

GLCD

Generated by Doxygen 1.7.6.1

Wed Oct 24 2012 22:48:50

Contents

1	Module Index	1
1.1	Modules	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Module Documentation	9
4.1	Primitives_api	9
4.1.1	Define Documentation	9
4.1.1.1	DPYCOLORTRANSLATE	9
4.1.2	Function Documentation	10
4.1.2.1	GrOffScreen1BPPInit	10
4.1.2.2	GrOffScreen4BPPInit	10
4.1.2.3	GrOffScreen4BPPPPaletteSet	10
4.1.2.4	GrOffScreen8BPPInit	11
4.1.2.5	GrOffScreen8BPPPPaletteSet	11
4.2	Listbox_api	13
4.2.1	Define Documentation	14
4.2.1.1	ListBox	14
4.2.1.2	LISTBOX_STYLE_LOCKED	16
4.2.1.3	LISTBOX_STYLE_OUTLINE	16
4.2.1.4	LISTBOX_STYLE_WRAP	16
4.2.1.5	ListBoxBackgroundColorSet	16

4.2.1.6	ListBoxCallbackSet	17
4.2.1.7	ListBoxClear	17
4.2.1.8	ListBoxFontSet	18
4.2.1.9	ListBoxLock	18
4.2.1.10	ListBoxOutlineColorSet	19
4.2.1.11	ListBoxOutlineOff	19
4.2.1.12	ListBoxOutlineOn	20
4.2.1.13	ListBoxSelectedBackgroundColorSet	20
4.2.1.14	ListBoxSelectedTextColorSet	21
4.2.1.15	ListBoxSelectionGet	21
4.2.1.16	ListBoxSelectionSet	21
4.2.1.17	ListBoxStruct	22
4.2.1.18	ListBoxTextColorSet	23
4.2.1.19	ListBoxTextSet	24
4.2.1.20	ListBoxUnlock	24
4.2.1.21	ListBoxWrapDisable	25
4.2.1.22	ListBoxWrapEnable	25
4.2.2	Function Documentation	26
4.2.2.1	ListBoxInit	26
4.2.2.2	ListBoxMsgProc	27
4.2.2.3	ListBoxTextAdd	27
4.2.3	Variable Documentation	27
4.2.3.1	IPointerY	28
4.2.3.2	pfnOnChange	28
4.2.3.3	pFont	28
4.2.3.4	ppcText	28
4.2.3.5	sBase	28
4.2.3.6	sSelected	28
4.2.3.7	ulBackgroundColor	28
4.2.3.8	ulOutlineColor	28
4.2.3.9	ulSelectedBackgroundColor	28
4.2.3.10	ulSelectedTextColor	29
4.2.3.11	ulStyle	29
4.2.3.12	ulTextColor	29

4.2.3.13	usMaxEntries	29
4.2.3.14	usOldestEntry	29
4.2.3.15	usPopulated	29
4.2.3.16	usScrolled	29
4.2.3.17	usStartEntry	29
4.3	PushButton_api	30
4.3.1	Define Documentation	31
4.3.1.1	CircularButton	31
4.3.1.2	CircularButtonStruct	33
4.3.1.3	PB_STYLE_AUTO_REPEAT	35
4.3.1.4	PB_STYLE_FILL	35
4.3.1.5	PB_STYLE_IMG	35
4.3.1.6	PB_STYLE_OUTLINE	35
4.3.1.7	PB_STYLE_PRESSED	35
4.3.1.8	PB_STYLE_RELEASE_NOTIFY	35
4.3.1.9	PB_STYLE_TEXT	35
4.3.1.10	PB_STYLE_TEXT_OPAQUE	36
4.3.1.11	PushButtonAutoRepeatDelaySet	36
4.3.1.12	PushButtonAutoRepeatOff	36
4.3.1.13	PushButtonAutoRepeatOn	37
4.3.1.14	PushButtonAutoRepeatRateSet	37
4.3.1.15	PushButtonCallbackSet	38
4.3.1.16	PushButtonFillColorPressedSet	38
4.3.1.17	PushButtonFillColorSet	39
4.3.1.18	PushButtonFillOff	39
4.3.1.19	PushButtonFillOn	40
4.3.1.20	PushButtonFontSet	40
4.3.1.21	PushButtonImageOff	41
4.3.1.22	PushButtonImageOn	41
4.3.1.23	PushButtonImagePressedSet	42
4.3.1.24	PushButtonImageSet	42
4.3.1.25	PushButtonOutlineColorSet	43
4.3.1.26	PushButtonOutlineOff	43
4.3.1.27	PushButtonOutlineOn	44

4.3.1.28	PushButtonTextColorSet	44
4.3.1.29	PushButtonTextOff	45
4.3.1.30	PushButtonTextOn	45
4.3.1.31	PushButtonTextOpaqueOff	46
4.3.1.32	PushButtonTextOpaqueOn	46
4.3.1.33	PushButtonTextSet	47
4.3.1.34	RectangularButton	47
4.3.1.35	RectangularButtonStruct	49
4.3.2	Function Documentation	51
4.3.2.1	CircularButtonInit	51
4.3.2.2	CircularButtonMsgProc	51
4.3.2.3	RectangularButtonInit	52
4.3.2.4	RectangularButtonMsgProc	52
4.4	Radiobutton_api	54
4.4.1	Define Documentation	55
4.4.1.1	RadioButton	55
4.4.1.2	RadioButtonCallbackSet	56
4.4.1.3	RadioButtonCircleSizeSet	56
4.4.1.4	RadioButtonFillColorSet	57
4.4.1.5	RadioButtonFillOff	57
4.4.1.6	RadioButtonFillOn	58
4.4.1.7	RadioButtonFontSet	58
4.4.1.8	RadioButtonImageOff	59
4.4.1.9	RadioButtonImageOn	59
4.4.1.10	RadioButtonImageSet	60
4.4.1.11	RadioButtonOutlineColorSet	60
4.4.1.12	RadioButtonOutlineOff	61
4.4.1.13	RadioButtonOutlineOn	61
4.4.1.14	RadioButtonStruct	62
4.4.1.15	RadioButtonTextColorSet	64
4.4.1.16	RadioButtonTextOff	64
4.4.1.17	RadioButtonTextOn	65
4.4.1.18	RadioButtonTextOpaqueOff	65
4.4.1.19	RadioButtonTextOpaqueOn	66

4.4.1.20	RadioButtonTextSet	66
4.4.1.21	RB_STYLE_FILL	67
4.4.1.22	RB_STYLE_IMG	67
4.4.1.23	RB_STYLE_OUTLINE	67
4.4.1.24	RB_STYLE_SELECTED	67
4.4.1.25	RB_STYLE_TEXT	67
4.4.1.26	RB_STYLE_TEXT_OPAQUE	67
4.4.2	Function Documentation	67
4.4.2.1	RadioButtonInit	67
4.4.2.2	RadioButtonMsgProc	68
5	Data Structure Documentation	69
5.1	<u>Widget Struct Reference</u>	69
5.1.1	Detailed Description	69
5.1.2	Field Documentation	70
5.1.2.1	ISize	70
5.1.2.2	pChild	70
5.1.2.3	pDisplay	70
5.1.2.4	pfnMsgProc	70
5.1.2.5	pNext	70
5.1.2.6	pParent	70
5.1.2.7	sPosition	70
5.2	<u>tCanvasWidget Struct Reference</u>	70
5.2.1	Detailed Description	71
5.2.2	Field Documentation	71
5.2.2.1	pcText	71
5.2.2.2	pfnOnPaint	71
5.2.2.3	pFont	71
5.2.2.4	puImage	71
5.2.2.5	sBase	71
5.2.2.6	ulFillColor	72
5.2.2.7	ulOutlineColor	72
5.2.2.8	ulStyle	72
5.2.2.9	ulTextColor	72

5.3	tCheckBoxWidget Struct Reference	72
5.3.1	Detailed Description	73
5.3.2	Field Documentation	73
5.3.2.1	pcText	73
5.3.2.2	pfnOnChange	73
5.3.2.3	pFont	73
5.3.2.4	puImage	73
5.3.2.5	sBase	73
5.3.2.6	ulFillColor	73
5.3.2.7	ulOutlineColor	73
5.3.2.8	ulTextColor	73
5.3.2.9	usBoxSize	74
5.3.2.10	usStyle	74
5.4	tContainerWidget Struct Reference	74
5.4.1	Detailed Description	74
5.4.2	Field Documentation	74
5.4.2.1	pcText	74
5.4.2.2	pFont	74
5.4.2.3	sBase	75
5.4.2.4	ulFillColor	75
5.4.2.5	ulOutlineColor	75
5.4.2.6	ulStyle	75
5.4.2.7	ulTextColor	75
5.5	tContext Struct Reference	75
5.5.1	Detailed Description	76
5.5.2	Field Documentation	76
5.5.2.1	ISize	76
5.5.2.2	pDisplay	76
5.5.2.3	pFont	76
5.5.2.4	sClipRegion	76
5.5.2.5	ulBackground	76
5.5.2.6	ulForeground	76
5.6	tDisplay Struct Reference	76
5.6.1	Detailed Description	77

5.6.2	Field Documentation	77
5.6.2.1	ISize	77
5.6.2.2	pfnColorTranslate	77
5.6.2.3	pfnFlush	78
5.6.2.4	pfnLineDrawH	78
5.6.2.5	pfnLineDrawV	78
5.6.2.6	pfnPixelDraw	78
5.6.2.7	pfnPixelDrawMultiple	78
5.6.2.8	pfnRectFill	78
5.6.2.9	pvDisplayData	78
5.6.2.10	usHeight	78
5.6.2.11	usWidth	78
5.7	tFont Struct Reference	79
5.7.1	Detailed Description	79
5.7.2	Field Documentation	79
5.7.2.1	pucData	79
5.7.2.2	pusOffset	79
5.7.2.3	ucBaseline	79
5.7.2.4	ucFormat	79
5.7.2.5	ucHeight	80
5.7.2.6	ucMaxWidth	80
5.8	tlImageButtonWidget Struct Reference	80
5.8.1	Detailed Description	80
5.8.2	Field Documentation	81
5.8.2.1	pcText	81
5.8.2.2	pfnOnClick	81
5.8.2.3	pFont	81
5.8.2.4	puImage	81
5.8.2.5	pucKeycapImage	81
5.8.2.6	pucPressImage	81
5.8.2.7	sBase	81
5.8.2.8	sXOffset	81
5.8.2.9	sYOffset	81
5.8.2.10	uiAutoRepeatCount	82

5.8.2.11	ulBackgroundColor	82
5.8.2.12	ulForegroundColor	82
5.8.2.13	ulPressedColor	82
5.8.2.14	ulStyle	82
5.8.2.15	usAutoRepeatDelay	82
5.8.2.16	usAutoRepeatRate	82
5.9	tListBoxWidget Struct Reference	83
5.9.1	Detailed Description	83
5.10	tPushButtonWidget Struct Reference	83
5.10.1	Detailed Description	84
5.10.2	Field Documentation	84
5.10.2.1	pcText	84
5.10.2.2	pfnOnClick	84
5.10.2.3	pFont	84
5.10.2.4	puImage	84
5.10.2.5	pucPressImage	85
5.10.2.6	sBase	85
5.10.2.7	ulAutoRepeatCount	85
5.10.2.8	ulFillColor	85
5.10.2.9	ulOutlineColor	85
5.10.2.10	ulPressFillColor	85
5.10.2.11	ulStyle	85
5.10.2.12	ulTextColor	85
5.10.2.13	usAutoRepeatDelay	85
5.10.2.14	usAutoRepeatRate	86
5.11	tRadioButtonWidget Struct Reference	86
5.11.1	Detailed Description	86
5.11.2	Field Documentation	86
5.11.2.1	pcText	86
5.11.2.2	pfnOnChange	86
5.11.2.3	pFont	87
5.11.2.4	puImage	87
5.11.2.5	sBase	87
5.11.2.6	ulFillColor	87

5.11.2.7 ulOutlineColor	87
5.11.2.8 ulTextColor	87
5.11.2.9 usCircleSize	87
5.11.2.10 usStyle	87
5.12 tRectangle Struct Reference	88
5.12.1 Detailed Description	88
5.12.2 Field Documentation	88
5.12.2.1 sXMax	88
5.12.2.2 sXMin	88
5.12.2.3 sYMax	88
5.12.2.4 sYMin	88
5.13 tSliderWidget Struct Reference	89
5.13.1 Detailed Description	89
5.13.2 Field Documentation	89
5.13.2.1 lMax	89
5.13.2.2 lMin	89
5.13.2.3 lValue	90
5.13.2.4 pcText	90
5.13.2.5 pfnOnChange	90
5.13.2.6 pFont	90
5.13.2.7 pucBackgroundImage	90
5.13.2.8 puImage	90
5.13.2.9 sBase	90
5.13.2.10 sPos	90
5.13.2.11 ulBackgroundColor	90
5.13.2.12 ulBackgroundTextColor	91
5.13.2.13 ulFillColor	91
5.13.2.14 ulOutlineColor	91
5.13.2.15 ulStyle	91
5.13.2.16 ulTextColor	91
5.14 tWidgetMessageQueue Struct Reference	91
5.14.1 Field Documentation	91
5.14.1.1 pWidget	91
5.14.1.2 ulFlags	91

5.14.1.3	ulMessage	92
5.14.1.4	ulParam1	92
5.14.1.5	ulParam2	92
6	File Documentation	93
6.1	driver/sed1335-AT91.c File Reference	93
6.1.1	Define Documentation	94
6.1.1.1	SED1335_A0	94
6.1.1.2	SED1335_CS	94
6.1.1.3	SED1335_CTRL_CLR	94
6.1.1.4	SED1335_CTRL_INP	94
6.1.1.5	SED1335_CTRL_IO_EN	94
6.1.1.6	SED1335_CTRL_OUT	94
6.1.1.7	SED1335_CTRL_PIN	94
6.1.1.8	SED1335_CTRL_SET	94
6.1.1.9	SED1335_D0	94
6.1.1.10	SED1335_DATA_CLR	94
6.1.1.11	SED1335_DATA_INP	94
6.1.1.12	SED1335_DATA_IO_EN	94
6.1.1.13	SED1335_DATA_OUT	94
6.1.1.14	SED1335_DATA_PIN	94
6.1.1.15	SED1335_DATA_SET	94
6.1.1.16	SED1335_RD	94
6.1.1.17	SED1335_RES	94
6.1.1.18	SED1335_WR	94
6.1.2	Function Documentation	94
6.1.2.1	GLCD_InitializePorts	94
6.1.2.2	GLCD_ReadByteFromROMMemory	95
6.1.2.3	GLCD_ReadData	95
6.1.2.4	GLCD_WriteCommand	95
6.1.2.5	GLCD_WriteData	95
6.1.2.6	Init	95
6.2	driver/sed1335-avr.c File Reference	95
6.2.1	Define Documentation	95

6.2.1.1	SED1335_A0	95
6.2.1.2	SED1335_CONTROL_DIR	95
6.2.1.3	SED1335_CONTROL_PIN	95
6.2.1.4	SED1335_CONTROL_PORT	96
6.2.1.5	SED1335_CS	96
6.2.1.6	SED1335_DATA_DIR	96
6.2.1.7	SED1335_DATA_PIN	96
6.2.1.8	SED1335_DATA_PORT	96
6.2.1.9	SED1335_RD	96
6.2.1.10	SED1335_RES	96
6.2.1.11	SED1335_WR	96
6.2.2	Function Documentation	96
6.2.2.1	GLCD_InitializePorts	96
6.2.2.2	GLCD_ReadByteFromROMMemory	96
6.2.2.3	GLCD_ReadData	96
6.2.2.4	GLCD_WriteCommand	96
6.2.2.5	GLCD_WriteData	96
6.3	driver/sed1335-LPC2100.c File Reference	96
6.3.1	Define Documentation	97
6.3.1.1	SED1335_A0	97
6.3.1.2	SED1335_CLR	97
6.3.1.3	SED1335_CS1	97
6.3.1.4	SED1335_D0	97
6.3.1.5	SED1335_DIR	97
6.3.1.6	SED1335_PIN	97
6.3.1.7	SED1335_RD	97
6.3.1.8	SED1335_RES	97
6.3.1.9	SED1335_SET	97
6.3.1.10	SED1335_WR	97
6.3.2	Function Documentation	97
6.3.2.1	GLCD_InitializePorts	97
6.3.2.2	GLCD_ReadByteFromROMMemory	97
6.3.2.3	GLCD_ReadData	97
6.3.2.4	GLCD_ReadStatus	97

6.3.2.5	GLCD_WriteCommand	97
6.3.2.6	GLCD_WriteData	97
6.4	driver/sed1335-MSP430.c File Reference	98
6.4.1	Define Documentation	98
6.4.1.1	SED1335_A0	98
6.4.1.2	SED1335_CS1	98
6.4.1.3	SED1335_D0	98
6.4.1.4	SED1335_DATA	98
6.4.1.5	SED1335_DATA_DIR	98
6.4.1.6	SED1335_DIR	98
6.4.1.7	SED1335_PIN	98
6.4.1.8	SED1335_RD	98
6.4.1.9	SED1335_RES	98
6.4.1.10	SED1335_WR	99
6.4.2	Function Documentation	99
6.4.2.1	GLCD_InitializePorts	99
6.4.2.2	GLCD_ReadByteFromROMMemory	99
6.4.2.3	GLCD_ReadData	99
6.4.2.4	GLCD_ReadStatus	99
6.4.2.5	GLCD_WriteCommand	99
6.4.2.6	GLCD_WriteData	99
6.5	driver/sed1335-STM32.c File Reference	99
6.5.1	Define Documentation	100
6.5.1.1	SED1335_A0	100
6.5.1.2	SED1335_CS	100
6.5.1.3	SED1335_D0	100
6.5.1.4	SED1335_PORT	100
6.5.1.5	SED1335_RD	100
6.5.1.6	SED1335_RES	100
6.5.1.7	SED1335_WR	100
6.5.2	Function Documentation	100
6.5.2.1	GLCD_InitPorts	100
6.5.2.2	GLCD_ReadByteFromROMMemory	100
6.5.2.3	GLCD_ReadData	100

6.5.2.4	GLCD_WriteCommand	100
6.5.2.5	GLCD_WriteData	100
6.5.3	Variable Documentation	100
6.5.3.1	GPIO_InitStructure	100
6.6	driver/sed1335.c File Reference	100
6.6.1	Function Documentation	101
6.6.1.1	GLCD_Bitmap	101
6.6.1.2	GLCD_ClearGraphic	101
6.6.1.3	GLCD_ClearText	101
6.6.1.4	GLCD_GraphicGoTo	101
6.6.1.5	GLCD_Initialize	102
6.6.1.6	GLCD_InitializePorts	102
6.6.1.7	GLCD_SetCursorAddress	102
6.6.1.8	GLCD_SetPixel	102
6.6.1.9	GLCD_TextGoTo	103
6.6.1.10	GLCD_WriteText	103
6.6.1.11	GLCD_WriteTextP	103
6.7	grlib/canvas.c File Reference	104
6.7.1	Function Documentation	104
6.7.1.1	CanvasInit	104
6.7.1.2	CanvasMsgProc	104
6.8	grlib/checkbox.c File Reference	105
6.8.1	Function Documentation	105
6.8.1.1	CheckBoxInit	105
6.8.1.2	CheckBoxMsgProc	106
6.9	grlib/circle.c File Reference	106
6.9.1	Function Documentation	106
6.9.1.1	GrCircleDraw	106
6.9.1.2	GrCircleFill	107
6.10	grlib/container.c File Reference	107
6.10.1	Function Documentation	108
6.10.1.1	ContainerInit	108
6.10.1.2	ContainerMsgProc	108
6.11	grlib/context.c File Reference	108

6.11.1	Function Documentation	109
6.11.1.1	GrContextClipRegionSet	109
6.11.1.2	GrContextInit	109
6.12	glib/fonts/fontcm12.c File Reference	109
6.12.1	Variable Documentation	110
6.12.1.1	g_sFontCm12	110
6.13	glib/fonts/fontcm12b.c File Reference	110
6.13.1	Variable Documentation	110
6.13.1.1	g_sFontCm12b	110
6.14	glib/fonts/fontcm12i.c File Reference	110
6.14.1	Variable Documentation	110
6.14.1.1	g_sFontCm12i	110
6.15	glib/fonts/fontcm14.c File Reference	110
6.15.1	Variable Documentation	111
6.15.1.1	g_sFontCm14	111
6.16	glib/fonts/fontcm14b.c File Reference	111
6.16.1	Variable Documentation	111
6.16.1.1	g_sFontCm14b	111
6.17	glib/fonts/fontcm14i.c File Reference	111
6.17.1	Variable Documentation	111
6.17.1.1	g_sFontCm14i	111
6.18	glib/fonts/fontcm16.c File Reference	111
6.18.1	Variable Documentation	111
6.18.1.1	g_sFontCm16	111
6.19	glib/fonts/fontcm16b.c File Reference	112
6.19.1	Variable Documentation	112
6.19.1.1	g_sFontCm16b	112
6.20	glib/fonts/fontcm16i.c File Reference	112
6.20.1	Variable Documentation	112
6.20.1.1	g_sFontCm16i	112
6.21	glib/fonts/fontcm18.c File Reference	112
6.21.1	Variable Documentation	112
6.21.1.1	g_sFontCm18	112
6.22	glib/fonts/fontcm18b.c File Reference	112

6.22.1	Variable Documentation	113
6.22.1.1	g_sFontCm18b	113
6.23	grlib/fonts/fontcm18i.c File Reference	113
6.23.1	Variable Documentation	113
6.23.1.1	g_sFontCm18i	113
6.24	grlib/fonts/fontcm20.c File Reference	113
6.24.1	Variable Documentation	113
6.24.1.1	g_sFontCm20	113
6.25	grlib/fonts/fontcm20b.c File Reference	113
6.25.1	Variable Documentation	114
6.25.1.1	g_sFontCm20b	114
6.26	grlib/fonts/fontcm20i.c File Reference	114
6.26.1	Variable Documentation	114
6.26.1.1	g_sFontCm20i	114
6.27	grlib/fonts/fontcm22.c File Reference	114
6.27.1	Variable Documentation	114
6.27.1.1	g_sFontCm22	114
6.28	grlib/fonts/fontcm22b.c File Reference	114
6.28.1	Variable Documentation	114
6.28.1.1	g_sFontCm22b	114
6.29	grlib/fonts/fontcm22i.c File Reference	115
6.29.1	Variable Documentation	115
6.29.1.1	g_sFontCm22i	115
6.30	grlib/fonts/fontcm24.c File Reference	115
6.30.1	Variable Documentation	115
6.30.1.1	g_sFontCm24	115
6.31	grlib/fonts/fontcm24b.c File Reference	115
6.31.1	Variable Documentation	115
6.31.1.1	g_sFontCm24b	115
6.32	grlib/fonts/fontcm24i.c File Reference	115
6.32.1	Variable Documentation	116
6.32.1.1	g_sFontCm24i	116
6.33	grlib/fonts/fontcm26.c File Reference	116
6.33.1	Variable Documentation	116

6.33.1.1	g_sFontCm26	116
6.34	grlib/fonts/fontcm26b.c File Reference	116
6.34.1	Variable Documentation	116
6.34.1.1	g_sFontCm26b	116
6.35	grlib/fonts/fontcm26i.c File Reference	116
6.35.1	Variable Documentation	117
6.35.1.1	g_sFontCm26i	117
6.36	grlib/fonts/fontcm28.c File Reference	117
6.36.1	Variable Documentation	117
6.36.1.1	g_sFontCm28	117
6.37	grlib/fonts/fontcm28b.c File Reference	117
6.37.1	Variable Documentation	117
6.37.1.1	g_sFontCm28b	117
6.38	grlib/fonts/fontcm28i.c File Reference	117
6.38.1	Variable Documentation	117
6.38.1.1	g_sFontCm28i	117
6.39	grlib/fonts/fontcm30.c File Reference	118
6.39.1	Variable Documentation	118
6.39.1.1	g_sFontCm30	118
6.40	grlib/fonts/fontcm30b.c File Reference	118
6.40.1	Variable Documentation	118
6.40.1.1	g_sFontCm30b	118
6.41	grlib/fonts/fontcm30i.c File Reference	118
6.41.1	Variable Documentation	118
6.41.1.1	g_sFontCm30i	118
6.42	grlib/fonts/fontcm32.c File Reference	118
6.42.1	Variable Documentation	119
6.42.1.1	g_sFontCm32	119
6.43	grlib/fonts/fontcm32b.c File Reference	119
6.43.1	Variable Documentation	119
6.43.1.1	g_sFontCm32b	119
6.44	grlib/fonts/fontcm32i.c File Reference	119
6.44.1	Variable Documentation	119
6.44.1.1	g_sFontCm32i	119

6.45 grlib/fonts/fontcm34.c File Reference	119
6.45.1 Variable Documentation	120
6.45.1.1 g_sFontCm34	120
6.46 grlib/fonts/fontcm34b.c File Reference	120
6.46.1 Variable Documentation	120
6.46.1.1 g_sFontCm34b	120
6.47 grlib/fonts/fontcm34i.c File Reference	120
6.47.1 Variable Documentation	120
6.47.1.1 g_sFontCm34i	120
6.48 grlib/fonts/fontcm36.c File Reference	120
6.48.1 Variable Documentation	120
6.48.1.1 g_sFontCm36	120
6.49 grlib/fonts/fontcm36b.c File Reference	121
6.49.1 Variable Documentation	121
6.49.1.1 g_sFontCm36b	121
6.50 grlib/fonts/fontcm36i.c File Reference	121
6.50.1 Variable Documentation	121
6.50.1.1 g_sFontCm36i	121
6.51 grlib/fonts/fontcm38.c File Reference	121
6.51.1 Variable Documentation	121
6.51.1.1 g_sFontCm38	121
6.52 grlib/fonts/fontcm38b.c File Reference	121
6.52.1 Variable Documentation	122
6.52.1.1 g_sFontCm38b	122
6.53 grlib/fonts/fontcm38i.c File Reference	122
6.53.1 Variable Documentation	122
6.53.1.1 g_sFontCm38i	122
6.54 grlib/fonts/fontcm40.c File Reference	122
6.54.1 Variable Documentation	122
6.54.1.1 g_sFontCm40	122
6.55 grlib/fonts/fontcm40b.c File Reference	122
6.55.1 Variable Documentation	123
6.55.1.1 g_sFontCm40b	123
6.56 grlib/fonts/fontcm40i.c File Reference	123

6.56.1	Variable Documentation	123
6.56.1.1	g_sFontCm40i	123
6.57	grlib/fonts/fontcm42.c File Reference	123
6.57.1	Variable Documentation	123
6.57.1.1	g_sFontCm42	123
6.58	grlib/fonts/fontcm42b.c File Reference	123
6.58.1	Variable Documentation	123
6.58.1.1	g_sFontCm42b	123
6.59	grlib/fonts/fontcm42i.c File Reference	124
6.59.1	Variable Documentation	124
6.59.1.1	g_sFontCm42i	124
6.60	grlib/fonts/fontcm44.c File Reference	124
6.60.1	Variable Documentation	124
6.60.1.1	g_sFontCm44	124
6.61	grlib/fonts/fontcm44b.c File Reference	124
6.61.1	Variable Documentation	124
6.61.1.1	g_sFontCm44b	124
6.62	grlib/fonts/fontcm44i.c File Reference	124
6.62.1	Variable Documentation	125
6.62.1.1	g_sFontCm44i	125
6.63	grlib/fonts/fontcm46.c File Reference	125
6.63.1	Variable Documentation	125
6.63.1.1	g_sFontCm46	125
6.64	grlib/fonts/fontcm46b.c File Reference	125
6.64.1	Variable Documentation	125
6.64.1.1	g_sFontCm46b	125
6.65	grlib/fonts/fontcm46i.c File Reference	125
6.65.1	Variable Documentation	126
6.65.1.1	g_sFontCm46i	126
6.66	grlib/fonts/fontcm48.c File Reference	126
6.66.1	Variable Documentation	126
6.66.1.1	g_sFontCm48	126
6.67	grlib/fonts/fontcm48b.c File Reference	126
6.67.1	Variable Documentation	126

6.67.1.1	g_sFontCm48b	126
6.68	grlib/fonts/fontcm48i.c File Reference	126
6.68.1	Variable Documentation	126
6.68.1.1	g_sFontCm48i	126
6.69	grlib/fonts/fontcmsc12.c File Reference	127
6.69.1	Variable Documentation	127
6.69.1.1	g_sFontCmsc12	127
6.70	grlib/fonts/fontcmsc14.c File Reference	127
6.70.1	Variable Documentation	127
6.70.1.1	g_sFontCmsc14	127
6.71	grlib/fonts/fontcmsc16.c File Reference	127
6.71.1	Variable Documentation	127
6.71.1.1	g_sFontCmsc16	127
6.72	grlib/fonts/fontcmsc18.c File Reference	127
6.72.1	Variable Documentation	128
6.72.1.1	g_sFontCmsc18	128
6.73	grlib/fonts/fontcmsc20.c File Reference	128
6.73.1	Variable Documentation	128
6.73.1.1	g_sFontCmsc20	128
6.74	grlib/fonts/fontcmsc22.c File Reference	128
6.74.1	Variable Documentation	128
6.74.1.1	g_sFontCmsc22	128
6.75	grlib/fonts/fontcmsc24.c File Reference	128
6.75.1	Variable Documentation	129
6.75.1.1	g_sFontCmsc24	129
6.76	grlib/fonts/fontcmsc26.c File Reference	129
6.76.1	Variable Documentation	129
6.76.1.1	g_sFontCmsc26	129
6.77	grlib/fonts/fontcmsc28.c File Reference	129
6.77.1	Variable Documentation	129
6.77.1.1	g_sFontCmsc28	129
6.78	grlib/fonts/fontcmsc30.c File Reference	129
6.78.1	Variable Documentation	129
6.78.1.1	g_sFontCmsc30	129

6.79 grlib/fonts/fontcmsc32.c File Reference	130
6.79.1 Variable Documentation	130
6.79.1.1 g_sFontCmsc32	130
6.80 grlib/fonts/fontcmsc34.c File Reference	130
6.80.1 Variable Documentation	130
6.80.1.1 g_sFontCmsc34	130
6.81 grlib/fonts/fontcmsc36.c File Reference	130
6.81.1 Variable Documentation	130
6.81.1.1 g_sFontCmsc36	130
6.82 grlib/fonts/fontcmsc38.c File Reference	130
6.82.1 Variable Documentation	131
6.82.1.1 g_sFontCmsc38	131
6.83 grlib/fonts/fontcmsc40.c File Reference	131
6.83.1 Variable Documentation	131
6.83.1.1 g_sFontCmsc40	131
6.84 grlib/fonts/fontcmsc42.c File Reference	131
6.84.1 Variable Documentation	131
6.84.1.1 g_sFontCmsc42	131
6.85 grlib/fonts/fontcmsc44.c File Reference	131
6.85.1 Variable Documentation	132
6.85.1.1 g_sFontCmsc44	132
6.86 grlib/fonts/fontcmsc46.c File Reference	132
6.86.1 Variable Documentation	132
6.86.1.1 g_sFontCmsc46	132
6.87 grlib/fonts/fontcmsc48.c File Reference	132
6.87.1 Variable Documentation	132
6.87.1.1 g_sFontCmsc48	132
6.88 grlib/fonts/fontcmss12.c File Reference	132
6.88.1 Variable Documentation	132
6.88.1.1 g_sFontCmss12	132
6.89 grlib/fonts/fontcmss12b.c File Reference	133
6.89.1 Variable Documentation	133
6.89.1.1 g_sFontCmss12b	133
6.90 grlib/fonts/fontcmss12i.c File Reference	133

6.90.1	Variable Documentation	133
6.90.1.1	g_sFontCmss12i	133
6.91	grlib/fonts/fontcmss14.c File Reference	133
6.91.1	Variable Documentation	133
6.91.1.1	g_sFontCmss14	133
6.92	grlib/fonts/fontcmss14b.c File Reference	133
6.92.1	Variable Documentation	134
6.92.1.1	g_sFontCmss14b	134
6.93	grlib/fonts/fontcmss14i.c File Reference	134
6.93.1	Variable Documentation	134
6.93.1.1	g_sFontCmss14i	134
6.94	grlib/fonts/fontcmss16.c File Reference	134
6.94.1	Variable Documentation	134
6.94.1.1	g_sFontCmss16	134
6.95	grlib/fonts/fontcmss16b.c File Reference	134
6.95.1	Variable Documentation	135
6.95.1.1	g_sFontCmss16b	135
6.96	grlib/fonts/fontcmss16i.c File Reference	135
6.96.1	Variable Documentation	135
6.96.1.1	g_sFontCmss16i	135
6.97	grlib/fonts/fontcmss18.c File Reference	135
6.97.1	Variable Documentation	135
6.97.1.1	g_sFontCmss18	135
6.98	grlib/fonts/fontcmss18b.c File Reference	135
6.98.1	Variable Documentation	135
6.98.1.1	g_sFontCmss18b	135
6.99	grlib/fonts/fontcmss18i.c File Reference	136
6.99.1	Variable Documentation	136
6.99.1.1	g_sFontCmss18i	136
6.100	grlib/fonts/fontcmss20.c File Reference	136
6.100.1	Variable Documentation	136
6.100.1.1	g_sFontCmss20	136
6.101	grlib/fonts/fontcmss20b.c File Reference	136
6.101.1	Variable Documentation	136

6.101.1.1 g_sFontCmss20b	136
6.102grlib/fonts/fontcmss20i.c File Reference	136
6.102.1 Variable Documentation	137
6.102.1.1 g_sFontCmss20i	137
6.103grlib/fonts/fontcmss22.c File Reference	137
6.103.1 Variable Documentation	137
6.103.1.1 g_sFontCmss22	137
6.104grlib/fonts/fontcmss22b.c File Reference	137
6.104.1 Variable Documentation	137
6.104.1.1 g_sFontCmss22b	137
6.105grlib/fonts/fontcmss22i.c File Reference	137
6.105.1 Variable Documentation	138
6.105.1.1 g_sFontCmss22i	138
6.106grlib/fonts/fontcmss24.c File Reference	138
6.106.1 Variable Documentation	138
6.106.1.1 g_sFontCmss24	138
6.107grlib/fonts/fontcmss24b.c File Reference	138
6.107.1 Variable Documentation	138
6.107.1.1 g_sFontCmss24b	138
6.108grlib/fonts/fontcmss24i.c File Reference	138
6.108.1 Variable Documentation	138
6.108.1.1 g_sFontCmss24i	138
6.109grlib/fonts/fontcmss26.c File Reference	139
6.109.1 Variable Documentation	139
6.109.1.1 g_sFontCmss26	139
6.110grlib/fonts/fontcmss26b.c File Reference	139
6.110.1 Variable Documentation	139
6.110.1.1 g_sFontCmss26b	139
6.111grlib/fonts/fontcmss26i.c File Reference	139
6.111.1 Variable Documentation	139
6.111.1.1 g_sFontCmss26i	139
6.112grlib/fonts/fontcmss28.c File Reference	139
6.112.1 Variable Documentation	140
6.112.1.1 g_sFontCmss28	140

6.113grlib/fonts/fontcmss28b.c File Reference	140
6.113.1 Variable Documentation	140
6.113.1.1 g_sFontCmss28b	140
6.114grlib/fonts/fontcmss28i.c File Reference	140
6.114.1 Variable Documentation	140
6.114.1.1 g_sFontCmss28i	140
6.115grlib/fonts/fontcmss30.c File Reference	140
6.115.1 Variable Documentation	141
6.115.1.1 g_sFontCmss30	141
6.116grlib/fonts/fontcmss30b.c File Reference	141
6.116.1 Variable Documentation	141
6.116.1.1 g_sFontCmss30b	141
6.117grlib/fonts/fontcmss30i.c File Reference	141
6.117.1 Variable Documentation	141
6.117.1.1 g_sFontCmss30i	141
6.118grlib/fonts/fontcmss32.c File Reference	141
6.118.1 Variable Documentation	141
6.118.1.1 g_sFontCmss32	141
6.119grlib/fonts/fontcmss32b.c File Reference	142
6.119.1 Variable Documentation	142
6.119.1.1 g_sFontCmss32b	142
6.120grlib/fonts/fontcmss32i.c File Reference	142
6.120.1 Variable Documentation	142
6.120.1.1 g_sFontCmss32i	142
6.121grlib/fonts/fontcmss34.c File Reference	142
6.121.1 Variable Documentation	142
6.121.1.1 g_sFontCmss34	142
6.122grlib/fonts/fontcmss34b.c File Reference	142
6.122.1 Variable Documentation	143
6.122.1.1 g_sFontCmss34b	143
6.123grlib/fonts/fontcmss34i.c File Reference	143
6.123.1 Variable Documentation	143
6.123.1.1 g_sFontCmss34i	143
6.124grlib/fonts/fontcmss36.c File Reference	143

6.124.1 Variable Documentation	143
6.124.1.1 g_sFontCmss36	143
6.125grlib/fonts/fontcmss36b.c File Reference	143
6.125.1 Variable Documentation	144
6.125.1.1 g_sFontCmss36b	144
6.126grlib/fonts/fontcmss36i.c File Reference	144
6.126.1 Variable Documentation	144
6.126.1.1 g_sFontCmss36i	144
6.127grlib/fonts/fontcmss38.c File Reference	144
6.127.1 Variable Documentation	144
6.127.1.1 g_sFontCmss38	144
6.128grlib/fonts/fontcmss38b.c File Reference	144
6.128.1 Variable Documentation	144
6.128.1.1 g_sFontCmss38b	144
6.129grlib/fonts/fontcmss38i.c File Reference	145
6.129.1 Variable Documentation	145
6.129.1.1 g_sFontCmss38i	145
6.130grlib/fonts/fontcmss40.c File Reference	145
6.130.1 Variable Documentation	145
6.130.1.1 g_sFontCmss40	145
6.131grlib/fonts/fontcmss40b.c File Reference	145
6.131.1 Variable Documentation	145
6.131.1.1 g_sFontCmss40b	145
6.132grlib/fonts/fontcmss40i.c File Reference	145
6.132.1 Variable Documentation	146
6.132.1.1 g_sFontCmss40i	146
6.133grlib/fonts/fontcmss42.c File Reference	146
6.133.1 Variable Documentation	146
6.133.1.1 g_sFontCmss42	146
6.134grlib/fonts/fontcmss42b.c File Reference	146
6.134.1 Variable Documentation	146
6.134.1.1 g_sFontCmss42b	146
6.135grlib/fonts/fontcmss42i.c File Reference	146
6.135.1 Variable Documentation	147

6.135.1.1 <code>g_sFontCmss42i</code>	147
6.136grlib/fonts/fontcmss44.c File Reference	147
6.136.1 Variable Documentation	147
6.136.1.1 <code>g_sFontCmss44</code>	147
6.137grlib/fonts/fontcmss44b.c File Reference	147
6.137.1 Variable Documentation	147
6.137.1.1 <code>g_sFontCmss44b</code>	147
6.138grlib/fonts/fontcmss44i.c File Reference	147
6.138.1 Variable Documentation	147
6.138.1.1 <code>g_sFontCmss44i</code>	147
6.139grlib/fonts/fontcmss46.c File Reference	148
6.139.1 Variable Documentation	148
6.139.1.1 <code>g_sFontCmss46</code>	148
6.140grlib/fonts/fontcmss46b.c File Reference	148
6.140.1 Variable Documentation	148
6.140.1.1 <code>g_sFontCmss46b</code>	148
6.141grlib/fonts/fontcmss46i.c File Reference	148
6.141.1 Variable Documentation	148
6.141.1.1 <code>g_sFontCmss46i</code>	148
6.142grlib/fonts/fontcmss48.c File Reference	148
6.142.1 Variable Documentation	149
6.142.1.1 <code>g_sFontCmss48</code>	149
6.143grlib/fonts/fontcmss48b.c File Reference	149
6.143.1 Variable Documentation	149
6.143.1.1 <code>g_sFontCmss48b</code>	149
6.144grlib/fonts/fontcmss48i.c File Reference	149
6.144.1 Variable Documentation	149
6.144.1.1 <code>g_sFontCmss48i</code>	149
6.145grlib/fonts/fontfixed6x8.c File Reference	149
6.145.1 Variable Documentation	150
6.145.1.1 <code>g_sFontFixed6x8</code>	150
6.146grlib/image.c File Reference	150
6.146.1 Function Documentation	150
6.146.1.1 <code>GrlImageDraw</code>	150

6.147grlib/imgbutton.c File Reference	150
6.147.1 Function Documentation	151
6.147.1.1 ImageButtonInit	151
6.147.1.2 ImageButtonMsgProc	151
6.148grlib/line.c File Reference	152
6.148.1 Function Documentation	152
6.148.1.1 GrLineDraw	152
6.148.1.2 GrLineDrawH	152
6.148.1.3 GrLineDrawV	153
6.149grlib/listbox.c File Reference	153
6.149.1 Define Documentation	154
6.149.1.1 abs	154
6.149.1.2 max	154
6.149.1.3 min	154
6.150grlib/offscr1bpp.c File Reference	154
6.151grlib/offscr4bpp.c File Reference	154
6.152grlib/offscr8bpp.c File Reference	155
6.153grlib/pushbutton.c File Reference	155
6.154grlib/radiobutton.c File Reference	155
6.155grlib/rectangle.c File Reference	155
6.155.1 Define Documentation	156
6.155.1.1 max	156
6.155.1.2 min	156
6.155.2 Function Documentation	156
6.155.2.1 GrRectDraw	156
6.155.2.2 GrRectFill	156
6.155.2.3 GrRectIntersectGet	157
6.155.2.4 GrRectOverlapCheck	157
6.156grlib/slider.c File Reference	158
6.156.1 Define Documentation	158
6.156.1.1 max	158
6.156.1.2 min	158
6.156.2 Function Documentation	158
6.156.2.1 SliderInit	158

6.156.2.2 SliderMsgProc	159
6.157 grlib/string.c File Reference	159
6.157.1 Define Documentation	160
6.157.1.1 ABSENT_CHAR_REPLACEMENT	160
6.157.1.2 SC_FLAG_COMPRESSED	160
6.157.1.3 SC_GET_INDEX	160
6.157.1.4 SC_GET_LEN	160
6.157.1.5 SC_GET_OFF	160
6.157.1.6 SC_IS_NULL	160
6.157.1.7 SC_MAX_INDEX	160
6.157.1.8 SC_OFFSET_M	160
6.157.2 Function Documentation	160
6.157.2.1 GrStringDraw	160
6.157.2.2 GrStringGet	161
6.157.2.3 GrStringLanguageSet	161
6.157.2.4 GrStringTableSet	161
6.157.2.5 GrStringWidthGet	162
6.157.2.6 NumLeadingZeros	162
6.158 grlib/widget.c File Reference	162
6.158.1 Define Documentation	163
6.158.1.1 MQ_FLAG_POST_ORDER	163
6.158.1.2 MQ_FLAG_STOP_ON_SUCCESS	163
6.158.1.3 QUEUE_SIZE	163
6.158.2 Function Documentation	163
6.158.2.1 WidgetAdd	163
6.158.2.2 WidgetDefaultMsgProc	164
6.158.2.3 WidgetMessageQueueAdd	164
6.158.2.4 WidgetMessageQueueProcess	165
6.158.2.5 WidgetMessageSendPostOrder	165
6.158.2.6 WidgetMessageSendPreOrder	166
6.158.2.7 WidgetMutexGet	167
6.158.2.8 WidgetMutexInit	167
6.158.2.9 WidgetMutexPut	168
6.158.2.10 WidgetPointerMessage	168

6.158.2.11WidgetRemove	168
6.158.3 Variable Documentation	169
6.158.3.1 g_sRoot	169
6.159 include/build/lpc210x.h File Reference	169
6.159.1 Define Documentation	173
6.159.1.1 ALDOM	173
6.159.1.2 ALDOW	173
6.159.1.3 ALDOY	173
6.159.1.4 ALHOUR	173
6.159.1.5 ALMIN	173
6.159.1.6 ALMON	174
6.159.1.7 ALSEC	174
6.159.1.8 ALYEAR	174
6.159.1.9 AMR	174
6.159.1.10CCR	174
6.159.1.11CIIR	174
6.159.1.12CTC	174
6.159.1.13CTIME0	174
6.159.1.14CTIME1	174
6.159.1.15CTIME2	174
6.159.1.16DOM	174
6.159.1.17DOW	174
6.159.1.18DOY	174
6.159.1.19EXTINT	174
6.159.1.20EXTWAKE	174
6.159.1.21HOUR	174
6.159.1.22I2C_I2ADR	174
6.159.1.23I2C_I2CONCLR	174
6.159.1.24I2C_I2CONSET	174
6.159.1.25I2C_I2DAT	174
6.159.1.26I2C_I2SCLH	174
6.159.1.27I2C_I2SCLL	174
6.159.1.28I2C_I2STAT	174
6.159.1.29LR	174

6.159.1.30OCLR	175
6.159.1.31ODIR	175
6.159.1.32OPIN	175
6.159.1.33OSET	175
6.159.1.34MAMCR	175
6.159.1.35MAMMAP	175
6.159.1.36MAMTIM	175
6.159.1.37MIN	175
6.159.1.38MONTH	175
6.159.1.39PCON	175
6.159.1.40PCONP	175
6.159.1.41PINSEL0	175
6.159.1.42PINSEL1	175
6.159.1.43PLLCFG	175
6.159.1.44PLLCON	175
6.159.1.45PLLFEED	175
6.159.1.46PLLSTAT	175
6.159.1.47PREFRAC	175
6.159.1.48PREINT	175
6.159.1.49PWM_CCR	175
6.159.1.50PWM_CR0	175
6.159.1.51PWM_CR1	175
6.159.1.52PWM_CR2	175
6.159.1.53PWM_CR3	175
6.159.1.54PWM_EMR	176
6.159.1.55PWM_IR	176
6.159.1.56PWM_LER	176
6.159.1.57PWM_MCR	176
6.159.1.58PWM_MR0	176
6.159.1.59PWM_MR1	176
6.159.1.60PWM_MR2	176
6.159.1.61PWM_MR3	176
6.159.1.62PWM_MR4	176
6.159.1.63PWM_MR5	176

6.159.1.64PWM_MR6	176
6.159.1.65PWM_PC	176
6.159.1.66PWM_PCR	176
6.159.1.67PWM_PR	176
6.159.1.68PWM_TC	176
6.159.1.69PWM_TCR	176
6.159.1.70SEC	176
6.159.1.71SPI_SPCCR	176
6.159.1.72SPI_SPCR	176
6.159.1.73SPI_SPDR	176
6.159.1.74SPI_SPINT	176
6.159.1.75SPI_SPSR	176
6.159.1.76SPI_SPTCR	176
6.159.1.77SPI_SPTOR	176
6.159.1.78SPI_SPTSR	177
6.159.1.79T0CCR	177
6.159.1.80T0CR0	177
6.159.1.81T0CR1	177
6.159.1.82T0CR2	177
6.159.1.83T0CR3	177
6.159.1.84T0EMR	177
6.159.1.85T0IR	177
6.159.1.86T0MCR	177
6.159.1.87T0MR0	177
6.159.1.88T0MR1	177
6.159.1.89T0MR2	177
6.159.1.90T0MR3	177
6.159.1.91T0PC	177
6.159.1.92T0PR	177
6.159.1.93T0TC	177
6.159.1.94T0TCR	177
6.159.1.95T1CCR	177
6.159.1.96T1CR0	177
6.159.1.97T1CR1	177

6.159.1.98T1CR2	177
6.159.1.99T1CR3	177
6.159.1.1001EMR	177
6.159.1.1011IR	177
6.159.1.1021MCR	178
6.159.1.1031MR0	178
6.159.1.1041MR1	178
6.159.1.1051MR2	178
6.159.1.1061MR3	178
6.159.1.1071PC	178
6.159.1.1081PR	178
6.159.1.1091TC	178
6.159.1.1101TCR	178
6.159.1.1110DLL	178
6.159.1.1120DLM	178
6.159.1.1130FCR	178
6.159.1.1140IER	178
6.159.1.1150IIR	178
6.159.1.1160LCR	178
6.159.1.1170LSR	178
6.159.1.1180MCR	178
6.159.1.1190MSR	178
6.159.1.1200RBR	178
6.159.1.1210SCR	178
6.159.1.1220THR	178
6.159.1.1231DLL	178
6.159.1.1241DLM	178
6.159.1.1251FCR	178
6.159.1.1261IER	179
6.159.1.1271IIR	179
6.159.1.1281LCR	179
6.159.1.1291LSR	179
6.159.1.1301MCR	179
6.159.1.1311MSR	179

6.159.1.1301RBR	179
6.159.1.1301SCR	179
6.159.1.1301THR	179
6.159.1.1301ICDefVectAddr	179
6.159.1.1301ICFIQStatus	179
6.159.1.1301ICIntEnable	179
6.159.1.1301ICIntEnClr	179
6.159.1.1301ICIntSelect	179
6.159.1.1401CIRQStatus	179
6.159.1.1401VICProtection	179
6.159.1.1401ICRawIntr	179
6.159.1.1401ICSoflnt	179
6.159.1.1401ICSoftIntClr	179
6.159.1.1401ICVectAddr	179
6.159.1.1401ICVectAddr0	179
6.159.1.1401ICVectAddr1	179
6.159.1.1401ICVectAddr10	179
6.159.1.1401ICVectAddr11	179
6.159.1.1501ICVectAddr12	180
6.159.1.1501ICVectAddr13	180
6.159.1.1501ICVectAddr14	180
6.159.1.1501ICVectAddr15	180
6.159.1.1501ICVectAddr2	180
6.159.1.1501ICVectAddr3	180
6.159.1.1501ICVectAddr4	180
6.159.1.1501ICVectAddr5	180
6.159.1.1501ICVectAddr6	180
6.159.1.1501ICVectAddr7	180
6.159.1.1601ICVectAddr8	180
6.159.1.1601ICVectAddr9	180
6.159.1.1601ICVectCtl0	180
6.159.1.1601ICVectCtl1	180
6.159.1.1601ICVectCtl10	180
6.159.1.1601ICVectCtl11	180

6.159.1.16 ⁸ ICVectCntl12	180
6.159.1.16 ⁷ ICVectCntl13	180
6.159.1.16 ⁸ ICVectCntl14	180
6.159.1.16 ⁰ ICVectCntl15	180
6.159.1.17 ⁰ ICVectCntl2	180
6.159.1.17 ¹ ICVectCntl3	180
6.159.1.17 ² ICVectCntl4	180
6.159.1.17 ³ ICVectCntl5	180
6.159.1.17 ⁴ ICVectCntl6	181
6.159.1.17 ⁵ ICVectCntl7	181
6.159.1.17 ⁶ ICVectCntl8	181
6.159.1.17 ⁷ ICVectCntl9	181
6.159.1.17 ⁸ PBDIV	181
6.159.1.17 ⁰ DFEED	181
6.159.1.18 ⁰ DMOD	181
6.159.1.18 ¹ WDTC	181
6.159.1.18 ² WDTV	181
6.159.1.18 ³ EAR	181
6.160include/build/lpc213x.h File Reference	181
6.160.1 Define Documentation	186
6.160.1.1 AD0CR	186
6.160.1.2 AD0DR	186
6.160.1.3 AD1CR	186
6.160.1.4 AD1DR	186
6.160.1.5 ALDOM	186
6.160.1.6 ALDOW	186
6.160.1.7 ALDOY	186
6.160.1.8 ALHOUR	186
6.160.1.9 ALMIN	186
6.160.1.10ALMON	186
6.160.1.11ALSEC	186
6.160.1.12ALYEAR	186
6.160.1.13AMR	186
6.160.1.14CCR	186

6.160.1.15CIIR	187
6.160.1.16CSPR	187
6.160.1.17CTC	187
6.160.1.18CTIME0	187
6.160.1.19CTIME1	187
6.160.1.20CTIME2	187
6.160.1.21DACR	187
6.160.1.22DOM	187
6.160.1.23DOW	187
6.160.1.24DOY	187
6.160.1.25EXINT	187
6.160.1.26EXTINT	187
6.160.1.27EXTMODE	187
6.160.1.28EXTPOLAR	187
6.160.1.29FIO0CLR	187
6.160.1.30FIO0DIR	187
6.160.1.31FIO0MASK	187
6.160.1.32FIO0PIN	187
6.160.1.33FIO0SET	187
6.160.1.34FIO1CLR	187
6.160.1.35FIO1DIR	187
6.160.1.36FIO1MASK	187
6.160.1.37FIO1PIN	187
6.160.1.38FIO1SET	187
6.160.1.39HOUR	188
6.160.1.40I20ADR	188
6.160.1.41I20CONCLR	188
6.160.1.42I20CONSET	188
6.160.1.43I20DAT	188
6.160.1.44I20SCLH	188
6.160.1.45I20SCLL	188
6.160.1.46I20STAT	188
6.160.1.47I21ADR	188
6.160.1.48I21CONCLR	188

6.160.1.49	CONSET	188
6.160.1.50	21DAT	188
6.160.1.51	121SCLH	188
6.160.1.52	21SCLL	188
6.160.1.53	21STAT	188
6.160.1.54	LR	188
6.160.1.55	NTWAKE	188
6.160.1.56	OCLR0	188
6.160.1.57	OCLR1	188
6.160.1.58	ODIR0	188
6.160.1.59	ODIR1	188
6.160.1.60	OPIN0	188
6.160.1.61	OPIN1	188
6.160.1.62	OSET0	188
6.160.1.63	OSET1	189
6.160.1.64	MAMCR	189
6.160.1.65	MAMTIM	189
6.160.1.66	MEMMAP	189
6.160.1.67	MIN	189
6.160.1.68	MONTH	189
6.160.1.69	PCON	189
6.160.1.70	PCONP	189
6.160.1.71	PINSEL0	189
6.160.1.72	PINSEL1	189
6.160.1.73	PINSEL2	189
6.160.1.74	PLLCFG	189
6.160.1.75	PLLCON	189
6.160.1.76	PLLFEED	189
6.160.1.77	PLLSTAT	189
6.160.1.78	PREFRAC	189
6.160.1.79	PREINT	189
6.160.1.80	PWMEMR	189
6.160.1.81	PWMIR	189
6.160.1.82	PWMLER	189

6.160.1.83PWMMCR	189
6.160.1.84PWMMR0	189
6.160.1.85PWMMR1	189
6.160.1.86PWMMR2	189
6.160.1.87PWMMR3	190
6.160.1.88PWMMR4	190
6.160.1.89PWMMR5	190
6.160.1.90PWMMR6	190
6.160.1.91PWMPC	190
6.160.1.92PWMPCR	190
6.160.1.93PWMPR	190
6.160.1.94PWMTC	190
6.160.1.95PWMTCR	190
6.160.1.96RSID	190
6.160.1.97S0SPCCR	190
6.160.1.98S0SPCR	190
6.160.1.99S0SPDR	190
6.160.1.10S0SPINT	190
6.160.1.10S0SPSR	190
6.160.1.10S0SPTCR	190
6.160.1.10S0SPTOR	190
6.160.1.10S0SPTSR	190
6.160.1.10SEC	190
6.160.1.10SPCPSR	190
6.160.1.10SSPCR0	190
6.160.1.10SSPCR1	190
6.160.1.10SSPDMACR	190
6.160.1.11SSPDR	190
6.160.1.11SSPICR	191
6.160.1.11SSPIMSC	191
6.160.1.11SSPMIS	191
6.160.1.11SSPRIS	191
6.160.1.11SSPSR	191
6.160.1.11T0CCR	191

6.160.1.11 T 0CR0	191
6.160.1.11 B 0CR1	191
6.160.1.11 T 0CR2	191
6.160.1.12 D 0CR3	191
6.160.1.12 T 0CTCR	191
6.160.1.12 D 0EMR	191
6.160.1.12 B 0IR	191
6.160.1.12 D 0MCR	191
6.160.1.12 E 0MR0	191
6.160.1.12 E 0MR1	191
6.160.1.12 T 0MR2	191
6.160.1.12 B 0MR3	191
6.160.1.12 D 0PC	191
6.160.1.13 D 0PR	191
6.160.1.13 T 0TC	191
6.160.1.13 D 0TCR	191
6.160.1.13 B 1CCR	191
6.160.1.13 D 1CR0	191
6.160.1.13 B 1CR1	192
6.160.1.13 B 1CR2	192
6.160.1.13 T 1CR3	192
6.160.1.13 B 1CTCR	192
6.160.1.13 D 1EMR	192
6.160.1.14 D 1IR	192
6.160.1.14 T 1MCR	192
6.160.1.14 D 1MR0	192
6.160.1.14 B 1MR1	192
6.160.1.14 D 1MR2	192
6.160.1.14 B 1MR3	192
6.160.1.14 D 1PC	192
6.160.1.14 T 1PR	192
6.160.1.14 B 1TC	192
6.160.1.14 D 1TCR	192
6.160.1.15 D 0DLL	192

6.160.1.150DLM	192
6.160.1.150FCR	192
6.160.1.150IER	192
6.160.1.150IIR	192
6.160.1.150LCR	192
6.160.1.150LSR	192
6.160.1.150RBR	192
6.160.1.150SCR	192
6.160.1.150TER	193
6.160.1.1600THR	193
6.160.1.1601DLL	193
6.160.1.1601DLM	193
6.160.1.1601FCR	193
6.160.1.1601IER	193
6.160.1.1601IIR	193
6.160.1.1601LCR	193
6.160.1.1601LSR	193
6.160.1.1601MCR	193
6.160.1.1601MSR	193
6.160.1.1701RBR	193
6.160.1.1701SCR	193
6.160.1.1701TER	193
6.160.1.1701THR	193
6.160.1.170ICDefVectAddr	193
6.160.1.170ICFIQStatus	193
6.160.1.170ICIntEnable	193
6.160.1.170ICIntEnClr	193
6.160.1.170ICIntSelect	193
6.160.1.170ICIRQStatus	193
6.160.1.180ICProtection	193
6.160.1.180ICRawIntr	193
6.160.1.180ICSoftInt	193
6.160.1.180ICSoftIntClr	194
6.160.1.180ICVectAddr	194

6.160.1.18 ⁸ ICVectAddr0	194
6.160.1.18 ⁹ ICVectAddr1	194
6.160.1.18 ⁷ ICVectAddr10	194
6.160.1.18 ⁸ ICVectAddr11	194
6.160.1.18 ⁹ ICVectAddr12	194
6.160.1.19 ⁰ ICVectAddr13	194
6.160.1.19 ¹ ICVectAddr14	194
6.160.1.19 ² ICVectAddr15	194
6.160.1.19 ³ ICVectAddr2	194
6.160.1.19 ⁴ ICVectAddr3	194
6.160.1.19 ⁵ ICVectAddr4	194
6.160.1.19 ⁶ ICVectAddr5	194
6.160.1.19 ⁷ ICVectAddr6	194
6.160.1.19 ⁸ ICVectAddr7	194
6.160.1.19 ⁹ ICVectAddr8	194
6.160.1.20 ⁰ ICVectAddr9	194
6.160.1.20 ¹ ICVectCtl0	194
6.160.1.20 ² ICVectCtl1	194
6.160.1.20 ³ ICVectCtl10	194
6.160.1.20 ⁴ ICVectCtl11	194
6.160.1.20 ⁵ ICVectCtl12	194
6.160.1.20 ⁶ ICVectCtl13	194
6.160.1.20 ⁷ ICVectCtl14	195
6.160.1.20 ⁸ ICVectCtl15	195
6.160.1.20 ⁹ ICVectCtl2	195
6.160.1.21 ⁰ ICVectCtl3	195
6.160.1.21 ¹ ICVectCtl4	195
6.160.1.21 ² ICVectCtl5	195
6.160.1.21 ³ ICVectCtl6	195
6.160.1.21 ⁴ ICVectCtl7	195
6.160.1.21 ⁵ ICVectCtl8	195
6.160.1.21 ⁶ ICVectCtl9	195
6.160.1.21 ⁷ PBDIV	195
6.160.1.21 ⁸ DFEED	195

6.160.1.21WDMOD	195
6.160.1.22WDTC	195
6.160.1.22WDTV	195
6.160.1.22WEAR	195
6.161include/build/lpc23xx.h File Reference	195
6.161.1 Define Documentation	219
6.161.1.1 __LPC23xx_H	219
6.161.1.2 AD0_BASE_ADDR	219
6.161.1.3 AD0CR	219
6.161.1.4 AD0DR0	219
6.161.1.5 AD0DR1	219
6.161.1.6 AD0DR2	220
6.161.1.7 AD0DR3	220
6.161.1.8 AD0DR4	220
6.161.1.9 AD0DR5	220
6.161.1.10AD0DR6	220
6.161.1.11AD0DR7	220
6.161.1.12AD0GDR	220
6.161.1.13AD0INTEN	220
6.161.1.14AD0STAT	220
6.161.1.15AHBCFG1	220
6.161.1.16AHBCFG2	220
6.161.1.17CAN1_BASE_ADDR	220
6.161.1.18CAN1BTR	220
6.161.1.19CAN1CMR	220
6.161.1.20CAN1EWL	220
6.161.1.21CAN1GSR	220
6.161.1.22CAN1ICR	220
6.161.1.23CAN1IER	220
6.161.1.24CAN1MOD	220
6.161.1.25CAN1RDA	220
6.161.1.26CAN1RDB	220
6.161.1.27CAN1RFS	220
6.161.1.28CAN1RID	220

6.161.1.29CAN1SR	220
6.161.1.30CAN1TDA1	221
6.161.1.31CAN1TDA2	221
6.161.1.32CAN1TDA3	221
6.161.1.33CAN1TDB1	221
6.161.1.34CAN1TDB2	221
6.161.1.35CAN1TDB3	221
6.161.1.36CAN1TFI1	221
6.161.1.37CAN1TFI2	221
6.161.1.38CAN1TFI3	221
6.161.1.39CAN1TID1	221
6.161.1.40CAN1TID2	221
6.161.1.41CAN1TID3	221
6.161.1.42CAN2_BASE_ADDR	221
6.161.1.43CAN2BTR	221
6.161.1.44CAN2CMR	221
6.161.1.45CAN2EWL	221
6.161.1.46CAN2GSR	221
6.161.1.47CAN2ICR	221
6.161.1.48CAN2IER	221
6.161.1.49CAN2MOD	221
6.161.1.50CAN2RDA	221
6.161.1.51CAN2RDB	221
6.161.1.52CAN2RFS	221
6.161.1.53CAN2RID	221
6.161.1.54CAN2SR	222
6.161.1.55CAN2TDA1	222
6.161.1.56CAN2TDA2	222
6.161.1.57CAN2TDA3	222
6.161.1.58CAN2TDB1	222
6.161.1.59CAN2TDB2	222
6.161.1.60CAN2TDB3	222
6.161.1.61CAN2TFI1	222
6.161.1.62CAN2TFI2	222

6.161.1.63CAN2TFI3	222
6.161.1.64CAN2TID1	222
6.161.1.65CAN2TID2	222
6.161.1.66CAN2TID3	222
6.161.1.67CAN_ACCEPT_BASE_ADDR	222
6.161.1.68CAN_AFMR	222
6.161.1.69CAN_CENTRAL_BASE_ADDR	222
6.161.1.70CAN_EFF_GRP_SA	222
6.161.1.71CAN_EFF_SA	222
6.161.1.72CAN_EOT	222
6.161.1.73CAN_LUT_ERR	222
6.161.1.74CAN_LUT_ERR_ADR	222
6.161.1.75CAN_MSR	223
6.161.1.76CAN_RX_SR	223
6.161.1.77CAN_SFF_GRP_SA	223
6.161.1.78CAN_SFF_SA	223
6.161.1.79CAN_TX_SR	223
6.161.1.80CCLKCFG	223
6.161.1.81CLKSRCSEL	223
6.161.1.82CMD_CODE	223
6.161.1.83CMD_DATA	223
6.161.1.84CSPR	223
6.161.1.85DAC_BASE_ADDR	223
6.161.1.86DACR	223
6.161.1.87DEV_INT_CLR	223
6.161.1.88DEV_INT_EN	223
6.161.1.89DEV_INT_PRIO	223
6.161.1.90DEV_INT_SET	223
6.161.1.91DEV_INT_STAT	223
6.161.1.92DMA_BASE_ADDR	223
6.161.1.93DMA_INT_EN	223
6.161.1.94DMA_INT_STAT	223
6.161.1.95DMA_REQ_CLR	224
6.161.1.96DMA_REQ_SET	224

6.161.1.97DMA_REQ_STAT	224
6.161.1.98DYNAMIC_MEM0_BASE	224
6.161.1.99DYNAMIC_MEM1_BASE	224
6.161.1.100DYNAMIC_MEM2_BASE	224
6.161.1.101DYNAMIC_MEM3_BASE	224
6.161.1.102BMC_BASE_ADDR	224
6.161.1.103BMC_CONFIG	224
6.161.1.104BMC_CTRL	224
6.161.1.105BMC_DYN_APP	224
6.161.1.106BMC_DYN_CFG0	224
6.161.1.107BMC_DYN_CFG1	224
6.161.1.108BMC_DYN_CFG2	224
6.161.1.109BMC_DYN_CFG3	224
6.161.1.110BMC_DYN_CTRL	224
6.161.1.111BMC_DYN_DAL	224
6.161.1.112BMC_DYN_MRД	224
6.161.1.113BMC_DYN_RAS	225
6.161.1.114BMC_DYN_RASCAS0	225
6.161.1.115BMC_DYN_RASCAS1	225
6.161.1.116BMC_DYN_RASCAS2	225
6.161.1.117BMC_DYN_RASCAS3	225
6.161.1.118BMC_DYN_RC	225
6.161.1.119BMC_DYN_RD_CFG	225
6.161.1.120BMC_DYN_RFC	225
6.161.1.121BMC_DYN_RFSH	225
6.161.1.122BMC_DYN_RP	225
6.161.1.123BMC_DYN_RRD	225
6.161.1.124BMC_DYN_SREX	225
6.161.1.125BMC_DYN_WR	225
6.161.1.126BMC_DYN_XSR	225
6.161.1.127BMC_STA_CFG0	225
6.161.1.128BMC_STA_CFG1	225
6.161.1.129BMC_STA_CFG2	226
6.161.1.130BMC_STA_CFG3	226

6.161.1.13 ¹ MC_STA_EXT_WAIT	226
6.161.1.13 ² MC_STA_WAITOEN0	226
6.161.1.13 ³ MC_STA_WAITOEN1	226
6.161.1.13 ⁴ MC_STA_WAITOEN2	226
6.161.1.13 ⁵ MC_STA_WAITOEN3	226
6.161.1.13 ⁶ MC_STA_WAITPAGE0	226
6.161.1.13 ⁷ MC_STA_WAITPAGE1	226
6.161.1.13 ⁸ MC_STA_WAITPAGE2	226
6.161.1.13 ⁹ MC_STA_WAITPAGE3	226
6.161.1.14 ¹⁰ MC_STA_WAITRD0	226
6.161.1.14 ¹¹ MC_STA_WAITRD1	226
6.161.1.14 ¹² MC_STA_WAITRD2	226
6.161.1.14 ¹³ MC_STA_WAITRD3	226
6.161.1.14 ¹⁴ MC_STA_WAITTURN0	226
6.161.1.14 ¹⁵ MC_STA_WAITTURN1	227
6.161.1.14 ¹⁶ MC_STA_WAITTURN2	227
6.161.1.14 ¹⁷ MC_STA_WAITTURN3	227
6.161.1.14 ¹⁸ MC_STA_WAITWEN0	227
6.161.1.14 ¹⁹ MC_STA_WAITWEN1	227
6.161.1.15 ²⁰ MC_STA_WAITWEN2	227
6.161.1.15 ²¹ MC_STA_WAITWEN3	227
6.161.1.15 ²² MC_STA_WAITWR0	227
6.161.1.15 ²³ MC_STA_WAITWR1	227
6.161.1.15 ²⁴ MC_STA_WAITWR2	227
6.161.1.15 ²⁵ MC_STA_WAITWR3	227
6.161.1.15 ²⁶ MC_STAT	227
6.161.1.15 ²⁷ OT_INT_CLR	227
6.161.1.15 ²⁸ OT_INT_SET	227
6.161.1.15 ²⁹ OT_INT_STAT	227
6.161.1.16 ³⁰ P_DMA_DIS	227
6.161.1.16 ³¹ P_DMA_EN	227
6.161.1.16 ³² P_DMA_STAT	228
6.161.1.16 ³³ P_INDEX	228
6.161.1.16 ³⁴ P_INT_CLR	228

6.161.1.16 ⁵ P_INT_EN	228
6.161.1.16 ⁶ P_INT_PRIO	228
6.161.1.16 ⁷ P_INT_SET	228
6.161.1.16 ⁸ P_INT_STAT	228
6.161.1.16 ⁹ XTINT	228
6.161.1.17 ⁰ XTMODE	228
6.161.1.17 ¹ XTPOLAR	228
6.161.1.17 ² IO0CLR	228
6.161.1.17 ³ IO0CLR0	228
6.161.1.17 ⁴ IO0CLR1	228
6.161.1.17 ⁵ IO0CLR2	228
6.161.1.17 ⁶ IO0CLR3	228
6.161.1.17 ⁷ IO0CLRL	228
6.161.1.17 ⁸ IO0CLRU	228
6.161.1.17 ⁹ IO0DIR	228
6.161.1.18 ⁰ IO0DIR0	228
6.161.1.18 ¹ IO0DIR1	228
6.161.1.18 ² IO0DIR2	228
6.161.1.18 ³ IO0DIR3	228
6.161.1.18 ⁴ IO0DIRL	228
6.161.1.18 ⁵ IO0DIRU	229
6.161.1.18 ⁶ IO0MASK	229
6.161.1.18 ⁷ IO0MASK0	229
6.161.1.18 ⁸ IO0MASK1	229
6.161.1.18 ⁹ IO0MASK2	229
6.161.1.19 ⁰ IO0MASK3	229
6.161.1.19 ¹ IO0MASKL	229
6.161.1.19 ² IO0MASKU	229
6.161.1.19 ³ IO0PIN	229
6.161.1.19 ⁴ IO0PINO	229
6.161.1.19 ⁵ IO0PIN1	229
6.161.1.19 ⁶ IO0PIN2	229
6.161.1.19 ⁷ IO0PIN3	229
6.161.1.19 ⁸ IO0PINL	229

6.161.1.19IO0PINU	229
6.161.1.20IO0SET	229
6.161.1.20IO0SET0	229
6.161.1.20IO0SET1	229
6.161.1.20IO0SET2	229
6.161.1.20IO0SET3	229
6.161.1.20IO0SETL	229
6.161.1.20IO0SETU	229
6.161.1.20IO1CLR	229
6.161.1.20IO1CLR0	229
6.161.1.20IO1CLR1	230
6.161.1.21IO1CLR2	230
6.161.1.21IO1CLR3	230
6.161.1.21IO1CLRL	230
6.161.1.21IO1CLRU	230
6.161.1.21IO1DIR	230
6.161.1.21IO1DIR0	230
6.161.1.21IO1DIR1	230
6.161.1.21IO1DIR2	230
6.161.1.21IO1DIR3	230
6.161.1.21IO1DIRL	230
6.161.1.22IO1DIRU	230
6.161.1.22IO1MASK	230
6.161.1.22IO1MASK0	230
6.161.1.22IO1MASK1	230
6.161.1.22IO1MASK2	230
6.161.1.22IO1MASK3	230
6.161.1.22IO1MASKL	230
6.161.1.22IO1MASKU	230
6.161.1.22IO1PIN	230
6.161.1.22IO1PIN0	230
6.161.1.23IO1PIN1	230
6.161.1.23IO1PIN2	230
6.161.1.23IO1PIN3	230

6.161.1.23 5 IO1PINL	231
6.161.1.23 4 IO1PINU	231
6.161.1.23 5 IO1SET	231
6.161.1.23 6 IO1SET0	231
6.161.1.23 7 IO1SET1	231
6.161.1.23 8 IO1SET2	231
6.161.1.23 9 IO1SET3	231
6.161.1.24 0 IO1SETL	231
6.161.1.24 1 IO1SETU	231
6.161.1.24 2 IO2CLR	231
6.161.1.24 3 IO2CLR0	231
6.161.1.24 4 IO2CLR1	231
6.161.1.24 5 IO2CLR2	231
6.161.1.24 6 IO2CLR3	231
6.161.1.24 7 IO2CLRL	231
6.161.1.24 8 IO2CLRU	231
6.161.1.24 9 IO2DIR	231
6.161.1.25 0 IO2DIR0	231
6.161.1.25 1 IO2DIR1	231
6.161.1.25 2 IO2DIR2	231
6.161.1.25 3 IO2DIR3	231
6.161.1.25 4 IO2DIRL	231
6.161.1.25 5 IO2DIRU	231
6.161.1.25 6 IO2MASK	231
6.161.1.25 7 IO2MASK0	232
6.161.1.25 8 IO2MASK1	232
6.161.1.25 9 IO2MASK2	232
6.161.1.26 0 IO2MASK3	232
6.161.1.26 1 IO2MASKL	232
6.161.1.26 2 IO2MASKU	232
6.161.1.26 3 IO2PIN	232
6.161.1.26 4 IO2PIN0	232
6.161.1.26 5 IO2PIN1	232
6.161.1.26 6 IO2PIN2	232

6.161.1.267IO2PIN3	232
6.161.1.268IO2PINL	232
6.161.1.269IO2PINU	232
6.161.1.270IO2SET	232
6.161.1.271IO2SET0	232
6.161.1.272IO2SET1	232
6.161.1.273IO2SET2	232
6.161.1.274IO2SET3	232
6.161.1.275IO2SETL	232
6.161.1.276IO2SETU	232
6.161.1.277IO3CLR	232
6.161.1.278IO3CLR0	232
6.161.1.279IO3CLR1	232
6.161.1.280IO3CLR2	232
6.161.1.281IO3CLR3	233
6.161.1.282IO3CLRL	233
6.161.1.283IO3CLRU	233
6.161.1.284IO3DIR	233
6.161.1.285IO3DIR0	233
6.161.1.286IO3DIR1	233
6.161.1.287IO3DIR2	233
6.161.1.288IO3DIR3	233
6.161.1.289IO3DIRL	233
6.161.1.290IO3DIRU	233
6.161.1.291IO3MASK	233
6.161.1.292IO3MASK0	233
6.161.1.293IO3MASK1	233
6.161.1.294IO3MASK2	233
6.161.1.295IO3MASK3	233
6.161.1.296IO3MASKL	233
6.161.1.297IO3MASKU	233
6.161.1.298IO3PIN	233
6.161.1.299IO3PIN0	233
6.161.1.300IO3PIN1	233

