc\_tsGui::nDispH

Height of the display (pixels)

4.9.2.6 uint16\_t gslc\_tsGui::nDispW

Width of the display (pixels)

4.9.2.7 uint8\_t gslc\_tsGui::nFontCnt

Number of fonts allocated.

4.9.2.8 uint8\_t gslc\_tsGui::nFontMax

Maximum number of fonts to allocate.

4.9.2.9 uint8\_t gslc\_tsGui::nFrameRateCnt

Diagnostic frame rate count.

4.9.2.10 uint8\_t gslc\_tsGui::nFrameRateStart

Diagnostic frame rate timestamp.

4.9.2.11 uint8\_t gslc\_tsGui::nPageCnt

Current page index.

4.9.2.12 uint8\_t gslc\_tsGui::nPageMax

Maximum number of pages.

4.9.2.13 uint16\_t gslc\_tsGui::nTouchLastPress

Last touch event pressure (0=none))

4.9.2.14 int16\_t gslc\_tsGui::nTouchLastX

Last touch event X coord.

4.9.2.15 int16\_t gslc\_tsGui::nTouchLastY

Last touch event Y coord.

4.9.2.16 gslc\_tsPage\* gslc\_tsGui::pCurPage

Currently active page.

 $4.9.2.17 \quad gslc\_tsCollect* gslc\_tsGui::pCurPageCollect$ 

Ptr to active page collection.

4.9.2.18 GSLC\_CB\_EVENT gslc\_tsGui::pfuncXEvent

Callback func ptr for events.

```
4.9.2.19 void* gslc_tsGui::pvDriver
```

Driver-specific members (gslc\_tsDriver\*)

4.9.2.20 gslc\_tsElem gslc\_tsGui::sElemTmp

Temporary element.

4.9.2.21 gslc\_tslmgRef gslc\_tsGui::slmgRefBkgnd

Image reference for background.

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

• src/GUIslice.h

# 4.10 gslc\_tslmgRef Struct Reference

Image reference structure.

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

#### **Public Attributes**

· 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)

# 4.10.1 Detailed Description

Image reference structure.

## 4.10.2 Member Data Documentation

4.10.2.1 gslc\_telmgRefFlags gslc\_tslmgRef::elmgFlags

Image reference flags.

4.10.2.2 const char\* gslc\_tslmgRef::pFname

Pathname to input image file [FILE,SD].

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

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

4.10.2.4 void\* gslc\_tslmgRef::pvlmgRaw

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

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

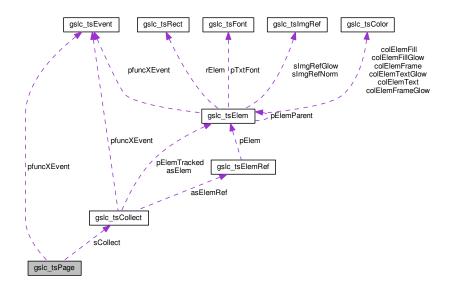
· src/GUIslice.h

# 4.11 gslc\_tsPage Struct Reference

Page structure.

#include <GUIslice.h>

Collaboration diagram for gslc\_tsPage:



# **Public Attributes**

• gslc\_tsCollect sCollect

Collection of elements on page.

• int8\_t nPageId

Page identifier.

bool bPageNeedRedraw

Page require a redraw.

bool bPageNeedFlip

Screen requires a page flip.

GSLC\_CB\_EVENT pfuncXEvent

Callback func ptr for events.

# 4.11.1 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

# 4.11.2 Member Data Documentation

4.11.2.1 bool gslc\_tsPage::bPageNeedFlip

Screen requires a page flip.

4.11.2.2 bool gslc\_tsPage::bPageNeedRedraw

Page require a redraw.

4.11.2.3 int8\_t gslc\_tsPage::nPageId

Page identifier.

4.11.2.4 GSLC\_CB\_EVENT gslc\_tsPage::pfuncXEvent

Callback func ptr for events.

4.11.2.5 gslc\_tsCollect gslc\_tsPage::sCollect

Collection of elements on page.

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

• src/GUIslice.h

# 4.12 gslc\_tsPt Struct Reference

Define point coordinates.

#include <GUIslice.h>

# **Public Attributes**

• int x

X coordinate.

int y

Y coordinate.

# 4.12.1 Detailed Description

Define point coordinates.

# 4.12.2 Member Data Documentation

```
4.12.2.1 int gslc_tsPt::x
```

X coordinate.

```
4.12.2.2 int gslc_tsPt::y
```

Y coordinate.

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

• src/GUIslice.h

# 4.13 gslc\_tsRect Struct Reference

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

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

# **Public Attributes**

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.

# 4.13.1 Detailed Description

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

# 4.13.2 Member Data Documentation

```
4.13.2.1 uint16_t gslc_tsRect::h
```

Height of region.

4.13.2.2 uint16\_t gslc\_tsRect::w

Width of region.

4.13.2.3 int16\_t gslc\_tsRect::x

X coordinate of corner.

4.13.2.4 int16\_t gslc\_tsRect::y

Y coordinate of corner.

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

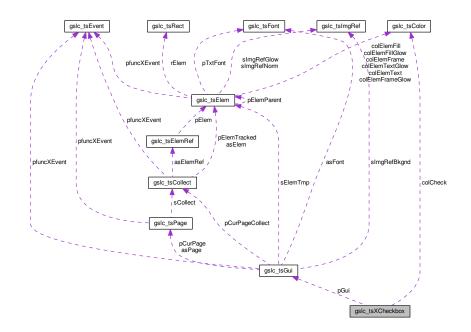
· src/GUIslice.h

# 4.14 gslc\_tsXCheckbox Struct Reference

Extended data for Checkbox element.

#include <GUIslice\_ex.h>

Collaboration diagram for gslc tsXCheckbox:



# **Public Attributes**

• gslc\_tsGui \* pGui

Ptr to GUI (for radio group control)

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.

# 4.14.1 Detailed Description

Extended data for Checkbox element.

# 4.14.2 Member Data Documentation

4.14.2.1 bool gslc\_tsXCheckbox::bChecked

Indicates if it is selected (checked)

4.14.2.2 bool gslc\_tsXCheckbox::bRadio

Radio-button operation if true.

4.14.2.3 gslc\_tsColor gslc\_tsXCheckbox::colCheck

Color of checked inner fill.

4.14.2.4 gslc\_teXCheckboxStyle gslc\_tsXCheckbox::nStyle

Drawing style for element.

4.14.2.5 gslc\_tsGui\* gslc\_tsXCheckbox::pGui

Ptr to GUI (for radio group control)

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

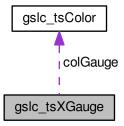
• src/GUIslice\_ex.h

# 4.15 gslc\_tsXGauge Struct Reference

Extended data for Gauge element.

#include <GUIslice\_ex.h>

Collaboration diagram for gslc\_tsXGauge:



# **Public Attributes**

• int16\_t nGaugeMin

Minimum control value.

int16\_t nGaugeMax

Maximum control value.

int16\_t nGaugeVal

Current control value.

gslc\_tsColor colGauge

Color of gauge fill bar.

bool bGaugeVert

Vertical if true, else Horizontal.

bool bGaugeFlip

Reverse direction of gauge.

# 4.15.1 Detailed Description

Extended data for Gauge element.

# 4.15.2 Member Data Documentation

4.15.2.1 bool gslc\_tsXGauge::bGaugeFlip

Reverse direction of gauge.

4.15.2.2 bool gslc\_tsXGauge::bGaugeVert

Vertical if true, else Horizontal.

4.15.2.3 gslc\_tsColor gslc\_tsXGauge::colGauge

Color of gauge fill bar.

4.15.2.4 int16\_t gslc\_tsXGauge::nGaugeMax

Maximum control value.

4.15.2.5 int16\_t gslc\_tsXGauge::nGaugeMin

Minimum control value.

4.15.2.6 int16\_t gslc\_tsXGauge::nGaugeVal

Current control value.

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

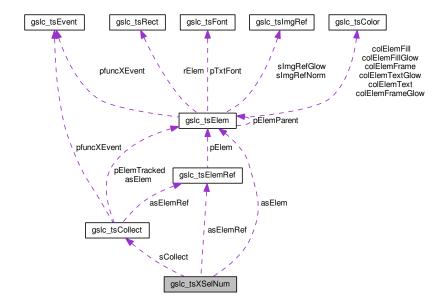
• src/GUIslice\_ex.h

# 4.16 gslc\_tsXSelNum Struct Reference

Extended data for SelNum element.

#include <GUIslice\_ex.h>

Collaboration diagram for gslc\_tsXSelNum:



# **Public Attributes**

• int16\_t nCounter

Counter for demo purposes.

gslc\_tsCollect sCollect

Collection management for sub-elements.

• gslc\_tsElemRef asElemRef [4]

Storage for sub-element references.

• gslc\_tsElem asElem [4]

Storage for sub-elements.

char acElemTxt [4][SELNUM\_STR\_LEN]

Storage for strings.

# 4.16.1 Detailed Description

Extended data for SelNum element.

# 4.16.2 Member Data Documentation

4.16.2.1 char gslc\_tsXSelNum::acElemTxt[4][SELNUM\_STR\_LEN]

Storage for strings.

4.16.2.2 gslc\_tsElem gslc\_tsXSelNum::asElem[4]

Storage for sub-elements.

4.16.2.3 gslc\_tsElemRef gslc\_tsXSelNum::asElemRef[4]

Storage for sub-element references.

4.16.2.4 int16\_t gslc\_tsXSelNum::nCounter

Counter for demo purposes.

4.16.2.5 gslc\_tsCollect gslc\_tsXSelNum::sCollect

Collection management for sub-elements.

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

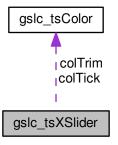
• src/GUIslice\_ex.h

# 4.17 gslc\_tsXSlider Struct Reference

Extended data for Slider element.

#include <GUIslice\_ex.h>

Collaboration diagram for gslc\_tsXSlider:



#### **Public Attributes**

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.

# 4.17.1 Detailed Description

Extended data for Slider element.

#### 4.17.2 Member Data Documentation

4.17.2.1 bool gslc\_tsXSlider::bTrim

Style: show a trim color.

4.17.2.2 bool gslc\_tsXSlider::bVert

Orientation: true if vertical, else horizontal.

4.17.2.3 gslc\_tsColor gslc\_tsXSlider::colTick

Style: color of ticks.

4.17.2.4 gslc\_tsColor gslc\_tsXSlider::colTrim

Style: color of trim.

4.17.2.5 int16\_t gslc\_tsXSlider::nPos

Current position value of the slider.

4.17.2.6 int16\_t gslc\_tsXSlider::nPosMax

Maximum position value of the slider.

4.17.2.7 int16\_t gslc\_tsXSlider::nPosMin

Minimum position value of the slider.

4.17.2.8 int16\_t gslc\_tsXSlider::nThumbSz

Size of the thumb control.

4.17.2.9 uint16\_t gslc\_tsXSlider::nTickDiv

Style: number of tickmark divisions (0 for none)

4.17.2.10 int16\_t gslc\_tsXSlider::nTickLen

Style: length of tickmarks.

4.17.2.11 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/GUIslice\_ex.h

# **Chapter 5**

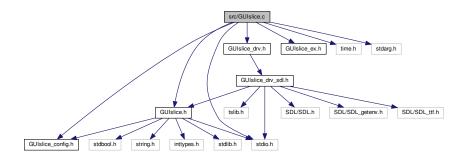
# **File Documentation**

# 5.1 README.md File Reference

# 5.2 src/GUIslice.c File Reference

```
#include "GUIslice_config.h"
#include "GUIslice.h"
#include "GUIslice_ex.h"
#include "GUIslice_drv.h"
#include <stdio.h>
#include <time.h>
#include <stdarg.h>
```

Include dependency graph for GUIslice.c:



### Macros

• #define GUISLICE\_VER "0.8.6"

# **Functions**

- char \* gslc\_GetVer (gslc\_tsGui \*pGui)
  - Get the GUIslice version number.
- 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.

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\_teEventType eType, uint8\_t nSubType, void \*pvScope, 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\_lsInWH (gslc\_tsGui \*pGui, 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\_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\_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\_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.

bool gslc FontAdd (gslc tsGui \*pGui, int16 t nFontId, const char \*acFontName, 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\_ts
 ElemRef \*psElemRef, uint16 t nMaxElemRef)

Add a page to the GUI.

int gslc\_GetPageCur (gslc\_tsGui \*pGui)

Fetch the current page ID.

void gslc\_SetPageCur (gslc\_tsGui \*pGui, int16\_t nPageId)

Select a new page for display.

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 PageFindByld (gslc tsGui \*pGui, int16 t nPageId)

Find a page in the GUI by its ID.

gslc\_tsElem \* gslc\_PageFindElemByld (gslc\_tsGui \*pGui, int16\_t nPageId, int16\_t nElemId)

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

void gslc PageSetEventFunc (gslc tsPage \*pPage, GSLC CB EVENT funcCb)

Assign the event callback function for a page.

int gslc\_ElemGetId (gslc\_tsElem \*pElem)

Get an Element ID from an element structure.

 gslc\_tsElem \* 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\_tsElem \* 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\_tsElem \* gslc\_ElemCreateBtnImg (gslc\_tsGui \*pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsRect r←
Elem, gslc\_tsImgRef sImgRef, gslc\_tsImgRef sImgRefSel, GSLC\_CB\_TOUCH cbTouch)

Create a graphical Button Element.

• gslc\_tsElem \* gslc\_ElemCreateBox (gslc\_tsGui \*pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsRect rElem)

Create a Box Element.

• gslc\_tsElem \* 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\_tsElem \* gslc\_ElemCreateImg (gslc\_tsGui \*pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsRect rElem, 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\_tsElem \*pElem)

Draw an element to the active display.

void gslc\_ElemSetFillEn (gslc\_tsElem \*pElem, bool bFillEn)

Set the fill state for an Element.

void gslc\_ElemSetFrameEn (gslc\_tsElem \*pElem, bool bFrameEn)

Set the frame state for an Element.

 void gslc\_ElemSetCol (gslc\_tsElem \*pElem, gslc\_tsColor colFrame, gslc\_tsColor colFill, gslc\_tsColor col← FillGlow)

Update the common color selection for an Element.

void gslc\_ElemSetGlowCol (gslc\_tsElem \*pElem, gslc\_tsColor colFrameGlow, gslc\_tsColor colFillGlow, gslc\_tsColor colTxtGlow)

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

void gslc\_ElemSetGroup (gslc\_tsElem \*pElem, int nGroupId)

Set the group ID for an element.

int gslc\_ElemGetGroup (gslc\_tsElem \*pElem)

Get the group ID for an element.

void gslc\_ElemSetTxtAlign (gslc\_tsElem \*pElem, unsigned nAlign)

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

void gslc\_ElemSetTxtMargin (gslc\_tsElem \*pElem, unsigned nMargin)

Set the margin around of a textual element.

void gslc\_ElemSetTxtStr (gslc\_tsElem \*pElem, const char \*pStr)

Update the text string associated with an Element ID.

void gslc\_ElemSetTxtCol (gslc\_tsElem \*pElem, gslc\_tsColor colVal)

Update the text string color associated with an Element ID.

void gslc\_ElemSetTxtMem (gslc\_tsElem \*pElem, gslc\_teTxtFlags eFlags)

Update the text string location in memory.

void gslc\_ElemUpdateFont (gslc\_tsGui \*pGui, gslc\_tsElem \*pElem, int nFontId)

Update the Font selected for an Element's text.

void gslc\_ElemSetRedraw (gslc\_tsElem \*pElem, bool bRedraw)

Update the need-redraw status for an element.

bool gslc\_ElemGetRedraw (gslc\_tsElem \*pElem)

Get the need-redraw status for an element.

void gslc\_ElemSetGlow (gslc\_tsElem \*pElem, bool bGlowing)

Update the glowing indicator for an element.

bool gslc\_ElemGetGlow (gslc\_tsElem \*pElem)

Get the glowing indicator for an element.

void gslc\_ElemSetGlowEn (gslc\_tsElem \*pElem, bool bGlowEn)

Update the glowing enable for an element.

bool gslc ElemGetGlowEn (gslc tsElem \*pElem)

Get the glowing enable for an element.

void gslc\_ElemSetStyleFrom (gslc\_tsElem \*pElemSrc, gslc\_tsElem \*pElemDest)

Copy style settings from one element to another.

void gslc ElemSetEventFunc (gslc tsElem \*pElem, GSLC CB EVENT funcCb)

Assign the event callback function for a element.

void gslc ElemSetDrawFunc (gslc tsElem \*pElem, GSLC CB DRAW funcCb)

Assign the drawing callback function for an element.

void gslc ElemSetTickFunc (gslc tsElem \*pElem, GSLC CB TICK funcCb)

Assign the tick callback function for an element.

• bool gslc\_ElemOwnsCoord (gslc\_tsElem \*pElem, int16\_t nX, int16\_t nY, bool bOnlyClickEn)

Determine if a coordinate is inside of an element.

void gslc\_CollectTouch (gslc\_tsGui \*pGui, gslc\_tsCollect \*pCollect, gslc\_tsEventTouch \*pEventTouch)

Handle touch 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

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)

Initialize the touchscreen device driver.

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.

Add an element to a collection.

bool gslc\_CollectGetRedraw (gslc\_tsCollect \*pCollect)

Determine if any elements in a collection need redraw.

 gslc\_tsElem \* 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.

bool gslc\_SetClipRect (gslc\_tsGui \*pGui, gslc\_tsRect \*pRect)

Set the clipping rectangle for further drawing.

void gslc\_ElemSetImage (gslc\_tsGui \*pGui, gslc\_tsElem \*pElem, gslc\_tsImgRef sImgRef, gslc\_tsImgRef sImgRef sImgRe

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\_ElemSendEventTouch (gslc\_tsGui \*pGui, gslc\_tsElem \*pElemTracked, gslc\_teTouch eTouch, 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\_tsCollect \*pCollect)

Free up any members associated with an element collection.

void gslc\_PageDestruct (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.

gslc\_tsElem \* gslc\_CollectFindElemById (gslc\_tsCollect \*pCollect, int16\_t nElemId)

Find an element in a collection by its Element ID.

• int gslc\_CollectGetNextId (gslc\_tsCollect \*pCollect)

Allocate the next available Element ID in a collection.

gslc\_tsElem \* gslc\_CollectGetElemTracked (gslc\_tsCollect \*pCollect)

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

void gslc\_CollectSetElemTracked (gslc\_tsCollect \*pCollect, gslc\_tsElem \*pElem)

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

• gslc\_tsElem \* gslc\_CollectFindElemFromCoord (gslc\_tsCollect \*pCollect, int16\_t nX, int16\_t nY)

Find an element in a collection by a coordinate coordinate.

void gslc\_CollectSetParent (gslc\_tsCollect \*pCollect, gslc\_tsElem \*pElemParent)

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

• void gslc\_CollectSetEventFunc (gslc\_tsCollect \*pCollect, GSLC\_CB\_EVENT funcCb)

Assign the event callback function for an element collection.

#### **Variables**

• GSLC\_CB\_DEBUG\_OUT g\_pfDebugOut = NULL

Global debug output function.

#### 5.2.1 Macro Definition Documentation

5.2.1.1 #define GUISLICE\_VER "0.8.6"

#### 5.2.2 Function Documentation

5.2.2.1 bool gslc\_ClipLine ( gslc\_tsRect \* pClipRect, int16\_t \* pnX0, int16\_t \* pnY0, 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

5.2.2.2 bool gslc\_ClipPt ( gslc\_tsRect \* pClipRect, int16\_t nX, int16\_t nY )

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

#### **Parameters**

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

#### Returns

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

5.2.2.3 bool gslc\_ClipRect ( gslc\_tsRect \* pClipRect, gslc\_tsRect \* pRect )

Perform basic clipping of a rectangle to a clipping region.