6.161.1.30 I IO3PIN2	233
6.161.1.30 I IO3PIN3	233
6.161.1.30 I IO3PINL	233
6.161.1.30 I IO3PINU	233
6.161.1.30 I IO3SET	234
6.161.1.30 I IO3SET0	234
6.161.1.30 I IO3SET1	234
6.161.1.30 I IO3SET2	234
6.161.1.30 I IO3SET3	234
6.161.1.31 I IO3SETL	234
6.161.1.31 I IO3SETU	234
6.161.1.31 I IO4CLR	234
6.161.1.31 I IO4CLR0	234
6.161.1.31 I IO4CLR1	234
6.161.1.31 I IO4CLR2	234
6.161.1.31 I IO4CLR3	234
6.161.1.31 I IO4CLRL	234
6.161.1.31 I IO4CLRU	234
6.161.1.31 I IO4DIR	234
6.161.1.32 I IO4DIR0	234
6.161.1.32 I IO4DIR1	234
6.161.1.32 I IO4DIR2	234
6.161.1.32 I IO4DIR3	234
6.161.1.32 I IO4DIRL	234
6.161.1.32 I IO4DIRU	234
6.161.1.32 I IO4MASK	234
6.161.1.32 I IO4MASK0	234
6.161.1.32 I IO4MASK1	234
6.161.1.32 I IO4MASK2	235
6.161.1.32 I IO4MASK3	235
6.161.1.33 I IO4MASKL	235
6.161.1.33 I IO4MASKU	235
6.161.1.33 I IO4PIN	235
6.161.1.33 I IO4PIN0	235

CONTENTS

6.161.1.335IO4PIN1	235
6.161.1.336IO4PIN2	235
6.161.1.337IO4PIN3	235
6.161.1.338IO4PINL	235
6.161.1.339IO4PINU	235
6.161.1.340IO4SET	235
6.161.1.341IO4SET0	235
6.161.1.342IO4SET1	235
6.161.1.343IO4SET2	235
6.161.1.344IO4SET3	235
6.161.1.345IO4SETL	235
6.161.1.346IO4SETU	235
6.161.1.347IO_BASE_ADDR	235
6.161.1.348PDMA_CH0_CFG	235
6.161.1.349PDMA_CH0_CTRL	235
6.161.1.350PDMA_CH0_DEST	235
6.161.1.350PDMA_CH0_LLI	236
6.161.1.350PDMA_CH0_SRC	236
6.161.1.353PDMA_CH1_CFG	236
6.161.1.353PDMA_CH1_CTRL	236
6.161.1.355PDMA_CH1_DEST	236
6.161.1.356PDMA_CH1_LLI	236
6.161.1.356PDMA_CH1_SRC	236
6.161.1.358PDMA_CONFIG	236
6.161.1.359PDMA_ENABLED_CHNS	236
6.161.1.360PDMA_INT_ERR_CLR	236
6.161.1.360PDMA_INT_ERR_STAT	236
6.161.1.360PDMA_INT_STAT	236
6.161.1.363PDMA_INT_TCCLR	236
6.161.1.363PDMA_INT_TCSTAT	236
6.161.1.365PDMA_RAW_INT_ERR_STAT	236
6.161.1.366PDMA_RAW_INT_TCSTAT	236
6.161.1.367PDMA_SOFT_BREQ	237
6.161.1.368PDMA_SOFT_LBREQ	237

6.161.1.369PDMA_SOFT_LSREQ	237
6.161.1.370PDMA_SOFT_SREQ	237
6.161.1.371PDMA_SYNC	237
6.161.1.372PIO_BASE_ADDR	237
6.161.1.373BC_BULK_CUR_ED	237
6.161.1.374BC_BULK_HEAD_ED	237
6.161.1.375BC_CMD_STAT	237
6.161.1.376BC_CONTROL	237
6.161.1.377BC_CTRL_CUR_ED	237
6.161.1.378BC_CTRL_HEAD_ED	237
6.161.1.379BC_DONE_HEAD	237
6.161.1.380BC_FM_INTERVAL	237
6.161.1.381BC_FM_NUMBER	237
6.161.1.382BC_FM_REMAINING	237
6.161.1.383BC_HCCA	237
6.161.1.384BC_INT_DIS	238
6.161.1.385BC_INT_EN	238
6.161.1.386BC_INT_STAT	238
6.161.1.387BC_LS THRHL	238
6.161.1.388BC_PERIOD_CUR_ED	238
6.161.1.389BC_PERIOD_START	238
6.161.1.390BC_REVISION	238
6.161.1.391BC_RH_DESCA	238
6.161.1.392BC_RH_DESCB	238
6.161.1.393BC_RH_PORT_STAT1	238
6.161.1.394BC_RH_PORT_STAT2	238
6.161.1.395BC_RH_STAT	238
6.161.1.396ADR	238
6.161.1.397CONCLR	238
6.161.1.398CONSET	238
6.161.1.399DAT	238
6.161.1.400SCLH	238
6.161.1.401SCLL	238
6.161.1.402STAT	239

6.161.1.40 1 1ADR	239
6.161.1.40 1 1CONCLR	239
6.161.1.40 1 1CONSET	239
6.161.1.40 1 1DAT	239
6.161.1.40 1 1SCLH	239
6.161.1.40 1 1SCLL	239
6.161.1.40 1 1STAT	239
6.161.1.41 1 2ADR	239
6.161.1.41 1 2CONCLR	239
6.161.1.41 1 2CONSET	239
6.161.1.41 1 2DAT	239
6.161.1.41 1 2SCLH	239
6.161.1.41 1 2SCLL	239
6.161.1.41 1 2STAT	239
6.161.1.41 1 C0_BASE_ADDR	239
6.161.1.41 1 C1_BASE_ADDR	239
6.161.1.41 1 C2_BASE_ADDR	239
6.161.1.42 1 S_BASE_ADDR	239
6.161.1.42 1 S_DAI	239
6.161.1.42 1 S.DAO	239
6.161.1.42 1 S.DMA1	239
6.161.1.42 1 S.DMA2	239
6.161.1.42 1 S.IRQ	239
6.161.1.42 1 S.RX_FIFO	240
6.161.1.42 1 S.RXRATE	240
6.161.1.42 1 S.STATE	240
6.161.1.42 1 S.TX_FIFO	240
6.161.1.43 1 S.TXRATE	240
6.161.1.43 1 NTWAKE	240
6.161.1.43 1 O0.INT_CLR	240
6.161.1.43 1 O0.INT_EN_F	240
6.161.1.43 1 O0.INT_EN_R	240
6.161.1.43 1 O0.INT_STAT_F	240
6.161.1.43 1 O0.INT_STAT_R	240

6.161.1.43 02 _INT_CLR	240
6.161.1.43 02 _INT_EN_F	240
6.161.1.43 02 _INT_EN_R	240
6.161.1.44 02 _INT_STAT_F	240
6.161.1.44 02 _INT_STAT_R	240
6.161.1.44 02 _INT_STAT	240
6.161.1.44 02 _CLR0	240
6.161.1.44 02 _CLR1	240
6.161.1.44 02 _DIR0	241
6.161.1.44 02 _DIR1	241
6.161.1.44 02 _PIN0	241
6.161.1.44 02 _PIN1	241
6.161.1.44 02 _SET0	241
6.161.1.45 02 _SET1	241
6.161.1.45 02 _MAC_BASE_ADDR	241
6.161.1.45 02 _MAC_CLRT	241
6.161.1.45 02 _MAC_COMMAND	241
6.161.1.45 02 _MAC_FLOWCONTROLCNT	241
6.161.1.45 02 _MAC_FLOWCONTROLSTS	241
6.161.1.45 02 _MAC_HASHFILTERH	241
6.161.1.45 02 _MAC_HASHFILTERL	241
6.161.1.45 02 _MAC_INTCLEAR	241
6.161.1.45 02 _MAC_INTENABLE	241
6.161.1.46 02 _MAC_INTSET	241
6.161.1.46 02 _MAC_INTSTATUS	241
6.161.1.46 02 _MAC_IPGR	241
6.161.1.46 02 _MAC_IPGT	242
6.161.1.46 02 _MAC_MAC1	242
6.161.1.46 02 _MAC_MAC2	242
6.161.1.46 02 _MAC_MADR	242
6.161.1.46 02 _MAC_MAXF	242
6.161.1.46 02 _MAC_MCFG	242
6.161.1.46 02 _MAC_MCMD	242
6.161.1.47 02 _MAC_MIND	242

6.161.1.47MAC_MODULEID	242
6.161.1.47MAC_MRDD	242
6.161.1.47MAC_MWTD	242
6.161.1.47MAC_POWERDOWN	242
6.161.1.47MAC_RSV	242
6.161.1.47MAC_RXCONSUMEINDEX	242
6.161.1.47MAC_RXDESCRIPTOR	242
6.161.1.47MAC_RXDESCRIPTORMNUM	242
6.161.1.47MAC_RXFILTERCTRL	243
6.161.1.48MAC_RXFILTERWOLCLR	243
6.161.1.48MAC_RXFILTERWOLSTS	243
6.161.1.48MAC_RXPRODUCEINDEX	243
6.161.1.48MAC_RXSTATUS	243
6.161.1.48MAC_SA0	243
6.161.1.48MAC_SA1	243
6.161.1.48MAC_SA2	243
6.161.1.48MAC_STATUS	243
6.161.1.48MAC_SUPP	243
6.161.1.48MAC_TEST	243
6.161.1.49MAC_TSV0	243
6.161.1.49MAC_TSV1	243
6.161.1.49MAC_TXCONSUMEINDEX	243
6.161.1.49MAC_TXDESCRIPTOR	243
6.161.1.49MAC_TXDESCRIPTORMNUM	243
6.161.1.49MAC_TXPRODUCEINDEX	244
6.161.1.49MAC_TXSTATUS	244
6.161.1.49MAMCR	244
6.161.1.49MAMTIM	244
6.161.1.49MAXPACKET_SIZE	244
6.161.1.50MCI_ARGUMENT	244
6.161.1.50MCI_BASE_ADDR	244
6.161.1.50MCI_CLEAR	244
6.161.1.50MCI_CLOCK	244
6.161.1.50MCI_COMMAND	244

6.161.1.50 <u>MCI_DATA_CNT</u>	244
6.161.1.50 <u>MCI_DATA_CTRL</u>	244
6.161.1.50 <u>MCI_DATA_LEN</u>	244
6.161.1.50 <u>MCI_DATA_TMR</u>	244
6.161.1.50 <u>MCI_FIFO</u>	244
6.161.1.51 <u>MCI_FIFO_CNT</u>	244
6.161.1.51 <u>MCI_MASK0</u>	244
6.161.1.51 <u>MCI_MASK1</u>	244
6.161.1.51 <u>MCI_POWER</u>	244
6.161.1.51 <u>MCI_RESP0</u>	245
6.161.1.51 <u>MCI_RESP1</u>	245
6.161.1.51 <u>MCI_RESP2</u>	245
6.161.1.51 <u>MCI_RESP3</u>	245
6.161.1.51 <u>MCI_RESP_CMD</u>	245
6.161.1.51 <u>MCI_STATUS</u>	245
6.161.1.52 <u>MEMMAP</u>	245
6.161.1.52 <u>REQ_INT_CLR</u>	245
6.161.1.52 <u>REQ_INT_SET</u>	245
6.161.1.52 <u>REQ_INT_STAT</u>	245
6.161.1.52 <u>TG_CLK_CTRL</u>	245
6.161.1.52 <u>TG_CLK_STAT</u>	245
6.161.1.52 <u>TG_I2C_CLKHI</u>	245
6.161.1.52 <u>TG_I2C_CLKLO</u>	245
6.161.1.52 <u>TG_I2C_CTL</u>	245
6.161.1.52 <u>TG_I2C_RX</u>	245
6.161.1.53 <u>TG_I2C_STS</u>	245
6.161.1.53 <u>TG_I2C_TX</u>	245
6.161.1.53 <u>TG_INT_CLR</u>	246
6.161.1.53 <u>TG_INT_EN</u>	246
6.161.1.53 <u>TG_INT_SET</u>	246
6.161.1.53 <u>TG_INT_STAT</u>	246
6.161.1.53 <u>TG_STAT_CTRL</u>	246
6.161.1.53 <u>TG_TIMER</u>	246
6.161.1.53 <u>BARTCFG</u>	246

6.161.1.539ARTCFG_BASE_ADDR	246
6.161.1.540CLKSEL0	246
6.161.1.541CLKSEL1	246
6.161.1.542CON	246
6.161.1.543CONP	246
6.161.1.544INMODE0	246
6.161.1.545INMODE1	246
6.161.1.546INMODE2	246
6.161.1.547INMODE3	246
6.161.1.548INMODE4	246
6.161.1.549INMODE5	246
6.161.1.550INMODE6	247
6.161.1.551INMODE7	247
6.161.1.552INMODE8	247
6.161.1.553INMODE9	247
6.161.1.554INSEL0	247
6.161.1.555INSEL1	247
6.161.1.556INSEL10	247
6.161.1.557INSEL2	247
6.161.1.558INSEL3	247
6.161.1.559INSEL4	247
6.161.1.560INSEL5	247
6.161.1.561INSEL6	247
6.161.1.562INSEL7	247
6.161.1.563INSEL8	247
6.161.1.564INSEL9	247
6.161.1.565INSEL_BASE_ADDR	247
6.161.1.566LLCFG	247
6.161.1.567LLCON	247
6.161.1.568LLFEED	247
6.161.1.569LLSTAT	247
6.161.1.570WM0_BASE_ADDR	247
6.161.1.571WM0CCR	247
6.161.1.572WM0CR0	248

6.161.1.57 ^B WM0CR1	248
6.161.1.57 ^D WM0CR2	248
6.161.1.57 ^E WM0CR3	248
6.161.1.57 ^F WM0CTCR	248
6.161.1.57 ^F WM0EMR	248
6.161.1.57 ^G WM0IR	248
6.161.1.57 ^H WM0LER	248
6.161.1.58 ^B WM0MCR	248
6.161.1.58 ^B WM0MR0	248
6.161.1.58 ^C WM0MR1	248
6.161.1.58 ^B WM0MR2	248
6.161.1.58 ^C WM0MR3	248
6.161.1.58 ^B WM0MR4	248
6.161.1.58 ^B WM0MR5	248
6.161.1.58 ^F WM0MR6	248
6.161.1.58 ^B WM0PC	248
6.161.1.58 ^H WM0PCR	248
6.161.1.59 ^B WM0PR	248
6.161.1.59 ^F WM0TC	248
6.161.1.59 ^C WM0TCR	248
6.161.1.59 ^B WM1_BASE_ADDR	248
6.161.1.59 ^F WM1CCR	248
6.161.1.59 ^B WM1CR0	249
6.161.1.59 ^B WM1CR1	249
6.161.1.59 ^F WM1CR2	249
6.161.1.59 ^B WM1CR3	249
6.161.1.59 ^H WM1CTCR	249
6.161.1.60 ^B WM1EMR	249
6.161.1.60 ^F WM1IR	249
6.161.1.60 ^C WM1LER	249
6.161.1.60 ^B WM1MCR	249
6.161.1.60 ^F WM1MR0	249
6.161.1.60 ^B WM1MR1	249
6.161.1.60 ^B WM1MR2	249

6.161.1.60 ⁷ WM1MR3	249
6.161.1.60 ⁸ WM1MR4	249
6.161.1.60 ⁹ WM1MR5	249
6.161.1.61 ⁰ WM1MR6	249
6.161.1.61 ¹ WM1PC	249
6.161.1.61 ² WM1PCR	249
6.161.1.61 ³ WM1PR	249
6.161.1.61 ⁴ WM1TC	249
6.161.1.61 ⁵ WM1TCR	249
6.161.1.61 ⁶ REALIZE_EP	249
6.161.1.61 ⁷ SIR	249
6.161.1.61 ⁸ TC_ALDOM	250
6.161.1.61 ⁹ TC_ALDOW	250
6.161.1.62 ⁰ TC_ALDOY	250
6.161.1.62 ¹ TC_ALHOUR	250
6.161.1.62 ² TC_ALMIN	250
6.161.1.62 ³ TC_ALMON	250
6.161.1.62 ⁴ TC_ALSEC	250
6.161.1.62 ⁵ TC_ALYEAR	250
6.161.1.62 ⁶ TC_AMR	250
6.161.1.62 ⁷ TC_BASE_ADDR	250
6.161.1.62 ⁸ TC_CCR	250
6.161.1.62 ⁹ TC_CIIR	250
6.161.1.63 ⁰ TC_CISS	250
6.161.1.63 ¹ TC_CTC	250
6.161.1.63 ² TC_CTIME0	250
6.161.1.63 ³ TC_CTIME1	250
6.161.1.63 ⁴ TC_CTIME2	250
6.161.1.63 ⁵ TC_DOM	250
6.161.1.63 ⁶ TC_DOW	250
6.161.1.63 ⁷ TC_DOY	250
6.161.1.63 ⁸ TC_HOUR	250
6.161.1.63 ⁹ TC_ILR	250
6.161.1.64 ⁰ TC_MIN	250

6.161.1.64RTC_MONTH	251
6.161.1.64RTC_PREFRAC	251
6.161.1.64RTC_PREINT	251
6.161.1.64RTC_SEC	251
6.161.1.64RTC_YEAR	251
6.161.1.64BX_DATA	251
6.161.1.64RX_PLENGTH	251
6.161.1.64SPCCR	251
6.161.1.64SPPCR	251
6.161.1.65SPDR	251
6.161.1.65SPINT	251
6.161.1.65SPSR	251
6.161.1.65SCB_BASE_ADDR	251
6.161.1.65SCS	251
6.161.1.65SPI0_BASE_ADDR	251
6.161.1.65SP0_BASE_ADDR	251
6.161.1.65SP0CPSR	251
6.161.1.65SP0CR0	251
6.161.1.65SP0CR1	251
6.161.1.66SP0DMACR	251
6.161.1.66SP0DR	251
6.161.1.66SP0ICR	251
6.161.1.66SP0IMSC	252
6.161.1.66SP0MIS	252
6.161.1.66SP0RIS	252
6.161.1.66SP0SR	252
6.161.1.66SP1_BASE_ADDR	252
6.161.1.66SP1CPSR	252
6.161.1.66SP1CR0	252
6.161.1.67SP1CR1	252
6.161.1.67SP1DMACR	252
6.161.1.67SP1DR	252
6.161.1.67SP1ICR	252
6.161.1.67SP1IMSC	252

6.161.1.67\$SP1MIS	252
6.161.1.67\$SP1RIS	252
6.161.1.67\$SP1SR	252
6.161.1.67\$STATIC_MEM0_BASE	252
6.161.1.67\$STATIC_MEM1_BASE	252
6.161.1.68\$STATIC_MEM2_BASE	252
6.161.1.68\$STATIC_MEM3_BASE	252
6.161.1.68\$SYS_ERR_INT_CLR	252
6.161.1.68\$SYS_ERR_INT_SET	252
6.161.1.68\$SYS_ERR_INT_STAT	252
6.161.1.68\$CCR	253
6.161.1.68\$CR0	253
6.161.1.68\$CR1	253
6.161.1.68\$CR2	253
6.161.1.68\$CR3	253
6.161.1.69\$CTCR	253
6.161.1.69\$EMR	253
6.161.1.69\$IR	253
6.161.1.69\$MCR	253
6.161.1.69\$MR0	253
6.161.1.69\$MR1	253
6.161.1.69\$MR2	253
6.161.1.69\$MR3	253
6.161.1.69\$PC	253
6.161.1.69\$PR	253
6.161.1.70\$TC	253
6.161.1.70\$TCR	253
6.161.1.70\$CCR	253
6.161.1.70\$B1CR0	253
6.161.1.70\$B1CR1	253
6.161.1.70\$B1CR2	253
6.161.1.70\$B1CR3	253
6.161.1.70\$V1CTCR	253
6.161.1.70\$B1EMR	253

6.161.1.70 ⁰¹ IR	254
6.161.1.71 ⁰¹ MCR	254
6.161.1.71 ¹¹ MR0	254
6.161.1.71 ²¹ MR1	254
6.161.1.71 ³¹ MR2	254
6.161.1.71 ⁴¹ MR3	254
6.161.1.71 ⁵¹ PC	254
6.161.1.71 ⁶¹ PR	254
6.161.1.71 ⁷¹ TC	254
6.161.1.71 ⁸¹ TCR	254
6.161.1.71 ⁹² CCR	254
6.161.1.72 ⁰² CR0	254
6.161.1.72 ¹² CR1	254
6.161.1.72 ²² CR2	254
6.161.1.72 ³² CR3	254
6.161.1.72 ⁴² CTCR	254
6.161.1.72 ⁵² EMR	254
6.161.1.72 ⁶² IR	254
6.161.1.72 ⁷² MCR	254
6.161.1.72 ⁸² MR0	254
6.161.1.72 ⁹² MR1	254
6.161.1.73 ⁰² MR2	254
6.161.1.73 ¹² MR3	254
6.161.1.73 ²² PC	254
6.161.1.73 ³² PR	255
6.161.1.73 ⁴² TC	255
6.161.1.73 ⁵² TCR	255
6.161.1.73 ⁶³ CCR	255
6.161.1.73 ⁷³ CR0	255
6.161.1.73 ⁸³ CR1	255
6.161.1.73 ⁹³ CR2	255
6.161.1.74 ⁰³ CR3	255
6.161.1.74 ¹³ CTCR	255
6.161.1.74 ²³ EMR	255

6.161.1.74B3IR	255
6.161.1.74T3MCR	255
6.161.1.74B3MR0	255
6.161.1.74B3MR1	255
6.161.1.74T3MR2	255
6.161.1.74B3MR3	255
6.161.1.74B3PC	255
6.161.1.75T3PR	255
6.161.1.75T3TC	255
6.161.1.75T3TCR	255
6.161.1.75BMR0_BASE_ADDR	255
6.161.1.75BMR1_BASE_ADDR	255
6.161.1.75BMR2_BASE_ADDR	255
6.161.1.75BMR3_BASE_ADDR	255
6.161.1.75TX_DATA	256
6.161.1.75BX_PLENGTH	256
6.161.1.7590ACR	256
6.161.1.7600DLL	256
6.161.1.7610DLM	256
6.161.1.7620FCR	256
6.161.1.7630FDR	256
6.161.1.7640ICR	256
6.161.1.7650IER	256
6.161.1.7660IIR	256
6.161.1.7670LCR	256
6.161.1.7680LSR	256
6.161.1.7690RBR	256
6.161.1.7700SCR	256
6.161.1.7710TER	256
6.161.1.7720THR	256
6.161.1.7731ACR	256
6.161.1.7741DLL	256
6.161.1.7751DLM	256
6.161.1.7761FCR	256

6.161.1.7701FDR	256
6.161.1.7701IER	256
6.161.1.7701IIR	256
6.161.1.7801LCR	257
6.161.1.7801LSR	257
6.161.1.7801MCR	257
6.161.1.7801MSR	257
6.161.1.7801RBR	257
6.161.1.7801SCR	257
6.161.1.7801TER	257
6.161.1.7801THR	257
6.161.1.7802ACR	257
6.161.1.7802DLL	257
6.161.1.7902DLM	257
6.161.1.7902FCR	257
6.161.1.7902FDR	257
6.161.1.7902ICR	257
6.161.1.7902IER	257
6.161.1.7902IIR	257
6.161.1.7902LCR	257
6.161.1.7902LSR	257
6.161.1.7902RBR	257
6.161.1.7902SCR	257
6.161.1.8002TER	257
6.161.1.8002THR	257
6.161.1.8003ACR	257
6.161.1.8003DLL	257
6.161.1.8003DLM	258
6.161.1.8003FCR	258
6.161.1.8003FDR	258
6.161.1.8003ICR	258
6.161.1.8003IER	258
6.161.1.8003IIR	258
6.161.1.8003LCR	258

6.161.1.81 <u>3</u> LSR	258
6.161.1.81 <u>3</u> RBR	258
6.161.1.81 <u>3</u> SCR	258
6.161.1.81 <u>3</u> TER	258
6.161.1.81 <u>3</u> THR	258
6.161.1.81 <u>ART0</u> _BASE_ADDR	258
6.161.1.81 <u>ART1</u> _BASE_ADDR	258
6.161.1.81 <u>ART2</u> _BASE_ADDR	258
6.161.1.81 <u>ART3</u> _BASE_ADDR	258
6.161.1.82 <u>DCA</u> _HEAD	258
6.161.1.82 <u>USB</u> _BASE_ADDR	258
6.161.1.82 <u>SB</u> _CTRL	258
6.161.1.82 <u>SB</u> _INT_BASE_ADDR	258
6.161.1.82 <u>SB</u> _INT_STAT	258
6.161.1.82 <u>SBCLKCFG</u>	258
6.161.1.82 <u>SBClkCtrl</u>	259
6.161.1.82 <u>SBClkSt</u>	259
6.161.1.82 <u>SBHC</u> _BASE_ADDR	259
6.161.1.82 <u>SBOTG</u> _BASE_ADDR	259
6.161.1.82 <u>SBOTG_CLK</u> _BASE_ADDR	259
6.161.1.82 <u>SBOTG_I2C</u> _BASE_ADDR	259
6.161.1.83 <u>SBPortSel</u>	259
6.161.1.83 <u>IIC</u> _BASE_ADDR	259
6.161.1.83 <u>IICFIQStatus</u>	259
6.161.1.83 <u>IICIntEnable</u>	259
6.161.1.83 <u>IICIntEnClr</u>	259
6.161.1.83 <u>IICIntSelect</u>	259
6.161.1.83 <u>IICIRQStatus</u>	259
6.161.1.83 <u>IICProtection</u>	259
6.161.1.84 <u>IICRawIntr</u>	259
6.161.1.84 <u>IICSoftInt</u>	259
6.161.1.84 <u>IICSoftIntClr</u>	259
6.161.1.84 <u>ICSWPrioMask</u>	259
6.161.1.84 <u>IICVectAddr</u>	259

6.161.1.84 ⁸ ICVectAddr0	259
6.161.1.84 ⁸ ICVectAddr1	259
6.161.1.84 ⁷ ICVectAddr10	260
6.161.1.84 ⁸ ICVectAddr11	260
6.161.1.84 ⁹ ICVectAddr12	260
6.161.1.85 ⁰ ICVectAddr13	260
6.161.1.85 ¹ ICVectAddr14	260
6.161.1.85 ² ICVectAddr15	260
6.161.1.85 ³ ICVectAddr16	260
6.161.1.85 ⁴ ICVectAddr17	260
6.161.1.85 ⁵ ICVectAddr18	260
6.161.1.85 ⁶ ICVectAddr19	260
6.161.1.85 ⁷ ICVectAddr2	260
6.161.1.85 ⁸ ICVectAddr20	260
6.161.1.85 ⁹ ICVectAddr21	260
6.161.1.86 ⁰ ICVectAddr22	260
6.161.1.86 ¹ ICVectAddr23	260
6.161.1.86 ² ICVectAddr24	260
6.161.1.86 ³ ICVectAddr25	261
6.161.1.86 ⁴ ICVectAddr26	261
6.161.1.86 ⁵ ICVectAddr27	261
6.161.1.86 ⁶ ICVectAddr28	261
6.161.1.86 ⁷ ICVectAddr29	261
6.161.1.86 ⁸ ICVectAddr3	261
6.161.1.86 ⁹ ICVectAddr30	261
6.161.1.87 ⁰ ICVectAddr31	261
6.161.1.87 ¹ ICVectAddr4	261
6.161.1.87 ² ICVectAddr5	261
6.161.1.87 ³ ICVectAddr6	261
6.161.1.87 ⁴ ICVectAddr7	261
6.161.1.87 ⁵ ICVectAddr8	261
6.161.1.87 ⁶ ICVectAddr9	261
6.161.1.87 ⁷ ICVectCntrl0	261
6.161.1.87 ⁸ ICVectCntrl1	261

6.161.1.87VICVectCtrl10	261
6.161.1.88VICVectCtrl11	262
6.161.1.88VICVectCtrl12	262
6.161.1.88VICVectCtrl13	262
6.161.1.88VICVectCtrl14	262
6.161.1.88VICVectCtrl15	262
6.161.1.88VICVectCtrl16	262
6.161.1.88VICVectCtrl17	262
6.161.1.88VICVectCtrl18	262
6.161.1.88VICVectCtrl19	262
6.161.1.88VICVectCtrl20	262
6.161.1.89VICVectCtrl21	262
6.161.1.89VICVectCtrl22	262
6.161.1.89VICVectCtrl23	262
6.161.1.89VICVectCtrl24	262
6.161.1.89VICVectCtrl25	262
6.161.1.89VICVectCtrl26	263
6.161.1.89VICVectCtrl27	263
6.161.1.89VICVectCtrl28	263
6.161.1.89VICVectCtrl29	263
6.161.1.90VICVectCtrl3	263
6.161.1.90VICVectCtrl30	263
6.161.1.90VICVectCtrl31	263
6.161.1.90VICVectCtrl4	263
6.161.1.90VICVectCtrl5	263
6.161.1.90VICVectCtrl6	263
6.161.1.90VICVectCtrl7	263
6.161.1.90VICVectCtrl8	263
6.161.1.90VICVectCtrl9	263
6.161.1.90PBDIV	263
6.161.1.91WDCLKSEL	263
6.161.1.91WDFEED	263
6.161.1.91WDG_BASE_ADDR	263

6.161.1.91WDMOD	263
6.161.1.91WDTC	263
6.161.1.91WDTV	263
6.162include/canvas.h File Reference	263
6.162.1 Define Documentation	265
6.162.1.1 Canvas	265
6.162.1.2 CANVAS_STYLE_ALIGN_HMASK	266
6.162.1.3 CANVAS_STYLE_ALIGN_MASK	267
6.162.1.4 CANVAS_STYLE_ALIGN_VMASK	267
6.162.1.5 CANVAS_STYLE_APP_DRAWN	267
6.162.1.6 CANVAS_STYLE_FILL	267
6.162.1.7 CANVAS_STYLE_IMG	267
6.162.1.8 CANVAS_STYLE_OUTLINE	267
6.162.1.9 CANVAS_STYLE_TEXT	267
6.162.1.10CANVAS_STYLE_TEXT_BOTTOM	267
6.162.1.11CANVAS_STYLE_TEXT_HCENTER	268
6.162.1.12CANVAS_STYLE_TEXT_LEFT	268
6.162.1.13CANVAS_STYLE_TEXT_OPAQUE	268
6.162.1.14CANVAS_STYLE_TEXT_RIGHT	268
6.162.1.15CANVAS_STYLE_TEXT_TOP	268
6.162.1.16CANVAS_STYLE_TEXT_VCENTER	268
6.162.1.17CanvasAppDrawnOff	268
6.162.1.18CanvasAppDrawnOn	269
6.162.1.19CanvasCallbackSet	269
6.162.1.20CanvasFillColorSet	270
6.162.1.21CanvasFillOff	270
6.162.1.22CanvasFillOn	271
6.162.1.23CanvasFontSet	271
6.162.1.24CanvasImageOff	272
6.162.1.25CanvasImageOn	272
6.162.1.26CanvasImageSet	273
6.162.1.27CanvasOutlineColorSet	273
6.162.1.28CanvasOutlineOff	274
6.162.1.29CanvasOutlineOn	274

6.162.1.30CanvasStruct	275
6.162.1.31CanvasTextAlignment	277
6.162.1.32CanvasTextColorSet	277
6.162.1.33CanvasTextOff	278
6.162.1.34CanvasTextOn	278
6.162.1.35CanvasTextOpaqueOff	279
6.162.1.36CanvasTextOpaqueOn	279
6.162.1.37CanvasTextSet	280
6.162.2 Function Documentation	280
6.162.2.1 CanvasInit	280
6.162.2.2 CanvasMsgProc	281
6.163include/checkbox.h File Reference	281
6.163.1 Define Documentation	282
6.163.1.1 CB_STYLE_FILL	282
6.163.1.2 CB_STYLE_IMG	282
6.163.1.3 CB_STYLE_OUTLINE	282
6.163.1.4 CB_STYLE_SELECTED	283
6.163.1.5 CB_STYLE_TEXT	283
6.163.1.6 CB_STYLE_TEXT_OPAQUE	283
6.163.1.7 CheckBox	283
6.163.1.8 CheckBoxBoxSizeSet	284
6.163.1.9 CheckBoxCallbackSet	285
6.163.1.10CheckBoxFillColorSet	285
6.163.1.11CheckBoxFillOff	286
6.163.1.12CheckBoxFillOn	286
6.163.1.13CheckBoxFontSet	286
6.163.1.14CheckBoxImageOff	287
6.163.1.15CheckBoxImageOn	287
6.163.1.16CheckBoxImageSet	288
6.163.1.17CheckBoxOutlineColorSet	288
6.163.1.18CheckBoxOutlineOff	289
6.163.1.19CheckBoxOutlineOn	289
6.163.1.20CheckBoxStruct	290
6.163.1.21CheckBoxTextColorSet	292

6.163.1.22CheckBoxTextOff	292
6.163.1.23CheckBoxTextOn	293
6.163.1.24CheckBoxTextOpaqueOff	293
6.163.1.25CheckBoxTextOpaqueOn	294
6.163.1.26CheckBoxTextSet	294
6.163.2 Function Documentation	295
6.163.2.1 CheckBoxInit	295
6.163.2.2 CheckBoxMsgProc	295
6.164 include/container.h File Reference	296
6.164.1 Define Documentation	297
6.164.1.1 Container	297
6.164.1.2 ContainerFillColorSet	298
6.164.1.3 ContainerFillOff	298
6.164.1.4 ContainerFillOn	299
6.164.1.5 ContainerFontSet	299
6.164.1.6 ContainerOutlineColorSet	300
6.164.1.7 ContainerOutlineOff	300
6.164.1.8 ContainerOutlineOn	301
6.164.1.9 ContainerStruct	301
6.164.1.10ContainerTextCenterOff	303
6.164.1.11ContainerTextCenterOn	303
6.164.1.12ContainerTextColorSet	304
6.164.1.13ContainerTextOff	304
6.164.1.14ContainerTextOn	305
6.164.1.15ContainerTextOpaqueOff	305
6.164.1.16ContainerTextOpaqueOn	306
6.164.1.17ContainerTextSet	306
6.164.1.18CTR_STYLE_FILL	307
6.164.1.19CTR_STYLE_OUTLINE	307
6.164.1.20CTR_STYLE_TEXT	307
6.164.1.21CTR_STYLE_TEXT_CENTER	307
6.164.1.22CTR_STYLE_TEXT_OPAQUE	307
6.164.2 Function Documentation	307
6.164.2.1 ContainerInit	307

6.164.2.2 ContainerMsgProc	308
6.165include/graphic.h File Reference	308
6.165.1 Function Documentation	308
6.165.1.1 GLCD_Circle	308
6.165.1.2 GLCD_Line	308
6.165.1.3 GLCD_Rectangle	308
6.166include/grlib.h File Reference	309
6.166.1 Define Documentation	317
6.166.1.1 ClrAliceBlue	317
6.166.1.2 ClrAntiqueWhite	317
6.166.1.3 ClrAqua	317
6.166.1.4 ClrAquamarine	318
6.166.1.5 ClrAzure	318
6.166.1.6 ClrBeige	318
6.166.1.7 ClrBisque	318
6.166.1.8 ClrBlack	318
6.166.1.9 ClrBlanchedAlmond	318
6.166.1.10ClrBlue	318
6.166.1.11ClrBlueMask	318
6.166.1.12ClrBlueShift	318
6.166.1.13ClrBlueViolet	318
6.166.1.14ClrBrown	318
6.166.1.15ClrBurlyWood	318
6.166.1.16ClrCadetBlue	318
6.166.1.17ClrChartreuse	318
6.166.1.18ClrChocolate	318
6.166.1.19ClrCoral	318
6.166.1.20ClrCornflowerBlue	318
6.166.1.21ClrCornsilk	318
6.166.1.22ClrCrimson	318
6.166.1.23ClrCyan	318
6.166.1.24ClrDarkBlue	318
6.166.1.25ClrDarkCyan	318
6.166.1.26ClrDarkGoldenrod	318

6.166.1.27ClrDarkGray	318
6.166.1.28ClrDarkGreen	319
6.166.1.29ClrDarkKhaki	319
6.166.1.30ClrDarkMagenta	319
6.166.1.31ClrDarkOliveGreen	319
6.166.1.32ClrDarkOrange	319
6.166.1.33ClrDarkOrchid	319
6.166.1.34ClrDarkRed	319
6.166.1.35ClrDarkSalmon	319
6.166.1.36ClrDarkSeaGreen	319
6.166.1.37ClrDarkSlateBlue	319
6.166.1.38ClrDarkSlateGray	319
6.166.1.39ClrDarkTurquoise	319
6.166.1.40ClrDarkViolet	319
6.166.1.41ClrDeepPink	319
6.166.1.42ClrDeepSkyBlue	319
6.166.1.43ClrDimGray	319
6.166.1.44ClrDodgerBlue	319
6.166.1.45ClrFireBrick	319
6.166.1.46ClrFloralWhite	319
6.166.1.47ClrForestGreen	319
6.166.1.48ClrFuchsia	319
6.166.1.49ClrGainsboro	319
6.166.1.50ClrGhostWhite	319
6.166.1.51ClrGold	319
6.166.1.52ClrGoldenrod	320
6.166.1.53ClrGray	320
6.166.1.54ClrGreen	320
6.166.1.55ClrGreenMask	320
6.166.1.56ClrGreenShift	320
6.166.1.57ClrGreenYellow	320
6.166.1.58ClrHoneydew	320
6.166.1.59ClrHotPink	320
6.166.1.60ClrIndianRed	320

6.166.1.61ClrIndigo	320
6.166.1.62ClrIvory	320
6.166.1.63ClrKhaki	320
6.166.1.64ClrLavender	320
6.166.1.65ClrLavenderBlush	320
6.166.1.66ClrLawnGreen	320
6.166.1.67ClrLemonChiffon	320
6.166.1.68ClrLightBlue	320
6.166.1.69ClrLightCoral	320
6.166.1.70ClrLightCyan	320
6.166.1.71ClrLightGoldenrodYellow	320
6.166.1.72ClrLightGreen	320
6.166.1.73ClrLightGrey	320
6.166.1.74ClrLightPink	320
6.166.1.75ClrLightSalmon	320
6.166.1.76ClrLightSeaGreen	321
6.166.1.77ClrLightSkyBlue	321
6.166.1.78ClrLightSlateGray	321
6.166.1.79ClrLightSteelBlue	321
6.166.1.80ClrLightYellow	321
6.166.1.81ClrLime	321
6.166.1.82ClrLimeGreen	321
6.166.1.83ClrLinen	321
6.166.1.84ClrMagenta	321
6.166.1.85ClrMaroon	321
6.166.1.86ClrMediumAquamarine	321
6.166.1.87ClrMediumBlue	321
6.166.1.88ClrMediumOrchid	321
6.166.1.89ClrMediumPurple	321
6.166.1.90ClrMediumSeaGreen	321
6.166.1.91ClrMediumSlateBlue	321
6.166.1.92ClrMediumSpringGreen	321
6.166.1.93ClrMediumTurquoise	321
6.166.1.94ClrMediumVioletRed	321

6.166.1.95ClrMidnightBlue	321
6.166.1.96ClrMintCream	321
6.166.1.97ClrMistyRose	321
6.166.1.98ClrMoccasin	321
6.166.1.99ClrNavajoWhite	321
6.166.1.100IrNavy	322
6.166.1.103IrOldLace	322
6.166.1.102IrOlive	322
6.166.1.103IrOliveDrab	322
6.166.1.104IrOrange	322
6.166.1.105IrOrangeRed	322
6.166.1.106IrOrchid	322
6.166.1.107IrPaleGoldenrod	322
6.166.1.108IrPaleGreen	322
6.166.1.109IrPaleTurquoise	322
6.166.1.110IrPaleVioletRed	322
6.166.1.110IrPapayaWhip	322
6.166.1.112IrPeachPuff	322
6.166.1.113IrPeru	322
6.166.1.114IrPink	322
6.166.1.115IrPlum	322
6.166.1.116IrPowderBlue	322
6.166.1.117IrPurple	322
6.166.1.118IrRed	322
6.166.1.119IrRedMask	322
6.166.1.120IrRedShift	322
6.166.1.121IrRosyBrown	322
6.166.1.122IrRoyalBlue	322
6.166.1.123IrSaddleBrown	322
6.166.1.124IrSalmon	323
6.166.1.125IrSandyBrown	323
6.166.1.126IrSeaGreen	323
6.166.1.127IrSeashell	323
6.166.1.128IrSienna	323

6.166.1.129lrSilver	323
6.166.1.130lrSkyBlue	323
6.166.1.131lrSlateBlue	323
6.166.1.132lrSlateGray	323
6.166.1.133lrSnow	323
6.166.1.134lrSpringGreen	323
6.166.1.135lrSteelBlue	323
6.166.1.136lrTan	323
6.166.1.137lrTeal	323
6.166.1.138lrThistle	323
6.166.1.139lrTomato	323
6.166.1.140lrTurquoise	323
6.166.1.141lrViolet	323
6.166.1.142lrWheat	323
6.166.1.143lrWhite	323
6.166.1.144lrWhiteSmoke	323
6.166.1.145lrYellow	323
6.166.1.146lrYellowGreen	323
6.166.1.147DpyColorTranslate	324
6.166.1.148DpyFlush	324
6.166.1.149DpyHeightGet	324
6.166.1.150DpyLineDrawH	325
6.166.1.151DpyLineDrawV	325
6.166.1.152DpyPixelDraw	326
6.166.1.153DpyPixelDrawMultiple	326
6.166.1.154DpyRectFill	327
6.166.1.155DpyWidthGet	328
6.166.1.156FONT_FMT_PIXEL_RLE	328
6.166.1.157FONT_FMT_UNCOMPRESSED	328
6.166.1.158rContextBackgroundSet	328
6.166.1.159rContextBackgroundSetTranslated	329
6.166.1.160rContextDpyHeightGet	329
6.166.1.161rContextDpyWidthGet	329
6.166.1.162rContextFontSet	330

6.166.1.163rContextForegroundSet	330
6.166.1.164rContextForegroundSetTranslated	331
6.166.1.165rFlush	331
6.166.1.166rFontBaselineGet	332
6.166.1.167rFontHeightGet	332
6.166.1.168rFontMaxWidthGet	332
6.166.1.169rImageColorsGet	333
6.166.1.170rImageHeightGet	333
6.166.1.171rImageWidthGet	333
6.166.1.172rLangDe	334
6.166.1.173rLangEnAUS	334
6.166.1.174rLangEnCA	334
6.166.1.175rLangEnNZ	334
6.166.1.176rLangEnUK	334
6.166.1.177rLangEnUS	334
6.166.1.178rLangEsMX	334
6.166.1.179rLangEsSP	334
6.166.1.180rLangFr	334
6.166.1.181rLangHi	334
6.166.1.182rLangIt	334
6.166.1.183rLangJp	334
6.166.1.184rLangKo	334
6.166.1.185rLangSwKE	334
6.166.1.186rLangUrIN	334
6.166.1.187rLangUrPK	334
6.166.1.188rLangZhPRC	334
6.166.1.189rLangZhTW	334
6.166.1.190rOffScreen1BPPSize	334
6.166.1.191rOffScreen4BPPSize	335
6.166.1.192rOffScreen8BPPSize	335
6.166.1.193rPixelDraw	335
6.166.1.194rRectContainsPoint	336
6.166.1.195rStringBaselineGet	336
6.166.1.196rStringDrawCentered	337

6.166.1.19GrStringHeightGet	338
6.166.1.19GrStringMaxWidthGet	338
6.166.1.19IMAGE_FMT_16BPP_UNCOMP	338
6.166.1.20IMAGE_FMT_1BPP_COMP	338
6.166.1.20IMAGE_FMT_1BPP_UNCOMP	339
6.166.1.20IMAGE_FMT_4BPP_COMP	339
6.166.1.20IMAGE_FMT_4BPP_UNCOMP	339
6.166.1.20IMAGE_FMT_8BPP_COMP	339
6.166.1.20IMAGE_FMT_8BPP_UNCOMP	339
6.166.2 Function Documentation	339
6.166.2.1 GrCircleDraw	339
6.166.2.2 GrCircleFill	340
6.166.2.3 GrContextClipRegionSet	340
6.166.2.4 GrContextInit	340
6.166.2.5 GrImageDraw	341
6.166.2.6 GrLineDraw	341
6.166.2.7 GrLineDrawH	342
6.166.2.8 GrLineDrawV	342
6.166.2.9 GrRectDraw	343
6.166.2.10GrRectFill	343
6.166.2.11GrRectIntersectGet	343
6.166.2.12GrRectOverlapCheck	344
6.166.2.13GrStringDraw	344
6.166.2.14GrStringGet	345
6.166.2.15GrStringLanguageSet	345
6.166.2.16GrStringTableSet	345
6.166.2.17GrStringWidthGet	346
6.166.3 Variable Documentation	346
6.166.3.1 g_sFontCm12	346
6.166.3.2 g_sFontCm12b	346
6.166.3.3 g_sFontCm12i	346
6.166.3.4 g_sFontCm14	346
6.166.3.5 g_sFontCm14b	346
6.166.3.6 g_sFontCm14i	346

6.166.3.7 g_sFontCm16	347
6.166.3.8 g_sFontCm16b	347
6.166.3.9 g_sFontCm16i	347
6.166.3.10g_sFontCm18	347
6.166.3.11g_sFontCm18b	347
6.166.3.12g_sFontCm18i	347
6.166.3.13g_sFontCm20	347
6.166.3.14g_sFontCm20b	347
6.166.3.15g_sFontCm20i	347
6.166.3.16g_sFontCm22	347
6.166.3.17g_sFontCm22b	347
6.166.3.18g_sFontCm22i	347
6.166.3.19g_sFontCm24	347
6.166.3.20g_sFontCm24b	347
6.166.3.21g_sFontCm24i	347
6.166.3.22g_sFontCm26	347
6.166.3.23g_sFontCm26b	347
6.166.3.24g_sFontCm26i	347
6.166.3.25g_sFontCm28	347
6.166.3.26g_sFontCm28b	347
6.166.3.27g_sFontCm28i	347
6.166.3.28g_sFontCm30	347
6.166.3.29g_sFontCm30b	347
6.166.3.30g_sFontCm30i	347
6.166.3.31g_sFontCm32	348
6.166.3.32g_sFontCm32b	348
6.166.3.33g_sFontCm32i	348
6.166.3.34g_sFontCm34	348
6.166.3.35g_sFontCm34b	348
6.166.3.36g_sFontCm34i	348
6.166.3.37g_sFontCm36	348
6.166.3.38g_sFontCm36b	348
6.166.3.39g_sFontCm36i	348
6.166.3.40g_sFontCm38	348

6.166.3.41g_sFontCm38b	348
6.166.3.42g_sFontCm38i	348
6.166.3.43g_sFontCm40	348
6.166.3.44g_sFontCm40b	348
6.166.3.45g_sFontCm40i	348
6.166.3.46g_sFontCm42	348
6.166.3.47g_sFontCm42b	348
6.166.3.48g_sFontCm42i	348
6.166.3.49g_sFontCm44	348
6.166.3.50g_sFontCm44b	348
6.166.3.51g_sFontCm44i	348
6.166.3.52g_sFontCm46	348
6.166.3.53g_sFontCm46b	348
6.166.3.54g_sFontCm46i	348
6.166.3.55g_sFontCm48	349
6.166.3.56g_sFontCm48b	349
6.166.3.57g_sFontCm48i	349
6.166.3.58g_sFontCmsc12	349
6.166.3.59g_sFontCmsc14	349
6.166.3.60g_sFontCmsc16	349
6.166.3.61g_sFontCmsc18	349
6.166.3.62g_sFontCmsc20	349
6.166.3.63g_sFontCmsc22	349
6.166.3.64g_sFontCmsc24	349
6.166.3.65g_sFontCmsc26	349
6.166.3.66g_sFontCmsc28	349
6.166.3.67g_sFontCmsc30	349
6.166.3.68g_sFontCmsc32	349
6.166.3.69g_sFontCmsc34	349
6.166.3.70g_sFontCmsc36	349
6.166.3.71g_sFontCmsc38	349
6.166.3.72g_sFontCmsc40	349
6.166.3.73g_sFontCmsc42	349
6.166.3.74g_sFontCmsc44	349

6.166.3.75g_sFontCmsc46	349
6.166.3.76g_sFontCmsc48	349
6.166.3.77g_sFontCmss12	349
6.166.3.78g_sFontCmss12b	349
6.166.3.79g_sFontCmss12i	350
6.166.3.80g_sFontCmss14	350
6.166.3.81g_sFontCmss14b	350
6.166.3.82g_sFontCmss14i	350
6.166.3.83g_sFontCmss16	350
6.166.3.84g_sFontCmss16b	350
6.166.3.85g_sFontCmss16i	350
6.166.3.86g_sFontCmss18	350
6.166.3.87g_sFontCmss18b	350
6.166.3.88g_sFontCmss18i	350
6.166.3.89g_sFontCmss20	350
6.166.3.90g_sFontCmss20b	350
6.166.3.91g_sFontCmss20i	350
6.166.3.92g_sFontCmss22	350
6.166.3.93g_sFontCmss22b	350
6.166.3.94g_sFontCmss22i	350
6.166.3.95g_sFontCmss24	350
6.166.3.96g_sFontCmss24b	350
6.166.3.97g_sFontCmss24i	350
6.166.3.98g_sFontCmss26	350
6.166.3.99g_sFontCmss26b	350
6.166.3.100g_sFontCmss26i	350
6.166.3.101g_sFontCmss28	350
6.166.3.102g_sFontCmss28b	350
6.166.3.103g_sFontCmss28i	351
6.166.3.104g_sFontCmss30	351
6.166.3.105g_sFontCmss30b	351
6.166.3.106g_sFontCmss30i	351
6.166.3.107g_sFontCmss32	351
6.166.3.108g_sFontCmss32b	351

6.166.3.109_sFontCmss32i	351
6.166.3.110_sFontCmss34	351
6.166.3.111_sFontCmss34b	351
6.166.3.112_sFontCmss34i	351
6.166.3.113_sFontCmss36	351
6.166.3.114_sFontCmss36b	351
6.166.3.115_sFontCmss36i	351
6.166.3.116_sFontCmss38	351
6.166.3.117_sFontCmss38b	351
6.166.3.118_sFontCmss38i	351
6.166.3.119_sFontCmss40	351
6.166.3.120_sFontCmss40b	351
6.166.3.121_sFontCmss40i	351
6.166.3.122_sFontCmss42	351
6.166.3.123_sFontCmss42b	351
6.166.3.124_sFontCmss42i	351
6.166.3.125_sFontCmss44	351
6.166.3.126_sFontCmss44b	351
6.166.3.127_sFontCmss44i	352
6.166.3.128_sFontCmss46	352
6.166.3.129_sFontCmss46b	352
6.166.3.130_sFontCmss46i	352
6.166.3.131_sFontCmss48	352
6.166.3.132_sFontCmss48b	352
6.166.3.133_sFontCmss48i	352
6.166.3.134_sFontFixed6x8	352
6.167 include/imgbutton.h File Reference	352
6.167.1 Define Documentation	353
6.167.1.1 IB_STYLE_AUTO_REPEAT	353
6.167.1.2 IB_STYLE_FILL	353
6.167.1.3 IB_STYLE_IMAGE_OFF	353
6.167.1.4 IB_STYLE_KEYCAP_OFF	354
6.167.1.5 IB_STYLE_PRESSED	354
6.167.1.6 IB_STYLE_RELEASE_NOTIFY	354

6.167.1.7 IB_STYLE_TEXT	354
6.167.1.8 ImageButton	354
6.167.1.9 ImageButtonAutoRepeatDelaySet	356
6.167.1.10 ImageButtonAutoRepeatOff	356
6.167.1.11 ImageButtonAutoRepeatOn	357
6.167.1.12 ImageButtonAutoRepeatRateSet	357
6.167.1.13 ImageButtonBackgroundColorSet	358
6.167.1.14 ImageButtonCallbackSet	358
6.167.1.15 ImageButtonFillColorSet	359
6.167.1.16 ImageButtonFillOff	359
6.167.1.17 ImageButtonFillOn	360
6.167.1.18 ImageButtonForegroundColorSet	360
6.167.1.19 ImageButtonImageKeycapSet	361
6.167.1.20 ImageButtonImageOff	361
6.167.1.21 ImageButtonImageOn	362
6.167.1.22 ImageButtonImagePressedSet	362
6.167.1.23 ImageButtonImageSet	363
6.167.1.24 ImageButtonKeycapOff	363
6.167.1.25 ImageButtonKeycapOffsetSet	364
6.167.1.26 ImageButtonKeycapOn	364
6.167.1.27 ImageButtonPressedColorSet	365
6.167.1.28 ImageButtonStruct	365
6.167.1.29 ImageButtonTextOff	367
6.167.1.30 ImageButtonTextOn	368
6.167.1.31 ImageButtonTextSet	368
6.167.2 Function Documentation	368
6.167.2.1 ImageButtonInit	369
6.167.2.2 ImageButtonMsgProc	369
6.168 include/listbox.h File Reference	369
6.169 include/pushbutton.h File Reference	370
6.170 include/radiobutton.h File Reference	372
6.171 include/sed1335.h File Reference	373
6.171.1 Define Documentation	375
6.171.1.1 SED1335_APH	375

6.171.1.2 SED1335_APL	375
6.171.1.3 SED1335_CGRAM_ADR	375
6.171.1.4 SED1335_CM	375
6.171.1.5 SED1335_CR	375
6.171.1.6 SED1335_CRX	375
6.171.1.7 SED1335_CRY	375
6.171.1.8 SED1335_CSRDIR_D	375
6.171.1.9 SED1335_CSRDIR_L	375
6.171.1.10 SED1335_CSRDIR_R	375
6.171.1.11 SED1335_CSRDIR_U	375
6.171.1.12 SED1335_CSRF_P2	375
6.171.1.13 SED1335_CSRFORM	376
6.171.1.14 SED1335_CSRR	376
6.171.1.15 SED1335_CSRW	376
6.171.1.16 SED1335_DISP_OFF	376
6.171.1.17 SED1335_DISP_ON	376
6.171.1.18 SED1335_DM1	376
6.171.1.19 SED1335_DM2	376
6.171.1.20 SED1335_FLASH	376
6.171.1.21 SED1335_FX	376
6.171.1.22 SED1335_FY	376
6.171.1.23 SED1335_GRAPHICSIZE	376
6.171.1.24 SED1335_GRAPHICSTART	376
6.171.1.25 SED1335_HDOT_SCR	376
6.171.1.26 SED1335_IV	376
6.171.1.27 SED1335_LF	376
6.171.1.28 SED1335_LINES	376
6.171.1.29 SED1335_M0	376
6.171.1.30 SED1335_M1	376
6.171.1.31 SED1335_M2	376
6.171.1.32 SED1335_MEM_END	376
6.171.1.33 SED1335_MREAD	376
6.171.1.34 SED1335_MWRITE	376
6.171.1.35 SED1335_MX0	376

6.171.1.36SED1335_MX1	377
6.171.1.37SED1335_OV	377
6.171.1.38SED1335_OVLAY	377
6.171.1.39SED1335_OVLAY_P1	377
6.171.1.40SED1335_SAD1H	377
6.171.1.41SED1335_SAD1L	377
6.171.1.42SED1335_SAD2H	377
6.171.1.43SED1335_SAD2L	377
6.171.1.44SED1335_SAD3H	377
6.171.1.45SED1335_SAD3L	377
6.171.1.46SED1335_SAD4H	377
6.171.1.47SED1335_SAD4L	377
6.171.1.48SED1335_SAGH	377
6.171.1.49SED1335_SAGL	377
6.171.1.50SED1335_SCR_WIDTH	377
6.171.1.51SED1335_SCRD	377
6.171.1.52SED1335_SCROLL	377
6.171.1.53SED1335_SL1	377
6.171.1.54SED1335_SL2	377
6.171.1.55SED1335_SLEEP_IN	377
6.171.1.56SED1335_SYS_P1	377
6.171.1.57SED1335_SYS_P2	377
6.171.1.58SED1335_SYSTEM_SET	377
6.171.1.59SED1335_TCR	378
6.171.1.60SED1335_TEXTSIZE	378
6.171.1.61SED1335_WF	378
6.171.1.62SED1335_WS	378
6.171.2 Function Documentation	378
6.171.2.1 GLCD_Bitmap	378
6.171.2.2 GLCD_ClearGraphic	378
6.171.2.3 GLCD_ClearText	378
6.171.2.4 GLCD_Initialize	379
6.171.2.5 GLCD_ReadByteFromROMMemory	379
6.171.2.6 GLCD_ReadData	379

6.171.2.7 GLCD_SetCursorAddress	379
6.171.2.8 GLCD_SetPixel	379
6.171.2.9 GLCD_TextGoTo	379
6.171.2.10 GLCD_WriteCommand	380
6.171.2.11 GLCD_WriteData	380
6.171.2.12 GLCD_WriteText	380
6.172 include/slider.h File Reference	380
6.172.1 Define Documentation	382
6.172.1.1 SL_STYLE_BACKG_FILL	382
6.172.1.2 SL_STYLE_BACKG_IMG	382
6.172.1.3 SL_STYLE_BACKG_TEXT	382
6.172.1.4 SL_STYLE_BACKG_TEXT_OPAQUE	382
6.172.1.5 SL_STYLE_FILL	382
6.172.1.6 SL_STYLE_IMG	382
6.172.1.7 SL_STYLE_LOCKED	382
6.172.1.8 SL_STYLE_OUTLINE	383
6.172.1.9 SL_STYLE_TEXT	383
6.172.1.10 SL_STYLE_TEXT_OPAQUE	383
6.172.1.11 SL_STYLE_VERTICAL	383
6.172.1.12 Slider	383
6.172.1.13 SliderBackgroundImageOff	385
6.172.1.14 SliderBackgroundImageOn	385
6.172.1.15 SliderBackgroundImageSet	386
6.172.1.16 SliderBackgroundTextColorSet	386
6.172.1.17 SliderBackgroundTextOff	387
6.172.1.18 SliderBackgroundTextOn	387
6.172.1.19 SliderBackgroundTextOpaqueOff	388
6.172.1.20 SliderBackgroundTextOpaqueOn	388
6.172.1.21 SliderCallbackSet	389
6.172.1.22 SliderFillColorBackgroundedSet	389
6.172.1.23 SliderFillColorSet	390
6.172.1.24 SliderFillOff	390
6.172.1.25 SliderFillOn	391
6.172.1.26 SliderFontSet	391

6.172.1.27SliderImageOff	392
6.172.1.28SliderImageOn	392
6.172.1.29SliderImageSet	393
6.172.1.30SliderLock	393
6.172.1.31SliderOutlineColorSet	394
6.172.1.32SliderOutlineOff	394
6.172.1.33SliderOutlineOn	395
6.172.1.34SliderRangeSet	395
6.172.1.35SliderStruct	396
6.172.1.36SliderTextColorSet	398
6.172.1.37SliderTextOff	398
6.172.1.38SliderTextOn	399
6.172.1.39SliderTextOpaqueOff	399
6.172.1.40SliderTextOpaqueOn	400
6.172.1.41SliderTextSet	400
6.172.1.42SliderUnlock	401
6.172.1.43SliderValueSet	401
6.172.2 Function Documentation	402
6.172.2.1 SliderInit	402
6.172.2.2 SliderMsgProc	402
6.173include/widget.h File Reference	403
6.173.1 Define Documentation	404
6.173.1.1 WIDGET_MSG_PAINT	404
6.173.1.2 WIDGET_MSG_PTR_DOWN	404
6.173.1.3 WIDGET_MSG_PTR_MOVE	404
6.173.1.4 WIDGET_MSG_PTR_UP	404
6.173.1.5 WIDGET_ROOT	404
6.173.1.6 WidgetPaint	404
6.173.2 Typedef Documentation	405
6.173.2.1 tWidget	405
6.173.3 Function Documentation	405
6.173.3.1 WidgetAdd	405
6.173.3.2 WidgetDefaultMsgProc	406
6.173.3.3 WidgetMessageQueueAdd	406

6.173.3.4 WidgetMessageQueueProcess	407
6.173.3.5 WidgetMessageSendPostOrder	407
6.173.3.6 WidgetMessageSendPreOrder	408
6.173.3.7 WidgetMutexGet	408
6.173.3.8 WidgetMutexInit	409
6.173.3.9 WidgetMutexPut	409
6.173.3.10 WidgetPointerMessage	410
6.173.3.11 WidgetRemove	410
6.173.4 Variable Documentation	410
6.173.4.1 g_sRoot	410
6.174 OLED/graphic.c File Reference	410
6.174.1 Function Documentation	411
6.174.1.1 GLCD_Circle	411
6.174.1.2 GLCD_Line	411
6.174.1.3 GLCD_Rectangle	411
6.174.1.4 GLCD_SetPixel	411
6.174.2 Variable Documentation	411
6.174.2.1 color	411
6.175 src/main.c File Reference	411
6.175.1 Function Documentation	411
6.175.1.1 main	411
6.176 src/msp430-main.c File Reference	411
6.176.1 Function Documentation	412
6.176.1.1 main	412
6.177 src/old_main.c File Reference	412
6.177.1 Function Documentation	412
6.177.1.1 main	412

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Primitives_api	9
Listbox_api	13
Pushbutton_api	30
RadioButton_api	54

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

__Widget	69
tCanvasWidget	The structure that describes a canvas widget	70
tCheckBoxWidget	The structure that describes a check box widget	72
tContainerWidget	The structure that describes a container widget	74
tContext	75
tDisplay	This structure defines the characteristics of a display driver	76
tFont	This structure describes a font used for drawing text onto the screen	79
tImageButtonWidget	The structure that describes a image button widget	80
tListBoxWidget	The structure that describes a listbox widget	83
tPushButtonWidget	The structure that describes a push button widget	83
tRadioButtonWidget	The structure that describes a radio button widget	86
tRectangle	88
tSliderWidget	The structure that describes a slider widget	89
tWidgetMessageQueue	91

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

driver/ sed1335-AT91.c	93
driver/ sed1335-avr.c	95
driver/ sed1335-LPC2100.c	96
driver/ sed1335-MSP430.c	98
driver/ sed1335-STM32.c	99
driver/ sed1335.c	100
gllib/ canvas.c	104
gllib/ checkbox.c	105
gllib/ circle.c	106
gllib/ container.c	107
gllib/ context.c	108
gllib/ image.c	150
gllib/ imgbutton.c	150
gllib/ line.c	152
gllib/ listbox.c	153
gllib/ offscr1bpp.c	154
gllib/ offscr4bpp.c	154
gllib/ offscr8bpp.c	155
gllib/ pushbutton.c	155
gllib/ radiobutton.c	155
gllib/ rectangle.c	155
gllib/ slider.c	158
gllib/ string.c	159
gllib/ widget.c	162
gllib/fonts/ fontcm12.c	109
gllib/fonts/ fontcm12b.c	110
gllib/fonts/ fontcm12i.c	110
gllib/fonts/ fontcm14.c	110
gllib/fonts/ fontcm14b.c	111

gllib/fonts/fontcm14i.c	111
gllib/fonts/fontcm16.c	111
gllib/fonts/fontcm16b.c	112
gllib/fonts/fontcm16i.c	112
gllib/fonts/fontcm18.c	112
gllib/fonts/fontcm18b.c	112
gllib/fonts/fontcm18i.c	113
gllib/fonts/fontcm20.c	113
gllib/fonts/fontcm20b.c	113
gllib/fonts/fontcm20i.c	114
gllib/fonts/fontcm22.c	114
gllib/fonts/fontcm22b.c	114
gllib/fonts/fontcm22i.c	115
gllib/fonts/fontcm24.c	115
gllib/fonts/fontcm24b.c	115
gllib/fonts/fontcm24i.c	115
gllib/fonts/fontcm26.c	116
gllib/fonts/fontcm26b.c	116
gllib/fonts/fontcm26i.c	116
gllib/fonts/fontcm28.c	117
gllib/fonts/fontcm28b.c	117
gllib/fonts/fontcm28i.c	117
gllib/fonts/fontcm30.c	118
gllib/fonts/fontcm30b.c	118
gllib/fonts/fontcm30i.c	118
gllib/fonts/fontcm32.c	118
gllib/fonts/fontcm32b.c	119
gllib/fonts/fontcm32i.c	119
gllib/fonts/fontcm34.c	119
gllib/fonts/fontcm34b.c	120
gllib/fonts/fontcm34i.c	120
gllib/fonts/fontcm36.c	120
gllib/fonts/fontcm36b.c	121
gllib/fonts/fontcm36i.c	121
gllib/fonts/fontcm38.c	121
gllib/fonts/fontcm38b.c	121
gllib/fonts/fontcm38i.c	122
gllib/fonts/fontcm40.c	122
gllib/fonts/fontcm40b.c	122
gllib/fonts/fontcm40i.c	123
gllib/fonts/fontcm42.c	123
gllib/fonts/fontcm42b.c	123
gllib/fonts/fontcm42i.c	124
gllib/fonts/fontcm44.c	124
gllib/fonts/fontcm44b.c	124
gllib/fonts/fontcm44i.c	124
gllib/fonts/fontcm46.c	125
gllib/fonts/fontcm46b.c	125
gllib/fonts/fontcm46i.c	125
gllib/fonts/fontcm48.c	126

gllib/fonts/ fontcm48b.c	126
gllib/fonts/ fontcm48i.c	126
gllib/fonts/ fontcmsc12.c	127
gllib/fonts/ fontcmsc14.c	127
gllib/fonts/ fontcmsc16.c	127
gllib/fonts/ fontcmsc18.c	127
gllib/fonts/ fontcmsc20.c	128
gllib/fonts/ fontcmsc22.c	128
gllib/fonts/ fontcmsc24.c	128
gllib/fonts/ fontcmsc26.c	129
gllib/fonts/ fontcmsc28.c	129
gllib/fonts/ fontcmsc30.c	129
gllib/fonts/ fontcmsc32.c	130
gllib/fonts/ fontcmsc34.c	130
gllib/fonts/ fontcmsc36.c	130
gllib/fonts/ fontcmsc38.c	130
gllib/fonts/ fontcmsc40.c	131
gllib/fonts/ fontcmsc42.c	131
gllib/fonts/ fontcmsc44.c	131
gllib/fonts/ fontcmsc46.c	132
gllib/fonts/ fontcmsc48.c	132
gllib/fonts/ fontcmss12.c	132
gllib/fonts/ fontcmss12b.c	133
gllib/fonts/ fontcmss12i.c	133
gllib/fonts/ fontcmss14.c	133
gllib/fonts/ fontcmss14b.c	133
gllib/fonts/ fontcmss14i.c	134
gllib/fonts/ fontcmss16.c	134
gllib/fonts/ fontcmss16b.c	134
gllib/fonts/ fontcmss16i.c	135
gllib/fonts/ fontcmss18.c	135
gllib/fonts/ fontcmss18b.c	135
gllib/fonts/ fontcmss18i.c	136
gllib/fonts/ fontcmss20.c	136
gllib/fonts/ fontcmss20b.c	136
gllib/fonts/ fontcmss20i.c	136
gllib/fonts/ fontcmss22.c	137
gllib/fonts/ fontcmss22b.c	137
gllib/fonts/ fontcmss22i.c	137
gllib/fonts/ fontcmss24.c	138
gllib/fonts/ fontcmss24b.c	138
gllib/fonts/ fontcmss24i.c	138
gllib/fonts/ fontcmss26.c	139
gllib/fonts/ fontcmss26b.c	139
gllib/fonts/ fontcmss26i.c	139
gllib/fonts/ fontcmss28.c	139
gllib/fonts/ fontcmss28b.c	140
gllib/fonts/ fontcmss28i.c	140
gllib/fonts/ fontcmss30.c	140
gllib/fonts/ fontcmss30b.c	141

gllib/fonts/fontcmss30i.c	141
gllib/fonts/fontcmss32.c	141
gllib/fonts/fontcmss32b.c	142
gllib/fonts/fontcmss32i.c	142
gllib/fonts/fontcmss34.c	142
gllib/fonts/fontcmss34b.c	142
gllib/fonts/fontcmss34i.c	143
gllib/fonts/fontcmss36.c	143
gllib/fonts/fontcmss36b.c	143
gllib/fonts/fontcmss36i.c	144
gllib/fonts/fontcmss38.c	144
gllib/fonts/fontcmss38b.c	144
gllib/fonts/fontcmss38i.c	145
gllib/fonts/fontcmss40.c	145
gllib/fonts/fontcmss40b.c	145
gllib/fonts/fontcmss40i.c	145
gllib/fonts/fontcmss42.c	146
gllib/fonts/fontcmss42b.c	146
gllib/fonts/fontcmss42i.c	146
gllib/fonts/fontcmss44.c	147
gllib/fonts/fontcmss44b.c	147
gllib/fonts/fontcmss44i.c	147
gllib/fonts/fontcmss46.c	148
gllib/fonts/fontcmss46b.c	148
gllib/fonts/fontcmss46i.c	148
gllib/fonts/fontcmss48.c	148
gllib/fonts/fontcmss48b.c	149
gllib/fonts/fontcmss48i.c	149
gllib/fonts/fontfixed6x8.c	149
include/canvas.h	263
include/checkbox.h	281
include/container.h	296
include/graphic.h	308
include/gplib.h	309
include/imgbutton.h	352
include/listbox.h	369
include/pushbutton.h	370
include/radiobutton.h	372
include/sed1335.h	373
include/slider.h	380
include/widget.h	403
include/build/lpc210x.h	169
include/build/lpc213x.h	181
include/build/lpc23xx.h	195
OLED/graphic.c	410
src/main.c	411
src/msp430-main.c	411
src/old_main.c	412

Chapter 4

Module Documentation

4.1 Primitives_api

Defines

- #define DPYCOLORTRANSLATE(c)

Functions

- void GrOffScreen1BPPInit (tDisplay *pDisplay, unsigned char *puImage, long lWidth, long lHeight)
- void GrOffScreen4BPPInit (tDisplay *pDisplay, unsigned char *puImage, long lWidth, long lHeight)
- void GrOffScreen4BPPPPaletteSet (tDisplay *pDisplay, unsigned long *pulPalette, unsigned long ulOffset, unsigned long ulCount)
- void GrOffScreen8BPPInit (tDisplay *pDisplay, unsigned char *puImage, long lWidth, long lHeight)
- void GrOffScreen8BPPPPaletteSet (tDisplay *pDisplay, unsigned long *pulPalette, unsigned long ulOffset, unsigned long ulCount)

4.1.1 Define Documentation

4.1.1.1 #define DPYCOLORTRANSLATE(c)

Value:

```
(((((c) & 0x00ff0000) >> 16) * 19661) + \
    (((c) & 0x0000ff00) >> 8) * 38666) + \
    (((c) & 0x000000ff) * 7209)) / \
    (65536 * 128))
```

4.1.2 Function Documentation

4.1.2.1 void GrOffScreen1BPPInit (tDisplay * *pDisplay*, unsigned char * *puclImage*, long *lWidth*, long *lHeight*)

Initializes a 1 BPP off-screen buffer.

Parameters

<i>pDisplay</i>	is a pointer to the display structure to be configured for the 1 BPP off-screen buffer.
<i>puclImage</i>	is a pointer to the image buffer to be used for the off-screen buffer.
<i>lWidth</i>	is the width of the image buffer in pixels.
<i>lHeight</i>	is the height of the image buffer in pixels.

This function initializes a display structure, preparing it to draw into the supplied image buffer. The image buffer is assumed to be large enough to hold an image of the specified geometry.

Returns

None.

4.1.2.2 void GrOffScreen4BPPInit (tDisplay * *pDisplay*, unsigned char * *puclImage*, long *lWidth*, long *lHeight*)

Initializes a 4 BPP off-screen buffer.

Parameters

<i>pDisplay</i>	is a pointer to the display structure to be configured for the 4 BPP off-screen buffer.
<i>puclImage</i>	is a pointer to the image buffer to be used for the off-screen buffer.
<i>lWidth</i>	is the width of the image buffer in pixels.
<i>lHeight</i>	is the height of the image buffer in pixels.

This function initializes a display structure, preparing it to draw into the supplied image buffer. The image buffer is assumed to be large enough to hold an image of the specified geometry.

Returns

None.

4.1.2.3 void GrOffScreen4BPPPaletteSet (tDisplay * *pDisplay*, unsigned long * *pulPalette*, unsigned long *ulOffset*, unsigned long *ulCount*)

Sets the palette of a 4 BPP off-screen buffer.

Parameters

<i>pDisplay</i>	is a pointer to the display structure for the 4 BPP off-screen buffer.
<i>pulPalette</i>	is a pointer to the array of 24-bit RGB values to be placed into the palette.
<i>ulOffset</i>	is the starting offset into the image palette.
<i>ulCount</i>	is the number of palette entries to set.

This function sets the entries of the palette used by the 4 BPP off-screen buffer. The palette is used to select colors for drawing via `GrOffScreen4BPPColorTranslate()`, and for the final rendering of the image to a real display via `GrlImageDraw()`.

Returns

None.

4.1.2.4 void GrOffScreen8BPPInit (*tDisplay* * *pDisplay*, *unsigned char* * *puclImage*, *long* *IWidth*, *long* *IHeight*)

Initializes an 8 BPP off-screen buffer.

Parameters

<i>pDisplay</i>	is a pointer to the display structure to be configured for the 4 BPP off-screen buffer.
<i>puclImage</i>	is a pointer to the image buffer to be used for the off-screen buffer.
<i>IWidth</i>	is the width of the image buffer in pixels.
<i>IHeight</i>	is the height of the image buffer in pixels.

This function initializes a display structure, preparing it to draw into the supplied image buffer. The image buffer is assumed to be large enough to hold an image of the specified geometry.

Returns

None.

4.1.2.5 void GrOffScreen8BPPPaletteSet (*tDisplay* * *pDisplay*, *unsigned long* * *pulPalette*, *unsigned long* *ulOffset*, *unsigned long* *ulCount*)

Sets the palette of an 8 BPP off-screen buffer.

Parameters

<i>pDisplay</i>	is a pointer to the display structure for the 4 BPP off-screen buffer.
<i>pulPalette</i>	is a pointer to the array of 24-bit RGB values to be placed into the palette.
<i>ulOffset</i>	is the starting offset into the image palette.
<i>ulCount</i>	is the number of palette entries to set.

This function sets the entries of the palette used by the 8 BPP off-screen buffer. The palette is used to select colors for drawing via `GrOffScreen4BPPColorTranslate()`, and for the final rendering of the image to a real display via `GrlImageDraw()`.

Returns

None.

4.2 Listbox_api

Data Structures

- struct `tListBoxWidget`

The structure that describes a listbox widget.

Defines

- #define `LISTBOX_STYLE_OUTLINE` 0x0001
- #define `LISTBOX_STYLE_LOCKED` 0x0002
- #define `LISTBOX_STYLE_WRAP` 0x0004
- #define `ListBoxStruct`(pParent, pNext, pChild, pDisplay, IX, IY, IWidth,IHeight, ulStyle, ulBgColor, ulSelBgColor,ulTextColor, ulSelTextColor, ulOutlineColor, p-Font,ppcText, usMaxEntries, usPopulatedEntries,pfnOnChange)
- #define `ListBox`(sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth,IHeight, ulStyle, ulBgColor, ulSelBgColor, ulTextColor,ulSelTextColor, ulOutlineColor, p-Font, ppcText, usMaxEntries, usPopulatedEntries, pfnOnChange)
- #define `ListBoxCallbackSet`(pWidget, pfnCallback)
- #define `ListBoxBackgroundColorSet`(pWidget, ulColor)
- #define `ListBoxSelectedBackgroundColorSet`(pWidget, ulColor)
- #define `ListBoxFontSet`(pWidget, pFnt)
- #define `ListBoxOutlineColorSet`(pWidget, ulColor)
- #define `ListBoxOutlineOff`(pWidget)
- #define `ListBoxOutlineOn`(pWidget)
- #define `ListBoxTextColorSet`(pWidget, ulColor)
- #define `ListBoxSelectedTextColorSet`(pWidget, ulColor)
- #define `ListBoxTextSet`(pWidget, pcTxt, ullIndex)
- #define `ListBoxLock`(pWidget)
- #define `ListBoxUnlock`(pWidget)
- #define `ListBoxWrapEnable`(pWidget)
- #define `ListBoxWrapDisable`(pWidget)
- #define `ListBoxClear`(pWidget)
- #define `ListBoxSelectionSet`(pWidget, sSel)
- #define `ListBoxSelectionGet`(pWidget) (((`tListBoxWidget` *) (pWidget))->s- Selected)

Functions

- int `ListBoxMsgProc` (`tWidget` *pWidget, unsigned ulMsg, unsigned ulParam1, un- signed ulParam2)
- void `ListBoxInit` (`tListBoxWidget` *pWidget, const `tDisplay` *pDisplay, const char **ppcText, unsigned short usMaxEntries, unsigned short usPopulatedEntries, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)
- int `ListBoxTextAdd` (`tListBoxWidget` *pWidget, const char *pcTxt)

Variables

- **tWidget tListBoxWidget::sBase**
The generic widget information.
- **unsigned tListBoxWidget::ulStyle**
- **unsigned long tListBoxWidget::ulBackgroundColor**
The 24-bit RGB color used as the background for the listbox.
- **unsigned long tListBoxWidget::ulSelectedBackgroundColor**
- **unsigned long tListBoxWidget::ulTextColor**
The 24-bit RGB color used to draw text on this listbox.
- **unsigned long tListBoxWidget::ulSelectedTextColor**
The 24-bit RGB color used to draw the selected text on this listbox.
- **unsigned long tListBoxWidget::ulOutlineColor**
- **const tFont * tListBoxWidget::pFont**
A pointer to the font used to render the listbox text.
- **const char ** tListBoxWidget::ppcText**
- **unsigned short tListBoxWidget::usMaxEntries**
The number of elements in the array pointed to by pccText.
- **unsigned short tListBoxWidget::usPopulated**
- **short tListBoxWidget::sSelected**
- **unsigned short tListBoxWidget::usStartEntry**
- **unsigned short tListBoxWidget::usOldestEntry**
- **unsigned short tListBoxWidget::usScrolled**
- **unsigned tListBoxWidget::lPointerY**
- **void(* tListBoxWidget::pfnOnChange)(tWidget *pWidget, short sSelIndex)**

4.2.1 Define Documentation

```
4.2.1.1 #define ListBox( sName, pParent, pNext, pChild, pDisplay, lX, lY, lWidth,
    lHeight, ulStyle, ulBgColor, ulSelBgColor, ulTextColor, ulSelTextColor,
    ulOutlineColor, pFont, ppcText, usMaxEntries, usPopulatedEntries, pfnOnChange )
```

Value:

```
tListBoxWidget sName =
    ListBoxStruct(pParent, pNext, pChild, pDisplay, lX, lY, lWidth, lHeight, \
        ulStyle, ulBgColor, ulSelBgColor, ulTextColor, \
        ulSelTextColor, ulOutlineColor, pFont, ppcText, \
        usMaxEntries, usPopulatedEntries, pfnOnChange)
```

Declares an initialized variable containing a listbox widget data structure.

Parameters

<i>sName</i>	is the name of the variable to be declared.
<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the listbox.
<i>IX</i>	is the X coordinate of the upper left corner of the listbox.
<i>IY</i>	is the Y coordinate of the upper left corner of the listbox.
<i>IWidth</i>	is the width of the listbox.
<i>IHeight</i>	is the height of the listbox.
<i>ulStyle</i>	is the style to be applied to the listbox.
<i>ulBgColor</i>	is the background color for the listbox.
<i>ulSelBg-Color</i>	is the background color for the selected element in the listbox.
<i>ulTextColor</i>	is the color used to draw text on the listbox.
<i>ulSelTextColor</i>	is the color used to draw the selected element text in the listbox.
<i>ulOutline-Color</i>	is the color used to outline the listbox.
<i>pFont</i>	is a pointer to the font to be used to draw text on the listbox.
<i>ppcText</i>	is a pointer to the string table for the listbox.
<i>usMaxEntries</i>	provides the number of entries in the <i>ppcText</i> array and represents the maximum number of strings the listbox can hold.
<i>usPopulatedEntries</i>	indicates the number of entries in the <i>ppcText</i> array that currently hold valid string for the listbox.
<i>pfnOnChange</i>	is a pointer to the application callback for the listbox.

This macro declares a variable containing an initialized listbox widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

ulStyle is the logical OR of the following:

- **LISTBOX_STYLE_OUTLINE** to indicate that the listbox should be outlined.
- **LISTBOX_STYLE_LOCKED** to indicate that the listbox should ignore user input and merely display its contents.
- **LISTBOX_STYLE_WRAP** to indicate that the listbox should discard the oldest string it contains if asked to add a new string while the string table is already full.

Returns

Nothing; this is not a function.

4.2.1.2 #define LISTBOX_STYLE_LOCKED 0x0002

This flag indicates that the listbox is not interactive but merely displays strings. Scrolling of the listbox content is supported when this flag is set but widgets using this style do not make callbacks to the application and do not support selection and deselection of entries. This may be used if a listbox is intended, for example, as a text output or status reporting control.

4.2.1.3 #define LISTBOX_STYLE_OUTLINE 0x0001

This flag indicates that the listbox should be outlined. If enabled, the widget is drawn with a two pixel border, the outer, single pixel rectangle of which is in the color found in the ulOutlineColor field of the widget structure and the inner rectangle in color ulBackgroundColor.

4.2.1.4 #define LISTBOX_STYLE_WRAP 0x0004

This flag controls the behavior of the listbox if a new string is added when the string table (ppcText) is already full. If this style is set, the oldest string in the table is replaced with new one and, if the discarded string was currently displayed, the display positions will be fixed up to ensure that the (new) oldest string remains at the top of the listbox. If this style is not set, the attempt to set a new string will fail if the table is full.

4.2.1.5 #define ListBoxBackgroundColorSet(pWidget, ulColor)

Value:

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    pW->ulBackgroundColor = ulColor; \
}                                \
while (0)                         \
```

Sets the background color of a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use for the listbox background.

This function changes the color used for the listbox background on the display. The display is not updated until the next paint request.

Returns

None.

4.2.1.6 #define ListBoxCallbackSet(pWidget, pfnCallback)

Value:

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    pW->pfnOnChange = pfnCallback; \
}                                \
while(0)
```

Sets the function to call when the listbox selection changes.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
<i>pfnCallback</i>	is a pointer to the function to call.

This function sets the function to be called when the selected element in this listbox changes. If style **LISTBOX_STYLE_LOCKED** is selected, or the callback function pointer set is NULL, no callbacks will be made.

Returns

None.

4.2.1.7 #define ListBoxClear(pWidget)

Value:

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    pW->usPopulated = 0;          \
    pW->sSelected = (short)0xFFFF; \
    pW->usStartEntry = 0;          \
    pW->usOldestEntry = 0;         \
}                                \
while(0)
```

Empties the listbox.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
----------------	---

This function removes all text from a listbox widget. The display is not updated until the next paint request.

Returns

None.

4.2.1.8 #define ListBoxFontSet(*pWidget*, *pFnt*)

Value:

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget; \
    const tFont *pF = pFnt;     \
    pW->pFont = pF;           \
}                                \
while (0)
```

Sets the font for a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
<i>pFnt</i>	is a pointer to the font to use to draw text on the listbox.

This function changes the font used to draw text on the listbox. The display is not updated until the next paint request.

Returns

None.

4.2.1.9 #define ListBoxLock(*pWidget*)

Value:

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget; \
    pW->ulStyle |= LISTBOX_STYLE_LOCKED; \
}                                \
while (0)
```

Locks a listbox making it ignore attempts to select elements.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
----------------	---

This function locks a listbox widget and makes it ignore attempts to select or deselect an element. When locked, a listbox acts as a passive indicator. Strings may be added and the selected element changed via calls to `ListBoxSelectioSet()` but pointer activity will not change the selection and no callbacks will be made. In this mode, the user may still use the pointer to scroll the content of the listbox assuming it contains more strings that can be displayed in the widget area.

Returns

None.

4.2.1.10 #define ListBoxOutlineColorSet(pWidget, ulColor)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;      \
    pW->ulOutlineColor = ulColor;      \
}                                \
while(0)
```

Sets the outline color of a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to outline the listbox.

This function changes the color used to outline the listbox on the display. The display is not updated until the next paint request.

Returns

None.

4.2.1.11 #define ListBoxOutlineOff(pWidget)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;      \
    pW->ulStyle &= ~ (LISTBOX_STYLE_OUTLINE); \
}                                \
while(0)
```

Disables outlining of a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
----------------	---

This function disables the outlining of a listbox widget. The display is not updated until the next paint request.

Returns

None.

4.2.1.12 #define ListBoxOutlineOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    pW->ulStyle |= LISTBOX_STYLE_OUTLINE; \
}                                \
while(0)
```

Enables outlining of a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
----------------	---

This function enables the outlining of a listbox widget. The display is not updated until the next paint request.

Returns

None.

4.2.1.13 #define ListBoxSelectedBackgroundColorSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    pW->ulSelectedBackgroundColor = ulColor; \
}                                \
while(0)
```

Sets the background color of the selected element in a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use for the background of the selected element.

This function changes the color used for the background of the selected line of text on the display. The display is not updated until the next paint request.

Returns

None.

4.2.1.14 #define ListBoxSelectedTextColorSet(pWidget, ulColor)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;      \
    pW->ulSelectedTextColor = ulColor; \
}                                \
while(0)                            \
```

Sets the text color of the selected element in a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw the selected text on the listbox.

This function changes the color used to draw the selected element text on the display. The display is not updated until the next paint request.

Returns

None.

**4.2.1.15 #define ListBoxSelectionGet(pWidget) (((tListBoxWidget
*)(pWidget))->sSelected)**

Gets the index of the current selection within the listbox.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to be queried.
----------------	---

This function returns the index of the item currently selected in a listbox. If no selection has been made, 0xFFFF (-1) is returned.

Returns

None.

4.2.1.16 #define ListBoxSelectionSet(pWidget, sSel)**Value:**

```

do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    if((sSel) < pW->usPopulated)  \
    {                                \
        pW->sSelected = (sSel);    \
    }                                \
}                                \
while(0)

```

Sets the current selection within the listbox.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
<i>sSel</i>	is the index of the item to select.

This function selects an item within the list box. The display is not updated until the next paint request.

Returns

None.

```
4.2.1.17 #define ListBoxStruct( pParent, pNext, pChild, pDisplay, IX, IY, IWidth,
                           IHeight, ulStyle, ulBgColor, ulSelBgColor, ulTextColor, ulSelTextColor,
                           ulOutlineColor, pFont, ppcText, usMaxEntries, usPopulatedEntries, pfnOnChange
                           )
```

Declares an initialized listbox widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the listbox.
<i>IX</i>	is the X coordinate of the upper left corner of the listbox.
<i>IY</i>	is the Y coordinate of the upper left corner of the listbox.
<i>IWidth</i>	is the width of the listbox.
<i>IHeight</i>	is the height of the listbox.
<i>ulStyle</i>	is the style to be applied to the listbox.
<i>ulBgColor</i>	is the background color for the listbox.
<i>ulSelBg-Color</i>	is the background color for the selected element in the listbox.
<i>ulTextColor</i>	is the color used to draw text on the listbox.
<i>ulSelTextColor</i>	is the color used to draw the selected element text in the listbox.
<i>ulOutline-Color</i>	is the color used to outline the listbox.
<i>pFont</i>	is a pointer to the font to be used to draw text on the listbox.

<i>ppcText</i>	is a pointer to the string table for the listbox.
<i>usMaxEntries</i>	provides the number of entries in the <i>ppcText</i> array and represents the maximum number of strings the listbox can hold.
<i>usPopulatedEntries</i>	indicates the number of entries in the <i>ppcText</i> array that currently hold valid string for the listbox.
<i>pfnOnChange</i>	is a pointer to the application callback for the listbox.

This macro provides an initialized listbox widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
//!     tListBoxWidget g_sListBox = ListBoxStruct(...);  
//!
```

Or, in an array of variables:

```
//!     tListBoxWidget g_psListBox[] =  
//!     {  
//!         ListBoxStruct(...),  
//!         ListBoxStruct(...)  
//!     };  
//!
```

ulStyle is the logical OR of the following:

- **LISTBOX_STYLE_OUTLINE** to indicate that the listbox should be outlined.
- **LISTBOX_STYLE_LOCKED** to indicate that the listbox should ignore user input and merely display its contents.
- **LISTBOX_STYLE_WRAP** to indicate that the listbox should discard the oldest string it contains if asked to add a new string while the string table is already full.

Returns

Nothing; this is not a function.

4.2.1.18 #define ListBoxTextColorSet(pWidget, ulColor)

Value:

```
do                                \  
{                                \  
    tListBoxWidget *pW = pWidget;    \  
    pW->ulTextColor = ulColor;    \  
}                                \  
while(0)
```

Sets the text color of a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw text on the listbox.

This function changes the color used to draw text on the listbox on the display. The display is not updated until the next paint request.

Returns

None.

4.2.1.19 #define ListBoxTextSet(*pWidget*, *pcTxt*, *ullIndex*)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    const char *pcT = pcTxt;       \
    if(ullIndex < pW->usMaxEntries) \
    {                                \
        pW->ppcText[ullIndex] = pcT; \
    }                                \
}                                \
while(0)
```

Changes the text associated with an element in the listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to be modified.
<i>pcTxt</i>	is a pointer to the new text string.
<i>ullIndex</i>	is the index of the element whose string is to be replaced.

This function replaces the string associated with one of the listbox elements. This call should only be used to replace a string for an already-populated element. To add a new string, use [ListBoxTextAdd\(\)](#). The display is not updated until the next paint request.

Returns

None.

4.2.1.20 #define ListBoxUnlock(*pWidget*)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    pW->ulStyle &= ~LISTBOX_STYLE_LOCKED; \
}
```

```
    }                                \
    while(0)
```

Unlocks a listbox making it respond to pointer input.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
----------------	---

This function unlocks a listbox widget. When unlocked, a listbox will respond to pointer input by setting its selected element appropriately and informing the application of changes via callbacks.

Returns

None.

4.2.1.21 #define ListBoxWrapDisable(*pWidget*)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    pW->ulStyle |= LISTBOX_STYLE_WRAP; \
}                                \
while(0)
```

Disables text wrapping in a listbox.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
----------------	---

This function disables text wrapping in a listbox widget. With wrapping enabled, calls to [ListBoxTextAdd\(\)](#) made when the widget string table is full will discard the oldest string in favor of the new one. If wrapping is disabled, these calls will fail.

Returns

None.

4.2.1.22 #define ListBoxWrapEnable(*pWidget*)**Value:**

```
do                                \
{                                \
    tListBoxWidget *pW = pWidget;   \
    pW->ulStyle &= ~LISTBOX_STYLE_WRAP; \
}
```

```

}
while(0)
\
```

Enables wrapping in a listbox.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to modify.
----------------	---

This function enables text wrapping in a listbox widget. With wrapping enabled, calls to [ListBoxTextAdd\(\)](#) made when the widget string table is full will discard the oldest string in favor of the new one. If wrapping is disabled, these calls will fail.

Returns

None.

4.2.2 Function Documentation

4.2.2.1 void ListBoxInit (tListBoxWidget * *pWidget*, const tDisplay * *pDisplay*, const char ** *ppcText*, unsigned short *usMaxEntries*, unsigned short *usPopulatedEntries*, unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)

Initializes a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the listbox.
<i>ppcText</i>	is a pointer to an array of character pointers which will hold the strings that the listbox displays.
<i>usMax-Entries</i>	provides the total number of entries in the <i>ppcText</i> array.
<i>us- Populated- Entries</i>	provides the number of entries in the <i>ppcText</i> array which are populated.
<i>IX</i>	is the X coordinate of the upper left corner of the listbox.
<i>IY</i>	is the Y coordinate of the upper left corner of the listbox.
<i>IWidth</i>	is the width of the listbox.
<i>IHeight</i>	is the height of the listbox.

This function initializes the provided listbox widget.

Returns

None.

4.2.2.2 int ListBoxMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a listbox widget.

Parameters

<i>pWidget</i>	is a pointer to the listbox widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this listbox widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

4.2.2.3 int ListBoxTextAdd (tListBoxWidget * pListBox, const char * pcTxt)

Adds a line of text to a listbox.

Parameters

<i>pListBox</i>	is a pointer to the listbox widget that is to receive the new text string.
<i>pcTxt</i>	is a pointer to the string that is to be added to the listbox.

This function adds a new string to the listbox. If the listbox has style [LISTBOX_STYLE_WRAP](#) and the current string table is full, this function will discard the oldest string and replace it with the one passed here. If this style flag is absent, the function will return -1 if no empty entries exist in the string table for the widget.

The display is not automatically updated as a result of this function call. An application must call [WidgetPaint\(\)](#) to update the display after adding a new string to the listbox.

Note

To replace the string associated with a particular, existing element in the listbox, use [ListBoxTextSet\(\)](#).

Returns

Returns the string table index into which the new string has been placed if successful or -1 if the string table is full and [LISTBOX_STYLE_WRAP](#) is not set.

4.2.3 Variable Documentation

4.2.3.1 unsigned tListBoxWidget::lPointerY

The Y coordinate of the last pointer position we received. This is an internal variable used to manage scrolling of the listbox contents and must not be modified by an application using this widget class.

4.2.3.2 void(* tListBoxWidget::pfnOnChange)(tWidget *pWidget, short sSelIndex)

A pointer to the application-supplied callback function. This function will be called each time the selected element in the list box changes. The sSelIndex parameter contains the index of the selected string in ppcText array or, if no element is selected, 0xFFFF (-1).

4.2.3.3 const tFont* tListBoxWidget::pFont

A pointer to the font used to render the listbox text.

4.2.3.4 const char tListBoxWidget::ppcText**

A pointer to the array of string pointers representing the contents of the list box.

4.2.3.5 tWidget tListBoxWidget::sBase

The generic widget information.

4.2.3.6 short tListBoxWidget::sSelected

The index of the string currently selected in the list box. If no selection has been made, this will be set to 0xFFFF (-1).

4.2.3.7 unsigned long tListBoxWidget::ulBackgroundColor

The 24-bit RGB color used as the background for the listbox.

4.2.3.8 unsigned long tListBoxWidget::ulOutlineColor

The 24-bit RGB color used to outline this listbox, if LISTBOX_STYLE_OUTLINE is selected.

4.2.3.9 unsigned long tListBoxWidget::ulSelectedBackgroundColor

The 24-bit RGB color used as the background for the selected entry in the listbox.

4.2.3.10 unsigned long tListBoxWidget::uiSelectedTextColor

The 24-bit RGB color used to draw the selected text on this listbox.

4.2.3.11 unsigned tListBoxWidget::uiStyle

The style for this widget. This is a set of flags defined by LISTBOX_STYLE_xxx.

4.2.3.12 unsigned long tListBoxWidget::uiTextColor

The 24-bit RGB color used to draw text on this listbox.

4.2.3.13 unsigned short tListBoxWidget::usMaxEntries

The number of elements in the array pointed to by pccText.

4.2.3.14 unsigned short tListBoxWidget::usOldestEntry

The index of the oldest entry in the pccText array. This is used by the widget class to determine where to add a new string if the array is full and the listbox has style LISTBOX_STYLE_WRAP. This is an internal variable and must not be modified by an application using this widget class.

4.2.3.15 unsigned short tListBoxWidget::usPopulated

The number of elements in the array pointed to by pccText which are currently populated with strings.

4.2.3.16 unsigned short tListBoxWidget::usScrolled

A flag which we use to determine whether to change the selected element when the pointer is lifted. The listbox will change the selection if no scrolling was performed since the last WIDGET_MSG_PTR_DOWN was received. This is an internal variable and must not be modified by an application using this widget class.

4.2.3.17 unsigned short tListBoxWidget::usStartEntry

The index of the string that appears at the top of the list box. This is used by the widget class to control scrolling of the box content. This is an internal variable and must not be modified by an application using this widget class.

4.3 Pushbutton_api

Data Structures

- struct `tPushButtonWidget`

The structure that describes a push button widget.

Defines

- `#define PB_STYLE_OUTLINE 0x0001`
This flag indicates that the push button should be outlined.
- `#define PB_STYLE_FILL 0x0002`
This flag indicates that the push button should be filled.
- `#define PB_STYLE_TEXT 0x0004`
This flag indicates that the push button should have text drawn on it.
- `#define PB_STYLE_IMG 0x0008`
This flag indicates that the push button should have an image drawn on it.
- `#define PB_STYLE_TEXT_OPAQUE 0x0010`
- `#define PB_STYLE_AUTO_REPEAT 0x0020`
- `#define PB_STYLE_PRESSED 0x0040`
This flag indicates that the push button is pressed.
- `#define PB_STYLE_RELEASE_NOTIFY 0x0080`
- `#define CircularButtonStruct(pParent, pNext, pChild, pDisplay, IX, IY, IR, ulStyle, ulFillColor, ulPressFillColor,ulOutlineColor, ulTextColor, pFont, pcText,puclImage, pucPressImage, usAutoRepeatDelay,usAutoRepeatRate, pfnOnClick)`
- `#define CircularButton(sName, pParent, pNext, pChild, pDisplay, IX, IY, IR,ulStyle, ulFillColor, ulPressFillColor,ulOutlineColor, ulTextColor, pFont, pcText, puclImage, pucPressImage, usAutoRepeatDelay, usAutoRepeatRate,pfnOnClick)`
- `#define RectangularButtonStruct(pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, ulStyle, ulFillColor,ulPressFillColor, ulOutlineColor,ulTextColor, pFont, pcText, puclImage, pucPressImage, usAutoRepeatDelay,usAutoRepeatRate, pfnOnClick)`
- `#define RectangularButton(sName, pParent, pNext, pChild, pDisplay, IX, IY,IWidth, IHeight, ulStyle, ulFillColor,ulPressFillColor, ulOutlineColor,ulTextColor, pFont, pcText, puclImage, pucPressImage,usAutoRepeatDelay, usAutoRepeatRate, usAutoRepeatRate, pfnOnClick)`
- `#define PushButtonAutoRepeatDelaySet(pWidget, usDelay)`
- `#define PushButtonAutoRepeatOff(pWidget)`
- `#define PushButtonAutoRepeatOn(pWidget)`
- `#define PushButtonAutoRepeatRateSet(pWidget, usRate)`
- `#define PushButtonCallbackSet(pWidget, pfnOnClik)`
- `#define PushButtonFillColorSet(pWidget, ulColor)`
- `#define PushButtonFillColorPressedSet(pWidget, ulColor)`
- `#define PushButtonFillOff(pWidget)`
- `#define PushButtonFillOn(pWidget)`

- #define PushButtonFontSet(pWidget, pFnt)
- #define PushButtonImageSet(pWidget, plmg)
- #define PushButtonImagePressedSet(pWidget, plmg)
- #define PushButtonImageOff(pWidget)
- #define PushButtonImageOn(pWidget)
- #define PushButtonOutlineColorSet(pWidget, ulColor)
- #define PushButtonOutlineOff(pWidget)
- #define PushButtonOutlineOn(pWidget)
- #define PushButtonTextColorSet(pWidget, ulColor)
- #define PushButtonTextOff(pWidget)
- #define PushButtonTextOn(pWidget)
- #define PushButtonTextOpaqueOff(pWidget)
- #define PushButtonTextOpaqueOn(pWidget)
- #define PushButtonTextSet(pWidget, pcTxt)

Functions

- int **RectangularButtonMsgProc** (tWidget *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void **RectangularButtonInit** (tPushButtonWidget *pWidget, const tDisplay *pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)
- int **CircularButtonMsgProc** (tWidget *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void **CircularButtonInit** (tPushButtonWidget *pWidget, const tDisplay *pDisplay, unsigned IX, unsigned IY, unsigned IR)

4.3.1 Define Documentation

4.3.1.1 #define **CircularButton(sName, pParent, pNext, pChild, pDisplay, IX, IY, IR, ulStyle, ulFillColor, ulPressFillColor, ulOutlineColor, ulTextColor, pFont, pcText, pucImage, pucPressImage, usAutoRepeatDelay, usAutoRepeatRate, pfnOnClick)**

Value:

```
tPushButtonWidget sName =
    CircularButtonStruct (pParent, pNext, pChild, pDisplay, IX, IY, \
    IR, ulStyle, ulFillColor, ulPressFillColor, \
    ulOutlineColor, ulTextColor, pFont, pcText, \
    pucImage, pucPressImage, usAutoRepeatDelay, \
    usAutoRepeatRate, pfnOnClick)
```

Declares an initialized variable containing a circular push button widget data structure.

Parameters

<i>sName</i>	is the name of the variable to be declared.
<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the center of the push button.
<i>IY</i>	is the Y coordinate of the center of the push button.
<i>IR</i>	is the radius of the push button.
<i>ulStyle</i>	is the style to be applied to the push button.
<i>ulFillColor</i>	is the color used to fill in the push button.
<i>ulPressFill-Color</i>	is the color used to fill in the push button when it is pressed.
<i>ulOutline-Color</i>	is the color used to outline the push button.
<i>ulTextColor</i>	is the color used to draw text on the push button.
<i>pFont</i>	is a pointer to the font to be used to draw text on the push button.
<i>pcText</i>	is a pointer to the text to draw on this push button.
<i>pucImage</i>	is a pointer to the image to draw on this push button.
<i>pucPress-Image</i>	is a pointer to the image to draw on this push button when it is pressed.
<i>usAuto-Repeat-Delay</i>	is the delay before starting auto-repeat.
<i>usAuto-RepeatRate</i>	is the rate at which auto-repeat events are generated.
<i>pfnOnClick</i>	is a pointer to the function that is called when the push button is pressed.

This macro provides an initialized circular push button widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

ulStyle is the logical OR of the following:

- **PB_STYLE_OUTLINE** to indicate that the push button should be outlined.
- **PB_STYLE_FILL** to indicate that the push button should be filled.
- **PB_STYLE_TEXT** to indicate that the push button should have text drawn on it (using *pFont* and *pcText*).
- **PB_STYLE_IMG** to indicate that the push button should have an image drawn on it (using *pucImage*).
- **PB_STYLE_TEXT_OPAQUE** to indicate that the push button text should be drawn opaque (in other words, drawing the background pixels).
- **PB_STYLE_AUTO_REPEAT** to indicate that auto-repeat should be used.

- **PB_STYLE_RELEASE_NOTIFY** to indicate that the callback should be made when the button is released. If absent, the callback is called when the button is initially pressed.

Returns

Nothing; this is not a function.

4.3.1.2 #define CircularButtonStruct(pParent, pNext, pChild, pDisplay, IX, IY, IR, ulStyle, ulFillColor, ulPressFillColor, ulOutlineColor, ulTextColor, pFont, pcText, pucImage, pucPressImage, usAutoRepeatDelay, usAutoRepeatRate, pfnOnClick)

Value:

```
{
{
    sizeof(tPushButtonWidget),
    (tWidget *) (pParent),
    (tWidget *) (pNext),
    (tWidget *) (pChild),
    pDisplay,
    {
        (lX) - (lR),
        (lY) - (lR),
        (lX) + (lR),
        (lY) + (lR)
    },
    CircularButtonMsgProc
},
ulStyle,
ulFillColor,
ulPressFillColor,
ulOutlineColor,
ulTextColor,
pFont,
pcText,
pucImage,
pucPressImage,
usAutoRepeatDelay,
usAutoRepeatRate,
0,
pfnOnClick
}
```

Declares an initialized circular push button widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the center of the push button.
<i>IY</i>	is the Y coordinate of the center of the push button.
<i>IR</i>	is the radius of the push button.

<i>ulStyle</i>	is the style to be applied to the push button.
<i>ulFillColor</i>	is the color used to fill in the push button.
<i>ulPressFill-Color</i>	is the color used to fill in the push button when it is pressed.
<i>ulOutline-Color</i>	is the color used to outline the push button.
<i>ulTextColor</i>	is the color used to draw text on the push button.
<i>pFont</i>	is a pointer to the font to be used to draw text on the push button.
<i>pcText</i>	is a pointer to the text to draw on this push button.
<i>puclImage</i>	is a pointer to the image to draw on this push button.
<i>pucPress-Image</i>	is a pointer to the image to draw on this push button when it is pressed.
<i>usAuto-Repeat-Delay</i>	is the delay before starting auto-repeat.
<i>usAuto-RepeatRate</i>	is the rate at which auto-repeat events are generated.
<i>pfnOnClick</i>	is a pointer to the function that is called when the push button is pressed.

This macro provides an initialized circular push button widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
///!      tPushButtonWidget g_sPushButton = CircularButtonStruct(...);
///!
```

Or, in an array of variables:

```
///!      tPushButtonWidget g_psPushButtons[] =
///!      {
///!          CircularButtonStruct(...),
///!          CircularButtonStruct(...)
///!      };
///!
```

ulStyle is the logical OR of the following:

- **PB_STYLE_OUTLINE** to indicate that the push button should be outlined.
- **PB_STYLE_FILL** to indicate that the push button should be filled.
- **PB_STYLE_TEXT** to indicate that the push button should have text drawn on it (using *pFont* and *pcText*).
- **PB_STYLE_IMG** to indicate that the push button should have an image drawn on it (using *puclImage*).
- **PB_STYLE_TEXT_OPAQUE** to indicate that the push button text should be drawn opaque (in other words, drawing the background pixels).

- **PB_STYLE_AUTO_REPEAT** to indicate that auto-repeat should be used.
- **PB_STYLE_RELEASE_NOTIFY** to indicate that the callback should be made when the button is released. If absent, the callback is called when the button is initially pressed.

Returns

Nothing; this is not a function.

4.3.1.3 #define PB_STYLE_AUTO_REPEAT 0x0020

This flag indicates that the push button should auto-repeat, generating repeated click events while it is pressed.

4.3.1.4 #define PB_STYLE_FILL 0x0002

This flag indicates that the push button should be filled.

4.3.1.5 #define PB_STYLE_IMG 0x0008

This flag indicates that the push button should have an image drawn on it.

4.3.1.6 #define PB_STYLE_OUTLINE 0x0001

This flag indicates that the push button should be outlined.

4.3.1.7 #define PB_STYLE_PRESSED 0x0040

This flag indicates that the push button is pressed.

4.3.1.8 #define PB_STYLE_RELEASE_NOTIFY 0x0080

This flag indicates that the push button callback should be made when the button is released rather than when it is pressed. This does not affect the operation of auto repeat buttons.

4.3.1.9 #define PB_STYLE_TEXT 0x0004

This flag indicates that the push button should have text drawn on it.

4.3.1.10 #define PB_STYLE_TEXT_OPAQUE 0x0010

This flag indicates that the push button text should be drawn opaque (in other words, drawing the background pixels as well as the foreground pixels).

4.3.1.11 #define PushButtonAutoRepeatDelaySet(pWidget, usDelay)

Value:

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->usAutoRepeatDelay = usDelay; \
}                                \
while (0)
```

Sets the auto-repeat delay for a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
<i>usDelay</i>	is the number of pointer events before auto-repeat starts.

This function sets the delay before auto-repeat begins. Unpredictable behavior will occur if this is called while the push button is pressed.

Returns

None.

4.3.1.12 #define PushButtonAutoRepeatOff(pWidget)

Value:

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulStyle &= ~(PB_STYLE_AUTO_REPEAT); \
}                                \
while (0)
```

Disables auto-repeat for a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function disables the auto-repeat behavior of a push button.

Returns

None.

4.3.1.13 #define PushButtonAutoRepeatOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget;   \
    pW->ulStyle |= PB_STYLE_AUTO_REPEAT; \
}                                \
while(0)
```

Enables auto-repeat for a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function enables the auto-repeat behavior of a push button. Unpredictable behavior will occur if this is called while the push button is pressed.

Returns

None.

4.3.1.14 #define PushButtonAutoRepeatRateSet(*pWidget*, *usRate*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget;   \
    pW->usAutoRepeatRate = usRate;   \
}                                \
while(0)
```

Sets the auto-repeat rate for a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
<i>usRate</i>	is the number of pointer events between auto-repeat events.

This function sets the rate at which auto-repeat events occur. Unpredictable behavior will occur if this is called while the push button is pressed.

Returns

None.

4.3.1.15 #define PushButtonCallbackSet(*pWidget*, *pfnOnClik*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->pfnOnClick = pfnOnClik;  \
}                                \
while(0)
```

Sets the function to call when this push button widget is pressed.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
<i>pfnOnClik</i>	is a pointer to the function to call.

This function sets the function to be called when this push button is pressed. The supplied function is called when the push button is first pressed, and then repeated while the push button is pressed if auto-repeat is enabled.

Returns

None.

4.3.1.16 #define PushButtonFillColorPressedSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulPressFillColor = ulColor; \
}                                \
while(0)
```

Sets the fill color of a push button widget when it is pressed.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the push button when it is pressed.

This function changes the color used to fill the push button on the display when it is pressed. The display is not updated until the next paint request.

Returns

None.

4.3.1.17 #define PushButtonFillColorSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget;   \
    pW->ulFillColor = ulColor;       \
}                                \
while(0)
```

Sets the fill color of a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the push button.

This function changes the color used to fill the push button on the display. The display is not updated until the next paint request.

Returns

None.

4.3.1.18 #define PushButtonFillOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget;   \
    pW->ulStyle &= ~(PB_STYLE_FILL); \
}                                \
while(0)
```

Disables filling of a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function disables the filling of a push button widget. The display is not updated until the next paint request.

Returns

None.

4.3.1.19 #define PushButtonFillOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulStyle |= PB_STYLE_FILL; \
}                                \
while(0)
```

Enables filling of a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function enables the filling of a push button widget. The display is not updated until the next paint request.

Returns

None.

4.3.1.20 #define PushButtonFontSet(*pWidget*, *pFnt*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    const tFont *pF = pFnt;        \
    pW->pFont = pF;              \
}                                \
while(0)
```

Sets the font for a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
<i>pFnt</i>	is a pointer to the font to use to draw text on the push button.

This function changes the font used to draw text on the push button. The display is not updated until the next paint request.

Returns

None.

4.3.1.21 #define PushButtonImageOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulStyle &= ~(PB_STYLE_IMG); \
}                                \
while(0)
```

Disables the image on a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function disables the drawing of an image on a push button widget. The display is not updated until the next paint request.

Returns

None.

4.3.1.22 #define PushButtonImageOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulStyle |= PB_STYLE_IMG; \
}                                \
while(0)
```

Enables the image on a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function enables the drawing of an image on a push button widget. The display is not updated until the next paint request.

Returns

None.

4.3.1.23 #define PushButtonImagePressedSet(*pWidget*, *plmg*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    const unsigned char *pI = plmg; \
    pW->pucPressImage = pI;       \
}                                \
while(0)
```

Changes the image drawn on a push button widget when it is pressed.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to be modified.
<i>plmg</i>	is a pointer to the image to draw onto the push button when it is pressed.

This function changes the image that is drawn onto the push button when it is pressed. The display is not updated until the next paint request.

Returns

None.

4.3.1.24 #define PushButtonImageSet(*pWidget*, *plmg*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    const unsigned char *pI = plmg; \
    pW->pucImage = pI;           \
}                                \
while(0)
```

Changes the image drawn on a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to be modified.
<i>plmg</i>	is a pointer to the image to draw onto the push button.

This function changes the image that is drawn onto the push button. The display is not updated until the next paint request.

Returns

None.

4.3.1.25 #define PushButtonOutlineColorSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget;   \
    pW->ulOutlineColor = ulColor;    \
}                                \
while(0)
```

Sets the outline color of a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to outline the push button.

This function changes the color used to outline the push button on the display. The display is not updated until the next paint request.

Returns

None.

4.3.1.26 #define PushButtonOutlineOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget;   \
    pW->ulStyle &= ~(PB_STYLE_OUTLINE); \
}                                \
while(0)
```

Disables outlining of a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function disables the outlining of a push button widget. The display is not updated until the next paint request.

Returns

None.

4.3.1.27 #define PushButtonOutlineOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulStyle |= PB_STYLE_OUTLINE; \
}                                \
while(0)
```

Enables outlining of a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function enables the outlining of a push button widget. The display is not updated until the next paint request.

Returns

None.

4.3.1.28 #define PushButtonTextColorSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulTextColor = ulColor;     \
}                                \
while(0)
```

Sets the text color of a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw text on the push button.

This function changes the color used to draw text on the push button on the display. The display is not updated until the next paint request.

Returns

None.

4.3.1.29 #define PushButtonTextOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulStyle &= ~(PB_STYLE_TEXT); \
}                                \
while(0)
```

Disables the text on a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function disables the drawing of text on a push button widget. The display is not updated until the next paint request.

Returns

None.

4.3.1.30 #define PushButtonTextOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    pW->ulStyle |= PB_STYLE_TEXT; \
}                                \
while(0)
```

Enables the text on a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function enables the drawing of text on a push button widget. The display is not updated until the next paint request.

Returns

None.

4.3.1.31 #define PushButtonTextOpaqueOff(pWidget)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget;   \
    pW->ulStyle &= ~PB_STYLE_TEXT_OPAQUE; \
}                                \
while(0)
```

Disables opaque text on a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function disables the use of opaque text on this push button. When not using opaque text, only the foreground pixels of the text are drawn on the screen, allowing the previously drawn pixels (such as the push button image) to show through the text.

Returns

None.

4.3.1.32 #define PushButtonTextOpaqueOn(pWidget)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget;   \
    pW->ulStyle |= PB_STYLE_TEXT_OPAQUE; \
}                                \
while(0)
```

Enables opaque text on a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to modify.
----------------	---

This function enables the use of opaque text on this push button. When using opaque text, both the foreground and background pixels of the text are drawn on the screen, blocking out the previously drawn pixels.

Returns

None.

4.3.1.33 #define PushButtonTextSet(pWidget, pcTxt)**Value:**

```
do                                \
{                                \
    tPushButtonWidget *pW = pWidget; \
    const char *pcT = pcTxt;       \
    pW->pcText = pcT;            \
}                                \
while(0)
```

Changes the text drawn on a push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to be modified.
<i>pcTxt</i>	is a pointer to the text to draw onto the push button.

This function changes the text that is drawn onto the push button. The display is not updated until the next paint request.

Returns

None.

4.3.1.34 #define RectangularButton(sName, pParent, pNext, pChild, pDisplay, lX, lY, lWidth, lHeight, ulStyle, ulFillColor, ulPressFillColor, ulOutlineColor, ulTextColor, pFont, pcText, pucImage, pucPressImage, usAutoRepeatDelay, usAutoRepeatRate, pfnOnClick)**Value:**

```
tPushButtonWidget sName = \
    RectangularButtonStruct(pParent, pNext, pChild, pDisplay, lX, lY, \
    lWidth, lHeight, ulStyle, ulFillColor, \
    ulPressFillColor, ulOutlineColor, \
    ulTextColor, pFont, pcText, pucImage, \
    pucPressImage, usAutoRepeatDelay, \
    usAutoRepeatRate, pfnOnClick)
```

Declares an initialized variable containing a rectangular push button widget data structure.

Parameters

<i>sName</i>	is the name of the variable to be declared.
<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the upper left corner of the push button.
<i>IY</i>	is the Y coordinate of the upper left corner of the push button.
<i>IWidth</i>	is the width of the push button.
<i>IHeight</i>	is the height of the push button.
<i>ulStyle</i>	is the style to be applied to the push button.
<i>ulFillColor</i>	is the color used to fill in the push button.
<i>ulPressFill-Color</i>	is the color used to fill in the push button when it is pressed.
<i>ulOutline-Color</i>	is the color used to outline the push button.
<i>ulTextColor</i>	is the color used to draw text on the push button.
<i>pFont</i>	is a pointer to the font to be used to draw text on the push button.
<i>pcText</i>	is a pointer to the text to draw on this push button.
<i>puclImage</i>	is a pointer to the image to draw on this push button.
<i>pucPress-Image</i>	is a pointer to the image to draw on this push button when it is pressed.
<i>usAuto-Repeat-Delay</i>	is the delay before starting auto-repeat.
<i>usAuto-RepeatRate</i>	is the rate at which auto-repeat events are generated.
<i>pfnOnClick</i>	is a pointer to the function that is called when the push button is pressed.

This macro provides an initialized rectangular push button widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

ulStyle is the logical OR of the following:

- **PB_STYLE_OUTLINE** to indicate that the push button should be outlined.
- **PB_STYLE_FILL** to indicate that the push button should be filled.
- **PB_STYLE_TEXT** to indicate that the push button should have text drawn on it (using *pFont* and *pcText*).
- **PB_STYLE_IMG** to indicate that the push button should have an image drawn on it (using *puclImage*).
- **PB_STYLE_TEXT_OPAQUE** to indicate that the push button text should be drawn opaque (in other words, drawing the background pixels).
- **PB_STYLE_AUTO_REPEAT** to indicate that auto-repeat should be used.

- **PB_STYLE_RELEASE_NOTIFY** to indicate that the callback should be made when the button is released. If absent, the callback is called when the button is initially pressed.

Returns

Nothing; this is not a function.

```
4.3.1.35 #define RectangularButtonStruct( pParent, pNext, pChild, pDisplay, lX, lY,
    lWidth, lHeight, ulStyle, ulFillColor, ulPressFillColor, ulOutlineColor, ulTextColor,
    pFont, pcText, pucImage, pucPressImage, usAutoRepeatDelay, usAutoRepeatRate,
    pfnOnClick )
```

Value:

```
{
{
    sizeof(tPushButtonWidget),
    (tWidget *) (pParent),
    (tWidget *) (pNext),
    (tWidget *) (pChild),
    pDisplay,
    {
        lX,
        lY,
        (lX) + (lWidth) - 1,
        (lY) + (lHeight) - 1
    },
    RectangularButtonMsgProc
},
ulStyle,
ulFillColor,
ulPressFillColor,
ulOutlineColor,
ulTextColor,
pFont,
pcText,
pucImage,
pucPressImage,
usAutoRepeatDelay,
usAutoRepeatRate,
0,
pfnOnClick
}
```

Declares an initialized rectangular push button widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>lX</i>	is the X coordinate of the upper left corner of the push button.

<i>IY</i>	is the Y coordinate of the upper left corner of the push button.
<i>IWidth</i>	is the width of the push button.
<i>IHeight</i>	is the height of the push button.
<i>ulStyle</i>	is the style to be applied to the push button.
<i>ulFillColor</i>	is the color used to fill in the push button.
<i>ulPressFill-Color</i>	is the color used to fill in the push button when it is pressed.
<i>ulOutline-Color</i>	is the color used to outline the push button.
<i>ulTextColor</i>	is the color used to draw text on the push button.
<i>pFont</i>	is a pointer to the font to be used to draw text on the push button.
<i>pcText</i>	is a pointer to the text to draw on this push button.
<i>puclImage</i>	is a pointer to the image to draw on this push button.
<i>pucPress-Image</i>	is a pointer to the image to draw on this push button when it is pressed.
<i>usAuto-Repeat-Delay</i>	is the delay before starting auto-repeat.
<i>usAuto-RepeatRate</i>	is the rate at which auto-repeat events are generated.
<i>pfnOnClick</i>	is a pointer to the function that is called when the push button is pressed.

This macro provides an initialized rectangular push button widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
///!      tPushButtonWidget g_sPushButton = RectangularButtonStruct(...);  
///!
```

Or, in an array of variables:

```
///!      tPushButtonWidget g_psPushButtons [ ] =  
///!      {  
///!          RectangularButtonStruct(...),  
///!          RectangularButtonStruct(...)  
///!      };  
///!
```

ulStyle is the logical OR of the following:

- **PB_STYLE_OUTLINE** to indicate that the push button should be outlined.
- **PB_STYLE_FILL** to indicate that the push button should be filled.
- **PB_STYLE_TEXT** to indicate that the push button should have text drawn on it (using *pFont* and *pcText*).
- **PB_STYLE_IMG** to indicate that the push button should have an image drawn on it (using *puclImage*).

- **PB_STYLE_TEXT_OPAQUE** to indicate that the push button text should be drawn opaque (in other words, drawing the background pixels).
- **PB_STYLE_AUTO_REPEAT** to indicate that auto-repeat should be used.
- **PB_STYLE_RELEASE_NOTIFY** to indicate that the callback should be made when the button is released. If absent, the callback is called when the button is initially pressed.

Returns

Nothing; this is not a function.

4.3.2 Function Documentation

4.3.2.1 **void CircularButtonInit (tPushButtonWidget * pWidget, const tDisplay * pDisplay, unsigned IX, unsigned IY, unsigned IR)**

Initializes a circular push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the upper left corner of the push button.
<i>IY</i>	is the Y coordinate of the upper left corner of the push button.
<i>IR</i>	is the radius of the push button.

This function initializes the provided push button widget so that it will be a circular push button.

Returns

None.

4.3.2.2 **int CircularButtonMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)**

Handles messages for a circular push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this push button widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

4.3.2.3 void RectangularButtonInit (tPushButtonWidget * pWidget, const tDisplay * pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

Initializes a rectangular push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the upper left corner of the push button.
<i>IY</i>	is the Y coordinate of the upper left corner of the push button.
<i>IWidth</i>	is the width of the push button.
<i>IHeight</i>	is the height of the push button.

This function initializes the provided push button widget so that it will be a rectangular push button.

Returns

None.

4.3.2.4 int RectangularButtonMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a rectangular push button widget.

Parameters

<i>pWidget</i>	is a pointer to the push button widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this push button widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

4.4 Radiobutton_api

Data Structures

- struct [tRadioButtonWidget](#)
The structure that describes a radio button widget.

Defines

- #define [RB_STYLE_OUTLINE](#) 0x0001
This flag indicates that the radio button should be outlined.
- #define [RB_STYLE_FILL](#) 0x0002
This flag indicates that the radio button should be filled.
- #define [RB_STYLE_TEXT](#) 0x0004
This flag indicates that the radio button should have text drawn on it.
- #define [RB_STYLE_IMG](#) 0x0008
This flag indicates that the radio button should have an image drawn on it.
- #define [RB_STYLE_TEXT_OPAQUE](#) 0x0010
- #define [RB_STYLE_SELECTED](#) 0x0020
This flag indicates that the radio button is selected.
- #define [RadioButtonStruct](#)(pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, usStyle, usCircleSize, ulFillColor,ulOutlineColor, ulTextColor, pFont, pcText,puclImage, pfnOnChange)
- #define [RadioButton](#)(sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, usStyle, usCircleSize, ulFillColor,ulOutlineColor, ulTextColor, pFont, pcText,puclImage,pfnOnChange)
- #define [RadioButtonCircleSizeSet](#)(pWidget, usSize)
- #define [RadioButtonCallbackSet](#)(pWidget, pfnOnChg)
- #define [RadioButtonFillColorSet](#)(pWidget, ulColor)
- #define [RadioButtonFillOff](#)(pWidget)
- #define [RadioButtonFillOn](#)(pWidget)
- #define [RadioButtonFontSet](#)(pWidget, pFnt)
- #define [RadioButtonImageSet](#)(pWidget, plmg)
- #define [RadioButtonImageOff](#)(pWidget)
- #define [RadioButtonImageOn](#)(pWidget)
- #define [RadioButtonOutlineColorSet](#)(pWidget, ulColor)
- #define [RadioButtonOutlineOff](#)(pWidget)
- #define [RadioButtonOutlineOn](#)(pWidget)
- #define [RadioButtonTextColorSet](#)(pWidget, ulColor)
- #define [RadioButtonTextOff](#)(pWidget)
- #define [RadioButtonTextOn](#)(pWidget)
- #define [RadioButtonTextOpaqueOff](#)(pWidget)
- #define [RadioButtonTextOpaqueOn](#)(pWidget)
- #define [RadioButtonTextSet](#)(pWidget, pcTxt)

Functions

- int `RadioButtonMsgProc` (`tWidget` *`pWidget`, unsigned `ulMsg`, unsigned `ulParam1`, unsigned `ulParam2`)
- void `RadioButtonInit` (`tRadioButtonWidget` *`pWidget`, const `tDisplay` *`pDisplay`, unsigned `IX`, unsigned `IY`, unsigned `IWidth`, unsigned `IHeight`)

4.4.1 Define Documentation

```
4.4.1.1 #define RadioButton( sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth,
    IHeight, usStyle, usCircleSize, ulFillColor, ulOutlineColor, ulTextColor, pFont,
    pcText, pucImage, pfnOnChange )
```

Value:

```
tRadioButtonWidget sName =
    RadioButtonStruct(pParent, pNext, pChild, pDisplay, lX, lY,
                      lWidth, lHeight, usStyle, usCircleSize,
                      ulFillColor, ulOutlineColor, ulTextColor,
                      pFont, pcText, pucImage, pfnOnChange)
```

Declares an initialized variable containing a radio button widget data structure.

Parameters

<code>sName</code>	is the name of the variable to be declared.
<code>pParent</code>	is a pointer to the parent widget.
<code>pNext</code>	is a pointer to the sibling widget.
<code>pChild</code>	is a pointer to the first child widget.
<code>pDisplay</code>	is a pointer to the display on which to draw the radio button.
<code>IX</code>	is the X coordinate of the upper left corner of the radio button.
<code>IY</code>	is the Y coordinate of the upper left corner of the radio button.
<code>IWidth</code>	is the width of the radio button.
<code>IHeight</code>	is the height of the radio button.
<code>usStyle</code>	is the style to be applied to this radio button.
<code>usCircleSize</code>	is the size of the circle that is filled.
<code>ulFillColor</code>	is the color used to fill in the radio button.
<code>ulOutline-Color</code>	is the color used to outline the radio button.
<code>ulTextColor</code>	is the color used to draw text on the radio button.
<code>pFont</code>	is a pointer to the font to be used to draw text on the radio button.
<code>pcText</code>	is a pointer to the text to draw on this radio button.
<code>pucImage</code>	is a pointer to the image to draw on this radio button.
<code>pfnOn-Change</code>	is a pointer to the function that is called when the radio button is pressed.

This macro provides an initialized radio button widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

usStyle is the logical OR of the following:

- **RB_STYLE_OUTLINE** to indicate that the radio button should be outlined.
- **RB_STYLE_FILL** to indicate that the radio button should be filled.
- **RB_STYLE_TEXT** to indicate that the radio button should have text drawn on it (using *pFont* and *pcText*).
- **RB_STYLE_IMG** to indicate that the radio button should have an image drawn on it (using *puclImage*).
- **RB_STYLE_TEXT_OPAQUE** to indicate that the radio button text should be drawn opaque (in other words, drawing the background pixels).
- **RB_STYLE_SELECTED** to indicate that the radio button is selected.

Returns

Nothing; this is not a function.

4.4.1.2 #define RadioButtonCallbackSet(*pWidget*, *pfnOnChg*)

Value:

```
do                                \
{                                \
    tRadioButtonWidget *pW = pWidget;   \
    pW->pfnOnChange = pfnOnChg;       \
}                                \
while(0)
```

Sets the function to call when this radio button widget is toggled.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
<i>pfnOnChg</i>	is a pointer to the function to call.

This function sets the function to be called when this radio button is toggled.

Returns

None.

4.4.1.3 #define RadioButtonCircleSizeSet(*pWidget*, *usSize*)

Value:

```
do                                \

```

```

{
    tRadioButtonWidget *pW = pWidget;      \
    pW->usCircleSize = usSize;           \
}
while(0)

```

Sets size of the circle to be filled.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
<i>usSize</i>	is the size of the circle, in pixels.

This function sets the size of the circle that is drawn as part of the radio button.

Returns

None.

4.4.1.4 #define RadioButtonFillColorSet(*pWidget*, *ulColor*)

Value:

```

do                                \
{
    tRadioButtonWidget *pW = pWidget;      \
    pW->ulFillColor = ulColor;           \
}
while(0)

```

Sets the fill color of a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the radio button.

This function changes the color used to fill the radio button on the display. The display is not updated until the next paint request.

Returns

None.

4.4.1.5 #define RadioButtonFillOff(*pWidget*)

Value:

```

do                                \
{

```

```

    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle &= ~(RB_STYLE_FILL); \
}
while(0)

```

Disables filling of a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function disables the filling of a radio button widget. The display is not updated until the next paint request.

Returns

None.

4.4.1.6 #define RadioButtonFillOn(*pWidget*)

Value:

```

do \
{ \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle |= RB_STYLE_FILL; \
} \
while(0)

```

Enables filling of a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function enables the filling of a radio button widget. The display is not updated until the next paint request.

Returns

None.

4.4.1.7 #define RadioButtonFontSet(*pWidget*, *pFnt*)

Value:

```

do \
{ \
    tRadioButtonWidget *pW = pWidget; \
    const tFont *pF = pFnt; \
}

```

```

    pW->pFont = pF;           \
}                           \
while(0)

```

Sets the font for a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
<i>pFnt</i>	is a pointer to the font to use to draw text on the radio button.

This function changes the font used to draw text on the radio button. The display is not updated until the next paint request.

Returns

None.

4.4.1.8 #define RadioButtonImageOff(*pWidget*)

Value:

```

do                                \
{                               \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle &= ~RB_STYLE_IMG; \
}                               \
while(0)

```

Disables the image on a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function disables the drawing of an image on a radio button widget. The display is not updated until the next paint request.

Returns

None.

4.4.1.9 #define RadioButtonImageOn(*pWidget*)

Value:

```

do                                \
{                               \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle |= RB_STYLE_IMG; \
}

```

```

}
while(0)                                \

```

Enables the image on a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function enables the drawing of an image on a radio button widget. The display is not updated until the next paint request.

Returns

None.

4.4.1.10 #define RadioButtonImageSet(*pWidget*, *plmg*)

Value:

```

do                                         \
{
    tRadioButtonWidget *pW = pWidget; \
    const unsigned char *pI = plmg;  \
    pW->pucImage = pI;           \
}                                         \
while(0)

```

Changes the image drawn on a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to be modified.
<i>plmg</i>	is a pointer to the image to draw onto the radio button.

This function changes the image that is drawn onto the radio button. The display is not updated until the next paint request.

Returns

None.

4.4.1.11 #define RadioButtonOutlineColorSet(*pWidget*, *ulColor*)

Value:

```

do                                         \
{
    tRadioButtonWidget *pW = pWidget; \
    pW->ulOutlineColor = ulColor; \
}

```

```
    }                                \
    while(0)
```

Sets the outline color of a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to outline the radio button.

This function changes the color used to outline the radio button on the display. The display is not updated until the next paint request.

Returns

None.

4.4.1.12 #define RadioButtonOutlineOff(*pWidget*)

Value:

```
do                                \
{                                \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle &= ~ (RB_STYLE_OUTLINE); \
}                                \
while(0)
```

Disables outlining of a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function disables the outlining of a radio button widget. The display is not updated until the next paint request.

Returns

None.

4.4.1.13 #define RadioButtonOutlineOn(*pWidget*)

Value:

```
do                                \
{                                \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle |= RB_STYLE_OUTLINE; \
}                                \
while(0)
```

Enables outlining of a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function enables the outlining of a radio button widget. The display is not updated until the next paint request.

Returns

None.

```
4.4.1.14 #define RadioButtonStruct( pParent, pNext, pChild, pDisplay, IX, IY, lWidth,
    lHeight, usStyle, usCircleSize, ulFillColor, ulOutlineColor, ulTextColor, pFont,
    pcText, pucImage, pfnOnChange )
```

Value:

```
{
    {
        sizeof(tRadioButtonWidget),
        (tWidget *) (pParent),
        (tWidget *) (pNext),
        (tWidget *) (pChild),
        pDisplay,
        {
            IX,
            IY,
            (IX) + (lWidth) - 1,
            (IY) + (lHeight) - 1
        },
        RadioButtonMsgProc
    },
    usStyle,
    usCircleSize,
    ulFillColor,
    ulOutlineColor,
    ulTextColor,
    pFont,
    pcText,
    pucImage,
    pfnOnChange
}
```

Declares an initialized radio button widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the radio button.
<i>IX</i>	is the X coordinate of the upper left corner of the radio button.

<i>IY</i>	is the Y coordinate of the upper left corner of the radio button.
<i>IWidth</i>	is the width of the radio button.
<i>IHeight</i>	is the height of the radio button.
<i>usStyle</i>	is the style to be applied to this radio button.
<i>usCircleSize</i>	is the size of the circle that is filled.
<i>ulFillColor</i>	is the color used to fill in the radio button.
<i>ulOutline-Color</i>	is the color used to outline the radio button.
<i>ulTextColor</i>	is the color used to draw text on the radio button.
<i>pFont</i>	is a pointer to the font to be used to draw text on the radio button.
<i>pcText</i>	is a pointer to the text to draw on this radio button.
<i>puclImage</i>	is a pointer to the image to draw on this radio button.
<i>pfnOn-Change</i>	is a pointer to the function that is called when the radio button is pressed.

This macro provides an initialized radio button widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
///!      tRadioButtonWidget g_sRadioButton = RadioButtonStruct(...);
///!
```

Or, in an array of variables:

```
///!      tRadioButtonWidget g_psButtons[ ] =
///!      {
///!          RadioButtonStruct(...),
///!          RadioButtonStruct(...)
///!      };
///!
```

usStyle is the logical OR of the following:

- **RB_STYLE_OUTLINE** to indicate that the radio button should be outlined.
- **RB_STYLE_FILL** to indicate that the radio button should be filled.
- **RB_STYLE_TEXT** to indicate that the radio button should have text drawn on it (using *pFont* and *pcText*).
- **RB_STYLE_IMG** to indicate that the radio button should have an image drawn on it (using *puclImage*).
- **RB_STYLE_TEXT_OPAQUE** to indicate that the radio button text should be drawn opaque (in other words, drawing the background pixels).
- **RB_STYLE_SELECTED** to indicate that the radio button is selected.

Returns

Nothing; this is not a function.

4.4.1.15 #define RadioButtonTextColorSet(*pWidget*, *ulColor*)

Value:

```
do                                \
{                                \
    tRadioButtonWidget *pW = pWidget; \
    pW->ulTextColor = ulColor;     \
}                                \
while(0)
```

Sets the text color of a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw text on the radio button.

This function changes the color used to draw text on the radio button on the display. The display is not updated until the next paint request.

Returns

None.

4.4.1.16 #define RadioButtonTextOff(*pWidget*)

Value:

```
do                                \
{                                \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle &= ~(RB_STYLE_TEXT); \
}                                \
while(0)
```

Disables the text on a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function disables the drawing of text on a radio button widget. The display is not updated until the next paint request.

Returns

None.

4.4.1.17 #define RadioButtonTextOn(pWidget)**Value:**

```
do                                \
{                                \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle |= RB_STYLE_TEXT;   \
}                                \
while(0)
```

Enables the text on a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function enables the drawing of text on a radio button widget. The display is not updated until the next paint request.

Returns

None.

4.4.1.18 #define RadioButtonTextOpaqueOff(pWidget)**Value:**

```
do                                \
{                                \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle &= ~(RB_STYLE_TEXT_OPAQUE); \
}                                \
while(0)
```

Disables opaque text on a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function disables the use of opaque text on this radio button. When not using opaque text, only the foreground pixels of the text are drawn on the screen, allowing the previously drawn pixels (such as the radio button image) to show through the text.

Returns

None.

4.4.1.19 #define RadioButtonTextOpaqueOn(pWidget)

Value:

```
do                                \
{                                 \
    tRadioButtonWidget *pW = pWidget; \
    pW->usStyle |= RB_STYLE_TEXT_OPAQUE; \
}                                 \
while(0)
```

Enables opaque text on a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to modify.
----------------	--

This function enables the use of opaque text on this radio button. When using opaque text, both the foreground and background pixels of the text are drawn on the screen, blocking out the previously drawn pixels.

Returns

None.

4.4.1.20 #define RadioButtonTextSet(pWidget, pcTxt)

Value:

```
do                                \
{                                 \
    tRadioButtonWidget *pW = pWidget; \
    const char *pcT = pcTxt;        \
    pW->pcText = pcT;             \
}                                 \
while(0)
```

Changes the text drawn on a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to be modified.
<i>pcTxt</i>	is a pointer to the text to draw onto the radio button.

This function changes the text that is drawn onto the radio button. The display is not updated until the next paint request.

Returns

None.

4.4.1.21 #define RB_STYLE_FILL 0x0002

This flag indicates that the radio button should be filled.

4.4.1.22 #define RB_STYLE_IMG 0x0008

This flag indicates that the radio button should have an image drawn on it.

4.4.1.23 #define RB_STYLE_OUTLINE 0x0001

This flag indicates that the radio button should be outlined.

4.4.1.24 #define RB_STYLE_SELECTED 0x0020

This flag indicates that the radio button is selected.

4.4.1.25 #define RB_STYLE_TEXT 0x0004

This flag indicates that the radio button should have text drawn on it.

4.4.1.26 #define RB_STYLE_TEXT_OPAQUE 0x0010

This flag indicates that the radio button text should be drawn opaque (in other words, drawing the background pixels as well as the foreground pixels).

4.4.2 Function Documentation

4.4.2.1 void RadioButtonInit (tRadioButtonWidget * pWidget, const tDisplay * pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

Initializes a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the upper left corner of the radio button.
<i>IY</i>	is the Y coordinate of the upper left corner of the radio button.
<i>IWidth</i>	is the width of the radio button.
<i>IHeight</i>	is the height of the radio button.

This function initializes the provided radio button widget.

Returns

None.

4.4.2.2 int RadioButtonMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a radio button widget.

Parameters

<i>pWidget</i>	is a pointer to the radio button widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this radio button widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

Chapter 5

Data Structure Documentation

5.1 __Widget Struct Reference

```
#include <widget.h>
```

Data Fields

- long **ISize**
- struct **__Widget * pParent**

A pointer to this widget's parent widget.

- struct **__Widget * pNext**

A pointer to this widget's first sibling widget.

- struct **__Widget * pChild**

A pointer to this widget's first child widget.

- const **tDisplay * pDisplay**

A pointer to the display on which this widget resides.

- **tRectangle sPosition**

The rectangle that encloses this widget.

- int(* **pfnMsgProc**)(struct **__Widget** *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2)

The procedure that handles messages sent to this widget.

5.1.1 Detailed Description

The structure that describes a generic widget. This structure is the base “class” for all other widgets.

5.1.2 Field Documentation

5.1.2.1 long __Widget::lSize

The size of this structure. This will be the size of the full structure, not just the generic widget subset.

5.1.2.2 struct __Widget* __Widget::pChild

A pointer to this widget's first child widget.

5.1.2.3 const tDisplay* __Widget::pDisplay

A pointer to the display on which this widget resides.

5.1.2.4 int(* __Widget::pfnMsgProc)(struct __Widget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2)

The procedure that handles messages sent to this widget.

5.1.2.5 struct __Widget* __Widget::pNext

A pointer to this widget's first sibling widget.

5.1.2.6 struct __Widget* __Widget::pParent

A pointer to this widget's parent widget.

5.1.2.7 tRectangle __Widget::sPosition

The rectangle that encloses this widget.

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

- include/widget.h

5.2 tCanvasWidget Struct Reference

The structure that describes a canvas widget.

```
#include <canvas.h>
```

Data Fields

- **tWidget sBase**
The generic widget information.
- unsigned **ulStyle**
- unsigned long **ulFillColor**
- unsigned long **ulOutlineColor**
- unsigned long **ulTextColor**
- const **tFont * pFont**
- const char * **pcText**
- const unsigned char * **puImage**
- void(* **pfnOnPaint**)(tWidget *pWidget, tContext *pContext)

5.2.1 Detailed Description

The structure that describes a canvas widget.

5.2.2 Field Documentation

5.2.2.1 const char* tCanvasWidget::pcText

A pointer to the text to draw on this canvas, if CANVAS_STYLE_TEXT is selected.

5.2.2.2 void(* tCanvasWidget::pfnOnPaint)(tWidget *pWidget, tContext *pContext)

A pointer to the application-supplied drawing function used to draw onto this canvas, if CANVAS_STYLE_APP_DRAWN is selected.

5.2.2.3 const tFont* tCanvasWidget::pFont

A pointer to the font used to render the canvas text, if CANVAS_STYLE_TEXT is selected.

5.2.2.4 const unsigned char* tCanvasWidget::puImage

A pointer to the image to be drawn onto this canvas, if CANVAS_STYLE_IMG is selected.

5.2.2.5 tWidget tCanvasWidget::sBase

The generic widget information.

5.2.2.6 unsigned long tCanvasWidget::ulFillColor

The 24-bit RGB color used to fill this canvas, if CANVAS_STYLE_FILL is selected, and to use as the background color if CANVAS_STYLE_TEXT_OPAQUE is selected.

5.2.2.7 unsigned long tCanvasWidget::ulOutlineColor

The 24-bit RGB color used to outline this canvas, if CANVAS_STYLE_OUTLINE is selected.

5.2.2.8 unsigned tCanvasWidget::ulStyle

The style for this widget. This is a set of flags defined by CANVAS_STYLE_xxx.

5.2.2.9 unsigned long tCanvasWidget::ulTextColor

The 24-bit RGB color used to draw text on this canvas, if CANVAS_STYLE_TEXT is selected.

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

- include/[canvas.h](#)

5.3 tCheckBoxWidget Struct Reference

The structure that describes a check box widget.

```
#include <checkbox.h>
```

Data Fields

- [tWidget sBase](#)

The generic widget information.

- unsigned short [usStyle](#)
- unsigned short [usBoxSize](#)
- unsigned long [ulFillColor](#)
- unsigned long [ulOutlineColor](#)
- unsigned long [ulTextColor](#)
- const [tFont * pFont](#)

The font used to draw the check box text, if CB_STYLE_TEXT is selected.

- const char * [pcText](#)
- const unsigned char * [puImage](#)
- void(* [pfnOnChange](#))(tWidget *pWidget, unsigned bSelected)

5.3.1 Detailed Description

The structure that describes a check box widget.

5.3.2 Field Documentation

5.3.2.1 const char* tCheckBoxWidget::pcText

A pointer to the text to draw on this check box, if CB_STYLE_TEXT is selected.

5.3.2.2 void(* tCheckBoxWidget::pfnOnChange)(tWidget *pWidget, unsigned bSelected)

A pointer to the function to be called when the check box is pressed. This function is called when the state of the check box is changed.

5.3.2.3 const tFont* tCheckBoxWidget::pFont

The font used to draw the check box text, if CB_STYLE_TEXT is selected.

5.3.2.4 const unsigned char* tCheckBoxWidget::puImage

A pointer to the image to be drawn onto this check box, if CB_STYLE_IMG is selected.

5.3.2.5 tWidget tCheckBoxWidget::sBase

The generic widget information.

5.3.2.6 unsigned long tCheckBoxWidget::ulFillColor

The 24-bit RGB color used to fill this check box, if CB_STYLE_FILL is selected, and to use as the background color if CB_STYLE_TEXT_OPAQUE is selected.

5.3.2.7 unsigned long tCheckBoxWidget::ulOutlineColor

The 24-bit RGB color used to outline this check box, if CB_STYLE_OUTLINE is selected.

5.3.2.8 unsigned long tCheckBoxWidget::ulTextColor

The 24-bit RGB color used to draw text on this check box, if CB_STYLE_TEXT is selected.

5.3.2.9 unsigned short tCheckBoxWidget::usBoxSize

The size of the check box itself, not including the text and/or image that accompanies it (in other words, the size of the actual box that is checked or unchecked).

5.3.2.10 unsigned short tCheckBoxWidget::usStyle

The style for this check box. This is a set of flags defined by CB_STYLE_xxx.

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

- include/checkbox.h

5.4 tContainerWidget Struct Reference

The structure that describes a container widget.

```
#include <container.h>
```

Data Fields

- **tWidget sBase**
The generic widget information.
- unsigned **ulStyle**
- unsigned long **ulFillColor**
- unsigned long **ulOutlineColor**
- unsigned long **ulTextColor**
- const **tFont * pFont**
- const char * **pcText**

5.4.1 Detailed Description

The structure that describes a container widget.

5.4.2 Field Documentation

5.4.2.1 const char* tContainerWidget::pcText

A pointer to the text to draw on this container widget, if CTR_STYLE_TEXT is selected.

5.4.2.2 const tFont* tContainerWidget::pFont

A pointer to the font used to render the container text, if CTR_STYLE_TEXT is selected.

5.4.2.3 tWidget tContainerWidget::sBase

The generic widget information.

5.4.2.4 unsigned long tContainerWidget::ulFillColor

The 24-bit RGB color used to fill this container widget, if CTR_STYLE_FILL is selected, and to use as the background color if CTR_STYLE_TEXT_OPAQUE is selected.

5.4.2.5 unsigned long tContainerWidget::ulOutlineColor

The 24-bit RGB color used to outline this container widget, if CTR_STYLE_OUTLINE is selected.

5.4.2.6 unsigned tContainerWidget::ulStyle

The style for this widget. This is a set of flags defined by CTR_STYLE_xxx.

5.4.2.7 unsigned long tContainerWidget::ulTextColor

The 24-bit RGB color used to draw text on this container widget, if CTR_STYLE_TEXT is selected.

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

- include/container.h

5.5 tContext Struct Reference

```
#include <grlib.h>
```

Data Fields

- long [lSize](#)
The size of this structure.
- const [tDisplay * pDisplay](#)
The screen onto which drawing operations are performed.
- [tRectangle sClipRegion](#)
The clipping region to be used when drawing onto the screen.
- unsigned long [ulForeground](#)
The color used to draw primitives onto the screen.
- unsigned long [ulBackground](#)
The background color used to draw primitives onto the screen.

- const [tFont](#) * [pFont](#)

The font used to render text onto the screen.

5.5.1 Detailed Description

This structure defines a drawing context to be used to draw onto the screen. Multiple drawing contexts may exist at any time.

5.5.2 Field Documentation

5.5.2.1 long [tContext::lSize](#)

The size of this structure.

5.5.2.2 const [tDisplay](#)* [tContext::pDisplay](#)

The screen onto which drawing operations are performed.

5.5.2.3 const [tFont](#)* [tContext::pFont](#)

The font used to render text onto the screen.

5.5.2.4 [tRectangle](#) [tContext::sClipRegion](#)

The clipping region to be used when drawing onto the screen.

5.5.2.5 unsigned long [tContext::ulBackground](#)

The background color used to draw primitives onto the screen.

5.5.2.6 unsigned long [tContext::ulForeground](#)

The color used to draw primitives onto the screen.

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

- include/[grlib.h](#)

5.6 [tDisplay](#) Struct Reference

This structure defines the characteristics of a display driver.

```
#include <grlib.h>
```

Data Fields

- long **lSize**
The size of this structure.
- void * **pvDisplayData**
A pointer to display driver-specific data.
- unsigned short **usWidth**
The width of this display.
- unsigned short **usHeight**
The height of this display.
- void(*) **pfnPixelDraw**)(void ***pvDisplayData**, unsigned IX, unsigned IY, unsigned ulValue)
A pointer to the function to draw a pixel on this display.
- void(*) **pfnPixelDrawMultiple**)(void ***pvDisplayData**, unsigned IX, unsigned IY, unsigned IX0, unsigned ICount, unsigned IBPP, const unsigned char *pucData, const unsigned char *pucPalette)
A pointer to the function to draw multiple pixels on this display.
- void(*) **pfnLineDrawH**)(void ***pvDisplayData**, unsigned IX1, unsigned IX2, unsigned IY, unsigned ulValue)
A pointer to the function to draw a horizontal line on this display.
- void(*) **pfnLineDrawV**)(void ***pvDisplayData**, unsigned IX, unsigned IY1, unsigned IY2, unsigned ulValue)
A pointer to the function to draw a vertical line on this display.
- void(*) **pfnRectFill**)(void ***pvDisplayData**, const **tRectangle** *pRect, unsigned ulValue)
A pointer to the function to draw a filled rectangle on this display.
- unsigned(* **pfnColorTranslate**)(void ***pvDisplayData**, unsigned long ulValue)
- void(* **pfnFlush**)(void ***pvDisplayData**)

5.6.1 Detailed Description

This structure defines the characteristics of a display driver.

5.6.2 Field Documentation

5.6.2.1 long tDisplay::lSize

The size of this structure.

5.6.2.2 unsigned(* tDisplay::pfnColorTranslate)(void *pvDisplayData, unsigned long ulValue)

A pointer to the function to translate 24-bit RGB colors to display-specific colors.

5.6.2.3 void(* tDisplay::pfnFlush)(void *pvDisplayData)

A pointer to the function to flush any cached drawing operations on this display.

5.6.2.4 void(* tDisplay::pfnLineDrawH)(void *pvDisplayData, unsigned lX1, unsigned lX2, unsigned lY, unsigned ulValue)

A pointer to the function to draw a horizontal line on this display.

5.6.2.5 void(* tDisplay::pfnLineDrawV)(void *pvDisplayData, unsigned lX, unsigned lY1, unsigned lY2, unsigned ulValue)

A pointer to the function to draw a vertical line on this display.

5.6.2.6 void(* tDisplay::pfnPixelDraw)(void *pvDisplayData, unsigned lX, unsigned lY, unsigned ulValue)

A pointer to the function to draw a pixel on this display.

5.6.2.7 void(* tDisplay::pfnPixelDrawMultiple)(void *pvDisplayData, unsigned lX, unsigned lY, unsigned lX0, unsigned lCount, unsigned lBPP, const unsigned char *pucData, const unsigned char *pucPalette)

A pointer to the function to draw multiple pixels on this display.

5.6.2.8 void(* tDisplay::pfnRectFill)(void *pvDisplayData, const tRectangle *pRect, unsigned ulValue)

A pointer to the function to draw a filled rectangle on this display.

5.6.2.9 void* tDisplay::pvDisplayData

A pointer to display driver-specific data.

5.6.2.10 unsigned short tDisplay::usHeight

The height of this display.

5.6.2.11 unsigned short tDisplay::usWidth

The width of this display.

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

- include/grlib.h

5.7 tFont Struct Reference

This structure describes a font used for drawing text onto the screen.