• Coordinates in parameter rect are modified to fit the region

#### **Parameters**

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

#### Returns

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

5.2.2.4 void gslc\_CollectDestruct ( gslc\_tsCollect \* pCollect )

Free up any members associated with an element collection.

# Parameters

in	pCollect	Pointer to collection
----	----------	-----------------------

#### Returns

none

5.2.2.5 gslc\_tsElem\* gslc\_CollectElemAdd ( 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	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 in the collection that has been added or NULL if there was an error

5.2.2.6 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

5.2.2.7 gslc\_tsElem\* gslc\_CollectFindElemByld ( gslc\_tsCollect \* pCollect, int16\_t nElemId )

Find an element in a collection by its Element ID.

#### **Parameters**

in	pCollect	Pointer to the collection
in	nElemId	Element ID to search for

# Returns

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

5.2.2.8 gslc\_tsElem\* gslc\_CollectFindElemFromCoord ( 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	pCollect	Pointer to the collection
in	nX	Absolute X coordinate to use for search
in	nY	Absolute Y coordinate to use for search

## Returns

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

5.2.2.9 gslc\_tsElem\* gslc\_CollectGetElemTracked ( gslc\_tsCollect \* pCollect )

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

#### **Parameters**

in	pCollect	Pointer to the collection
----	----------	---------------------------

#### Returns

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

5.2.2.10 int gslc\_CollectGetNextId ( gslc\_tsCollect \* pCollect )

Allocate the next available Element ID in a collection.

#### **Parameters**

in	pCollect	Pointer to the collection
----	----------	---------------------------

#### Returns

Element ID that is reserved for use

5.2.2.11 bool gslc\_CollectGetRedraw ( gslc\_tsCollect \* pCollect )

Determine if any elements in a collection need redraw.

#### **Parameters**

in	pCollect	Pointer to Element collection
----	----------	-------------------------------

#### Returns

True if redraw required, false otherwise

5.2.2.12 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

5.2.2.13 void gslc\_CollectSetElemTracked ( gslc\_tsCollect \* pCollect, gslc\_tsElem \* pElem )

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

#### **Parameters**

in	pCollect	Pointer to the collection
in	pElem	Ptr to element to mark as being tracked

#### Returns

none

5.2.2.14 void gslc\_CollectSetEventFunc ( gslc\_tsCollect \* pCollect, GSLC\_CB\_EVENT funcCb )

Assign the event callback function for an element collection.

#### **Parameters**

in	pCollect	Pointer to collection
in	funcCb	Function pointer to event routine (or NULL for default))

#### Returns

none

5.2.2.15 void gslc\_CollectSetParent ( gslc\_tsCollect \* pCollect, gslc\_tsElem \* pElemParent )

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	pCollect	Pointer to the collection
in	pElemParent	Ptr to element that is the parent

#### Returns

none

5.2.2.16 void gslc\_CollectTouch ( gslc\_tsGui \* pGui, gslc\_tsCollect \* pCollect, gslc\_tsEventTouch \* pEventTouch )

Handle touch events within the element collection.

#### **Parameters**

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

#### Returns

none

5.2.2.17 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 <a href="mailto:gslc\_InitDebug">gslc\_InitDebug</a>()

#### **Parameters**

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

#### Returns

none

5.2.2.18 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

5.2.2.19 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

5.2.2.20 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

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

Draw an arbitrary line using Bresenham's algorithm.

#### **Parameters**

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

#### Returns

none

5.2.2.22 void gslc\_DrawLineH ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, uint16\_t nW, gslc\_tsColor nCol )

Draw a horizontal line.

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

#### **Parameters**

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

## Returns

none

5.2.2.23 void gslc\_DrawLineV ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, uint16\_t nH, gslc\_tsColor nCol )

Draw a vertical line.

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

# **Parameters**

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

## Returns

none

5.2.2.24 void gslc\_DrawSetPixel (  $gslc_tsGui*pGui$ , int16\_t nX, int16\_t nY,  $gslc_tsColor nCol$ )

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

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

#### **Parameters**

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

#### Returns

none

5.2.2.25 gslc\_tsElem\* gslc\_ElemAdd ( gslc\_tsGui \* pGui, int16\_t nPageld, 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	nPageld	Page ID to add element to (GSLC_PAGE_NONE to skip in case of temporary
		creation for compound elements)
in	pElem	Pointer to Element to add
in	eFlags	Flags describing the element (eg. whether the element should be stored in
		internal RAM array or is located in Flash/PROGMEM).

#### Returns

Pointer to Element or NULL if fail

5.2.2.26 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	nPageld	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

5.2.2.27 gslc\_tsElem\* 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	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 box size

#### Returns

Pointer to the Element or NULL if failure

5.2.2.28 gslc\_tsElem\* 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 or NULL if failure

5.2.2.29 gslc\_tsElem\* gslc\_ElemCreateBtnTxt ( gslc\_tsGui \* pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsRect rElem, char \* pStrBuf, uint8\_t nStrBufMax, int16\_t nFontId, GSLC\_CB\_TOUCH cbTouch )

Create a textual Button Element.

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

#### **Parameters**

in	pGui	Pointer to GUI

in	nElemId	Element ID to assign (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
in	cbTouch	Callback for touch events

#### Returns

Pointer to the Element or NULL if failure

5.2.2.30 gslc\_tsElem\* 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	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 box size
in	sImgRef	Image reference to load

# Returns

Pointer to the Element or NULL if failure

5.2.2.31 gslc\_tsElem\* gslc\_ElemCreateLine ( gslc\_tsGui \* pGui, int16\_t nElemId, int16\_t nPage, int16\_t nX0, int16\_t nY0, int16\_t nX1, int16\_t nY1 )

Create a Line Element.

· Draws a line with fill color

#### **Parameters**

in	pGui	Pointer to GUI
in	nElemId	Element ID to assign (016383 or GSLC_ID_AUTO to autogen)
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 or NULL if failure

5.2.2.32 gslc\_tsElem\* 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	nFontld	Font ID to use for text display

#### Returns

Pointer to the Element or NULL if failure

5.2.2.33 void gslc\_ElemDestruct ( gslc\_tsElem \* pElem )

Free up any members associated with an element.

#### **Parameters**

- 1			
	in	pElem	Pointer to element

## Returns

none

5.2.2.34 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	nPageld	ID of page containing element
in	nElemId	ID of element

## Returns

none

5.2.2.35 bool gslc\_ElemDrawByRef (  $gslc_tsGui*pGui, gslc_tsElem*pElem$  )

Draw an element to the active display.

· Element is referenced by an element pointer

#### **Parameters**

	in	pGui	Pointer to GUI
Г	in	pElem	Ptr to Element to draw

# Returns

true if success, false otherwise

5.2.2.36 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

5.2.2.37 bool gslc\_ElemGetGlow (  $gslc_tsElem*pElem*$ )

Get the glowing indicator for an element.

#### **Parameters**

in	pElem	Pointer to Element

#### Returns

True if element is glowing

5.2.2.38 bool gslc\_ElemGetGlowEn ( gslc\_tsElem \* pElem )

Get the glowing enable for an element.

# **Parameters**

in	pElem	Pointer to Element
----	-------	--------------------

## Returns

True if element supports glowing

5.2.2.39 int gslc\_ElemGetGroup ( gslc\_tsElem \* pElem )

Get the group ID for an element.

**Parameters** 

in	pElem	Pointer to Element

## **Returns**

Group ID or GSLC\_GROUP\_ID\_NONE if unassigned

5.2.2.40 int gslc\_ElemGetId ( gslc\_tsElem \* pElem )

Get an Element ID from an element structure.

#### **Parameters**

in	pElem	Pointer to element structure
----	-------	------------------------------

#### Returns

ID of element or GSLC ID NONE if not found

5.2.2.41 bool gslc\_ElemGetRedraw ( gslc\_tsElem \* pElem )

Get the need-redraw status for an element.

#### **Parameters**

in	pElem	Pointer to Element
----	-------	--------------------

## Returns

True if redraw required, false otherwise

5.2.2.42 bool gslc\_ElemOwnsCoord ( gslc\_tsElem \* pElem, 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	pElem	Element used for boundary test
in	nX	X coordinate to test
in	nY	Y coordinate to test
in	bOnlyClickEn	Only output true if element was also marked as "clickable" (eg. bClickEn=true)

# Returns

true if inside element, false otherwise

5.2.2.43 bool gslc\_ElemSendEventTouch ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElemTracked, 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	pElemTracked	Pointer to tracked Element (or NULL for none))
in	eTouch	Touch event type
in	nX	X coordinate of event (absolute coordinate)
in	nY	Y coordinate of event (absolute coordinate)

#### Returns

true if success, false if error

5.2.2.44 void gslc\_ElemSetCol ( gslc\_tsElem \* pElem, gslc\_tsColor colFrame, gslc\_tsColor colFill, gslc\_tsColor colFillGlow )

Update the common color selection for an Element.

## **Parameters**

in	pElem	Pointer to Element
in	colFrame	Color for the frame
in	colFill	Color for the fill
in	colFillGlow	Color for the fill when glowing

# Returns

none

5.2.2.45 void gslc\_ElemSetDrawFunc (  $gslc_tsElem*pElem$ ,  $GSLC_CB_DRAW$  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	pElem	Pointer to Element
in	funcCb	Function pointer to drawing routine (or NULL for default))

## Returns

none

5.2.2.46 void gslc\_ElemSetEventFunc ( gslc\_tsElem \* pElem, GSLC\_CB\_EVENT funcCb )

Assign the event callback function for a element.

# **Parameters**

in	pElem	Pointer to element
----	-------	--------------------

in	funcCb	Function pointer to event routine (or NULL for default))

# Returns

none

5.2.2.47 void gslc\_ElemSetFillEn ( gslc\_tsElem \* pElem, bool bFillEn )

Set the fill state for an Element.

## **Parameters**

in	pElem	Pointer to Element
in	bFillEn	True if filled, false otherwise

## Returns

none

5.2.2.48 void gslc\_ElemSetFrameEn (  $gslc_tsElem*pElem$ , bool bFrameEn )

Set the frame state for an Element.

## **Parameters**

in	pElem	Pointer to Element
in	bFrameEn	True if framed, false otherwise

#### Returns

none

5.2.2.49 void gslc\_ElemSetGlow (  $gslc\_tsElem*pElem$ , bool bGlowing )

Update the glowing indicator for an element.

## **Parameters**

in	pElem	Pointer to Element
in	bGlowing	True if element is glowing

## Returns

none

5.2.2.50 void gslc\_ElemSetGlowCol ( gslc\_tsElem \* pElem, gslc\_tsColor colFrameGlow, gslc\_tsColor colTillGlow, gslc\_tsColor colTxtGlow )

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

#### **Parameters**

in	pElem	Pointer to Element
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

5.2.2.51 void gslc\_ElemSetGlowEn ( gslc\_tsElem \* pElem, bool bGlowEn )

Update the glowing enable for an element.

#### **Parameters**

in	pElem	Pointer to Element
in	bGlowEn	True if element should support glowing

## **Returns**

none

5.2.2.52 void gslc\_ElemSetGroup (  $gslc\_tsElem*pElem$ , int nGroupId )

Set the group ID for an element.

• Typically used to associate radio button elements together

## **Parameters**

-	in	pElem	Pointer to Element
	in	nGroupId	Group ID to assign

# Returns

none

5.2.2.53 void gslc\_ElemSetImage ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElem, gslc\_tsImgRef sImgRef, gslc\_tsImgRef sImgRefSel )

Set an element to use a bitmap image.

## **Parameters**

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

## Returns

none

5.2.2.54 void gslc\_ElemSetRedraw ( gslc\_tsElem \* pElem, bool bRedraw )

Update the need-redraw status for an element.

#### **Parameters**

in	pElem	Pointer to Element
in	bRedraw	True if redraw required, false otherwise

## Returns

none

5.2.2.55 void gslc\_ElemSetStyleFrom ( gslc\_tsElem \* pElemSrc, gslc\_tsElem \* pElemDest )

Copy style settings from one element to another.

# **Parameters**

in	pElemSrc	Pointer to source Element
in	pElemDest	Pointer to destination Element

# Returns

none

5.2.2.56 void gslc\_ElemSetTickFunc ( gslc\_tsElem \* pElem, 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	pElem	Pointer to Element
in	funcCb	Function pointer to tick routine (or NULL for none))

Returns

none

5.2.2.57 void gslc\_ElemSetTxtAlign (  $gslc_tsElem*pElem$ , unsigned nAlign )

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

## **Parameters**

in	pElem	Pointer to Element
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

5.2.2.58 void gslc\_ElemSetTxtCol (  $gslc_tsElem*pElem*, gslc_tsColor* colVal$  )

Update the text string color associated with an Element ID.

## **Parameters**

in	pElem	Pointer to Element
in	colVal	RGB color to change to

# Returns

none

5.2.2.59 void gslc\_ElemSetTxtMargin (  $gslc\_tsElem*pElem$ , unsigned nMargin )

Set the margin around of a textual element.

## **Parameters**

in pElem   Pointer to Element
-------------------------------

1	in	nMargin	Number of pixels gap to leave surrounding text
		a. g	i tambor or pinore gap to real or arraning tone

## Returns

none

 $5.2.2.60 \quad \text{void gslc\_ElemSetTxtMem ( } \textbf{gslc\_tsElem} * \textbf{pElem}, \textbf{gslc\_teTxtFlags} \textit{eFlags} \textbf{ )}$ 

Update the text string location in memory.

## **Parameters**

in	pElem	Pointer to Element
in	eFlags	Flags associated with text memory location (GSLC_TXT_MEM_*)

## Returns

none

5.2.2.61 void gslc\_ElemSetTxtStr ( gslc\_tsElem \* pElem, const char \* pStr )

Update the text string associated with an Element ID.

## **Parameters**

in	pElem	Pointer to Element
in	pStr	String to copy into element

## Returns

none

5.2.2.62 void gslc\_ElemUpdateFont ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElem, int nFontId )

Update the Font selected for an Element's text.

## **Parameters**

in	pGui	Pointer to GUI
in	pElem	Pointer to Element
in	nFontld	Font ID to select

# Returns

none

5.2.2.63 gslc\_tsEvent gslc\_EventCreate ( gslc\_teEventType eType, uint8\_t nSubType, void \* pvScope, void \* pvData )

Create an event structure.

#### **Parameters**

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

5.2.2.64 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 neg-
		ative)
in	nExpandH	Number of pixels to expand the height (if positive) of contract the height (if
		negative)

## Returns

gslc\_tsRect() with resized dimensions

5.2.2.65 bool gslc\_FontAdd ( gslc\_tsGui \* pGui, int16\_t nFontId, const char \* acFontName, uint16\_t nFontSz )

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

## **Parameters**

in	pGui	Pointer to GUI
in	nFontld	ID to use when referencing this font
in	acFontName	Filename path to the font
in	nFontSz	Typeface size to use

## Returns

true if load was successful, false otherwise

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

Fetch a font from its ID value.

## **Parameters**

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

# Returns

A pointer to the font structure or NULL if error

 $5.2.2.67 \quad gslc\_tslmgRef \ gslc\_GetlmageFromFile \ ( \ const \ char * pFname, \ gslc\_telmgRefFlags \ eFmt \ )$ 

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

 $5.2.2.68 \quad gslc\_tslmgRef \ gslc\_detlmageFromProg \ ( \ const \ unsigned \ char * \textit{plmgBuf}, \ gslc\_telmgRefFlags \ \textit{eFmt} \ )$ 

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

#### **Parameters**

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

## Returns

Loaded image reference

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

Create an image reference to a bitmap in SRAM.

#### **Parameters**

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

# Returns

Loaded image reference

5.2.2.70 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

5.2.2.71 int gslc\_GetPageCur (  $gslc_tsGui * pGui$  )

Fetch the current page ID.

#### **Parameters**

in	pGui	Pointer to GUI

#### Returns

Page ID

5.2.2.72 bool gslc\_GetTouch (  $gslc_tsGui*pGui$ ,  $int16_t*pnX$ ,  $int16_t*pnY$ ,  $uint16_t*pnPress$  )

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

## Returns

true if touch event, false otherwise

5.2.2.73 char\* gslc\_GetVer ( gslc\_tsGui \* pGui )

Get the GUIslice version number.

Returns

String containing version number

5.2.2.74 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

5.2.2.75 bool gslc\_lnit ( 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	d pointer to Driver struct (gslc_tsDriver*)			
in	asPage	Pointer to Page array			
in	nMaxPage	ize of Page array			
in	asFont	Pointer to Font array			
in	nMaxFont	Size of Font array			

#### Returns

true if success, false if fail

5.2.2.76 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

5.2.2.77 bool gslc\_lnitTouch ( gslc\_tsGui \* pGui, const char \* acDev )

Initialize the touchscreen device driver.

# **Parameters**

in	pGui	Pointer t	o GUI								
in	acDev	Device	path	to	touchscreen	(or	""	if	not	applicable))	eg.
		"/dev/inp	out/touc	hscre	en"						

# Returns

true if successful

5.2.2.78 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**

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

## **Returns**

true if inside region, false otherwise

5.2.2.79 bool gslc\_lslnWH (  $gslc_tsGui * pGui$ , 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	pGui	Pointer to GUI	
in	nSelX	X coordinate to test	
in	nSelY	X coordinate to test	
in	nWidth	dth to test against	
in	nHeight	Height to test against	

## Returns

true if inside region, false otherwise

```
5.2.2.80 void gslc_OrderCoord ( int16_t * pnX0, int16_t * pnY0, int16_t * pnX1, int16_t * pnY1 )
```

5.2.2.81 void gslc\_PageAdd ( gslc\_tsGui \* pGui, int16\_t nPageId, gslc\_tsElem \* psElem, uint16\_t nMaxElem, gslc\_tsElemRef \* psElemRef, uint16\_t nMaxElemRef )

Add a page to the GUI.

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

# **Parameters**

in	pGui	Pointer to GUI	
in	nPageld	Page ID to assign	
in	psElem	ternal 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 ele-	
		ments, 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

5.2.2.82 void gslc\_PageDestruct ( gslc\_tsPage \* pPage )

Free up any members associated with a page.

## **Parameters**

in	pPage	Pointer to Page
----	-------	-----------------

## Returns

none

5.2.2.83 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

5.2.2.84 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	nPageId	Page ID to search

# Returns

Ptr to a page or NULL if none found

 $5.2.2.85 \quad \textbf{gslc\_tsElem} * \textbf{gslc\_PageFindElemByld (} \ \textbf{gslc\_tsGui} * \textbf{\textit{pGui}}, \ \textbf{int16\_t} \ \textbf{\textit{nPageId}}, \ \textbf{int16\_t} \ \textbf{\textit{nElemId}})$ 

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

#### **Parameters**

in	pGui	Pointer to GUI	
in	nPageld	age ID to search	
in	nElemId	Element ID to search	

#### Returns

Ptr to an element or NULL if none found

5.2.2.86 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

5.2.2.87 void gslc\_PageFlipGo ( gslc\_tsGui \* pGui )

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

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

#### **Parameters**

in	pGui	Pointer to GUI

## Returns

None

5.2.2.88 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

5.2.2.89 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

5.2.2.90 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

5.2.2.91 void gslc\_PageRedrawGo ( gslc\_tsGui \* pGui )

Redraw all elements on the active page.

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

## **Parameters**

in	pGui	Pointer to GUI

# Returns

none

5.2.2.92 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

5.2.2.93 void gslc\_PageSetEventFunc ( gslc\_tsPage \* pPage, GSLC\_CB\_EVENT funcCb )

Assign the event callback function for a page.

#### **Parameters**

in	pPage	Pointer to page
in	funcCb	Function pointer to event routine (or NULL for default))

## Returns

none

5.2.2.94 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

5.2.2.95 void gslc\_ResetElem ( gslc\_tsElem \* pElem )

Initialize an Element struct.

## **Parameters**

in	pElem	Pointer to Element

## Returns

none

5.2.2.96 void gslc\_ResetFont ( gslc\_tsFont \* pFont )

Initialize a Font struct.