```
#include <grlib.h>
```

Data Fields

- unsigned char **ucFormat**
- unsigned char **ucMaxWidth**
- unsigned char **ucHeight**
- unsigned char **ucBaseline**
- unsigned short **pusOffset** [96]

The offset within pucData to the data for each character in the font.

- const unsigned char * **pucData**

A pointer to the data for the font.

5.7.1 Detailed Description

This structure describes a font used for drawing text onto the screen.

5.7.2 Field Documentation

5.7.2.1 const unsigned char* tFont::pucData

A pointer to the data for the font.

5.7.2.2 unsigned short tFont::pusOffset[96]

The offset within pucData to the data for each character in the font.

5.7.2.3 unsigned char tFont::ucBaseline

The offset between the top of the character cell and the baseline of the glyph. The baseline is the bottom row of a capital letter, below which only the descenders of the lower case letters occur.

5.7.2.4 unsigned char tFont::ucFormat

The format of the font. Can be one of FONT_FMT_UNCOMPRESSED or FONT_FMT_PIXEL_RLE.

5.7.2.5 `unsigned char tFont::ucHeight`

The height of the character cell; this may be taller than the font data for the characters (to provide inter-line spacing).

5.7.2.6 `unsigned char tFont::ucMaxWidth`

The maximum width of a character; this is the width of the widest character in the font, though any individual character may be narrower than this width.

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

- include/grlib.h

5.8 `tImageButtonWidget` Struct Reference

The structure that describes a image button widget.

```
#include <imgbutton.h>
```

Data Fields

- `tWidget sBase`
The generic widget information.
- `unsigned ulStyle`
- `unsigned long ulForegroundColor`
- `unsigned long ulPressedColor`
- `unsigned long ulBackgroundColor`
- `const tFont * pFont`
- `const char * pcText`
- `const unsigned char * puImage`
- `const unsigned char * pucPressImage`
- `const unsigned char * pucKeycapImage`
- `short sXOffset`
- `short sYOffset`
- `unsigned short usAutoRepeatDelay`
- `unsigned short usAutoRepeatRate`
- `unsigned ulAutoRepeatCount`
- `void(* pfnOnClick)(tWidget *pWidget)`

5.8.1 Detailed Description

The structure that describes a image button widget.

5.8.2 Field Documentation

5.8.2.1 const char* tImageButtonWidget::pcText

A pointer to the text to draw on this push button, if IB_STYLE_TEXT is selected.

5.8.2.2 void(* tImageButtonWidget::pfnOnClick)(tWidget *pWidget)

A pointer to the function to be called when the button is pressed. This is repeatedly called when IB_STYLE_AUTO_REPEAT is selected.

5.8.2.3 const tFont* tImageButtonWidget::pFont

A pointer to the font used to render the button text, if IB_STYLE_TEXT is selected.

5.8.2.4 const unsigned char* tImageButtonWidget::puclImage

A pointer to the image to be drawn onto this image button, if IB_STYLE_IMG is selected.

5.8.2.5 const unsigned char* tImageButtonWidget::pucKeycapImage

A pointer to the image to be drawn above the background image for the button.

5.8.2.6 const unsigned char* tImageButtonWidget::pucPressImage

A pointer to the image to be drawn onto this image button when it is pressed.

5.8.2.7 tWidget tImageButtonWidget::sBase

The generic widget information.

5.8.2.8 short tImageButtonWidget::sXOffset

The number of pixels to move the keycap image horizontally when the button is drawn in its pressed state.

5.8.2.9 short tImageButtonWidget::sYOffset

The number of pixels to move the keycap image vertically when the button is drawn in its pressed state.

5.8.2.10 unsigned tImageButtonWidget::ulAutoRepeatCount

The number of pointer events that have occurred. This is used when IB_STYLE_AUTO_REPEAT is selected to generate the auto-repeat events.

5.8.2.11 unsigned long tImageButtonWidget::ulBackgroundColor

The color to use for background pixels when the button is released and a 1bpp image is in use. This value is ignored for all other image bit depths. If IB_STYLE_FILL is specified, this is also the color that will be used to fill the widget when it is in the unpressed state.

5.8.2.12 unsigned long tImageButtonWidget::ulForegroundColor

The color to use for foreground pixels when a 1bpp image or text is in use. This value is ignored for all other image bit depths.

5.8.2.13 unsigned long tImageButtonWidget::ulPressedColor

The color to use for background pixels when the button is pressed and a 1bpp image is in use. This value is ignored for all other image bit depths. If IB_STYLE_FILL is specified, this is also the color that will be used to fill the widget when it is in the pressed state.

5.8.2.14 unsigned tImageButtonWidget::ulStyle

The style for this widget. This is a set of flags defined by IB_STYLE_xxx.

5.8.2.15 unsigned short tImageButtonWidget::usAutoRepeatDelay

The number of pointer events to delay before starting to auto-repeat, if IB_STYLE_AUTO_REPEAT is selected. The amount of time to which this corresponds is dependent upon the rate at which pointer events are generated by the pointer driver.

5.8.2.16 unsigned short tImageButtonWidget::usAutoRepeatRate

The number of pointer events between button presses generated by the auto-repeat function, if IB_STYLE_AUTO_REPEAT is selected. The amount of time to which this corresponds is dependent up on the rate at which pointer events are generated by the pointer driver.

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

- include/[imgbutton.h](#)

5.9 tListBoxWidget Struct Reference

The structure that describes a listbox widget.

```
#include <listbox.h>
```

Data Fields

- **tWidget sBase**
The generic widget information.
- **unsigned ulStyle**
- **unsigned long ulBackgroundColor**
The 24-bit RGB color used as the background for the listbox.
- **unsigned long ulSelectedBackgroundColor**
- **unsigned long ulTextColor**
The 24-bit RGB color used to draw text on this listbox.
- **unsigned long ulSelectedTextColor**
The 24-bit RGB color used to draw the selected text on this listbox.
- **unsigned long ulOutlineColor**
- **const tFont * pFont**
A pointer to the font used to render the listbox text.
- **const char ** pccText**
- **unsigned short usMaxEntries**
The number of elements in the array pointed to by pccText.
- **unsigned short usPopulated**
- **short sSelected**
- **unsigned short usStartEntry**
- **unsigned short usOldestEntry**
- **unsigned short usScrolled**
- **unsigned IPointerY**
- **void(* pfnOnChange)(tWidget *pWidget, short sSelIndex)**

5.9.1 Detailed Description

The structure that describes a listbox widget.

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

- include/[listbox.h](#)

5.10 tPushButtonWidget Struct Reference

The structure that describes a push button widget.

```
#include <pushbutton.h>
```

Data Fields

- `tWidget sBase`
The generic widget information.
- `unsigned ulStyle`
- `unsigned long ulFillColor`
- `unsigned long ulPressFillColor`
- `unsigned long ulOutlineColor`
- `unsigned long ulTextColor`
- `const tFont * pFont`
- `const char * pcText`
- `const unsigned char * puImage`
- `const unsigned char * pucPressImage`
- `unsigned short usAutoRepeatDelay`
- `unsigned short usAutoRepeatRate`
- `unsigned ulAutoRepeatCount`
- `void(* pfnOnClick)(tWidget *pWidget)`

5.10.1 Detailed Description

The structure that describes a push button widget.

5.10.2 Field Documentation

5.10.2.1 `const char* tPushButtonWidget::pcText`

A pointer to the text to draw on this push button, if PB_STYLE_TEXT is selected.

5.10.2.2 `void(* tPushButtonWidget::pfnOnClick)(tWidget *pWidget)`

A pointer to the function to be called when the button is pressed. This is repeatedly called when PB_STYLE_AUTO_REPEAT is selected.

5.10.2.3 `const tFont* tPushButtonWidget::pFont`

A pointer to the font used to render the push button text, if PB_STYLE_TEXT is selected.

5.10.2.4 `const unsigned char* tPushButtonWidget::puImage`

A pointer to the image to be drawn onto this push button, if PB_STYLE_IMG is selected.

5.10.2.5 const unsigned char* tPushButtonWidget::pucPressImage

A pointer to the image to be drawn onto this push button when it is pressed, if PB_STYLE_PRESSEDIMG is selected.

5.10.2.6 tWidget tPushButtonWidget::sBase

The generic widget information.

5.10.2.7 unsigned tPushButtonWidget::ulAutoRepeatCount

The number of pointer events that have occurred. This is used when PB_STYLE_AUTO_REPEAT is selected to generate the auto-repeat events.

5.10.2.8 unsigned long tPushButtonWidget::ulFillColor

The 24-bit RGB color used to fill this push button, if PB_STYLE_FILL is selected, and to use as the background color if PB_STYLE_TEXT_OPAQUE is selected.

5.10.2.9 unsigned long tPushButtonWidget::ulOutlineColor

The 24-bit RGB color used to outline this push button, if PB_STYLE_OUTLINE is selected.

5.10.2.10 unsigned long tPushButtonWidget::ulPressFillColor

The 24-bit RGB color used to fill this push button when it is pressed, if PB_STYLE_FILL is selected, and to use as the background color if PB_STYLE_TEXT_OPAQUE is selected.

5.10.2.11 unsigned tPushButtonWidget::ulStyle

The style for this widget. This is a set of flags defined by PB_STYLE_xxx.

5.10.2.12 unsigned long tPushButtonWidget::ulTextColor

The 24-bit RGB color used to draw text on this push button, if PB_STYLE_TEXT is selected.

5.10.2.13 unsigned short tPushButtonWidget::usAutoRepeatDelay

The number of pointer events to delay before starting to auto-repeat, if PB_STYLE_AUTO_REPEAT is selected. The amount of time to which this corresponds is dependent upon the rate at which pointer events are generated by the pointer driver.

5.10.2.14 unsigned short tPushButtonWidget::usAutoRepeatRate

The number of pointer events between button presses generated by the auto-repeat function, if PB_STYLE_AUTO_REPEAT is selected. The amount of time to which this corresponds is dependent up on the rate at which pointer events are generated by the pointer driver.

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

- include/[pushbutton.h](#)

5.11 tRadioButtonWidget Struct Reference

The structure that describes a radio button widget.

```
#include <radiobutton.h>
```

Data Fields

- **tWidget sBase**
The generic widget information.
- unsigned short **usStyle**
- unsigned short **usCircleSize**
- unsigned long **ulFillColor**
- unsigned long **ulOutlineColor**
- unsigned long **ulTextColor**
- const **tFont * pFont**
- const char * **pcText**
- const unsigned char * **puImage**
- void(* **pfnOnChange**)(tWidget *pWidget, int bSelected)

5.11.1 Detailed Description

The structure that describes a radio button widget.

5.11.2 Field Documentation

5.11.2.1 const char* tRadioButtonWidget::pcText

A pointer to the text to draw on this radio button, if RB_STYLE_TEXT is selected.

5.11.2.2 void(* tRadioButtonWidget::pfnOnChange)(tWidget *pWidget, int bSelected)

A pointer to the function to be called when the radio button is pressed. This function is called when the state of the radio button is changed.

5.11.2.3 const tFont* tRadioButtonWidget::pFont

The font used to draw the radio button text, if RB_STYLE_TEXT is selected.

5.11.2.4 const unsigned char* tRadioButtonWidget::puImage

A pointer to the image to be drawn onto this radio button, if RB_STYLE_IMG is selected.

5.11.2.5 tWidget tRadioButtonWidget::sBase

The generic widget information.

5.11.2.6 unsigned long tRadioButtonWidget::ulFillColor

The 24-bit RGB color used to fill this radio button, if RB_STYLE_FILL is selected, and to use as the background color if RB_STYLE_TEXT_OPAQUE is selected.

5.11.2.7 unsigned long tRadioButtonWidget::ulOutlineColor

The 24-bit RGB color used to outline this radio button, if RB_STYLE_OUTLINE is selected.

5.11.2.8 unsigned long tRadioButtonWidget::ulTextColor

The 24-bit RGB color used to draw text on this radio button, if RB_STYLE_TEXT is selected.

5.11.2.9 unsigned short tRadioButtonWidget::usCircleSize

The size of the radio button itself, not including the text and/or image that accompanies it (in other words, the size of the actual circle that is filled or unfilled).

5.11.2.10 unsigned short tRadioButtonWidget::usStyle

The style for this radio button. This is a set of flags defined by RB_STYLE_xxx.

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

- include/radiobutton.h

5.12 tRectangle Struct Reference

```
#include <grlib.h>
```

Data Fields

- short **sXMin**
The minimum X coordinate of the rectangle.
- short **sYMin**
The minimum Y coordinate of the rectangle.
- short **sXMax**
The maximum X coordinate of the rectangle.
- short **sYMax**
The maximum Y coordinate of the rectangle.

5.12.1 Detailed Description

This structure defines the extents of a rectangle. All points greater than or equal to the minimum and less than or equal to the maximum are part of the rectangle.

5.12.2 Field Documentation

5.12.2.1 short tRectangle::sXMax

The maximum X coordinate of the rectangle.

5.12.2.2 short tRectangle::sXMin

The minimum X coordinate of the rectangle.

5.12.2.3 short tRectangle::sYMax

The maximum Y coordinate of the rectangle.

5.12.2.4 short tRectangle::sYMin

The minimum Y coordinate of the rectangle.

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

- include/[grlib.h](#)

5.13 tSliderWidget Struct Reference

The structure that describes a slider widget.

```
#include <slider.h>
```

Data Fields

- **tWidget sBase**
The generic widget information.
- unsigned **ulStyle**
- unsigned long **ulFillColor**
- unsigned long **ulBackgroundFillColor**
- unsigned long **ulOutlineColor**
- unsigned long **ulTextColor**
- unsigned long **ulBackgroundTextColor**
- const **tFont * pFont**
- const char * **pcText**
- const unsigned char * **puImage**
- const unsigned char * **pucBackgroundImage**
- void(* **pfnOnChange**)(tWidget *pWidget, unsigned **IValue**)
- unsigned **IMin**
- unsigned **IMax**
- unsigned **IValue**
- short **sPos**

5.13.1 Detailed Description

The structure that describes a slider widget.

5.13.2 Field Documentation

5.13.2.1 unsigned tSliderWidget::IMax

The value represented by the slider at its maximum position. This value is returned if a horizontal slider is pulled to the far right or a vertical slider is pulled to the top of the widget's bounding rectangle.

5.13.2.2 unsigned tSliderWidget::IMin

The value represented by the slider at its zero position. This value is returned if a horizontal slider is pulled to the far left or a vertical slider is pulled to the bottom of widget's bounding rectangle.

5.13.2.3 unsigned tSliderWidget::lValue

The current slider value scaled according to the minimum and maximum values for the control.

5.13.2.4 const char* tSliderWidget::pcText

A pointer to the text to draw on this slider, if SL_STYLE_TEXT is selected.

5.13.2.5 void(* tSliderWidget::pfnOnChange)(tWidget *pWidget, unsigned lValue)

A pointer to the function to be called when the state of the slider changes.

5.13.2.6 const tFont* tSliderWidget::pFont

A pointer to the font used to render the slider text, if SL_STYLE_TEXT is selected.

5.13.2.7 const unsigned char* tSliderWidget::pucBackgroundImage

A pointer to the image to be drawn onto this slider background if SL_STYLE_BACKG_IMG is selected.

5.13.2.8 const unsigned char* tSliderWidget::puclImage

A pointer to the image to be drawn onto this slider, if SL_STYLE_IMG is selected.

5.13.2.9 tWidget tSliderWidget::sBase

The generic widget information.

5.13.2.10 short tSliderWidget::sPos

This internal work variable stores the pixel position representing the current slider value.

5.13.2.11 unsigned long tSliderWidget::ulBackgroundColor

The 24-bit RGB color used to fill the background portion of the slider if SL_STYLE_FILL is selected, and to use as the background color if SL_STYLE_TEXT_OPAQUE is selected.

5.13.2.12 unsigned long tSliderWidget::ulBackgroundTextColor

The 24-bit RGB color used to draw text on the background portion of this slider, if SL_STYLE_TEXT is selected.

5.13.2.13 unsigned long tSliderWidget::ulFillColor

The 24-bit RGB color used to fill this slider, if SL_STYLE_FILL is selected, and to use as the background color if SL_STYLE_TEXT_OPAQUE is selected.

5.13.2.14 unsigned long tSliderWidget::ulOutlineColor

The 24-bit RGB color used to outline this slider, if SL_STYLE_OUTLINE is selected.

5.13.2.15 unsigned tSliderWidget::ulStyle

The style for this widget. This is a set of flags defined by SL_STYLE_xxx.

5.13.2.16 unsigned long tSliderWidget::ulTextColor

The 24-bit RGB color used to draw text on the "active" portion of this slider, if SL_STYLE_TEXT is selected.

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

- include/[slider.h](#)

5.14 tWidgetMessageQueue Struct Reference

Data Fields

- unsigned [ulFlags](#)
- [tWidget](#) * [pWidget](#)
- unsigned [ulMessage](#)
- unsigned [ulParam1](#)
- unsigned [ulParam2](#)

5.14.1 Field Documentation

5.14.1.1 [tWidget*](#) tWidgetMessageQueue::pWidget**5.14.1.2 unsigned tWidgetMessageQueue::ulFlags**

5.14.1.3 `unsigned tWidgetMessageQueue::ulMessage`

5.14.1.4 `unsigned tWidgetMessageQueue::ulParam1`

5.14.1.5 `unsigned tWidgetMessageQueue::ulParam2`

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

- `grlib/widget.c`

Chapter 6

File Documentation

6.1 driver/sed1335-AT91.c File Reference

```
#include "build/AT91SAM7S256.h"
```

Defines

- #define SED1335_DATA_SET (*AT91C_PIOA_SODR)
- #define SED1335_DATA_CLR (*AT91C_PIOA_CODR)
- #define SED1335_DATA_OUT (*AT91C_PIOA_OER)
- #define SED1335_DATA_INP (*AT91C_PIOA_ODR)
- #define SED1335_DATA_PIN (*AT91C_PIOA_PDSR)
- #define SED1335_DATA_IO_EN (*AT91C_PIOA_PER)
- #define SED1335_D0 16
- #define SED1335_CTRL_SET (*AT91C_PIOA_SODR)
- #define SED1335_CTRL_CLR (*AT91C_PIOA_CODR)
- #define SED1335_CTRL_OUT (*AT91C_PIOA_OER)
- #define SED1335_CTRL_INP (*AT91C_PIOA_ODR)
- #define SED1335_CTRL_PIN (*AT91C_PIOA_PDSR)
- #define SED1335_CTRL_IO_EN (*AT91C_PIOA_PER)
- #define SED1335_WR (1 << 8)
- #define SED1335_RD (1 << 9)
- #define SED1335_A0 (1 << 10)
- #define SED1335_CS (1 << 11)
- #define SED1335_RES (1 << 12)

Functions

- void [Init](#) (void)
- void [GLCD_InitializePorts](#) (void)
- void [GLCD_WriteCommand](#) (unsigned char commandToWrite)

- void **GLCD_WriteData** (unsigned char dataToWrite)
- unsigned char **GLCD_ReadData** (void)
- char **GLCD_ReadByteFromROMMemory** (char *ptr)

6.1.1 Define Documentation

- 6.1.1.1 #define **SED1335_A0** (1 << 10)
- 6.1.1.2 #define **SED1335_CS** (1 << 11)
- 6.1.1.3 #define **SED1335_CTRL_CLR** (*AT91C_PIOA_CODR)
- 6.1.1.4 #define **SED1335_CTRL_INP** (*AT91C_PIOA_ODR)
- 6.1.1.5 #define **SED1335_CTRL_IO_EN** (*AT91C_PIOA_PER)
- 6.1.1.6 #define **SED1335_CTRL_OUT** (*AT91C_PIOA_OER)
- 6.1.1.7 #define **SED1335_CTRL_PIN** (*AT91C_PIOA_PDSR)
- 6.1.1.8 #define **SED1335_CTRL_SET** (*AT91C_PIOA_SODR)
- 6.1.1.9 #define **SED1335_D0** 16
- 6.1.1.10 #define **SED1335_DATA_CLR** (*AT91C_PIOA_CODR)
- 6.1.1.11 #define **SED1335_DATA_INP** (*AT91C_PIOA_ODR)
- 6.1.1.12 #define **SED1335_DATA_IO_EN** (*AT91C_PIOA_PER)
- 6.1.1.13 #define **SED1335_DATA_OUT** (*AT91C_PIOA_OER)
- 6.1.1.14 #define **SED1335_DATA_PIN** (*AT91C_PIOA_PDSR)
- 6.1.1.15 #define **SED1335_DATA_SET** (*AT91C_PIOA_SODR)
- 6.1.1.16 #define **SED1335_RD** (1 << 9)
- 6.1.1.17 #define **SED1335_RES** (1 << 12)
- 6.1.1.18 #define **SED1335_WR** (1 << 8)

6.1.2 Function Documentation

- 6.1.2.1 void **GLCD_InitializePorts** (void)

- 6.1.2.2 char **GLCD_ReadByteFromROMMemory** (char * *ptr*)
- 6.1.2.3 unsigned char **GLCD_ReadData** (void)
- 6.1.2.4 void **GLCD_WriteCommand** (unsigned char *commandToWrite*)
- 6.1.2.5 void **GLCD_WriteData** (unsigned char *dataToWrite*)
- 6.1.2.6 void **Init** (void)

6.2 driver/sed1335-avr.c File Reference

```
#include <avr/io.h> #include <avr/pgmspace.h>
```

Defines

- #define **SED1335_DATA_PORT** PORTC
- #define **SED1335_DATA_DIR** DDRC
- #define **SED1335_DATA_PIN** PINC
- #define **SED1335_CONTROL_PORT** PORTD
- #define **SED1335_CONTROL_DIR** DDRD
- #define **SED1335_CONTROL_PIN** PIND
- #define **SED1335_A0** (1 << PD0)
- #define **SED1335_WR** (1 << PD1)
- #define **SED1335_RD** (1 << PD2)
- #define **SED1335_CS** (1 << PD3)
- #define **SED1335_RES** (1 << PD4)

Functions

- void **GLCD_InitializePorts** (void)
- void **GLCD_WriteData** (unsigned char *dataToWrite*)
- void **GLCD_WriteCommand** (unsigned char *commandToWrite*)
- unsigned char **GLCD_ReadData** (void)
- unsigned char **GLCD_ReadByteFromROMMemory** (unsigned char **ptr*)

6.2.1 Define Documentation

- 6.2.1.1 #define **SED1335_A0** (1 << PD0)
- 6.2.1.2 #define **SED1335_CONTROL_DIR** DDRD
- 6.2.1.3 #define **SED1335_CONTROL_PIN** PIND

6.2.1.4 #define SED1335_CONTROL_PORT PORTD

6.2.1.5 #define SED1335_CS (1 << PD3)

6.2.1.6 #define SED1335_DATA_DIR DDRC

6.2.1.7 #define SED1335_DATA_PIN PINC

6.2.1.8 #define SED1335_DATA_PORT PORTC

6.2.1.9 #define SED1335_RD (1 << PD2)

6.2.1.10 #define SED1335_RES (1 << PD4)

6.2.1.11 #define SED1335_WR (1 << PD1)

6.2.2 Function Documentation

6.2.2.1 void GLCD_InitializePorts (void)

6.2.2.2 unsigned char GLCD_ReadByteFromROMMemory (unsigned char * ptr)

6.2.2.3 unsigned char GLCD_ReadData (void)

6.2.2.4 void GLCD_WriteCommand (unsigned char commandToWrite)

6.2.2.5 void GLCD_WriteData (unsigned char dataToWrite)

6.3 driver/sed1335-LPC2100.c File Reference

```
#include "build/lpc213x.h"
```

Defines

- #define SED1335_SET IOSET0
- #define SED1335_CLR IOCLR0
- #define SED1335_DIR IODIRO
- #define SED1335_PIN IOPINO
- #define SED1335_RD (1 << 4) /* pin PA8 -> E */
- #define SED1335_WR (1 << 5) /* pin PA9 -> RW */
- #define SED1335_A0 (1 << 6) /* pin PA10 -> RS */
- #define SED1335_RES (1 << 7) /* pin PA11 -> RST */
- #define SED1335_CS1 (1 << 8) /* pin PA12 -> CS1B */
- #define SED1335_D0 16 /* first databus bit is pin PA0 */

Functions

- void **GLCD_InitializePorts** (void)
- unsigned char **GLCD_ReadStatus** (void)
- unsigned char **GLCD_ReadData** (void)
- void **GLCD_WriteData** (unsigned char *dataToWrite*)
- void **GLCD_WriteCommand** (unsigned char *commandToWrite*)
- char **GLCD_ReadByteFromROMMemory** (char **ptr*)

6.3.1 Define Documentation

- 6.3.1.1 #define **SED1335_A0** (1 << 6) /* pin PA10 -> RS */
- 6.3.1.2 #define **SED1335_CLR** IOCLR0
- 6.3.1.3 #define **SED1335_CS1** (1 << 8) /* pin PA12 -> CS1B */
- 6.3.1.4 #define **SED1335_D0** 16 /* first databus bit is pin PA0 */
- 6.3.1.5 #define **SED1335_DIR** IODIR0
- 6.3.1.6 #define **SED1335_PIN** IOPIN0
- 6.3.1.7 #define **SED1335_RD** (1 << 4) /* pin PA8 -> E */
- 6.3.1.8 #define **SED1335_RES** (1 << 7) /* pin PA11 -> RST */
- 6.3.1.9 #define **SED1335_SET** IOSET0
- 6.3.1.10 #define **SED1335_WR** (1 << 5) /* pin PA9 -> RW */

6.3.2 Function Documentation

- 6.3.2.1 void **GLCD_InitializePorts** (void)
- 6.3.2.2 char **GLCD_ReadByteFromROMMemory** (char * *ptr*)
- 6.3.2.3 unsigned char **GLCD_ReadData** (void)
- 6.3.2.4 unsigned char **GLCD_ReadStatus** (void)
- 6.3.2.5 void **GLCD_WriteCommand** (unsigned char *commandToWrite*)
- 6.3.2.6 void **GLCD_WriteData** (unsigned char *dataToWrite*)

6.4 driver/sed1335-MSP430.c File Reference

```
#include <stdlib.h> #include <msp430.h>
```

Defines

- #define SED1335_DIR P1DIR
- #define SED1335_PIN P1OUT
- #define SED1335_RD BIT0 /* pin 0 -> E */
- #define SED1335_WR BIT1 /* pin 1 -> RW */
- #define SED1335_A0 BIT2 /* pin 2 -> RS */
- #define SED1335_RES BIT3 /* pin 3 -> RST */
- #define SED1335_CS1 BIT4 /* pin 4 -> CS1B */
- #define SED1335_D0 0 /* first databus bit is pin P2.0 */
- #define SED1335_DATA P2OUT /* port databus is pin P2 */
- #define SED1335_DATA_DIR P2DIR /* port databus dir */

Functions

- void GLCD_InitializePorts (void)
- unsigned char GLCD_ReadStatus (void)
- unsigned char GLCD_ReadData (void)
- void GLCD_WriteData (unsigned char dataToWrite)
- void GLCD_WriteCommand (unsigned char commandToWrite)
- char GLCD_ReadByteFromROMMemory (char *ptr)

6.4.1 Define Documentation

- 6.4.1.1 #define SED1335_A0 BIT2 /* pin 2 -> RS */
- 6.4.1.2 #define SED1335_CS1 BIT4 /* pin 4 -> CS1B */
- 6.4.1.3 #define SED1335_D0 0 /* first databus bit is pin P2.0 */
- 6.4.1.4 #define SED1335_DATA P2OUT /* port databus is pin P2 */
- 6.4.1.5 #define SED1335_DATA_DIR P2DIR /* port databus dir */
- 6.4.1.6 #define SED1335_DIR P1DIR
- 6.4.1.7 #define SED1335_PIN P1OUT
- 6.4.1.8 #define SED1335_RD BIT0 /* pin 0 -> E */
- 6.4.1.9 #define SED1335_RES BIT3 /* pin 3 -> RST */

6.4.1.10 #define SED1335_WR BIT1 /* pin 1 -> RW */

6.4.2 Function Documentation

6.4.2.1 void GLCD_InitializePorts (void)

6.4.2.2 char GLCD_ReadByteFromROMMemory (char *ptr)

6.4.2.3 unsigned char GLCD_ReadData (void)

6.4.2.4 unsigned char GLCD_ReadStatus (void)

6.4.2.5 void GLCD_WriteCommand (unsigned char commandToWrite)

6.4.2.6 void GLCD_WriteData (unsigned char dataToWrite)

6.5 driver/sed1335-STM32.c File Reference

```
#include "stm32f10x_lib.h"
```

Defines

- #define SED1335_PORT GPIOE
- #define SED1335_A0 GPIO_Pin_10
- #define SED1335_WR GPIO_Pin_8
- #define SED1335_RD GPIO_Pin_9
- #define SED1335_CS GPIO_Pin_11
- #define SED1335_RES GPIO_Pin_12
- #define SED1335_D0 0

Functions

- void GLCD_InitPorts (void)
- void GLCD_WriteData (unsigned char dataToWrite)
- void GLCD_WriteCommand (unsigned char commandToWrite)
- unsigned char GLCD_ReadData (void)
- unsigned char GLCD_ReadByteFromROMMemory (unsigned char *ptr)

Variables

- GPIO_InitTypeDef GPIO_InitStructure

6.5.1 Define Documentation

- 6.5.1.1 #define SED1335_A0 GPIO_Pin_10
- 6.5.1.2 #define SED1335_CS GPIO_Pin_11
- 6.5.1.3 #define SED1335_D0 0
- 6.5.1.4 #define SED1335_PORT GPIOE
- 6.5.1.5 #define SED1335_RD GPIO_Pin_9
- 6.5.1.6 #define SED1335_RES GPIO_Pin_12
- 6.5.1.7 #define SED1335_WR GPIO_Pin_8

6.5.2 Function Documentation

- 6.5.2.1 void GLCD_InitPorts(void)
- 6.5.2.2 unsigned char GLCD_ReadByteFromROMMemory(unsigned char *ptr)
- 6.5.2.3 unsigned char GLCD_ReadData(void)
- 6.5.2.4 void GLCD_WriteCommand(unsigned char commandToWrite)
- 6.5.2.5 void GLCD_WriteData(unsigned char dataToWrite)

6.5.3 Variable Documentation

- 6.5.3.1 GPIO_InitTypeDef GPIO_InitStructure

6.6 driver/sed1335.c File Reference

```
#include "sed1335.h"
```

Functions

- void GLCD_InitializePorts(void)
- void GLCD_Initialize(void)
- void GLCD_SetPixel(unsigned int x, unsigned int y, int color)
- void GLCD_WriteText(char *tekst)
- void GLCD_WriteTextP(char *tekst)
- void GLCD_SetCursorAddress(unsigned int address)
- void GLCD_TextGoTo(unsigned char x, unsigned char y)
- void GLCD_GraphicGoTo(unsigned int x, unsigned int y)

- void `GLCD_ClearText` (void)
- void `GLCD_ClearGraphic` (void)
- void `GLCD_Bitmap` (char **bmp*, int *x*, int *y*, int *width*, int *height*)

6.6.1 Function Documentation

6.6.1.1 void `GLCD_Bitmap` (char **bmp*, int *x*, int *y*, int *width*, int *height*)

Draw a bit

Parameters

<i>bmp</i>	is the pointer that contains the values of the Bitmap
<i>x</i>	is the position where the bitmap will start in X
<i>y</i>	is the position where the bitmap will start in Y.
<i>width</i>	is the width of the Bitmap
<i>height</i>	is the height of the Bitmap

This function draw a Bitmap on GLCD, starting from (X,Y)

Returns

None.

6.6.1.2 void `GLCD_ClearGraphic` (void)

Clean the screen

This function clean the screen of the GLCD for graphics.

Returns

None.

6.6.1.3 void `GLCD_ClearText` (void)

Clean the screen

This function clean the screen of the GLCD for texts.

Returns

None.

6.6.1.4 void `GLCD_GraphicGoTo` (unsigned int *x*, unsigned int *y*)

Set the graphic's cursor on a address

Parameters

<i>x</i>	is the position where the cursor will be send in X.
<i>y</i>	is the position where the cursor will be send in Y.

This function addresses the GLCD's graphic cursor on a specific address.

Returns

None.

6.6.1.5 void GLCD_Initialize(void)

Initialize the GLCD

This function initialize the ports and start up the GLCD with the default options.

Returns

None.

6.6.1.6 void GLCD_InitializePorts(void)**6.6.1.7 void GLCD_SetCursorPosition(unsigned int *address*)**

Set the cursor on a address of the ROM

Parameters

<i>address</i>	is the local where the cursor will point.
----------------	---

This function addresses the GLCD's cursor on a specific address.

Returns

None.

6.6.1.8 void GLCD_SetPixel(unsigned int *x*, unsigned int *y*, int *color*)

Draws a pixel point

Parameters

<i>x</i>	is the position where the point will be print in X.
<i>y</i>	is the position where the point will be print in Y.
<i>color</i>	is the color of the pixel printed in (X,Y).

This function draws pixel point in position (X,Y) with the color chosen.

Returns

None.

6.6.1.9 void GLCD_TextGoTo (unsigned char x, unsigned char y)

Set the text's cursor on a address

Parameters

x	is the position where the cursor will be send in X.
y	is the position where the cursor will be send in Y.

This function addresses the GLCD's text cursor on a specific address.

Returns

None.

6.6.1.10 void GLCD_WriteText (char * *tekst*)

Print a message on GLCD

Parameters

<i>tekst</i>	is a string that will be send to the GLCD.
--------------	--

This function prints a message on GLCD.

Returns

None.

6.6.1.11 void GLCD_WriteTextP (char * *tekst*)

Print a message on GLCD

Parameters

<i>tekst</i>	is a pointer that will be used to read a message from the ROM of the GLCD.
--------------	--

This function prints a message on GLCD located on local pointed by *teskt

Returns

None.

6.7 grlib/canvas.c File Reference

```
#include "grlib.h" #include "widget.h" #include "canvas.h"
```

Functions

- int [CanvasMsgProc](#) (*tWidget* **pWidget*, unsigned *ulMsg*, unsigned *ulParam1*, unsigned *ulParam2*)
- void [CanvasInit](#) (*tCanvasWidget* **pWidget*, const *tDisplay* **pDisplay*, unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)

6.7.1 Function Documentation

6.7.1.1 void [CanvasInit](#) (*tCanvasWidget* * *pWidget*, const *tDisplay* * *pDisplay*, unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)

Initializes a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the canvas.
<i>IX</i>	is the X coordinate of the upper left corner of the canvas.
<i>IY</i>	is the Y coordinate of the upper left corner of the canvas.
<i>IWidth</i>	is the width of the canvas.
<i>IHeight</i>	is the height of the canvas.

This function initializes the provided canvas widget.

Returns

None.

6.7.1.2 int [CanvasMsgProc](#) (*tWidget* * *pWidget*, unsigned *ulMsg*, unsigned *ulParam1*, unsigned *ulParam2*)

Handles messages for a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this canvas widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.8 grlib/checkbox.c File Reference

```
#include "grlib.h" #include "widget.h" #include "checkbox.-  
h"
```

Functions

- int [CheckBoxMsgProc](#) (*tWidget* **pWidget*, unsigned *ulMsg*, unsigned *ulParam1*, unsigned *ulParam2*)
- void [CheckBoxInit](#) (*tCheckBoxWidget* **pWidget*, const *tDisplay* **pDisplay*, unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)

6.8.1 Function Documentation

6.8.1.1 void CheckBoxInit (*tCheckBoxWidget* * *pWidget*, const *tDisplay* * *pDisplay*, unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)

Initializes a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the check box.
<i>IX</i>	is the X coordinate of the upper left corner of the check box.
<i>IY</i>	is the Y coordinate of the upper left corner of the check box.
<i>IWidth</i>	is the width of the check box.
<i>IHeight</i>	is the height of the check box.

This function initializes the provided check box widget.

Returns

None.

6.8.1.2 int CheckBoxMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this check box widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.9 grlib/circle.c File Reference

```
#include "grlib.h"
```

Functions

- void [GrCircleDraw](#) (const [tContext](#) *pContext, unsigned IX, unsigned IY, unsigned IRaduis)
- void [GrCircleFill](#) (const [tContext](#) *pContext, unsigned IX, unsigned IY, unsigned IRaduis)

6.9.1 Function Documentation

6.9.1.1 void GrCircleDraw (const [tContext](#) * pContext, unsigned IX, unsigned IY, unsigned IRaduis)

Draws a circle.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX</i>	is the X coordinate of the center of the circle.
<i>IY</i>	is the Y coordinate of the center of the circle.
<i>IRadius</i>	is the radius of the circle.

This function draws a circle, utilizing the Bresenham circle drawing algorithm. The extent of the circle is from *IX* - *IRadius* to *IX* + *IRadius* and *IY* - *IRadius* to *IY* + *IRadius*, inclusive.

Returns

None.

6.9.1.2 void GrCircleFill (const tContext * *pContext*, unsigned *IX*, unsigned *IY*, unsigned *IRadius*)

Draws a filled circle.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX</i>	is the X coordinate of the center of the circle.
<i>IY</i>	is the Y coordinate of the center of the circle.
<i>IRadius</i>	is the radius of the circle.

This function draws a filled circle, utilizing the Bresenham circle drawing algorithm. The extent of the circle is from *IX* - *IRadius* to *IX* + *IRadius* and *IY* - *IRadius* to *IY* + *IRadius*, inclusive.

Returns

None.

6.10 grlib/container.c File Reference

```
#include "grlib.h" #include "widget.h" #include "container.h"
```

Functions

- int **ContainerMsgProc** (tWidget **pWidget*, unsigned *ulMsg*, unsigned *ulParam1*, unsigned *ulParam2*)
- void **ContainerInit** (tContainerWidget **pWidget*, const tDisplay **pDisplay*, unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)

6.10.1 Function Documentation

**6.10.1.1 void ContainerInit (tContainerWidget * *pWidget*, const tDisplay * *pDisplay*,
unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)**

Initializes a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the container widget.
<i>IX</i>	is the X coordinate of the upper left corner of the container widget.
<i>IY</i>	is the Y coordinate of the upper left corner of the container widget.
<i>IWidth</i>	is the width of the container widget.
<i>IHeight</i>	is the height of the container widget.

This function initializes a container widget, preparing it for placement into the widget tree.

Returns

none.

**6.10.1.2 int ContainerMsgProc (tWidget * *pWidget*, unsigned *ulMsg*, unsigned *ulParam1*,
unsigned *ulParam2*)**

Handles messages for a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this container widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.11 grlib/context.c File Reference

```
#include "grlib.h"
```

Functions

- void `GrContextInit (tContext *pContext, const tDisplay *pDisplay)`
- void `GrContextClipRegionSet (tContext *pContext, tRectangle *pRect)`

6.11.1 Function Documentation

6.11.1.1 void GrContextClipRegionSet (tContext * pContext, tRectangle * pRect)

Sets the extents of the clipping region.

Parameters

<code>pContext</code>	is a pointer to the drawing context to use.
<code>pRect</code>	is a pointer to the structure containing the extents of the clipping region.

This function sets the extents of the clipping region. The clipping region is not allowed to exceed the extents of the screen, but may be a portion of the screen.

The supplied coordinate are inclusive; `sXMin` of 1 and `sXMax` of 1 will define a clipping region that will display only the pixels in the `X = 1` column. A consequence of this is that the clipping region must contain at least one row and one column.

Returns

None.

6.11.1.2 void GrContextInit (tContext * pContext, const tDisplay * pDisplay)

Initializes a drawing context.

Parameters

<code>pContext</code>	is a pointer to the drawing context to initialize.
<code>pDisplay</code>	is a pointer to the <code>tDisplayInfo</code> structure that describes the display driver to use.

This function initializes a drawing context, preparing it for use. The provided display driver will be used for all subsequent graphics operations, and the default clipping region will be set to the extent of the screen.

Returns

None.

6.12 grlib/fonts/fontcm12.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm12

6.12.1 Variable Documentation

6.12.1.1 const tFont g_sFontCm12

6.13 grlib/fonts/fontcm12b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm12b

6.13.1 Variable Documentation

6.13.1.1 const tFont g_sFontCm12b

6.14 grlib/fonts/fontcm12i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm12i

6.14.1 Variable Documentation

6.14.1.1 const tFont g_sFontCm12i

6.15 grlib/fonts/fontcm14.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm14

6.15.1 Variable Documentation

6.15.1.1 const tFont g_sFontCm14

6.16 grlib/fonts/fontcm14b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm14b

6.16.1 Variable Documentation

6.16.1.1 const tFont g_sFontCm14b

6.17 grlib/fonts/fontcm14i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm14i

6.17.1 Variable Documentation

6.17.1.1 const tFont g_sFontCm14i

6.18 grlib/fonts/fontcm16.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm16

6.18.1 Variable Documentation

6.18.1.1 const tFont g_sFontCm16

6.19 grlib/fonts/fontcm16b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm16b

6.19.1 Variable Documentation

6.19.1.1 const tFont g_sFontCm16b

6.20 grlib/fonts/fontcm16i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm16i

6.20.1 Variable Documentation

6.20.1.1 const tFont g_sFontCm16i

6.21 grlib/fonts/fontcm18.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm18

6.21.1 Variable Documentation

6.21.1.1 const tFont g_sFontCm18

6.22 grlib/fonts/fontcm18b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm18b

6.22.1 Variable Documentation

6.22.1.1 const tFont g_sFontCm18b

6.23 grlib/fonts/fontcm18i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm18i

6.23.1 Variable Documentation

6.23.1.1 const tFont g_sFontCm18i

6.24 grlib/fonts/fontcm20.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm20

6.24.1 Variable Documentation

6.24.1.1 const tFont g_sFontCm20

6.25 grlib/fonts/fontcm20b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm20b

6.25.1 Variable Documentation

6.25.1.1 const tFont g_sFontCm20b

6.26 grlib/fonts/fontcm20i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm20i

6.26.1 Variable Documentation

6.26.1.1 const tFont g_sFontCm20i

6.27 grlib/fonts/fontcm22.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm22

6.27.1 Variable Documentation

6.27.1.1 const tFont g_sFontCm22

6.28 grlib/fonts/fontcm22b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm22b

6.28.1 Variable Documentation

6.28.1.1 const tFont g_sFontCm22b

6.29 grlib/fonts/fontcm22i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const [tFont g_sFontCm22i](#)

6.29.1 Variable Documentation

6.29.1.1 const [tFont g_sFontCm22i](#)

6.30 grlib/fonts/fontcm24.c File Reference

```
#include "..\grlib.h"
```

Variables

- const [tFont g_sFontCm24](#)

6.30.1 Variable Documentation

6.30.1.1 const [tFont g_sFontCm24](#)

6.31 grlib/fonts/fontcm24b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const [tFont g_sFontCm24b](#)

6.31.1 Variable Documentation

6.31.1.1 const [tFont g_sFontCm24b](#)

6.32 grlib/fonts/fontcm24i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm24i

6.32.1 Variable Documentation

6.32.1.1 const tFont g_sFontCm24i

6.33 grlib/fonts/fontcm26.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm26

6.33.1 Variable Documentation

6.33.1.1 const tFont g_sFontCm26

6.34 grlib/fonts/fontcm26b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm26b

6.34.1 Variable Documentation

6.34.1.1 const tFont g_sFontCm26b

6.35 grlib/fonts/fontcm26i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm26i

6.35.1 Variable Documentation

6.35.1.1 const tFont g_sFontCm26i

6.36 grlib/fonts/fontcm28.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm28

6.36.1 Variable Documentation

6.36.1.1 const tFont g_sFontCm28

6.37 grlib/fonts/fontcm28b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm28b

6.37.1 Variable Documentation

6.37.1.1 const tFont g_sFontCm28b

6.38 grlib/fonts/fontcm28i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm28i

6.38.1 Variable Documentation

6.38.1.1 const tFont g_sFontCm28i

6.39 grlib/fonts/fontcm30.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm30

6.39.1 Variable Documentation

6.39.1.1 const tFont g_sFontCm30

6.40 grlib/fonts/fontcm30b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm30b

6.40.1 Variable Documentation

6.40.1.1 const tFont g_sFontCm30b

6.41 grlib/fonts/fontcm30i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm30i

6.41.1 Variable Documentation

6.41.1.1 const tFont g_sFontCm30i

6.42 grlib/fonts/fontcm32.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm32

6.42.1 Variable Documentation

6.42.1.1 const tFont g_sFontCm32

6.43 grlib/fonts/fontcm32b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm32b

6.43.1 Variable Documentation

6.43.1.1 const tFont g_sFontCm32b

6.44 grlib/fonts/fontcm32i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm32i

6.44.1 Variable Documentation

6.44.1.1 const tFont g_sFontCm32i

6.45 grlib/fonts/fontcm34.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm34

6.45.1 Variable Documentation

6.45.1.1 const tFont g_sFontCm34

6.46 grlib/fonts/fontcm34b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm34b

6.46.1 Variable Documentation

6.46.1.1 const tFont g_sFontCm34b

6.47 grlib/fonts/fontcm34i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm34i

6.47.1 Variable Documentation

6.47.1.1 const tFont g_sFontCm34i

6.48 grlib/fonts/fontcm36.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm36

6.48.1 Variable Documentation

6.48.1.1 const tFont g_sFontCm36

6.49 grlib/fonts/fontcm36b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm36b

6.49.1 Variable Documentation

6.49.1.1 const tFont g_sFontCm36b

6.50 grlib/fonts/fontcm36i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm36i

6.50.1 Variable Documentation

6.50.1.1 const tFont g_sFontCm36i

6.51 grlib/fonts/fontcm38.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm38

6.51.1 Variable Documentation

6.51.1.1 const tFont g_sFontCm38

6.52 grlib/fonts/fontcm38b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm38b

6.52.1 Variable Documentation

6.52.1.1 const tFont g_sFontCm38b

6.53 grlib/fonts/fontcm38i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm38i

6.53.1 Variable Documentation

6.53.1.1 const tFont g_sFontCm38i

6.54 grlib/fonts/fontcm40.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm40

6.54.1 Variable Documentation

6.54.1.1 const tFont g_sFontCm40

6.55 grlib/fonts/fontcm40b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm40b

6.55.1 Variable Documentation

6.55.1.1 const tFont g_sFontCm40b

6.56 grlib/fonts/fontcm40i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm40i

6.56.1 Variable Documentation

6.56.1.1 const tFont g_sFontCm40i

6.57 grlib/fonts/fontcm42.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm42

6.57.1 Variable Documentation

6.57.1.1 const tFont g_sFontCm42

6.58 grlib/fonts/fontcm42b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm42b

6.58.1 Variable Documentation

6.58.1.1 const tFont g_sFontCm42b

6.59 grlib/fonts/fontcm42i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm42i

6.59.1 Variable Documentation

6.59.1.1 const tFont g_sFontCm42i

6.60 grlib/fonts/fontcm44.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm44

6.60.1 Variable Documentation

6.60.1.1 const tFont g_sFontCm44

6.61 grlib/fonts/fontcm44b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm44b

6.61.1 Variable Documentation

6.61.1.1 const tFont g_sFontCm44b

6.62 grlib/fonts/fontcm44i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm44i

6.62.1 Variable Documentation

6.62.1.1 const tFont g_sFontCm44i

6.63 grlib/fonts/fontcm46.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm46

6.63.1 Variable Documentation

6.63.1.1 const tFont g_sFontCm46

6.64 grlib/fonts/fontcm46b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm46b

6.64.1 Variable Documentation

6.64.1.1 const tFont g_sFontCm46b

6.65 grlib/fonts/fontcm46i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm46i

6.65.1 Variable Documentation

6.65.1.1 const tFont g_sFontCm46i

6.66 grlib/fonts/fontcm48.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm48

6.66.1 Variable Documentation

6.66.1.1 const tFont g_sFontCm48

6.67 grlib/fonts/fontcm48b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm48b

6.67.1 Variable Documentation

6.67.1.1 const tFont g_sFontCm48b

6.68 grlib/fonts/fontcm48i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCm48i

6.68.1 Variable Documentation

6.68.1.1 const tFont g_sFontCm48i

6.69 grlib/fonts/fontcmsc12.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc12

6.69.1 Variable Documentation

6.69.1.1 const tFont g_sFontCmsc12

6.70 grlib/fonts/fontcmsc14.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc14

6.70.1 Variable Documentation

6.70.1.1 const tFont g_sFontCmsc14

6.71 grlib/fonts/fontcmsc16.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc16

6.71.1 Variable Documentation

6.71.1.1 const tFont g_sFontCmsc16

6.72 grlib/fonts/fontcmsc18.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc18

6.72.1 Variable Documentation

6.72.1.1 const tFont g_sFontCmsc18

6.73 grlib/fonts/fontcmsc20.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc20

6.73.1 Variable Documentation

6.73.1.1 const tFont g_sFontCmsc20

6.74 grlib/fonts/fontcmsc22.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc22

6.74.1 Variable Documentation

6.74.1.1 const tFont g_sFontCmsc22

6.75 grlib/fonts/fontcmsc24.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc24

6.75.1 Variable Documentation

6.75.1.1 const tFont g_sFontCmsc24

6.76 grlib/fonts/fontcmsc26.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc26

6.76.1 Variable Documentation

6.76.1.1 const tFont g_sFontCmsc26

6.77 grlib/fonts/fontcmsc28.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc28

6.77.1 Variable Documentation

6.77.1.1 const tFont g_sFontCmsc28

6.78 grlib/fonts/fontcmsc30.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc30

6.78.1 Variable Documentation

6.78.1.1 const tFont g_sFontCmsc30

6.79 grlib/fonts/fontcmsc32.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc32

6.79.1 Variable Documentation

6.79.1.1 const tFont g_sFontCmsc32

6.80 grlib/fonts/fontcmsc34.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc34

6.80.1 Variable Documentation

6.80.1.1 const tFont g_sFontCmsc34

6.81 grlib/fonts/fontcmsc36.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc36

6.81.1 Variable Documentation

6.81.1.1 const tFont g_sFontCmsc36

6.82 grlib/fonts/fontcmsc38.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc38

6.82.1 Variable Documentation

6.82.1.1 const tFont g_sFontCmsc38

6.83 grlib/fonts/fontcmsc40.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc40

6.83.1 Variable Documentation

6.83.1.1 const tFont g_sFontCmsc40

6.84 grlib/fonts/fontcmsc42.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc42

6.84.1 Variable Documentation

6.84.1.1 const tFont g_sFontCmsc42

6.85 grlib/fonts/fontcmsc44.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc44

6.85.1 Variable Documentation

6.85.1.1 const tFont g_sFontCmsc44

6.86 grlib/fonts/fontcmsc46.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc46

6.86.1 Variable Documentation

6.86.1.1 const tFont g_sFontCmsc46

6.87 grlib/fonts/fontcmsc48.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmsc48

6.87.1 Variable Documentation

6.87.1.1 const tFont g_sFontCmsc48

6.88 grlib/fonts/fontcmss12.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss12

6.88.1 Variable Documentation

6.88.1.1 const tFont g_sFontCmss12

6.89 grlib/fonts/fontcmss12b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss12b

6.89.1 Variable Documentation

6.89.1.1 const tFont g_sFontCmss12b

6.90 grlib/fonts/fontcmss12i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss12i

6.90.1 Variable Documentation

6.90.1.1 const tFont g_sFontCmss12i

6.91 grlib/fonts/fontcmss14.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss14

6.91.1 Variable Documentation

6.91.1.1 const tFont g_sFontCmss14

6.92 grlib/fonts/fontcmss14b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss14b

6.92.1 Variable Documentation

6.92.1.1 const tFont g_sFontCmss14b

6.93 grlib/fonts/fontcmss14i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss14i

6.93.1 Variable Documentation

6.93.1.1 const tFont g_sFontCmss14i

6.94 grlib/fonts/fontcmss16.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss16

6.94.1 Variable Documentation

6.94.1.1 const tFont g_sFontCmss16

6.95 grlib/fonts/fontcmss16b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss16b

6.95.1 Variable Documentation

6.95.1.1 const tFont g_sFontCmss16b

6.96 grlib/fonts/fontcmss16i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss16i

6.96.1 Variable Documentation

6.96.1.1 const tFont g_sFontCmss16i

6.97 grlib/fonts/fontcmss18.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss18

6.97.1 Variable Documentation

6.97.1.1 const tFont g_sFontCmss18

6.98 grlib/fonts/fontcmss18b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss18b

6.98.1 Variable Documentation

6.98.1.1 const tFont g_sFontCmss18b

6.99 grlib/fonts/fontcmss18i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss18i

6.99.1 Variable Documentation

6.99.1.1 const tFont g_sFontCmss18i

6.100 grlib/fonts/fontcmss20.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss20

6.100.1 Variable Documentation

6.100.1.1 const tFont g_sFontCmss20

6.101 grlib/fonts/fontcmss20b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss20b

6.101.1 Variable Documentation

6.101.1.1 const tFont g_sFontCmss20b

6.102 grlib/fonts/fontcmss20i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss20i

6.102.1 Variable Documentation

6.102.1.1 const tFont g_sFontCmss20i

6.103 grlib/fonts/fontcmss22.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss22

6.103.1 Variable Documentation

6.103.1.1 const tFont g_sFontCmss22

6.104 grlib/fonts/fontcmss22b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss22b

6.104.1 Variable Documentation

6.104.1.1 const tFont g_sFontCmss22b

6.105 grlib/fonts/fontcmss22i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss22i

6.105.1 Variable Documentation

6.105.1.1 const tFont g_sFontCmss22i

6.106 grlib/fonts/fontcmss24.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss24

6.106.1 Variable Documentation

6.106.1.1 const tFont g_sFontCmss24

6.107 grlib/fonts/fontcmss24b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss24b

6.107.1 Variable Documentation

6.107.1.1 const tFont g_sFontCmss24b

6.108 grlib/fonts/fontcmss24i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss24i

6.108.1 Variable Documentation

6.108.1.1 const tFont g_sFontCmss24i

6.109 grlib/fonts/fontcmss26.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss26

6.109.1 Variable Documentation

6.109.1.1 const tFont g_sFontCmss26

6.110 grlib/fonts/fontcmss26b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss26b

6.110.1 Variable Documentation

6.110.1.1 const tFont g_sFontCmss26b

6.111 grlib/fonts/fontcmss26i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss26i

6.111.1 Variable Documentation

6.111.1.1 const tFont g_sFontCmss26i

6.112 grlib/fonts/fontcmss28.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss28

6.112.1 Variable Documentation

6.112.1.1 const tFont g_sFontCmss28

6.113 grlib/fonts/fontcmss28b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss28b

6.113.1 Variable Documentation

6.113.1.1 const tFont g_sFontCmss28b

6.114 grlib/fonts/fontcmss28i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss28i

6.114.1 Variable Documentation

6.114.1.1 const tFont g_sFontCmss28i

6.115 grlib/fonts/fontcmss30.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss30

6.115.1 Variable Documentation

6.115.1.1 const tFont g_sFontCmss30

6.116 grlib/fonts/fontcmss30b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss30b

6.116.1 Variable Documentation

6.116.1.1 const tFont g_sFontCmss30b

6.117 grlib/fonts/fontcmss30i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss30i

6.117.1 Variable Documentation

6.117.1.1 const tFont g_sFontCmss30i

6.118 grlib/fonts/fontcmss32.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss32

6.118.1 Variable Documentation

6.118.1.1 const tFont g_sFontCmss32

6.119 grlib/fonts/fontcmss32b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss32b

6.119.1 Variable Documentation

6.119.1.1 const tFont g_sFontCmss32b

6.120 grlib/fonts/fontcmss32i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss32i

6.120.1 Variable Documentation

6.120.1.1 const tFont g_sFontCmss32i

6.121 grlib/fonts/fontcmss34.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss34

6.121.1 Variable Documentation

6.121.1.1 const tFont g_sFontCmss34

6.122 grlib/fonts/fontcmss34b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss34b

6.122.1 Variable Documentation

6.122.1.1 const tFont g_sFontCmss34b

6.123 grlib/fonts/fontcmss34i.c File Reference

#include "..\grlib.h"

Variables

- const tFont g_sFontCmss34i

6.123.1 Variable Documentation

6.123.1.1 const tFont g_sFontCmss34i

6.124 grlib/fonts/fontcmss36.c File Reference

#include "..\grlib.h"

Variables

- const tFont g_sFontCmss36

6.124.1 Variable Documentation

6.124.1.1 const tFont g_sFontCmss36

6.125 grlib/fonts/fontcmss36b.c File Reference

#include "..\grlib.h"

Variables

- const tFont g_sFontCmss36b

6.125.1 Variable Documentation

6.125.1.1 const tFont g_sFontCmss36b

6.126 grlib/fonts/fontcmss36i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss36i

6.126.1 Variable Documentation

6.126.1.1 const tFont g_sFontCmss36i

6.127 grlib/fonts/fontcmss38.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss38

6.127.1 Variable Documentation

6.127.1.1 const tFont g_sFontCmss38

6.128 grlib/fonts/fontcmss38b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss38b

6.128.1 Variable Documentation

6.128.1.1 const tFont g_sFontCmss38b

6.129 grlib/fonts/fontcmss38i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss38i

6.129.1 Variable Documentation

6.129.1.1 const tFont g_sFontCmss38i

6.130 grlib/fonts/fontcmss40.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss40

6.130.1 Variable Documentation

6.130.1.1 const tFont g_sFontCmss40

6.131 grlib/fonts/fontcmss40b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss40b

6.131.1 Variable Documentation

6.131.1.1 const tFont g_sFontCmss40b

6.132 grlib/fonts/fontcmss40i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss40i

6.132.1 Variable Documentation

6.132.1.1 const tFont g_sFontCmss40i

6.133 grlib/fonts/fontcmss42.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss42

6.133.1 Variable Documentation

6.133.1.1 const tFont g_sFontCmss42

6.134 grlib/fonts/fontcmss42b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss42b

6.134.1 Variable Documentation

6.134.1.1 const tFont g_sFontCmss42b

6.135 grlib/fonts/fontcmss42i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss42i

6.135.1 Variable Documentation

6.135.1.1 const tFont g_sFontCmss42i

6.136 grlib/fonts/fontcmss44.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss44

6.136.1 Variable Documentation

6.136.1.1 const tFont g_sFontCmss44

6.137 grlib/fonts/fontcmss44b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss44b

6.137.1 Variable Documentation

6.137.1.1 const tFont g_sFontCmss44b

6.138 grlib/fonts/fontcmss44i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss44i

6.138.1 Variable Documentation

6.138.1.1 const tFont g_sFontCmss44i

6.139 grlib/fonts/fontcmss46.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss46

6.139.1 Variable Documentation

6.139.1.1 const tFont g_sFontCmss46

6.140 grlib/fonts/fontcmss46b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss46b

6.140.1 Variable Documentation

6.140.1.1 const tFont g_sFontCmss46b

6.141 grlib/fonts/fontcmss46i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss46i

6.141.1 Variable Documentation

6.141.1.1 const tFont g_sFontCmss46i

6.142 grlib/fonts/fontcmss48.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss48

6.142.1 Variable Documentation

6.142.1.1 const tFont g_sFontCmss48

6.143 grlib/fonts/fontcmss48b.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss48b

6.143.1 Variable Documentation

6.143.1.1 const tFont g_sFontCmss48b

6.144 grlib/fonts/fontcmss48i.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontCmss48i

6.144.1 Variable Documentation

6.144.1.1 const tFont g_sFontCmss48i

6.145 grlib/fonts/fontfixed6x8.c File Reference

```
#include "..\grlib.h"
```

Variables

- const tFont g_sFontFixed6x8

6.145.1 Variable Documentation

6.145.1.1 const tFont g_sFontFixed6x8

6.146 grlib/image.c File Reference

```
#include "grlib.h"
```

Functions

- void [GrlImageDraw](#) (const tContext *pContext, const unsigned char *puclImage, unsigned IX, unsigned IY)

6.146.1 Function Documentation

6.146.1.1 void [GrlImageDraw](#) (const tContext * *pContext*, const unsigned char * *puclImage*, unsigned *IX*, unsigned *IY*)

Draws a bitmap image.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>puclImage</i>	is a pointer to the image to draw.
<i>IX</i>	is the X coordinate of the upper left corner of the image.
<i>IY</i>	is the Y coordinate of the upper left corner of the image.

This function draws a bitmap image. The image may be 1 bit per pixel (using the foreground and background color from the drawing context), 4 bits per pixel (using a palette supplied in the image data), or 8 bits per pixel (using a palette supplied in the image data). It can be uncompressed data, or it can be compressed using the Lempel-Ziv--Storer-Szymanski algorithm (as published in the Journal of the ACM, 29(4):928-951, October 1982).

Returns

None.

6.147 grlib/imgbutton.c File Reference

```
#include "grlib.h" #include "widget.h" #include "imgbutton.h"
```

Functions

- int `ImageButtonMsgProc` (`tWidget` **pWidget*, unsigned *ulMsg*, unsigned *ulParam1*, unsigned *ulParam2*)
- void `ImageButtonInit` (`tImageButtonWidget` **pWidget*, const `tDisplay` **pDisplay*, unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)

6.147.1 Function Documentation

6.147.1.1 void `ImageButtonInit` (`tImageButtonWidget` **pWidget*, const `tDisplay` **pDisplay*, unsigned *IX*, unsigned *IY*, unsigned *IWidth*, unsigned *IHeight*)

Initializes an image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the upper left corner of the image button.
<i>IY</i>	is the Y coordinate of the upper left corner of the image button.
<i>IWidth</i>	is the width of the image button.
<i>IHeight</i>	is the height of the image button.

This function initializes the provided image button widget.

Returns

None.

6.147.1.2 int `ImageButtonMsgProc` (`tWidget` **pWidget*, unsigned *ulMsg*, unsigned *ulParam1*, unsigned *ulParam2*)

Handles messages for an image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this image button widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling `WidgetDefaultMsgProc()`.

Returns

Returns a value appropriate to the supplied message.

6.148 grlib/line.c File Reference

```
#include "grlib.h"
```

Functions

- void [GrLineDrawH](#) (const [tContext](#) *pContext, unsigned IX1, unsigned IX2, unsigned IY)
- void [GrLineDrawV](#) (const [tContext](#) *pContext, unsigned IX, unsigned IY1, unsigned IY2)
- void [GrLineDraw](#) (const [tContext](#) *pContext, unsigned IX1, unsigned IY1, unsigned IX2, unsigned IY2)

6.148.1 Function Documentation

6.148.1.1 void GrLineDraw (const [tContext](#) * pContext, unsigned IX1, unsigned IY1, unsigned IX2, unsigned IY2)

Draws a line.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX1</i>	is the X coordinate of the start of the line.
<i>IY1</i>	is the Y coordinate of the start of the line.
<i>IX2</i>	is the X coordinate of the end of the line.
<i>IY2</i>	is the Y coordinate of the end of the line.

This function draws a line, utilizing [GrLineDrawH\(\)](#) and [GrLineDrawV\(\)](#) to draw the line as efficiently as possible. The line is clipped to the clipping rectangle using the Cohen-Sutherland clipping algorithm, and then scan converted using Bresenham's line drawing algorithm.

Returns

None.

6.148.1.2 void GrLineDrawH (const [tContext](#) * pContext, unsigned IX1, unsigned IX2, unsigned IY)

Draws a horizontal line.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX1</i>	is the X coordinate of one end of the line.
<i>IX2</i>	is the X coordinate of the other end of the line.
<i>Y</i>	is the Y coordinate of the line.

This function draws a horizontal line, taking advantage of the fact that the line is horizontal to draw it more efficiently. The clipping of the horizontal line to the clipping rectangle is performed within this routine; the display driver's horizontal line routine is used to perform the actual line drawing.

Returns

None.

6.148.1.3 void GrLineDrawV (const tContext * *pContext*, unsigned *IX*, unsigned *Y1*, unsigned *Y2*)

Draws a vertical line.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX</i>	is the X coordinate of the line.
<i>Y1</i>	is the Y coordinate of one end of the line.
<i>Y2</i>	is the Y coordinate of the other end of the line.

This function draws a vertical line, taking advantage of the fact that the line is vertical to draw it more efficiently. The clipping of the vertical line to the clipping rectangle is performed within this routine; the display driver's vertical line routine is used to perform the actual line drawing.

Returns

None.

6.149 grlib/listbox.c File Reference

```
#include "grlib.h" #include "widget.h" #include "listbox.-  
h"
```

Defines

- #define **abs**(a) (((a) >= 0) ? (a) : (-(a)))
- #define **min**(a, b) (((a) < (b)) ? (a) : (b))
- #define **max**(a, b) (((a) < (b)) ? (b) : (a))

Functions

- int [ListBoxMsgProc](#) (tWidget *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void [ListBoxInit](#) (tListBoxWidget *pWidget, const tDisplay *pDisplay, const char **ppcText, unsigned short usMaxEntries, unsigned short usPopulatedEntries, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)
- int [ListBoxTextAdd](#) (tListBoxWidget *pListBox, const char *pcTxt)

6.149.1 Define Documentation

6.149.1.1 #define **abs(a)** (((a) >= 0) ? (a) : (-(a)))

6.149.1.2 #define **max(a, b)** (((a) < (b)) ? (b) : (a))

6.149.1.3 #define **min(a, b)** (((a) < (b)) ? (a) : (b))

6.150 grlib/offscr1bpp.c File Reference

```
#include "grlib.h"
```

Defines

- #define [DPYCOLORTRANSLATE\(c\)](#)

Functions

- void [GrOffScreen1BPPInit](#) (tDisplay *pDisplay, unsigned char *puImage, long IWidth, long IHeight)

6.151 grlib/offscr4bpp.c File Reference

```
#include "grlib.h"
```

Functions

- void [GrOffScreen4BPPInit](#) (tDisplay *pDisplay, unsigned char *puImage, long IWidth, long IHeight)
- void [GrOffScreen4BPPPaletteSet](#) (tDisplay *pDisplay, unsigned long *pulPalette, unsigned long ulOffset, unsigned long ulCount)

6.152 grlib/offscr8bpp.c File Reference

```
#include "grlib.h"
```

Functions

- void `GrOffScreen8BPPInit` (`tDisplay` *`pDisplay`, unsigned char *`puclImage`, long `lWidth`, long `lHeight`)
- void `GrOffScreen8BPPPPaletteSet` (`tDisplay` *`pDisplay`, unsigned long *`pulPalette`, unsigned long `ulOffset`, unsigned long `ulCount`)

6.153 grlib/pushbutton.c File Reference

```
#include "grlib.h" #include "widget.h" #include "pushbutton.-h"
```

Functions

- int `RectangularButtonMsgProc` (`tWidget` *`pWidget`, unsigned `ulMsg`, unsigned `ulParam1`, unsigned `ulParam2`)
- void `RectangularButtonInit` (`tPushButtonWidget` *`pWidget`, const `tDisplay` *`pDisplay`, unsigned `IX`, unsigned `IY`, unsigned `lWidth`, unsigned `lHeight`)
- int `CircularButtonMsgProc` (`tWidget` *`pWidget`, unsigned `ulMsg`, unsigned `ulParam1`, unsigned `ulParam2`)
- void `CircularButtonInit` (`tPushButtonWidget` *`pWidget`, const `tDisplay` *`pDisplay`, unsigned `IX`, unsigned `IY`, unsigned `IR`)

6.154 grlib/radiobutton.c File Reference

```
#include "grlib.h" #include "widget.h" #include "radiobutton.-h"
```

Functions

- int `RadioButtonMsgProc` (`tWidget` *`pWidget`, unsigned `ulMsg`, unsigned `ulParam1`, unsigned `ulParam2`)
- void `RadioButtonInit` (`tRadioButtonWidget` *`pWidget`, const `tDisplay` *`pDisplay`, unsigned `IX`, unsigned `IY`, unsigned `lWidth`, unsigned `lHeight`)

6.155 grlib/rectangle.c File Reference

```
#include "grlib.h"
```

Defines

- #define `min(a, b)` (((a) < (b)) ? (a) : (b))
- #define `max(a, b)` (((a) < (b)) ? (b) : (a))

Functions

- void `GrRectDraw` (const `tContext` *`pContext`, const `tRectangle` *`pRect`)
- void `GrRectFill` (const `tContext` *`pContext`, const `tRectangle` *`pRect`)
- int `GrRectOverlapCheck` (`tRectangle` *`psRect1`, `tRectangle` *`psRect2`)
- int `GrRectIntersectGet` (`tRectangle` *`psRect1`, `tRectangle` *`psRect2`, `tRectangle` *`psIntersect`)

6.155.1 Define Documentation

6.155.1.1 #define `max(a, b)` (((a) < (b)) ? (b) : (a))

6.155.1.2 #define `min(a, b)` (((a) < (b)) ? (a) : (b))

6.155.2 Function Documentation

6.155.2.1 void `GrRectDraw` (const `tContext` * `pContext`, const `tRectangle` * `pRect`)

Draws a rectangle.

Parameters

<code>pContext</code>	is a pointer to the drawing context to use.
<code>pRect</code>	is a pointer to the structure containing the extents of the rectangle.

This function draws a rectangle. The rectangle will extend from `IXMin` to `IXMax` and `IYMin` to `IYMax`, inclusive.

Returns

None.

6.155.2.2 void `GrRectFill` (const `tContext` * `pContext`, const `tRectangle` * `pRect`)

Draws a filled rectangle.

Parameters

<code>pContext</code>	is a pointer to the drawing context to use.
<code>pRect</code>	is a pointer to the structure containing the extents of the rectangle.

This function draws a filled rectangle. The rectangle will extend from *IXMin* to *IXMax* and *IYMin* to *IYMax*, inclusive. The clipping of the rectangle to the clipping rectangle is performed within this routine; the display driver's rectangle fill routine is used to perform the actual rectangle fill.

Returns

None.

6.155.2.3 int GrRectIntersectGet (tRectangle * psRect1, tRectangle * psRect2, tRectangle * psIntersect)

Determines the intersection of two rectangles.

Parameters

<i>psRect1</i>	is a pointer to the first rectangle.
<i>psRect2</i>	is a pointer to the second rectangle.
<i>psIntersect</i>	is a pointer to a rectangle which will be written with the intersection of <i>psRect1</i> and <i>psRect2</i> .

This function determines if two rectangles overlap and, if they do, calculates the rectangle representing their intersection. If the rectangles do not overlap, 0 is returned and *psIntersect* is not written.

Returns

Returns 1 if there is an overlap or 0 if not.

6.155.2.4 int GrRectOverlapCheck (tRectangle * psRect1, tRectangle * psRect2)

Determines if two rectangles overlap.

Parameters

<i>psRect1</i>	is a pointer to the first rectangle.
<i>psRect2</i>	is a pointer to the second rectangle.

This function determines whether two rectangles overlap. It assumes that rectangles *psRect1* and *psRect2* are valid with *sXMin* < *sXMax* and *sYMin* < *sYMax*.

Returns

Returns 1 if there is an overlap or 0 if not.

6.156 grlib/slider.c File Reference

```
#include <grlib.h> #include "widget.h" #include "slider.h"
```

Defines

- #define **min**(a, b) (((a) < (b)) ? (a) : (b))
- #define **max**(a, b) (((a) < (b)) ? (b) : (a))

Functions

- int **SliderMsgProc** (tWidget *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void **SliderInit** (tSliderWidget *pWidget, const tDisplay *pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

6.156.1 Define Documentation

6.156.1.1 #define **max(a, b)** (((a) < (b)) ? (b) : (a))

6.156.1.2 #define **min(a, b)** (((a) < (b)) ? (a) : (b))

6.156.2 Function Documentation

6.156.2.1 void **SliderInit** (tSliderWidget * pWidget, const tDisplay * pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

Initializes a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the slider.
<i>IX</i>	is the X coordinate of the upper left corner of the slider.
<i>IY</i>	is the Y coordinate of the upper left corner of the slider.
<i>IWidth</i>	is the width of the slider.
<i>IHeight</i>	is the height of the slider.

This function initializes the provided slider widget.

Returns

None.

6.156.2.2 int SliderMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this slider widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.157 grlib/string.c File Reference

```
#include "grlib.h"
```

Defines

- #define **ABSENT_CHAR_REPLACEMENT** ''
- #define **SC_MAX_INDEX** 2047
- #define **SC_IS_NULL** 0xffff
- #define **SC_GET_LEN(v)** ((v) >> (15 - 5))
- #define **SC_GET_INDEX(v)** (((v) >> 8) & **SC_MAX_INDEX**)
- #define **SC_GET_OFF(v)** ((v) & **SC_IS_NULL**)
- #define **SC_FLAG_COMPRESSED** 0x8000
- #define **SC_OFFSET_M** 0x7fff

Functions

- unsigned char **NumLeadingZeros** (unsigned long x)
- int **GrStringWidthGet** (const **tContext** *pContext, const char *pcString, int Length)

- void `GrStringDraw` (const `tContext` **pContext*, const char **pcString*, int *lLength*, unsigned *IX*, unsigned *IY*, int *bOpaque*)
- void `GrStringTableSet` (const void **pvTable*)
- int `GrStringLanguageSet` (unsigned short *usLangID*)
- unsigned `GrStringGet` (int *iIndex*, char **pcData*, unsigned *ulSize*)

6.157.1 Define Documentation

```
6.157.1.1 #define ABSENT_CHAR_REPLACEMENT ' '

6.157.1.2 #define SC_FLAG_COMPRESSED 0x8000

6.157.1.3 #define SC_GET_INDEX( v ) (((v) >> 8) & SC_MAX_INDEX)

6.157.1.4 #define SC_GET_LEN( v ) ((v) >> (15 - 5))

6.157.1.5 #define SC_GET_OFF( v ) ((v) & SC_IS_NULL)