# Parameters

in	pFont	Pointer to Font

## Returns

none

5.2.2.97 gslc\_tslmgRef gslc\_ResetImage ( )

Create a blank image reference structure.

## Returns

Image reference struct

5.2.2.98 bool gslc\_SetBkgndColor ( gslc\_tsGui \* pGui, gslc\_tsColor nCol )

Configure the background to use a solid color.

• The background is used when redrawing the entire page

#### **Parameters**

in	pGui	Pointer to GUI
in	nCol	RGB Color to use

#### Returns

true if success, false if fail

5.2.2.99 bool gslc\_SetBkgndlmage ( 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

5.2.2.100 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

5.2.2.101 void gslc\_SetPageCur ( gslc\_tsGui \* pGui, int16\_t nPageld )

Select a new page for display.

# Parameters

in	pGui	Pointer to GUI

in	nPageId	Page ID to select as current	
----	---------	------------------------------	--

## Returns

none

5.2.2.102 void gslc\_TrackTouch (  $gslc_tsGui*pGui*, gslc_tsPage*pPage*, int16_t nX*, int16_t nY*, uint16_t nPress*)$ 

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

## **Parameters**

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

#### Returns

none

5.2.2.103 void gslc\_Update ( gslc\_tsGui \* pGui )

Perform main GUIslice handling functions.

- · Handles any touch events
- · Performs any necessary screen redraw

# **Parameters**

in
----

#### Returns

None

# 5.2.3 Variable Documentation

5.2.3.1 GSLC\_CB\_DEBUG\_OUT g\_pfDebugOut = NULL

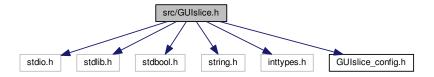
Global debug output function.

• The user assigns this function via gslc\_InitDebug()

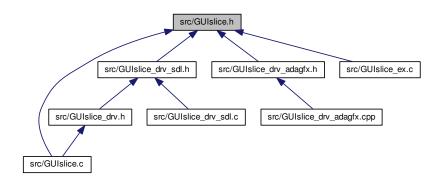
# 5.3 src/GUIslice.h File Reference

```
#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:



# Classes

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\_tsElem

Element Struct.

struct gslc\_tsElemRef

Element reference structure.

struct gslc\_tsCollect

Element collection struct.

struct gslc\_tsPage

Page structure.

• struct gslc\_tsGui

GUI structure.

## **Macros**

• #define GSLC\_ALIGNV\_TOP 0x10

Vertical align to top.

• #define GSLC\_ALIGNV\_MID 0x20

Vertical align to middle.

#define GSLC\_ALIGNV\_BOT 0x40

Vertical align to bottom.

• #define GSLC\_ALIGNH\_LEFT 0x01

Horizontal align to left.

#define GSLC\_ALIGNH\_MID 0x02

Horizontal align to middle.

#define GSLC\_ALIGNH\_RIGHT 0x04

Horizontal align to right.

- #define GSLC\_ALIGN\_TOP\_LEFT GSLC\_ALIGNH\_LEFT | GSLC\_ALIGNV\_TOP
   Align to top-left.
- #define GSLC\_ALIGN\_TOP\_MID GSLC\_ALIGNH\_MID | GSLC\_ALIGNV\_TOP Align to middle of top.
- #define GSLC\_ALIGN\_TOP\_RIGHT GSLC\_ALIGNH\_RIGHT | GSLC\_ALIGNV\_TOP
   Align to top-right.
- #define GSLC\_ALIGN\_MID\_LEFT GSLC\_ALIGNH\_LEFT | GSLC\_ALIGNV\_MID Align to middle of left side.
- #define GSLC\_ALIGN\_MID\_MID GSLC\_ALIGNH\_MID | GSLC\_ALIGNV\_MID Align to center.
- #define GSLC\_ALIGN\_MID\_RIGHT GSLC\_ALIGNH\_RIGHT | GSLC\_ALIGNV\_MID Align to middle of right side.
- #define GSLC\_ALIGN\_BOT\_LEFT GSLC\_ALIGNH\_LEFT | GSLC\_ALIGNV\_BOT Align to bottom-left.
- #define GSLC\_ALIGN\_BOT\_MID GSLC\_ALIGNH\_MID | GSLC\_ALIGNV\_BOT Align to middle of bottom.
- #define GSLC\_ALIGN\_BOT\_RIGHT GSLC\_ALIGNH\_RIGHT | GSLC\_ALIGNV\_BOT Align to bottom-right.
- #define GSLC\_COL\_RED\_DK4 (gslc\_tsColor) {128, 0, 0}

Red (dark4)

#define GSLC\_COL\_RED\_DK3 (gslc\_tsColor) {160, 0, 0}

Red (dark3)

#define GSLC\_COL\_RED\_DK2 (gslc\_tsColor) {192, 0, 0}

Red (dark2)

#define GSLC\_COL\_RED\_DK1 (gslc\_tsColor) {224, 0, 0}

Red (dark1)

#define GSLC\_COL\_RED (gslc\_tsColor) {255, 0, 0}

Red.

#define GSLC\_COL\_RED\_LT1 (gslc\_tsColor) {255, 32, 32}

Red (light1)

• #define GSLC\_COL\_RED\_LT2 (gslc\_tsColor) {255, 64, 64}

Red (light2)

```
#define GSLC_COL_RED_LT3 (gslc_tsColor) {255, 96, 96}
     Red (light3)
• #define GSLC COL RED LT4 (gslc tsColor) {255,128,128}
     Red (light4)
#define GSLC_COL_GREEN_DK4 (gslc_tsColor) { 0,128, 0}
     Green (dark4)
• #define GSLC_COL_GREEN_DK3 (gslc_tsColor) { 0,160, 0}
     Green (dark3)
#define GSLC_COL_GREEN_DK2 (gslc_tsColor) { 0,192, 0}
     Green (dark2)

    #define GSLC COL GREEN DK1 (gslc tsColor) { 0,224, 0}

     Green (dark1)

    #define GSLC_COL_GREEN (gslc_tsColor) { 0,255, 0}

     Green.
#define GSLC_COL_GREEN_LT1 (gslc_tsColor) { 32,255, 32}
     Green (light1)
#define GSLC_COL_GREEN_LT2 (gslc_tsColor) { 64,255, 64}
     Green (light2)
#define GSLC_COL_GREEN_LT3 (gslc_tsColor) { 96,255, 96}
     Green (light3)
#define GSLC_COL_GREEN_LT4 (gslc_tsColor) {128,255,128}
     Green (light4)
#define GSLC_COL_BLUE_DK4 (gslc_tsColor) { 0, 0,128}
     Blue (dark4)
#define GSLC_COL_BLUE_DK3 (gslc_tsColor) { 0, 0,160}
     Blue (dark3)
#define GSLC_COL_BLUE_DK2 (gslc_tsColor) { 0, 0,192}
     Blue (dark2)
• #define GSLC_COL_BLUE_DK1 (gslc_tsColor) { 0, 0,224}
     Blue (dark1)
#define GSLC_COL_BLUE (gslc_tsColor) { 0, 0,255}
     Blue.
#define GSLC_COL_BLUE_LT1 (gslc_tsColor) { 32, 32,255}
     Blue (light1)
#define GSLC_COL_BLUE_LT2 (gslc_tsColor) { 64, 64,255}
     Blue (light2)
#define GSLC COL BLUE LT3 (gslc tsColor) { 96, 96,255}
     Blue (light3)
#define GSLC_COL_BLUE_LT4 (gslc_tsColor) {128,128,255}
     Blue (light4)
• #define GSLC_COL_BLACK (gslc_tsColor) { 0, 0, 0}
#define GSLC_COL_GRAY_DK3 (gslc_tsColor) { 32, 32, 32}
     Gray (dark)
#define GSLC_COL_GRAY_DK2 (gslc_tsColor) { 64, 64, 64}
     Gray (dark)
#define GSLC_COL_GRAY_DK1 (gslc_tsColor) { 96, 96, 96}
     Gray (dark)

    #define GSLC COL GRAY (gslc tsColor) {128,128,128}
```

#define GSLC\_COL\_GRAY\_LT1 (gslc\_tsColor) {160,160,160}

Gray (light1)

#define GSLC\_COL\_GRAY\_LT2 (gslc\_tsColor) {192,192,192}

Gray (light2)

• #define GSLC COL GRAY LT3 (gslc tsColor) {224,224,224}

Gray (light3)

#define GSLC\_COL\_WHITE (gslc\_tsColor) {255,255,255}

White

• #define GSLC COL YELLOW (gslc tsColor) {255,255,0}

Yellow

#define GSLC\_COL\_YELLOW\_DK (gslc\_tsColor) {64,64,0}

Yellow (dark)

• #define GSLC\_COL\_PURPLE (gslc\_tsColor) {128,0,128}

Purple.

#define GSLC\_COL\_CYAN (gslc\_tsColor) {0,255,255}

Cvan.

#define GSLC COL MAGENTA (gslc tsColor) {255,0,255}

Magenta.

#define GSLC\_COL\_TEAL (gslc\_tsColor) {0,128,128}

Teal

#define GSLC\_COL\_ORANGE (gslc\_tsColor) {255,165,0}

Orange.

• #define GSLC\_COL\_BROWN (gslc\_tsColor) {165,42,42}

Brown.

#define GSLC\_DEBUG\_PRINT(sFmt,...)

Macro to enable optional debug output.

#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\_ElemCreateBox\_P(pGui, nElemId, nPage, nX, nY, nW, nH, colFrame, colFill, bFrameEn, bFillEn)

Create a read-only box 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 \*pvElem)

Callback function for element drawing.

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

Callback function for element touch tracking.

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

Callback function for element tick.

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 = 0, GSLC_ID_NONE = -1999, GSLC_ID_AUTO, GSLC_ID_TEMP,
     GSLC ID AUTO BASE = 16384 }
        Element ID enumerations.

    enum gslc tePageId { GSLC PAGE USER BASE = 0, GSLC PAGE NONE = -2999 }

        Page ID enumerations.

    enum gslc teGroupId { GSLC GROUP ID USER BASE = 0, GSLC GROUP ID NONE = -6999 }

        Group ID enumerations.
   • enum gslc_teFontId { GSLC_FONT_USER_BASE = 0, GSLC_FONT_NONE = -4999 }
        Font ID enumerations.

    enum gslc teElemInd { GSLC IND NONE = -9999, GSLC IND FIRST = 0 }

        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 = 0x1000 }
        Flement type.

    enum gslc teTouch {

     GSLC TOUCH NONE = 0, GSLC TOUCH DOWN = (1 << 4), GSLC TOUCH MOVE = (1 << 5), GSLC \leftrightarrow
     TOUCH UP = (1 << 6),
     GSLC_TOUCH_IN = (1<<0), GSLC_TOUCH_OUT = (1<<1), GSLC_TOUCH_INOUT_MASK = GSLC_←
     TOUCH_IN | GSLC_TOUCH_OUT, GSLC_TOUCH_DOWN_IN = GSLC_TOUCH_DOWN | GSLC_TOUC
     H IN,
     GSLC TOUCH MOVE IN = GSLC TOUCH MOVE | GSLC TOUCH IN, GSLC TOUCH MOVE OUT =
     GSLC_TOUCH_MOVE | GSLC_TOUCH_OUT, GSLC_TOUCH_UP_IN = GSLC_TOUCH_UP | GSLC_TO⊷
     UCH IN, GSLC TOUCH UP OUT = GSLC TOUCH UP | GSLC TOUCH OUT }
        Touch event type for element touch tracking.

    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 teElemRefFlags { GSLC ELEMREF NONE = 0, GSLC ELEMREF SRC RAM = (1<<0), GS</li>

     LC_ELEMREF_SRC_PROG = (2<<0), GSLC_ELEMREF_SRC = (7<<0) }
        Element reference flags: Describes characteristics of an element.

    enum gslc telmgRefFlags {

     GSLC_IMGREF_NONE = 0, GSLC_IMGREF_SRC_FILE = (1<<0), GSLC_IMGREF_SRC_SD = (2<<0),
     GSLC_IMGREF_SRC_RAM = (3 << 0),
     GSLC_IMGREF_SRC_PROG = (4 << 0), GSLC_IMGREF_FMT_BMP24 = (1 << 4), GSLC_IMGREF_FM\leftrightarrow
     T_BMP16 = (2 << 4), GSLC_IMGREF_FMT_RAW1 = (3 << 4),
     GSLC_IMGREF_SRC = (7 << 0), GSLC_IMGREF_FMT = (7 << 4) 
        Image reference flags: Describes characteristics of an image reference.

    enum gslc teTxtFlags {
```

GSLC TXT MEM RAM = (0 << 0), GSLC TXT MEM PROG = (1 << 0), GSLC TXT ALLOC NONE =

 $GSLC_TXT_ALLOC_EXT = (2 << 2), GSLC_TXT_MEM = (3 << 0), GSLC_TXT_ALLOC = (3 << 2), GSLC_TXT_ALLOC = (3 << 0), GSLC_TXT_ALLOC =$ 

(0 < < 2), GSLC TXT ALLOC INT = (1 < < 2),

\_TXT\_DEFAULT = GSLC\_TXT\_MEM\_RAM | GSLC\_TXT\_ALLOC\_NONE }

Text reference flags: Describes the characteristics of a text string (ie.

enum gslc\_teDebugPrintState { GSLC\_DEBUG\_PRINT\_NORM, GSLC\_DEBUG\_PRINT\_TOKEN, GSLC\_
DEBUG\_PRINT\_UINT16, GSLC\_DEBUG\_PRINT\_STR }

## **Functions**

char \* gslc GetVer (gslc tsGui \*pGui)

Get the GUIslice version number.

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.

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\_teEventType eType, uint8\_t nSubType, void \*pvScope, void \*pvData)

  Create an event structure.
- bool gslc IsInRect (int16 t nSelX, int16 t nSelY, gslc tsRect rRect)

Determine if a coordinate is inside of a rectangular region.

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 (gslc\_tsGui \*pGui, 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\_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\_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 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.

Draw an arbitrary line using Bresenham's algorithm.

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

bool gslc\_FontAdd (gslc\_tsGui \*pGui, int16\_t nFontId, const char \*acFontName, 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\_PageSetEventFunc (gslc\_tsPage \*pPage, GSLC\_CB\_EVENT funcCb)

Assign the event callback function for a page.

int gslc\_GetPageCur (gslc\_tsGui \*pGui)

Fetch the current page ID.

void gslc\_SetPageCur (gslc\_tsGui \*pGui, int16\_t nPageId)

Select a new page for display.

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 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.

void gslc\_PageAdd (gslc\_tsGui \*pGui, int16\_t nPageId, gslc\_tsElem \*psElem, uint16\_t nMaxElem, gslc\_ts
 ElemRef \*psElemRef, uint16 t nMaxElemRef)

Add a page to the GUI.

• gslc\_tsPage \* gslc\_PageFindByld (gslc\_tsGui \*pGui, int16\_t nPageId)

Find a page in the GUI by its ID.

gslc\_tsElem \* gslc\_PageFindElemByld (gslc\_tsGui \*pGui, int16\_t nPageId, int16\_t nElemId)

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

void gslc PageRedrawCalc (gslc tsGui \*pGui)

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

 gslc\_tsElem \* 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\_tsElem \* 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\_tsElem \* gslc\_ElemCreateBtnImg (gslc\_tsGui \*pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsRect r←
 Elem, gslc\_tsImgRef sImgRef, gslc\_tsImgRef sImgRefSel, GSLC\_CB\_TOUCH cbTouch)

Create a graphical Button Element.

• gslc\_tsElem \* gslc\_ElemCreateBox (gslc\_tsGui \*pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsRect rElem)

Create a Box Element.

gslc\_tsElem \* 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\_tsElem \* gslc\_ElemCreateImg (gslc\_tsGui \*pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsRect rElem, gslc\_tsImgRef sImgRef)

Create an image Element.

int gslc ElemGetId (gslc tsElem \*pElem)

Get an Element ID from an element structure.

void gslc\_ElemSetFillEn (gslc\_tsElem \*pElem, bool bFillEn)

Set the fill state for an Element.

void gslc ElemSetFrameEn (gslc tsElem \*pElem, bool bFrameEn)

Set the frame state for an Element.

 void gslc\_ElemSetCol (gslc\_tsElem \*pElem, gslc\_tsColor colFrame, gslc\_tsColor colFill, gslc\_tsColor col← FillGlow)

Update the common color selection for an Element.

void gslc\_ElemSetGlowCol (gslc\_tsElem \*pElem, gslc\_tsColor colFrameGlow, gslc\_tsColor colFillGlow, gslc tsColor colTxtGlow)

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

void gslc\_ElemSetGroup (gslc\_tsElem \*pElem, int nGroupId)

Set the group ID for an element.

int gslc\_ElemGetGroup (gslc\_tsElem \*pElem)

Get the group ID for an element.

void gslc\_ElemSetTxtAlign (gslc\_tsElem \*pElem, unsigned nAlign)

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

void gslc\_ElemSetTxtMargin (gslc\_tsElem \*pElem, unsigned nMargin)

Set the margin around of a textual element.

void gslc ElemSetTxtStr (gslc tsElem \*pElem, const char \*pStr)

Update the text string associated with an Element ID.

void gslc\_ElemSetTxtCol (gslc\_tsElem \*pElem, gslc\_tsColor colVal)

Update the text string color associated with an Element ID.

void gslc\_ElemSetTxtMem (gslc\_tsElem \*pElem, gslc\_teTxtFlags eFlags)

Update the text string location in memory.

void gslc\_ElemUpdateFont (gslc\_tsGui \*pGui, gslc\_tsElem \*pElem, int nFontId)

Update the Font selected for an Element's text.

void gslc\_ElemSetRedraw (gslc\_tsElem \*pElem, bool bRedraw)

Update the need-redraw status for an element.

bool gslc\_ElemGetRedraw (gslc\_tsElem \*pElem)

Get the need-redraw status for an element.

void gslc ElemSetGlowEn (gslc tsElem \*pElem, bool bGlowEn)

Update the glowing enable for an element.

• void gslc\_ElemSetStyleFrom (gslc\_tsElem \*pElemSrc, gslc\_tsElem \*pElemDest)

Copy style settings from one element to another.

bool gslc\_ElemGetGlowEn (gslc\_tsElem \*pElem)

Get the glowing enable for an element.

void gslc ElemSetGlow (gslc tsElem \*pElem, bool bGlowing)

Update the glowing indicator for an element.

bool gslc\_ElemGetGlow (gslc\_tsElem \*pElem)

Get the glowing indicator for an element.

• void gslc\_ElemSetEventFunc (gslc\_tsElem \*pElem, GSLC\_CB\_EVENT funcCb)

Assign the event callback function for a element.

• void gslc\_ElemSetDrawFunc (gslc\_tsElem \*pElem, GSLC\_CB\_DRAW funcCb)

Assign the drawing callback function for an element.

void gslc\_ElemSetTickFunc (gslc\_tsElem \*pElem, GSLC\_CB\_TICK funcCb)

Assign the tick callback function for an element.

bool gslc\_ElemOwnsCoord (gslc\_tsElem \*pElem, int16\_t nX, int16\_t nY, bool bOnlyClickEn)

Determine if a coordinate is inside of an element.

bool gslc\_ElemEvent (void \*pvGui, gslc\_tsEvent sEvent)

Common event handler function for an element.

void gslc\_ElemDraw (gslc\_tsGui \*pGui, int16\_t nPageId, int16\_t nElemId)

Draw an element to the active display.

void gslc\_CollectReset (gslc\_tsCollect \*pCollect, gslc\_tsElem \*asElem, uint16\_t nElemMax, gslc\_tsElemRef
 \*asElemRef, uint16\_t nElemRefMax)

Reset the members of an element collection.