6.157.1.6 #define SC_IS_NULL 0xffff

6.157.1.7 #define SC_MAX_INDEX 2047

6.157.1.8 #define SC_OFFSET_M 0x7fff
```

6.157.2 Function Documentation

```
6.157.2.1 void GrStringDraw ( const tContext * pContext, const char * pcString, int
    lLength, unsigned IX, unsigned IY, int bOpaque )
```

Draws a string.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>pcString</i>	is a pointer to the string to be drawn.
<i>lLength</i>	is the number of characters from the string that should be drawn on the screen.
<i>IX</i>	is the X coordinate of the upper left corner of the string position on the screen.
<i>IY</i>	is the Y coordinate of the upper left corner of the string position on the screen.
<i>bOpaque</i>	is true if the background of each character should be drawn and false if it should not (leaving the background as is).

This function draws a string of text on the screen. The *lLength* parameter allows a portion of the string to be examined without having to insert a NULL character at the stopping point (which would not be possible if the string was located in flash); specifying a length of -1 will cause the entire string to be rendered (subject to clipping).

Returns

None.

6.157.2.2 unsigned GrStringGet (int *iIndex*, char * *pcData*, unsigned *ulSize*)

This function returns a string from the current string table.

Parameters

<i>iIndex</i>	is the index of the string to retrieve.
<i>pcData</i>	is the pointer to the buffer to store the string into.
<i>ulSize</i>	is the size of the buffer provided by <i>pcData</i> .

This function will return a string from the string table in the language set by the [GrStringLanguageSet\(\)](#) function. The value passed in *iIndex* parameter is the string that is being requested and will be returned in the buffer provided in the *pcData* parameter. The amount of data returned will be limited by the *ulSize* parameter.

Returns

Returns the number of valid bytes returned in the *pcData* buffer.

6.157.2.3 int GrStringLanguageSet (unsigned short *usLangID*)

This function sets the current language for strings returned by the [GrStringGet\(\)](#) function.

Parameters

<i>usLangID</i>	is one of the language identifiers provided in the string table.
-----------------	--

This function is used to set the language identifier for the strings returned by the [GrStringGet\(\)](#) function. The *usLangID* parameter should match one of the identifiers that was included in the string table. These are provided in a header file in the graphics library and must match the values that were passed through the sting compression utility.

Returns

This function returns 0 if the language was not found and a non-zero value if the lagauge was found.

6.157.2.4 void GrStringTableSet (const void * *pvTable*)

This function sets the location of the current string table.

Parameters

<i>pvTable</i>	is a pointer to a string table that was generated by the string compression utility.
----------------	--

This function is used to set the string table to use for strings in an application. This string table is created by the string compression utility. This function is used to swap out multiple string tables if the application requires more than one table. It does not allow using more than one string table at a time.

Returns

None.

6.157.2.5 int GrStringWidthGet (const tContext * *pContext*, const char * *pcString*, int *lLength*)

Determines the width of a string.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>pcString</i>	is the string in question.
<i>lLength</i>	is the length of the string.

This function determines the width of a string (or portion of the string) when drawn with a particular font. The *lLength* parameter allows a portion of the string to be examined without having to insert a NULL character at the stopping point (would not be possible if the string was located in flash); specifying a length of -1 will cause the width of the entire string to be computed.

Returns

Returns the width of the string in pixels.

6.157.2.6 unsigned char NumLeadingZeros (unsigned long *x*)

6.158 grlib/widget.c File Reference

```
#include "grlib.h" #include "widget.h"
```

Data Structures

- struct [tWidgetMessageQueue](#)

Defines

- #define MQ_FLAG_POST_ORDER 1
- #define MQ_FLAG_STOP_ON_SUCCESS 2
- #define QUEUE_SIZE 16

Functions

- void WidgetMutexInit (int *pcMutex)
- int WidgetMutexGet (int *pcMutex)
- void WidgetMutexPut (int *pcMutex)
- int WidgetDefaultMsgProc (tWidget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2)
- void WidgetAdd (tWidget *pParent, tWidget *pWidget)
- void WidgetRemove (tWidget *pWidget)
- int WidgetMessageSendPreOrder (tWidget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2, int bStopOnSuccess)
- int WidgetMessageSendPostOrder (tWidget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2, int bStopOnSuccess)
- int WidgetMessageQueueAdd (tWidget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2, unsigned bPostOrder, int bStopOnSuccess)
- void WidgetMessageQueueProcess (void)
- int WidgetPointerMessage (unsigned ulMessage, unsigned IX, unsigned IY)

Variables

- tWidget g_sRoot

6.158.1 Define Documentation

6.158.1.1 #define MQ_FLAG_POST_ORDER 1

6.158.1.2 #define MQ_FLAG_STOP_ON_SUCCESS 2

6.158.1.3 #define QUEUE_SIZE 16

6.158.2 Function Documentation

6.158.2.1 void WidgetAdd (tWidget * pParent, tWidget * pWidget)

Adds a widget to the widget tree.

Parameters

<i>pParent</i>	is the parent for the widget. To add to the root of the tree set this parameter to WIDGET_ROOT .
<i>pWidget</i>	is the widget to add.

Generated on Wed Oct 24 2012 22:48:50 for GLCD by Doxygen

This function adds a widget to the widget tree at the given position within the tree. The widget will become the last child of its parent, and will therefore be searched after the existing children.

The added widget can be a full widget tree, allowing addition of an entire hierarchy all at once (for example, adding an entire screen to the widget tree all at once). In this case, it is the responsibility of the caller to ensure that the pParent field of each widget in the added tree is correctly set (in other words, only the widget pointed to by *pWidget* is updated to properly reside in the tree).

It is the responsibility of the caller to initialize the pNext and pChild field of the added widget; either of these fields being non-zero results in a pre-defined tree of widgets being added instead of a single one.

Returns

None.

**6.158.2.2 int WidgetDefaultMsgProc (tWidget * *pWidget*, unsigned *ulMessage*,
unsigned *ulParam1*, unsigned *ulParam2*)**

Handles widget messages.

Parameters

<i>pWidget</i>	is a pointer to the widget.
<i>ulMessage</i>	is the message to be processed.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function is a default handler for widget messages; it simply ignores all messages sent to it. This is used as the message handler for the root widget, and should be called by the message handler for other widgets when they do not explicitly handle the provided message (in case new messages are added that require some default but override-able processing).

Returns

Always returns 0.

**6.158.2.3 int WidgetMessageQueueAdd (tWidget * *pWidget*, unsigned *ulMessage*,
unsigned *ulParam1*, unsigned *ulParam2*, unsigned *bPostOrder*, int *bStopOnSuccess*
)**

Adds a message to the widget message queue.

Parameters

<i>pWidget</i>	is the widget to which the message should be sent.
<i>ulMessage</i>	is the message to be sent.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.
<i>bPostOrder</i>	is true if the message should be sent via a post-order search, and false if it should be sent via a pre-order search.
<i>bStopOn-Success</i>	is true if the message should be sent to widgets until one returns success, and false if it should be sent to all widgets.

This function places a widget message into the message queue for later processing. The messages are removed from the queue by [WidgetMessageQueueProcess\(\)](#) and sent to the appropriate place.

It is safe for code which interrupts [WidgetMessageQueueProcess\(\)](#) (or called by it) to call this function to send a message. It is not safe for code which interrupts this function to call this function as well; it is up to the caller to guarantee that the later sequence never occurs.

Returns

Returns 1 if the message was added to the queue, and 0 if it could not be added since either the queue is full or another context is currently adding a message to the queue.

6.158.2.4 void WidgetMessageQueueProcess(void)

Processes the messages in the widget message queue.

This function extracts messages from the widget message queue one at a time and processes them. If the processing of a widget message requires that a new message be sent, it is acceptable to call [WidgetMessageQueueAdd\(\)](#). It is also acceptable for code which interrupts this function to call [WidgetMessageQueueAdd\(\)](#) to send more messages. In both cases, the newly added message will also be processed before this function returns.

Returns

None.

6.158.2.5 int WidgetMessageSendPostOrder(tWidget * pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2, int bStopOnSuccess)

Sends a message to a widget tree via a post-order, depth-first search.

Parameters

<i>pWidget</i>	is a pointer to the widget tree; if this is zero then the root of the widget tree will be used.
<i>ulMessage</i>	is the message to send.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.
<i>bStopOnSuccess</i>	is true if the search should be stopped when the first widget is found that returns success in response to the message.

This function performs a post-order, depth-first search of the widget tree, sending a message to each widget encountered. In a depth-first search, the children of a widget are searched before its sibling (preferring to go deeper into the tree, hence the name depth-first). A post-order search means that the message is sent to a widget after all of its children are searched.

An example use of the post-order search is for pointer-related messages; those messages should be delivered to the lowest widget in the tree before its parents (in other words, the widget deepest in the tree that has a hit should get the message, not the higher up widgets that also include the hit location).

Special handling is performed for pointer-related messages. The widget that accepts **WIDGET_MSG_PTR_DOWN** is remembered and subsequent **WIDGET_MSG_PTR_MOVE** and **WIDGET_MSG_PTR_UP** messages are sent directly to that widget.

Returns

Returns 0 if *bStopOnSuccess* is false or no widget returned success in response to the message, or the value returned by the first widget to successfully process the message.

6.158.2.6 int WidgetMessageSendPreOrder (tWidget * *pWidget*, unsigned *ulMessage*, unsigned *ulParam1*, unsigned *ulParam2*, int *bStopOnSuccess*)

Sends a message to a widget tree via a pre-order, depth-first search.

Parameters

<i>pWidget</i>	is a pointer to the widget tree.
<i>ulMessage</i>	is the message to send.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.
<i>bStopOnSuccess</i>	is true if the search should be stopped when the first widget is found that returns success in response to the message.

This function performs a pre-order, depth-first search of the widget tree, sending a message to each widget encountered. In a depth-first search, the children of a widget are searched before its siblings (preferring to go deeper into the tree, hence the name depth-first). A pre-order search means that the message is sent to a widget before any of its children are searched.

An example use of the pre-order search is for paint messages; the larger enclosing widgets should be drawn on the screen before the smaller widgets that reside within the parent widget (otherwise, the children would be overwritten by the parent).

Returns

Returns 0 if *bStopOnSuccess* is false or no widget returned success in response to the message, or the value returned by the first widget to successfully process the message.

6.158.2.7 int WidgetMutexGet (int * *pcMutex*)

Attempts to acquire a mutex.

Parameters

<i>pcMutex</i>	is a pointer to mutex that is to be acquired.
----------------	---

This function attempts to acquire a mutual exclusion semaphore (mutex) on behalf of the caller. If the mutex is not already held, 0 is returned to indicate that the caller may safely access whichever resource the mutex is protecting. If the mutex is already held, 1 is returned and the caller must not access the shared resource.

When access to the shared resource is complete, the mutex owner should call [WidgetMutexPut\(\)](#) to release the mutex and relinquish ownership of the shared resource.

Returns

Returns 0 if the mutex is acquired successfully or 1 if it is already held by another caller.

6.158.2.8 void WidgetMutexInit (int * *pcMutex*)

Initializes a mutex to the unowned state.

Parameters

<i>pcMutex</i>	is a pointer to mutex that is to be initialized.
----------------	--

This function initializes a mutual exclusion semaphore (mutex) to its unowned state in preparation for use with [WidgetMutexGet\(\)](#) and [WidgetMutexPut\(\)](#). A mutex is a two state object typically used to serialize access to a shared resource. An application will call [WidgetMutexGet\(\)](#) to request ownership of the mutex. If ownership is granted, the caller may safely access the resource then release the mutex using [WidgetMutexPut\(\)](#) once it is finished. If ownership is not granted, the caller knows that some other context is currently modifying the shared resource and it must not access the resource at that time.

Note that this function must not be called if the mutex passed in *pcMutex* is already in use since this will have the effect of releasing the lock even if some caller currently owns it.

Returns

None.

6.158.2.9 void WidgetMutexPut (int * *pcMutex*)

Release a mutex.

Parameters

<i>pcMutex</i>	is a pointer to mutex that is to be released.
----------------	---

This function releases a mutual exclusion semaphore (mutex), leaving it in the unowned state.

Returns

None.

6.158.2.10 int WidgetPointerMessage (unsigned *ulMessage*, unsigned *IX*, unsigned *IY*)

Sends a pointer message.

Parameters

<i>ulMessage</i>	is the pointer message to be sent.
<i>IX</i>	is the X coordinate associated with the message.
<i>IY</i>	is the Y coordinate associated with the message.

This function sends a pointer message to the root widget. A pointer driver (such as a touch screen driver) can use this function to deliver pointer activity to the widget tree without having to have direct knowledge of the structure of the widget framework.

Returns

Returns 1 if the message was added to the queue, and 0 if it could not be added since the queue is full.

6.158.2.11 void WidgetRemove (tWidget * *pWidget*)

Removes a widget from the widget tree.

Parameters

<i>pWidget</i>	is the widget to be removed.
----------------	------------------------------

This function removes a widget from the widget tree. The removed widget can be a full widget tree, allowing removal of an entire hierarchy all at once (for example, removing an entire screen from the widget tree).

Returns

None.

6.158.3 Variable Documentation

6.158.3.1 tWidget g_sRoot

Initial value:

```
{  
    sizeof(tWidget),  
    0,  
    0,  
    0,  
    0,  
    {  
        0,  
        0,  
        0,  
        0,  
    },  
    WidgetDefaultMsgProc  
}
```

6.159 include/build/lpc210x.h File Reference

Defines

- #define **VICIRQStatus** (*((volatile unsigned long *) 0xFFFFF000))
- #define **VICFIQStatus** (*((volatile unsigned long *) 0xFFFFF004))
- #define **VICRawIntr** (*((volatile unsigned long *) 0xFFFFF008))
- #define **VICIntSelect** (*((volatile unsigned long *) 0xFFFFF00C))
- #define **VICIntEnable** (*((volatile unsigned long *) 0xFFFFF010))
- #define **VICIntEnClr** (*((volatile unsigned long *) 0xFFFFF014))
- #define **VICSoftInt** (*((volatile unsigned long *) 0xFFFFF018))
- #define **VICSoftIntClr** (*((volatile unsigned long *) 0xFFFFF01C))
- #define **VICProtection** (*((volatile unsigned long *) 0xFFFFF020))
- #define **VICVectAddr** (*((volatile unsigned long *) 0xFFFFF030))
- #define **VICDefVectAddr** (*((volatile unsigned long *) 0xFFFFF034))
- #define **VICVectAddr0** (*((volatile unsigned long *) 0xFFFFF100))

```
• #define VICVectAddr1 (*((volatile unsigned long *) 0xFFFFF104))
• #define VICVectAddr2 (*((volatile unsigned long *) 0xFFFFF108))
• #define VICVectAddr3 (*((volatile unsigned long *) 0xFFFFF10C))
• #define VICVectAddr4 (*((volatile unsigned long *) 0xFFFFF110))
• #define VICVectAddr5 (*((volatile unsigned long *) 0xFFFFF114))
• #define VICVectAddr6 (*((volatile unsigned long *) 0xFFFFF118))
• #define VICVectAddr7 (*((volatile unsigned long *) 0xFFFFF11C))
• #define VICVectAddr8 (*((volatile unsigned long *) 0xFFFFF120))
• #define VICVectAddr9 (*((volatile unsigned long *) 0xFFFFF124))
• #define VICVectAddr10 (*((volatile unsigned long *) 0xFFFFF128))
• #define VICVectAddr11 (*((volatile unsigned long *) 0xFFFFF12C))
• #define VICVectAddr12 (*((volatile unsigned long *) 0xFFFFF130))
• #define VICVectAddr13 (*((volatile unsigned long *) 0xFFFFF134))
• #define VICVectAddr14 (*((volatile unsigned long *) 0xFFFFF138))
• #define VICVectAddr15 (*((volatile unsigned long *) 0xFFFFF13C))
• #define VICVectCntl0 (*((volatile unsigned long *) 0xFFFFF200))
• #define VICVectCntl1 (*((volatile unsigned long *) 0xFFFFF204))
• #define VICVectCntl2 (*((volatile unsigned long *) 0xFFFFF208))
• #define VICVectCntl3 (*((volatile unsigned long *) 0xFFFFF20C))
• #define VICVectCntl4 (*((volatile unsigned long *) 0xFFFFF210))
• #define VICVectCntl5 (*((volatile unsigned long *) 0xFFFFF214))
• #define VICVectCntl6 (*((volatile unsigned long *) 0xFFFFF218))
• #define VICVectCntl7 (*((volatile unsigned long *) 0xFFFFF21C))
• #define VICVectCntl8 (*((volatile unsigned long *) 0xFFFFF220))
• #define VICVectCntl9 (*((volatile unsigned long *) 0xFFFFF224))
• #define VICVectCntl10 (*((volatile unsigned long *) 0xFFFFF228))
• #define VICVectCntl11 (*((volatile unsigned long *) 0xFFFFF22C))
• #define VICVectCntl12 (*((volatile unsigned long *) 0xFFFFF230))
• #define VICVectCntl13 (*((volatile unsigned long *) 0xFFFFF234))
• #define VICVectCntl14 (*((volatile unsigned long *) 0xFFFFF238))
• #define VICVectCntl15 (*((volatile unsigned long *) 0xFFFFF23C))
• #define PINSEL0 (*((volatile unsigned long *) 0xE002C000))
• #define PINSEL1 (*((volatile unsigned long *) 0xE002C004))
• #define IOPIN (*((volatile unsigned long *) 0xE0028000))
• #define IOSET (*((volatile unsigned long *) 0xE0028004))
• #define IODIR (*((volatile unsigned long *) 0xE0028008))
• #define IOCLR (*((volatile unsigned long *) 0xE002800C))
• #define MAMCR (*((volatile unsigned char *) 0xE01FC000))
• #define MAMTIM (*((volatile unsigned char *) 0xE01FC004))
• #define MAMMAP (*((volatile unsigned char *) 0xE01FC040))
• #define PLLCON (*((volatile unsigned char *) 0xE01FC080))
• #define PLLCFG (*((volatile unsigned char *) 0xE01FC084))
• #define PLLSTAT (*((volatile unsigned short*) 0xE01FC088))
• #define PLLFEED (*((volatile unsigned char *) 0xE01FC08C))
• #define VPBDIV (*((volatile unsigned char *) 0xE01FC100))
• #define PCON (*((volatile unsigned char *) 0xE01FC0C0))
```

- #define **PCONP** (*((volatile unsigned long *) 0xE01FC0C4))
- #define **EXTINT** (*((volatile unsigned char *) 0xE01FC140))
- #define **EXTWAKE** (*((volatile unsigned char *) 0xE01FC144))
- #define **T0IR** (*((volatile unsigned long *) 0xE0004000))
- #define **TOTCR** (*((volatile unsigned long *) 0xE0004004))
- #define **TOTC** (*((volatile unsigned long *) 0xE0004008))
- #define **TOPR** (*((volatile unsigned long *) 0xE000400C))
- #define **TOPC** (*((volatile unsigned long *) 0xE0004010))
- #define **TOMCR** (*((volatile unsigned long *) 0xE0004014))
- #define **TOMR0** (*((volatile unsigned long *) 0xE0004018))
- #define **TOMR1** (*((volatile unsigned long *) 0xE000401C))
- #define **TOMR2** (*((volatile unsigned long *) 0xE0004020))
- #define **TOMR3** (*((volatile unsigned long *) 0xE0004024))
- #define **TOCCR** (*((volatile unsigned long *) 0xE0004028))
- #define **TOCR0** (*((volatile unsigned long *) 0xE000402C))
- #define **TOCR1** (*((volatile unsigned long *) 0xE0004030))
- #define **TOCR2** (*((volatile unsigned long *) 0xE0004034))
- #define **TOCR3** (*((volatile unsigned long *) 0xE0004038))
- #define **TOEMR** (*((volatile unsigned long *) 0xE000403C))
- #define **T1IR** (*((volatile unsigned long *) 0xE0008000))
- #define **T1TCR** (*((volatile unsigned long *) 0xE0008004))
- #define **T1TC** (*((volatile unsigned long *) 0xE0008008))
- #define **T1PR** (*((volatile unsigned long *) 0xE000800C))
- #define **T1PC** (*((volatile unsigned long *) 0xE0008010))
- #define **T1MCR** (*((volatile unsigned long *) 0xE0008014))
- #define **T1MR0** (*((volatile unsigned long *) 0xE0008018))
- #define **T1MR1** (*((volatile unsigned long *) 0xE000801C))
- #define **T1MR2** (*((volatile unsigned long *) 0xE0008020))
- #define **T1MR3** (*((volatile unsigned long *) 0xE0008024))
- #define **T1CCR** (*((volatile unsigned long *) 0xE0008028))
- #define **T1CR0** (*((volatile unsigned long *) 0xE000802C))
- #define **T1CR1** (*((volatile unsigned long *) 0xE0008030))
- #define **T1CR2** (*((volatile unsigned long *) 0xE0008034))
- #define **T1CR3** (*((volatile unsigned long *) 0xE0008038))
- #define **T1EMR** (*((volatile unsigned long *) 0xE000803C))
- #define **PWM_IR** (*((volatile unsigned long *) 0xE0014000))
- #define **PWM_TCR** (*((volatile unsigned long *) 0xE0014004))
- #define **PWM_TC** (*((volatile unsigned long *) 0xE0014008))
- #define **PWM_PR** (*((volatile unsigned long *) 0xE001400C))
- #define **PWM_PC** (*((volatile unsigned long *) 0xE0014010))
- #define **PWM_MCR** (*((volatile unsigned long *) 0xE0014014))
- #define **PWM_MR0** (*((volatile unsigned long *) 0xE0014018))
- #define **PWM_MR1** (*((volatile unsigned long *) 0xE001401C))
- #define **PWM_MR2** (*((volatile unsigned long *) 0xE0014020))
- #define **PWM_MR3** (*((volatile unsigned long *) 0xE0014024))
- #define **PWM_MR4** (*((volatile unsigned long *) 0xE0014040))

- #define **PWM_MR5** (*((volatile unsigned long *) 0xE0014044))
- #define **PWM_MR6** (*((volatile unsigned long *) 0xE0014048))
- #define **PWM_CCR** (*((volatile unsigned long *) 0xE0014028))
- #define **PWM_CR0** (*((volatile unsigned long *) 0xE001402C))
- #define **PWM_CR1** (*((volatile unsigned long *) 0xE0014030))
- #define **PWM_CR2** (*((volatile unsigned long *) 0xE0014034))
- #define **PWM_CR3** (*((volatile unsigned long *) 0xE0014038))
- #define **PWM_EMR** (*((volatile unsigned long *) 0xE001403C))
- #define **PWM_PCR** (*((volatile unsigned long *) 0xE001404C))
- #define **PWM_LER** (*((volatile unsigned long *) 0xE0014050))
- #define **U0RBR** (*((volatile unsigned char *) 0xE000C000))
- #define **U0THR** (*((volatile unsigned char *) 0xE000C000))
- #define **U0IER** (*((volatile unsigned char *) 0xE000C004))
- #define **U0IIR** (*((volatile unsigned char *) 0xE000C008))
- #define **U0FCR** (*((volatile unsigned char *) 0xE000C008))
- #define **U0LCR** (*((volatile unsigned char *) 0xE000C00C))
- #define **U0LSR** (*((volatile unsigned char *) 0xE000C014))
- #define **U0SCR** (*((volatile unsigned char *) 0xE000C01C))
- #define **U0DLL** (*((volatile unsigned char *) 0xE000C000))
- #define **U0DLM** (*((volatile unsigned char *) 0xE000C004))
- #define **U0MSR** (*((volatile unsigned char *) 0xE000C018))
- #define **U0MCR** (*((volatile unsigned char *) 0xE000C010))
- #define **U1RBR** (*((volatile unsigned char *) 0xE0010000))
- #define **U1THR** (*((volatile unsigned char *) 0xE0010000))
- #define **U1IER** (*((volatile unsigned char *) 0xE0010004))
- #define **U1IIR** (*((volatile unsigned char *) 0xE0010008))
- #define **U1FCR** (*((volatile unsigned char *) 0xE0010008))
- #define **U1LCR** (*((volatile unsigned char *) 0xE001000C))
- #define **U1MCR** (*((volatile unsigned char *) 0xE0010010))
- #define **U1LSR** (*((volatile unsigned char *) 0xE0010014))
- #define **U1MSR** (*((volatile unsigned char *) 0xE0010018))
- #define **U1SCR** (*((volatile unsigned char *) 0xE001001C))
- #define **U1DLL** (*((volatile unsigned char *) 0xE0010000))
- #define **U1DLM** (*((volatile unsigned char *) 0xE0010004))
- #define **I2C_I2CONSET** (*((volatile unsigned char *) 0xE001C000))
- #define **I2C_I2STAT** (*((volatile unsigned char *) 0xE001C004))
- #define **I2C_I2DAT** (*((volatile unsigned char *) 0xE001C008))
- #define **I2C_I2ADR** (*((volatile unsigned char *) 0xE001C00C))
- #define **I2C_I2SCLH** (*((volatile unsigned short*) 0xE001C010))
- #define **I2C_I2SCLL** (*((volatile unsigned short*) 0xE001C014))
- #define **I2C_I2CONCLR** (*((volatile unsigned char *) 0xE001C018))
- #define **SPI_SPCR** (*((volatile unsigned char *) 0xE0020000))
- #define **SPI_SPSR** (*((volatile unsigned char *) 0xE0020004))
- #define **SPI_SPDR** (*((volatile unsigned char *) 0xE0020008))
- #define **SPI_SPCCR** (*((volatile unsigned char *) 0xE002000C))
- #define **SPI_SPTCR** (*((volatile unsigned char *) 0xE0020010))

- #define **SPI_SPTSR** (*((volatile unsigned char *) 0xE0020014))
- #define **SPI_SPTOR** (*((volatile unsigned char *) 0xE0020018))
- #define **SPI_SPINT** (*((volatile unsigned char *) 0xE002001C))
- #define **ILR** (*((volatile unsigned char *) 0xE0024000))
- #define **CTC** (*((volatile unsigned short*) 0xE0024004))
- #define **CCR** (*((volatile unsigned char *) 0xE0024008))
- #define **CIIR** (*((volatile unsigned char *) 0xE002400C))
- #define **AMR** (*((volatile unsigned char *) 0xE0024010))
- #define **CTIME0** (*((volatile unsigned long *) 0xE0024014))
- #define **CTIME1** (*((volatile unsigned long *) 0xE0024018))
- #define **CTIME2** (*((volatile unsigned long *) 0xE002401C))
- #define **SEC** (*((volatile unsigned char *) 0xE0024020))
- #define **MIN** (*((volatile unsigned char *) 0xE0024024))
- #define **HOUR** (*((volatile unsigned char *) 0xE0024028))
- #define **DOM** (*((volatile unsigned char *) 0xE002402C))
- #define **DOW** (*((volatile unsigned char *) 0xE0024030))
- #define **DOY** (*((volatile unsigned short*) 0xE0024034))
- #define **MONTH** (*((volatile unsigned char *) 0xE0024038))
- #define **YEAR** (*((volatile unsigned short*) 0xE002403C))
- #define **ALSEC** (*((volatile unsigned char *) 0xE0024060))
- #define **ALMIN** (*((volatile unsigned char *) 0xE0024064))
- #define **ALHOUR** (*((volatile unsigned char *) 0xE0024068))
- #define **ALDOM** (*((volatile unsigned char *) 0xE002406C))
- #define **ALDOW** (*((volatile unsigned char *) 0xE0024070))
- #define **ALDOY** (*((volatile unsigned short*) 0xE0024074))
- #define **ALMON** (*((volatile unsigned char *) 0xE0024078))
- #define **ALYEAR** (*((volatile unsigned short*) 0xE002407C))
- #define **PREINT** (*((volatile unsigned short*) 0xE0024080))
- #define **PREFRAC** (*((volatile unsigned short*) 0xE0024084))
- #define **WDMOD** (*((volatile unsigned char *) 0xE0000000))
- #define **WDTC** (*((volatile unsigned long *) 0xE0000004))
- #define **WDFEED** (*((volatile unsigned char *) 0xE0000008))
- #define **WDTV** (*((volatile unsigned long *) 0xE000000C))

6.159.1 Define Documentation

6.159.1.1 #define **ALDOM** (*((volatile unsigned char *) 0xE002406C))

6.159.1.2 #define **ALDOW** (*((volatile unsigned char *) 0xE0024070))

6.159.1.3 #define **ALDOY** (*((volatile unsigned short*) 0xE0024074))

6.159.1.4 #define **ALHOUR** (*((volatile unsigned char *) 0xE0024068))

6.159.1.5 #define **ALMIN** (*((volatile unsigned char *) 0xE0024064))

```
6.159.1.6 #define ALMON (*((volatile unsigned char *) 0xE0024078))

6.159.1.7 #define ALSEC (*((volatile unsigned char *) 0xE0024060))

6.159.1.8 #define ALYEAR (*((volatile unsigned short*) 0xE002407C))

6.159.1.9 #define AMR (*((volatile unsigned char *) 0xE0024010))

6.159.1.10 #define CCR (*((volatile unsigned char *) 0xE0024008))

6.159.1.11 #define CIIR (*((volatile unsigned char *) 0xE002400C))

6.159.1.12 #define CTC (*((volatile unsigned short*) 0xE0024004))

6.159.1.13 #define CTIME0 (*((volatile unsigned long *) 0xE0024014))

6.159.1.14 #define CTIME1 (*((volatile unsigned long *) 0xE0024018))

6.159.1.15 #define CTIME2 (*((volatile unsigned long *) 0xE002401C))

6.159.1.16 #define DOM (*((volatile unsigned char *) 0xE002402C))

6.159.1.17 #define DOW (*((volatile unsigned char *) 0xE0024030))

6.159.1.18 #define DOY (*((volatile unsigned short*) 0xE0024034))

6.159.1.19 #define EXTINT (*((volatile unsigned char *) 0xE01FC140))

6.159.1.20 #define EXTWAKE (*((volatile unsigned char *) 0xE01FC144))

6.159.1.21 #define HOUR (*((volatile unsigned char *) 0xE0024028))

6.159.1.22 #define I2C_I2ADR (*((volatile unsigned char *) 0xE001C00C))

6.159.1.23 #define I2C_I2CONCLR (*((volatile unsigned char *) 0xE001C018))

6.159.1.24 #define I2C_I2CONSET (*((volatile unsigned char *) 0xE001C000))

6.159.1.25 #define I2C_I2DAT (*((volatile unsigned char *) 0xE001C008))

6.159.1.26 #define I2C_I2SCLH (*((volatile unsigned short*) 0xE001C010))

6.159.1.27 #define I2C_I2SCLL (*((volatile unsigned short*) 0xE001C014))

6.159.1.28 #define I2C_I2STAT (*((volatile unsigned char *) 0xE001C004))

6.159.1.29 #define ILR (*((volatile unsigned char *) 0xE0024000))
```

```
6.159.1.30 #define IOCLR (*((volatile unsigned long *) 0xE002800C))

6.159.1.31 #define IODIR (*((volatile unsigned long *) 0xE0028008))

6.159.1.32 #define IOPIN (*((volatile unsigned long *) 0xE0028000))

6.159.1.33 #define IOSET (*((volatile unsigned long *) 0xE0028004))

6.159.1.34 #define MAMCR (*((volatile unsigned char *) 0xE01FC000))

6.159.1.35 #define MAMMAP (*((volatile unsigned char *) 0xE01FC040))

6.159.1.36 #define MAMTIM (*((volatile unsigned char *) 0xE01FC004))

6.159.1.37 #define MIN (*((volatile unsigned char *) 0xE0024024))

6.159.1.38 #define MONTH (*((volatile unsigned char *) 0xE0024038))

6.159.1.39 #define PCON (*((volatile unsigned char *) 0xE01FC0C0))

6.159.1.40 #define PCOMP (*((volatile unsigned long *) 0xE01FC0C4))

6.159.1.41 #define PINSEL0 (*((volatile unsigned long *) 0xE002C000))

6.159.1.42 #define PINSEL1 (*((volatile unsigned long *) 0xE002C004))

6.159.1.43 #define PLLCFG (*((volatile unsigned char *) 0xE01FC084))

6.159.1.44 #define PLLCON (*((volatile unsigned char *) 0xE01FC080))

6.159.1.45 #define PLLFEED (*((volatile unsigned char *) 0xE01FC08C))

6.159.1.46 #define PLLSTAT (*((volatile unsigned short*) 0xE01FC088))

6.159.1.47 #define PREFRAC (*((volatile unsigned short*) 0xE0024084))

6.159.1.48 #define PREINT (*((volatile unsigned short*) 0xE0024080))

6.159.1.49 #define PWM_CCR (*((volatile unsigned long *) 0xE0014028))

6.159.1.50 #define PWM_CRO (*((volatile unsigned long *) 0xE001402C))

6.159.1.51 #define PWM_CR1 (*((volatile unsigned long *) 0xE0014030))

6.159.1.52 #define PWM_CR2 (*((volatile unsigned long *) 0xE0014034))

6.159.1.53 #define PWM_CR3 (*((volatile unsigned long *) 0xE0014038))
```

```
6.159.1.54 #define PWM_EMR (*((volatile unsigned long *) 0xE001403C))

6.159.1.55 #define PWM_IR (*((volatile unsigned long *) 0xE0014000))

6.159.1.56 #define PWM_LER (*((volatile unsigned long *) 0xE0014050))

6.159.1.57 #define PWM_MCR (*((volatile unsigned long *) 0xE0014014))

6.159.1.58 #define PWM_MR0 (*((volatile unsigned long *) 0xE0014018))

6.159.1.59 #define PWM_MR1 (*((volatile unsigned long *) 0xE001401C))

6.159.1.60 #define PWM_MR2 (*((volatile unsigned long *) 0xE0014020))

6.159.1.61 #define PWM_MR3 (*((volatile unsigned long *) 0xE0014024))

6.159.1.62 #define PWM_MR4 (*((volatile unsigned long *) 0xE0014040))

6.159.1.63 #define PWM_MR5 (*((volatile unsigned long *) 0xE0014044))

6.159.1.64 #define PWM_MR6 (*((volatile unsigned long *) 0xE0014048))

6.159.1.65 #define PWM_PC (*((volatile unsigned long *) 0xE0014010))

6.159.1.66 #define PWM_PCR (*((volatile unsigned long *) 0xE001404C))

6.159.1.67 #define PWM_PR (*((volatile unsigned long *) 0xE001400C))

6.159.1.68 #define PWM_TC (*((volatile unsigned long *) 0xE0014008))

6.159.1.69 #define PWM_TCR (*((volatile unsigned long *) 0xE0014004))

6.159.1.70 #define SEC (*((volatile unsigned char *) 0xE0024020))

6.159.1.71 #define SPI_SPCCR (*((volatile unsigned char *) 0xE002000C))

6.159.1.72 #define SPI_SPCR (*((volatile unsigned char *) 0xE0020000))

6.159.1.73 #define SPI_SPDR (*((volatile unsigned char *) 0xE0020008))

6.159.1.74 #define SPI_SPINT (*((volatile unsigned char *) 0xE002001C))

6.159.1.75 #define SPI_SPSR (*((volatile unsigned char *) 0xE0020004))

6.159.1.76 #define SPI_SPTCR (*((volatile unsigned char *) 0xE0020010))

6.159.1.77 #define SPI_SPTOR (*((volatile unsigned char *) 0xE0020018))
```

```
6.159.1.78 #define SPI_SPTSR (*((volatile unsigned char *) 0xE0020014))

6.159.1.79 #define T0CCR (*((volatile unsigned long *) 0xE0004028))

6.159.1.80 #define T0CR0 (*((volatile unsigned long *) 0xE000402C))

6.159.1.81 #define T0CR1 (*((volatile unsigned long *) 0xE0004030))

6.159.1.82 #define T0CR2 (*((volatile unsigned long *) 0xE0004034))

6.159.1.83 #define T0CR3 (*((volatile unsigned long *) 0xE0004038))

6.159.1.84 #define T0EMR (*((volatile unsigned long *) 0xE000403C))

6.159.1.85 #define T0IR (*((volatile unsigned long *) 0xE0004000))

6.159.1.86 #define T0MCR (*((volatile unsigned long *) 0xE0004014))

6.159.1.87 #define T0MR0 (*((volatile unsigned long *) 0xE0004018))

6.159.1.88 #define T0MR1 (*((volatile unsigned long *) 0xE000401C))

6.159.1.89 #define T0MR2 (*((volatile unsigned long *) 0xE0004020))

6.159.1.90 #define T0MR3 (*((volatile unsigned long *) 0xE0004024))

6.159.1.91 #define T0PC (*((volatile unsigned long *) 0xE0004010))

6.159.1.92 #define T0PR (*((volatile unsigned long *) 0xE000400C))

6.159.1.93 #define T0TC (*((volatile unsigned long *) 0xE0004008))

6.159.1.94 #define T0TCR (*((volatile unsigned long *) 0xE0004004))

6.159.1.95 #define T1CCR (*((volatile unsigned long *) 0xE0008028))

6.159.1.96 #define T1CR0 (*((volatile unsigned long *) 0xE000802C))

6.159.1.97 #define T1CR1 (*((volatile unsigned long *) 0xE0008030))

6.159.1.98 #define T1CR2 (*((volatile unsigned long *) 0xE0008034))

6.159.1.99 #define T1CR3 (*((volatile unsigned long *) 0xE0008038))

6.159.1.100 #define T1EMR (*((volatile unsigned long *) 0xE000803C))

6.159.1.101 #define T1IR (*((volatile unsigned long *) 0xE0008000))
```

```
6.159.1.102 #define T1MCR (*((volatile unsigned long *) 0xE0008014))  
6.159.1.103 #define T1MRO (*((volatile unsigned long *) 0xE0008018))  
6.159.1.104 #define T1MR1 (*((volatile unsigned long *) 0xE000801C))  
6.159.1.105 #define T1MR2 (*((volatile unsigned long *) 0xE0008020))  
6.159.1.106 #define T1MR3 (*((volatile unsigned long *) 0xE0008024))  
6.159.1.107 #define T1PC (*((volatile unsigned long *) 0xE0008010))  
6.159.1.108 #define T1PR (*((volatile unsigned long *) 0xE000800C))  
6.159.1.109 #define T1TC (*((volatile unsigned long *) 0xE0008008))  
6.159.1.110 #define T1TCR (*((volatile unsigned long *) 0xE0008004))  
6.159.1.111 #define U0DLL (*((volatile unsigned char *) 0xE000C000))  
6.159.1.112 #define U0DLM (*((volatile unsigned char *) 0xE000C004))  
6.159.1.113 #define U0FCR (*((volatile unsigned char *) 0xE000C008))  
6.159.1.114 #define U0IER (*((volatile unsigned char *) 0xE000C004))  
6.159.1.115 #define U0IIR (*((volatile unsigned char *) 0xE000C008))  
6.159.1.116 #define U0LCR (*((volatile unsigned char *) 0xE000C00C))  
6.159.1.117 #define U0LSR (*((volatile unsigned char *) 0xE000C014))  
6.159.1.118 #define U0MCR (*((volatile unsigned char *) 0xE000C010))  
6.159.1.119 #define U0MSR (*((volatile unsigned char *) 0xE000C018))  
6.159.1.120 #define U0RBR (*((volatile unsigned char *) 0xE000C000))  
6.159.1.121 #define U0SCR (*((volatile unsigned char *) 0xE000C01C))  
6.159.1.122 #define U0THR (*((volatile unsigned char *) 0xE000C000))  
6.159.1.123 #define U1DLL (*((volatile unsigned char *) 0xE0010000))  
6.159.1.124 #define U1DLM (*((volatile unsigned char *) 0xE0010004))  
6.159.1.125 #define U1FCR (*((volatile unsigned char *) 0xE0010008))
```

```
6.159.1.126 #define U1IER (*((volatile unsigned char *) 0xE0010004))  
6.159.1.127 #define U1IIR (*((volatile unsigned char *) 0xE0010008))  
6.159.1.128 #define U1LCR (*((volatile unsigned char *) 0xE001000C))  
6.159.1.129 #define U1LSR (*((volatile unsigned char *) 0xE0010014))  
6.159.1.130 #define U1MCR (*((volatile unsigned char *) 0xE0010010))  
6.159.1.131 #define U1MSR (*((volatile unsigned char *) 0xE0010018))  
6.159.1.132 #define U1RBR (*((volatile unsigned char *) 0xE0010000))  
6.159.1.133 #define U1SCR (*((volatile unsigned char *) 0xE001001C))  
6.159.1.134 #define U1THR (*((volatile unsigned char *) 0xE0010000))  
6.159.1.135 #define VICDefVectAddr (*((volatile unsigned long *) 0xFFFFF034))  
6.159.1.136 #define VICFIQStatus (*((volatile unsigned long *) 0xFFFFF004))  
6.159.1.137 #define VICIntEnable (*((volatile unsigned long *) 0xFFFFF010))  
6.159.1.138 #define VICIntEnClr (*((volatile unsigned long *) 0xFFFFF014))  
6.159.1.139 #define VICIntSelect (*((volatile unsigned long *) 0xFFFFF00C))  
6.159.1.140 #define VICIRQStatus (*((volatile unsigned long *) 0xFFFFF000))  
6.159.1.141 #define VICProtection (*((volatile unsigned long *) 0xFFFFF020))  
6.159.1.142 #define VICRawIntr (*((volatile unsigned long *) 0xFFFFF008))  
6.159.1.143 #define VICSoftInt (*((volatile unsigned long *) 0xFFFFF018))  
6.159.1.144 #define VICSoftIntClr (*((volatile unsigned long *) 0xFFFFF01C))  
6.159.1.145 #define VICVectAddr (*((volatile unsigned long *) 0xFFFFF030))  
6.159.1.146 #define VICVectAddr0 (*((volatile unsigned long *) 0xFFFFF100))  
6.159.1.147 #define VICVectAddr1 (*((volatile unsigned long *) 0xFFFFF104))  
6.159.1.148 #define VICVectAddr10 (*((volatile unsigned long *) 0xFFFFF128))  
6.159.1.149 #define VICVectAddr11 (*((volatile unsigned long *) 0xFFFFF12C))
```

```
6.159.1.150 #define VICVectAddr12 (*((volatile unsigned long *) 0xFFFFF130))  
6.159.1.151 #define VICVectAddr13 (*((volatile unsigned long *) 0xFFFFF134))  
6.159.1.152 #define VICVectAddr14 (*((volatile unsigned long *) 0xFFFFF138))  
6.159.1.153 #define VICVectAddr15 (*((volatile unsigned long *) 0xFFFFF13C))  
6.159.1.154 #define VICVectAddr2 (*((volatile unsigned long *) 0xFFFFF108))  
6.159.1.155 #define VICVectAddr3 (*((volatile unsigned long *) 0xFFFFF10C))  
6.159.1.156 #define VICVectAddr4 (*((volatile unsigned long *) 0xFFFFF110))  
6.159.1.157 #define VICVectAddr5 (*((volatile unsigned long *) 0xFFFFF114))  
6.159.1.158 #define VICVectAddr6 (*((volatile unsigned long *) 0xFFFFF118))  
6.159.1.159 #define VICVectAddr7 (*((volatile unsigned long *) 0xFFFFF11C))  
6.159.1.160 #define VICVectAddr8 (*((volatile unsigned long *) 0xFFFFF120))  
6.159.1.161 #define VICVectAddr9 (*((volatile unsigned long *) 0xFFFFF124))  
6.159.1.162 #define VICVectCntl0 (*((volatile unsigned long *) 0xFFFFF200))  
6.159.1.163 #define VICVectCntl1 (*((volatile unsigned long *) 0xFFFFF204))  
6.159.1.164 #define VICVectCntl10 (*((volatile unsigned long *) 0xFFFFF228))  
6.159.1.165 #define VICVectCntl11 (*((volatile unsigned long *) 0xFFFFF22C))  
6.159.1.166 #define VICVectCntl12 (*((volatile unsigned long *) 0xFFFFF230))  
6.159.1.167 #define VICVectCntl13 (*((volatile unsigned long *) 0xFFFFF234))  
6.159.1.168 #define VICVectCntl14 (*((volatile unsigned long *) 0xFFFFF238))  
6.159.1.169 #define VICVectCntl15 (*((volatile unsigned long *) 0xFFFFF23C))  
6.159.1.170 #define VICVectCntl2 (*((volatile unsigned long *) 0xFFFFF208))  
6.159.1.171 #define VICVectCntl3 (*((volatile unsigned long *) 0xFFFFF20C))  
6.159.1.172 #define VICVectCntl4 (*((volatile unsigned long *) 0xFFFFF210))  
6.159.1.173 #define VICVectCntl5 (*((volatile unsigned long *) 0xFFFFF214))
```

```
6.159.1.174 #define VICVectCntl6 (*((volatile unsigned long *) 0xFFFFF218))  
6.159.1.175 #define VICVectCntl7 (*((volatile unsigned long *) 0xFFFFF21C))  
6.159.1.176 #define VICVectCntl8 (*((volatile unsigned long *) 0xFFFFF220))  
6.159.1.177 #define VICVectCntl9 (*((volatile unsigned long *) 0xFFFFF224))  
6.159.1.178 #define VPBDIV (*((volatile unsigned char *) 0xE01FC100))  
6.159.1.179 #define WDFEED (*((volatile unsigned char *) 0xE0000008))  
6.159.1.180 #define WDMOD (*((volatile unsigned char *) 0xE0000000))  
6.159.1.181 #define WDTC (*((volatile unsigned long *) 0xE0000004))  
6.159.1.182 #define WDTV (*((volatile unsigned long *) 0xE000000C))  
6.159.1.183 #define YEAR (*((volatile unsigned short*) 0xE002403C))
```

6.160 include/build/lpc213x.h File Reference

Defines

- #define **VICIRQStatus** (*((volatile unsigned long *) 0xFFFFF000))
- #define **VICFIQStatus** (*((volatile unsigned long *) 0xFFFFF004))
- #define **VICRawIntr** (*((volatile unsigned long *) 0xFFFFF008))
- #define **VICIntSelect** (*((volatile unsigned long *) 0xFFFFF00C))
- #define **VICIntEnable** (*((volatile unsigned long *) 0xFFFFF010))
- #define **VICIntEnClr** (*((volatile unsigned long *) 0xFFFFF014))
- #define **VICSoftInt** (*((volatile unsigned long *) 0xFFFFF018))
- #define **VICSoftIntClr** (*((volatile unsigned long *) 0xFFFFF01C))
- #define **VICProtection** (*((volatile unsigned long *) 0xFFFFF020))
- #define **VICVectAddr** (*((volatile unsigned long *) 0xFFFFF030))
- #define **VICDefVectAddr** (*((volatile unsigned long *) 0xFFFFF034))
- #define **VICVectAddr0** (*((volatile unsigned long *) 0xFFFFF100))
- #define **VICVectAddr1** (*((volatile unsigned long *) 0xFFFFF104))
- #define **VICVectAddr2** (*((volatile unsigned long *) 0xFFFFF108))
- #define **VICVectAddr3** (*((volatile unsigned long *) 0xFFFFF10C))
- #define **VICVectAddr4** (*((volatile unsigned long *) 0xFFFFF110))
- #define **VICVectAddr5** (*((volatile unsigned long *) 0xFFFFF114))
- #define **VICVectAddr6** (*((volatile unsigned long *) 0xFFFFF118))
- #define **VICVectAddr7** (*((volatile unsigned long *) 0xFFFFF11C))
- #define **VICVectAddr8** (*((volatile unsigned long *) 0xFFFFF120))
- #define **VICVectAddr9** (*((volatile unsigned long *) 0xFFFFF124))
- #define **VICVectAddr10** (*((volatile unsigned long *) 0xFFFFF128))

- #define **VICVectAddr11** (*((volatile unsigned long *) 0xFFFFF12C))
- #define **VICVectAddr12** (*((volatile unsigned long *) 0xFFFFF130))
- #define **VICVectAddr13** (*((volatile unsigned long *) 0xFFFFF134))
- #define **VICVectAddr14** (*((volatile unsigned long *) 0xFFFFF138))
- #define **VICVectAddr15** (*((volatile unsigned long *) 0xFFFFF13C))
- #define **VICVectCntl0** (*((volatile unsigned long *) 0xFFFFF200))
- #define **VICVectCntl1** (*((volatile unsigned long *) 0xFFFFF204))
- #define **VICVectCntl2** (*((volatile unsigned long *) 0xFFFFF208))
- #define **VICVectCntl3** (*((volatile unsigned long *) 0xFFFFF20C))
- #define **VICVectCntl4** (*((volatile unsigned long *) 0xFFFFF210))
- #define **VICVectCntl5** (*((volatile unsigned long *) 0xFFFFF214))
- #define **VICVectCntl6** (*((volatile unsigned long *) 0xFFFFF218))
- #define **VICVectCntl7** (*((volatile unsigned long *) 0xFFFFF21C))
- #define **VICVectCntl8** (*((volatile unsigned long *) 0xFFFFF220))
- #define **VICVectCntl9** (*((volatile unsigned long *) 0xFFFFF224))
- #define **VICVectCntl10** (*((volatile unsigned long *) 0xFFFFF228))
- #define **VICVectCntl11** (*((volatile unsigned long *) 0xFFFFF22C))
- #define **VICVectCntl12** (*((volatile unsigned long *) 0xFFFFF230))
- #define **VICVectCntl13** (*((volatile unsigned long *) 0xFFFFF234))
- #define **VICVectCntl14** (*((volatile unsigned long *) 0xFFFFF238))
- #define **VICVectCntl15** (*((volatile unsigned long *) 0xFFFFF23C))
- #define **PINSEL0** (*((volatile unsigned long *) 0xE002C000))
- #define **PINSEL1** (*((volatile unsigned long *) 0xE002C004))
- #define **PINSEL2** (*((volatile unsigned long *) 0xE002C014))
- #define **IOPIN0** (*((volatile unsigned long *) 0xE0028000))
- #define **IOSET0** (*((volatile unsigned long *) 0xE0028004))
- #define **IODIR0** (*((volatile unsigned long *) 0xE0028008))
- #define **IOCLR0** (*((volatile unsigned long *) 0xE002800C))
- #define **IOPIN1** (*((volatile unsigned long *) 0xE0028010))
- #define **IOSET1** (*((volatile unsigned long *) 0xE0028014))
- #define **IODIR1** (*((volatile unsigned long *) 0xE0028018))
- #define **IOCLR1** (*((volatile unsigned long *) 0xE002801C))
- #define **FIO0PIN** (*((volatile unsigned long *) 0x3FFC014))
- #define **FIO0SET** (*((volatile unsigned long *) 0x3FFC018))
- #define **FIO0DIR** (*((volatile unsigned long *) 0x3FFC000))
- #define **FIO0CLR** (*((volatile unsigned long *) 0x3FFC01C))
- #define **FIO0MASK** (*((volatile unsigned long *) 0x3FFC010))
- #define **FIO1PIN** (*((volatile unsigned long *) 0x3FFC034))
- #define **FIO1SET** (*((volatile unsigned long *) 0x3FFC038))
- #define **FIO1DIR** (*((volatile unsigned long *) 0x3FFC020))
- #define **FIO1CLR** (*((volatile unsigned long *) 0x3FFC03C))
- #define **FIO1MASK** (*((volatile unsigned long *) 0x3FFC030))
- #define **MAMCR** (*((volatile unsigned char *) 0xE01FC000))
- #define **MAMTIM** (*((volatile unsigned char *) 0xE01FC004))
- #define **MEMMAP** (*((volatile unsigned char *) 0xE01FC040))
- #define **PLLCON** (*((volatile unsigned char *) 0xE01FC080))

- #define **PLLCFG** (*((volatile unsigned char *) 0xE01FC084))
- #define **PLLSTAT** (*((volatile unsigned short*) 0xE01FC088))
- #define **PLLFEED** (*((volatile unsigned char *) 0xE01FC08C))
- #define **VPBDIV** (*((volatile unsigned char *) 0xE01FC100))
- #define **PCON** (*((volatile unsigned char *) 0xE01FC0C0))
- #define **PCONP** (*((volatile unsigned long *) 0xE01FC0C4))
- #define **EXTINT** (*((volatile unsigned char *) 0xE01FC140))
- #define **EXINT** (*((volatile unsigned char *) 0xE01FC140))
- #define **INTWAKE** (*((volatile unsigned char *) 0xE01FC144))
- #define **EXTMODE** (*((volatile unsigned char *) 0xE01FC148))
- #define **EXTPOLAR** (*((volatile unsigned char *) 0xE01FC14C))
- #define **RSID** (*((volatile unsigned char *) 0xE01FC180))
- #define **CSPR** (*((volatile unsigned char *) 0xE01FC184))
- #define **TOIR** (*((volatile unsigned long *) 0xE0004000))
- #define **TOTCR** (*((volatile unsigned long *) 0xE0004004))
- #define **TOTC** (*((volatile unsigned long *) 0xE0004008))
- #define **TOPR** (*((volatile unsigned long *) 0xE000400C))
- #define **TOPC** (*((volatile unsigned long *) 0xE0004010))
- #define **TOMCR** (*((volatile unsigned long *) 0xE0004014))
- #define **TOMR0** (*((volatile unsigned long *) 0xE0004018))
- #define **TOMR1** (*((volatile unsigned long *) 0xE000401C))
- #define **TOMR2** (*((volatile unsigned long *) 0xE0004020))
- #define **TOMR3** (*((volatile unsigned long *) 0xE0004024))
- #define **TOCCR** (*((volatile unsigned long *) 0xE0004028))
- #define **TOCR0** (*((volatile unsigned long *) 0xE000402C))
- #define **TOCR1** (*((volatile unsigned long *) 0xE0004030))
- #define **TOCR2** (*((volatile unsigned long *) 0xE0004034))
- #define **TOCR3** (*((volatile unsigned long *) 0xE0004038))
- #define **TOEMR** (*((volatile unsigned long *) 0xE000403C))
- #define **TOCTCR** (*((volatile unsigned long *) 0xE0004070))
- #define **T1IR** (*((volatile unsigned long *) 0xE0008000))
- #define **T1TCR** (*((volatile unsigned long *) 0xE0008004))
- #define **T1TC** (*((volatile unsigned long *) 0xE0008008))
- #define **T1PR** (*((volatile unsigned long *) 0xE000800C))
- #define **T1PC** (*((volatile unsigned long *) 0xE0008010))
- #define **T1MCR** (*((volatile unsigned long *) 0xE0008014))
- #define **T1MR0** (*((volatile unsigned long *) 0xE0008018))
- #define **T1MR1** (*((volatile unsigned long *) 0xE000801C))
- #define **T1MR2** (*((volatile unsigned long *) 0xE0008020))
- #define **T1MR3** (*((volatile unsigned long *) 0xE0008024))
- #define **T1CCR** (*((volatile unsigned long *) 0xE0008028))
- #define **T1CR0** (*((volatile unsigned long *) 0xE000802C))
- #define **T1CR1** (*((volatile unsigned long *) 0xE0008030))
- #define **T1CR2** (*((volatile unsigned long *) 0xE0008034))
- #define **T1CR3** (*((volatile unsigned long *) 0xE0008038))
- #define **T1EMR** (*((volatile unsigned long *) 0xE000803C))

- #define **T1CTCR** (*((volatile unsigned long *) 0xE0008070))
- #define **PWMIR** (*((volatile unsigned long *) 0xE0014000))
- #define **PWMTCR** (*((volatile unsigned long *) 0xE0014004))
- #define **PWMTC** (*((volatile unsigned long *) 0xE0014008))
- #define **PWMPR** (*((volatile unsigned long *) 0xE001400C))
- #define **PWMPC** (*((volatile unsigned long *) 0xE0014010))
- #define **PWMMCR** (*((volatile unsigned long *) 0xE0014014))
- #define **PWMMR0** (*((volatile unsigned long *) 0xE0014018))
- #define **PWMMR1** (*((volatile unsigned long *) 0xE001401C))
- #define **PWMMR2** (*((volatile unsigned long *) 0xE0014020))
- #define **PWMMR3** (*((volatile unsigned long *) 0xE0014024))
- #define **PWMMR4** (*((volatile unsigned long *) 0xE0014040))
- #define **PWMMR5** (*((volatile unsigned long *) 0xE0014044))
- #define **PWMMR6** (*((volatile unsigned long *) 0xE0014048))
- #define **PWMEMR** (*((volatile unsigned long *) 0xE001403C))
- #define **PWMPCR** (*((volatile unsigned long *) 0xE001404C))
- #define **PWMLER** (*((volatile unsigned long *) 0xE0014050))
- #define **U0RBR** (*((volatile unsigned char *) 0xE000C000))
- #define **U0THR** (*((volatile unsigned char *) 0xE000C000))
- #define **U0IER** (*((volatile unsigned char *) 0xE000C004))
- #define **U0IIR** (*((volatile unsigned char *) 0xE000C008))
- #define **U0FCR** (*((volatile unsigned char *) 0xE000C008))
- #define **U0LCR** (*((volatile unsigned char *) 0xE000C00C))
- #define **U0LSR** (*((volatile unsigned char *) 0xE000C014))
- #define **U0SCR** (*((volatile unsigned char *) 0xE000C01C))
- #define **U0DLL** (*((volatile unsigned char *) 0xE000C000))
- #define **U0DLM** (*((volatile unsigned char *) 0xE000C004))
- #define **U0TER** (*((volatile unsigned char *) 0xE000C030))
- #define **U1RBR** (*((volatile unsigned char *) 0xE0010000))
- #define **U1THR** (*((volatile unsigned char *) 0xE0010000))
- #define **U1IER** (*((volatile unsigned char *) 0xE0010004))
- #define **U1IIR** (*((volatile unsigned char *) 0xE0010008))
- #define **U1FCR** (*((volatile unsigned char *) 0xE0010008))
- #define **U1LCR** (*((volatile unsigned char *) 0xE001000C))
- #define **U1MCR** (*((volatile unsigned char *) 0xE0010010))
- #define **U1LSR** (*((volatile unsigned char *) 0xE0010014))
- #define **U1MSR** (*((volatile unsigned char *) 0xE0010018))
- #define **U1SCR** (*((volatile unsigned char *) 0xE001001C))
- #define **U1DLL** (*((volatile unsigned char *) 0xE0010000))
- #define **U1DLM** (*((volatile unsigned char *) 0xE0010004))
- #define **U1TER** (*((volatile unsigned char *) 0xE0010030))
- #define **I20CONSET** (*((volatile unsigned char *) 0xE001C000))
- #define **I20STAT** (*((volatile unsigned char *) 0xE001C004))
- #define **I20DAT** (*((volatile unsigned char *) 0xE001C008))
- #define **I20ADR** (*((volatile unsigned char *) 0xE001C00C))
- #define **I20SCLH** (*((volatile unsigned short*) 0xE001C010))

- #define **I20SCLL** (*((volatile unsigned short*) 0xE001C014))
- #define **I20CONCLR** (*((volatile unsigned char *) 0xE001C018))
- #define **I21CONSET** (*((volatile unsigned char *) 0xE005C000))
- #define **I21STAT** (*((volatile unsigned char *) 0xE005C004))
- #define **I21DAT** (*((volatile unsigned char *) 0xE005C008))
- #define **I21ADR** (*((volatile unsigned char *) 0xE005C00C))
- #define **I21SCLH** (*((volatile unsigned short*) 0xE005C010))
- #define **I21SCLL** (*((volatile unsigned short*) 0xE005C014))
- #define **I21CONCLR** (*((volatile unsigned char *) 0xE005C018))
- #define **S0SPCR** (*((volatile unsigned char *) 0xE0020000))
- #define **S0SPSR** (*((volatile unsigned char *) 0xE0020004))
- #define **S0SPDR** (*((volatile unsigned char *) 0xE0020008))
- #define **S0SPCCR** (*((volatile unsigned char *) 0xE002000C))
- #define **S0SPTCR** (*((volatile unsigned char *) 0xE0020010))
- #define **S0SPTSR** (*((volatile unsigned char *) 0xE0020014))
- #define **S0SPTOR** (*((volatile unsigned char *) 0xE0020018))
- #define **S0SPINT** (*((volatile unsigned char *) 0xE002001C))
- #define **SSPCR0** (*((volatile unsigned short*) 0xE0068000))
- #define **SSPCR1** (*((volatile unsigned char *) 0xE0068004))
- #define **SSPDR** (*((volatile unsigned short*) 0xE0068008))
- #define **SSPSR** (*((volatile unsigned char *) 0xE006800C))
- #define **SSPCPSR** (*((volatile unsigned char *) 0xE0068010))
- #define **SSPIMSC** (*((volatile unsigned char *) 0xE0068014))
- #define **SSPRIS** (*((volatile unsigned char *) 0xE0068018))
- #define **SSPMIS** (*((volatile unsigned char *) 0xE006801C))
- #define **SSPICR** (*((volatile unsigned char *) 0xE0068020))
- #define **SSPDAMCR** (*((volatile unsigned char *) 0xE0068024))
- #define **ILR** (*((volatile unsigned char *) 0xE0024000))
- #define **CTC** (*((volatile unsigned short*) 0xE0024004))
- #define **CCR** (*((volatile unsigned char *) 0xE0024008))
- #define **CIIR** (*((volatile unsigned char *) 0xE002400C))
- #define **AMR** (*((volatile unsigned char *) 0xE0024010))
- #define **CTIME0** (*((volatile unsigned long *) 0xE0024014))
- #define **CTIME1** (*((volatile unsigned long *) 0xE0024018))
- #define **CTIME2** (*((volatile unsigned long *) 0xE002401C))
- #define **SEC** (*((volatile unsigned char *) 0xE0024020))
- #define **MIN** (*((volatile unsigned char *) 0xE0024024))
- #define **HOUR** (*((volatile unsigned char *) 0xE0024028))
- #define **DOM** (*((volatile unsigned char *) 0xE002402C))
- #define **DOW** (*((volatile unsigned char *) 0xE0024030))
- #define **DOY** (*((volatile unsigned short*) 0xE0024034))
- #define **MONTH** (*((volatile unsigned char *) 0xE0024038))
- #define **YEAR** (*((volatile unsigned short*) 0xE002403C))
- #define **ALSEC** (*((volatile unsigned char *) 0xE0024060))
- #define **ALMIN** (*((volatile unsigned char *) 0xE0024064))
- #define **ALHOUR** (*((volatile unsigned char *) 0xE0024068))

- #define **ALDOM** (*((volatile unsigned char *) 0xE002406C))
- #define **ALDOW** (*((volatile unsigned char *) 0xE0024070))
- #define **ALDOY** (*((volatile unsigned short*) 0xE0024074))
- #define **ALMON** (*((volatile unsigned char *) 0xE0024078))
- #define **ALYEAR** (*((volatile unsigned short*) 0xE002407C))
- #define **PREINT** (*((volatile unsigned short*) 0xE0024080))
- #define **PREFRAC** (*((volatile unsigned short*) 0xE0024084))
- #define **AD0CR** (*((volatile unsigned long *) 0xE0034000))
- #define **AD0DR** (*((volatile unsigned long *) 0xE0034004))
- #define **AD1CR** (*((volatile unsigned long *) 0xE0060000))
- #define **AD1DR** (*((volatile unsigned long *) 0xE0060004))
- #define **DACR** (*((volatile unsigned long *) 0xE006C000))
- #define **WDMOD** (*((volatile unsigned char *) 0xE0000000))
- #define **WDTC** (*((volatile unsigned long *) 0xE0000004))
- #define **WDFEED** (*((volatile unsigned char *) 0xE0000008))
- #define **WDTV** (*((volatile unsigned long *) 0xE000000C))

6.160.1 Define Documentation

6.160.1.1 #define **AD0CR** (*((volatile unsigned long *) 0xE0034000))

6.160.1.2 #define **AD0DR** (*((volatile unsigned long *) 0xE0034004))

6.160.1.3 #define **AD1CR** (*((volatile unsigned long *) 0xE0060000))

6.160.1.4 #define **AD1DR** (*((volatile unsigned long *) 0xE0060004))

6.160.1.5 #define **ALDOM** (*((volatile unsigned char *) 0xE002406C))

6.160.1.6 #define **ALDOW** (*((volatile unsigned char *) 0xE0024070))

6.160.1.7 #define **ALDOY** (*((volatile unsigned short*) 0xE0024074))

6.160.1.8 #define **ALHOUR** (*((volatile unsigned char *) 0xE0024068))

6.160.1.9 #define **ALMIN** (*((volatile unsigned char *) 0xE0024064))

6.160.1.10 #define **ALMON** (*((volatile unsigned char *) 0xE0024078))

6.160.1.11 #define **ALSEC** (*((volatile unsigned char *) 0xE0024060))

6.160.1.12 #define **ALYEAR** (*((volatile unsigned short*) 0xE002407C))

6.160.1.13 #define **AMR** (*((volatile unsigned char *) 0xE0024010))

6.160.1.14 #define **CCR** (*((volatile unsigned char *) 0xE0024008))

```
6.160.1.15 #define CIIR (*((volatile unsigned char *) 0xE002400C))

6.160.1.16 #define CSPR (*((volatile unsigned char *) 0xE01FC184))

6.160.1.17 #define CTC (*((volatile unsigned short*) 0xE0024004))

6.160.1.18 #define CTIME0 (*((volatile unsigned long *) 0xE0024014))

6.160.1.19 #define CTIME1 (*((volatile unsigned long *) 0xE0024018))

6.160.1.20 #define CTIME2 (*((volatile unsigned long *) 0xE002401C))

6.160.1.21 #define DACR (*((volatile unsigned long *) 0xE006C000))

6.160.1.22 #define DOM (*((volatile unsigned char *) 0xE002402C))

6.160.1.23 #define DOW (*((volatile unsigned char *) 0xE0024030))

6.160.1.24 #define DOY (*((volatile unsigned short*) 0xE0024034))

6.160.1.25 #define EXINT (*((volatile unsigned char *) 0xE01FC140))

6.160.1.26 #define EXTINT (*((volatile unsigned char *) 0xE01FC140))

6.160.1.27 #define EXTMODE (*((volatile unsigned char *) 0xE01FC148))

6.160.1.28 #define EXTPOLAR (*((volatile unsigned char *) 0xE01FC14C))

6.160.1.29 #define FIO0CLR (*((volatile unsigned long *) 0x3FFFC01C))

6.160.1.30 #define FIO0DIR (*((volatile unsigned long *) 0x3FFFC000))

6.160.1.31 #define FIO0MASK (*((volatile unsigned long *) 0x3FFFC010))

6.160.1.32 #define FIO0PIN (*((volatile unsigned long *) 0x3FFFC014))

6.160.1.33 #define FIO0SET (*((volatile unsigned long *) 0x3FFFC018))

6.160.1.34 #define FIO1CLR (*((volatile unsigned long *) 0x3FFFC03C))

6.160.1.35 #define FIO1DIR (*((volatile unsigned long *) 0x3FFFC020))

6.160.1.36 #define FIO1MASK (*((volatile unsigned long *) 0x3FFFC030))

6.160.1.37 #define FIO1PIN (*((volatile unsigned long *) 0x3FFFC034))

6.160.1.38 #define FIO1SET (*((volatile unsigned long *) 0x3FFFC038))
```

```
6.160.1.39 #define HOUR (*((volatile unsigned char *) 0xE0024028))

6.160.1.40 #define I20ADR (*((volatile unsigned char *) 0xE001C00C))

6.160.1.41 #define I20CONCLR (*((volatile unsigned char *) 0xE001C018))

6.160.1.42 #define I20CONSET (*((volatile unsigned char *) 0xE001C000))

6.160.1.43 #define I20DAT (*((volatile unsigned char *) 0xE001C008))

6.160.1.44 #define I20SCLH (*((volatile unsigned short*) 0xE001C010))

6.160.1.45 #define I20SCLL (*((volatile unsigned short*) 0xE001C014))

6.160.1.46 #define I20STAT (*((volatile unsigned char *) 0xE001C004))

6.160.1.47 #define I21ADR (*((volatile unsigned char *) 0xE005C00C))

6.160.1.48 #define I21CONCLR (*((volatile unsigned char *) 0xE005C018))

6.160.1.49 #define I21CONSET (*((volatile unsigned char *) 0xE005C000))

6.160.1.50 #define I21DAT (*((volatile unsigned char *) 0xE005C008))

6.160.1.51 #define I21SCLH (*((volatile unsigned short*) 0xE005C010))

6.160.1.52 #define I21SCLL (*((volatile unsigned short*) 0xE005C014))

6.160.1.53 #define I21STAT (*((volatile unsigned char *) 0xE005C004))

6.160.1.54 #define ILR (*((volatile unsigned char *) 0xE0024000))

6.160.1.55 #define INTWAKE (*((volatile unsigned char *) 0xE01FC144))

6.160.1.56 #define IOCLR0 (*((volatile unsigned long *) 0xE002800C))

6.160.1.57 #define IOCLR1 (*((volatile unsigned long *) 0xE002801C))

6.160.1.58 #define IODIR0 (*((volatile unsigned long *) 0xE0028008))

6.160.1.59 #define IODIR1 (*((volatile unsigned long *) 0xE0028018))

6.160.1.60 #define IOPIN0 (*((volatile unsigned long *) 0xE0028000))

6.160.1.61 #define IOPIN1 (*((volatile unsigned long *) 0xE0028010))

6.160.1.62 #define IOSET0 (*((volatile unsigned long *) 0xE0028004))
```

```
6.160.1.63 #define IOSET1 (*((volatile unsigned long *) 0xE0028014))

6.160.1.64 #define MAMCR (*((volatile unsigned char *) 0xE01FC000))

6.160.1.65 #define MAMTIM (*((volatile unsigned char *) 0xE01FC004))

6.160.1.66 #define MEMMAP (*((volatile unsigned char *) 0xE01FC040))

6.160.1.67 #define MIN (*((volatile unsigned char *) 0xE0024024))

6.160.1.68 #define MONTH (*((volatile unsigned char *) 0xE0024038))

6.160.1.69 #define PCON (*((volatile unsigned char *) 0xE01FC0C0))

6.160.1.70 #define PCOMP (*((volatile unsigned long *) 0xE01FC0C4))

6.160.1.71 #define PINSEL0 (*((volatile unsigned long *) 0xE002C000))

6.160.1.72 #define PINSEL1 (*((volatile unsigned long *) 0xE002C004))

6.160.1.73 #define PINSEL2 (*((volatile unsigned long *) 0xE002C014))

6.160.1.74 #define PLLCFG (*((volatile unsigned char *) 0xE01FC084))

6.160.1.75 #define PLLCON (*((volatile unsigned char *) 0xE01FC080))

6.160.1.76 #define PLLFEED (*((volatile unsigned char *) 0xE01FC08C))

6.160.1.77 #define PLLSTAT (*((volatile unsigned short*) 0xE01FC088))

6.160.1.78 #define PREFRAC (*((volatile unsigned short*) 0xE0024084))

6.160.1.79 #define PREINT (*((volatile unsigned short*) 0xE0024080))

6.160.1.80 #define PWMEMR (*((volatile unsigned long *) 0xE001403C))

6.160.1.81 #define PWMIR (*((volatile unsigned long *) 0xE0014000))

6.160.1.82 #define PWMLER (*((volatile unsigned long *) 0xE0014050))

6.160.1.83 #define PWMMCR (*((volatile unsigned long *) 0xE0014014))

6.160.1.84 #define PWMMR0 (*((volatile unsigned long *) 0xE0014018))

6.160.1.85 #define PWMMR1 (*((volatile unsigned long *) 0xE001401C))

6.160.1.86 #define PWMMR2 (*((volatile unsigned long *) 0xE0014020))
```

```
6.160.1.87 #define PWMMR3 (*((volatile unsigned long *) 0xE0014024))

6.160.1.88 #define PWMMR4 (*((volatile unsigned long *) 0xE0014040))

6.160.1.89 #define PWMMR5 (*((volatile unsigned long *) 0xE0014044))

6.160.1.90 #define PWMMR6 (*((volatile unsigned long *) 0xE0014048))

6.160.1.91 #define PWMPC (*((volatile unsigned long *) 0xE0014010))

6.160.1.92 #define PWMPCR (*((volatile unsigned long *) 0xE001404C))

6.160.1.93 #define PWMPR (*((volatile unsigned long *) 0xE001400C))

6.160.1.94 #define PWMTC (*((volatile unsigned long *) 0xE0014008))

6.160.1.95 #define PWMTCR (*((volatile unsigned long *) 0xE0014004))

6.160.1.96 #define RSID (*((volatile unsigned char *) 0xE01FC180))

6.160.1.97 #define S0SPCCR (*((volatile unsigned char *) 0xE002000C))

6.160.1.98 #define S0SPCR (*((volatile unsigned char *) 0xE0020000))

6.160.1.99 #define S0SPDR (*((volatile unsigned char *) 0xE0020008))

6.160.1.100 #define SOSPINT (*((volatile unsigned char *) 0xE002001C))

6.160.1.101 #define SOSPSR (*((volatile unsigned char *) 0xE0020004))

6.160.1.102 #define SOSPTCR (*((volatile unsigned char *) 0xE0020010))

6.160.1.103 #define SOSPTOR (*((volatile unsigned char *) 0xE0020018))

6.160.1.104 #define SOSPTSR (*((volatile unsigned char *) 0xE0020014))

6.160.1.105 #define SEC (*((volatile unsigned char *) 0xE0024020))

6.160.1.106 #define SSPCPSR (*((volatile unsigned char *) 0xE0068010))

6.160.1.107 #define SSPCR0 (*((volatile unsigned short *) 0xE0068000))

6.160.1.108 #define SSPCR1 (*((volatile unsigned char *) 0xE0068004))

6.160.1.109 #define SSPDMACR (*((volatile unsigned char *) 0xE0068024))

6.160.1.110 #define SSPDPR (*((volatile unsigned short *) 0xE0068008))
```

```
6.160.1.111 #define SSPICR (*((volatile unsigned char *) 0xE0068020))

6.160.1.112 #define SSPIMSC (*((volatile unsigned char *) 0xE0068014))

6.160.1.113 #define SSPMIS (*((volatile unsigned char *) 0xE006801C))

6.160.1.114 #define SSPRIS (*((volatile unsigned char *) 0xE0068018))

6.160.1.115 #define SSPSR (*((volatile unsigned char *) 0xE006800C))

6.160.1.116 #define T0CCR (*((volatile unsigned long *) 0xE0004028))

6.160.1.117 #define T0CR0 (*((volatile unsigned long *) 0xE000402C))

6.160.1.118 #define T0CR1 (*((volatile unsigned long *) 0xE0004030))

6.160.1.119 #define T0CR2 (*((volatile unsigned long *) 0xE0004034))

6.160.1.120 #define T0CR3 (*((volatile unsigned long *) 0xE0004038))

6.160.1.121 #define T0CTCR (*((volatile unsigned long *) 0xE0004070))

6.160.1.122 #define T0EMR (*((volatile unsigned long *) 0xE000403C))

6.160.1.123 #define T0IR (*((volatile unsigned long *) 0xE0004000))

6.160.1.124 #define T0MCR (*((volatile unsigned long *) 0xE0004014))

6.160.1.125 #define T0MR0 (*((volatile unsigned long *) 0xE0004018))

6.160.1.126 #define T0MR1 (*((volatile unsigned long *) 0xE000401C))

6.160.1.127 #define T0MR2 (*((volatile unsigned long *) 0xE0004020))

6.160.1.128 #define T0MR3 (*((volatile unsigned long *) 0xE0004024))

6.160.1.129 #define T0PC (*((volatile unsigned long *) 0xE0004010))

6.160.1.130 #define T0PR (*((volatile unsigned long *) 0xE000400C))

6.160.1.131 #define T0TC (*((volatile unsigned long *) 0xE0004008))

6.160.1.132 #define T0TCR (*((volatile unsigned long *) 0xE0004004))

6.160.1.133 #define T1CCR (*((volatile unsigned long *) 0xE0008028))

6.160.1.134 #define T1CR0 (*((volatile unsigned long *) 0xE000802C))
```

```
6.160.1.135 #define T1CR1 (*((volatile unsigned long *) 0xE0008030))

6.160.1.136 #define T1CR2 (*((volatile unsigned long *) 0xE0008034))

6.160.1.137 #define T1CR3 (*((volatile unsigned long *) 0xE0008038))

6.160.1.138 #define T1CTCR (*((volatile unsigned long *) 0xE0008070))

6.160.1.139 #define T1EMR (*((volatile unsigned long *) 0xE000803C))

6.160.1.140 #define T1IR (*((volatile unsigned long *) 0xE0008000))

6.160.1.141 #define T1MCR (*((volatile unsigned long *) 0xE0008014))

6.160.1.142 #define T1MRO (*((volatile unsigned long *) 0xE0008018))

6.160.1.143 #define T1MR1 (*((volatile unsigned long *) 0xE000801C))

6.160.1.144 #define T1MR2 (*((volatile unsigned long *) 0xE0008020))

6.160.1.145 #define T1MR3 (*((volatile unsigned long *) 0xE0008024))

6.160.1.146 #define T1PC (*((volatile unsigned long *) 0xE0008010))

6.160.1.147 #define T1PR (*((volatile unsigned long *) 0xE000800C))

6.160.1.148 #define T1TC (*((volatile unsigned long *) 0xE0008008))

6.160.1.149 #define T1TCR (*((volatile unsigned long *) 0xE0008004))

6.160.1.150 #define U0DLL (*((volatile unsigned char *) 0xE000C000))

6.160.1.151 #define U0DLM (*((volatile unsigned char *) 0xE000C004))

6.160.1.152 #define U0FCR (*((volatile unsigned char *) 0xE000C008))

6.160.1.153 #define U0IER (*((volatile unsigned char *) 0xE000C004))

6.160.1.154 #define U0IIR (*((volatile unsigned char *) 0xE000C008))

6.160.1.155 #define U0LCR (*((volatile unsigned char *) 0xE000C00C))

6.160.1.156 #define U0LSR (*((volatile unsigned char *) 0xE000C014))

6.160.1.157 #define U0RBR (*((volatile unsigned char *) 0xE000C000))

6.160.1.158 #define U0SCR (*((volatile unsigned char *) 0xE000C01C))
```

```
6.160.1.159 #define U0TER (*((volatile unsigned char *) 0xE000C030))

6.160.1.160 #define U0THR (*((volatile unsigned char *) 0xE000C000))

6.160.1.161 #define U1DLL (*((volatile unsigned char *) 0xE0010000))

6.160.1.162 #define U1DLM (*((volatile unsigned char *) 0xE0010004))

6.160.1.163 #define U1FCR (*((volatile unsigned char *) 0xE0010008))

6.160.1.164 #define U1IER (*((volatile unsigned char *) 0xE0010004))

6.160.1.165 #define U1IIR (*((volatile unsigned char *) 0xE0010008))

6.160.1.166 #define U1LCR (*((volatile unsigned char *) 0xE001000C))

6.160.1.167 #define U1LSR (*((volatile unsigned char *) 0xE0010014))

6.160.1.168 #define U1MCR (*((volatile unsigned char *) 0xE0010010))

6.160.1.169 #define U1MSR (*((volatile unsigned char *) 0xE0010018))

6.160.1.170 #define U1RBR (*((volatile unsigned char *) 0xE0010000))

6.160.1.171 #define U1SCR (*((volatile unsigned char *) 0xE001001C))

6.160.1.172 #define U1TER (*((volatile unsigned char *) 0xE0010030))

6.160.1.173 #define U1THR (*((volatile unsigned char *) 0xE0010000))

6.160.1.174 #define VICDefVectAddr (*((volatile unsigned long *) 0xFFFFF034))

6.160.1.175 #define VICFIQStatus (*((volatile unsigned long *) 0xFFFFF004))

6.160.1.176 #define VICIntEnable (*((volatile unsigned long *) 0xFFFFF010))

6.160.1.177 #define VICIntEnClr (*((volatile unsigned long *) 0xFFFFF014))

6.160.1.178 #define VICIntSelect (*((volatile unsigned long *) 0xFFFFF00C))

6.160.1.179 #define VICIRQStatus (*((volatile unsigned long *) 0xFFFFF000))

6.160.1.180 #define VICProtection (*((volatile unsigned long *) 0xFFFFF020))

6.160.1.181 #define VICRawIntr (*((volatile unsigned long *) 0xFFFFF008))

6.160.1.182 #define VICSoftInt (*((volatile unsigned long *) 0xFFFFF018))
```

```
6.160.1.183 #define VICSoftIntClr (*((volatile unsigned long *) 0xFFFFF01C))

6.160.1.184 #define VICVectAddr (*((volatile unsigned long *) 0xFFFFF030))

6.160.1.185 #define VICVectAddr0 (*((volatile unsigned long *) 0xFFFFF100))

6.160.1.186 #define VICVectAddr1 (*((volatile unsigned long *) 0xFFFFF104))

6.160.1.187 #define VICVectAddr10 (*((volatile unsigned long *) 0xFFFFF128))

6.160.1.188 #define VICVectAddr11 (*((volatile unsigned long *) 0xFFFFF12C))

6.160.1.189 #define VICVectAddr12 (*((volatile unsigned long *) 0xFFFFF130))

6.160.1.190 #define VICVectAddr13 (*((volatile unsigned long *) 0xFFFFF134))

6.160.1.191 #define VICVectAddr14 (*((volatile unsigned long *) 0xFFFFF138))

6.160.1.192 #define VICVectAddr15 (*((volatile unsigned long *) 0xFFFFF13C))

6.160.1.193 #define VICVectAddr2 (*((volatile unsigned long *) 0xFFFFF108))

6.160.1.194 #define VICVectAddr3 (*((volatile unsigned long *) 0xFFFFF10C))

6.160.1.195 #define VICVectAddr4 (*((volatile unsigned long *) 0xFFFFF110))

6.160.1.196 #define VICVectAddr5 (*((volatile unsigned long *) 0xFFFFF114))

6.160.1.197 #define VICVectAddr6 (*((volatile unsigned long *) 0xFFFFF118))

6.160.1.198 #define VICVectAddr7 (*((volatile unsigned long *) 0xFFFFF11C))

6.160.1.199 #define VICVectAddr8 (*((volatile unsigned long *) 0xFFFFF120))

6.160.1.200 #define VICVectAddr9 (*((volatile unsigned long *) 0xFFFFF124))

6.160.1.201 #define VICVectCntl0 (*((volatile unsigned long *) 0xFFFFF200))

6.160.1.202 #define VICVectCntl1 (*((volatile unsigned long *) 0xFFFFF204))

6.160.1.203 #define VICVectCntl10 (*((volatile unsigned long *) 0xFFFFF228))

6.160.1.204 #define VICVectCntl11 (*((volatile unsigned long *) 0xFFFFF22C))

6.160.1.205 #define VICVectCntl12 (*((volatile unsigned long *) 0xFFFFF230))

6.160.1.206 #define VICVectCntl13 (*((volatile unsigned long *) 0xFFFFF234))
```

```
6.160.1.207 #define VICVectCntl14 (*((volatile unsigned long *) 0xFFFFF238))  
6.160.1.208 #define VICVectCntl15 (*((volatile unsigned long *) 0xFFFFF23C))  
6.160.1.209 #define VICVectCntl2 (*((volatile unsigned long *) 0xFFFFF208))  
6.160.1.210 #define VICVectCntl3 (*((volatile unsigned long *) 0xFFFFF20C))  
6.160.1.211 #define VICVectCntl4 (*((volatile unsigned long *) 0xFFFFF210))  
6.160.1.212 #define VICVectCntl5 (*((volatile unsigned long *) 0xFFFFF214))  
6.160.1.213 #define VICVectCntl6 (*((volatile unsigned long *) 0xFFFFF218))  
6.160.1.214 #define VICVectCntl7 (*((volatile unsigned long *) 0xFFFFF21C))  
6.160.1.215 #define VICVectCntl8 (*((volatile unsigned long *) 0xFFFFF220))  
6.160.1.216 #define VICVectCntl9 (*((volatile unsigned long *) 0xFFFFF224))  
6.160.1.217 #define VPBDIV (*((volatile unsigned char *) 0xE01FC100))  
6.160.1.218 #define WDFEED (*((volatile unsigned char *) 0xE0000008))  
6.160.1.219 #define WDMOD (*((volatile unsigned char *) 0xE0000000))  
6.160.1.220 #define WDTC (*((volatile unsigned long *) 0xE0000004))  
6.160.1.221 #define WDTV (*((volatile unsigned long *) 0xE000000C))  
6.160.1.222 #define YEAR (*((volatile unsigned short*) 0xE002403C))
```

6.161 include/build/lpc23xx.h File Reference

Defines

- #define __LPC23xx_H
- #define VIC_BASE_ADDR 0xFFFFF000
- #define VICIRQStatus (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x000)))
- #define VICFIQStatus (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x004)))
- #define VICRawIntr (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x008)))
- #define VICIntSelect (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x00C)))
- #define VICIntEnable (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x010)))
- #define VICIntEnClr (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x014)))
- #define VICSoftInt (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x018)))
- #define VICSoftIntClr (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x01C)))
- #define VICProtection (*((volatile unsigned long *) (VIC_BASE_ADDR + 0x020)))

- #define VICSWPrioMask (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x024))
- #define VICVectAddr0 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x100))
- #define VICVectAddr1 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x104))
- #define VICVectAddr2 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x108))
- #define VICVectAddr3 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x10C))
- #define VICVectAddr4 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x110))
- #define VICVectAddr5 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x114))
- #define VICVectAddr6 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x118))
- #define VICVectAddr7 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x11C))
- #define VICVectAddr8 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x120))
- #define VICVectAddr9 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x124))
- #define VICVectAddr10 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x128))
- #define VICVectAddr11 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x12C))
- #define VICVectAddr12 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x130))
- #define VICVectAddr13 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x134))
- #define VICVectAddr14 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x138))
- #define VICVectAddr15 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x13C))
- #define VICVectAddr16 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x140))
- #define VICVectAddr17 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x144))
- #define VICVectAddr18 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x148))
- #define VICVectAddr19 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x14C))
- #define VICVectAddr20 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x150))
- #define VICVectAddr21 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x154))
- #define VICVectAddr22 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x158))
- #define VICVectAddr23 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x15C))
- #define VICVectAddr24 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x160))
- #define VICVectAddr25 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x164))
- #define VICVectAddr26 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x168))
- #define VICVectAddr27 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x16C))
- #define VICVectAddr28 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x170))
- #define VICVectAddr29 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x174))
- #define VICVectAddr30 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x178))
- #define VICVectAddr31 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x17C))
- #define VICVectCntl0 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x200))
- #define VICVectCntl1 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x204))
- #define VICVectCntl2 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x208))
- #define VICVectCntl3 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x20C))
- #define VICVectCntl4 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x210))
- #define VICVectCntl5 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x214))
- #define VICVectCntl6 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x218))
- #define VICVectCntl7 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x21C))
- #define VICVectCntl8 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x220))
- #define VICVectCntl9 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x224))
- #define VICVectCntl10 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x228))
- #define VICVectCntl11 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x22C))

- #define **VICVectCntl12** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x230))
- #define **VICVectCntl13** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x234))
- #define **VICVectCntl14** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x238))
- #define **VICVectCntl15** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x23C))
- #define **VICVectCntl16** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x240))
- #define **VICVectCntl17** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x244))
- #define **VICVectCntl18** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x248))
- #define **VICVectCntl19** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x24C))
- #define **VICVectCntl20** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x250))
- #define **VICVectCntl21** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x254))
- #define **VICVectCntl22** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x258))
- #define **VICVectCntl23** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x25C))
- #define **VICVectCntl24** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x260))
- #define **VICVectCntl25** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x264))
- #define **VICVectCntl26** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x268))
- #define **VICVectCntl27** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x26C))
- #define **VICVectCntl28** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x270))
- #define **VICVectCntl29** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x274))
- #define **VICVectCntl30** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x278))
- #define **VICVectCntl31** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0x27C))
- #define **VICVectAddr** (*(volatile unsigned long *)(**VIC_BASE_ADDR** + 0xF00))
- #define **PINSEL_BASE_ADDR** 0xE002C000
- #define **PINSEL0** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x00))
- #define **PINSEL1** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x04))
- #define **PINSEL2** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x08))
- #define **PINSEL3** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x0C))
- #define **PINSEL4** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x10))
- #define **PINSEL5** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x14))
- #define **PINSEL6** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x18))
- #define **PINSEL7** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x1C))
- #define **PINSEL8** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x20))
- #define **PINSEL9** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x24))
- #define **PINSEL10** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x28))
- #define **PINMODE0** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x40))
- #define **PINMODE1** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x44))
- #define **PINMODE2** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x48))
- #define **PINMODE3** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x4C))
- #define **PINMODE4** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x50))
- #define **PINMODE5** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x54))
- #define **PINMODE6** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x58))
- #define **PINMODE7** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x5C))
- #define **PINMODE8** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x60))
- #define **PINMODE9** (*(volatile unsigned long *)(**PINSEL_BASE_ADDR** + 0x64))
- #define **GPIO_BASE_ADDR** 0xE0028000
- #define **IOPIN0** (*(volatile unsigned long *)(**GPIO_BASE_ADDR** + 0x00))
- #define **IOSET0** (*(volatile unsigned long *)(**GPIO_BASE_ADDR** + 0x04))

- #define IODIR0 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x08))
- #define IOCLR0 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x0C))
- #define IOPIN1 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x10))
- #define IOSET1 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x14))
- #define IODIR1 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x18))
- #define IOCLR1 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x1C))
- #define IO0_INT_EN_R (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x90))
- #define IO0_INT_EN_F (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x94))
- #define IO0_INT_STAT_R (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x84))
- #define IO0_INT_STAT_F (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x88))
- #define IO0_INT_CLR (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x8C))
- #define IO2_INT_EN_R (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0xB0))
- #define IO2_INT_EN_F (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0xB4))
- #define IO2_INT_STAT_R (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0xA4))
- #define IO2_INT_STAT_F (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0xA8))
- #define IO2_INT_CLR (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0xAC))
- #define IO_INT_STAT (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x80))
- #define PARTCFG_BASE_ADDR 0x3FFF8000
- #define PARTCFG (*(volatile unsigned long *) (PARTCFG_BASE_ADDR + 0x00))
- #define FIO_BASE_ADDR 0x3FFFC000
- #define FIO0DIR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x00))
- #define FIO0MASK (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x10))
- #define FIO0PIN (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x14))
- #define FIO0SET (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x18))
- #define FIO0CLR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x1C))
- #define FIO1DIR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x20))
- #define FIO1MASK (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x30))
- #define FIO1PIN (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x34))
- #define FIO1SET (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x38))
- #define FIO1CLR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x3C))
- #define FIO2DIR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x40))
- #define FIO2MASK (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x50))
- #define FIO2PIN (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x54))
- #define FIO2SET (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x58))
- #define FIO2CLR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x5C))
- #define FIO3DIR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x60))
- #define FIO3MASK (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x70))
- #define FIO3PIN (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x74))

- #define **FIO3SET** (*(volatile unsigned long *)(**FIO_BASE_ADDR** + 0x78))
- #define **FIO3CLR** (*(volatile unsigned long *)(**FIO_BASE_ADDR** + 0x7C))
- #define **FIO4DIR** (*(volatile unsigned long *)(**FIO_BASE_ADDR** + 0x80))
- #define **FIO4MASK** (*(volatile unsigned long *)(**FIO_BASE_ADDR** + 0x90))
- #define **FIO4PIN** (*(volatile unsigned long *)(**FIO_BASE_ADDR** + 0x94))
- #define **FIO4SET** (*(volatile unsigned long *)(**FIO_BASE_ADDR** + 0x98))
- #define **FIO4CLR** (*(volatile unsigned long *)(**FIO_BASE_ADDR** + 0x9C))
- #define **FIO0DIR0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x00))
- #define **FIO1DIR0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x20))
- #define **FIO2DIR0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x40))
- #define **FIO3DIR0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x60))
- #define **FIO4DIR0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x80))
- #define **FIO0DIR1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x01))
- #define **FIO1DIR1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x21))
- #define **FIO2DIR1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x41))
- #define **FIO3DIR1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x61))
- #define **FIO4DIR1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x81))
- #define **FIO0DIR2** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x02))
- #define **FIO1DIR2** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x22))
- #define **FIO2DIR2** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x42))
- #define **FIO3DIR2** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x62))
- #define **FIO4DIR2** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x82))
- #define **FIO0DIR3** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x03))
- #define **FIO1DIR3** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x23))
- #define **FIO2DIR3** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x43))
- #define **FIO3DIR3** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x63))
- #define **FIO4DIR3** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x83))
- #define **FIO0DIRL** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x00))
- #define **FIO1DIRL** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x20))
- #define **FIO2DIRL** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x40))
- #define **FIO3DIRL** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x60))
- #define **FIO4DIRL** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x80))
- #define **FIO0DIRU** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x02))
- #define **FIO1DIRU** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x22))
- #define **FIO2DIRU** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x42))
- #define **FIO3DIRU** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x62))
- #define **FIO4DIRU** (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x82))
- #define **FIO0MASK0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x10))
- #define **FIO1MASK0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x30))
- #define **FIO2MASK0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x50))
- #define **FIO3MASK0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x70))
- #define **FIO4MASK0** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x90))
- #define **FIO0MASK1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x11))
- #define **FIO1MASK1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x21))
- #define **FIO2MASK1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x51))
- #define **FIO3MASK1** (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x71))

- #define FIO4MASK1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x91))
- #define FIO0MASK2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x12))
- #define FIO1MASK2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x32))
- #define FIO2MASK2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x52))
- #define FIO3MASK2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x72))
- #define FIO4MASK2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x92))
- #define FIO0MASK3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x13))
- #define FIO1MASK3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x33))
- #define FIO2MASK3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x53))
- #define FIO3MASK3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x73))
- #define FIO4MASK3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x93))
- #define FIO0MASKL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x10))
- #define FIO1MASKL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x30))
- #define FIO2MASKL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x50))
- #define FIO3MASKL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x70))
- #define FIO4MASKL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x90))
- #define FIO0MASKU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x12))
- #define FIO1MASKU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x32))
- #define FIO2MASKU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x52))
- #define FIO3MASKU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x72))
- #define FIO4MASKU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x92))
- #define FIO0PIN0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x14))
- #define FIO1PIN0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x34))
- #define FIO2PIN0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x54))
- #define FIO3PIN0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x74))
- #define FIO4PIN0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x94))
- #define FIO0PIN1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x15))
- #define FIO1PIN1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x25))
- #define FIO2PIN1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x55))
- #define FIO3PIN1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x75))
- #define FIO4PIN1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x95))
- #define FIO0PIN2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x16))
- #define FIO1PIN2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x36))
- #define FIO2PIN2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x56))
- #define FIO3PIN2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x76))
- #define FIO4PIN2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x96))
- #define FIO0PIN3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x17))
- #define FIO1PIN3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x37))
- #define FIO2PIN3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x57))
- #define FIO3PIN3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x77))
- #define FIO4PIN3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x97))
- #define FIO0PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x14))
- #define FIO1PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x34))
- #define FIO2PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x54))
- #define FIO3PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x74))
- #define FIO4PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x94))

- #define FIO0PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x16))
- #define FIO1PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x36))
- #define FIO2PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x56))
- #define FIO3PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x76))
- #define FIO4PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x96))
- #define FIO0SET0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x18))
- #define FIO1SET0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x38))
- #define FIO2SET0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x58))
- #define FIO3SET0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x78))
- #define FIO4SET0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x98))
- #define FIO0SET1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x19))
- #define FIO1SET1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x29))
- #define FIO2SET1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x59))
- #define FIO3SET1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x79))
- #define FIO4SET1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x99))
- #define FIO0SET2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1A))
- #define FIO1SET2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x3A))
- #define FIO2SET2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5A))
- #define FIO3SET2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x7A))
- #define FIO4SET2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x9A))
- #define FIO0SET3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1B))
- #define FIO1SET3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x3B))
- #define FIO2SET3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5B))
- #define FIO3SET3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x7B))
- #define FIO4SET3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x9B))
- #define FIO0SETL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x18))
- #define FIO1SETL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x38))
- #define FIO2SETL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x58))
- #define FIO3SETL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x78))
- #define FIO4SETL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x98))
- #define FIO0SETU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x1A))
- #define FIO1SETU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x3A))
- #define FIO2SETU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x5A))
- #define FIO3SETU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x7A))
- #define FIO4SETU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x9A))
- #define FIO0CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1C))
- #define FIO1CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x3C))
- #define FIO2CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5C))
- #define FIO3CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x7C))
- #define FIO4CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x9C))
- #define FIO0CLR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1D))
- #define FIO1CLR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x2D))
- #define FIO2CLR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5D))
- #define FIO3CLR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x7D))
- #define FIO4CLR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x9D))
- #define FIO0CLR2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1E))

- #define FIO1CLR2 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x3E))
- #define FIO2CLR2 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x5E))
- #define FIO3CLR2 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x7E))
- #define FIO4CLR2 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x9E))
- #define FIO0CLR3 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x1F))
- #define FIO1CLR3 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x3F))
- #define FIO2CLR3 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x5F))
- #define FIO3CLR3 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x7F))
- #define FIO4CLR3 (*(volatile unsigned char *)(**FIO_BASE_ADDR** + 0x9F))
- #define FIO0CLRL (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x1C))
- #define FIO1CLRL (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x3C))
- #define FIO2CLRL (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x5C))
- #define FIO3CLRL (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x7C))
- #define FIO4CLRL (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x9C))
- #define FIO0CLRU (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x1E))
- #define FIO1CLRU (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x3E))
- #define FIO2CLRU (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x5E))
- #define FIO3CLRU (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x7E))
- #define FIO4CLRU (*(volatile unsigned short *)(**FIO_BASE_ADDR** + 0x9E))
- #define **SCB_BASE_ADDR** 0xE01FC000
- #define MAMCR (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x000))
- #define MAMTIM (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x004))
- #define MEMMAP (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x040))
- #define PLLCON (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x080))
- #define PLLCFG (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x084))
- #define PLLSTAT (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x088))
- #define PLLFEED (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x08C))
- #define VPBDIV (((volatile unsigned char *) 0xE01FC100))
- #define PCON (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x0C0))
- #define PCONP (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x0C4))
- #define CCLKCFG (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x104))
- #define USBCLKCFG (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x108))
- #define CLKSRCSEL (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x10C))
- #define PCLKSEL0 (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x1A8))
- #define PCLKSEL1 (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x1AC))
- #define EXTINT (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x140))
- #define INTWAKE (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x144))
- #define EXTMODE (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x148))
- #define EXTPOLAR (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x14C))
- #define RSIR (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x180))
- #define CSPR (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x184))
- #define AHBCFG1 (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x188))
- #define AHBCFG2 (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x18C))
- #define SCS (*(volatile unsigned long *)(**SCB_BASE_ADDR** + 0x1A0))
- #define STATIC_MEM0_BASE 0x80000000
- #define STATIC_MEM1_BASE 0x81000000

- #define STATIC_MEM2_BASE 0x82000000
- #define STATIC_MEM3_BASE 0x83000000
- #define DYNAMIC_MEM0_BASE 0xA0000000
- #define DYNAMIC_MEM1_BASE 0xB0000000
- #define DYNAMIC_MEM2_BASE 0xC0000000
- #define DYNAMIC_MEM3_BASE 0xD0000000
- #define EMC_BASE_ADDR 0xFFE08000
- #define EMC_CTRL (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x000))
- #define EMC_STAT (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x004))
- #define EMC_CONFIG (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x008))
- #define EMC_DYN_CTRL (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x020))
- #define EMC_DYN_RFSH (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x024))
- #define EMC_DYN_RD_CFG (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x028))
- #define EMC_DYN_RP (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x030))
- #define EMC_DYN_RAS (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x034))
- #define EMC_DYN_SREX (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x038))
- #define EMC_DYN_APW (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x03C))
- #define EMC_DYN_DAL (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x040))
- #define EMC_DYN_WR (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x044))
- #define EMC_DYN_RC (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x048))
- #define EMC_DYN_RFC (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x04C))
- #define EMC_DYN_XSR (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x050))
- #define EMC_DYN_RRD (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x054))
- #define EMC_DYN_MR0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x058))
- #define EMC_DYN_CFG0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x100))
- #define EMC_DYN_RASCAS0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x104))
- #define EMC_DYN_CFG1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x140))
- #define EMC_DYN_RASCAS1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x144))

- #define EMC_DYN_CFG2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x160))
- #define EMC_DYN_RASCAS2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x164))
- #define EMC_DYN_CFG3 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x180))
- #define EMC_DYN_RASCAS3 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x184))
- #define EMC_STA_CFG0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x200))
- #define EMC_STA_WAITWEN0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x204))
- #define EMC_STA_WAITOEN0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x208))
- #define EMC_STA_WAITRD0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x20C))
- #define EMC_STA_WAITPAGE0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x210))
- #define EMC_STA_WAITWR0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x214))
- #define EMC_STA_WAITTURN0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x218))
- #define EMC_STA_CFG1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x220))
- #define EMC_STA_WAITWEN1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x224))
- #define EMC_STA_WAITOEN1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x228))
- #define EMC_STA_WAITRD1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x22C))
- #define EMC_STA_WAITPAGE1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x230))
- #define EMC_STA_WAITWR1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x234))
- #define EMC_STA_WAITTURN1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x238))
- #define EMC_STA_CFG2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x240))
- #define EMC_STA_WAITWEN2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x244))
- #define EMC_STA_WAITOEN2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x248))
- #define EMC_STA_WAITRD2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x24C))
- #define EMC_STA_WAITPAGE2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x250))
- #define EMC_STA_WAITWR2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x254))

- #define **EMC_STA_WAITTURN2** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x258))
- #define **EMC_STA_CFG3** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x260))
- #define **EMC_STA_WAITWEN3** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x264))
- #define **EMC_STA_WAITOEN3** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x268))
- #define **EMC_STA_WAITRD3** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x26C))
- #define **EMC_STA_WAITPAGE3** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x270))
- #define **EMC_STA_WAITWR3** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x274))
- #define **EMC_STA_WAITTURN3** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x278))
- #define **EMC_STA_EXT_WAIT** (*(volatile unsigned long *)(**EMC_BASE_ADDR** + 0x880))
- #define **TMR0_BASE_ADDR** 0xE0004000
- #define **T0IR** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x00))
- #define **T0TCR** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x04))
- #define **T0TC** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x08))
- #define **T0PR** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x0C))
- #define **T0PC** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x10))
- #define **T0MCR** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x14))
- #define **T0MR0** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x18))
- #define **T0MR1** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x1C))
- #define **T0MR2** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x20))
- #define **T0MR3** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x24))
- #define **T0CCR** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x28))
- #define **T0CR0** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x2C))
- #define **T0CR1** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x30))
- #define **T0CR2** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x34))
- #define **T0CR3** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x38))
- #define **T0EMR** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x3C))
- #define **T0CTCR** (*(volatile unsigned long *)(**TMR0_BASE_ADDR** + 0x70))
- #define **TMR1_BASE_ADDR** 0xE0008000
- #define **T1IR** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x00))
- #define **T1TCR** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x04))
- #define **T1TC** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x08))
- #define **T1PR** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x0C))
- #define **T1PC** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x10))
- #define **T1MCR** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x14))
- #define **T1MR0** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x18))
- #define **T1MR1** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x1C))
- #define **T1MR2** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x20))
- #define **T1MR3** (*(volatile unsigned long *)(**TMR1_BASE_ADDR** + 0x24))

- #define T1CCR (*(volatile unsigned long *) (TMR1_BASE_ADDR + 0x28))
- #define T1CR0 (*(volatile unsigned long *) (TMR1_BASE_ADDR + 0x2C))
- #define T1CR1 (*(volatile unsigned long *) (TMR1_BASE_ADDR + 0x30))
- #define T1CR2 (*(volatile unsigned long *) (TMR1_BASE_ADDR + 0x34))
- #define T1CR3 (*(volatile unsigned long *) (TMR1_BASE_ADDR + 0x38))
- #define T1EMR (*(volatile unsigned long *) (TMR1_BASE_ADDR + 0x3C))
- #define T1CTCR (*(volatile unsigned long *) (TMR1_BASE_ADDR + 0x70))
- #define TMR2_BASE_ADDR 0xE0070000
- #define T2IR (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x00))
- #define T2TCR (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x04))
- #define T2TC (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x08))
- #define T2PR (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x0C))
- #define T2PC (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x10))
- #define T2MCR (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x14))
- #define T2MR0 (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x18))
- #define T2MR1 (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x1C))
- #define T2MR2 (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x20))
- #define T2MR3 (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x24))
- #define T2CCR (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x28))
- #define T2CR0 (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x2C))
- #define T2CR1 (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x30))
- #define T2CR2 (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x34))
- #define T2CR3 (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x38))
- #define T2EMR (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x3C))
- #define T2CTCR (*(volatile unsigned long *) (TMR2_BASE_ADDR + 0x70))
- #define TMR3_BASE_ADDR 0xE0074000
- #define T3IR (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x00))
- #define T3TCR (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x04))
- #define T3TC (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x08))
- #define T3PR (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x0C))
- #define T3PC (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x10))
- #define T3MCR (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x14))
- #define T3MR0 (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x18))
- #define T3MR1 (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x1C))
- #define T3MR2 (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x20))
- #define T3MR3 (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x24))
- #define T3CCR (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x28))
- #define T3CR0 (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x2C))
- #define T3CR1 (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x30))
- #define T3CR2 (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x34))
- #define T3CR3 (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x38))
- #define T3EMR (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x3C))
- #define T3CTCR (*(volatile unsigned long *) (TMR3_BASE_ADDR + 0x70))
- #define PWM0_BASE_ADDR 0xE0014000
- #define PWM0IR (*(volatile unsigned long *) (PWM0_BASE_ADDR + 0x00))
- #define PWM0TCR (*(volatile unsigned long *) (PWM0_BASE_ADDR + 0x04))

- #define **PWM0TC** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x08))
- #define **PWM0PR** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x0C))
- #define **PWM0PC** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x10))
- #define **PWM0MCR** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x14))
- #define **PWM0MR0** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x18))
- #define **PWM0MR1** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x1C))
- #define **PWM0MR2** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x20))
- #define **PWM0MR3** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x24))
- #define **PWM0CCR** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x28))
- #define **PWM0CR0** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x2C))
- #define **PWM0CR1** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x30))
- #define **PWM0CR2** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x34))
- #define **PWM0CR3** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x38))
- #define **PWM0EMR** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x3C))
- #define **PWM0MR4** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x40))
- #define **PWM0MR5** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x44))
- #define **PWM0MR6** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x48))
- #define **PWM0PCR** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x4C))
- #define **PWM0LER** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x50))
- #define **PWM0CTCR** (*(volatile unsigned long *)(**PWM0_BASE_ADDR** + 0x70))
- #define **PWM1_BASE_ADDR** 0xE0018000
- #define **PWM1IR** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x00))
- #define **PWM1TCR** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x04))
- #define **PWM1TC** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x08))
- #define **PWM1PR** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x0C))
- #define **PWM1PC** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x10))
- #define **PWM1MCR** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x14))
- #define **PWM1MR0** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x18))
- #define **PWM1MR1** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x1C))
- #define **PWM1MR2** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x20))
- #define **PWM1MR3** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x24))
- #define **PWM1CCR** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x28))
- #define **PWM1CR0** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x2C))
- #define **PWM1CR1** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x30))
- #define **PWM1CR2** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x34))
- #define **PWM1CR3** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x38))
- #define **PWM1EMR** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x3C))
- #define **PWM1MR4** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x40))
- #define **PWM1MR5** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x44))
- #define **PWM1MR6** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x48))
- #define **PWM1PCR** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x4C))
- #define **PWM1LER** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x50))
- #define **PWM1CTCR** (*(volatile unsigned long *)(**PWM1_BASE_ADDR** + 0x70))
- #define **UART0_BASE_ADDR** 0xE000C000
- #define **U0RBR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x00))
- #define **U0THR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x00))

- #define **U0DLL** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x00))
- #define **U0DLM** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x04))
- #define **U0IER** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x04))
- #define **U0IIR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x08))
- #define **U0FCR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x08))
- #define **U0LCR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x0C))
- #define **U0LSR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x14))
- #define **U0SCR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x1C))
- #define **U0ACR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x20))
- #define **U0ICR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x24))
- #define **U0FDR** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x28))
- #define **U0TER** (*(volatile unsigned long *)(**UART0_BASE_ADDR** + 0x30))
- #define **UART1_BASE_ADDR** 0xE0010000
- #define **U1RBR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x00))
- #define **U1THR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x00))
- #define **U1DLL** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x00))
- #define **U1DLM** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x04))
- #define **U1IER** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x04))
- #define **U1IIR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x08))
- #define **U1FCR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x08))
- #define **U1LCR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x0C))
- #define **U1MCR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x10))
- #define **U1LSR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x14))
- #define **U1MSR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x18))
- #define **U1SCR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x1C))
- #define **U1ACR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x20))
- #define **U1FDR** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x28))
- #define **U1TER** (*(volatile unsigned long *)(**UART1_BASE_ADDR** + 0x30))
- #define **UART2_BASE_ADDR** 0xE0078000
- #define **U2RBR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x00))
- #define **U2THR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x00))
- #define **U2DLL** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x00))
- #define **U2DLM** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x04))
- #define **U2IER** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x04))
- #define **U2IIR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x08))
- #define **U2FCR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x08))
- #define **U2LCR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x0C))
- #define **U2LSR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x14))
- #define **U2SCR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x1C))
- #define **U2ACR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x20))
- #define **U2ICR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x24))
- #define **U2FDR** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x28))
- #define **U2TER** (*(volatile unsigned long *)(**UART2_BASE_ADDR** + 0x30))
- #define **UART3_BASE_ADDR** 0xE007C000
- #define **U3RBR** (*(volatile unsigned long *)(**UART3_BASE_ADDR** + 0x00))
- #define **U3THR** (*(volatile unsigned long *)(**UART3_BASE_ADDR** + 0x00))

- #define **U3DLL** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x00))
- #define **U3DLM** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x04))
- #define **U3IER** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x04))
- #define **U3IIR** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x08))
- #define **U3FCR** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x08))
- #define **U3LCR** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x0C))
- #define **U3LSR** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x14))
- #define **U3SCR** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x1C))
- #define **U3ACR** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x20))
- #define **U3ICR** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x24))
- #define **U3FDR** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x28))
- #define **U3TER** (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x30))
- #define **I2C0_BASE_ADDR** 0xE001C000
- #define **I2CONSET** (*(volatile unsigned long *)(I2C0_BASE_ADDR + 0x00))
- #define **I2STAT** (*(volatile unsigned long *)(I2C0_BASE_ADDR + 0x04))
- #define **I2DAT** (*(volatile unsigned long *)(I2C0_BASE_ADDR + 0x08))
- #define **I2ADDR** (*(volatile unsigned long *)(I2C0_BASE_ADDR + 0x0C))
- #define **I2SCLH** (*(volatile unsigned long *)(I2C0_BASE_ADDR + 0x10))
- #define **I2SCLL** (*(volatile unsigned long *)(I2C0_BASE_ADDR + 0x14))
- #define **I2CONCLR** (*(volatile unsigned long *)(I2C0_BASE_ADDR + 0x18))
- #define **I2C1_BASE_ADDR** 0xE005C000
- #define **I21CONSET** (*(volatile unsigned long *)(I2C1_BASE_ADDR + 0x00))
- #define **I21STAT** (*(volatile unsigned long *)(I2C1_BASE_ADDR + 0x04))
- #define **I21DAT** (*(volatile unsigned long *)(I2C1_BASE_ADDR + 0x08))
- #define **I21ADR** (*(volatile unsigned long *)(I2C1_BASE_ADDR + 0x0C))
- #define **I21SCLH** (*(volatile unsigned long *)(I2C1_BASE_ADDR + 0x10))
- #define **I21SCLL** (*(volatile unsigned long *)(I2C1_BASE_ADDR + 0x14))
- #define **I21CONCLR** (*(volatile unsigned long *)(I2C1_BASE_ADDR + 0x18))
- #define **I2C2_BASE_ADDR** 0xE0080000
- #define **I22CONSET** (*(volatile unsigned long *)(I2C2_BASE_ADDR + 0x00))
- #define **I22STAT** (*(volatile unsigned long *)(I2C2_BASE_ADDR + 0x04))
- #define **I22DAT** (*(volatile unsigned long *)(I2C2_BASE_ADDR + 0x08))
- #define **I22ADDR** (*(volatile unsigned long *)(I2C2_BASE_ADDR + 0x0C))
- #define **I22SCLH** (*(volatile unsigned long *)(I2C2_BASE_ADDR + 0x10))
- #define **I22SCLL** (*(volatile unsigned long *)(I2C2_BASE_ADDR + 0x14))
- #define **I22CONCLR** (*(volatile unsigned long *)(I2C2_BASE_ADDR + 0x18))
- #define **SPI0_BASE_ADDR** 0xE0020000
- #define **S0SPCR** (*(volatile unsigned long *)(SPI0_BASE_ADDR + 0x00))
- #define **S0SPSR** (*(volatile unsigned long *)(SPI0_BASE_ADDR + 0x04))
- #define **S0SPDR** (*(volatile unsigned long *)(SPI0_BASE_ADDR + 0x08))
- #define **S0SPCCR** (*(volatile unsigned long *)(SPI0_BASE_ADDR + 0x0C))
- #define **S0SPINT** (*(volatile unsigned long *)(SPI0_BASE_ADDR + 0x1C))
- #define **SSP0_BASE_ADDR** 0xE0068000
- #define **SSP0CR0** (*(volatile unsigned long *)(SSP0_BASE_ADDR + 0x00))
- #define **SSP0CR1** (*(volatile unsigned long *)(SSP0_BASE_ADDR + 0x04))
- #define **SSP0DR** (*(volatile unsigned long *)(SSP0_BASE_ADDR + 0x08))

- #define SSP0SR (*(volatile unsigned long *)(**SSP0_BASE_ADDR** + 0x0C))
- #define SSP0CPSR (*(volatile unsigned long *)(**SSP0_BASE_ADDR** + 0x10))
- #define SSP0IMSC (*(volatile unsigned long *)(**SSP0_BASE_ADDR** + 0x14))
- #define SSP0RIS (*(volatile unsigned long *)(**SSP0_BASE_ADDR** + 0x18))
- #define SSP0MIS (*(volatile unsigned long *)(**SSP0_BASE_ADDR** + 0x1C))
- #define SSP0ICR (*(volatile unsigned long *)(**SSP0_BASE_ADDR** + 0x20))
- #define SSP0DMACR (*(volatile unsigned long *)(**SSP0_BASE_ADDR** + 0x24))
- #define SSP1_BASE_ADDR 0xE0030000
- #define SSP1CR0 (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x00))
- #define SSP1CR1 (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x04))
- #define SSP1DR (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x08))
- #define SSP1SR (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x0C))
- #define SSP1CPSR (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x10))
- #define SSP1IMSC (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x14))
- #define SSP1RIS (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x18))
- #define SSP1MIS (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x1C))
- #define SSP1ICR (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x20))
- #define SSP1DMACR (*(volatile unsigned long *)(**SSP1_BASE_ADDR** + 0x24))
- #define RTC_BASE_ADDR 0xE0024000
- #define RTC_ILR (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x00))
- #define RTC_CTC (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x04))
- #define RTC_CCR (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x08))
- #define RTC_CIIR (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x0C))
- #define RTC_AMR (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x10))
- #define RTC_CTIME0 (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x14))
- #define RTC_CTIME1 (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x18))
- #define RTC_CTIME2 (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x1C))
- #define RTC_SEC (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x20))
- #define RTC_MIN (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x24))
- #define RTC_HOUR (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x28))
- #define RTC_DOM (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x2C))
- #define RTC_DOW (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x30))
- #define RTC_DOY (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x34))
- #define RTC_MONTH (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x38))
- #define RTC_YEAR (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x3C))
- #define RTC_CISSL (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x40))
- #define RTC_ALSEC (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x60))
- #define RTC_ALMIN (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x64))
- #define RTC_ALHOUR (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x68))
- #define RTC_ALDOM (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x6C))
- #define RTC_ALDOW (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x70))
- #define RTC_ALDOY (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x74))
- #define RTC_ALMON (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x78))
- #define RTC_ALYEAR (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x7C))
- #define RTC_PREINT (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x80))
- #define RTC_PREFRAC (*(volatile unsigned long *)(**RTC_BASE_ADDR** + 0x84))

- #define AD0_BASE_ADDR 0xE0034000
- #define AD0CR (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x00))
- #define AD0GDR (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x04))
- #define AD0INTEN (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x0C))
- #define AD0DR0 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x10))
- #define AD0DR1 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x14))
- #define AD0DR2 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x18))
- #define AD0DR3 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x1C))
- #define AD0DR4 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x20))
- #define AD0DR5 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x24))
- #define AD0DR6 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x28))
- #define AD0DR7 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x2C))
- #define AD0STAT (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x30))
- #define DAC_BASE_ADDR 0xE006C000
- #define DACR (*(volatile unsigned long *) (DAC_BASE_ADDR + 0x00))
- #define WDG_BASE_ADDR 0xE0000000
- #define WDMOD (*(volatile unsigned long *) (WDG_BASE_ADDR + 0x00))
- #define WDTC (*(volatile unsigned long *) (WDG_BASE_ADDR + 0x04))
- #define WDFEED (*(volatile unsigned long *) (WDG_BASE_ADDR + 0x08))
- #define WDTV (*(volatile unsigned long *) (WDG_BASE_ADDR + 0x0C))
- #define WDCLKSEL (*(volatile unsigned long *) (WDG_BASE_ADDR + 0x10))
- #define CAN_ACCEPT_BASE_ADDR 0xE003C000
- #define CAN_AFMR (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR + 0x00))
- #define CAN_SFF_SA (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR + 0x04))
- #define CAN_SFF_GRP_SA (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR + 0x08))
- #define CAN_EFF_SA (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR + 0x0C))
- #define CAN_EFF_GRP_SA (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR + 0x10))
- #define CAN_EOT (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR + 0x14))
- #define CAN_LUT_ERR_ADR (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR + 0x18))
- #define CAN_LUT_ERR (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR + 0x1C))
- #define CAN_CENTRAL_BASE_ADDR 0xE0040000
- #define CAN_TX_SR (*(volatile unsigned long *) (CAN_CENTRAL_BASE_ADDR + 0x00))
- #define CAN_RX_SR (*(volatile unsigned long *) (CAN_CENTRAL_BASE_ADDR + 0x04))
- #define CAN_MSR (*(volatile unsigned long *) (CAN_CENTRAL_BASE_ADDR + 0x08))
- #define CAN1_BASE_ADDR 0xE0044000
- #define CAN1MOD (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x00))

- #define CAN1CMR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x04))
- #define CAN1GSR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x08))
- #define CAN1ICR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x0C))
- #define CAN1IER (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x10))
- #define CAN1BTR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x14))
- #define CAN1EWL (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x18))
- #define CAN1SR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x1C))
- #define CAN1RFS (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x20))
- #define CAN1RID (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x24))
- #define CAN1RDA (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x28))
- #define CAN1RDB (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x2C))
- #define CAN1TFI1 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x30))
- #define CAN1TID1 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x34))
- #define CAN1TDA1 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x38))
- #define CAN1TDB1 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x3C))
- #define CAN1TFI2 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x40))
- #define CAN1TID2 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x44))
- #define CAN1TDA2 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x48))
- #define CAN1TDB2 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x4C))
- #define CAN1TFI3 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x50))
- #define CAN1TID3 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x54))
- #define CAN1TDA3 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x58))
- #define CAN1TDB3 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x5C))
- #define CAN2_BASE_ADDR 0xE0048000
- #define CAN2MOD (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x00))
- #define CAN2CMR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x04))
- #define CAN2GSR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x08))
- #define CAN2ICR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x0C))
- #define CAN2IER (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x10))
- #define CAN2BTR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x14))
- #define CAN2EWL (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x18))
- #define CAN2SR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x1C))
- #define CAN2RFS (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x20))
- #define CAN2RID (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x24))
- #define CAN2RDA (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x28))
- #define CAN2RDB (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x2C))
- #define CAN2TFI1 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x30))
- #define CAN2TID1 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x34))
- #define CAN2TDA1 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x38))
- #define CAN2TDB1 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x3C))
- #define CAN2TFI2 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x40))
- #define CAN2TID2 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x44))
- #define CAN2TDA2 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x48))
- #define CAN2TDB2 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x4C))
- #define CAN2TFI3 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x50))
- #define CAN2TID3 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x54))

- #define CAN2TDA3 (*(volatile unsigned long *)(**CAN2_BASE_ADDR** + 0x58))
- #define CAN2TDB3 (*(volatile unsigned long *)(**CAN2_BASE_ADDR** + 0x5C))
- #define MCI_BASE_ADDR 0xE008C000
- #define MCI_POWER (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x00))
- #define MCI_CLOCK (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x04))
- #define MCI_ARGUMENT (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x08))
- #define MCI_COMMAND (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x0C))
- #define MCI_RESP_CMD (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x10))
- #define MCI_RESP0 (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x14))
- #define MCI_RESP1 (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x18))
- #define MCI_RESP2 (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x1C))
- #define MCI_RESP3 (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x20))
- #define MCI_DATA_TMR (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x24))
- #define MCI_DATA_LEN (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x28))
- #define MCI_DATA_CTRL (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x2C))
- #define MCI_DATA_CNT (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x30))
- #define MCI_STATUS (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x34))
- #define MCI_CLEAR (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x38))
- #define MCI_MASK0 (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x3C))
- #define MCI_MASK1 (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x40))
- #define MCI_FIFO_CNT (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x48))
- #define MCI_FIFO (*(volatile unsigned long *)(**MCI_BASE_ADDR** + 0x80))
- #define I2S_BASE_ADDR 0xE0088000
- #define I2S_DAO (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x00))
- #define I2S_DAI (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x04))
- #define I2S_TX_FIFO (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x08))
- #define I2S_RX_FIFO (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x0C))
- #define I2S_STATE (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x10))
- #define I2S_DMA1 (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x14))
- #define I2S_DMA2 (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x18))
- #define I2S_IRQ (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x1C))
- #define I2S_TXRATE (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x20))
- #define I2S_RXRATE (*(volatile unsigned long *)(**I2S_BASE_ADDR** + 0x24))
- #define DMA_BASE_ADDR 0xFFE04000
- #define GPDMA_INT_STAT (*(volatile unsigned long *)(**DMA_BASE_ADDR** + 0x000))
- #define GPDMA_INT_TCSTAT (*(volatile unsigned long *)(**DMA_BASE_ADDR** + 0x004))
- #define GPDMA_INT_TCCLR (*(volatile unsigned long *)(**DMA_BASE_ADDR** + 0x008))

- #define GPDMA_INT_ERR_STAT (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x00C))
- #define GPDMA_INT_ERR_CLR (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x010))
- #define GPDMA_RAW_INT_TCSTAT (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x014))
- #define GPDMA_RAW_INT_ERR_STAT (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x018))
- #define GPDMA_ENABLED_CHNS (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x01C))
- #define GPDMA_SOFT_BREQ (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x020))
- #define GPDMA_SOFT_SREQ (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x024))
- #define GPDMA_SOFT_LBREQ (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x028))
- #define GPDMA_SOFT_LSREQ (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x02C))
- #define GPDMA_CONFIG (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x030))
- #define GPDMA_SYNC (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x034))
- #define GPDMA_CH0_SRC (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x100))
- #define GPDMA_CH0_DEST (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x104))
- #define GPDMA_CH0_LLI (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x108))
- #define GPDMA_CH0_CTRL (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x10C))
- #define GPDMA_CH0_CFG (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x110))
- #define GPDMA_CH1_SRC (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x120))
- #define GPDMA_CH1_DEST (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x124))
- #define GPDMA_CH1_LLI (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x128))
- #define GPDMA_CH1_CTRL (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x12C))
- #define GPDMA_CH1_CFG (*(volatile unsigned long *) (DMA_BASE_ADDR + 0x130))
- #define USB_INT_BASE_ADDR 0xE01FC1C0
- #define USB_BASE_ADDR 0xFFE0C200 /* USB Base Address */
- #define USB_INT_STAT (*(volatile unsigned long *) (USB_INT_BASE_ADDR + 0x00))
- #define DEV_INT_STAT (*(volatile unsigned long *) (USB_BASE_ADDR + 0x00))

- #define **DEV_INT_EN** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x04))
- #define **DEV_INT_CLR** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x08))
- #define **DEV_INT_SET** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x0C))
- #define **DEV_INT_PRIO** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x2C))
- #define **EP_INT_STAT** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x30))
- #define **EP_INT_EN** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x34))
- #define **EP_INT_CLR** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x38))
- #define **EP_INT_SET** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x3C))
- #define **EP_INT_PRIO** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x40))
- #define **REALIZE_EP** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x44))
- #define **EP_INDEX** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x48))
- #define **MAXPACKET_SIZE** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x4C))
- #define **CMD_CODE** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x10))
- #define **CMD_DATA** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x14))
- #define **RX_DATA** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x18))
- #define **TX_DATA** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x1C))
- #define **RX_PLENGTH** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x20))
- #define **TX_PLENGTH** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x24))
- #define **USB_CTRL** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x28))
- #define **DMA_REQ_STAT** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x50))
- #define **DMA_REQ_CLR** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x54))
- #define **DMA_REQ_SET** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x58))
- #define **UDCA_HEAD** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x80))
- #define **EP_DMA_STAT** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x84))
- #define **EP_DMA_EN** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x88))
- #define **EP_DMA_DIS** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x8C))
- #define **DMA_INT_STAT** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x90))
- #define **DMA_INT_EN** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0x94))
- #define **EOT_INT_STAT** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0xA0))
- #define **EOT_INT_CLR** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0xA4))
- #define **EOT_INT_SET** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0xA8))
- #define **NDD_REQ_INT_STAT** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0xAC))
- #define **NDD_REQ_INT_CLR** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0xB0))
- #define **NDD_REQ_INT_SET** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0xB4))
- #define **SYS_ERR_INT_STAT** (*(volatile unsigned long *)(**USB_BASE_ADDR** + 0xB8))

- #define SYS_ERR_INT_CLR (*(volatile unsigned long *) (USB_BASE_ADDR + 0xBC))
- #define SYS_ERR_INT_SET (*(volatile unsigned long *) (USB_BASE_ADDR + 0xC0))
- #define USBHC_BASE_ADDR 0xFFE0C000
- #define HC_REVISION (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x00))
- #define HC_CONTROL (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x04))
- #define HC_CMD_STAT (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x08))
- #define HC_INT_STAT (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x0C))
- #define HC_INT_EN (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x10))
- #define HC_INT_DIS (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x14))
- #define HC_HCCA (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x18))
- #define HC_PERIOD_CUR_ED (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x1C))
- #define HC_CTRL_HEAD_ED (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x20))
- #define HC_CTRL_CUR_ED (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x24))
- #define HC_BULK_HEAD_ED (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x28))
- #define HC_BULK_CUR_ED (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x2C))
- #define HC_DONE_HEAD (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x30))
- #define HC_FM_INTERVAL (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x34))
- #define HC_FM_REMAINING (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x38))
- #define HC_FM_NUMBER (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x3C))
- #define HC_PERIOD_START (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x40))
- #define HC_LS THRHL (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x44))
- #define HC_RH_DESCA (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x48))
- #define HC_RH_DESCB (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x4C))
- #define HC_RH_STAT (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x50))
- #define HC_RH_PORT_STAT1 (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x54))

- #define **HC_RH_PORT_STAT2** (*(volatile unsigned long *)(**USBHC_BASE_ADDR** + 0x58))
- #define **USBOTG_BASE_ADDR** 0xFFE0C100
- #define **OTG_INT_STAT** (*(volatile unsigned long *)(**USBOTG_BASE_ADDR** + 0x00))
- #define **OTG_INT_EN** (*(volatile unsigned long *)(**USBOTG_BASE_ADDR** + 0x04))
- #define **OTG_INT_SET** (*(volatile unsigned long *)(**USBOTG_BASE_ADDR** + 0x08))
- #define **OTG_INT_CLR** (*(volatile unsigned long *)(**USBOTG_BASE_ADDR** + 0x0C))
- #define **OTG_STAT_CTRL** (*(volatile unsigned long *)(**USBOTG_BASE_ADDR** + 0x10))
- #define **OTG_TIMER** (*(volatile unsigned long *)(**USBOTG_BASE_ADDR** + 0x14))
- #define **USBOTG_I2C_BASE_ADDR** 0xFFE0C300
- #define **OTG_I2C_RX** (*(volatile unsigned long *)(**USBOTG_I2C_BASE_ADDR** + 0x00))
- #define **OTG_I2C_TX** (*(volatile unsigned long *)(**USBOTG_I2C_BASE_ADDR** + 0x00))
- #define **OTG_I2C_STS** (*(volatile unsigned long *)(**USBOTG_I2C_BASE_ADDR** + 0x04))
- #define **OTG_I2C_CTL** (*(volatile unsigned long *)(**USBOTG_I2C_BASE_ADDR** + 0x08))
- #define **OTG_I2C_CLKHI** (*(volatile unsigned long *)(**USBOTG_I2C_BASE_ADDR** + 0x0C))
- #define **OTG_I2C_CLKLO** (*(volatile unsigned long *)(**USBOTG_I2C_BASE_ADDR** + 0x10))
- #define **USBOTG_CLK_BASE_ADDR** 0xFFE0cff0
- #define **OTG_CLK_CTRL** (*(volatile unsigned long *)(**USBOTG_CLK_BASE_ADDR** + 0x04))
- #define **OTG_CLK_STAT** (*(volatile unsigned long *)(**USBOTG_CLK_BASE_ADDR** + 0x08))
- #define **USBPortSel** (*(volatile unsigned long *)(**USBOTG_BASE_ADDR** + 0x10))
- #define **USBClkCtrl** (*(volatile unsigned long *)(**USBOTG_CLK_BASE_ADDR** + 0x04))
- #define **USBClkSt** (*(volatile unsigned long *)(**USBOTG_CLK_BASE_ADDR** + 0x08))
- #define **MAC_BASE_ADDR** 0xFFE00000 /* AHB Peripheral # 0 */
- #define **MAC_MAC1** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x000)) /* MAC config reg 1 */
- #define **MAC_MAC2** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x004)) /* MAC config reg 2 */
- #define **MAC_IPGT** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x008)) /* b2b InterPacketGap reg */
- #define **MAC_IPGR** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x00C)) /* non b2b InterPacketGap reg */

- #define **MAC_CLRT** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x010))
/* CoLision window/ReTry reg */
- #define **MAC_MAXF** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x014))
/* MAXimum Frame reg */
- #define **MAC_SUPP** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x018))
/* PHY SUPPort reg */
- #define **MAC_TEST** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x01C))
/* TEST reg */
- #define **MAC_MCFG** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x020))
/* MII Mgmt ConFiG reg */
- #define **MAC_MCMD** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x024))
/* MII Mgmt CoMmanD reg */
- #define **MAC_MADR** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x028))
/* MII Mgmt ADdRess reg */
- #define **MAC_MWTD** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x02C))
/* MII Mgmt WriTe Data reg (WO) */
- #define **MAC_MRDD** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x030))
/* MII Mgmt ReaD Data reg (RO) */
- #define **MAC_MIND** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x034))
/* MII Mgmt INDicators reg (RO) */
- #define **MAC_SA0** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x040)) /* Station Address 0 reg */
- #define **MAC_SA1** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x044)) /* Station Address 1 reg */
- #define **MAC_SA2** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x048)) /* Station Address 2 reg */
- #define **MAC_COMMAND** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x100)) /* Command reg */
- #define **MAC_STATUS** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x104)) /* Status reg (RO) */
- #define **MAC_RXDESCRIPTOR** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x108)) /* Rx descriptor base address reg */
- #define **MAC_RXSTATUS** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x10C)) /* Rx status base address reg */
- #define **MAC_RXDESCRIPTORMNUM** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x110)) /* Rx number of descriptors reg */
- #define **MAC_RXPRODUCEINDEX** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x114)) /* Rx produce index reg (RO) */
- #define **MAC_RXCONSUMEINDEX** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x118)) /* Rx consume index reg */
- #define **MAC_TXDESCRIPTOR** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x11C)) /* Tx descriptor base address reg */
- #define **MAC_TXSTATUS** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x120)) /* Tx status base address reg */
- #define **MAC_TXDESCRIPTORMNUM** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x124)) /* Tx number of descriptors reg */
- #define **MAC_TXPRODUCEINDEX** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x128)) /* Tx produce index reg */

- #define **MAC_TXCONSUMEINDEX** (*(volatile unsigned long *)(**MAC_BASE_A-DDR** + 0x12C)) /* Tx consume index reg (RO) */
- #define **MAC_TSV0** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x158)) /* Tx status vector 0 reg (RO) */
- #define **MAC_TSV1** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x15C)) /* Tx status vector 1 reg (RO) */
- #define **MAC_RSV** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x160)) /* Rx status vector reg (RO) */
- #define **MAC_FLOWCONTROLCNT** (*(volatile unsigned long *)(**MAC_BASE_A-ADDR** + 0x170)) /* Flow control counter reg */
- #define **MAC_FLOWCONTROLSTS** (*(volatile unsigned long *)(**MAC_BASE_A-DDR** + 0x174)) /* Flow control status reg */
- #define **MAC_RXFILTERCTRL** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x200)) /* Rx filter ctrl reg */
- #define **MAC_RXFILTERWOLSTS** (*(volatile unsigned long *)(**MAC_BASE_AD-DR** + 0x204)) /* Rx filter WoL status reg (RO) */
- #define **MAC_RXFILTERWOLCLR** (*(volatile unsigned long *)(**MAC_BASE_AD-DR** + 0x208)) /* Rx filter WoL clear reg (WO) */
- #define **MAC_HASHFILTERL** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x210)) /* Hash filter LSBs reg */
- #define **MAC_HASHFILTERH** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x214)) /* Hash filter MSBs reg */
- #define **MAC_INTSTATUS** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x-
FE0)) /* Interrupt status reg (RO) */
- #define **MAC_INENABLE** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x-
FE4)) /* Interrupt enable reg */
- #define **MAC_INTCLEAR** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x-
FE8)) /* Interrupt clear reg (WO) */
- #define **MAC_INTSET** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0xFE-
C)) /* Interrupt set reg (WO) */
- #define **MAC_POWERDOWN** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0xFF4)) /* Power-down reg */
- #define **MAC_MODULEID** (*(volatile unsigned long *)(**MAC_BASE_ADDR** + 0x-
FFC)) /* Module ID reg (RO) */

6.161.1 Define Documentation

6.161.1.1 #define __LPC23xx_H

6.161.1.2 #define AD0_BASE_ADDR 0xE0034000

6.161.1.3 #define AD0CR (*(volatile unsigned long *)(**AD0_BASE_ADDR** + 0x00))

6.161.1.4 #define AD0DR0 (*(volatile unsigned long *)(**AD0_BASE_ADDR** + 0x10))

6.161.1.5 #define AD0DR1 (*(volatile unsigned long *)(**AD0_BASE_ADDR** + 0x14))

```
6.161.1.6 #define AD0DR2 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x18))

6.161.1.7 #define AD0DR3 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x1C))

6.161.1.8 #define AD0DR4 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x20))

6.161.1.9 #define AD0DR5 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x24))

6.161.1.10 #define AD0DR6 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x28))

6.161.1.11 #define AD0DR7 (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x2C))

6.161.1.12 #define AD0GDR (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x04))

6.161.1.13 #define AD0INTEN (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x0C))

6.161.1.14 #define AD0STAT (*(volatile unsigned long *) (AD0_BASE_ADDR + 0x30))

6.161.1.15 #define AHBCFG1 (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x188))

6.161.1.16 #define AHBCFG2 (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x18C))

6.161.1.17 #define CAN1_BASE_ADDR 0xE0044000

6.161.1.18 #define CAN1BTR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x14))

6.161.1.19 #define CAN1CMR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x04))

6.161.1.20 #define CAN1EVL (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x18))

6.161.1.21 #define CAN1GSR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x08))

6.161.1.22 #define CAN1ICR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x0C))

6.161.1.23 #define CAN1IER (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x10))

6.161.1.24 #define CAN1MOD (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x00))

6.161.1.25 #define CAN1RDA (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x28))

6.161.1.26 #define CAN1RDB (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x2C))

6.161.1.27 #define CAN1RFS (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x20))

6.161.1.28 #define CAN1RID (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x24))

6.161.1.29 #define CAN1SR (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x1C))
```

```
6.161.1.30 #define CAN1TDA1 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x38))  
6.161.1.31 #define CAN1TDA2 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x48))  
6.161.1.32 #define CAN1TDA3 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x58))  
6.161.1.33 #define CAN1TDB1 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x3C))  
6.161.1.34 #define CAN1TDB2 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x4C))  
6.161.1.35 #define CAN1TDB3 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x5C))  
6.161.1.36 #define CAN1TFI1 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x30))  
6.161.1.37 #define CAN1TFI2 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x40))  
6.161.1.38 #define CAN1TFI3 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x50))  
6.161.1.39 #define CAN1TID1 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x34))  
6.161.1.40 #define CAN1TID2 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x44))  
6.161.1.41 #define CAN1TID3 (*(volatile unsigned long *) (CAN1_BASE_ADDR + 0x54))  
6.161.1.42 #define CAN2_BASE_ADDR 0xE0048000  
6.161.1.43 #define CAN2BTR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x14))  
6.161.1.44 #define CAN2CMR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x04))  
6.161.1.45 #define CAN2EWL (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x18))  
6.161.1.46 #define CAN2GSR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x08))  
6.161.1.47 #define CAN2ICR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x0C))  
6.161.1.48 #define CAN2IER (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x10))  
6.161.1.49 #define CAN2MOD (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x00))  
6.161.1.50 #define CAN2RDA (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x28))  
6.161.1.51 #define CAN2RDB (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x2C))  
6.161.1.52 #define CAN2RFS (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x20))  
6.161.1.53 #define CAN2RID (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x24))
```

```
6.161.1.54 #define CAN2SR (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x1C))

6.161.1.55 #define CAN2TDA1 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x38))

6.161.1.56 #define CAN2TDA2 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x48))

6.161.1.57 #define CAN2TDA3 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x58))

6.161.1.58 #define CAN2TDB1 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x3C))

6.161.1.59 #define CAN2TDB2 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x4C))

6.161.1.60 #define CAN2TDB3 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x5C))

6.161.1.61 #define CAN2TFI1 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x30))

6.161.1.62 #define CAN2TFI2 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x40))

6.161.1.63 #define CAN2TFI3 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x50))

6.161.1.64 #define CAN2TID1 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x34))

6.161.1.65 #define CAN2TID2 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x44))

6.161.1.66 #define CAN2TID3 (*(volatile unsigned long *) (CAN2_BASE_ADDR + 0x54))

6.161.1.67 #define CAN_ACCEPT_BASE_ADDR 0xE003C000

6.161.1.68 #define CAN_AFMR (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR
+ 0x00))

6.161.1.69 #define CAN_CENTRAL_BASE_ADDR 0xE0040000

6.161.1.70 #define CAN_EFF_GRP_SA (*(volatile unsigned long
*)(CAN_ACCEPT_BASE_ADDR + 0x10))

6.161.1.71 #define CAN_EFF_SA (*(volatile unsigned long
*)(CAN_ACCEPT_BASE_ADDR + 0x0C))

6.161.1.72 #define CAN_EOT (*(volatile unsigned long *) (CAN_ACCEPT_BASE_ADDR +
0x14))

6.161.1.73 #define CAN_LUT_ERR (*(volatile unsigned long
*)(CAN_ACCEPT_BASE_ADDR + 0x1C))

6.161.1.74 #define CAN_LUT_ERR_ADR (*(volatile unsigned long
*)(CAN_ACCEPT_BASE_ADDR + 0x18))
```

```
6.161.1.75 #define CAN_MSR (*(volatile unsigned long *)(CAN_CENTRAL_BASE_ADDR  
+ 0x08))  
  
6.161.1.76 #define CAN_RX_SR (*(volatile unsigned long  
*)(CAN_CENTRAL_BASE_ADDR + 0x04))  
  
6.161.1.77 #define CAN_SFF_GRP_SA (*(volatile unsigned long  
*)(CAN_ACCEPT_BASE_ADDR + 0x08))  
  
6.161.1.78 #define CAN_SFF_SA (*(volatile unsigned long  
*)(CAN_ACCEPT_BASE_ADDR + 0x04))  
  
6.161.1.79 #define CAN_TX_SR (*(volatile unsigned long  
*)(CAN_CENTRAL_BASE_ADDR + 0x00))  
  
6.161.1.80 #define CCLKCFG (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x104))  
  
6.161.1.81 #define CLKSRCSEL (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x10C))  
  
6.161.1.82 #define CMD_CODE (*(volatile unsigned long *)(USB_BASE_ADDR + 0x10))  
  
6.161.1.83 #define CMD_DATA (*(volatile unsigned long *)(USB_BASE_ADDR + 0x14))  
  
6.161.1.84 #define CSPR (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x184))  
  
6.161.1.85 #define DAC_BASE_ADDR 0xE006C000  
  
6.161.1.86 #define DACR (*(volatile unsigned long *)(DAC_BASE_ADDR + 0x00))  
  
6.161.1.87 #define DEV_INT_CLR (*(volatile unsigned long *)(USB_BASE_ADDR + 0x08))  
  
6.161.1.88 #define DEV_INT_EN (*(volatile unsigned long *)(USB_BASE_ADDR + 0x04))  
  
6.161.1.89 #define DEV_INT_PRIO (*(volatile unsigned long *)(USB_BASE_ADDR +  
0x2C))  
  
6.161.1.90 #define DEV_INT_SET (*(volatile unsigned long *)(USB_BASE_ADDR + 0x0C))  
  
6.161.1.91 #define DEV_INT_STAT (*(volatile unsigned long *)(USB_BASE_ADDR +  
0x00))  
  
6.161.1.92 #define DMA_BASE_ADDR 0xFFE04000  
  
6.161.1.93 #define DMA_INT_EN (*(volatile unsigned long *)(USB_BASE_ADDR + 0x94))  
  
6.161.1.94 #define DMA_INT_STAT (*(volatile unsigned long *)(USB_BASE_ADDR +  
0x90))
```

```
6.161.1.95 #define DMA_REQ_CLR (*(volatile unsigned long *) (USB_BASE_ADDR +  
0x54))  
  
6.161.1.96 #define DMA_REQ_SET (*(volatile unsigned long *) (USB_BASE_ADDR +  
0x58))  
  
6.161.1.97 #define DMA_REQ_STAT (*(volatile unsigned long *) (USB_BASE_ADDR +  
0x50))  
  
6.161.1.98 #define DYNAMIC_MEM0_BASE 0xA0000000  
  
6.161.1.99 #define DYNAMIC_MEM1_BASE 0xB0000000  
  
6.161.1.100 #define DYNAMIC_MEM2_BASE 0xC0000000  
  
6.161.1.101 #define DYNAMIC_MEM3_BASE 0xD0000000  
  
6.161.1.102 #define EMC_BASE_ADDR 0xFFE08000  
  
6.161.1.103 #define EMC_CONFIG (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x008))  
  
6.161.1.104 #define EMC_CTRL (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x000))  
  
6.161.1.105 #define EMC_DYN_APRL (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x03C))  
  
6.161.1.106 #define EMC_DYN_CFG0 (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x100))  
  
6.161.1.107 #define EMC_DYN_CFG1 (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x140))  
  
6.161.1.108 #define EMC_DYN_CFG2 (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x160))  
  
6.161.1.109 #define EMC_DYN_CFG3 (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x180))  
  
6.161.1.110 #define EMC_DYN_CTRL (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x020))  
  
6.161.1.111 #define EMC_DYN_DAL (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x040))  
  
6.161.1.112 #define EMC_DYN_MRDL (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x058))
```

```
6.161.1.113 #define EMC_DYN_RAS (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x034))  
  
6.161.1.114 #define EMC_DYN_RASCAS0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x104))  
  
6.161.1.115 #define EMC_DYN_RASCAS1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x144))  
  
6.161.1.116 #define EMC_DYN_RASCAS2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x164))  
  
6.161.1.117 #define EMC_DYN_RASCAS3 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x184))  
  
6.161.1.118 #define EMC_DYN_RC (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x048))  
  
6.161.1.119 #define EMC_DYN_RD_CFG (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x028))  
  
6.161.1.120 #define EMC_DYN_RFC (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x04C))  
  
6.161.1.121 #define EMC_DYN_RFSH (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x024))  
  
6.161.1.122 #define EMC_DYN_RP (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x030))  
  
6.161.1.123 #define EMC_DYN_RRD (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x054))  
  
6.161.1.124 #define EMC_DYN_SREX (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x038))  
  
6.161.1.125 #define EMC_DYN_WR (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x044))  
  
6.161.1.126 #define EMC_DYN_XSR (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x050))  
  
6.161.1.127 #define EMC_STA_CFG0 (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x200))  
  
6.161.1.128 #define EMC_STA_CFG1 (*(volatile unsigned long *) (EMC_BASE_ADDR +  
0x220))
```

```
6.161.1.129 #define EMC_STA_CFG2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x240))  
6.161.1.130 #define EMC_STA_CFG3 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x260))  
6.161.1.131 #define EMC_STA_EXT_WAIT (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x880))  
6.161.1.132 #define EMC_STA_WAITOENO (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x208))  
6.161.1.133 #define EMC_STA_WAITOEN1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x228))  
6.161.1.134 #define EMC_STA_WAITOEN2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x248))  
6.161.1.135 #define EMC_STA_WAITOEN3 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x268))  
6.161.1.136 #define EMC_STA_WAITPAGE0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x210))  
6.161.1.137 #define EMC_STA_WAITPAGE1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x230))  
6.161.1.138 #define EMC_STA_WAITPAGE2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x250))  
6.161.1.139 #define EMC_STA_WAITPAGE3 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x270))  
6.161.1.140 #define EMC_STA_WAITRD0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x20C))  
6.161.1.141 #define EMC_STA_WAITRD1 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x22C))  
6.161.1.142 #define EMC_STA_WAITRD2 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x24C))  
6.161.1.143 #define EMC_STA_WAITRD3 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x26C))  
6.161.1.144 #define EMC_STA_WAITTURN0 (*(volatile unsigned long *) (EMC_BASE_ADDR + 0x218))
```

```
6.161.1.145 #define EMC_STA_WAITTURN1 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x238))  
  
6.161.1.146 #define EMC_STA_WAITTURN2 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x258))  
  
6.161.1.147 #define EMC_STA_WAITTURN3 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x278))  
  
6.161.1.148 #define EMC_STA_WAITWEN0 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x204))  
  
6.161.1.149 #define EMC_STA_WAITWEN1 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x224))  
  
6.161.1.150 #define EMC_STA_WAITWEN2 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x244))  
  
6.161.1.151 #define EMC_STA_WAITWEN3 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x264))  
  
6.161.1.152 #define EMC_STA_WAITWR0 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x214))  
  
6.161.1.153 #define EMC_STA_WAITWR1 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x234))  
  
6.161.1.154 #define EMC_STA_WAITWR2 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x254))  
  
6.161.1.155 #define EMC_STA_WAITWR3 (*(volatile unsigned long  
*)(EMC_BASE_ADDR + 0x274))  
  
6.161.1.156 #define EMC_STAT (*(volatile unsigned long *)(EMC_BASE_ADDR + 0x004))  
  
6.161.1.157 #define EOT_INT_CLR (*(volatile unsigned long *)(USB_BASE_ADDR +  
0xA4))  
  
6.161.1.158 #define EOT_INT_SET (*(volatile unsigned long *)(USB_BASE_ADDR +  
0xA8))  
  
6.161.1.159 #define EOT_INT_STAT (*(volatile unsigned long *)(USB_BASE_ADDR +  
0xA0))  
  
6.161.1.160 #define EP_DMA_DIS (*(volatile unsigned long *)(USB_BASE_ADDR + 0x8C))  
  
6.161.1.161 #define EP_DMA_EN (*(volatile unsigned long *)(USB_BASE_ADDR + 0x88))
```

```
6.161.1.162 #define EP_DMA_STAT (*(volatile unsigned long *) (USB_BASE_ADDR + 0x84))  
6.161.1.163 #define EP_INDEX (*(volatile unsigned long *) (USB_BASE_ADDR + 0x48))  
6.161.1.164 #define EP_INT_CLR (*(volatile unsigned long *) (USB_BASE_ADDR + 0x38))  
6.161.1.165 #define EP_INT_EN (*(volatile unsigned long *) (USB_BASE_ADDR + 0x34))  
6.161.1.166 #define EP_INT_PRIO (*(volatile unsigned long *) (USB_BASE_ADDR + 0x40))  
6.161.1.167 #define EP_INT_SET (*(volatile unsigned long *) (USB_BASE_ADDR + 0x3C))  
6.161.1.168 #define EP_INT_STAT (*(volatile unsigned long *) (USB_BASE_ADDR + 0x30))  
6.161.1.169 #define EXTINT (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x140))  
6.161.1.170 #define EXTMODE (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x148))  
6.161.1.171 #define EXTPOLAR (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x14C))  
6.161.1.172 #define FIO0CLR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x1C))  
6.161.1.173 #define FIO0CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1C))  
6.161.1.174 #define FIO0CLR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1D))  
6.161.1.175 #define FIO0CLR2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1E))  
6.161.1.176 #define FIO0CLR3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1F))  
6.161.1.177 #define FIO0CLRL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x1C))  
6.161.1.178 #define FIO0CLRU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x1E))  
6.161.1.179 #define FIO0DIR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x00))  
6.161.1.180 #define FIO0DIR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x00))  
6.161.1.181 #define FIO0DIR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x01))  
6.161.1.182 #define FIO0DIR2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x02))  
6.161.1.183 #define FIO0DIR3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x03))  
6.161.1.184 #define FIO0DIRL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x00))
```