Add an element to a collection.

bool gslc CollectGetRedraw (gslc tsCollect \*pCollect)

Determine if any elements in a collection need redraw.

gslc tsElem \* gslc CollectFindElemById (gslc tsCollect \*pCollect, int16 t nElemId)

Find an element in a collection by its Element ID.

gslc\_tsElem \* gslc\_CollectFindElemFromCoord (gslc\_tsCollect \*pCollect, int16\_t nX, int16\_t nY)

Find an element in a collection by a coordinate coordinate.

int gslc CollectGetNextId (gslc tsCollect \*pCollect)

Allocate the next available Element ID in a collection.

gslc\_tsElem \* gslc\_CollectGetElemTracked (gslc\_tsCollect \*pCollect)

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

void gslc\_CollectSetElemTracked (gslc\_tsCollect \*pCollect, gslc\_tsElem \*pElem)

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

void gslc\_CollectSetParent (gslc\_tsCollect \*pCollect, gslc\_tsElem \*pElemParent)

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

• void gslc\_CollectSetEventFunc (gslc\_tsCollect \*pCollect, GSLC\_CB\_EVENT funcCb)

Assign the event callback function for an element 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 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

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)

Initialize the touchscreen device driver.

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\_tsElem \* 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.

• bool gslc\_SetClipRect (gslc\_tsGui \*pGui, gslc\_tsRect \*pRect)

Set the clipping rectangle for further drawing.

void gslc\_ElemSetImage (gslc\_tsGui \*pGui, gslc\_tsElem \*pElem, gslc\_tsImgRef sImgRef, gslc\_tsImgRef sImgRef, gslc\_tsImgRef

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\_ElemDrawByRef (gslc\_tsGui \*pGui, gslc\_tsElem \*pElem)

Draw an element to the active display.

void gslc\_GuiDestruct (gslc\_tsGui \*pGui)

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

void gslc\_PageDestruct (gslc\_tsPage \*pPage)

Free up any members associated with a page.

void gslc\_CollectDestruct (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.

bool gslc\_ElemSendEventTouch (gslc\_tsGui \*pGui, gslc\_tsElem \*pElemTracked, gslc\_teTouch eTouch, int16 t nX, int16 t nY)

Trigger an element's touch event.

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.

#### 5.3.1 Macro Definition Documentation

5.3.1.1 #define GSLC\_ALIGN\_BOT\_LEFT GSLC\_ALIGNH\_LEFT | GSLC\_ALIGNV\_BOT

Align to bottom-left.

5.3.1.2 #define GSLC\_ALIGN\_BOT\_MID GSLC\_ALIGNH\_MID | GSLC\_ALIGNV\_BOT

Align to middle of bottom.

5.3.1.3 #define GSLC\_ALIGN\_BOT\_RIGHT GSLC\_ALIGNH\_RIGHT | GSLC\_ALIGNV\_BOT

Align to bottom-right.

5.3.1.4 #define GSLC\_ALIGN\_MID\_LEFT GSLC\_ALIGNH\_LEFT | GSLC\_ALIGNV\_MID

Align to middle of left side.

5.3.1.5 #define GSLC\_ALIGN\_MID\_MID GSLC\_ALIGNH\_MID | GSLC\_ALIGNV\_MID

Align to center.

#define GSLC\_ALIGN\_MID\_RIGHT GSLC\_ALIGNH\_RIGHT | GSLC\_ALIGNV\_MID Align to middle of right side. 5.3.1.7 #define GSLC\_ALIGN\_TOP\_LEFT GSLC\_ALIGNH\_LEFT | GSLC\_ALIGNV\_TOP Align to top-left. 5.3.1.8 #define GSLC\_ALIGN\_TOP\_MID GSLC\_ALIGNH\_MID | GSLC\_ALIGNV\_TOP Align to middle of top. 5.3.1.9 #define GSLC\_ALIGN\_TOP\_RIGHT GSLC\_ALIGNH\_RIGHT | GSLC\_ALIGNV\_TOP Align to top-right. 5.3.1.10 #define GSLC\_ALIGNH\_LEFT 0x01 Horizontal align to left. 5.3.1.11 #define GSLC\_ALIGNH\_MID 0x02 Horizontal align to middle. 5.3.1.12 #define GSLC\_ALIGNH\_RIGHT 0x04 Horizontal align to right. 5.3.1.13 #define GSLC\_ALIGNV\_BOT 0x40 Vertical align to bottom. 5.3.1.14 #define GSLC\_ALIGNV\_MID 0x20 Vertical align to middle. 5.3.1.15 #define GSLC\_ALIGNV\_TOP 0x10 Vertical align to top. 5.3.1.16 #define GSLC\_COL\_BLACK (gslc\_tsColor) { 0, 0, 0}

5.3.1.17 #define GSLC\_COL\_BLUE (gslc\_tsColor) { 0, 0,255} Blue.

Black.

```
5.3.1.18 #define GSLC_COL_BLUE_DK1 (gslc_tsColor) { 0, 0,224}
Blue (dark1)
5.3.1.19 #define GSLC_COL_BLUE_DK2 (gslc_tsColor) { 0, 0,192}
Blue (dark2)
5.3.1.20 #define GSLC_COL_BLUE_DK3 (gslc_tsColor) { 0, 0,160}
Blue (dark3)
5.3.1.21 #define GSLC_COL_BLUE_DK4 (gslc_tsColor) { 0, 0,128}
Blue (dark4)
5.3.1.22 #define GSLC_COL_BLUE_LT1 (gslc_tsColor) { 32, 32,255}
Blue (light1)
5.3.1.23 #define GSLC_COL_BLUE_LT2 (gslc_tsColor) { 64, 64,255}
Blue (light2)
5.3.1.24 #define GSLC_COL_BLUE_LT3 (gslc_tsColor) { 96, 96,255}
Blue (light3)
5.3.1.25 #define GSLC_COL_BLUE_LT4 (gslc_tsColor) {128,128,255}
Blue (light4)
5.3.1.26 #define GSLC_COL_BROWN (gslc_tsColor) {165,42,42}
Brown.
5.3.1.27 #define GSLC_COL_CYAN (gslc_tsColor) {0,255,255}
Cyan.
5.3.1.28 #define GSLC_COL_GRAY (gslc_tsColor) {128,128,128}
Gray.
5.3.1.29 #define GSLC_COL_GRAY_DK1 (gslc_tsColor) { 96, 96, 96}
Gray (dark)
```

```
5.3.1.30 #define GSLC_COL_GRAY_DK2 (gslc_tsColor) { 64, 64, 64}
Gray (dark)
5.3.1.31 #define GSLC_COL_GRAY_DK3 (gslc_tsColor) { 32, 32, 32}
Gray (dark)
5.3.1.32 #define GSLC_COL_GRAY_LT1 (gslc_tsColor) {160,160,160}
Gray (light1)
5.3.1.33 #define GSLC_COL_GRAY_LT2 (gslc_tsColor) {192,192,192}
Gray (light2)
5.3.1.34 #define GSLC_COL_GRAY_LT3 (gslc_tsColor) {224,224,224}
Gray (light3)
5.3.1.35 #define GSLC_COL_GREEN (gslc_tsColor) { 0,255, 0}
Green.
5.3.1.36 #define GSLC_COL_GREEN_DK1 (gslc_tsColor) { 0,224, 0}
Green (dark1)
5.3.1.37 #define GSLC_COL_GREEN_DK2 (gslc_tsColor) { 0,192, 0}
Green (dark2)
5.3.1.38 #define GSLC_COL_GREEN_DK3 (gslc_tsColor) { 0,160, 0}
Green (dark3)
5.3.1.39 #define GSLC_COL_GREEN_DK4 (gslc_tsColor) { 0,128, 0}
Green (dark4)
5.3.1.40 #define GSLC_COL_GREEN_LT1 (gslc_tsColor) { 32,255, 32}
Green (light1)
5.3.1.41 #define GSLC_COL_GREEN_LT2 (gslc_tsColor) { 64,255, 64}
Green (light2)
```

```
5.3.1.42 #define GSLC_COL_GREEN_LT3 (gslc_tsColor) { 96,255, 96}
Green (light3)
5.3.1.43 #define GSLC_COL_GREEN_LT4 (gslc_tsColor) {128,255,128}
Green (light4)
5.3.1.44 #define GSLC_COL_MAGENTA (gslc_tsColor) {255,0,255}
Magenta.
5.3.1.45 #define GSLC_COL_ORANGE (gslc_tsColor) {255,165,0}
Orange.
5.3.1.46 #define GSLC_COL_PURPLE (gslc_tsColor) {128,0,128}
Purple.
5.3.1.47 #define GSLC_COL_RED (gslc_tsColor) {255, 0, 0}
Red.
5.3.1.48 #define GSLC_COL_RED_DK1 (gslc_tsColor) {224, 0, 0}
Red (dark1)
5.3.1.49 #define GSLC_COL_RED_DK2 (gslc_tsColor) {192, 0, 0}
Red (dark2)
5.3.1.50 #define GSLC_COL_RED_DK3 (gslc_tsColor) {160, 0, 0}
Red (dark3)
5.3.1.51 #define GSLC_COL_RED_DK4 (gslc_tsColor) {128, 0, 0}
Red (dark4)
5.3.1.52 #define GSLC_COL_RED_LT1 (gslc_tsColor) {255, 32, 32}
Red (light1)
5.3.1.53 #define GSLC_COL_RED_LT2 (gslc_tsColor) {255, 64, 64}
Red (light2)
```

```
5.3.1.54 #define GSLC_COL_RED_LT3 (gslc_tsColor) {255, 96, 96}
Red (light3)
5.3.1.55 #define GSLC_COL_RED_LT4 (gslc_tsColor) {255,128,128}
Red (light4)
5.3.1.56 #define GSLC_COL_TEAL (gslc_tsColor) {0,128,128}
Teal.
        #define GSLC_COL_WHITE (gslc_tsColor) {255,255,255}
5.3.1.57
White.
5.3.1.58
        #define GSLC_COL_YELLOW (gslc_tsColor) {255,255,0}
Yellow.
5.3.1.59 #define GSLC_COL_YELLOW_DK (gslc_tsColor) {64,64,0}
Yellow (dark)
5.3.1.60 #define GSLC_DEBUG_PRINT( sFmt, ... )
Value:
            if (DEBUG_ERR) {
              gslc_DebugPrintf(sFmt,__VA_ARGS__);
          } while (0)
```

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

5.3.1.61 #define gslc\_ElemCreateBox\_P( pGui, nElemId, nPage, nX, nY, nW, nH, colFrame, colFill, bFrameEn, bFillEn

Value:

```
static const gslc_tsElem sElem##nElemId = {
        nElemId,
        true,
        GSLC_TYPE_BOX,
        (gslc_tsRect) {nX,nY,nW,nH},
GSLC_GROUP_ID_NONE, false, false, bFrameEn, bFillEn,
colFrame, colFill, GSLC_COL_BLACK, GSLC_COL_BLACK,
(gslc_tsImgRef) {NULL, NULL, GSLC_IMGREF_NONE, NULL},
         (gslc_tsImgRef) {NULL, NULL, GSLC_IMGREF_NONE, NULL},
        NULL,
        NULL,
        Ο,
        GSLC_TXT_DEFAULT,
        GSLC_COL_WHITE,
        GSLC COL WHITE.
        GSLC_ALIGN_MID_MID,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        false,
        false,
   gslc_ElemAdd(pGui,nPage,(gslc_tsElem*)&sElem##nElemId,
        GSLC_ELEMREF_SRC_RAM);
```

## Create a read-only box element.

#### **Parameters**

in	pGui	Pointer to GUI
in	nElemId	Unique element ID to assign
in	nPage	Page ID to attach element to
in	nX	X coordinate of element
in	nY	Y coordinate of element
in	nW	Width of element
in	nH	Height of element
in	colFrame	Color for the frame
in	colFill	Color for the fill
in	bFrameEn	True if framed, false otherwise
in	bFillEn	True if filled, false otherwise

5.3.1.62 #define gslc\_ElemCreateTxt\_P( pGui, nElemId, nPage, nX, nY, nW, nH, strTxt, pFont, colTxt, colFrame, colFill, nAlignTxt, bFrameEn, bFillEn )

## Value:

```
static const char str##nElemId[] = strTxt;
   static const gslc_tsElem sElem##nElemId = {
         nElemId,
          true,
          GSLC_TYPE_TXT,
         (gslc_tsRect) {nX,nY,nW,nH},
GSLC_GROUP_ID_NONE, false, false, bFrameEn, bFillEn,
colFrame, colFill, GSLC_COL_BLACK, GSLC_COL_BLACK,
(gslc_tsImgRef) {NULL, NULL, GSLC_IMGREF_NONE, NULL},
(gslc_tsImgRef) {NULL, NULL, GSLC_IMGREF_NONE, NULL},
          NULL.
          (char*) str##nElemId,
          0,
          (gslc_teTxtFlags) (GSLC_TXT_MEM_RAM |
          GSLC_TXT_ALLOC_EXT),
          colTxt,
          colTxt,
          nAlignTxt,
          Ο,
          pFont,
          NULL,
```

```
NULL,
NULL,
NULL,
NULL,
NULL,
false,
false,
};
gslc_ElemAdd(pGui,nPage,(gslc_tsElem*)&sElem##nElemId,
GSLC_ELEMREF_SRC_RAM);
```

## Create a read-only text element.

## **Parameters**

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

# 5.3.2 Typedef Documentation

5.3.2.1 typedef int16\_t(\* GSLC\_CB\_DEBUG\_OUT)(char ch)

5.3.2.2 typedef bool(\* GSLC\_CB\_DRAW)(void \*pvGui, void \*pvElem)

Callback function for element drawing.

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

Callback function for element drawing.

5.3.2.4 typedef bool(\* GSLC\_CB\_TICK)(void \*pvGui, void \*pvElem)

Callback function for element tick.

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

Callback function for element touch tracking.

5.3.2.6 typedef struct gslc\_tsColor gslc\_tsColor

Color structure. Defines RGB triplet.

5.3.2.7 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.

5.3.2.8 typedef struct gslc\_tsEvent gslc\_tsEvent

Event structure.

5.3.2.9 typedef struct gslc\_tsEventTouch gslc\_tsEventTouch

Structure used to pass touch data through event.

5.3.2.10 typedef struct gslc\_tsPt gslc\_tsPt

Define point coordinates.

5.3.2.11 typedef struct gslc\_tsRect gslc\_tsRect

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

- 5.3.3 Enumeration Type Documentation
- 5.3.3.1 enum gslc\_teDebugPrintState

Enumerator

GSLC\_DEBUG\_PRINT\_NORM
GSLC\_DEBUG\_PRINT\_TOKEN
GSLC\_DEBUG\_PRINT\_UINT16
GSLC\_DEBUG\_PRINT\_STR

5.3.3.2 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)

### 5.3.3.3 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.

# 5.3.3.4 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 stored in RAM (internal element array)

**GSLC\_ELEMREF\_SRC\_PROG** Element is stored in program memory (PROGMEM, read-only, external to element array)

GSLC\_ELEMREF\_SRC Mask for Source flags.

5.3.3.5 enum gslc\_teEventSubType

Event sub-types.

### Enumerator

GSLC\_EVTSUB\_NONE

GSLC\_EVTSUB\_DRAW\_NEEDED

GSLC\_EVTSUB\_DRAW\_FORCE

# 5.3.3.6 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.
```

# 5.3.3.7 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.
```

# 5.3.3.8 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.
```

# 5.3.3.9 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.
```

### 5.3.3.10 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.
```

```
5.3.3.11 enum gslc_teTouch
```

Touch event type for element touch tracking.

#### Enumerator