```
6.161.1.185 #define FIO0DIRU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x02))  
6.161.1.186 #define FIO0MASK (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x10))  
6.161.1.187 #define FIO0MASK0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x10))  
6.161.1.188 #define FIO0MASK1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x11))  
6.161.1.189 #define FIO0MASK2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x12))  
6.161.1.190 #define FIO0MASK3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x13))  
6.161.1.191 #define FIO0MASKL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x10))  
6.161.1.192 #define FIO0MASKU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x12))  
6.161.1.193 #define FIO0PIN (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x14))  
6.161.1.194 #define FIO0PINO (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x14))  
6.161.1.195 #define FIO0PIN1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x15))  
6.161.1.196 #define FIO0PIN2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x16))  
6.161.1.197 #define FIO0PIN3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x17))  
6.161.1.198 #define FIO0PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x14))  
6.161.1.199 #define FIO0PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x16))  
6.161.1.200 #define FIO0SET (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x18))  
6.161.1.201 #define FIO0SET0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x18))  
6.161.1.202 #define FIO0SET1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x19))  
6.161.1.203 #define FIO0SET2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1A))  
6.161.1.204 #define FIO0SET3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x1B))  
6.161.1.205 #define FIO0SETL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x18))  
6.161.1.206 #define FIO0SETU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x1A))  
6.161.1.207 #define FIO1CLR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x3C))  
6.161.1.208 #define FIO1CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x3C))
```

```
6.161.1.209 #define FIO1CLR1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x2D))

6.161.1.210 #define FIO1CLR2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x3E))

6.161.1.211 #define FIO1CLR3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x3F))

6.161.1.212 #define FIO1CLRL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x3C))

6.161.1.213 #define FIO1CLRU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x3E))

6.161.1.214 #define FIO1DIR (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x20))

6.161.1.215 #define FIO1DIR0 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x20))

6.161.1.216 #define FIO1DIR1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x21))

6.161.1.217 #define FIO1DIR2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x22))

6.161.1.218 #define FIO1DIR3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x23))

6.161.1.219 #define FIO1DIRL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x20))

6.161.1.220 #define FIO1DIRU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x22))

6.161.1.221 #define FIO1MASK (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x30))

6.161.1.222 #define FIO1MASK0 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x30))

6.161.1.223 #define FIO1MASK1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x21))

6.161.1.224 #define FIO1MASK2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x32))

6.161.1.225 #define FIO1MASK3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x33))

6.161.1.226 #define FIO1MASKL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x30))

6.161.1.227 #define FIO1MASKU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x32))

6.161.1.228 #define FIO1PIN (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x34))

6.161.1.229 #define FIO1PIN0 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x34))

6.161.1.230 #define FIO1PIN1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x25))

6.161.1.231 #define FIO1PIN2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x36))

6.161.1.232 #define FIO1PIN3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x37))
```

```
6.161.1.233 #define FIO1PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x34))  
6.161.1.234 #define FIO1PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x36))  
6.161.1.235 #define FIO1SET (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x38))  
6.161.1.236 #define FIO1SET0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x38))  
6.161.1.237 #define FIO1SET1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x29))  
6.161.1.238 #define FIO1SET2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x3A))  
6.161.1.239 #define FIO1SET3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x3B))  
6.161.1.240 #define FIO1SETL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x38))  
6.161.1.241 #define FIO1SETU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x3A))  
6.161.1.242 #define FIO2CLR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x5C))  
6.161.1.243 #define FIO2CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5C))  
6.161.1.244 #define FIO2CLR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5D))  
6.161.1.245 #define FIO2CLR2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5E))  
6.161.1.246 #define FIO2CLR3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5F))  
6.161.1.247 #define FIO2CLRL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x5C))  
6.161.1.248 #define FIO2CLRU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x5E))  
6.161.1.249 #define FIO2DIR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x40))  
6.161.1.250 #define FIO2DIR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x40))  
6.161.1.251 #define FIO2DIR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x41))  
6.161.1.252 #define FIO2DIR2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x42))  
6.161.1.253 #define FIO2DIR3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x43))  
6.161.1.254 #define FIO2DIRL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x40))  
6.161.1.255 #define FIO2DIRU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x42))  
6.161.1.256 #define FIO2MASK (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x50))
```

```
6.161.1.257 #define FIO2MASK0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x50))

6.161.1.258 #define FIO2MASK1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x51))

6.161.1.259 #define FIO2MASK2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x52))

6.161.1.260 #define FIO2MASK3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x53))

6.161.1.261 #define FIO2MASKL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x50))

6.161.1.262 #define FIO2MASKU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x52))

6.161.1.263 #define FIO2PIN (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x54))

6.161.1.264 #define FIO2PINO (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x54))

6.161.1.265 #define FIO2PIN1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x55))

6.161.1.266 #define FIO2PIN2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x56))

6.161.1.267 #define FIO2PIN3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x57))

6.161.1.268 #define FIO2PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x54))

6.161.1.269 #define FIO2PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x56))

6.161.1.270 #define FIO2SET (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x58))

6.161.1.271 #define FIO2SET0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x58))

6.161.1.272 #define FIO2SET1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x59))

6.161.1.273 #define FIO2SET2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5A))

6.161.1.274 #define FIO2SET3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x5B))

6.161.1.275 #define FIO2SETL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x58))

6.161.1.276 #define FIO2SETU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x5A))

6.161.1.277 #define FIO3CLR (*(volatile unsigned long *) (FIO_BASE_ADDR + 0x7C))

6.161.1.278 #define FIO3CLR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x7C))

6.161.1.279 #define FIO3CLR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x7D))

6.161.1.280 #define FIO3CLR2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x7E))
```

```
6.161.1.281 #define FIO3CLR3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x7F))  
6.161.1.282 #define FIO3CLRL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x7C))  
6.161.1.283 #define FIO3CLRU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x7E))  
6.161.1.284 #define FIO3DIR (* (volatile unsigned long *) (FIO_BASE_ADDR + 0x60))  
6.161.1.285 #define FIO3DIR0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x60))  
6.161.1.286 #define FIO3DIR1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x61))  
6.161.1.287 #define FIO3DIR2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x62))  
6.161.1.288 #define FIO3DIR3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x63))  
6.161.1.289 #define FIO3DIRL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x60))  
6.161.1.290 #define FIO3DIRU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x62))  
6.161.1.291 #define FIO3MASK (* (volatile unsigned long *) (FIO_BASE_ADDR + 0x70))  
6.161.1.292 #define FIO3MASK0 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x70))  
6.161.1.293 #define FIO3MASK1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x71))  
6.161.1.294 #define FIO3MASK2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x72))  
6.161.1.295 #define FIO3MASK3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x73))  
6.161.1.296 #define FIO3MASKL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x70))  
6.161.1.297 #define FIO3MASKU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x72))  
6.161.1.298 #define FIO3PIN (* (volatile unsigned long *) (FIO_BASE_ADDR + 0x74))  
6.161.1.299 #define FIO3PINO (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x74))  
6.161.1.300 #define FIO3PIN1 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x75))  
6.161.1.301 #define FIO3PIN2 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x76))  
6.161.1.302 #define FIO3PIN3 (*(volatile unsigned char *) (FIO_BASE_ADDR + 0x77))  
6.161.1.303 #define FIO3PINL (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x74))  
6.161.1.304 #define FIO3PINU (*(volatile unsigned short *) (FIO_BASE_ADDR + 0x76))
```

```
6.161.1.305 #define FIO3SET (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x78))

6.161.1.306 #define FIO3SET0 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x78))

6.161.1.307 #define FIO3SET1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x79))

6.161.1.308 #define FIO3SET2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x7A))

6.161.1.309 #define FIO3SET3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x7B))

6.161.1.310 #define FIO3SETL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x78))

6.161.1.311 #define FIO3SETU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x7A))

6.161.1.312 #define FIO4CLR (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x9C))

6.161.1.313 #define FIO4CLR0 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x9C))

6.161.1.314 #define FIO4CLR1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x9D))

6.161.1.315 #define FIO4CLR2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x9E))

6.161.1.316 #define FIO4CLR3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x9F))

6.161.1.317 #define FIO4CLRL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x9C))

6.161.1.318 #define FIO4CLRU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x9E))

6.161.1.319 #define FIO4DIR (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x80))

6.161.1.320 #define FIO4DIR0 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x80))

6.161.1.321 #define FIO4DIR1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x81))

6.161.1.322 #define FIO4DIR2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x82))

6.161.1.323 #define FIO4DIR3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x83))

6.161.1.324 #define FIO4DIRL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x80))

6.161.1.325 #define FIO4DIRU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x82))

6.161.1.326 #define FIO4MASK (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x90))

6.161.1.327 #define FIO4MASK0 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x90))

6.161.1.328 #define FIO4MASK1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x91))
```

```
6.161.1.329 #define FIO4MASK2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x92))  
6.161.1.330 #define FIO4MASK3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x93))  
6.161.1.331 #define FIO4MASKL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x90))  
6.161.1.332 #define FIO4MASKU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x92))  
6.161.1.333 #define FIO4PIN (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x94))  
6.161.1.334 #define FIO4PINO (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x94))  
6.161.1.335 #define FIO4PIN1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x95))  
6.161.1.336 #define FIO4PIN2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x96))  
6.161.1.337 #define FIO4PIN3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x97))  
6.161.1.338 #define FIO4PINL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x94))  
6.161.1.339 #define FIO4PINU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x96))  
6.161.1.340 #define FIO4SET (*(volatile unsigned long *)(FIO_BASE_ADDR + 0x98))  
6.161.1.341 #define FIO4SET0 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x98))  
6.161.1.342 #define FIO4SET1 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x99))  
6.161.1.343 #define FIO4SET2 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x9A))  
6.161.1.344 #define FIO4SET3 (*(volatile unsigned char *)(FIO_BASE_ADDR + 0x9B))  
6.161.1.345 #define FIO4SETL (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x98))  
6.161.1.346 #define FIO4SETU (*(volatile unsigned short *)(FIO_BASE_ADDR + 0x9A))  
6.161.1.347 #define FIO_BASE_ADDR 0x3FFC000  
6.161.1.348 #define GPDMA_CH0_CFG (*(volatile unsigned long *)(DMA_BASE_ADDR  
+ 0x110))  
6.161.1.349 #define GPDMA_CH0_CTRL (*(volatile unsigned long *)(DMA_BASE_ADDR  
+ 0x10C))  
6.161.1.350 #define GPDMA_CH0_DEST (*(volatile unsigned long *)(DMA_BASE_ADDR  
+ 0x104))
```

```
6.161.1.351 #define GPDMA_CH0_LLI (*(volatile unsigned long *) (DMA_BASE_ADDR +  
+ 0x108))  
  
6.161.1.352 #define GPDMA_CH0_SRC (*(volatile unsigned long *) (DMA_BASE_ADDR  
+ 0x100))  
  
6.161.1.353 #define GPDMA_CH1_CFG (*(volatile unsigned long *) (DMA_BASE_ADDR  
+ 0x130))  
  
6.161.1.354 #define GPDMA_CH1_CTRL (*(volatile unsigned long *) (DMA_BASE_ADDR  
+ 0x12C))  
  
6.161.1.355 #define GPDMA_CH1_DEST (*(volatile unsigned long *) (DMA_BASE_ADDR  
+ 0x124))  
  
6.161.1.356 #define GPDMA_CH1_LLI (*(volatile unsigned long *) (DMA_BASE_ADDR +  
0x128))  
  
6.161.1.357 #define GPDMA_CH1_SRC (*(volatile unsigned long *) (DMA_BASE_ADDR +  
0x120))  
  
6.161.1.358 #define GPDMA_CONFIG (*(volatile unsigned long *) (DMA_BASE_ADDR +  
0x030))  
  
6.161.1.359 #define GPDMA_ENABLED_CHNS (*(volatile unsigned long *  
) (DMA_BASE_ADDR + 0x01C))  
  
6.161.1.360 #define GPDMA_INT_ERR_CLR (*(volatile unsigned long *  
) (DMA_BASE_ADDR + 0x010))  
  
6.161.1.361 #define GPDMA_INT_ERR_STAT (*(volatile unsigned long *  
) (DMA_BASE_ADDR + 0x00C))  
  
6.161.1.362 #define GPDMA_INT_STAT (*(volatile unsigned long *) (DMA_BASE_ADDR  
+ 0x000))  
  
6.161.1.363 #define GPDMA_INT_TCCLR (*(volatile unsigned long *  
) (DMA_BASE_ADDR + 0x008))  
  
6.161.1.364 #define GPDMA_INT_TCSTAT (*(volatile unsigned long *  
) (DMA_BASE_ADDR + 0x004))  
  
6.161.1.365 #define GPDMA_RAW_INT_ERR_STAT (*(volatile unsigned long *  
) (DMA_BASE_ADDR + 0x018))  
  
6.161.1.366 #define GPDMA_RAW_INT_TCSTAT (*(volatile unsigned long *  
) (DMA_BASE_ADDR + 0x014))
```

```
6.161.1.367 #define GPDMA_SOFT_BREQ (*(volatile unsigned long  
*)(DMA_BASE_ADDR + 0x020))  
  
6.161.1.368 #define GPDMA_SOFT_LBREQ (*(volatile unsigned long  
*)(DMA_BASE_ADDR + 0x028))  
  
6.161.1.369 #define GPDMA_SOFT_LSREQ (*(volatile unsigned long  
*)(DMA_BASE_ADDR + 0x02C))  
  
6.161.1.370 #define GPDMA_SOFT_SREQ (*(volatile unsigned long  
*)(DMA_BASE_ADDR + 0x024))  
  
6.161.1.371 #define GPDMA_SYNC (*(volatile unsigned long *)(DMA_BASE_ADDR +  
0x034))  
  
6.161.1.372 #define GPIO_BASE_ADDR 0xE0028000  
  
6.161.1.373 #define HC_BULK_CUR_ED (*(volatile unsigned long  
*)(USBHC_BASE_ADDR + 0x2C))  
  
6.161.1.374 #define HC_BULK_HEAD_ED (*(volatile unsigned long  
*)(USBHC_BASE_ADDR + 0x28))  
  
6.161.1.375 #define HC_CMD_STAT (*(volatile unsigned long *)(USBHC_BASE_ADDR +  
0x08))  
  
6.161.1.376 #define HC_CONTROL (*(volatile unsigned long *)(USBHC_BASE_ADDR +  
0x04))  
  
6.161.1.377 #define HC_CTRL_CUR_ED (*(volatile unsigned long  
*)(USBHC_BASE_ADDR + 0x24))  
  
6.161.1.378 #define HC_CTRL_HEAD_ED (*(volatile unsigned long  
*)(USBHC_BASE_ADDR + 0x20))  
  
6.161.1.379 #define HC_DONE_HEAD (*(volatile unsigned long *)(USBHC_BASE_ADDR  
+ 0x30))  
  
6.161.1.380 #define HC_FM_INTERVAL (*(volatile unsigned long  
*)(USBHC_BASE_ADDR + 0x34))  
  
6.161.1.381 #define HC_FM_NUMBER (*(volatile unsigned long *)(USBHC_BASE_ADDR  
+ 0x3C))  
  
6.161.1.382 #define HC_FM_REMAINING (*(volatile unsigned long  
*)(USBHC_BASE_ADDR + 0x38))  
  
6.161.1.383 #define HC_HCCA (*(volatile unsigned long *)(USBHC_BASE_ADDR + 0x18))
```

```
6.161.1.384 #define HC_INT_DIS (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x14))  
6.161.1.385 #define HC_INT_EN (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x10))  
6.161.1.386 #define HC_INT_STAT (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x0C))  
6.161.1.387 #define HC_LS_THRHLD (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x44))  
6.161.1.388 #define HC_PERIOD_CUR_ED (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x1C))  
6.161.1.389 #define HC_PERIOD_START (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x40))  
6.161.1.390 #define HC_REVISION (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x00))  
6.161.1.391 #define HC_RH_DESCA (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x48))  
6.161.1.392 #define HC_RH_DESCB (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x4C))  
6.161.1.393 #define HC_RH_PORT_STAT1 (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x54))  
6.161.1.394 #define HC_RH_PORT_STAT2 (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x58))  
6.161.1.395 #define HC_RH_STAT (*(volatile unsigned long *) (USBHC_BASE_ADDR + 0x50))  
6.161.1.396 #define I20ADR (*(volatile unsigned long *) (I2C0_BASE_ADDR + 0x0C))  
6.161.1.397 #define I20CONCLR (*(volatile unsigned long *) (I2C0_BASE_ADDR + 0x18))  
6.161.1.398 #define I20CONSET (*(volatile unsigned long *) (I2C0_BASE_ADDR + 0x00))  
6.161.1.399 #define I20DAT (*(volatile unsigned long *) (I2C0_BASE_ADDR + 0x08))  
6.161.1.400 #define I20SCLH (*(volatile unsigned long *) (I2C0_BASE_ADDR + 0x10))  
6.161.1.401 #define I20SCLL (*(volatile unsigned long *) (I2C0_BASE_ADDR + 0x14))
```

```
6.161.1.402 #define I20STAT (*(volatile unsigned long *) (I2C0_BASE_ADDR + 0x04))  
6.161.1.403 #define I21ADR (*(volatile unsigned long *) (I2C1_BASE_ADDR + 0x0C))  
6.161.1.404 #define I21CONCLR (*(volatile unsigned long *) (I2C1_BASE_ADDR + 0x18))  
6.161.1.405 #define I21CONSET (*(volatile unsigned long *) (I2C1_BASE_ADDR + 0x00))  
6.161.1.406 #define I21DAT (*(volatile unsigned long *) (I2C1_BASE_ADDR + 0x08))  
6.161.1.407 #define I21SCLH (*(volatile unsigned long *) (I2C1_BASE_ADDR + 0x10))  
6.161.1.408 #define I21SCLL (*(volatile unsigned long *) (I2C1_BASE_ADDR + 0x14))  
6.161.1.409 #define I21STAT (*(volatile unsigned long *) (I2C1_BASE_ADDR + 0x04))  
6.161.1.410 #define I22ADR (*(volatile unsigned long *) (I2C2_BASE_ADDR + 0x0C))  
6.161.1.411 #define I22CONCLR (*(volatile unsigned long *) (I2C2_BASE_ADDR + 0x18))  
6.161.1.412 #define I22CONSET (*(volatile unsigned long *) (I2C2_BASE_ADDR + 0x00))  
6.161.1.413 #define I22DAT (*(volatile unsigned long *) (I2C2_BASE_ADDR + 0x08))  
6.161.1.414 #define I22SCLH (*(volatile unsigned long *) (I2C2_BASE_ADDR + 0x10))  
6.161.1.415 #define I22SCLL (*(volatile unsigned long *) (I2C2_BASE_ADDR + 0x14))  
6.161.1.416 #define I22STAT (*(volatile unsigned long *) (I2C2_BASE_ADDR + 0x04))  
6.161.1.417 #define I2C0_BASE_ADDR 0xE001C000  
6.161.1.418 #define I2C1_BASE_ADDR 0xE005C000  
6.161.1.419 #define I2C2_BASE_ADDR 0xE0080000  
6.161.1.420 #define I2S_BASE_ADDR 0xE0088000  
6.161.1.421 #define I2S_DAI (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x04))  
6.161.1.422 #define I2S.DAO (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x00))  
6.161.1.423 #define I2S_DMA1 (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x14))  
6.161.1.424 #define I2S_DMA2 (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x18))  
6.161.1.425 #define I2S_IRQ (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x1C))
```

```
6.161.1.426 #define I2S_RX_FIFO (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x0C))

6.161.1.427 #define I2S_RXRATE (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x24))

6.161.1.428 #define I2S_STATE (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x10))

6.161.1.429 #define I2S_TX_FIFO (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x08))

6.161.1.430 #define I2S_TXRATE (*(volatile unsigned long *) (I2S_BASE_ADDR + 0x20))

6.161.1.431 #define INTWAKE (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x144))

6.161.1.432 #define IO0_INT_CLR (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0x8C))

6.161.1.433 #define IO0_INT_EN_F (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0x94))

6.161.1.434 #define IO0_INT_EN_R (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0x90))

6.161.1.435 #define IO0_INT_STAT_F (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0x88))

6.161.1.436 #define IO0_INT_STAT_R (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0x84))

6.161.1.437 #define IO2_INT_CLR (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0xAC))

6.161.1.438 #define IO2_INT_EN_F (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0xB4))

6.161.1.439 #define IO2_INT_EN_R (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0xB0))

6.161.1.440 #define IO2_INT_STAT_F (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0xA8))

6.161.1.441 #define IO2_INT_STAT_R (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0xA4))

6.161.1.442 #define IO_INT_STAT (*(volatile unsigned long *) (GPIO_BASE_ADDR +
0x80))

6.161.1.443 #define IOCLR0 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x0C))

6.161.1.444 #define IOCLR1 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x1C))
```

```
6.161.1.445 #define IODIR0 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x08))  
6.161.1.446 #define IODIR1 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x18))  
6.161.1.447 #define IOPIN0 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x00))  
6.161.1.448 #define IOPIN1 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x10))  
6.161.1.449 #define IOSET0 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x04))  
6.161.1.450 #define IOSET1 (*(volatile unsigned long *) (GPIO_BASE_ADDR + 0x14))  
6.161.1.451 #define MAC_BASE_ADDR 0xFFE00000 /* AHB Peripheral # 0 */  
6.161.1.452 #define MAC_CLRT (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x010))  
/* CoLlision window/ReTry reg */  
6.161.1.453 #define MAC_COMMAND (*(volatile unsigned long *) (MAC_BASE_ADDR +  
0x100)) /* Command reg */  
6.161.1.454 #define MAC_FLOWCONTROLCNT (*(volatile unsigned long *  
*)(MAC_BASE_ADDR + 0x170)) /* Flow control counter reg */  
6.161.1.455 #define MAC_FLOWCONTROLSTS (*(volatile unsigned long *  
*)(MAC_BASE_ADDR + 0x174)) /* Flow control status reg */  
6.161.1.456 #define MAC_HASHFILTERH (*(volatile unsigned long *  
*)(MAC_BASE_ADDR + 0x214)) /* Hash filter MSBs reg */  
6.161.1.457 #define MAC_HASHFILTERL (*(volatile unsigned long *  
*)(MAC_BASE_ADDR + 0x210)) /* Hash filter LSBs reg */  
6.161.1.458 #define MAC_INTCLEAR (*(volatile unsigned long *) (MAC_BASE_ADDR +  
0xFE8)) /* Interrupt clear reg (WO) */  
6.161.1.459 #define MAC_INTENABLE (*(volatile unsigned long *) (MAC_BASE_ADDR +  
0xFE4)) /* Interrupt enable reg */  
6.161.1.460 #define MAC_INTSET (*(volatile unsigned long *) (MAC_BASE_ADDR +  
0xFEC)) /* Interrupt set reg (WO) */  
6.161.1.461 #define MAC_INTSTATUS (*(volatile unsigned long *) (MAC_BASE_ADDR +  
0xFE0)) /* Interrupt status reg (RO) */  
6.161.1.462 #define MAC_IPGR (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x00C))  
/* non b2b InterPacketGap reg */
```

```
6.161.1.463 #define MAC_IPGT (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x008))  
/* b2b InterPacketGap reg */  
  
6.161.1.464 #define MAC_MAC1 (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x000))  
/* MAC config reg 1 */  
  
6.161.1.465 #define MAC_MAC2 (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x004))  
/* MAC config reg 2 */  
  
6.161.1.466 #define MAC_MADR (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x028))  
/* MII Mgmt ADDress reg */  
  
6.161.1.467 #define MAC_MAXF (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x014))  
/* MAXimum Frame reg */  
  
6.161.1.468 #define MAC_MCFG (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x020))  
/* MII Mgmt ConFiG reg */  
  
6.161.1.469 #define MAC_MCMD (*(volatile unsigned long *) (MAC_BASE_ADDR +  
0x024)) /* MII Mgmt CoMManD reg */  
  
6.161.1.470 #define MAC_MIND (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x034))  
/* MII Mgmt INDicators reg (RO) */  
  
6.161.1.471 #define MAC_MODULEID (*(volatile unsigned long *) (MAC_BASE_ADDR +  
0xFFC)) /* Module ID reg (RO) */  
  
6.161.1.472 #define MAC_MRDD (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x030))  
/* MII Mgmt ReaD Data reg (RO) */  
  
6.161.1.473 #define MAC_MWTD (*(volatile unsigned long *) (MAC_BASE_ADDR +  
0x02C)) /* MII Mgmt WriTe Data reg (WO) */  
  
6.161.1.474 #define MAC_POWERDOWN (*(volatile unsigned long *)  
(MAC_BASE_ADDR + 0xFF4)) /* Power-down reg */  
  
6.161.1.475 #define MAC_RSV (*(volatile unsigned long *) (MAC_BASE_ADDR + 0x160))  
/* Rx status vector reg (RO) */  
  
6.161.1.476 #define MAC_RXCONSUMEINDEX (*(volatile unsigned long *)  
(MAC_BASE_ADDR + 0x118)) /* Rx consume index reg */  
  
6.161.1.477 #define MAC_RXDESCRIPTOR (*(volatile unsigned long *)  
(MAC_BASE_ADDR + 0x108)) /* Rx descriptor base address reg */  
  
6.161.1.478 #define MAC_RXDESCRIPTORMNUM (*(volatile unsigned long *)  
(MAC_BASE_ADDR + 0x110)) /* Rx number of descriptors reg */
```

```
6.161.1.479 #define MAC_RXFILTERCTRL (*(volatile unsigned long  
*)(MAC_BASE_ADDR + 0x200)) /* Rx filter ctrl reg */  
  
6.161.1.480 #define MAC_RXFILTERWOLCLR (*(volatile unsigned long  
*)(MAC_BASE_ADDR + 0x208)) /* Rx filter WoL clear reg (WO) */  
  
6.161.1.481 #define MAC_RXFILTERWOLSTS (*(volatile unsigned long  
*)(MAC_BASE_ADDR + 0x204)) /* Rx filter WoL status reg (RO) */  
  
6.161.1.482 #define MAC_RXPRODUCEINDEX (*(volatile unsigned long  
*)(MAC_BASE_ADDR + 0x114)) /* Rx produce index reg (RO) */  
  
6.161.1.483 #define MAC_RXSTATUS (*(volatile unsigned long *)(MAC_BASE_ADDR +  
0x10C)) /* Rx status base address reg */  
  
6.161.1.484 #define MAC_SA0 (*(volatile unsigned long *)(MAC_BASE_ADDR + 0x040)) /*  
Station Address 0 reg */  
  
6.161.1.485 #define MAC_SA1 (*(volatile unsigned long *)(MAC_BASE_ADDR + 0x044)) /*  
Station Address 1 reg */  
  
6.161.1.486 #define MAC_SA2 (*(volatile unsigned long *)(MAC_BASE_ADDR + 0x048)) /*  
Station Address 2 reg */  
  
6.161.1.487 #define MAC_STATUS (*(volatile unsigned long *)(MAC_BASE_ADDR +  
0x104)) /* Status reg (RO) */  
  
6.161.1.488 #define MAC_SUPP (*(volatile unsigned long *)(MAC_BASE_ADDR + 0x018))  
/* PHY SUPPort reg */  
  
6.161.1.489 #define MAC_TEST (*(volatile unsigned long *)(MAC_BASE_ADDR + 0x01C))  
/* TEST reg */  
  
6.161.1.490 #define MAC_TSV0 (*(volatile unsigned long *)(MAC_BASE_ADDR + 0x158))  
/* Tx status vector 0 reg (RO) */  
  
6.161.1.491 #define MAC_TSV1 (*(volatile unsigned long *)(MAC_BASE_ADDR + 0x15C))  
/* Tx status vector 1 reg (RO) */  
  
6.161.1.492 #define MAC_TXCONSUMEINDEX (*(volatile unsigned long  
*)(MAC_BASE_ADDR + 0x12C)) /* Tx consume index reg (RO) */  
  
6.161.1.493 #define MAC_TXDESCRIPTOR (*(volatile unsigned long  
*)(MAC_BASE_ADDR + 0x11C)) /* Tx descriptor base address reg */  
  
6.161.1.494 #define MAC_TXDESCRIPTORMNUM (*(volatile unsigned long  
*)(MAC_BASE_ADDR + 0x124)) /* Tx number of descriptors reg */
```

```
6.161.1.495 #define MAC_TXPRODUCEINDEX (*(volatile unsigned long  
*)(MAC_BASE_ADDR + 0x128)) /* Tx produce index reg */  
  
6.161.1.496 #define MAC_TXSTATUS (*(volatile unsigned long *)(MAC_BASE_ADDR +  
0x120)) /* Tx status base address reg */  
  
6.161.1.497 #define MAMCR (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x000))  
  
6.161.1.498 #define MAMTIM (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x004))  
  
6.161.1.499 #define MAXPACKET_SIZE (*(volatile unsigned long *)(USB_BASE_ADDR  
+ 0x4C))  
  
6.161.1.500 #define MCI_ARGUMENT (*(volatile unsigned long *)(MCI_BASE_ADDR +  
0x08))  
  
6.161.1.501 #define MCI_BASE_ADDR 0xE008C000  
  
6.161.1.502 #define MCI_CLEAR (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x38))  
  
6.161.1.503 #define MCI_CLOCK (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x04))  
  
6.161.1.504 #define MCI_COMMAND (*(volatile unsigned long *)(MCI_BASE_ADDR +  
0x0C))  
  
6.161.1.505 #define MCI_DATA_CNT (*(volatile unsigned long *)(MCI_BASE_ADDR +  
0x30))  
  
6.161.1.506 #define MCI_DATA_CTRL (*(volatile unsigned long *)(MCI_BASE_ADDR +  
0x2C))  
  
6.161.1.507 #define MCI_DATA_LEN (*(volatile unsigned long *)(MCI_BASE_ADDR +  
0x28))  
  
6.161.1.508 #define MCI_DATA_TMR (*(volatile unsigned long *)(MCI_BASE_ADDR +  
0x24))  
  
6.161.1.509 #define MCI_FIFO (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x80))  
  
6.161.1.510 #define MCI_FIFO_CNT (*(volatile unsigned long *)(MCI_BASE_ADDR +  
0x48))  
  
6.161.1.511 #define MCI_MASK0 (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x3C))  
  
6.161.1.512 #define MCI_MASK1 (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x40))  
  
6.161.1.513 #define MCI_POWER (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x00))
```

```
6.161.1.514 #define MCI_RESP0 (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x14))

6.161.1.515 #define MCI_RESP1 (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x18))

6.161.1.516 #define MCI_RESP2 (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x1C))

6.161.1.517 #define MCI_RESP3 (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x20))

6.161.1.518 #define MCI_RESP_CMD (*(volatile unsigned long *)(MCI_BASE_ADDR +
0x10))

6.161.1.519 #define MCI_STATUS (*(volatile unsigned long *)(MCI_BASE_ADDR + 0x34))

6.161.1.520 #define MEMMAP (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x040))

6.161.1.521 #define NDD_REQ_INT_CLR (*(volatile unsigned long *)(USB_BASE_ADDR +
0xB0))

6.161.1.522 #define NDD_REQ_INT_SET (*(volatile unsigned long *)(USB_BASE_ADDR +
0xB4))

6.161.1.523 #define NDD_REQ_INT_STAT (*(volatile unsigned long
*)(USB_BASE_ADDR + 0xAC))

6.161.1.524 #define OTG_CLK_CTRL (*(volatile unsigned long
*)(USBOTG_CLK_BASE_ADDR + 0x04))

6.161.1.525 #define OTG_CLK_STAT (*(volatile unsigned long
*)(USBOTG_CLK_BASE_ADDR + 0x08))

6.161.1.526 #define OTG_I2C_CLKHI (*(volatile unsigned long
*)(USBOTG_I2C_BASE_ADDR + 0x0C))

6.161.1.527 #define OTG_I2C_CLKLO (*(volatile unsigned long
*)(USBOTG_I2C_BASE_ADDR + 0x10))

6.161.1.528 #define OTG_I2C_CTL (*(volatile unsigned long
*)(USBOTG_I2C_BASE_ADDR + 0x08))

6.161.1.529 #define OTG_I2C_RX (*(volatile unsigned long
*)(USBOTG_I2C_BASE_ADDR + 0x00))

6.161.1.530 #define OTG_I2C_STS (*(volatile unsigned long
*)(USBOTG_I2C_BASE_ADDR + 0x04))

6.161.1.531 #define OTG_I2C_TX (*(volatile unsigned long
*)(USBOTG_I2C_BASE_ADDR + 0x00))
```

```
6.161.1.532 #define OTG_INT_CLR (*(volatile unsigned long *) (USBOTG_BASE_ADDR + 0x0C))

6.161.1.533 #define OTG_INT_EN (*(volatile unsigned long *) (USBOTG_BASE_ADDR + 0x04))

6.161.1.534 #define OTG_INT_SET (*(volatile unsigned long *) (USBOTG_BASE_ADDR + 0x08))

6.161.1.535 #define OTG_INT_STAT (*(volatile unsigned long *) (USBOTG_BASE_ADDR + 0x00))

6.161.1.536 #define OTG_STAT_CTRL (*(volatile unsigned long *) (USBOTG_BASE_ADDR + 0x10))

6.161.1.537 #define OTG_TIMER (*(volatile unsigned long *) (USBOTG_BASE_ADDR + 0x14))

6.161.1.538 #define PARTCFG (*(volatile unsigned long *) (PARTCFG_BASE_ADDR + 0x00))

6.161.1.539 #define PARTCFG_BASE_ADDR 0x3FFF8000

6.161.1.540 #define PCLKSEL0 (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x1A8))

6.161.1.541 #define PCLKSEL1 (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x1AC))

6.161.1.542 #define PCON (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x0C0))

6.161.1.543 #define PCONP (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x0C4))

6.161.1.544 #define PINMODE0 (*(volatile unsigned long *) (PINSEL_BASE_ADDR + 0x40))

6.161.1.545 #define PINMODE1 (*(volatile unsigned long *) (PINSEL_BASE_ADDR + 0x44))

6.161.1.546 #define PINMODE2 (*(volatile unsigned long *) (PINSEL_BASE_ADDR + 0x48))

6.161.1.547 #define PINMODE3 (*(volatile unsigned long *) (PINSEL_BASE_ADDR + 0x4C))

6.161.1.548 #define PINMODE4 (*(volatile unsigned long *) (PINSEL_BASE_ADDR + 0x50))

6.161.1.549 #define PINMODE5 (*(volatile unsigned long *) (PINSEL_BASE_ADDR + 0x54))
```

```
6.161.1.550 #define PINMODE6 (*(volatile unsigned long *)(PINSEL_BASE_ADDR +  
0x58))  
  
6.161.1.551 #define PINMODE7 (*(volatile unsigned long *)(PINSEL_BASE_ADDR +  
0x5C))  
  
6.161.1.552 #define PINMODE8 (*(volatile unsigned long *)(PINSEL_BASE_ADDR +  
0x60))  
  
6.161.1.553 #define PINMODE9 (*(volatile unsigned long *)(PINSEL_BASE_ADDR +  
0x64))  
  
6.161.1.554 #define PINSEL0 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x00))  
  
6.161.1.555 #define PINSEL1 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x04))  
  
6.161.1.556 #define PINSEL10 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x28))  
  
6.161.1.557 #define PINSEL2 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x08))  
  
6.161.1.558 #define PINSEL3 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x0C))  
  
6.161.1.559 #define PINSEL4 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x10))  
  
6.161.1.560 #define PINSEL5 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x14))  
  
6.161.1.561 #define PINSEL6 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x18))  
  
6.161.1.562 #define PINSEL7 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x1C))  
  
6.161.1.563 #define PINSEL8 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x20))  
  
6.161.1.564 #define PINSEL9 (*(volatile unsigned long *)(PINSEL_BASE_ADDR + 0x24))  
  
6.161.1.565 #define PINSEL_BASE_ADDR 0xE002C000  
  
6.161.1.566 #define PLLCFG (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x084))  
  
6.161.1.567 #define PLLCON (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x080))  
  
6.161.1.568 #define PLLFEED (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x08C))  
  
6.161.1.569 #define PLLSTAT (*(volatile unsigned long *)(SCB_BASE_ADDR + 0x088))  
  
6.161.1.570 #define PWM0_BASE_ADDR 0xE0014000  
  
6.161.1.571 #define PWM0CCR (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x28))
```

```
6.161.1.572 #define PWM0CR0 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x2C))

6.161.1.573 #define PWM0CR1 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x30))

6.161.1.574 #define PWM0CR2 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x34))

6.161.1.575 #define PWM0CR3 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x38))

6.161.1.576 #define PWM0CTCR (*(volatile unsigned long *)(PWM0_BASE_ADDR +
0x70))

6.161.1.577 #define PWM0EMR (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x3C))

6.161.1.578 #define PWM0IR (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x00))

6.161.1.579 #define PWM0LER (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x50))

6.161.1.580 #define PWM0MCR (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x14))

6.161.1.581 #define PWM0MR0 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x18))

6.161.1.582 #define PWM0MR1 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x1C))

6.161.1.583 #define PWM0MR2 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x20))

6.161.1.584 #define PWM0MR3 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x24))

6.161.1.585 #define PWM0MR4 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x40))

6.161.1.586 #define PWM0MR5 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x44))

6.161.1.587 #define PWM0MR6 (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x48))

6.161.1.588 #define PWM0PC (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x10))

6.161.1.589 #define PWM0PCR (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x4C))

6.161.1.590 #define PWM0PR (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x0C))

6.161.1.591 #define PWM0TC (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x08))

6.161.1.592 #define PWM0TCR (*(volatile unsigned long *)(PWM0_BASE_ADDR + 0x04))

6.161.1.593 #define PWM1_BASE_ADDR 0xE0018000

6.161.1.594 #define PWM1CCR (*(volatile unsigned long *)(PWM1_BASE_ADDR + 0x28))
```

```
6.161.1.595 #define PWM1CR0 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x2C))  
6.161.1.596 #define PWM1CR1 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x30))  
6.161.1.597 #define PWM1CR2 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x34))  
6.161.1.598 #define PWM1CR3 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x38))  
6.161.1.599 #define PWM1CTCR (*(volatile unsigned long *) (PWM1_BASE_ADDR +  
0x70))  
6.161.1.600 #define PWM1EMR (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x3C))  
6.161.1.601 #define PWM1IR (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x00))  
6.161.1.602 #define PWM1LER (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x50))  
6.161.1.603 #define PWM1MCR (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x14))  
6.161.1.604 #define PWM1MRO (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x18))  
6.161.1.605 #define PWM1MR1 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x1C))  
6.161.1.606 #define PWM1MR2 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x20))  
6.161.1.607 #define PWM1MR3 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x24))  
6.161.1.608 #define PWM1MR4 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x40))  
6.161.1.609 #define PWM1MR5 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x44))  
6.161.1.610 #define PWM1MR6 (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x48))  
6.161.1.611 #define PWM1PC (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x10))  
6.161.1.612 #define PWM1PCR (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x4C))  
6.161.1.613 #define PWM1PR (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x0C))  
6.161.1.614 #define PWM1TC (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x08))  
6.161.1.615 #define PWM1TCR (*(volatile unsigned long *) (PWM1_BASE_ADDR + 0x04))  
6.161.1.616 #define REALIZE_EP (*(volatile unsigned long *) (USB_BASE_ADDR + 0x44))  
6.161.1.617 #define RSIR (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x180))
```

```
6.161.1.618 #define RTC_ALDOM (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x6C))  
6.161.1.619 #define RTC_ALDOW (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x70))  
6.161.1.620 #define RTC_ALDOY (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x74))  
6.161.1.621 #define RTC_ALHOUR (*(volatile unsigned long *) (RTC_BASE_ADDR +  
    0x68))  
6.161.1.622 #define RTC_ALMIN (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x64))  
6.161.1.623 #define RTC_ALMON (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x78))  
6.161.1.624 #define RTC_ALSEC (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x60))  
6.161.1.625 #define RTC_ALYEAR (*(volatile unsigned long *) (RTC_BASE_ADDR +  
    0x7C))  
6.161.1.626 #define RTC_AMR (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x10))  
6.161.1.627 #define RTC_BASE_ADDR 0xE0024000  
6.161.1.628 #define RTC_CCR (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x08))  
6.161.1.629 #define RTC_CIIR (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x0C))  
6.161.1.630 #define RTC_CISS (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x40))  
6.161.1.631 #define RTC_CTC (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x04))  
6.161.1.632 #define RTC_CTIME0 (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x14))  
6.161.1.633 #define RTC_CTIME1 (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x18))  
6.161.1.634 #define RTC_CTIME2 (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x1C))  
6.161.1.635 #define RTC_DOM (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x2C))  
6.161.1.636 #define RTC_DOW (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x30))  
6.161.1.637 #define RTC_DOY (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x34))  
6.161.1.638 #define RTC_HOUR (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x28))  
6.161.1.639 #define RTC_ILR (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x00))  
6.161.1.640 #define RTC_MIN (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x24))
```

```
6.161.1.641 #define RTC_MONTH (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x38))  
6.161.1.642 #define RTC_PREFRAC (*(volatile unsigned long *) (RTC_BASE_ADDR +  
0x84))  
6.161.1.643 #define RTC_PREINT (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x80))  
6.161.1.644 #define RTC_SEC (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x20))  
6.161.1.645 #define RTC_YEAR (*(volatile unsigned long *) (RTC_BASE_ADDR + 0x3C))  
6.161.1.646 #define RX_DATA (*(volatile unsigned long *) (USB_BASE_ADDR + 0x18))  
6.161.1.647 #define RX_PLENGTH (*(volatile unsigned long *) (USB_BASE_ADDR +  
0x20))  
6.161.1.648 #define S0SPCCR (*(volatile unsigned long *) (SPI0_BASE_ADDR + 0x0C))  
6.161.1.649 #define S0SPCR (*(volatile unsigned long *) (SPI0_BASE_ADDR + 0x00))  
6.161.1.650 #define S0SPDR (*(volatile unsigned long *) (SPI0_BASE_ADDR + 0x08))  
6.161.1.651 #define S0SPINT (*(volatile unsigned long *) (SPI0_BASE_ADDR + 0x1C))  
6.161.1.652 #define S0SPSR (*(volatile unsigned long *) (SPI0_BASE_ADDR + 0x04))  
6.161.1.653 #define SCB_BASE_ADDR 0xE01FC000  
6.161.1.654 #define SCS (*(volatile unsigned long *) (SCB_BASE_ADDR + 0x1A0))  
6.161.1.655 #define SPI0_BASE_ADDR 0xE0020000  
6.161.1.656 #define SSP0_BASE_ADDR 0xE0068000  
6.161.1.657 #define SSP0CPSR (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x10))  
6.161.1.658 #define SSP0CRO (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x00))  
6.161.1.659 #define SSP0CR1 (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x04))  
6.161.1.660 #define SSP0DMACR (*(volatile unsigned long *) (SSP0_BASE_ADDR +  
0x24))  
6.161.1.661 #define SSP0DR (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x08))  
6.161.1.662 #define SSP0ICR (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x20))
```

```
6.161.1.663 #define SSP0IMSC (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x14))

6.161.1.664 #define SSP0MIS (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x1C))

6.161.1.665 #define SSP0RIS (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x18))

6.161.1.666 #define SSP0SR (*(volatile unsigned long *) (SSP0_BASE_ADDR + 0x0C))

6.161.1.667 #define SSP1_BASE_ADDR 0xE0030000

6.161.1.668 #define SSP1CPSR (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x10))

6.161.1.669 #define SSP1CR0 (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x00))

6.161.1.670 #define SSP1CR1 (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x04))

6.161.1.671 #define SSP1DMACR (*(volatile unsigned long *) (SSP1_BASE_ADDR +
0x24))

6.161.1.672 #define SSP1DR (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x08))

6.161.1.673 #define SSP1ICR (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x20))

6.161.1.674 #define SSP1IMSC (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x14))

6.161.1.675 #define SSP1MIS (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x1C))

6.161.1.676 #define SSP1RIS (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x18))

6.161.1.677 #define SSP1SR (*(volatile unsigned long *) (SSP1_BASE_ADDR + 0x0C))

6.161.1.678 #define STATIC_MEM0_BASE 0x80000000

6.161.1.679 #define STATIC_MEM1_BASE 0x81000000

6.161.1.680 #define STATIC_MEM2_BASE 0x82000000

6.161.1.681 #define STATIC_MEM3_BASE 0x83000000

6.161.1.682 #define SYS_ERR_INT_CLR (*(volatile unsigned long *) (USB_BASE_ADDR +
0xBC))

6.161.1.683 #define SYS_ERR_INT_SET (*(volatile unsigned long *) (USB_BASE_ADDR +
0xC0))

6.161.1.684 #define SYS_ERR_INT_STAT (*(volatile unsigned long *
)(USB_BASE_ADDR + 0xB8))
```

```
6.161.1.685 #define T0CCR (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x28))  
6.161.1.686 #define T0CR0 (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x2C))  
6.161.1.687 #define T0CR1 (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x30))  
6.161.1.688 #define T0CR2 (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x34))  
6.161.1.689 #define T0CR3 (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x38))  
6.161.1.690 #define T0CTCR (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x70))  
6.161.1.691 #define T0EMR (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x3C))  
6.161.1.692 #define T0IR (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x00))  
6.161.1.693 #define T0MCR (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x14))  
6.161.1.694 #define T0MR0 (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x18))  
6.161.1.695 #define T0MR1 (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x1C))  
6.161.1.696 #define T0MR2 (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x20))  
6.161.1.697 #define T0MR3 (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x24))  
6.161.1.698 #define T0PC (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x10))  
6.161.1.699 #define T0PR (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x0C))  
6.161.1.700 #define T0TC (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x08))  
6.161.1.701 #define T0TCR (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x04))  
6.161.1.702 #define T1CCR (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x28))  
6.161.1.703 #define T1CR0 (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x2C))  
6.161.1.704 #define T1CR1 (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x30))  
6.161.1.705 #define T1CR2 (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x34))  
6.161.1.706 #define T1CR3 (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x38))  
6.161.1.707 #define T1CTCR (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x70))  
6.161.1.708 #define T1EMR (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x3C))
```

```
6.161.1.709 #define T1IR (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x00))

6.161.1.710 #define T1MCR (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x14))

6.161.1.711 #define T1MRO (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x18))

6.161.1.712 #define T1MR1 (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x1C))

6.161.1.713 #define T1MR2 (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x20))

6.161.1.714 #define T1MR3 (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x24))

6.161.1.715 #define T1PC (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x10))

6.161.1.716 #define T1PR (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x0C))

6.161.1.717 #define T1TC (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x08))

6.161.1.718 #define T1TCR (*(volatile unsigned long *)(TMR1_BASE_ADDR + 0x04))

6.161.1.719 #define T2CCR (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x28))

6.161.1.720 #define T2CRO (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x2C))

6.161.1.721 #define T2CR1 (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x30))

6.161.1.722 #define T2CR2 (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x34))

6.161.1.723 #define T2CR3 (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x38))

6.161.1.724 #define T2CTCR (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x70))

6.161.1.725 #define T2EMR (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x3C))

6.161.1.726 #define T2IR (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x00))

6.161.1.727 #define T2MCR (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x14))

6.161.1.728 #define T2MRO (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x18))

6.161.1.729 #define T2MR1 (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x1C))

6.161.1.730 #define T2MR2 (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x20))

6.161.1.731 #define T2MR3 (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x24))

6.161.1.732 #define T2PC (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x10))
```

```
6.161.1.733 #define T2PR (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x0C))  
6.161.1.734 #define T2TC (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x08))  
6.161.1.735 #define T2TCR (*(volatile unsigned long *)(TMR2_BASE_ADDR + 0x04))  
6.161.1.736 #define T3CCR (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x28))  
6.161.1.737 #define T3CR0 (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x2C))  
6.161.1.738 #define T3CR1 (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x30))  
6.161.1.739 #define T3CR2 (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x34))  
6.161.1.740 #define T3CR3 (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x38))  
6.161.1.741 #define T3CTCR (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x70))  
6.161.1.742 #define T3EMR (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x3C))  
6.161.1.743 #define T3IR (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x00))  
6.161.1.744 #define T3MCR (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x14))  
6.161.1.745 #define T3MR0 (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x18))  
6.161.1.746 #define T3MR1 (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x1C))  
6.161.1.747 #define T3MR2 (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x20))  
6.161.1.748 #define T3MR3 (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x24))  
6.161.1.749 #define T3PC (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x10))  
6.161.1.750 #define T3PR (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x0C))  
6.161.1.751 #define T3TC (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x08))  
6.161.1.752 #define T3TCR (*(volatile unsigned long *)(TMR3_BASE_ADDR + 0x04))  
6.161.1.753 #define TMR0_BASE_ADDR 0xE0004000  
6.161.1.754 #define TMR1_BASE_ADDR 0xE0008000  
6.161.1.755 #define TMR2_BASE_ADDR 0xE0070000  
6.161.1.756 #define TMR3_BASE_ADDR 0xE0074000
```

```
6.161.1.757 #define TX_DATA (*(volatile unsigned long *)(USB_BASE_ADDR + 0x1C))

6.161.1.758 #define TX_PLENGTH (*(volatile unsigned long *)(USB_BASE_ADDR +
0x24))

6.161.1.759 #define U0ACR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x20))

6.161.1.760 #define U0DLL (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x00))

6.161.1.761 #define U0DLM (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x04))

6.161.1.762 #define U0FCR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x08))

6.161.1.763 #define U0FDR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x28))

6.161.1.764 #define U0ICR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x24))

6.161.1.765 #define U0IER (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x04))

6.161.1.766 #define U0IIR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x08))

6.161.1.767 #define U0LCR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x0C))

6.161.1.768 #define U0LSR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x14))

6.161.1.769 #define U0RBR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x00))

6.161.1.770 #define U0SCR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x1C))

6.161.1.771 #define U0TER (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x30))

6.161.1.772 #define U0THR (*(volatile unsigned long *)(UART0_BASE_ADDR + 0x00))

6.161.1.773 #define U1ACR (*(volatile unsigned long *)(UART1_BASE_ADDR + 0x20))

6.161.1.774 #define U1DLL (*(volatile unsigned long *)(UART1_BASE_ADDR + 0x00))

6.161.1.775 #define U1DLM (*(volatile unsigned long *)(UART1_BASE_ADDR + 0x04))

6.161.1.776 #define U1FCR (*(volatile unsigned long *)(UART1_BASE_ADDR + 0x08))

6.161.1.777 #define U1FDR (*(volatile unsigned long *)(UART1_BASE_ADDR + 0x28))

6.161.1.778 #define U1IER (*(volatile unsigned long *)(UART1_BASE_ADDR + 0x04))

6.161.1.779 #define U1IIR (*(volatile unsigned long *)(UART1_BASE_ADDR + 0x08))
```

```
6.161.1.780 #define U1LCR (*(volatile unsigned long *) (UART1_BASE_ADDR + 0x0C))  
6.161.1.781 #define U1LSR (*(volatile unsigned long *) (UART1_BASE_ADDR + 0x14))  
6.161.1.782 #define U1MCR (*(volatile unsigned long *) (UART1_BASE_ADDR + 0x10))  
6.161.1.783 #define U1MSR (*(volatile unsigned long *) (UART1_BASE_ADDR + 0x18))  
6.161.1.784 #define U1RBR (*(volatile unsigned long *) (UART1_BASE_ADDR + 0x00))  
6.161.1.785 #define U1SCR (*(volatile unsigned long *) (UART1_BASE_ADDR + 0x1C))  
6.161.1.786 #define U1TER (*(volatile unsigned long *) (UART1_BASE_ADDR + 0x30))  
6.161.1.787 #define U1THR (*(volatile unsigned long *) (UART1_BASE_ADDR + 0x00))  
6.161.1.788 #define U2ACR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x20))  
6.161.1.789 #define U2DLL (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x00))  
6.161.1.790 #define U2DLM (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x04))  
6.161.1.791 #define U2FCR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x08))  
6.161.1.792 #define U2FDR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x28))  
6.161.1.793 #define U2ICR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x24))  
6.161.1.794 #define U2IER (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x04))  
6.161.1.795 #define U2IIR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x08))  
6.161.1.796 #define U2LCR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x0C))  
6.161.1.797 #define U2LSR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x14))  
6.161.1.798 #define U2RBR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x00))  
6.161.1.799 #define U2SCR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x1C))  
6.161.1.800 #define U2TER (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x30))  
6.161.1.801 #define U2THR (*(volatile unsigned long *) (UART2_BASE_ADDR + 0x00))  
6.161.1.802 #define U3ACR (*(volatile unsigned long *) (UART3_BASE_ADDR + 0x20))  
6.161.1.803 #define U3DLL (*(volatile unsigned long *) (UART3_BASE_ADDR + 0x00))
```

```
6.161.1.804 #define U3DLM (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x04))

6.161.1.805 #define U3FCR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x08))

6.161.1.806 #define U3FDR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x28))

6.161.1.807 #define U3ICR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x24))

6.161.1.808 #define U3IER (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x04))

6.161.1.809 #define U3IIR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x08))

6.161.1.810 #define U3LCR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x0C))

6.161.1.811 #define U3LSR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x14))

6.161.1.812 #define U3RBR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x00))

6.161.1.813 #define U3SCR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x1C))

6.161.1.814 #define U3TER (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x30))

6.161.1.815 #define U3THR (*(volatile unsigned long *)(UART3_BASE_ADDR + 0x00))

6.161.1.816 #define UART0_BASE_ADDR 0xE000C000

6.161.1.817 #define UART1_BASE_ADDR 0xE0010000

6.161.1.818 #define UART2_BASE_ADDR 0xE0078000

6.161.1.819 #define UART3_BASE_ADDR 0xE007C000

6.161.1.820 #define UDCA_HEAD (*(volatile unsigned long *)(USB_BASE_ADDR + 0x80))

6.161.1.821 #define USB_BASE_ADDR 0xFFE0C200 /* USB Base Address */

6.161.1.822 #define USB_CTRL (*(volatile unsigned long *)(USB_BASE_ADDR + 0x28))

6.161.1.823 #define USB_INT_BASE_ADDR 0xE01FC1C0

6.161.1.824 #define USB_INT_STAT (*(volatile unsigned long *)(USB_INT_BASE_ADDR +
+ 0x00))

6.161.1.825 #define USBCLKCFG (*(volatile unsigned long *)(SCB_BASE_ADDR +
+ 0x108))
```

```
6.161.1.826 #define USBClkCtrl (*(volatile unsigned long
    *)(USBOTG_CLK_BASE_ADDR + 0x04))

6.161.1.827 #define USBClkSt (*(volatile unsigned long
    *)(USBOTG_CLK_BASE_ADDR
    + 0x08))

6.161.1.828 #define USBHC_BASE_ADDR 0xFFE0C000

6.161.1.829 #define USBOTG_BASE_ADDR 0xFFE0C100

6.161.1.830 #define USBOTG_CLK_BASE_ADDR 0xFFE0CFF0

6.161.1.831 #define USBOTG_I2C_BASE_ADDR 0xFFE0C300

6.161.1.832 #define USBPortSel (*(volatile unsigned long
    *)(USBOTG_BASE_ADDR +
    0x10))

6.161.1.833 #define VIC_BASE_ADDR 0xFFFFF000

6.161.1.834 #define VICFIQStatus (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0x004))

6.161.1.835 #define VICIntEnable (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0x010))

6.161.1.836 #define VICIntEnClr (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0x014))

6.161.1.837 #define VICIntSelect (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0x00C))

6.161.1.838 #define VICIRQStatus (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0x000))

6.161.1.839 #define VICProtection (*(volatile unsigned long
    *)(VIC_BASE_ADDR +
    0x020))

6.161.1.840 #define VICRawIntr (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0x008))

6.161.1.841 #define VICSoflnt (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0x018))

6.161.1.842 #define VICSoflntClr (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0x01C))

6.161.1.843 #define VICSWPrioMask (*(volatile unsigned long
    *)(VIC_BASE_ADDR +
    0x024))

6.161.1.844 #define VICVectAddr (*(volatile unsigned long
    *)(VIC_BASE_ADDR + 0xF00))

6.161.1.845 #define VICVectAddr0 (*(volatile unsigned long
    *)(VIC_BASE_ADDR +
    0x100))

6.161.1.846 #define VICVectAddr1 (*(volatile unsigned long
    *)(VIC_BASE_ADDR +
    0x104))
```

```
6.161.1.847 #define VICVectAddr10 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x128))  
  
6.161.1.848 #define VICVectAddr11 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x12C))  
  
6.161.1.849 #define VICVectAddr12 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x130))  
  
6.161.1.850 #define VICVectAddr13 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x134))  
  
6.161.1.851 #define VICVectAddr14 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x138))  
  
6.161.1.852 #define VICVectAddr15 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x13C))  
  
6.161.1.853 #define VICVectAddr16 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x140))  
  
6.161.1.854 #define VICVectAddr17 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x144))  
  
6.161.1.855 #define VICVectAddr18 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x148))  
  
6.161.1.856 #define VICVectAddr19 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x14C))  
  
6.161.1.857 #define VICVectAddr20 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x108))  
  
6.161.1.858 #define VICVectAddr21 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x150))  
  
6.161.1.859 #define VICVectAddr22 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x154))  
  
6.161.1.860 #define VICVectAddr23 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x158))  
  
6.161.1.861 #define VICVectAddr24 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x160))
```

```
6.161.1.863 #define VICVectAddr25 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x164))  
  
6.161.1.864 #define VICVectAddr26 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x168))  
  
6.161.1.865 #define VICVectAddr27 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x16C))  
  
6.161.1.866 #define VICVectAddr28 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x170))  
  
6.161.1.867 #define VICVectAddr29 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x174))  
  
6.161.1.868 #define VICVectAddr30 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x10C))  
  
6.161.1.869 #define VICVectAddr31 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x178))  
  
6.161.1.870 #define VICVectAddr32 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x17C))  
  
6.161.1.871 #define VICVectAddr33 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x110))  
  
6.161.1.872 #define VICVectAddr34 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x114))  
  
6.161.1.873 #define VICVectAddr35 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x118))  
  
6.161.1.874 #define VICVectAddr36 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x11C))  
  
6.161.1.875 #define VICVectAddr37 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x120))  
  
6.161.1.876 #define VICVectAddr38 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x124))  
  
6.161.1.877 #define VICVectCntl0 (*(volatile unsigned long *) (VIC_BASE_ADDR + 0x200))  
  
6.161.1.878 #define VICVectCntl1 (*(volatile unsigned long *) (VIC_BASE_ADDR + 0x204))  
  
6.161.1.879 #define VICVectCntl2 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x228))
```

```
6.161.1.880 #define VICVectCntl11 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x22C))  
  
6.161.1.881 #define VICVectCntl12 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x230))  
  
6.161.1.882 #define VICVectCntl13 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x234))  
  
6.161.1.883 #define VICVectCntl14 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x238))  
  
6.161.1.884 #define VICVectCntl15 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x23C))  
  
6.161.1.885 #define VICVectCntl16 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x240))  
  
6.161.1.886 #define VICVectCntl17 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x244))  
  
6.161.1.887 #define VICVectCntl18 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x248))  
  
6.161.1.888 #define VICVectCntl19 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x24C))  
  
6.161.1.889 #define VICVectCntl2 (*(volatile unsigned long *) (VIC_BASE_ADDR + 0x208))  
  
6.161.1.890 #define VICVectCntl20 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x250))  
  
6.161.1.891 #define VICVectCntl21 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x254))  
  
6.161.1.892 #define VICVectCntl22 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x258))  
  
6.161.1.893 #define VICVectCntl23 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x25C))  
  
6.161.1.894 #define VICVectCntl24 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x260))  
  
6.161.1.895 #define VICVectCntl25 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x264))
```

```
6.161.1.896 #define VICVectCntl26 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x268))  
  
6.161.1.897 #define VICVectCntl27 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x26C))  
  
6.161.1.898 #define VICVectCntl28 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x270))  
  
6.161.1.899 #define VICVectCntl29 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x274))  
  
6.161.1.900 #define VICVectCntl3 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x20C))  
  
6.161.1.901 #define VICVectCntl30 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x278))  
  
6.161.1.902 #define VICVectCntl31 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x27C))  
  
6.161.1.903 #define VICVectCntl4 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x210))  
  
6.161.1.904 #define VICVectCntl5 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x214))  
  
6.161.1.905 #define VICVectCntl6 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x218))  
  
6.161.1.906 #define VICVectCntl7 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x21C))  
  
6.161.1.907 #define VICVectCntl8 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x220))  
  
6.161.1.908 #define VICVectCntl9 (*(volatile unsigned long *)(VIC_BASE_ADDR + 0x224))  
  
6.161.1.909 #define VPBDIV (*((volatile unsigned char *) 0xE01FC100))  
  
6.161.1.910 #define WDCLKSEL (*(volatile unsigned long *)(WDG_BASE_ADDR + 0x10))  
  
6.161.1.911 #define WDFEED (*(volatile unsigned long *)(WDG_BASE_ADDR + 0x08))  
  
6.161.1.912 #define WDG_BASE_ADDR 0xE0000000  
  
6.161.1.913 #define WDMOD (*(volatile unsigned long *)(WDG_BASE_ADDR + 0x00))  
  
6.161.1.914 #define WDTC (*(volatile unsigned long *)(WDG_BASE_ADDR + 0x04))  
  
6.161.1.915 #define WDTV (*(volatile unsigned long *)(WDG_BASE_ADDR + 0x0C))
```

6.162 include/canvas.h File Reference

Data Structures

- struct [tCanvasWidget](#)

The structure that describes a canvas widget.

Defines

- `#define CANVAS_STYLE_OUTLINE 0x0001`
This flag indicates that the canvas should be outlined.
- `#define CANVAS_STYLE_FILL 0x0002`
This flag indicates that the canvas should be filled.
- `#define CANVAS_STYLE_TEXT 0x0004`
This flag indicates that the canvas should have text drawn on it.
- `#define CANVAS_STYLE_IMG 0x0008`
This flag indicates that the canvas should have an image drawn on it.
- `#define CANVAS_STYLE_APP_DRAWN 0x0010`
- `#define CANVAS_STYLE_TEXT_OPAQUE 0x0020`
- `#define CANVAS_STYLE_TEXT_LEFT 0x0040`
- `#define CANVAS_STYLE_TEXT_RIGHT 0x0080`
- `#define CANVAS_STYLE_TEXT_TOP 0x0100`
- `#define CANVAS_STYLE_TEXT_BOTTOM 0x0200`
- `#define CANVAS_STYLE_TEXT_HCENTER 0x0000`
- `#define CANVAS_STYLE_TEXT_VCENTER 0x0000`
- `#define CANVAS_STYLE_ALIGN_MASK`
- `#define CANVAS_STYLE_ALIGN_HMASK`
- `#define CANVAS_STYLE_ALIGN_VMASK`
- `#define CanvasStruct(pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, ulStyle, ulFillColor, ulOutlineColor, ulTextColor, pFont, pcText, puImage, pfnOnPaint)`
- `#define Canvas(sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, ulStyle, ulFillColor, ulOutlineColor, ulTextColor, pFont, pcText, puImage, pfnOnPaint)`
- `#define CanvasAppDrawnOff(pWidget)`
- `#define CanvasAppDrawnOn(pWidget)`
- `#define CanvasCallbackSet(pWidget, pfnOnPnt)`
- `#define CanvasFillColorSet(pWidget, ulColor)`
- `#define CanvasFillOff(pWidget)`
- `#define CanvasFillOn(pWidget)`
- `#define CanvasFontSet(pWidget, pFnt)`
- `#define CanvasImageSet(pWidget, plmg)`
- `#define CanvasImageOff(pWidget)`
- `#define CanvasImageOn(pWidget)`
- `#define CanvasOutlineColorSet(pWidget, ulColor)`
- `#define CanvasOutlineOff(pWidget)`
- `#define CanvasOutlineOn(pWidget)`
- `#define CanvasTextColorSet(pWidget, ulColor)`

- #define `CanvasTextOff`(pWidget)
- #define `CanvasTextOn`(pWidget)
- #define `CanvasTextOpaqueOff`(pWidget)
- #define `CanvasTextOpaqueOn`(pWidget)
- #define `CanvasTextAlignment`(pWidget, ulAlign)
- #define `CanvasTextSet`(pWidget, pcTxt)

Functions

- int `CanvasMsgProc` (`tWidget` *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void `CanvasInit` (`tCanvasWidget` *pWidget, const `tDisplay` *pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

6.162.1 Define Documentation

6.162.1.1 #define `Canvas(sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth,`
`IHeight, ulStyle, ulFillColor, ulOutlineColor, ulTextColor, pFont, pcText,`
`pucImage, pfnOnPaint)`

Value:

```
tCanvasWidget sName =
    CanvasStruct(pParent, pNext, pChild, pDisplay, lX, lY, lWidth, \
    lHeight, ulStyle, ulFillColor, ulOutlineColor, \
    ulTextColor, pFont, pcText, pucImage, pfnOnPaint)
```

Declares an initialized variable containing a canvas widget data structure.

Parameters

<code>sName</code>	is the name of the variable to be declared.
<code>pParent</code>	is a pointer to the parent widget.
<code>pNext</code>	is a pointer to the sibling widget.
<code>pChild</code>	is a pointer to the first child widget.
<code>pDisplay</code>	is a pointer to the display on which to draw the canvas.
<code>IX</code>	is the X coordinate of the upper left corner of the canvas.
<code>IY</code>	is the Y coordinate of the upper left corner of the canvas.
<code>IWidth</code>	is the width of the canvas.
<code>IHeight</code>	is the height of the canvas.
<code>ulStyle</code>	is the style to be applied to the canvas.
<code>ulFillColor</code>	is the color used to fill in the canvas.
<code>ulOutlineColor</code>	is the color used to outline the canvas.
<code>ulTextColor</code>	is the color used to draw text on the canvas.
<code>pFont</code>	is a pointer to the font to be used to draw text on the canvas.
<code>pcText</code>	is a pointer to the text to draw on this canvas.
<code>pucImage</code>	is a pointer to the image to draw on this canvas.
<code>pfnOnPaint</code>	is a pointer to the application function to draw onto this canvas.

This macro declares a variable containing an initialized canvas widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

ulStyle is the logical OR of the following:

- **CANVAS_STYLE_OUTLINE** to indicate that the canvas should be outlined.
- **CANVAS_STYLE_FILL** to indicate that the canvas should be filled.
- **CANVAS_STYLE_TEXT** to indicate that the canvas should have text drawn on it (using *pFont* and *pcText*).
- **CANVAS_STYLE_IMG** to indicate that the canvas should have an image drawn on it (using *puImage*).
- **CANVAS_STYLE_APP_DRAWN** to indicate that the canvas should be drawn with the application-supplied drawing function (using *pfnOnPaint*).
- **CANVAS_STYLE_TEXT_OPAQUE** to indicate that the canvas text should be drawn opaque (in other words, drawing the background pixels).
- **CANVAS_STYLE_TEXT_LEFT** to indicate that the canvas text should be left aligned within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_HCENTER** to indicate that the canvas text should be horizontally centered within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_RIGHT** to indicate that the canvas text should be right aligned within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_TOP** to indicate that the canvas text should be top aligned within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_VCENTER** to indicate that the canvas text should be vertically centered within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_BOTTOM** to indicate that the canvas text should be bottom aligned within the widget bounding rectangle.

Returns

Nothing; this is not a function.

6.162.1.2 #define CANVAS_STYLE_ALIGN_HMASK

Value:

```
(CANVAS_STYLE_TEXT_LEFT |      \
           CANVAS_STYLE_TEXT_RIGHT)
```

6.162.1.3 #define CANVAS_STYLE_ALIGN_MASK**Value:**

```
(CANVAS_STYLE_TEXT_LEFT |      \
             CANVAS_STYLE_TEXT_RIGHT | \
             CANVAS_STYLE_TEXT_TOP |   \
             CANVAS_STYLE_TEXT_BOTTOM)
```

6.162.1.4 #define CANVAS_STYLE_ALIGN_VMASK**Value:**

```
(CANVAS_STYLE_TEXT_TOP |      \
             CANVAS_STYLE_TEXT_BOTTOM)
```

6.162.1.5 #define CANVAS_STYLE_APP_DRAWN 0x0010

This flag indicates that the canvas is drawn using the application-supplied drawing function.

6.162.1.6 #define CANVAS_STYLE_FILL 0x0002

This flag indicates that the canvas should be filled.

6.162.1.7 #define CANVAS_STYLE_IMG 0x0008

This flag indicates that the canvas should have an image drawn on it.

6.162.1.8 #define CANVAS_STYLE_OUTLINE 0x0001

This flag indicates that the canvas should be outlined.

6.162.1.9 #define CANVAS_STYLE_TEXT 0x0004

This flag indicates that the canvas should have text drawn on it.

6.162.1.10 #define CANVAS_STYLE_TEXT_BOTTOM 0x0200

This flag indicates that canvas text should be bottom-aligned. By default, text is centered in both X and Y within the canvas bounding rectangle.

6.162.1.11 #define CANVAS_STYLE_TEXT_HCENTER 0x0000

This flag indicates that canvas text should be centered horizontally. By default, text is centered in both X and Y within the canvas bounding rectangle.

6.162.1.12 #define CANVAS_STYLE_TEXT_LEFT 0x0040

This flag indicates that canvas text should be left-aligned. By default, text is centered in both X and Y within the canvas bounding rectangle.

6.162.1.13 #define CANVAS_STYLE_TEXT_OPAQUE 0x0020

This flag indicates that the canvas text should be drawn opaque (in other words, drawing the background pixels as well as the foreground pixels).

6.162.1.14 #define CANVAS_STYLE_TEXT_RIGHT 0x0080

This flag indicates that canvas text should be right-aligned. By default, text is centered in both X and Y within the canvas bounding rectangle.

6.162.1.15 #define CANVAS_STYLE_TEXT_TOP 0x0100

This flag indicates that canvas text should be top-aligned. By default, text is centered in both X and Y within the canvas bounding rectangle.

6.162.1.16 #define CANVAS_STYLE_TEXT_VCENTER 0x0000

This flag indicates that canvas text should be centered vertically. By default, text is centered in both X and Y within the canvas bounding rectangle.

6.162.1.17 #define CanvasAppDrawnOff(pWidget)
Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle &= ~(CANVAS_STYLE_APP_DRAWN); \
}                                \
while(0)
```

Disables application drawing of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function disables the use of the application callback to draw on a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.18 #define CanvasAppDrawnOn(*pWidget*)

Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle |= CANVAS_STYLE_APP_DRAWN; \
}                                \
while(0)
```

Enables application drawing of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function enables the use of the application callback to draw on a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.19 #define CanvasCallbackSet(*pWidget*, *pfnOnPnt*)

Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->pfnOnPaint = pfnOnPnt;   \
}                                \
while(0)
```

Sets the function to call when this canvas widget is drawn.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
<i>pfnOnPnt</i>	is a pointer to the function to call.

This function sets the function to be called when this canvas is drawn and **CANVAS_STYLE_APP_DRAWN** is selected.

Returns

None.

6.162.1.20 #define CanvasFillColorSet(pWidget, ulColor)

Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulFillColor = ulColor;   \
}                                \
while(0)
```

Sets the fill color of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the canvas.

This function changes the color used to fill the canvas on the display. The display is not updated until the next paint request.

Returns

None.

6.162.1.21 #define CanvasFillOff(pWidget)

Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle &= ~(CANVAS_STYLE_FILL); \
}                                \
while(0)
```

Disables filling of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function disables the filling of a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.22 #define CanvasFillOn(*pWidget*)

Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle |= CANVAS_STYLE_FILL; \
}                                \
while(0)
```

Enables filling of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function enables the filling of a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.23 #define CanvasFontSet(*pWidget*, *pFnt*)

Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    const tFont *pF = pFnt;       \
    pW->pFont = pF;             \
}                                \
while(0)
```

Sets the font for a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
<i>pFnt</i>	is a pointer to the font to use to draw text on the canvas.

This function changes the font used to draw text on the canvas. The display is not updated until the next paint request.

Returns

None.

6.162.1.24 #define CanvasImageOff(*pWidget*)

Value:

```
do                                \
{                                 \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle &= ~(CANVAS_STYLE_IMG); \
}                                 \
while(0)
```

Disables the image on a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function disables the drawing of an image on a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.25 #define CanvasImageOn(*pWidget*)

Value:

```
do                                \
{                                 \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle |= CANVAS_STYLE_IMG; \
}                                 \
while(0)
```

Enables the image on a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function enables the drawing of an image on a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.26 #define CanvasImageSet(pWidget, plmg)**Value:**

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    const unsigned char *pI = plmg; \
    pW->pucImage = pI;           \
}                                \
while(0)
```

Changes the image drawn on a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to be modified.
<i>plmg</i>	is a pointer to the image to draw onto the canvas.

This function changes the image that is drawn onto the canvas. The display is not updated until the next paint request.

Returns

None.

6.162.1.27 #define CanvasOutlineColorSet(pWidget, ulColor)**Value:**

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulOutlineColor = ulColor; \
}                                \
while(0)
```

Sets the outline color of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to outline the canvas.

This function changes the color used to outline the canvas on the display. The display is not updated until the next paint request.

Returns

None.

6.162.1.28 #define CanvasOutlineOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle &= ~CANVAS_STYLE_OUTLINE; \
}                                \
while(0)
```

Disables outlining of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function disables the outlining of a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.29 #define CanvasOutlineOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle |= CANVAS_STYLE_OUTLINE; \
}                                \
while(0)
```

Enables outlining of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function enables the outlining of a canvas widget. The display is not updated until the next paint request.