```
GSLC_TOUCH_NONE No touch event active.
```

GSLC\_TOUCH\_DOWN Touch event (down)

**GSLC\_TOUCH\_MOVE** Touch event (move)

GSLC\_TOUCH\_UP Touch event (up)

GSLC\_TOUCH\_IN Touch event inside element.

GSLC\_TOUCH\_OUT Touch event outside element.

GSLC\_TOUCH\_INOUT\_MASK Mask for in/out state.

GSLC\_TOUCH\_DOWN\_IN Touch down inside element (start tracking)

GSLC\_TOUCH\_MOVE\_IN Touch move inside tracked element.

GSLC\_TOUCH\_MOVE\_OUT Touch move outside tracked element.

GSLC\_TOUCH\_UP\_IN Touch up inside tracked element.

GSLC\_TOUCH\_UP\_OUT Touch up outside tracked element.

#### 5.3.3.12 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\_MEM Mask for updating text memory type.

GSLC\_TXT\_ALLOC Mask for updating location of text string buffer allocation.

GSLC\_TXT\_DEFAULT

### 5.3.3.13 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.

#### 5.3.4 Function Documentation

5.3.4.1 bool gslc\_ClipLine ( gslc\_tsRect \* pClipRect, int16\_t \* pnX0, int16\_t \* pnY0, 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

5.3.4.2 bool gslc\_ClipPt ( gslc\_tsRect \* pClipRect, int16\_t nX, int16\_t nY )

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

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

#### Returns

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

5.3.4.3 bool gslc\_ClipRect ( gslc\_tsRect \* pClipRect, gslc\_tsRect \* pRect )

Perform basic clipping of a rectangle to a clipping region.

• Coordinates in parameter rect are modified to fit the region

#### **Parameters**

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

#### Returns

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

5.3.4.4 void gslc\_CollectDestruct ( gslc\_tsCollect \* pCollect )

Free up any members associated with an element collection.

# **Parameters**

in	pCollect	Pointer to collection
----	----------	-----------------------

# Returns

none

5.3.4.5 gslc\_tsElem\* gslc\_CollectElemAdd ( 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	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 in the collection that has been added or NULL if there was an error

5.3.4.6 bool gslc\_CollectEvent ( void \* pvGui, gslc\_tsEvent sEvent )

Common event handler function for an element collection.

in	pvGui	Void pointer to GUI
in	sEvent	Event data structure

#### Returns

true if success, false if fail

5.3.4.7 gslc\_tsElem\* gslc\_CollectFindElemByld ( gslc\_tsCollect \* pCollect, int16\_t nElemId )

Find an element in a collection by its Element ID.

#### **Parameters**

in	pCollect	Pointer to the collection
in	nElemId	Element ID to search for

#### Returns

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

5.3.4.8 gslc\_tsElem\* gslc\_CollectFindElemFromCoord ( 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	pCollect	Pointer to the collection
in	nX	Absolute X coordinate to use for search
in	nY	Absolute Y coordinate to use for search

#### Returns

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

5.3.4.9  $gslc\_tsElem* gslc\_CollectGetElemTracked ( <math>gslc\_tsCollect* pCollect )$ 

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

#### **Parameters**

in	pCollect	Pointer to the collection

### Returns

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

5.3.4.10 int gslc\_CollectGetNextId ( gslc\_tsCollect \* pCollect )

Allocate the next available Element ID in a collection.

#### **Parameters**

in	pCollect	Pointer to the collection

#### Returns

Element ID that is reserved for use

5.3.4.11 bool gslc\_CollectGetRedraw ( gslc\_tsCollect \* pCollect )

Determine if any elements in a collection need redraw.

#### **Parameters**

in	pCollect	Pointer to Element collection
----	----------	-------------------------------

### Returns

True if redraw required, false otherwise

5.3.4.12 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

5.3.4.13 void gslc\_CollectSetElemTracked ( gslc\_tsCollect \* pCollect, gslc\_tsElem \* pElem )

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

#### **Parameters**

in	pCollect	Pointer to the collection
in	pElem	Ptr to element to mark as being tracked

### Returns

none

 $5.3.4.14 \quad \text{void gslc\_CollectSetEventFunc (} \quad \text{gslc\_tsCollect} \\ * \textit{pCollect,} \quad \text{GSLC\_CB\_EVENT} \quad \textit{funcCb} \quad \text{)}$ 

Assign the event callback function for an element collection.

in	pCollect	Pointer to collection
in	funcCb	Function pointer to event routine (or NULL for default))

### Returns

none

 $5.3.4.15 \quad \text{void gslc\_CollectSetParent (} \ \ \text{gslc\_tsCollect} * \textit{pCollect}, \ \ \text{gslc\_tsElem} * \textit{pElemParent} \ )$ 

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	pCollect	Pointer to the collection
in	pElemParent	Ptr to element that is the parent

#### Returns

none

5.3.4.16 void gslc\_CollectTouch ( gslc\_tsGui \* pGui, gslc\_tsCollect \* pCollect, gslc\_tsEventTouch \* pEventTouch )

Handle touch events within the element collection.

## **Parameters**

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

# Returns

none

5.3.4.17 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()

in	pFmt	Format string to use for printing

in		Variable parameter list
----	--	-------------------------

# Returns

none

5.3.4.18 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

5.3.4.19 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

5.3.4.20 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

5.3.4.21 void gslc\_DrawLine ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nX1, 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

5.3.4.22 void gslc\_DrawLineH ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, uint16\_t nW, gslc\_tsColor nCol )

Draw a horizontal line.

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

### **Parameters**

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

# Returns

none

 $5.3.4.23 \quad \text{void gslc\_DrawLineV (} \ \ \text{gslc\_tsGui} * \textit{pGui,} \ \ \text{int16\_t} \ \textit{nX,} \ \ \text{int16\_t} \ \textit{nY,} \ \ \text{uint16\_t} \ \textit{nH,} \ \ \text{gslc\_tsColor} \ \textit{nCol} \ \ \text{)}$ 

Draw a vertical line.

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

# **Parameters**

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

# Returns

none

5.3.4.24 void gslc\_DrawSetPixel (  $gslc_tsGui*pGui$ , int16\_t nX, int16\_t nY,  $gslc_tsColor nCol$ )

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

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

#### **Parameters**

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

#### Returns

none

5.3.4.25 gslc\_tsElem\* gslc\_telemAdd ( gslc\_tsGui \* pGui, int16\_t nPageld, 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	nPageld	Page ID to add element to (GSLC_PAGE_NONE to skip in case of temporary
		creation for compound elements)
in	pElem	Pointer to Element to add
in	eFlags	Flags describing the element (eg. whether the element should be stored in
		internal RAM array or is located in Flash/PROGMEM).

## Returns

Pointer to Element or NULL if fail

5.3.4.26 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	nPageld	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

5.3.4.27 gslc\_tsElem\* 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	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 box size

#### Returns

Pointer to the Element or NULL if failure

5.3.4.28 gslc\_tsElem\* 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 or NULL if failure

5.3.4.29 gslc\_tsElem\* 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
	ρ σ.α.	· cilitar to cita

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
in	cbTouch	Callback for touch events

### Returns

Pointer to the Element or NULL if failure

5.3.4.30 gslc\_tsElem\* 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	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 box size
in	sImgRef	Image reference to load

# Returns

Pointer to the Element or NULL if failure

5.3.4.31 gslc\_tsElem\* gslc\_ElemCreateLine ( gslc\_tsGui \* pGui, int16\_t nElemId, int16\_t nPage, int16\_t nX0, int16\_t nY0, int16\_t nX1, int16\_t nY1 )

Create a Line Element.

· Draws a line with fill color

### **Parameters**

in	pGui	Pointer to GUI
in	nElemId	Element ID to assign (016383 or GSLC_ID_AUTO to autogen)
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 or NULL if failure

5.3.4.32 gslc\_tsElem\* 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	nFontld	Font ID to use for text display

### Returns

Pointer to the Element or NULL if failure

5.3.4.33 void gslc\_ElemDestruct ( gslc\_tsElem \* pElem )

Free up any members associated with an element.

#### **Parameters**

	_	
in	pElem	Pointer to element

# Returns

none

5.3.4.34 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	nPageId	ID of page containing element
in	nElemId	ID of element

# Returns

none

5.3.4.35 bool gslc\_ElemDrawByRef (  $gslc_tsGui*pGui, gslc_tsElem*pElem$  )

Draw an element to the active display.

· Element is referenced by an element pointer

#### **Parameters**

in	pGui	Pointer to GUI
in	pElem	Ptr to Element to draw

# Returns

true if success, false otherwise

5.3.4.36 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

5.3.4.37 bool gslc\_ElemGetGlow (  $gslc_tsElem * pElem$  )

Get the glowing indicator for an element.

#### **Parameters**

in	pElem	Pointer to Element

### Returns

True if element is glowing

5.3.4.38 bool gslc\_ElemGetGlowEn ( gslc\_tsElem \* pElem )

Get the glowing enable for an element.

# **Parameters**

in pElem Pointer to Element
-----------------------------

# Returns

True if element supports glowing

5.3.4.39 int gslc\_ElemGetGroup ( gslc\_tsElem \* pElem )

Get the group ID for an element.

in	pElem	Pointer to Element
----	-------	--------------------

### Returns

Group ID or GSLC\_GROUP\_ID\_NONE if unassigned

5.3.4.40 int gslc\_ElemGetId ( gslc\_tsElem \* pElem )

Get an Element ID from an element structure.

#### **Parameters**

in	pElem	Pointer to element structure

#### Returns

ID of element or GSLC ID NONE if not found

5.3.4.41 bool gslc\_ElemGetRedraw ( gslc\_tsElem \* pElem )

Get the need-redraw status for an element.

#### **Parameters**

in	pElem	Pointer to Element
----	-------	--------------------

### Returns

True if redraw required, false otherwise

5.3.4.42 bool gslc\_ElemOwnsCoord ( gslc\_tsElem \* pElem, 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	pElem	Element used for boundary test
in	nX	X coordinate to test
in	nY	Y coordinate to test
in	bOnlyClickEn	Only output true if element was also marked as "clickable" (eg. bClickEn=true)

# Returns

true if inside element, false otherwise

5.3.4.43 bool gslc\_ElemSendEventTouch ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElemTracked, 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	pElemTracked	Pointer to tracked Element (or NULL for none))
in	eTouch	Touch event type
in	nX	X coordinate of event (absolute coordinate)
in	nY	Y coordinate of event (absolute coordinate)

#### Returns

true if success, false if error

5.3.4.44 void gslc\_ElemSetCol (  $gslc_tsElem*pElem*, gslc_tsColor* colFrame*, gslc_tsColor* colFill, gslc_tsColor* colFillGlow*)$ 

Update the common color selection for an Element.

### **Parameters**

in	pElem	Pointer to Element
in	colFrame	Color for the frame
in	colFill	Color for the fill
in	colFillGlow	Color for the fill when glowing

# Returns

none

 $5.3.4.45 \quad \text{void gslc\_ElemSetDrawFunc (} \ \ \text{gslc\_tsElem} * \textit{pElem,} \ \ \text{GSLC\_CB\_DRAW} \ \textit{funcCb} \ )$ 

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	pElem	Pointer to Element
in	funcCb	Function pointer to drawing routine (or NULL for default))

### Returns

none

5.3.4.46 void gslc\_ElemSetEventFunc ( gslc\_tsElem \* pElem, GSLC\_CB\_EVENT funcCb )

Assign the event callback function for a element.

in	pElem	Pointer to element
----	-------	--------------------

in	funcCb	Function pointer to event routine (or NULL for default))

# Returns

none

5.3.4.47 void gslc\_ElemSetFillEn ( gslc\_tsElem \* pElem, bool bFillEn )

Set the fill state for an Element.

### **Parameters**

in	pElem	Pointer to Element
in	bFillEn	True if filled, false otherwise

### Returns

none

5.3.4.48 void gslc\_ElemSetFrameEn (  $gslc_tsElem*pElem$ , bool bFrameEn )

Set the frame state for an Element.

### **Parameters**

in	pElem	Pointer to Element
in	bFrameEn	True if framed, false otherwise

#### Returns

none

5.3.4.49 void gslc\_ElemSetGlow (  $gslc\_tsElem*pElem$ , bool bGlowing )

Update the glowing indicator for an element.

### **Parameters**

in	pElem	Pointer to Element
in	bGlowing	True if element is glowing

## Returns

none

5.3.4.50 void gslc\_ElemSetGlowCol ( gslc\_tsElem \* pElem, gslc\_tsColor colFrameGlow, gslc\_tsColor colFillGlow, gslc\_tsColor colTxtGlow )

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

#### **Parameters**

in	pElem	Pointer to Element
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

5.3.4.51 void gslc\_ElemSetGlowEn ( gslc\_tsElem \* pElem, bool bGlowEn )

Update the glowing enable for an element.

#### **Parameters**

in	pElem	Pointer to Element
in	bGlowEn	True if element should support glowing

### Returns

none

5.3.4.52 void gslc\_ElemSetGroup (  $gslc\_tsElem*pElem$ , int nGroupId )

Set the group ID for an element.

• Typically used to associate radio button elements together

#### **Parameters**

-	in	pElem	Pointer to Element
	in	nGroupId	Group ID to assign

# Returns

none

5.3.4.53 void gslc\_ElemSetImage ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElem, gslc\_tsImgRef sImgRef, gslc\_tsImgRef sImgRefSel )

Set an element to use a bitmap image.

## **Parameters**

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

# Returns

none

5.3.4.54 void gslc\_ElemSetRedraw ( gslc\_tsElem \* pElem, bool bRedraw )

Update the need-redraw status for an element.

in	pElem	Pointer to Element
in	bRedraw	True if redraw required, false otherwise

### Returns

none

5.3.4.55 void gslc\_ElemSetStyleFrom ( gslc\_tsElem \* pElemSrc, gslc\_tsElem \* pElemDest )

Copy style settings from one element to another.

# **Parameters**

in	pElemSrc	Pointer to source Element
in	pElemDest	Pointer to destination Element

# Returns

none

5.3.4.56 void gslc\_ElemSetTickFunc ( gslc\_tsElem \* pElem, 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()

in	pElem	Pointer to Element
in	funcCb	Function pointer to tick routine (or NULL for none))

Returns

none

5.3.4.57 void gslc\_ElemSetTxtAlign (  $gslc_tsElem*pElem*, unsigned* nAlign*)$ 

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

#### **Parameters**

in	pElem	Pointer to Element
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

5.3.4.58 void gslc\_ElemSetTxtCol (  $gslc_tsElem*pElem*, gslc_tsColor* colVal$  )

Update the text string color associated with an Element ID.

# **Parameters**

in	pElem	Pointer to Element
in	colVal	RGB color to change to

# Returns

none

5.3.4.59 void gslc\_ElemSetTxtMargin (  $gslc_tsElem*pElem$ , unsigned nMargin )

Set the margin around of a textual element.

in	pElem	Pointer to Element

lin	nMarain	Number of pixels gap to leave surrounding text
711	inviaigiii	Trainbor or pixolo gap to loave barroanding toxt

### Returns

none

 $5.3.4.60 \quad \text{void gslc\_ElemSetTxtMem ( } \textbf{gslc\_tsElem} * \textbf{pElem}, \textbf{gslc\_teTxtFlags} \textit{eFlags} \textbf{ )}$ 

Update the text string location in memory.

### **Parameters**

in	pElem	Pointer to Element
in	eFlags	Flags associated with text memory location (GSLC_TXT_MEM_*)

### Returns

none

5.3.4.61 void gslc\_ElemSetTxtStr ( gslc\_tsElem \* pElem, const char \* pStr )

Update the text string associated with an Element ID.

### **Parameters**

in	pElem	Pointer to Element
in	pStr	String to copy into element

### Returns

none

5.3.4.62 void gslc\_ElemUpdateFont ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElem, int nFontId )

Update the Font selected for an Element's text.

## **Parameters**

in	pGui	Pointer to GUI
in	pElem	Pointer to Element
in	nFontld	Font ID to select

# Returns

none

5.3.4.63 gslc\_tsEvent gslc\_EventCreate ( gslc\_teEventType eType, uint8\_t nSubType, void \* pvScope, void \* pvData )

Create an event structure.

#### **Parameters**

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

5.3.4.64 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 neg-
		ative)
in	nExpandH	Number of pixels to expand the height (if positive) of contract the height (if
		negative)

### Returns

gslc\_tsRect() with resized dimensions

5.3.4.65 bool gslc\_FontAdd ( gslc\_tsGui \* pGui, int16\_t nFontId, const char \* acFontName, uint16\_t nFontSz )

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

# Parameters

in	pGui	Pointer to GUI
in	nFontld	ID to use when referencing this font
in	acFontName	Filename path to the font
in	nFontSz	Typeface size to use

# Returns

true if load was successful, false otherwise

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

Fetch a font from its ID value.

# **Parameters**

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

# Returns

A pointer to the font structure or NULL if error

 $5.3.4.67 \quad \text{gslc\_tsImgRef gslc\_GetImageFromFile ( const char} * \textit{pFname}, \ \text{gslc\_teImgRefFlags} \textit{eFmt} \ )$ 

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

 $5.3.4.68 \quad \text{gslc\_tsImgRef gslc\_GetImageFromProg ( const unsigned char} * \textit{plmgBuf, gslc\_teImgRefFlags eFmt )}$ 

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

#### **Parameters**

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

### Returns

Loaded image reference

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

Create an image reference to a bitmap in SRAM.

# **Parameters**

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

# Returns

Loaded image reference

5.3.4.70 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

5.3.4.71 int gslc\_GetPageCur (  $gslc_tsGui * pGui$  )

Fetch the current page ID.

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

#### Returns

Page ID

5.3.4.72 bool gslc\_GetTouch (  $gslc_tsGui*pGui$ ,  $int16_t*pnX$ ,  $int16_t*pnY$ ,  $uint16_t*pnPress$  )

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		

### Returns

true if touch event, false otherwise

5.3.4.73 char\* gslc\_GetVer ( gslc\_tsGui \* pGui )

Get the GUIslice version number.

Returns

String containing version number

5.3.4.74 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

5.3.4.75 bool gslc\_lnit ( 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	d 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

5.3.4.76 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

5.3.4.77 bool gslc\_lnitTouch ( gslc\_tsGui \* pGui, const char \* acDev )

Initialize the touchscreen device driver.

# **Parameters**

in	pGui	Pointer t	o GUI								
in	acDev	Device	path	to	touchscreen	(or	""	if	not	applicable))	eg.
		"/dev/inp	ut/toucl	nscre	en"						

# Returns

true if successful

5.3.4.78 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.

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

### Returns

true if inside region, false otherwise

5.3.4.79 bool gslc\_lslnWH (  $gslc_tsGui * pGui$ , 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	pGui	Pointer to GUI			
in	nSelX	pordinate 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

5.3.4.80 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

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 ele-	
		ments, 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

5.3.4.81 void gslc\_PageDestruct ( gslc\_tsPage \* pPage )

Free up any members associated with a page.

### **Parameters**

in	nPaga.	Pointer to Page
TU	pPage	Pointer to Page

# Returns

none

5.3.4.82 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

5.3.4.83 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	nPageId	Page ID to search

# Returns

Ptr to a page or NULL if none found

 $5.3.4.84 \quad \textbf{gslc\_tsElem}* \ \textbf{gslc\_PageFindElemByld} \ ( \ \textbf{gslc\_tsGui}* \ \textbf{pGui}, \ \textbf{int16\_t} \ \textbf{nPageId}, \ \textbf{int16\_t} \ \textbf{nElemId} \ )$ 

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

in	pGui	Pointer to GUI
in	nPageld	Page ID to search
in	nElemId	Element ID to search

#### Returns

Ptr to an element or NULL if none found

5.3.4.85 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

5.3.4.86 void gslc\_PageFlipGo ( gslc\_tsGui \* pGui )

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

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

#### **Parameters**

in	pGui	Pointer to GUI

## Returns

None

5.3.4.87 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

5.3.4.88 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

5.3.4.89 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

5.3.4.90 void gslc\_PageRedrawGo ( gslc\_tsGui \* pGui )

Redraw all elements on the active page.

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

### **Parameters**

in	pGui	Pointer to GUI

# Returns

none

5.3.4.91 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

5.3.4.92 void gslc\_PageSetEventFunc ( gslc\_tsPage \* pPage, GSLC\_CB\_EVENT funcCb )

Assign the event callback function for a page.

in	pPage	Pointer to page
in	funcCb	Function pointer to event routine (or NULL for default))

### Returns

none

5.3.4.93 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

5.3.4.94 void gslc\_ResetElem ( gslc\_tsElem \* pElem )

Initialize an Element struct.

## **Parameters**

in	pElem	Pointer to Element

### Returns

none

5.3.4.95 void gslc\_ResetFont ( gslc\_tsFont \* pFont )

Initialize a Font struct.

# Parameters

in	pFont	Pointer to Font

### Returns

none

5.3.4.96 gslc\_tslmgRef gslc\_ResetImage ( )

Create a blank image reference structure.

# Returns

Image reference struct

5.3.4.97 bool gslc\_SetBkgndColor ( gslc\_tsGui \* pGui, gslc\_tsColor nCol )

Configure the background to use a solid color.

• The background is used when redrawing the entire page

#### **Parameters**

in	pGui	Pointer to GUI
in	nCol	RGB Color to use

#### Returns

true if success, false if fail

5.3.4.98 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

5.3.4.99 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

5.3.4.100 void gslc\_SetPageCur ( gslc\_tsGui \* pGui, int16\_t nPageld )

Select a new page for display.

in	pGui	Pointer to GUI

in	nPageld	Page ID to select as current

# Returns

none

5.3.4.101 void gslc\_TrackTouch ( gslc\_tsGui \* pGui, gslc\_tsPage \* pPage, int16\_t nX, int16\_t nY, uint16\_t nPress )

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

### **Parameters**

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

#### Returns

none

5.3.4.102 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

# 5.3.5 Variable Documentation

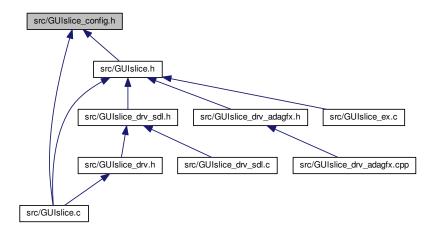
5.3.5.1 GSLC\_CB\_DEBUG\_OUT g\_pfDebugOut

Global debug output function.

• The user assigns this function via gslc\_InitDebug()

# 5.4 src/GUIslice\_config.h File Reference

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



# **Macros**

- #define DRV\_DISP\_SDL1
- #define DRV TOUCH TSLIB
- #define GSLC\_DEV\_FB "/dev/fb1"
- #define GSLC\_DEV\_TOUCH "/dev/input/touchscreen"
- #define GSLC\_DEV\_VID\_DRV "fbcon"
- #define DRV\_SDL\_FIX\_START 1
- #define DRV\_SDL\_MOUSE\_SHOW 0
- #define GSLC\_LOCAL\_STR 1
- #define DEBUG\_ERR 1
- #define ADATOUCH\_SWAP\_XY 1
- #define ADATOUCH FLIP X 0
- #define ADATOUCH\_FLIP\_Y 1
- #define GSLC\_LOCAL\_STR\_LEN 30
- #define GSLC\_BMP\_TRANS\_EN 1
- #define GSLC BMP TRANS RGB 0xFF,0x00,0xFF
- #define GSLC\_USE\_PROGMEM 0

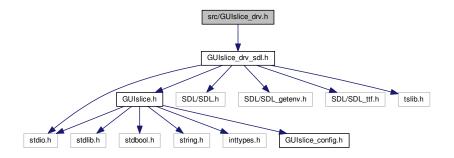
# 5.4.1 Macro Definition Documentation

- 5.4.1.1 #define ADATOUCH\_FLIP\_X 0
- 5.4.1.2 #define ADATOUCH\_FLIP\_Y 1
- 5.4.1.3 #define ADATOUCH\_SWAP\_XY 1
- 5.4.1.4 #define DEBUG\_ERR 1
- 5.4.1.5 #define DRV\_DISP\_SDL1

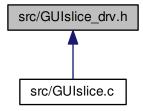
- 5.4.1.6 #define DRV\_SDL\_FIX\_START 1
- 5.4.1.7 #define DRV\_SDL\_MOUSE\_SHOW 0
- 5.4.1.8 #define DRV\_TOUCH\_TSLIB
- 5.4.1.9 #define GSLC\_BMP\_TRANS\_EN 1
- 5.4.1.10 #define GSLC\_BMP\_TRANS\_RGB 0xFF,0x00,0xFF
- 5.4.1.11 #define GSLC\_DEV\_FB "/dev/fb1"
- 5.4.1.12 #define GSLC\_DEV\_TOUCH "/dev/input/touchscreen"
- 5.4.1.13 #define GSLC\_DEV\_VID\_DRV "fbcon"
- 5.4.1.14 #define GSLC\_LOCAL\_STR 1
- 5.4.1.15 #define GSLC\_LOCAL\_STR\_LEN 30
- 5.4.1.16 #define GSLC\_USE\_PROGMEM 0

# 5.5 src/GUIslice\_drv.h File Reference

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



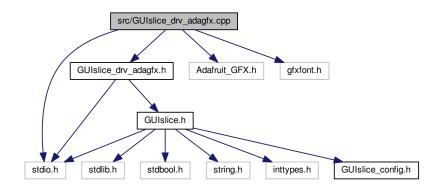
This graph shows which files directly or indirectly include this file:



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

```
#include "GUIslice_drv_adagfx.h"
#include <stdio.h>
#include <Adafruit_GFX.h>
#include <gfxfont.h>
```

Include dependency graph for GUIslice\_drv\_adagfx.cpp:



## **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.

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.

void \* gslc\_DrvFontAdd (const char \*acFontName, uint16\_t nFontSz)

Load a font from a file 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, 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)

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

Draw a filled rectangle.

bool gslc DrvDrawFrameRect (gslc tsGui \*pGui, gslc tsRect rRect, gslc tsColor nCol)

Draw a framed 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 \*, int16\_t nMidX, int16\_t nMidY, uint16\_t nRadius, gslc\_tsColor nCol)

Draw a framed circle.

- bool gslc\_DrvDrawFillCircle (gslc\_tsGui \*, int16\_t nMidX, int16\_t nMidY, uint16\_t nRadius, gslc\_tsColor nCol)

  Draw a filled circle.
- void gslc\_DrvDrawMonoFromMem (gslc\_tsGui \*pGui, int16\_t x, int16\_t y, const unsigned char \*bitmap, bool bProgMem)
- 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\_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)

Get the last touch event from the SDL handler.

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

## 5.6.1 Function Documentation

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

5.6.1.2 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

5.6.1.3 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

5.6.1.4 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

5.6.1.5 bool gslc\_DrvDrawFillRect (  $gslc_tsGui*pGui, gslc_tsRect rRect, gslc_tsColor nCol$  )

Draw a filled rectangle.

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

5.6.1.6 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

5.6.1.7 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

5.6.1.8 bool gslc\_DrvDrawImage ( gslc\_tsGui \* pGui, int16\_t nDstX, int16\_t nDstY, gslc\_tsImgRef sImgRef )

Copy all of source image to destination screen at specified coordinate.

## **Parameters**

in	pGui	Pointer to GUI
in	nDstX	Destination X coord for copy
in	nDstY	Destination Y coord for copy
in	sImgRef	Image reference

### Returns

true if success, false if fail

5.6.1.9 bool gslc\_DrvDrawLine ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nY0, int16\_t nX1, int16\_t nX1, int16\_t nY1, gslc\_tsColor nCol )

Draw a line.

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

- 5.6.1.10 void gslc\_DrvDrawMonoFromMem ( gslc\_tsGui \* pGui, int16\_t x, int16\_t y, const unsigned char \* bitmap, bool bProgMem )
- 5.6.1.11 bool gslc\_DrvDrawPoint ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, gslc\_tsColor nCol )

## Draw a point.

### **Parameters**

in	pGui	Pointer to GUI
in	nX	X coordinate of point
in	nY	Y coordinate of point
in	nCol	Color RGB value to draw

### Returns

true if success, false if error

 $5.6.1.12 \quad bool\ gslc\_DrvDrawPoints\ (\ gslc\_tsGui*pGui,\ gslc\_tsPt*asPt,\ uint16\_t\ nNumPt,\ gslc\_tsColor\ nCol\ )$ 

## Draw a point.

#### **Parameters**

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

## Returns

true if success, false if error

5.6.1.13 bool gslc\_DrvDrawTxt ( gslc\_tsGui \* pGui, int16\_t nTxtX, int16\_t nTxtY, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, gslc\_tsColor colTxt)

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

### Returns

true if success, false if failure

5.6.1.14 void\* gslc\_DrvFontAdd ( const char \* acFontName, uint16\_t nFontSz )

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

### **Parameters**

in	acFontName	Filename path to the font
in	nFontSz	Typeface size to use

## Returns

true if load was successful, false otherwise

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

Release all fonts defined in the GUI.

## **Parameters**

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

### Returns

none

5.6.1.16 bool gslc\_DrvGetTouch (  $gslc_tsGui*pGui$ ,  $int16_t*pnX$ ,  $int16_t*pnY$ ,  $uint16_t*pnPress$  )

Get the last touch event from the SDL handler.

Get the last touch event from the SDL\_Event handler.

## **Parameters**

in	pGui	Pointer to GUI
out	pnX	Ptr to X coordinate of last touch event
out	pnY	Ptr to Y coordinate of last touch event
out	pnPress	Ptr to Pressure level of last touch event (0 for none, >0 for touch)

## Returns

true if an event was detected or 0 otherwise

5.6.1.17 bool gslc\_DrvGetTxtSize ( gslc\_tsGui \* pGui, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, 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	pnTxtSzW	Ptr to width of text
out	pnTxtSzH	Ptr to height of text

### Returns

true if success, false if failure

5.6.1.18 void gslc\_DrvImageDestruct ( void \* pvImg )

Release an image surface.

## **Parameters**

in	pvImg	Void ptr to image

### Returns

none

5.6.1.19 bool gslc\_Drvlnit ( 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

5.6.1.20 bool gslc\_DrvInitTouch ( gslc\_tsGui \* pGui, const char \* acDev )

Perform any touchscreen-specific initialization.

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

#### Returns

true if successful

5.6.1.21 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) or NULL if error

5.6.1.22 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

5.6.1.23 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

5.6.1.24 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

 $5.6.1.25 \quad bool\ gslc\_brvSetClipRect\ (\ gslc\_tsGui*pGui,\ gslc\_tsRect*pRect\ )$ 

Set the clipping rectangle for future drawing updates.

#### **Parameters**

in	pGui	Pointer to GUI
in	pRect	Rectangular region to constrain edits

#### Returns

none

5.6.1.26 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

 $5.6.1.27 \quad bool\ gslc\_brvSetElemlmageNorm\ (\ gslc\_tsGui*pGui,\ gslc\_tsElem*pElem,\ gslc\_tsImgRef\ slmgRef\ )$ 

Set an element's normal-state image.

#### **Parameters**

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

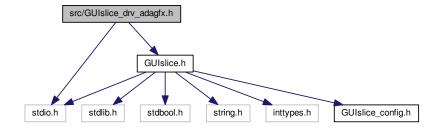
#### Returns

true if success, false if error

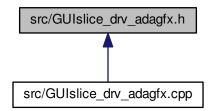
# 5.7 src/GUIslice\_drv\_adagfx.h File Reference

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

Include dependency graph for GUIslice\_drv\_adagfx.h:



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



## Classes

struct gslc\_tsDriver

## **Macros**

• #define DRV\_HAS\_DRAW\_POINT 1

Support gslc\_DrvDrawPoint()

#define DRV\_HAS\_DRAW\_POINTS 0

Support gslc\_DrvDrawPoints()

• #define DRV\_HAS\_DRAW\_LINE 1

Support gslc\_DrvDrawLine()

• #define DRV\_HAS\_DRAW\_RECT\_FRAME 1

Support gslc\_DrvDrawFrameRect()

• #define DRV\_HAS\_DRAW\_RECT\_FILL 1

Support gslc\_DrvDrawFillRect()

• #define DRV\_HAS\_DRAW\_CIRCLE\_FRAME 1

 $Support\ gslc\_DrvDrawFrameCircle()$ 

• #define DRV\_HAS\_DRAW\_CIRCLE\_FILL 1

Support gslc\_DrvDrawFillCircle()

• #define DRV\_HAS\_DRAW\_TEXT 1

Support gslc\_DrvDrawTxt()

#### **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.

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.

void \* gslc DrvFontAdd (const char \*acFontName, uint16 t nFontSz)

Load a font from a file 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, 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)

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\_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\_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)

Get the last touch event from the SDL\_Event handler.

uint16 t gslc DrvAdaptColorToRaw (gslc tsColor nCol)

#### 5.7.1 Macro Definition Documentation

5.7.1.1 #define DRV\_HAS\_DRAW\_CIRCLE\_FILL 1

Support gslc\_DrvDrawFillCircle()

5.7.1.2 #define DRV\_HAS\_DRAW\_CIRCLE\_FRAME 1

Support gslc\_DrvDrawFrameCircle()

5.7.1.3 #define DRV\_HAS\_DRAW\_LINE 1

Support gslc\_DrvDrawLine()

5.7.1.4 #define DRV\_HAS\_DRAW\_POINT 1

Support gslc\_DrvDrawPoint()

5.7.1.5 #define DRV\_HAS\_DRAW\_POINTS 0

Support gslc\_DrvDrawPoints()

5.7.1.6 #define DRV\_HAS\_DRAW\_RECT\_FILL 1

Support gslc\_DrvDrawFillRect()

5.7.1.7 #define DRV\_HAS\_DRAW\_RECT\_FRAME 1

Support gslc\_DrvDrawFrameRect()

5.7.1.8 #define DRV\_HAS\_DRAW\_TEXT 1

Support gslc\_DrvDrawTxt()

## 5.7.2 Function Documentation

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

5.7.2.2 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

5.7.2.3 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

Copy the background image to destination screen.

5.7.2.4 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

5.7.2.5 bool gslc\_DrvDrawFillRect ( gslc\_tsGui \* pGui, gslc\_tsRect rRect, gslc\_tsColor nCol )

Draw a filled rectangle.

#### **Parameters**

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

### Returns

true if success, false if error

5.7.2.6 bool gslc\_DrvDrawFrameCircle ( gslc\_tsGui \* pGui, int16\_t nMidX, int16\_t nMidY, uint16\_t nRadius, gslc\_tsColor nCol )

Draw a framed circle.

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

5.7.2.7 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

5.7.2.8 bool gslc\_DrvDrawlmage ( 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

5.7.2.9 bool gslc\_DrvDrawLine ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nX1, int16\_

Draw a line.

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

5.7.2.10 bool gslc\_DrvDrawPoint ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, gslc\_tsColor nCol )

## Draw a point.

### **Parameters**

in	pGui	Pointer to GUI
in	nX	X coordinate of point
in	nY	Y coordinate of point
in	nCol	Color RGB value to draw

## Returns

true if success, false if error

5.7.2.11 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	nNumPt	Number of points in array
in	nCol	Color RGB value to draw

## Returns

true if success, false if error

5.7.2.12 bool gslc\_DrvDrawTxt ( gslc\_tsGui \* pGui, int16\_t nTxtX, int16\_t nTxtY, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, gslc\_tsColor colTxt )

Draw a text string at the given coordinate.

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

### Returns

true if success, false if failure

5.7.2.13 void\* gslc\_DrvFontAdd ( const char \* acFontName, uint16\_t nFontSz )

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

### **Parameters**

in	acFontName	Filename path to the font	
in	nFontSz	Typeface size to use	

#### Returns

true if load was successful, false otherwise

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

Release all fonts defined in the GUI.

### **Parameters**

in	pGui	Pointer to GUI

## Returns

none

5.7.2.15 bool gslc\_DrvGetTouch ( gslc\_tsGui \* pGui, int16\_t \* pnX, int16\_t \* pnY, uint16\_t \* pnPress )

Get the last touch event from the SDL\_Event handler.

## **Parameters**

in	pGui	Pointer to GUI
out	pnX	Ptr to X coordinate of last touch event
out	pnY	Ptr to Y coordinate of last touch event
out	pnPress	Ptr to Pressure level of last touch event (0 for none, 1 for touch)

### Returns

true if an event was detected or false otherwise

Get the last touch event from the SDL\_Event handler.

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, >0 for touch)

### Returns

true if an event was detected or 0 otherwise

5.7.2.16 bool gslc\_DrvGetTxtSize ( gslc\_tsGui \* pGui, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, 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	pnTxtSzW	Ptr to width of text	
out	pnTxtSzH	Ptr to height of text	

### Returns

true if success, false if failure

5.7.2.17 void gslc\_DrvImageDestruct ( void \* pvImg )

Release an image surface.

### **Parameters**

in	pvlmg	Void ptr to image	

## Returns

none

5.7.2.18 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.

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

### Returns

true if success, false if fail

5.7.2.19 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

5.7.2.20 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

5.7.2.21 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

5.7.2.22 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

5.7.2.23 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

5.7.2.24 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

 $5.7.2.25 \quad \text{bool gslc\_DrvSetClipRect (} \ \, \text{gslc\_tsGui} * \textit{pGui}, \ \, \text{gslc\_tsRect} * \textit{pRect} \ \, \text{)}$ 

Set the clipping rectangle for future drawing updates.

## **Parameters**

in	pGui	Pointer to GUI
in	pRect	Rectangular region to constrain edits

### Returns

none

 $5.7.2.26 \quad bool\ gslc\_brvSetElemImageGlow\ (\ gslc\_tsGui*pGui,\ gslc\_tsElem*pElem,\ gslc\_tsImgRef\ sImgRef\ )$ 

Set an element's glow-state image.

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

#### Returns

true if success, false if error

5.7.2.27 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

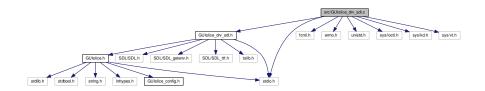
#### Returns

true if success, false if error

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

```
#include "GUIslice_drv_sdl.h"
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <sys/kd.h>
#include <sys/vt.h>
```

Include dependency graph for GUIslice\_drv\_sdl.c:



## **Macros**

#define DRV\_SDL\_FIX\_TTY "/dev/tty0"

## **Functions**

- $\bullet \ \ bool \ gslc\_DrvInit \ (gslc\_tsGui \ *pGui)$ 
  - Initialize the SDL library.
- void gslc\_DrvDestruct (gslc\_tsGui \*pGui)

Free up any members associated with the driver.

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.

void \* gslc DrvFontAdd (const char \*acFontName, uint16 t nFontSz)

Load a font from a file 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, 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)

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

Draw a filled rectangle.

bool gslc DrvDrawFrameRect (gslc tsGui \*pGui, gslc tsRect rRect, gslc tsColor nCol)

Draw a framed 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)

NOTE: Background image is stored in pGui->sImgRefBkgnd.

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)

Get the last touch event from the SDL Event handler.

bool gslc\_DrvCleanStart (const char \*sTTY)

Ensure SDL initializes cleanly to workaround possible issues if previous SDL application failed to close down gracefully.

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.

uint32\_t gslc\_DrvAdaptColorRaw (gslc\_tsGui \*pGui, gslc\_tsColor nCol)

Convert an RGB color triplet into the surface pixel value.

bool gslc\_DrvScreenLock (gslc\_tsGui \*pGui)

Lock an SDL surface so that direct pixel manipulation can be done safely.

void gslc\_DrvScreenUnlock (gslc\_tsGui \*pGui)

Unlock the SDL surface after pixel manipulation is complete.

uint32\_t gslc\_DrvDrawGetPixelRaw (gslc\_tsGui \*pGui, int16\_t nX, int16\_t nY)

Get the pixel at (X,Y) from the active screen.

• void gslc\_DrvDrawSetPixelRaw (gslc\_tsGui \*pGui, int16\_t nX, int16\_t nY, uint32\_t nPixelVal)

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

void gslc\_DrvPasteSurface (gslc\_tsGui \*pGui, int16\_t nX, int16\_t nY, void \*pvSrc, void \*pvDest)

Copy one image region to another.

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

Perform any touchscreen-specific initialization.

int gslc\_TDrvGetTouch (gslc\_tsGui \*pGui, int16\_t \*pnX, int16\_t \*pnY, uint16\_t \*pnPress)

Get the last touch event from the tslib handler.

#### 5.8.1 Macro Definition Documentation

5.8.1.1 #define DRV\_SDL\_FIX\_TTY "/dev/tty0"

### 5.8.2 Function Documentation

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

Translate a gslc\_tsColor into an SDL\_Color.

#### **Parameters**

in	sCol	gslc_tsColor

#### Returns

Converted SDL Color

5.8.2.2 uint32 t gslc DrvAdaptColorRaw ( gslc tsGui \* pGui, gslc tsColor nCol )

Convert an RGB color triplet into the surface pixel value.

This is called to produce the native pixel value required by the raw pixel manipulation routines.

### **Parameters**

in	pGui	Pointer to GUI
in	nCol	RGB value for conversion

## Returns

A pixel value for the current screen format

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

Translate a gslc\_tsRect into an SDL\_Rect.

#### **Parameters**

in	rRect	gslc_tsRect
----	-------	-------------

#### Returns

Converted SDL Rect

5.8.2.4 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

5.8.2.5 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

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

NOTE: Background image is stored in pGui->sImgRefBkgnd.

Copy the background image to destination screen.

5.8.2.7 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

5.8.2.8 bool gslc\_DrvDrawFrameRect ( gslc\_tsGui \* pGui, gslc\_tsRect rRect, gslc\_tsColor nCol )

Draw a framed rectangle.

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

5.8.2.9 uint32\_t gslc\_DrvDrawGetPixelRaw (  $gslc_tsGui * pGui$ , int16\_t nX, int16\_t nY)

Get the pixel at (X,Y) from the active screen.

## PRE:

· Screen surface must be locked

## **Parameters**

in	pGui	Pointer to GUI
in	nX	Pixel X coordinate
in	nY	Pixel Y coordinate

## Returns

Pixel color value from the coordinate or 0 if error

5.8.2.10 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

5.8.2.11 bool gslc\_DrvDrawLine ( gslc\_tsGui \* pGui, int16\_t nX0, int16\_t nY0, int16\_t nX1, int16\_t nY1, gslc\_tsColor nCol )

## Draw a line.

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

5.8.2.12 bool gslc\_DrvDrawPoint ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, gslc\_tsColor nCol )

### Draw a point.

### **Parameters**

in	pGui	Pointer to GUI
in	nX	X coordinate of point
in	nY	Y coordinate of point
in	nCol	Color RGB value to draw

### Returns

true if success, false if error

5.8.2.13 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	nNumPt	Number of points in array
in	nCol	Color RGB value to draw

### Returns

true if success, false if error

5.8.2.14 void gslc\_DrvDrawSetPixelRaw ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, uint32\_t nPixelCol )

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

## PRE:

· Screen surface must be locked

#### **Parameters**

in	pGui	Pointer to GUI
in	nX	Pixel X coordinate to set
in	nY	Pixel Y coordinate to set
in	nPixelCol	Raw color pixel value to assign

### **Returns**

none

5.8.2.15 bool gslc\_DrvDrawTxt ( gslc\_tsGui \* pGui, int16\_t nTxtX, int16\_t nTxtY, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, gslc\_tsColor colTxt)

Draw a text string at the given coordinate.

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

## Returns

true if success, false if failure

5.8.2.16 void\* gslc\_DrvFontAdd ( const char \* acFontName, uint16\_t nFontSz )

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

#### **Parameters**

in	acFontName	Filename path to the font
in	nFontSz	Typeface size to use

### Returns

true if load was successful, false otherwise

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

Release all fonts defined in the GUI.

## **Parameters**

in	pGui	Pointer to GUI

### Returns

none

5.8.2.18 bool gslc\_DrvGetTouch (  $gslc_tsGui*pGui$ ,  $int16_t*pnX$ ,  $int16_t*pnY$ ,  $uint16_t*pnPress$  )

Get the last touch event from the SDL\_Event handler.

### **Parameters**

in	pGui	Pointer to GUI
out	pnX	Ptr to X coordinate of last touch event
out	pnY	Ptr to Y coordinate of last touch event
out	pnPress	Ptr to Pressure level of last touch event (0 for none, 1 for touch)

## Returns

true if an event was detected or false otherwise

Get the last touch event from the SDL\_Event handler.

#### **Parameters**

in	pGui	Pointer to GUI
out	pnX	Ptr to X coordinate of last touch event
out	pnY	Ptr to Y coordinate of last touch event
out	pnPress	Ptr to Pressure level of last touch event (0 for none, >0 for touch)

#### Returns

true if an event was detected or 0 otherwise

5.8.2.19 bool gslc\_DrvGetTxtSize ( gslc\_tsGui \* pGui, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, 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	pnTxtSzW	Ptr to width of text
out	pnTxtSzH	Ptr to height of text

## Returns

true if success, false if failure

5.8.2.20 void gslc\_DrvImageDestruct (void \* pvImg)

Release an image surface.

#### **Parameters**

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

## Returns

none

5.8.2.21 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.

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

### Returns

true if success, false if fail

5.8.2.22 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

5.8.2.23 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) or NULL if error

5.8.2.24 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

5.8.2.25 void gslc\_DrvPasteSurface ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, void \* pvSrc, void \* pvDest )

Copy one image region to another.

• This is typically used to copy an image to the main screen surface

#### **Parameters**

in	pGui	Pointer to GUI
in	nX	Destination X coordinate of copy
in	nY	Destination Y coordinate of copy
in	pvSrc	Void Ptr to source surface (eg. a loaded image)
in	pvDest	Void Ptr to destination surface (typically the screen)

#### Returns

none

5.8.2.26 bool gslc\_DrvScreenLock (  $gslc_tsGui*pGui$  )

Lock an SDL surface so that direct pixel manipulation can be done safely.

This function is called before any direct pixel updates.

POST:

· Primary screen surface is locked

### **Parameters**

in	pGui	Pointer to GUI

#### Returns

true if success, false otherwise

5.8.2.27 void gslc\_DrvScreenUnlock ( gslc\_tsGui \* pGui )

Unlock the SDL surface after pixel manipulation is complete.

This function is called after all pixel updates are done.

POST:

· Primary screen surface is unlocked

## **Parameters**

in	pGui	Pointer to GUI

### Returns

none

 $5.8.2.28 \quad bool\ gslc\_DrvSetBkgndColor\ (\ gslc\_tsGui*pGui,\ gslc\_tsColor\ nCol\ )$ 

Configure the background to use a solid color.

· The background is used when redrawing the entire page

in	pGui	Pointer to GUI
in	nCol	RGB Color to use

#### Returns

true if success, false if fail

5.8.2.29 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

5.8.2.30 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

none

 $5.8.2.31 \quad \text{bool gslc\_DrvSetElemImageGlow (} \ \, \text{gslc\_tsGui} * \textit{pGui}, \ \, \text{gslc\_tsElem} * \textit{pElem}, \ \, \text{gslc\_tsImgRef} \; \text{slmgRef} \; )$ 

Set an element's glow-state image.

### **Parameters**

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

### Returns

true if success, false if error

 $5.8.2.32 \quad \text{bool gslc\_DrvSetElemImageNorm (} \ \, \text{gslc\_tsGui} * \textit{pGui}, \ \, \text{gslc\_tsElem} * \textit{pElem}, \ \, \text{gslc\_tsImgRef} \; \text{slmgRef} \; )$ 

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

5.8.2.33 int gslc\_TDrvGetTouch ( gslc\_tsGui \* pGui, int16\_t \* pnX, int16\_t \* pnY, uint16\_t \* pnPress )

Get the last touch event from the tslib 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, >0 for touch)

#### Returns

non-zero if an event was detected or 0 otherwise

5.8.2.34 bool gslc\_TDrvInitTouch ( 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"

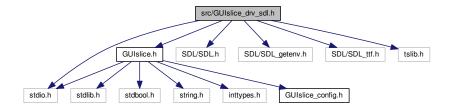
## Returns

true if successful

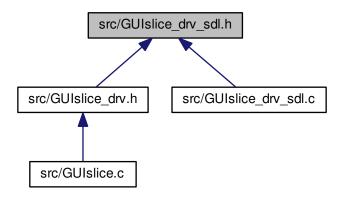
# 5.9 src/GUIslice\_drv\_sdl.h File Reference

```
#include "GUIslice.h"
#include <stdio.h>
#include <SDL/SDL.h>
#include <SDL/SDL_getenv.h>
#include <SDL/SDL_ttf.h>
#include "tslib.h"
```

Include dependency graph for GUIslice\_drv\_sdl.h:



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



## **Classes**

struct gslc\_tsDriver

## **Macros**

- #define DRV\_HAS\_DRAW\_POINT 1
  - Support gslc\_DrvDrawPoint()
- #define DRV\_HAS\_DRAW\_POINTS 1
  - Support gslc\_DrvDrawPoints()
- #define DRV\_HAS\_DRAW\_LINE 0
  - Support gslc\_DrvDrawLine()
- #define DRV\_HAS\_DRAW\_RECT\_FRAME 0
  - Support gslc\_DrvDrawFrameRect()
- #define DRV\_HAS\_DRAW\_RECT\_FILL 1
  - Support gslc\_DrvDrawFillRect()
- #define DRV HAS DRAW CIRCLE FRAME 0
  - Support gslc\_DrvDrawFrameCircle()
- #define DRV\_HAS\_DRAW\_CIRCLE\_FILL 0

Support gslc\_DrvDrawFillCircle()

• #define DRV\_HAS\_DRAW\_TEXT 1

Support gslc\_DrvDrawTxt()

#### **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.

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.

void \* gslc\_DrvFontAdd (const char \*acFontName, uint16\_t nFontSz)

Load a font from a file 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, 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)

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)

Get the last touch event from the SDL\_Event handler.

bool gslc\_DrvCleanStart (const char \*sTTY)

Ensure SDL initializes cleanly to workaround possible issues if previous SDL application failed to close down gracefully.

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 DrvScreenLock (gslc tsGui \*pGui)

Lock an SDL surface so that direct pixel manipulation can be done safely.

void gslc\_DrvScreenUnlock (gslc\_tsGui \*pGui)

Unlock the SDL surface after pixel manipulation is complete.

uint32\_t gslc\_DrvAdaptColorRaw (gslc\_tsGui \*pGui, gslc\_tsColor nCol)

Convert an RGB color triplet into the surface pixel value.

uint32\_t gslc\_DrvDrawGetPixelRaw (gslc\_tsGui \*pGui, int16\_t nX, int16\_t nY)

Get the pixel at (X,Y) from the active screen.

void gslc\_DrvDrawSetPixelRaw (gslc\_tsGui \*pGui, int16\_t nX, int16\_t nY, uint32\_t nPixelCol)

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

void gslc DrvPasteSurface (gslc tsGui \*pGui, int16 t nY, void \*pvSrc, void \*pvDest)

Copy one image region to another.

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

Perform any touchscreen-specific initialization.

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

Perform any touchscreen-specific initialization.

• int gslc TDrvGetTouch (gslc tsGui \*pGui, int16 t \*pnX, int16 t \*pnY, uint16 t \*pnPress)

Get the last touch event from the tslib handler.

### 5.9.1 Macro Definition Documentation

5.9.1.1 #define DRV\_HAS\_DRAW\_CIRCLE\_FILL 0

Support gslc DrvDrawFillCircle()

5.9.1.2 #define DRV\_HAS\_DRAW\_CIRCLE\_FRAME 0

Support gslc\_DrvDrawFrameCircle()

5.9.1.3 #define DRV\_HAS\_DRAW\_LINE 0

Support gslc\_DrvDrawLine()

5.9.1.4 #define DRV\_HAS\_DRAW\_POINT 1

Support gslc\_DrvDrawPoint()

5.9.1.5 #define DRV\_HAS\_DRAW\_POINTS 1

Support gslc\_DrvDrawPoints()

5.9.1.6 #define DRV\_HAS\_DRAW\_RECT\_FILL 1

Support gslc\_DrvDrawFillRect()

5.9.1.7 #define DRV\_HAS\_DRAW\_RECT\_FRAME 0

Support gslc\_DrvDrawFrameRect()

5.9.1.8 #define DRV\_HAS\_DRAW\_TEXT 1

Support gslc\_DrvDrawTxt()

## 5.9.2 Function Documentation

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

Translate a gslc\_tsColor into an SDL\_Color.

#### **Parameters**

	-0-1	
l ın	SUDI	asic istaniar
	0001	900_1000101

#### Returns

Converted SDL\_Color

5.9.2.2 uint32\_t gslc\_DrvAdaptColorRaw ( gslc\_tsGui \* pGui, gslc\_tsColor nCol )

Convert an RGB color triplet into the surface pixel value.

This is called to produce the native pixel value required by the raw pixel manipulation routines.

#### **Parameters**

in	pGui	Pointer to GUI
in	nCol	RGB value for conversion

# Returns

A pixel value for the current screen format

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

Translate a gslc\_tsRect into an SDL\_Rect.

# **Parameters**

in	rRect	gslc_tsRect

#### Returns

Converted SDL\_Rect

5.9.2.4 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

5.9.2.5 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

5.9.2.6 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

Copy the background image to destination screen.

5.9.2.7 bool gslc\_DrvDrawFillRect (  $gslc_tsGui*pGui, gslc_tsRect rRect, gslc_tsColor nCol$  )

Draw a filled rectangle.

#### **Parameters**

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

#### Returns

true if success, false if error

5.9.2.8 bool gslc\_DrvDrawFrameRect ( gslc\_tsGui \* pGui, gslc\_tsRect rRect, gslc\_tsColor nCol )

Draw a framed rectangle.

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

5.9.2.9 uint32\_t gslc\_DrvDrawGetPixelRaw (  $gslc_tsGui * pGui$ , int16\_t nX, int16\_t nY)

Get the pixel at (X,Y) from the active screen.

PRE:

· Screen surface must be locked

#### **Parameters**

in	pGui	Pointer to GUI
in	nX	Pixel X coordinate
in	nY	Pixel Y coordinate

## Returns

Pixel color value from the coordinate or 0 if error

 $5.9.2.10 \quad bool\ gslc\_DrvDrawlmage\ (\ gslc\_tsGui* \textit{pGui},\ int16\_t\ \textit{nDstX},\ int16\_t\ \textit{nDstY},\ gslc\_tslmgRef\ \textit{slmgRef}\ )$ 

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

5.9.2.11 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

5.9.2.12 bool gslc\_DrvDrawPoint ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, gslc\_tsColor nCol )

# Draw a point.

#### **Parameters**

in	pGui	Pointer to GUI
in	nX	X coordinate of point
in	nY	Y coordinate of point
in	nCol	Color RGB value to draw

#### Returns

true if success, false if error

5.9.2.13 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	nNumPt	Number of points in array
in	nCol	Color RGB value to draw

# Returns

true if success, false if error

5.9.2.14 void gslc\_DrvDrawSetPixelRaw (  $gslc_tsGui*pGui$ , int16\_t nX, int16\_t nY, uint32\_t nPixelCol )

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

PRE:

· Screen surface must be locked

# **Parameters**

in	pGui	Pointer to GUI
in	nX	Pixel X coordinate to set

in	nY	Pixel Y coordinate to set
in	nPixelCol	Raw color pixel value to assign

## Returns

none

5.9.2.15 bool gslc\_DrvDrawTxt ( gslc\_tsGui \* pGui, int16\_t nTxtX, int16\_t nTxtY, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, gslc\_tsColor colTxt )

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

#### Returns

true if success, false if failure

5.9.2.16 void\* gslc\_DrvFontAdd ( const char \* acFontName, uint16\_t nFontSz )

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

# Parameters

in	acFontName	Filename path to the font
in	nFontSz	Typeface size to use

# Returns

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

# **Parameters**

in	acFontName	Filename path to the font
in	nFontSz	Typeface size to use

# Returns

true if load was successful, false otherwise

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

Release all fonts defined in the GUI.

#### **Parameters**

in	pGui	Pointer to GUI

## Returns

none

 $5.9.2.18 \quad bool\ gslc\_DrvGetTouch\ (\ gslc\_tsGui*pGui,\ int16\_t*pnX,\ int16\_t*pnY,\ uint16\_t*pnPress\ )$ 

Get the last touch event from the SDL\_Event handler.

Get the last touch event from the SDL 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)

#### Returns

true if an event was detected or false otherwise

# **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, >0 for touch)

## Returns

true if an event was detected or 0 otherwise

Get the last touch event from the SDL\_Event handler.

## **Parameters**

in	pGui	Pointer to GUI
out	pnX	Ptr to X coordinate of last touch event
out	pnY	Ptr to Y coordinate of last touch event
out	pnPress	Ptr to Pressure level of last touch event (0 for none, >0 for touch)

# Returns

true if an event was detected or 0 otherwise

# **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)

# Returns

true if an event was detected or false otherwise

Get the last touch event from the SDL\_Event handler.

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, >0 for touch)

## Returns

true if an event was detected or 0 otherwise

5.9.2.19 bool gslc\_DrvGetTxtSize ( gslc\_tsGui \* pGui, gslc\_tsFont \* pFont, const char \* pStr, gslc\_teTxtFlags eTxtFlags, 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	pnTxtSzW	Ptr to width of text
out	pnTxtSzH	Ptr to height of text

#### Returns

true if success, false if failure

5.9.2.20 void gslc\_DrvImageDestruct ( void \* pvImg )

Release an image surface.

# **Parameters**

in	pvlmg	Void ptr to image

# Returns

none

5.9.2.21 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**

_			
ſ	in	pGui	Pointer to GUI

## Returns

true if success, false if fail

- 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

5.9.2.22 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

5.9.2.23 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/path) or NULL if error

Transparency is enabled by GSLC\_BMP\_TRANS\_EN through use of color (GSLC\_BMP\_TRANS\_RGB).

in	pGui	Pointer to GUI
in	sImgRef	Image reference

## Returns

Image pointer (surface/texture) or NULL if error

5.9.2.24 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

5.9.2.25 void gslc\_DrvPasteSurface ( gslc\_tsGui \* pGui, int16\_t nX, int16\_t nY, void \* pvSrc, void \* pvDest )

Copy one image region to another.

• This is typically used to copy an image to the main screen surface

# **Parameters**

in	pGui	Pointer to GUI
in	nX	Destination X coordinate of copy
in	nY	Destination Y coordinate of copy
in	pvSrc	Void Ptr to source surface (eg. a loaded image)
in	pvDest	Void Ptr to destination surface (typically the screen)

#### Returns

none

5.9.2.26 bool gslc\_DrvScreenLock (  $gslc_tsGui*pGui$  )

Lock an SDL surface so that direct pixel manipulation can be done safely.

This function is called before any direct pixel updates.

# POST:

· Primary screen surface is locked

#### **Parameters**

in	pGui	Pointer to GUI

## Returns

true if success, false otherwise

5.9.2.27 void gslc\_DrvScreenUnlock ( gslc\_tsGui \* pGui )

Unlock the SDL surface after pixel manipulation is complete.

This function is called after all pixel updates are done.

POST:

· Primary screen surface is unlocked

#### **Parameters**

in	pGui	Pointer to GUI

#### Returns

none

5.9.2.28 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

5.9.2.29 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

 $5.9.2.30 \quad \mathsf{bool} \ \mathsf{gslc\_brvSetClipRect} \ ( \ \mathsf{gslc\_tsGui} * \mathit{pGui}, \ \mathsf{gslc\_tsRect} * \mathit{pRect} \ )$ 

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

## **Parameters**

in	pGui	Pointer to GUI
in	pRect	Rectangular region to constrain edits

## Returns

none

5.9.2.31 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

5.9.2.32 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

5.9.2.33 int gslc\_TDrvGetTouch (  $gslc_tsGui*pGui$ , int16\_t\*pnX, int16\_t\*pnY, uint16\_t\*pnPress )

Get the last touch event from the tslib handler.

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, >0 for touch)

#### Returns

non-zero if an event was detected or 0 otherwise

5.9.2.34 bool gslc\_TDrvInitTouch ( 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"

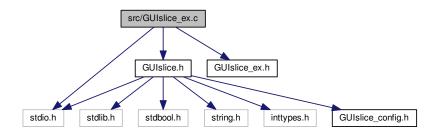
#### Returns

true if successful

# 5.10 src/GUIslice\_ex.c File Reference

```
#include "GUIslice.h"
#include "GUIslice_ex.h"
#include <stdio.h>
```

Include dependency graph for GUIslice\_ex.c:



## **Functions**

- gslc\_tsElem \* 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)
- void gslc\_ElemXGaugeUpdate (gslc\_tsElem \*pElem, int16\_t nVal)

Update a Gauge element's current value.

void gslc ElemXGaugeSetFlip (gslc tsGui \*pGui, gslc tsElem \*pElem, bool bFlip)

Set a Gauge element's fill direction.

Create a Gauge Element.

• bool gslc\_ElemXGaugeDraw (void \*pvGui, void \*pvElem)

Draw a gauge element on the screen.

gslc\_tsElem \* gslc\_ElemXCheckboxCreate (gslc\_tsGui \*pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsX
 Checkbox \*pXData, gslc\_tsRect rElem, bool bRadio, gslc\_teXCheckboxStyle nStyle, gslc\_tsColor colCheck, bool bChecked)

Create a Checkbox Element.

bool gslc\_ElemXCheckboxGetState (gslc\_tsElem \*pElem)

Get a Checkbox element's current state.

gslc tsElem \* gslc ElemXCheckboxFindChecked (gslc tsGui \*pGui, int16 t nGroupId)

Find the checkbox within a group that has been checked.

void gslc ElemXCheckboxSetState (gslc tsElem \*pElem, bool bChecked)

Set a Checkbox element's current state.

void gslc\_ElemXCheckboxToggleState (gslc\_tsElem \*pElem)

Toggle a Checkbox element's current state.

bool gslc ElemXCheckboxDraw (void \*pvGui, void \*pvElem)

Draw a Checkbox element on the screen.

bool gslc\_ElemXCheckboxTouch (void \*pvGui, void \*pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t nRelY)

Handle touch events to Checkbox element.

Create a Slider Element.

void gslc\_ElemXSliderSetStyle (gslc\_tsElem \*pElem, bool bTrim, gslc\_tsColor colTrim, uint16\_t nTickDiv, int16 t nTickLen, gslc tsColor colTick)

Set a Slider element's current position.

int gslc\_ElemXSliderGetPos (gslc\_tsElem \*pElem)

Get a Slider element's current position.

void gslc\_ElemXSliderSetPos (gslc\_tsGui \*pGui, gslc\_tsElem \*pElem, int16\_t nPos)

Set a Slider element's current position.

void gslc\_ElemXSliderSetPosFunc (gslc\_tsElem \*pElem, GSLC\_CB\_XSLIDER\_POS funcCb)

Assign the position callback function for a slider.

bool gslc\_ElemXSliderDraw (void \*pvGui, void \*pvElem)

Draw a Slider element on the screen.

Handle touch events to Slider element.

- bool gslc\_ElemXSliderTouch (void \*pvGui, void \*pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t nRelY)

Create a SelNum Element.

bool gslc\_ElemXSelNumDraw (void \*pvGui, void \*pvElem)

Draw a SelNum element on the screen.

int gslc\_ElemXSelNumGetCounter (gslc\_tsGui \*pGui, gslc\_tsXSelNum \*pSelNum)

Get the current counter associated with SelNum.

void gslc\_ElemXSelNumSetCounter (gslc\_tsXSelNum \*pSelNum, int16\_t nCount)

Set the current counter associated with SelNum.

- bool gslc\_ElemXSelNumClick (void \*pvGui, void \*pvElem, gslc\_teTouch eTouch, int16\_t nX, int16\_t nY)
  - Handle a click event within the SelNum.
- bool gslc\_ElemXSelNumTouch (void \*pvGui, void \*pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t n ← RelY)

Handle touch (up,down,move) events to SelNum element.

# **Variables**

- static const int16\_t SELNUM\_ID\_BTN\_INC = 100
- static const int16\_t SELNUM\_ID\_BTN\_DEC = 101
- static const int16\_t SELNUM\_ID\_TXT = 102

## 5.10.1 Function Documentation

5.10.1.1 gslc\_tsElem\* 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 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

Element pointer or NULL if failure

5.10.1.2 bool gslc\_ElemXCheckboxDraw (void \* pvGui, void \* pvElem)

Draw a Checkbox element on the screen.

• Called from gslc\_ElemDraw()

# **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)

# Returns

true if success, false otherwise

 $5.10.1.3 \quad gslc\_tsElem* gslc\_ElemXCheckboxFindChecked ( \ gslc\_tsGui* pGui, \ int16\_t \ nGroupId \ )$ 

Find the checkbox within a group that has been checked.

## **Parameters**

Generated on Fri Mar 17 2017 23:41:01 for GUIslice by Doxygen

in	pGui	Pointer to GUI
in	nGroupId	Group ID to search

## Returns

Element Ptr or NULL if none checked

5.10.1.4 bool gslc\_ElemXCheckboxGetState ( gslc\_tsElem \* pElem )

Get a Checkbox element's current state.

#### **Parameters**

in	pElem	Pointer to Element	

#### Returns

Current state

5.10.1.5 void gslc\_ElemXCheckboxSetState ( gslc\_tsElem \* pElem, bool bChecked )

Set a Checkbox element's current state.

#### **Parameters**

in	pElem	Pointer to Element
in	bChecked	New state

# Returns

none

5.10.1.6 void gslc\_ElemXCheckboxToggleState ( gslc\_tsElem \* pElem )

Toggle a Checkbox element's current state.

# Parameters

in	pElem	Pointer to Element

#### Returns

none

5.10.1.7 bool gslc\_ElemXCheckboxTouch ( void \* pvGui, void \* pvElem, 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	pvElem	Void ptr to Element (typecast to gslc_tsElem*)
in	eTouch	Touch event type
in	nRelX	Touch X coord relative to element
in	nRelY	Touch Y coord relative to element

## Returns

true if success, false otherwise

5.10.1.8 gslc\_tsElem\* 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 horizontal or vertical box with a filled region corresponding to the proportion that nVal represents between nMin and nMax.

#### **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 to fill the gauge with
in	bVert	Flag to indicate vertical vs horizontal action (true = vertical, false = horizontal)

## Returns

Pointer to Element or NULL if failure

5.10.1.9 bool gslc\_ElemXGaugeDraw ( void \* pvGui, void \* pvElem )

Draw a gauge element on the screen.

Called from gslc\_ElemDraw()

## **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)

# Returns

true if success, false otherwise

5.10.1.10 void gslc\_ElemXGaugeSetFlip ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElem, 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	pElem	Pointer to Element
in	bFlip	If set, reverse direction of fill from default

#### Returns

none

5.10.1.11 void gslc\_ElemXGaugeUpdate ( gslc\_tsElem \* pElem, int16\_t nVal )

Update a Gauge element's current value.

• Note that min & max values are assigned in create()

## **Parameters**

in	pElem	Pointer to Element
in	nVal	New value to show in gauge

#### Returns

none

5.10.1.12 bool gslc\_ElemXSelNumClick ( void \* pvGui, void \* pvElem, gslc\_teTouch eTouch, int16\_t nX, int16\_t nY )

Handle a click event within the SelNum.

• This is called internally by the SelNum touch handler

## **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)
in	eTouch	Touch event type
in	nX	Touch X coord
in	nY	Touch Y coord

## Returns

none

5.10.1.13 gslc\_tsElem\* gslc\_ElemXSelNumCreate ( gslc\_tsGui \* pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsXSelNum \* pXData, gslc\_tsRect rElem, int8\_t nFontId )

Create a SelNum 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 element size
in	nFontId	Font ID to use for drawing the element

# Returns

Pointer to Element or NULL if failure

5.10.1.14 bool gslc\_ElemXSelNumDraw ( void \*pvGui, void \*pvElem )

Draw a SelNum element on the screen.

· Called during redraw

## **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)

## Returns

true if success, false otherwise

 $5.10.1.15 \quad \text{int gslc\_ElemXSelNumGetCounter (} \ \ \text{gslc\_tsXSelNum} * \textit{pSelNum} \ )$ 

Get the current counter associated with SelNum.

# **Parameters**

in	pGui	Ptr to GUI
in	pSelNum	Ptr to Element

#### Returns

Current counter value

5.10.1.16 void gslc\_ElemXSelNumSetCounter ( gslc\_tsXSelNum \* pSelNum, int16\_t nCount )

Set the current counter associated with SelNum.

# **Parameters**

in	pSelNum	Ptr to Element
in	nCount	New counter value

# Returns

none

5.10.1.17 bool gslc\_ElemXSelNumTouch ( void \* pvGui, void \* pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t nRelX)

Handle touch (up,down,move) events to SelNum element.

Called from gslc\_ElemSendEventTouch()

#### **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)
in	eTouch	Touch event type
in	nRelX	Touch X coord relative to element
in	nRelY	Touch Y coord relative to element

#### Returns

true if success, false otherwise

5.10.1.18 gslc\_tsElem\* 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

Element pointer or NULL if failure

5.10.1.19 bool gslc\_ElemXSliderDraw ( void \* pvGui, void \* pvElem )

Draw a Slider element on the screen.

• Called from gslc\_ElemDraw()

# **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)

# Returns

true if success, false otherwise

 $5.10.1.20 \quad \text{int gslc\_ElemXSliderGetPos (} \ \ \text{gslc\_tsElem} * \textit{pElem} \ )$ 

Get a Slider element's current position.

#### **Parameters**

in	pElem	Pointer to Element
----	-------	--------------------

#### Returns

Current slider position

5.10.1.21 void gslc\_ElemXSliderSetPos ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElem, int16\_t nPos )

Set a Slider element's current position.

#### **Parameters**

in	pGui	Pointer to GUI
in	pElem	Pointer to Element
in	nPos	New position value

## Returns

none

5.10.1.22 void gslc\_ElemXSliderSetPosFunc ( gslc\_tsElem \* pElem, GSLC\_CB\_XSLIDER\_POS funcCb )

Assign the position callback function for a slider.

#### **Parameters**

in	pElem	Pointer to element
in	funcCb	Function pointer to position routine (or NULL for none)

# Returns

none

5.10.1.23 void gslc\_ElemXSliderSetStyle ( gslc\_tsElem \* pElem, bool bTrim, gslc\_tsColor colTrim, uint16\_t nTickDiv, int16\_t nTickLen, gslc\_tsColor colTick )

Set a Slider element's current position.

# **Parameters**

in	pElem	Pointer to Element
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

5.10.1.24 bool gslc\_ElemXSliderTouch ( void \* pvGui, void \* pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t nRelY)

Handle touch events to Slider element.

Called from gslc\_ElemSendEventTouch()

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)
in	eTouch	Touch event type
in	nRelX	Touch X coord relative to element
in	nRelY	Touch Y coord relative to element

#### Returns

true if success, false otherwise

#### 5.10.2 Variable Documentation

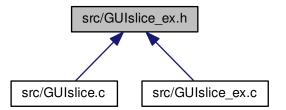
5.10.2.1 const int16\_t SELNUM\_ID\_BTN\_DEC = 101 [static]

5.10.2.2 const int16\_t SELNUM\_ID\_BTN\_INC = 100 [static]

5.10.2.3 const int16\_t SELNUM\_ID\_TXT = 102 [static]

# 5.11 src/GUIslice\_ex.h File Reference

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



# Classes

struct gslc\_tsXGauge

Extended data for Gauge element.

struct gslc\_tsXCheckbox

Extended data for Checkbox element.

struct gslc\_tsXSlider

Extended data for Slider element.

struct gslc\_tsXSelNum

Extended data for SelNum element.

## **Macros**

• #define SELNUM\_STR\_LEN 6

## **Typedefs**

typedef bool(\* GSLC\_CB\_XSLIDER\_POS )(void \*pvGui, void \*pvElem, int16\_t nPos)
 Callback function for slider feedback.

#### **Enumerations**

Extended Element types.

enum gslc\_teXCheckboxStyle { GSLCX\_CHECKBOX\_STYLE\_BOX, GSLCX\_CHECKBOX\_STYLE\_X, GS
 LCX\_CHECKBOX\_STYLE\_ROUND }

Checkbox drawing style.

#### **Functions**

gslc\_tsElem \* 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.

void gslc ElemXGaugeUpdate (gslc tsElem \*pElem, int16 t nVal)

Update a Gauge element's current value.

void gslc\_ElemXGaugeSetFlip (gslc\_tsGui \*pGui, gslc\_tsElem \*pElem, bool bFlip)

Set a Gauge element's fill direction.

bool gslc\_ElemXGaugeDraw (void \*pvGui, void \*pvElem)

Draw a gauge element on the screen.

gslc\_tsElem \* gslc\_ElemXCheckboxCreate (gslc\_tsGui \*pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsX
 Checkbox \*pXData, gslc\_tsRect rElem, bool bRadio, gslc\_teXCheckboxStyle nStyle, gslc\_tsColor colCheck, bool bChecked)

Create a Checkbox Element.

bool gslc ElemXCheckboxGetState (gslc tsElem \*pElem)

Get a Checkbox element's current state.

void gslc\_ElemXCheckboxSetState (gslc\_tsElem \*pElem, bool bChecked)

Set a Checkbox element's current state.

gslc\_tsElem \* gslc\_ElemXCheckboxFindChecked (gslc\_tsGui \*pGui, int16\_t nGroupId)

Find the checkbox within a group that has been checked.

void gslc\_ElemXCheckboxToggleState (gslc\_tsElem \*pElem)

Toggle a Checkbox element's current state.

bool gslc\_ElemXCheckboxDraw (void \*pvGui, void \*pvElem)

Draw a Checkbox element on the screen.

bool gslc\_ElemXCheckboxTouch (void \*pvGui, void \*pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t nRelY)

Handle touch events to Checkbox element.

 gslc\_tsElem \* 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 b← Vert)

Create a Slider Element.

void gslc\_ElemXSliderSetStyle (gslc\_tsElem \*pElem, bool bTrim, gslc\_tsColor colTrim, uint16\_t nTickDiv, int16\_t nTickLen, gslc\_tsColor colTick)

Set a Slider element's current position.

int gslc ElemXSliderGetPos (gslc tsElem \*pElem)

Get a Slider element's current position.

void gslc\_ElemXSliderSetPos (gslc\_tsGui \*pGui, gslc\_tsElem \*pElem, int16\_t nPos)

Set a Slider element's current position.

void gslc\_ElemXSliderSetPosFunc (gslc\_tsElem \*pElem, GSLC\_CB\_XSLIDER\_POS funcCb)

Assign the position callback function for a slider.

bool gslc\_ElemXSliderDraw (void \*pvGui, void \*pvElem)

Draw a Slider element on the screen.

- bool gslc\_ElemXSliderTouch (void \*pvGui, void \*pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t nRelY)

  Handle touch events to Slider element.

Create a SelNum Element.

bool gslc\_ElemXSelNumDraw (void \*pvGui, void \*pvElem)

Draw a SelNum element on the screen.

int gslc\_ElemXSelNumGetCounter (gslc\_tsGui \*pGui, gslc\_tsXSelNum \*pSelNum)

Get the current counter associated with SelNum.

• void gslc\_ElemXSelNumSetCounter (gslc\_tsXSelNum \*pSelNum, int16\_t nCount)

Set the current counter associated with SelNum.

• bool gslc\_ElemXSelNumClick (void \*pvGui, void \*pvElem, gslc\_teTouch eTouch, int16\_t nX, int16\_t nY)

Handle a click event within the SelNum.

bool gslc\_ElemXSelNumTouch (void \*pvGui, void \*pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t n RelY)

Handle touch (up,down,move) events to SelNum element.

#### 5.11.1 Macro Definition Documentation

5.11.1.1 #define SELNUM\_STR\_LEN 6

# 5.11.2 Typedef Documentation

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

Callback function for slider feedback.

# 5.11.3 Enumeration Type Documentation

5.11.3.1 enum gslc\_teTypeExtend

Extended Element types.

**Enumerator** 

GSLC\_TYPEX\_GAUGE Guage / progressbar extended element.

GSLC\_TYPEX\_CHECKBOX Checkbox extended element.

GSLC\_TYPEX\_SLIDER Slider extended element.

GSLC\_TYPEX\_SELNUM SelNum extended element.

# 5.11.3.2 enum gslc\_teXCheckboxStyle

Checkbox drawing style.

Enumerator

GSLCX\_CHECKBOX\_STYLE\_BOX Inner box.
GSLCX\_CHECKBOX\_STYLE\_X Crossed.

GSLCX\_CHECKBOX\_STYLE\_ROUND Circular.

# 5.11.4 Function Documentation

5.11.4.1 gslc\_tsElem\* 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 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

Element pointer or NULL if failure

5.11.4.2 bool gslc\_ElemXCheckboxDraw (void \* pvGui, void \* pvElem)

Draw a Checkbox element on the screen.

• Called from gslc\_ElemDraw()

## **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)

## Returns

true if success, false otherwise

5.11.4.3 gslc\_tsElem\* 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	nGroupId	Group ID to search

## Returns

Element Ptr or NULL if none checked

5.11.4.4 bool gslc\_ElemXCheckboxGetState ( gslc\_tsElem \* pElem )

Get a Checkbox element's current state.

in	pElem	Pointer to Element

#### Returns

Current state

 $5.11.4.5 \quad \text{void gslc\_ElemXCheckboxSetState (} \ \ \text{gslc\_tsElem} * \textit{pElem,} \ \ \text{bool } \textit{bChecked} \ \ )$ 

Set a Checkbox element's current state.

## **Parameters**

ſ	in	pElem	Pointer to Element
	in	bChecked	New state

#### Returns

none

5.11.4.6 void gslc\_ElemXCheckboxToggleState ( gslc\_tsElem \* pElem )

Toggle a Checkbox element's current state.

## **Parameters**

in	pElem	Pointer to Element

# Returns

none

5.11.4.7 bool gslc\_ElemXCheckboxTouch ( void \* pvGui, void \* pvElem, 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	pvElem	Void ptr to Element (typecast to gslc_tsElem*)
in	eTouch	Touch event type
in	nRelX	Touch X coord relative to element
in	nRelY	Touch Y coord relative to element

# Returns

true if success, false otherwise

5.11.4.8 gslc\_tsElem\* 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 horizontal or vertical box with a filled region corresponding to the proportion that nVal represents between nMin and nMax.

#### **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 to fill the gauge with
in	bVert	Flag to indicate vertical vs horizontal action (true = vertical, false = horizontal)

#### Returns

Pointer to Element or NULL if failure

5.11.4.9 bool gslc\_ElemXGaugeDraw ( void \* pvGui, void \* pvElem )

Draw a gauge element on the screen.

• Called from gslc\_ElemDraw()

# **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)

# Returns

true if success, false otherwise

5.11.4.10 void gslc\_ElemXGaugeSetFlip (  $gslc_tsGui*pGui, gslc_tsElem*pElem, bool bFlip$  )

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

in	pGui	Pointer to GUI
in	pElem	Pointer to Element
in	bFlip	If set, reverse direction of fill from default

## Returns

none

5.11.4.11 void gslc\_ElemXGaugeUpdate ( gslc\_tsElem \* pElem, int16\_t nVal )

Update a Gauge element's current value.

• Note that min & max values are assigned in create()

#### **Parameters**

in	pElem	Pointer to Element
in	nVal	New value to show in gauge

## Returns

none

5.11.4.12 bool gslc\_ElemXSelNumClick (void \* pvGui, void \* pvElem, gslc\_teTouch eTouch, int16\_t nX, int16\_t nY)

Handle a click event within the SelNum.

• This is called internally by the SelNum touch handler

## **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)
in	eTouch	Touch event type
in	nX	Touch X coord
in	nY	Touch Y coord

# Returns

none

5.11.4.13 gslc\_tsElem\* gslc\_ElemXSelNumCreate ( gslc\_tsGui \* pGui, int16\_t nElemId, int16\_t nPage, gslc\_tsXSelNum \* pXData, gslc\_tsRect rElem, int8\_t nFontId )

Create a SelNum 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 element size
in	nFontId	Font ID to use for drawing the element

#### Returns

Pointer to Element or NULL if failure

5.11.4.14 bool gslc\_ElemXSelNumDraw (void \* pvGui, void \* pvElem)

Draw a SelNum element on the screen.

· Called during redraw

#### **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)

## Returns

true if success, false otherwise

 $5.11.4.15 \quad \text{int gslc\_ElemXSelNumGetCounter (} \ \ \textbf{gslc\_tsGui} * \textit{pGui}, \ \ \textbf{gslc\_tsXSelNum} * \textit{pSelNum} \ )$ 

Get the current counter associated with SelNum.

# **Parameters**

		Division Olli
in	pGui	Ptr to GUI
in	pSelNum	Ptr to Element

# Returns

Current counter value

5.11.4.16 void gslc\_ElemXSelNumSetCounter( gslc\_tsXSelNum \* pSelNum, int16\_t nCount)

Set the current counter associated with SelNum.

# **Parameters**

in	pSelNum	Ptr to Element
in	nCount	New counter value

# Returns

none

5.11.4.17 bool gslc\_ElemXSelNumTouch ( void \* pvGui, void \* pvElem, gslc\_teTouch eTouch, int16\_t nRelX, int16\_t nRelY )

Handle touch (up,down,move) events to SelNum element.

Called from gslc\_ElemSendEventTouch()

#### **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)
in	eTouch	Touch event type
in	nRelX	Touch X coord relative to element
in	nRelY	Touch Y coord relative to element

#### Returns

true if success, false otherwise

5.11.4.18 gslc\_tsElem\* 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

Element pointer or NULL if failure

5.11.4.19 bool gslc\_ElemXSliderDraw ( void \* pvGui, void \* pvElem )

Draw a Slider element on the screen.

• Called from gslc\_ElemDraw()

# **Parameters**

in	pvGui	Void ptr to GUI (typecast to gslc_tsGui*)
in	pvElem	Void ptr to Element (typecast to gslc_tsElem*)

# Returns

true if success, false otherwise

5.11.4.20 int gslc\_ElemXSliderGetPos ( gslc\_tsElem \* pElem )

Get a Slider element's current position.

in	pElem	Pointer to Element

#### Returns

Current slider position

5.11.4.21 void gslc\_ElemXSliderSetPos ( gslc\_tsGui \* pGui, gslc\_tsElem \* pElem, int16\_t nPos )

Set a Slider element's current position.

## **Parameters**

in	pGui	Pointer to GUI
in	pElem	Pointer to Element
in	nPos	New position value

## Returns

none

5.11.4.22 void gslc\_ElemXSliderSetPosFunc ( gslc\_tsElem \* pElem, GSLC\_CB\_XSLIDER\_POS funcCb )

Assign the position callback function for a slider.

#### **Parameters**

in	pElem	Pointer to element
in	funcCb	Function pointer to position routine (or NULL for none)

# Returns

none

5.11.4.23 void gslc\_ElemXSliderSetStyle ( gslc\_tsElem \* pElem, bool bTrim, gslc\_tsColor colTrim, uint16\_t nTickDiv, int16\_t nTickLen, gslc\_tsColor colTick )

Set a Slider element's current position.

# **Parameters**

in	pElem	Pointer to Element
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

5.11.4.24 bool gslc\_ElemXSliderTouch ( void \* pvGui, void \* pvElem, 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	pvElem	Void ptr to Element (typecast to gslc_tsElem*)
in	eTouch	Touch event type
in	nRelX	Touch X coord relative to element
in	nRelY	Touch Y coord relative to element

# Returns

true if success, false otherwise