Returns

None.

```
6.162.1.30 #define CanvasStruct( pParent, pNext, pChild, pDisplay, lX, lY, lWidth,
    lHeight, ulStyle, ulFillColor, ulOutlineColor, ulTextColor, pFont, pcText,
    pucImage, pfnOnPaint )
```

Value:

```
{
{
    sizeof(tCanvasWidget),
    (tWidget *) (pParent),
    (tWidget *) (pNext),
    (tWidget *) (pChild),
    pDisplay,
    {
        lX,
        lY,
        (lX) + (lWidth) - 1,
        (lY) + (lHeight) - 1
    },
    CanvasMsgProc
},
ulStyle,
ulFillColor,
ulOutlineColor,
ulTextColor,
pFont,
pcText,
pucImage,
pfnOnPaint
}
```

Declares an initialized canvas widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the canvas.
<i>lX</i>	is the X coordinate of the upper left corner of the canvas.
<i>lY</i>	is the Y coordinate of the upper left corner of the canvas.
<i>lWidth</i>	is the width of the canvas.
<i>lHeight</i>	is the height of the canvas.
<i>ulStyle</i>	is the style to be applied to the canvas.
<i>ulFillColor</i>	is the color used to fill in the canvas.
<i>ulOutline-Color</i>	is the color used to outline the canvas.
<i>ulTextColor</i>	is the color used to draw text on the canvas.
<i>pFont</i>	is a pointer to the font to be used to draw text on the canvas.
<i>pcText</i>	is a pointer to the text to draw on this canvas.

<i>puImage</i>	is a pointer to the image to draw on this canvas.
<i>pfnOnPaint</i>	is a pointer to the application function to draw onto this canvas.

This macro provides an initialized canvas widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
///!      tCanvasWidget g_sCanvas = CanvasStruct(...);
///!
```

Or, in an array of variables:

```
///!      tCanvasWidget g_psCanvas[] =
///!      {
///!          CanvasStruct(...),
///!          CanvasStruct(...)
///!      };
///!
```

ulStyle is the logical OR of the following:

- **CANVAS_STYLE_OUTLINE** to indicate that the canvas should be outlined.
- **CANVAS_STYLE_FILL** to indicate that the canvas should be filled.
- **CANVAS_STYLE_TEXT** to indicate that the canvas should have text drawn on it (using *pFont* and *pcText*).
- **CANVAS_STYLE_IMG** to indicate that the canvas should have an image drawn on it (using *puImage*).
- **CANVAS_STYLE_APP_DRAWN** to indicate that the canvas should be drawn with the application-supplied drawing function (using *pfnOnPaint*).
- **CANVAS_STYLE_TEXT_OPAQUE** to indicate that the canvas text should be drawn opaque (in other words, drawing the background pixels).
- **CANVAS_STYLE_TEXT_LEFT** to indicate that the canvas text should be left aligned within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_HCENTER** to indicate that the canvas text should be horizontally centered within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_RIGHT** to indicate that the canvas text should be right aligned within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_TOP** to indicate that the canvas text should be top aligned within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_VCENTER** to indicate that the canvas text should be vertically centered within the widget bounding rectangle.
- **CANVAS_STYLE_TEXT_BOTTOM** to indicate that the canvas text should be bottom aligned within the widget bounding rectangle.

Returns

Nothing; this is not a function.

6.162.1.31 #define CanvasTextAlignment(pWidget, ulAlign)**Value:**

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;      \
    pW->ulStyle &= ~CANVAS_STYLE_ALIGN_MASK; \
    pW->ulStyle |= ((ulAlign) & CANVAS_STYLE_ALIGN_MASK); \
}                                \
while(0)
```

Sets the text alignment for a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
<i>ulAlign</i>	contains the required text alignment setting. This is a logical OR of style values CANVAS_STYLE_TEXT_LEFT , CANVAS_STYLE_TEXT_RIGHT , CANVAS_STYLE_TEXT_HCENTER , CANVAS_STYLE_TEXT_VCENTER , CANVAS_STYLE_TEXT_TOP and CANVAS_STYLE_TEXT_BOTTOM .

This function sets the alignment of the text drawn inside the widget. Independent alignment options for horizontal and vertical placement allow the text to be positioned in one of 9 positions within the bounding box of the widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.32 #define CanvasTextColorSet(pWidget, ulColor)**Value:**

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;      \
    pW->ulTextColor = ulColor;       \
}                                \
while(0)
```

Sets the text color of a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw text on the canvas. Generated on Wed Oct 24 2012 22:48:50 for GLCD by Doxygen

This function changes the color used to draw text on the canvas on the display. The display is not updated until the next paint request.

Returns

None.

6.162.1.33 #define CanvasTextOff(*pWidget*)

Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle &= ~(CANVAS_STYLE_TEXT); \
}                                \
while(0)
```

Disables the text on a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function disables the drawing of text on a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.34 #define CanvasTextOn(*pWidget*)

Value:

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;   \
    pW->ulStyle |= CANVAS_STYLE_TEXT; \
}                                \
while(0)
```

Enables the text on a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function enables the drawing of text on a canvas widget. The display is not updated until the next paint request.

Returns

None.

6.162.1.35 #define CanvasTextOpaqueOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;      \
    pW->ulStyle &= ~ (CANVAS_STYLE_TEXT_OPAQUE); \
}                                \
while(0)
```

Disables opaque text on a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function disables the use of opaque text on this canvas. When not using opaque text, only the foreground pixels of the text are drawn on the screen, allowing the previously drawn pixels (such as the canvas image) to show through the text.

Returns

None.

6.162.1.36 #define CanvasTextOpaqueOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tCanvasWidget *pW = pWidget;      \
    pW->ulStyle |= CANVAS_STYLE_TEXT_OPAQUE; \
}                                \
while(0)
```

Enables opaque text on a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to modify.
----------------	--

This function enables the use of opaque text on this canvas. When using opaque text, both the foreground and background pixels of the text are drawn on the screen, blocking out the previously drawn pixels.

Returns

None.

6.162.1.37 #define CanvasTextSet(*pWidget*, *pcTxt*)

Value:

```
do                                \
{                                 \
    tCanvasWidget *pW = pWidget; \
    const char *pcT = pcTxt;   \
    pW->pcText = pcT;        \
}                                 \
while(0)
```

Changes the text drawn on a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to be modified.
<i>pcTxt</i>	is a pointer to the text to draw onto the canvas.

This function changes the text that is drawn onto the canvas. The display is not updated until the next paint request.

Returns

None.

6.162.2 Function Documentation

**6.162.2.1 void CanvasInit (*tCanvasWidget* * *pWidget*, *const tDisplay* * *pDisplay*,
unsigned IX, *unsigned IY*, *unsigned IWidth*, *unsigned IHeight*)**

Initializes a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the canvas.
<i>IX</i>	is the X coordinate of the upper left corner of the canvas.
<i>IY</i>	is the Y coordinate of the upper left corner of the canvas.
<i>IWidth</i>	is the width of the canvas.
<i>IHeight</i>	is the height of the canvas.

This function initializes the provided canvas widget.

Returns

None.

6.162.2.2 int CanvasMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a canvas widget.

Parameters

<i>pWidget</i>	is a pointer to the canvas widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this canvas widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.163 include/checkbox.h File Reference

Data Structures

- struct [tCheckBoxWidget](#)

The structure that describes a check box widget.

Defines

- #define [CB_STYLE_OUTLINE](#) 0x0001

This flag indicates that the check box should be outlined.

- #define [CB_STYLE_FILL](#) 0x0002

This flag indicates that the check box should be filled.

- #define [CB_STYLE_TEXT](#) 0x0004

This flag indicates that the check box should have text drawn on it.

- #define [CB_STYLE_IMG](#) 0x0008

This flag indicates that the check box should have an image drawn on it.

- #define [CB_STYLE_TEXT_OPAQUE](#) 0x0010

- #define [CB_STYLE_SELECTED](#) 0x0020

This flag indicates that the check box is selected.

- #define `CheckBoxStruct`(pParent, pNext, pChild, pDisplay, IX, IY, IWidth,IHeight, usStyle, usBoxSize, ulFillColor,ulOutlineColor, uiTextColor, pFont, pcText, pucImage, pfnOnChange)
- #define `CheckBox`(sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth,IHeight, usStyle, usBoxSize, ulFillColor, ulOutlineColor,uiTextColor, pFont, pcText, puclImage, pfnOnChange)
- #define `CheckBoxBoxSizeSet`(pWidget, usSize)
- #define `CheckBoxCallbackSet`(pWidget, pfnOnChg)
- #define `CheckBoxFillColorSet`(pWidget, ulColor)
- #define `CheckBoxFillOff`(pWidget)
- #define `CheckBoxFillOn`(pWidget)
- #define `CheckBoxFontSet`(pWidget, pFnt)
- #define `CheckBoxImageSet`(pWidget, plmg)
- #define `CheckBoxImageOff`(pWidget)
- #define `CheckBoxImageOn`(pWidget)
- #define `CheckBoxOutlineColorSet`(pWidget, ulColor)
- #define `CheckBoxOutlineOff`(pWidget)
- #define `CheckBoxOutlineOn`(pWidget)
- #define `CheckBoxTextColorSet`(pWidget, ulColor)
- #define `CheckBoxTextOff`(pWidget)
- #define `CheckBoxTextOn`(pWidget)
- #define `CheckBoxTextOpaqueOff`(pWidget)
- #define `CheckBoxTextOpaqueOn`(pWidget)
- #define `CheckBoxTextSet`(pWidget, pcTxt)

Functions

- int `CheckBoxMsgProc` (`tWidget` *pWidget, unsigned ulMsg, unsigned uiParam1, unsigned uiParam2)
- void `CheckBoxInit` (`tCheckBoxWidget` *pWidget, const `tDisplay` *pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

6.163.1 Define Documentation

6.163.1.1 #define CB_STYLE_FILL 0x0002

This flag indicates that the check box should be filled.

6.163.1.2 #define CB_STYLE_IMG 0x0008

This flag indicates that the check box should have an image drawn on it.

6.163.1.3 #define CB_STYLE_OUTLINE 0x0001

This flag indicates that the check box should be outlined.

6.163.1.4 #define CB_STYLE_SELECTED 0x0020

This flag indicates that the check box is selected.

6.163.1.5 #define CB_STYLE_TEXT 0x0004

This flag indicates that the check box should have text drawn on it.

6.163.1.6 #define CB_STYLE_TEXT_OPAQUE 0x0010

This flag indicates that the check box text should be drawn opaque (in other words, drawing the background pixels as well as the foreground pixels).

6.163.1.7 #define CheckBox(*sName*, *pParent*, *pNext*, *pChild*, *pDisplay*, *lX*, *lY*, *lWidth*,
lHeight, *usStyle*, *usBoxSize*, *ulFillColor*, *ulOutlineColor*, *ulTextColor*, *pFont*,
pcText, *puclImage*, *pfnOnChange*)**Value:**

```
tCheckBoxWidget sName =
    CheckBoxStruct(pParent, pNext, pChild, pDisplay, lX, lY, lWidth, \
                    lHeight, usStyle, usBoxSize, ulFillColor, \
                    ulOutlineColor, ulTextColor, pFont, pcText, \
                    puclImage, pfnOnChange)
```

Declares an initialized variable containing a check box widget data structure.

Parameters

<i>sName</i>	is the name of the variable to be declared.
<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the check box.
<i>lX</i>	is the X coordinate of the upper left corner of the check box.
<i>lY</i>	is the Y coordinate of the upper left corner of the check box.
<i>lWidth</i>	is the width of the check box.
<i>lHeight</i>	is the height of the check box.
<i>usStyle</i>	is the style to be applied to this check box.
<i>usBoxSize</i>	is the size of the box that is checked.
<i>ulFillColor</i>	is the color used to fill in the check box.
<i>ulOutline-Color</i>	is the color used to outline the check box.
<i>ulTextColor</i>	is the color used to draw text on the check box.
<i>pFont</i>	is a pointer to the font to be used to draw text on the check box.
<i>pcText</i>	is a pointer to the text to draw on this check box.
<i>puclImage</i>	is a pointer to the image to draw on this check box.
<i>pfnOn-Change</i>	is a pointer to the function that is called when the check box is pressed.

This macro provides an initialized check box widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

usStyle is the logical OR of the following:

- **CB_STYLE_OUTLINE** to indicate that the check box should be outlined.
- **CB_STYLE_FILL** to indicate that the check box should be filled.
- **CB_STYLE_TEXT** to indicate that the check box should have text drawn on it (using *pFont* and *pcText*).
- **CB_STYLE_IMG** to indicate that the check box should have an image drawn on it (using *puImage*).
- **CB_STYLE_TEXT_OPAQUE** to indicate that the check box text should be drawn opaque (in other words, drawing the background pixels).
- **CB_STYLE_SELECTED** to indicate that the check box is selected.

Returns

Nothing; this is not a function.

6.163.1.8 #define CheckBoxBoxSizeSet(*pWidget*, *usSize*)

Value:

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget;  \
    pW->usBoxSize = usSize;       \
}                                \
while(0)
```

Sets size of the box to be checked.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
<i>usSize</i>	is the size of the box, in pixels.

This function sets the size of the box that is drawn as part of the check box.

Returns

None.

6.163.1.9 #define CheckBoxCallbackSet(*pWidget*, *pfnOnChg*)**Value:**

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget;   \
    pW->pfnOnChange = pfnOnChg;    \
}                                \
while(0)
```

Sets the function to call when this check box widget is toggled.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
<i>pfnOnChg</i>	is a pointer to the function to call.

This function sets the function to be called when this check box is toggled.

Returns

None.

6.163.1.10 #define CheckBoxFillColorSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget;   \
    pW->ulFillColor = ulColor;    \
}                                \
while(0)
```

Sets the fill color of a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the check box.

This function changes the color used to fill the check box on the display. The display is not updated until the next paint request.

Returns

None.

6.163.1.11 #define CheckBoxFillOff(*pWidget*)

Value:

```
do                                \
{                                 \
    tCheckBoxWidget *pW = pWidget; \
    pW->usStyle &= ~(CB_STYLE_FILL); \
}                                 \
while(0)
```

Disables filling of a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function disables the filling of a check box widget. The display is not updated until the next paint request.

Returns

None.

6.163.1.12 #define CheckBoxFillOn(*pWidget*)

Value:

```
do                                \
{                                 \
    tCheckBoxWidget *pW = pWidget; \
    pW->usStyle |= CB_STYLE_FILL; \
}                                 \
while(0)
```

Enables filling of a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function enables the filling of a check box widget. The display is not updated until the next paint request.

Returns

None.

6.163.1.13 #define CheckBoxFontSet(*pWidget*, *pFnt*)

Value:

```

do                                \
{                                \
    tCheckBoxWidget *pW = pWidget; \
    const tFont *pF = pFnt;      \
    pW->pFont = pF;            \
}                                \
while(0)

```

Sets the font for a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
<i>pFnt</i>	is a pointer to the font to use to draw text on the check box.

This function changes the font used to draw text on the check box. The display is not updated until the next paint request.

Returns

None.

6.163.1.14 #define CheckBoxImageOff(*pWidget*)

Value:

```

do                                \
{                                \
    tCheckBoxWidget *pW = pWidget; \
    pW->usStyle &= ~ (CB_STYLE_IMG); \
}                                \
while(0)

```

Disables the image on a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function disables the drawing of an image on a check box widget. The display is not updated until the next paint request.

Returns

None.

6.163.1.15 #define CheckBoxImageOn(*pWidget*)

Value:

```

do                                \
{                                \

```

```

{
    tCheckBoxWidget *pW = pWidget; \
    pW->usStyle |= CB_STYLE_IMG; \
}
while (0)

```

Enables the image on a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function enables the drawing of an image on a check box widget. The display is not updated until the next paint request.

Returns

None.

6.163.1.16 #define CheckBoxImageSet(*pWidget*, *pImg*)

Value:

```

do \
{
    tCheckBoxWidget *pW = pWidget; \
    const unsigned char *pI = pImg; \
    pW->pucImage = pI; \
}
while (0)

```

Changes the image drawn on a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to be modified.
<i>pImg</i>	is a pointer to the image to draw onto the check box.

This function changes the image that is drawn onto the check box. The display is not updated until the next paint request.

Returns

None.

6.163.1.17 #define CheckBoxOutlineColorSet(*pWidget*, *ulColor*)

Value:

```

do \

```

```

{
    tCheckBoxWidget *pW = pWidget;      \
    pW->ulOutlineColor = ulColor;     \
}
while(0)

```

Sets the outline color of a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to outline the check box.

This function changes the color used to outline the check box on the display. The display is not updated until the next paint request.

Returns

None.

6.163.1.18 #define CheckBoxOutlineOff(*pWidget*)

Value:

```

do                                \
{
    tCheckBoxWidget *pW = pWidget;      \
    pW->usStyle &= ~ (CB_STYLE_OUTLINE); \
}
while(0)

```

Disables outlining of a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function disables the outlining of a check box widget. The display is not updated until the next paint request.

Returns

None.

6.163.1.19 #define CheckBoxOutlineOn(*pWidget*)

Value:

```

do                                \
{

```

```

    tCheckBoxWidget *pW = pWidget; \
    pW->usStyle |= CB_STYLE_OUTLINE; \
}
while(0)

```

Enables outlining of a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function enables the outlining of a check box widget. The display is not updated until the next paint request.

Returns

None.

6.163.1.20 #define CheckBoxStruct(pParent, pNext, pChild, pDisplay, lX, lY, lWidth, lHeight, usStyle, usBoxSize, ulFillColor, ulOutlineColor, ulTextColor, pFont, pcText, pucImage, pfOnChange)

Value:

```

{
    sizeof(tCheckBoxWidget), \
    (tWidget *) (pParent), \
    (tWidget *) (pNext), \
    (tWidget *) (pChild), \
    pDisplay, \
    { \
        lX, \
        lY, \
        (lX) + (lWidth) - 1, \
        (lY) + (lHeight) - 1 \
    }, \
    CheckBoxMsgProc \
}, \
usStyle, \
usBoxSize, \
ulFillColor, \
ulOutlineColor, \
ulTextColor, \
pFont, \
pcText, \
pucImage, \
pfOnChange \
}

```

Declares an initialized check box widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the check box.
<i>IX</i>	is the X coordinate of the upper left corner of the check box.
<i>IY</i>	is the Y coordinate of the upper left corner of the check box.
<i>IWidth</i>	is the width of the check box.
<i>IHeight</i>	is the height of the check box.
<i>usStyle</i>	is the style to be applied to this check box.
<i>usBoxSize</i>	is the size of the box that is checked.
<i>ulFillColor</i>	is the color used to fill in the check box.
<i>ulOutline-Color</i>	is the color used to outline the check box.
<i>ulTextColor</i>	is the color used to draw text on the check box.
<i>pFont</i>	is a pointer to the font to be used to draw text on the check box.
<i>pcText</i>	is a pointer to the text to draw on this check box.
<i>puImage</i>	is a pointer to the image to draw on this check box.
<i>pfnOn-Change</i>	is a pointer to the function that is called when the check box is pressed.

This macro provides an initialized check box widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
///!      tCheckBoxWidget g_sCheckBox = CheckBoxStruct(...);
///!
```

Or, in an array of variables:

```
///!      tCheckBoxWidget g_psCheckboxes[] =
///!      {
///!          CheckBoxStruct(...),
///!          CheckBoxStruct(...)
///!      };
///!
```

usStyle is the logical OR of the following:

- **CB_STYLE_OUTLINE** to indicate that the check box should be outlined.
- **CB_STYLE_FILL** to indicate that the check box should be filled.
- **CB_STYLE_TEXT** to indicate that the check box should have text drawn on it (using *pFont* and *pcText*).
- **CB_STYLE_IMG** to indicate that the check box should have an image drawn on it (using *puImage*).
- **CB_STYLE_TEXT_OPAQUE** to indicate that the check box text should be drawn opaque (in other words, drawing the background pixels).

- **CB_STYLE_SELECTED** to indicate that the check box is selected.

Returns

Nothing; this is not a function.

6.163.1.21 #define CheckBoxTextColorSet(pWidget, ulColor)**Value:**

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget;   \
    pW->ulTextColor = ulColor;     \
}                                \
while(0)
```

Sets the text color of a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw text on the check box.

This function changes the color used to draw text on the check box on the display. The display is not updated until the next paint request.

Returns

None.

6.163.1.22 #define CheckBoxTextOff(pWidget)**Value:**

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget;   \
    pW->usStyle &= ~(CB_STYLE_TEXT); \
}                                \
while(0)
```

Disables the text on a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function disables the drawing of text on a check box widget. The display is not updated until the next paint request.

Returns

None.

6.163.1.23 #define CheckBoxTextOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget; \
    pW->usStyle |= CB_STYLE_TEXT; \
}                                \
while(0)
```

Enables the text on a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function enables the drawing of text on a check box widget. The display is not updated until the next paint request.

Returns

None.

6.163.1.24 #define CheckBoxTextOpaqueOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget; \
    pW->usStyle &= ~ (CB_STYLE_TEXT_OPAQUE); \
}                                \
while(0)
```

Disables opaque text on a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function disables the use of opaque text on this check box. When not using opaque text, only the foreground pixels of the text are drawn on the screen, allowing the previ-

ously drawn pixels (such as the check box image) to show through the text.

Returns

None.

6.163.1.25 #define CheckBoxTextOpaqueOn(*pWidget*)

Value:

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget; \
    pW->usStyle |= CB_STYLE_TEXT_OPAQUE; \
}                                \
while(0)
```

Enables opaque text on a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to modify.
----------------	---

This function enables the use of opaque text on this check box. When using opaque text, both the foreground and background pixels of the text are drawn on the screen, blocking out the previously drawn pixels.

Returns

None.

6.163.1.26 #define CheckBoxTextSet(*pWidget*, *pcTxt*)

Value:

```
do                                \
{                                \
    tCheckBoxWidget *pW = pWidget; \
    const char *pcT = pcTxt;     \
    pW->pcText = pcT;          \
}                                \
while(0)
```

Changes the text drawn on a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to be modified.
<i>pcTxt</i>	is a pointer to the text to draw onto the check box.

This function changes the text that is drawn onto the check box. The display is not updated until the next paint request.

Returns

None.

6.163.2 Function Documentation

6.163.2.1 void CheckBoxInit (tCheckBoxWidget * pWidget, const tDisplay * pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

Initializes a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the check box.
<i>IX</i>	is the X coordinate of the upper left corner of the check box.
<i>IY</i>	is the Y coordinate of the upper left corner of the check box.
<i>IWidth</i>	is the width of the check box.
<i>IHeight</i>	is the height of the check box.

This function initializes the provided check box widget.

Returns

None.

6.163.2.2 int CheckBoxMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a check box widget.

Parameters

<i>pWidget</i>	is a pointer to the check box widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this check box widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.164 include/container.h File Reference

Data Structures

- struct [tContainerWidget](#)

The structure that describes a container widget.

Defines

- #define [CTR_STYLE_OUTLINE](#) 0x0001
This flag indicates that the container widget should be outlined.
- #define [CTR_STYLE_FILL](#) 0x0002
This flag indicates that the container widget should be filled.
- #define [CTR_STYLE_TEXT](#) 0x0004
This flag indicates that the container widget should have text drawn on it.
- #define [CTR_STYLE_TEXT_OPAQUE](#) 0x0008
- #define [CTR_STYLE_TEXT_CENTER](#) 0x0010
- #define [ContainerStruct](#)(pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, ulStyle, ulFillColor, ulOutlineColor, ulTextColor, pFont, pcText)
- #define [Container](#)(sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, ulStyle, ulFillColor, ulOutlineColor, ulTextColor, pFont, pcText)
- #define [ContainerFillColorSet](#)(pWidget, ulColor)
- #define [ContainerFillOff](#)(pWidget)
- #define [ContainerFillOn](#)(pWidget)
- #define [ContainerFontSet](#)(pWidget, pFnt)
- #define [ContainerOutlineColorSet](#)(pWidget, ulColor)
- #define [ContainerOutlineOff](#)(pWidget)
- #define [ContainerOutlineOn](#)(pWidget)
- #define [ContainerTextCenterOff](#)(pWidget)
- #define [ContainerTextCenterOn](#)(pWidget)
- #define [ContainerTextColorSet](#)(pWidget, ulColor)
- #define [ContainerTextOff](#)(pWidget)
- #define [ContainerTextOn](#)(pWidget)
- #define [ContainerTextOpaqueOff](#)(pWidget)
- #define [ContainerTextOpaqueOn](#)(pWidget)
- #define [ContainerTextSet](#)(pWidget, pcTxt)

Functions

- int `ContainerMsgProc` (`tWidget *pWidget`, `unsigned ulMsg`, `unsigned ulParam1`, `unsigned ulParam2`)
- void `ContainerInit` (`tContainerWidget *pWidget`, `const tDisplay *pDisplay`, `unsigned IX`, `unsigned IY`, `unsigned IWidth`, `unsigned IHeight`)

6.164.1 Define Documentation

```
#define Container( sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth,
    IHeight, ulStyle, ulFillColor, ulOutlineColor, ulTextColor, pFont, pcText )
```

Value:

```
tContainerWidget sName =
    ContainerStruct(pParent, pNext, pChild, pDisplay, lX, lY, lWidth, \
                    lHeight, ulStyle, ulFillColor, ulOutlineColor, \
                    ultTextColor, pFont, pcText)
```

Declares an initialized variable containing a container widget data structure.

Parameters

<code>sName</code>	is the name of the variable to be declared.
<code>pParent</code>	is a pointer to the parent widget.
<code>pNext</code>	is a pointer to the sibling widget.
<code>pChild</code>	is a pointer to the first child widget.
<code>pDisplay</code>	is a pointer to the display on which to draw the container widget.
<code>IX</code>	is the X coordinate of the upper left corner of the container widget.
<code>IY</code>	is the Y coordinate of the upper left corner of the container widget.
<code>IWidth</code>	is the width of the container widget.
<code>IHeight</code>	is the height of the container widget.
<code>ulStyle</code>	is the style to be applied to the container widget.
<code>ulFillColor</code>	is the color used to fill in the container widget.
<code>ulOutline-Color</code>	is the color used to outline the container widget.
<code>ulTextColor</code>	is the color used to draw text on the container widget.
<code>pFont</code>	is a pointer to the font to be used to draw text on the container widget.
<code>pcText</code>	is a pointer to the text to draw on the container widget.

This macro provides an initialized container widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

`ulStyle` is the logical OR of the following:

- `CTR_STYLE_OUTLINE` to indicate that the container widget should be outlined.
- `CTR_STYLE_FILL` to indicate that the container widget should be filled.

- **CTR_STYLE_TEXT** to indicate that the container widget should have text drawn on it (using *pFont* and *pcText*).
- **CTR_STYLE_TEXT_OPAQUE** to indicate that the container widget text should be drawn opaque (in other words, drawing the background pixels).
- **CTR_STYLE_TEXT_CENTER** to indicate that the container widget text should be drawn centered horizontally.

Returns

Nothing; this is not a function.

6.164.1.2 #define ContainerFillColorSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                 \
    tContainerWidget *pW = pWidget;   \
    pW->ulFillColor = ulColor;      \
}                                 \
while(0)
```

Sets the fill color of a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the container widget.

This function changes the color used to fill the container widget on the display. The display is not updated until the next paint request.

Returns

None.

6.164.1.3 #define ContainerFillOff(*pWidget*)**Value:**

```
do                                \
{                                 \
    tContainerWidget *pW = pWidget;   \
    pW->ulStyle &= ~CTR_STYLE_FILL;  \
}                                 \
while(0)
```

Disables filling of a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function disables the filling of a container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.4 #define ContainerFillOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tContainerWidget *pW = pWidget; \
    pW->ulStyle |= CTR_STYLE_FILL; \
}                                \
while(0)
```

Enables filling of a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function enables the filling of a container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.5 #define ContainerFontSet(*pWidget*, *pFnt*)**Value:**

```
do                                \
{                                \
    tContainerWidget *pW = pWidget; \
    const tFont *pF = pFnt;        \
    pW->pFont = pF;              \
}                                \
while(0)
```

Sets the font for a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
<i>pFnt</i>	is a pointer to the font to use to draw text on the container widget.

Generated on Wed Oct 24 2012 22:40:50 for GLCD by Doxygen

This function changes the font used to draw text on the container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.6 #define ContainerOutlineColorSet(*pWidget*, *ulColor*)

Value:

```
do                                \
{                                \
    tContainerWidget *pW = pWidget;   \
    pW->ulOutlineColor = ulColor;    \
}                                \
while(0)
```

Sets the outline color of a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to outline the container widget.

This function changes the color used to outline the container widget on the display. The display is not updated until the next paint request.

Returns

None.

6.164.1.7 #define ContainerOutlineOff(*pWidget*)

Value:

```
do                                \
{                                \
    tContainerWidget *pW = pWidget;   \
    pW->ulStyle &= ~(CTR_STYLE_OUTLINE); \
}                                \
while(0)
```

Disables outlining of a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function disables the outlining of a container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.8 #define ContainerOutlineOn(*pWidget*)

Value:

```
do                                \
{                                \
    tContainerWidget *pW = pWidget; \
    pW->ulStyle |= CTR_STYLE_OUTLINE; \
}                                \
while(0)
```

Enables outlining of a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function enables the outlining of a container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.9 #define ContainerStruct(*pParent*, *pNext*, *pChild*, *pDisplay*, *lX*, *lY*, *lWidth*, *lHeight*, *ulStyle*, *ulFillColor*, *ulOutlineColor*, *ulTextColor*, *pFont*, *pcText*)

Value:

```
{                                \
{                                \
    sizeof(tContainerWidget), \
    (tWidget *) (pParent), \
    (tWidget *) (pNext), \
    (tWidget *) (pChild), \
    pDisplay, \
    { \
        lX, \
        lY, \
        (lX) + (lWidth) - 1, \
        (lY) + (lHeight) - 1 \
    }, \
    ContainerMsgProc \
}, \
ulStyle,
```

```

        ulFillColor,
        ulOutlineColor,
        ulTextColor,
        pFont,
        pcText
    }

```

Declares an initialized container widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the container widget.
<i>IX</i>	is the X coordinate of the upper left corner of the container widget.
<i>IY</i>	is the Y coordinate of the upper left corner of the container widget.
<i>IWidth</i>	is the width of the container widget.
<i>IHeight</i>	is the height of the container widget.
<i>ulStyle</i>	is the style to be applied to the container widget.
<i>ulFillColor</i>	is the color used to fill in the container widget.
<i>ulOutline-Color</i>	is the color used to outline the container widget.
<i>ulTextColor</i>	is the color used to draw text on the container widget.
<i>pFont</i>	is a pointer to the font to be used to draw text on the container widget.
<i>pcText</i>	is a poitner to the text to draw on the container widget.

This macro provides an initialized container widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
//!     tContainerWidget g_sContainer = ContainerStruct(...);
//!
```

Or, in an array of variables:

```
//!     tContainerWidget g_psContainers[] =
//!     {
//!         ContainerStruct(...),
//!         ContainerStruct(...)
//!     };
//!
```

ulStyle is the logical OR of the following:

- **CTR_STYLE_OUTLINE** to indicate that the container widget should be outlined.
- **CTR_STYLE_FILL** to indicate that the container widget should be filled.
- **CTR_STYLE_TEXT** to indicate that the container widget should have text drawn on it (using *pFont* and *pcText*).

- **CTR_STYLE_TEXT_OPAQUE** to indicate that the container widget text should be drawn opaque (in other words, drawing the background pixels).
- **CTR_STYLE_TEXT_CENTER** to indicate that the container widget text should be drawn centered horizontally.

Returns

Nothing; this is not a function.

6.164.1.10 #define ContainerTextCenterOff(pWidget)**Value:**

```
do                                \
{                                \
    tContainerWidget *pW = pWidget; \
    pW->ulStyle &= ~(CTR_STYLE_TEXT_CENTER); \
}                                \
while(0)
```

Disables the centering of text on a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function disables the centering of text on a container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.11 #define ContainerTextCenterOn(pWidget)**Value:**

```
do                                \
{                                \
    tContainerWidget *pW = pWidget; \
    pW->ulStyle |= CTR_STYLE_TEXT_CENTER; \
}                                \
while(0)
```

Enables the centering of text on a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function enables the centering of text on a container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.12 #define ContainerTextColorSet(*pWidget*, *ulColor*)

Value:

```
do                                \
{                                \
    tContainerWidget *pW = pWidget;   \
    pW->ulTextColor = ulColor;       \
}                                \
while(0)
```

Sets the text color of a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw text on the container widget.

This function changes the color used to draw text on the container widget on the display. The display is not updated until the next paint request.

Returns

None.

6.164.1.13 #define ContainerTextOff(*pWidget*)

Value:

```
do                                \
{                                \
    tContainerWidget *pW = pWidget;   \
    pW->ulStyle &= ~(CTR_STYLE_TEXT); \
}                                \
while(0)
```

Disables the text on a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function disables the drawing of text on a container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.14 #define ContainerTextOn(*pWidget*)**Value:**

```
do                                \
{                               \
    tContainerWidget *pW = pWidget; \
    pW->ulStyle |= CTR_STYLE_TEXT; \
}                               \
while(0)
```

Enables the text on a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function enables the drawing of text on a container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.15 #define ContainerTextOpaqueOff(*pWidget*)**Value:**

```
do                                \
{                               \
    tContainerWidget *pW = pWidget; \
    pW->ulStyle &= ~ (CTR_STYLE_TEXT_OPAQUE); \
}                               \
while(0)
```

Disables opaque text on a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function disables the use of opaque text on this container widget. When not using opaque text, only the foreground pixels of the text are drawn on the screen, allowing the

previously drawn pixels (such as the background) to show through the text.

Returns

None.

6.164.1.16 #define ContainerTextOpaqueOn(*pWidget*)

Value:

```
do                                \
{                                 \
    tContainerWidget *pW = pWidget; \
    pW->ulStyle |= CTR_STYLE_TEXT_OPAQUE; \
}                                 \
while(0)
```

Enables opaque text on a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to modify.
----------------	---

This function enables the use of opaque text on this container widget. When using opaque text, both the foreground and background pixels of the text are drawn on the screen, blocking out the previously drawn pixels.

Returns

None.

6.164.1.17 #define ContainerTextSet(*pWidget*, *pcTxt*)

Value:

```
do                                \
{                                 \
    tContainerWidget *pW = pWidget; \
    const char *pcT = pcTxt;       \
    pW->pcText = pcT;            \
}                                 \
while(0)
```

Changes the text drawn on a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to be modified.
<i>pcTxt</i>	is a pointer to the text to draw onto the container widget.

This function changes the text that is drawn onto the container widget. The display is not updated until the next paint request.

Returns

None.

6.164.1.18 #define CTR_STYLE_FILL 0x0002

This flag indicates that the container widget should be filled.

6.164.1.19 #define CTR_STYLE_OUTLINE 0x0001

This flag indicates that the container widget should be outlined.

6.164.1.20 #define CTR_STYLE_TEXT 0x0004

This flag indicates that the container widget should have text drawn on it.

6.164.1.21 #define CTR_STYLE_TEXT_CENTER 0x0010

This flag indicates that the container text should be drawn centered within the width of the container.

6.164.1.22 #define CTR_STYLE_TEXT_OPAQUE 0x0008

This flag indicates that the container text should be drawn opaque (in other words, drawing the background pixels as well as the foreground pixels).

6.164.2 Function Documentation

6.164.2.1 void ContainerInit(tContainerWidget * pWidget, const tDisplay * pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

Initializes a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the container widget.
<i>IX</i>	is the X coordinate of the upper left corner of the container widget.
<i>IY</i>	is the Y coordinate of the upper left corner of the container widget.
<i>IWidth</i>	is the width of the container widget.
<i>IHeight</i>	is the height of the container widget.

This function initializes a container widget, preparing it for placement into the widget tree.

Returns

none.

6.164.2.2 int ContainerMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a container widget.

Parameters

<i>pWidget</i>	is a pointer to the container widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this container widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.165 include/graphic.h File Reference

Functions

- void [GLCD_Rectangle](#) (unsigned int x, unsigned int y, unsigned int b, unsigned int a)
- void [GLCD_Circle](#) (unsigned int cx, unsigned int cy, unsigned int radius)
- void [GLCD_Line](#) (int X1, int Y1, int X2, int Y2)

6.165.1 Function Documentation

6.165.1.1 void GLCD_Circle (unsigned int cx, unsigned int cy, unsigned int radius)

6.165.1.2 void GLCD_Line (int X1, int Y1, int X2, int Y2)

6.165.1.3 void GLCD_Rectangle (unsigned int x, unsigned int y, unsigned int b, unsigned int a)

6.166 include/grlib.h File Reference

Data Structures

- struct [tRectangle](#)
- struct [tDisplay](#)

This structure defines the characteristics of a display driver.
- struct [tFont](#)

This structure describes a font used for drawing text onto the screen.
- struct [tContext](#)

Defines

- #define [FONT_FMT_UNCOMPRESSED](#) 0x00

Indicates that the font data is stored in an uncompressed format.
- #define [FONT_FMT_PIXEL_RLE](#) 0x01

Indicates that the font data is stored using a pixel-based RLE format.
- #define [IMAGE_FMT_1BPP_UNCOMP](#) 0x01
- #define [IMAGE_FMT_4BPP_UNCOMP](#) 0x04
- #define [IMAGE_FMT_8BPP_UNCOMP](#) 0x08
- #define [IMAGE_FMT_16BPP_UNCOMP](#) 0x10
- #define [IMAGE_FMT_1BPP_COMP](#) 0x81
- #define [IMAGE_FMT_4BPP_COMP](#) 0x84
- #define [IMAGE_FMT_8BPP_COMP](#) 0x88
- #define [GrContextBackgroundSet](#)(pContext, ulValue)
- #define [GrContextBackgroundSetTranslated](#)(pContext, ulValue)
- #define [GrContextDpyWidthGet](#)(pContext) ([DpyWidthGet](#)((pContext)->p-Display))
- #define [GrContextDpyHeightGet](#)(pContext) ([DpyHeightGet](#)((pContext)->p-Display))
- #define [GrContextFontSet](#)(pContext, pFnt)
- #define [GrContextForegroundSet](#)(pContext, ulValue)
- #define [GrContextForegroundSetTranslated](#)(pContext, ulValue)
- #define [GrFlush](#)(pContext)
- #define [GrFontBaselineGet](#)(pFont) ((pFont)->ucBaseline)
- #define [GrFontHeightGet](#)(pFont) ((pFont)->ucHeight)
- #define [GrFontMaxWidthGet](#)(pFont) ((pFont)->ucMaxWidth)
- #define [GrImageColorsGet](#)(puclImage) (((unsigned char *)puclImage)[5] + 1)
- #define [GrImageHeightGet](#)(puclImage) (*(unsigned short *)puclImage + 3))
- #define [GrImageWidthGet](#)(puclImage) (*(unsigned short *)puclImage + 1))
- #define [GrOffScreen1BPPSize](#)(lWidth, lHeight) (5 + ((lWidth + 7) / 8) * lHeight))
- #define [GrOffScreen4BPPSize](#)(lWidth, lHeight) (6 + (16 * 3) + ((lWidth + 1) / 2) * lHeight))
- #define [GrOffScreen8BPPSize](#)(lWidth, lHeight) (6 + (256 * 3) + (lWidth * lHeight))
- #define [GrPixelDraw](#)(pContext, IX, IY)

- #define `GrStringBaselineGet(pContext)` ((`pContext`)>`pFont`->`ucBaseline`)
- #define `GrStringDrawCentered(pContext, pcString, lLength, IX, IY, bOpaque)`
- #define `GrStringHeightGet(pContext)` ((`pContext`)>`pFont`->`ucHeight`)
- #define `GrStringMaxWidthGet(pContext)` ((`pContext`)>`pFont`->`ucMaxWidth`)
- #define `ClrAliceBlue` 0x00F0F8FF
- #define `ClrAntiqueWhite` 0x00FAEBD7
- #define `ClrAqua` 0x0000FFFF
- #define `ClrAquamarine` 0x007FFFFD4
- #define `ClrAzure` 0x00F0FFFF
- #define `ClrBeige` 0x00F5F5DC
- #define `ClrBisque` 0x00FFE4C4
- #define `ClrBlack` 0x00000000
- #define `ClrBlanchedAlmond` 0x00FFEBBC
- #define `ClrBlue` 0x000000FF
- #define `ClrBlueViolet` 0x008A2BE2
- #define `ClrBrown` 0x00A52A2A
- #define `ClrBurlyWood` 0x00DEB887
- #define `ClrCadetBlue` 0x005F9EA0
- #define `ClrChartreuse` 0x007FFF00
- #define `ClrChocolate` 0x00D2691E
- #define `ClrCoral` 0x00FF7F50
- #define `ClrCornflowerBlue` 0x006495ED
- #define `ClrCornsilk` 0x00FFF8DC
- #define `ClrCrimson` 0x00DC143C
- #define `ClrCyan` 0x0000FFFF
- #define `ClrDarkBlue` 0x0000008B
- #define `ClrDarkCyan` 0x00008B8B
- #define `ClrDarkGoldenrod` 0x00B8860B
- #define `ClrDarkGray` 0x00A9A9A9
- #define `ClrDarkGreen` 0x00006400
- #define `ClrDarkKhaki` 0x00BDB76B
- #define `ClrDarkMagenta` 0x008B008B
- #define `ClrDarkOliveGreen` 0x00556B2F
- #define `ClrDarkOrange` 0x00FF8C00
- #define `ClrDarkOrchid` 0x009932CC
- #define `ClrDarkRed` 0x008B0000
- #define `ClrDarkSalmon` 0x00E9967A
- #define `ClrDarkSeaGreen` 0x008FBC8F
- #define `ClrDarkSlateBlue` 0x00483D8B
- #define `ClrDarkSlateGray` 0x002F4F4F
- #define `ClrDarkTurquoise` 0x0000CED1
- #define `ClrDarkViolet` 0x009400D3
- #define `ClrDeepPink` 0x00FF1493
- #define `ClrDeepSkyBlue` 0x0000BFFF
- #define `ClrDimGray` 0x00696969
- #define `ClrDodgerBlue` 0x001E90FF

- #define ClrFireBrick 0x00B22222
- #define ClrFloralWhite 0x00FFFAF0
- #define ClrForestGreen 0x00228B22
- #define ClrFuchsia 0x00FF00FF
- #define ClrGainsboro 0x00DCDCDC
- #define ClrGhostWhite 0x00F8F8FF
- #define ClrGold 0x00FFD700
- #define ClrGoldenrod 0x00DAA520
- #define ClrGray 0x00808080
- #define ClrGreen 0x00008000
- #define ClrGreenYellow 0x00ADFF2F
- #define ClrHoneydew 0x00F0FFF0
- #define ClrHotPink 0x00FF69B4
- #define ClrIndianRed 0x00CD5C5C
- #define ClrIndigo 0x004B0082
- #define ClrIvory 0x00FFFFFF
- #define ClrKhaki 0x00F0E68C
- #define ClrLavender 0x00E6E6FA
- #define ClrLavenderBlush 0x00FFF0F5
- #define ClrLawnGreen 0x007CFC00
- #define ClrLemonChiffon 0x00FFFACD
- #define ClrLightBlue 0x00ADD8E6
- #define ClrLightCoral 0x00F08080
- #define ClrLightCyan 0x00E0FFFF
- #define ClrLightGoldenrodYellow 0x00FAFAD2
- #define ClrLightGreen 0x0090EE90
- #define ClrLightGrey 0x00D3D3D3
- #define ClrLightPink 0x00FFB6C1
- #define ClrLightSalmon 0x00FFA07A
- #define ClrLightSeaGreen 0x0020B2AA
- #define ClrLightSkyBlue 0x0087CEFA
- #define ClrLightSlateGray 0x00778899
- #define ClrLightSteelBlue 0x00B0C4DE
- #define ClrLightYellow 0x00FFFFE0
- #define ClrLime 0x0000FF00
- #define ClrLimeGreen 0x0032CD32
- #define ClrLinen 0x00FAF0E6
- #define ClrMagenta 0x00FF00FF
- #define ClrMaroon 0x00800000
- #define ClrMediumAquamarine 0x0066CDAA
- #define ClrMediumBlue 0x000000CD
- #define ClrMediumOrchid 0x00BA55D3
- #define ClrMediumPurple 0x009370DB
- #define ClrMediumSeaGreen 0x003CB371
- #define ClrMediumSlateBlue 0x007B68EE
- #define ClrMediumSpringGreen 0x0000FA9A

- #define ClrMediumTurquoise 0x0048D1CC
- #define ClrMediumVioletRed 0x00C71585
- #define ClrMidnightBlue 0x00191970
- #define ClrMintCream 0x00F5FFFF
- #define ClrMistyRose 0x00FFE4E1
- #define ClrMoccasin 0x00FFE4B5
- #define ClrNavajoWhite 0x00FFDEAD
- #define ClrNavy 0x00000080
- #define ClrOldLace 0x00FDF5E6
- #define ClrOlive 0x00808000
- #define ClrOliveDrab 0x006B8E23
- #define ClrOrange 0x00FFA500
- #define ClrOrangeRed 0x00FF4500
- #define ClrOrchid 0x00DA70D6
- #define ClrPaleGoldenrod 0x00EEE8AA
- #define ClrPaleGreen 0x0098FB98
- #define ClrPaleTurquoise 0x00AFEEEE
- #define ClrPaleVioletRed 0x00DB7093
- #define ClrPapayaWhip 0x00FFEF5
- #define ClrPeachPuff 0x00FFDAB9
- #define ClrPeru 0x00CD853F
- #define ClrPink 0x00FFC0CB
- #define ClrPlum 0x00DDA0DD
- #define ClrPowderBlue 0x00B0E0E6
- #define ClrPurple 0x00800080
- #define ClrRed 0x00FF0000
- #define ClrRosyBrown 0x00BC8F8F
- #define ClrRoyalBlue 0x004169E1
- #define ClrSaddleBrown 0x008B4513
- #define ClrSalmon 0x00FA8072
- #define ClrSandyBrown 0x00F4A460
- #define ClrSeaGreen 0x002E8B57
- #define ClrSeashell 0x00FFF5EE
- #define ClrSienna 0x00A0522D
- #define ClrSilver 0x00C0C0C0
- #define ClrSkyBlue 0x0087CEEB
- #define ClrSlateBlue 0x006A5ACD
- #define ClrSlateGray 0x00708090
- #define ClrSnow 0x00FFFAFA
- #define ClrSpringGreen 0x0000FF7F
- #define ClrSteelBlue 0x004682B4
- #define ClrTan 0x00D2B48C
- #define ClrTeal 0x00008080
- #define ClrThistle 0x00D8BFD8
- #define ClrTomato 0x00FF6347
- #define ClrTurquoise 0x0040E0D0

- #define **ClrViolet** 0x00EE82EE
- #define **ClrWheat** 0x00F5DEB3
- #define **ClrWhite** 0x00FFFFFF
- #define **ClrWhiteSmoke** 0x00F5F5F5
- #define **ClrYellow** 0x00FFFF00
- #define **ClrYellowGreen** 0x009ACD32
- #define **ClrRedMask** 0x00FF0000
- #define **ClrRedShift** 16
- #define **ClrGreenMask** 0x0000FF00
- #define **ClrGreenShift** 8
- #define **ClrBlueMask** 0x000000FF
- #define **ClrBlueShift** 0
- #define **GrLangZhPRC** 0x0804
- #define **GrLangZhTW** 0x0404
- #define **GrLangEnUS** 0x0409
- #define **GrLangEnUK** 0x0809
- #define **GrLangEnAUS** 0x0C09
- #define **GrLangEnCA** 0x1009
- #define **GrLangEnNZ** 0x1409
- #define **GrLangFr** 0x040C
- #define **GrLangDe** 0x0407
- #define **GrLangHi** 0x0439
- #define **GrLangIt** 0x0410
- #define **GrLangJp** 0x0411
- #define **GrLangKo** 0x0412
- #define **GrLangEsMX** 0x080A
- #define **GrLangEsSP** 0x0C0A
- #define **GrLangSwKE** 0x0441
- #define **GrLangUrIN** 0x0820
- #define **GrLangUrPK** 0x0420
- #define **DpyColorTranslate**(pDisplay, ulValue) ((pDisplay)->pfnColorTranslate((pDisplay)->pvDisplayData, ulValue))
- #define **DpyFlush**(pDisplay)
- #define **DpyHeightGet**(pDisplay) ((pDisplay)->usHeight)
- #define **DpyLineDrawH**(pDisplay, IX1, IX2, IY, ulValue)
- #define **DpyLineDrawV**(pDisplay, IX, IY1, IY2, ulValue)
- #define **DpyPixelDraw**(pDisplay, IX, IY, ulValue)
- #define **DpyPixelDrawMultiple**(pDisplay, IX, IY, IX0, ICount, IBPP, pucData,pucPalette)
- #define **DpyRectFill**(pDisplay, pRect, ulValue)
- #define **DpyWidthGet**(pDisplay) ((pDisplay)->usWidth)
- #define **GrRectContainsPoint**(pRect, IX, IY)

Functions

- void `GrCircleDraw` (const `tContext` *pContext, unsigned IX, unsigned IY, unsigned IRadius)
- void `GrCircleFill` (const `tContext` *pContext, unsigned IX, unsigned IY, unsigned IRadius)
- void `GrContextClipRegionSet` (`tContext` *pContext, `tRectangle` *pRect)
- void `GrContextInit` (`tContext` *pContext, const `tDisplay` *pDisplay)
- void `GrlImageDraw` (const `tContext` *pContext, const unsigned char *puImage, unsigned IX, unsigned IY)
- void `GrLineDraw` (const `tContext` *pContext, unsigned IX1, unsigned IY1, unsigned IX2, unsigned IY2)
- void `GrLineDrawH` (const `tContext` *pContext, unsigned IX1, unsigned IX2, unsigned IY)
- void `GrLineDrawV` (const `tContext` *pContext, unsigned IX, unsigned IY1, unsigned IY2)
- void `GrOffScreen1BPPInit` (`tDisplay` *pDisplay, unsigned char *puImage, long IWidth, long IHeight)
- void `GrOffScreen4BPPInit` (`tDisplay` *pDisplay, unsigned char *puImage, long IWidth, long IHeight)
- void `GrOffScreen4BPPPaletteSet` (`tDisplay` *pDisplay, unsigned long *pulPalette, unsigned long ulOffset, unsigned long ulCount)
- void `GrOffScreen8BPPInit` (`tDisplay` *pDisplay, unsigned char *puImage, long IWidth, long IHeight)
- void `GrOffScreen8BPPPaletteSet` (`tDisplay` *pDisplay, unsigned long *pulPalette, unsigned long ulOffset, unsigned long ulCount)
- void `GrRectDraw` (const `tContext` *pContext, const `tRectangle` *pRect)
- void `GrRectFill` (const `tContext` *pContext, const `tRectangle` *pRect)
- void `GrStringDraw` (const `tContext` *pContext, const char *pcString, int ILength, unsigned IX, unsigned IY, int bOpaque)
- int `GrStringWidthGet` (const `tContext` *pContext, const char *pcString, int ILength)
- void `GrStringTableSet` (const void *pvTable)
- int `GrStringLanguageSet` (unsigned short usLangID)
- unsigned `GrStringGet` (int iIndex, char *pcData, unsigned uSize)
- int `GrRectOverlapCheck` (`tRectangle` *psRect1, `tRectangle` *psRect2)
- int `GrRectIntersectGet` (`tRectangle` *psRect1, `tRectangle` *psRect2, `tRectangle` *psIntersect)

Variables

- const `tFont g_sFontCm12`
- const `tFont g_sFontCm12b`
- const `tFont g_sFontCm12i`
- const `tFont g_sFontCm14`
- const `tFont g_sFontCm14b`
- const `tFont g_sFontCm14i`

- const [tFont g_sFontCm16](#)
- const [tFont g_sFontCm16b](#)
- const [tFont g_sFontCm16i](#)
- const [tFont g_sFontCm18](#)
- const [tFont g_sFontCm18b](#)
- const [tFont g_sFontCm18i](#)
- const [tFont g_sFontCm20](#)
- const [tFont g_sFontCm20b](#)
- const [tFont g_sFontCm20i](#)
- const [tFont g_sFontCm22](#)
- const [tFont g_sFontCm22b](#)
- const [tFont g_sFontCm22i](#)
- const [tFont g_sFontCm24](#)
- const [tFont g_sFontCm24b](#)
- const [tFont g_sFontCm24i](#)
- const [tFont g_sFontCm26](#)
- const [tFont g_sFontCm26b](#)
- const [tFont g_sFontCm26i](#)
- const [tFont g_sFontCm28](#)
- const [tFont g_sFontCm28b](#)
- const [tFont g_sFontCm28i](#)
- const [tFont g_sFontCm30](#)
- const [tFont g_sFontCm30b](#)
- const [tFont g_sFontCm30i](#)
- const [tFont g_sFontCm32](#)
- const [tFont g_sFontCm32b](#)
- const [tFont g_sFontCm32i](#)
- const [tFont g_sFontCm34](#)
- const [tFont g_sFontCm34b](#)
- const [tFont g_sFontCm34i](#)
- const [tFont g_sFontCm36](#)
- const [tFont g_sFontCm36b](#)
- const [tFont g_sFontCm36i](#)
- const [tFont g_sFontCm38](#)
- const [tFont g_sFontCm38b](#)
- const [tFont g_sFontCm38i](#)
- const [tFont g_sFontCm40](#)
- const [tFont g_sFontCm40b](#)
- const [tFont g_sFontCm40i](#)
- const [tFont g_sFontCm42](#)
- const [tFont g_sFontCm42b](#)
- const [tFont g_sFontCm42i](#)
- const [tFont g_sFontCm44](#)
- const [tFont g_sFontCm44b](#)
- const [tFont g_sFontCm44i](#)
- const [tFont g_sFontCm46](#)

- const tFont g_sFontCm46b
- const tFont g_sFontCm46i
- const tFont g_sFontCm48
- const tFont g_sFontCm48b
- const tFont g_sFontCm48i
- const tFont g_sFontCmsc12
- const tFont g_sFontCmsc14
- const tFont g_sFontCmsc16
- const tFont g_sFontCmsc18
- const tFont g_sFontCmsc20
- const tFont g_sFontCmsc22
- const tFont g_sFontCmsc24
- const tFont g_sFontCmsc26
- const tFont g_sFontCmsc28
- const tFont g_sFontCmsc30
- const tFont g_sFontCmsc32
- const tFont g_sFontCmsc34
- const tFont g_sFontCmsc36
- const tFont g_sFontCmsc38
- const tFont g_sFontCmsc40
- const tFont g_sFontCmsc42
- const tFont g_sFontCmsc44
- const tFont g_sFontCmsc46
- const tFont g_sFontCmsc48
- const tFont g_sFontCmss12
- const tFont g_sFontCmss12b
- const tFont g_sFontCmss12i
- const tFont g_sFontCmss14
- const tFont g_sFontCmss14b
- const tFont g_sFontCmss14i
- const tFont g_sFontCmss16
- const tFont g_sFontCmss16b
- const tFont g_sFontCmss16i
- const tFont g_sFontCmss18
- const tFont g_sFontCmss18b
- const tFont g_sFontCmss18i
- const tFont g_sFontCmss20
- const tFont g_sFontCmss20b
- const tFont g_sFontCmss20i
- const tFont g_sFontCmss22
- const tFont g_sFontCmss22b
- const tFont g_sFontCmss22i
- const tFont g_sFontCmss24
- const tFont g_sFontCmss24b
- const tFont g_sFontCmss24i
- const tFont g_sFontCmss26

- const tFont g_sFontCmss26b
- const tFont g_sFontCmss26i
- const tFont g_sFontCmss28
- const tFont g_sFontCmss28b
- const tFont g_sFontCmss28i
- const tFont g_sFontCmss30
- const tFont g_sFontCmss30b
- const tFont g_sFontCmss30i
- const tFont g_sFontCmss32
- const tFont g_sFontCmss32b
- const tFont g_sFontCmss32i
- const tFont g_sFontCmss34
- const tFont g_sFontCmss34b
- const tFont g_sFontCmss34i
- const tFont g_sFontCmss36
- const tFont g_sFontCmss36b
- const tFont g_sFontCmss36i
- const tFont g_sFontCmss38
- const tFont g_sFontCmss38b
- const tFont g_sFontCmss38i
- const tFont g_sFontCmss40
- const tFont g_sFontCmss40b
- const tFont g_sFontCmss40i
- const tFont g_sFontCmss42
- const tFont g_sFontCmss42b
- const tFont g_sFontCmss42i
- const tFont g_sFontCmss44
- const tFont g_sFontCmss44b
- const tFont g_sFontCmss44i
- const tFont g_sFontCmss46
- const tFont g_sFontCmss46b
- const tFont g_sFontCmss46i
- const tFont g_sFontCmss48
- const tFont g_sFontCmss48b
- const tFont g_sFontCmss48i
- const tFont g_sFontFixed6x8

6.166.1 Define Documentation

6.166.1.1 #define ClrAliceBlue 0x00F0F8FF

6.166.1.2 #define ClrAntiqueWhite 0x00FAEBD7

6.166.1.3 #define ClrAqua 0x0000FFFF

```
6.166.1.4 #define ClrAquamarine 0x007FFFFD4  
6.166.1.5 #define ClrAzure 0x00F0FFFF  
6.166.1.6 #define ClrBeige 0x00F5F5DC  
6.166.1.7 #define ClrBisque 0x00FFE4C4  
6.166.1.8 #define ClrBlack 0x00000000  
6.166.1.9 #define ClrBlanchedAlmond 0x00FFEBBC  
6.166.1.10 #define ClrBlue 0x000000FF  
6.166.1.11 #define ClrBlueMask 0x000000FF  
6.166.1.12 #define ClrBlueShift 0  
6.166.1.13 #define ClrBlueViolet 0x008A2BE2  
6.166.1.14 #define ClrBrown 0x00A52A2A  
6.166.1.15 #define ClrBurlyWood 0x00DEB887  
6.166.1.16 #define ClrCadetBlue 0x005F9EA0  
6.166.1.17 #define ClrChartreuse 0x007FFF00  
6.166.1.18 #define ClrChocolate 0x00D2691E  
6.166.1.19 #define ClrCoral 0x00FF7F50  
6.166.1.20 #define ClrCornflowerBlue 0x006495ED  
6.166.1.21 #define ClrCornsilk 0x00FFF8DC  
6.166.1.22 #define ClrCrimson 0x00DC143C  
6.166.1.23 #define ClrCyan 0x0000FFFF  
6.166.1.24 #define ClrDarkBlue 0x0000008B  
6.166.1.25 #define ClrDarkCyan 0x00008B8B  
6.166.1.26 #define ClrDarkGoldenrod 0x00B8860B  
6.166.1.27 #define ClrDarkGray 0x00A9A9A9
```

```
6.166.1.28 #define ClrDarkGreen 0x00006400
6.166.1.29 #define ClrDarkKhaki 0x00BDB76B
6.166.1.30 #define ClrDarkMagenta 0x008B008B
6.166.1.31 #define ClrDarkOliveGreen 0x00556B2F
6.166.1.32 #define ClrDarkOrange 0x00FF8C00
6.166.1.33 #define ClrDarkOrchid 0x009932CC
6.166.1.34 #define ClrDarkRed 0x008B0000
6.166.1.35 #define ClrDarkSalmon 0x00E9967A
6.166.1.36 #define ClrDarkSeaGreen 0x008FBC8F
6.166.1.37 #define ClrDarkSlateBlue 0x00483D8B
6.166.1.38 #define ClrDarkSlateGray 0x002F4F4F
6.166.1.39 #define ClrDarkTurquoise 0x0000CED1
6.166.1.40 #define ClrDarkViolet 0x009400D3
6.166.1.41 #define ClrDeepPink 0x00FF1493
6.166.1.42 #define ClrDeepSkyBlue 0x0000BFFF
6.166.1.43 #define ClrDimGray 0x00696969
6.166.1.44 #define ClrDodgerBlue 0x001E90FF
6.166.1.45 #define ClrFireBrick 0x00B22222
6.166.1.46 #define ClrFloralWhite 0x00FFFFAF
6.166.1.47 #define ClrForestGreen 0x00228B22
6.166.1.48 #define ClrFuchsia 0x00FF00FF
6.166.1.49 #define ClrGainsboro 0x00DCDCDC
6.166.1.50 #define ClrGhostWhite 0x00F8F8FF
6.166.1.51 #define ClrGold 0x00FFD700
```

```
6.166.1.52 #define ClrGoldenrod 0x00DAA520
6.166.1.53 #define ClrGray 0x00808080
6.166.1.54 #define ClrGreen 0x00008000
6.166.1.55 #define ClrGreenMask 0x0000FF00
6.166.1.56 #define ClrGreenShift 8
6.166.1.57 #define ClrGreenYellow 0x00ADFF2F
6.166.1.58 #define ClrHoneydew 0x00F0FFFF
6.166.1.59 #define ClrHotPink 0x00FF69B4
6.166.1.60 #define ClrIndianRed 0x00CD5C5C
6.166.1.61 #define ClrIndigo 0x004B0082
6.166.1.62 #define ClrIvory 0x00FFFFFF
6.166.1.63 #define ClrKhaki 0x00F0E68C
6.166.1.64 #define ClrLavender 0x00E6E6FA
6.166.1.65 #define ClrLavenderBlush 0x00FFF0F5
6.166.1.66 #define ClrLawnGreen 0x007CFC00
6.166.1.67 #define ClrLemonChiffon 0x00FFFACD
6.166.1.68 #define ClrLightBlue 0x00ADD8E6
6.166.1.69 #define ClrLightCoral 0x00F08080
6.166.1.70 #define ClrLightCyan 0x00E0FFFF
6.166.1.71 #define ClrLightGoldenrodYellow 0x00FAFAD2
6.166.1.72 #define ClrLightGreen 0x0090EE90
6.166.1.73 #define ClrLightGrey 0x00D3D3D3
6.166.1.74 #define ClrLightPink 0x00FFB6C1
6.166.1.75 #define ClrLightSalmon 0x00FFA07A
```

```
6.166.1.76 #define ClrLightSeaGreen 0x0020B2AA  
6.166.1.77 #define ClrLightSkyBlue 0x0087CEFA  
6.166.1.78 #define ClrLightSlateGray 0x00778899  
6.166.1.79 #define ClrLightSteelBlue 0x00B0C4DE  
6.166.1.80 #define ClrLightYellow 0x00FFFFFF  
6.166.1.81 #define ClrLime 0x0000FF00  
6.166.1.82 #define ClrLimeGreen 0x0032CD32  
6.166.1.83 #define ClrLinen 0x00FAF0E6  
6.166.1.84 #define ClrMagenta 0x00FF00FF  
6.166.1.85 #define ClrMaroon 0x00800000  
6.166.1.86 #define ClrMediumAquamarine 0x0066CDAA  
6.166.1.87 #define ClrMediumBlue 0x000000CD  
6.166.1.88 #define ClrMediumOrchid 0x00BA55D3  
6.166.1.89 #define ClrMediumPurple 0x009370DB  
6.166.1.90 #define ClrMediumSeaGreen 0x003CB371  
6.166.1.91 #define ClrMediumSlateBlue 0x007B68EE  
6.166.1.92 #define ClrMediumSpringGreen 0x0000FA9A  
6.166.1.93 #define ClrMediumTurquoise 0x0048D1CC  
6.166.1.94 #define ClrMediumVioletRed 0x00C71585  
6.166.1.95 #define ClrMidnightBlue 0x00191970  
6.166.1.96 #define ClrMintCream 0x00F5FFFA  
6.166.1.97 #define ClrMistyRose 0x00FFE4E1  
6.166.1.98 #define ClrMoccasin 0x00FFE4B5  
6.166.1.99 #define ClrNavajoWhite 0x00FFDEAD
```

```
6.166.1.100 #define ClrNavy 0x00000080
6.166.1.101 #define ClrOldLace 0x00FDF5E6
6.166.1.102 #define ClrOlive 0x00808000
6.166.1.103 #define ClrOliveDrab 0x006B8E23
6.166.1.104 #define ClrOrange 0x00FFA500
6.166.1.105 #define ClrOrangeRed 0x00FF4500
6.166.1.106 #define ClrOrchid 0x00DA70D6
6.166.1.107 #define ClrPaleGoldenrod 0x00EEE8AA
6.166.1.108 #define ClrPaleGreen 0x0098FB98
6.166.1.109 #define ClrPaleTurquoise 0x00AFEEEE
6.166.1.110 #define ClrPaleVioletRed 0x00DB7093
6.166.1.111 #define ClrPapayaWhip 0x00FFEFDS
6.166.1.112 #define ClrPeachPuff 0x00FFDAB9
6.166.1.113 #define ClrPeru 0x00CD853F
6.166.1.114 #define ClrPink 0x00FFC0CB
6.166.1.115 #define ClrPlum 0x00DDA0DD
6.166.1.116 #define ClrPowderBlue 0x00B0E0E6
6.166.1.117 #define ClrPurple 0x00800080
6.166.1.118 #define ClrRed 0x00FF0000
6.166.1.119 #define ClrRedMask 0x00FF0000
6.166.1.120 #define ClrRedShift 16
6.166.1.121 #define ClrRosyBrown 0x00BC8F8F
6.166.1.122 #define ClrRoyalBlue 0x004169E1
6.166.1.123 #define ClrSaddleBrown 0x008B4513
```

```
6.166.1.124 #define ClrSalmon 0x00FA8072
6.166.1.125 #define ClrSandyBrown 0x00F4A460
6.166.1.126 #define ClrSeaGreen 0x002E8B57
6.166.1.127 #define ClrSeashell 0x00FFF5EE
6.166.1.128 #define ClrSienna 0x00A0522D
6.166.1.129 #define ClrSilver 0x00C0C0C0
6.166.1.130 #define ClrSkyBlue 0x0087CEEB
6.166.1.131 #define ClrSlateBlue 0x006A5ACD
6.166.1.132 #define ClrSlateGray 0x00708090
6.166.1.133 #define ClrSnow 0x00FFFAFA
6.166.1.134 #define ClrSpringGreen 0x0000FF7F
6.166.1.135 #define ClrSteelBlue 0x004682B4
6.166.1.136 #define ClrTan 0x00D2B48C
6.166.1.137 #define ClrTeal 0x00008080
6.166.1.138 #define ClrThistle 0x00D8BFD8
6.166.1.139 #define ClrTomato 0x00FF6347
6.166.1.140 #define ClrTurquoise 0x0040E0D0
6.166.1.141 #define ClrViolet 0x00EE82EE
6.166.1.142 #define ClrWheat 0x00F5DEB3
6.166.1.143 #define ClrWhite 0x00FFFFFF
6.166.1.144 #define ClrWhiteSmoke 0x00F5F5F5
6.166.1.145 #define ClrYellow 0x00FFFF00
6.166.1.146 #define ClrYellowGreen 0x009ACD32
```

```
6.166.1.147 #define DpyColorTranslate( pDisplay, ulValue
) ((pDisplay)->pfnColorTranslate((pDisplay)->pvDisplayData, ulValue))
```

Translates a 24-bit RGB color to a display driver-specific color.

Parameters

<i>pDisplay</i>	is the pointer to the display driver structure for the display to operate upon.
<i>ulValue</i>	is the 24-bit RGB color. The least-significant byte is the blue channel, the next byte is the green channel, and the third byte is the red channel.

This function translates a 24-bit RGB color into a value that can be written into the display's frame buffer in order to reproduce that color, or the closest possible approximation of that color.

Returns

Returns the display-driver specific color.

```
6.166.1.148 #define DpyFlush( pDisplay )
```

Value:

```
{
    \
    const tDisplay *pD = pDisplay; \
    pD->pfnFlush(pD->pvDisplayData); \
}
```

Flushes cached drawing operations.

Parameters

<i>pDisplay</i>	is the pointer to the display driver structure for the display to operate upon.
-----------------	---

This function flushes any cached drawing operations on a display.

Returns

None.

```
6.166.1.149 #define DpyHeightGet( pDisplay ) ((pDisplay)->usHeight)
```

Gets the height of the display.

Parameters

<i>pDisplay</i>	is a pointer to the display driver structure for the display to query.
-----------------	--

This function determines the height of the display.

Returns

Returns the height of the display in pixels.

6.166.1.150 #define DpyLineDrawH(*pDisplay*, *lX1*, *lX2*, *lY*, *ulValue*)

Value:

```
{
    const tDisplay *pD = pDisplay;
    pD->pfnLineDrawH(pD->pvDisplayData, lX1, lX2, lY, ulValue);
}
```

Draws a horizontal line on a display.

Parameters

<i>pDisplay</i>	is the pointer to the display driver structure for the display to operate upon.
<i>lX1</i>	is the starting X coordinate of the line.
<i>lX2</i>	is the ending X coordinate of the line.
<i>lY</i>	is the Y coordinate of the line.
<i>ulValue</i>	is the color to draw the line.

This function draws a horizontal line on a display. This assumes that clipping has already been performed, and that both end points of the line are within the extents of the display.

Returns

None.

6.166.1.151 #define DpyLineDrawV(*pDisplay*, *lX*, *lY1*, *lY2*, *ulValue*)

Value:

```
{
    const tDisplay *pD = pDisplay;
    pD->pfnLineDrawV(pD->pvDisplayData, lX, lY1, lY2, ulValue);
}
```

Draws a vertical line on a display.

Parameters

<i>pDisplay</i>	is the pointer to the display driver structure for the display to operate upon.
<i>lX</i>	is the X coordinate of the line.
<i>lY1</i>	is the starting Y coordinate of the line.
<i>lY2</i>	is the ending Y coordinate of the line.
Generated on Wed Oct 24 2012 22:18:50 for GLCD by Doxygen	is the color to draw the line.

This function draws a vertical line on a display. This assumes that clipping has already been performed, and that both end points of the line are within the extents of the display.

Returns

None.

6.166.1.152 #define DpyPixelDraw(*pDisplay*, *IX*, *IY*, *ulValue*)

Value:

```
{
    const tDisplay *pD = pDisplay;
    pD->pfnPixelDraw(pD->pvDisplayData, lX, lY, ulValue); \
}
```

Draws a pixel on a display.

Parameters

<i>pDisplay</i>	is the pointer to the display driver structure for the display to operate upon.
<i>IX</i>	is the X coordinate of the pixel.
<i>IY</i>	is the Y coordinate of the pixel.
<i>ulValue</i>	is the color to draw the pixel.

This function draws a pixel on a display. This assumes that clipping has already been performed.

Returns

None.

6.166.1.153 #define DpyPixelDrawMultiple(*pDisplay*, *IX*, *IY*, *lX0*, *lCount*, *lBPP*, *pucData*, *pucPalette*)

Value:

```
{
    const tDisplay *pD = pDisplay;
    pD->pfnPixelDrawMultiple(pD->pvDisplayData, lX, lY, lX0, lCount, \
                                lBPP, pucData, pucPalette); \
}
```

Draws a horizontal sequence of pixels on a display.

Parameters

<i>pDisplay</i>	is the pointer to the display driver structure for the display to operate upon.
<i>IX</i>	is the X coordinate of the first pixel.
<i>IY</i>	is the Y coordinate of the first pixel.
<i>IX0</i>	is sub-pixel offset within the pixel data, which is valid for 1 or 4 bit per pixel formats.
<i>ICount</i>	is the number of pixels to draw.
<i>IBPP</i>	is the number of bits per pixel; must be 1, 4, or 8.
<i>pucData</i>	is a pointer to the pixel data. For 1 and 4 bit per pixel formats, the most significant bit(s) represent the left-most pixel.
<i>pucPalette</i>	is a pointer to the palette used to draw the pixels.

This function draws a horizontal sequence of pixels on a display, using the supplied palette. For 1 bit per pixel format, the palette contains pre-translated colors; for 4 and 8 bit per pixel formats, the palette contains 24-bit RGB values that must be translated before being written to the display.

Returns

None.

6.166.1.154 #define DpyRectFill(*pDisplay*, *pRect*, *ulValue*)**Value:**

```
{
    const tDisplay *pD = pDisplay;
    pD->pfnRectFill(pD->pvDisplayData, pRect, ulValue); \
}
```

Fills a rectangle on a display.

Parameters

<i>pDisplay</i>	is the pointer to the display driver structure for the display to operate upon.
<i>pRect</i>	is a pointer to the structure describing the rectangle to fill.
<i>ulValue</i>	is the color to fill the rectangle.

This function fills a rectangle on the display. This assumes that clipping has already been performed, and that all sides of the rectangle are within the extents of the display.

Returns

None.

6.166.1.155 #define DpyWidthGet(*pDisplay*) ((*pDisplay*)>usWidth)

Gets the width of the display.

Parameters

<i>pDisplay</i>	is a pointer to the display driver structure for the display to query.
-----------------	--

This function determines the width of the display.

Returns

Returns the width of the display in pixels.

6.166.1.156 #define FONT_FMT_PIXEL_RLE 0x01

Indicates that the font data is stored using a pixel-based RLE format.

6.166.1.157 #define FONT_FMT_UNCOMPRESSED 0x00

Indicates that the font data is stored in an uncompressed format.

6.166.1.158 #define GrContextBackgroundSet(*pContext*, *ulValue*)**Value:**

```
do \\\n    { \\\n        tContext *pC = pContext; \\\n        pC->ulBackground = DpyColorTranslate(pC->pDisplay, ulValue); \\\n    } \\\n    while(0) \\\n
```

Sets the background color to be used.

Parameters

<i>pContext</i>	is a pointer to the drawing context to modify.
<i>ulValue</i>	is the 24-bit RGB color to be used.

This function sets the background color to be used for drawing operations in the specified drawing context.

Returns

None.

6.166.1.159 #define GrContextBackgroundSetTranslated(*pContext*, *ulValue*)

Value:

```
do \n{ \n    tContext *pC = pContext; \n    pC->ulBackground = ulValue; \n} \nwhile(0) \n
```

Sets the background color to be used.

Parameters

<i>pContext</i>	is a pointer to the drawing context to modify.
<i>ulValue</i>	is the display driver-specific color to be used.

This function sets the background color to be used for drawing operations in the specified drawing context, using a color that has been previously translated to a driver-specific color (for example, via [DpyColorTranslate\(\)](#)).

Returns

None.

6.166.1.160 #define GrContextDpyHeightGet(*pContext*
)(DpyHeightGet((*pContext*)->pDisplay))

Gets the height of the display being used by this drawing context.

Parameters

<i>pContext</i>	is a pointer to the drawing context to query.
-----------------	---

This function returns the height of the display that is being used by this drawing context.

Returns

Returns the height of the display in pixels.

6.166.1.161 #define GrContextDpyWidthGet(*pContext*
)(DpyWidthGet((*pContext*)->pDisplay))

Gets the width of the display being used by this drawing context.

Parameters

<i>pContext</i>	is a pointer to the drawing context to query.
-----------------	---

This function returns the width of the display that is being used by this drawing context.

Returns

Returns the width of the display in pixels.

6.166.1.162 #define GrContextFontSet(*pContext*, *pFnt*)

Value:

```
do                                \
{                                \
    tContext *pC = pContext;      \
    const tFont *pF = pFnt;       \
    pC->pFont = pF;             \
}                                \
while(0)
```

Sets the font to be used.

Parameters

<i>pContext</i>	is a pointer to the drawing context to modify.
<i>pFnt</i>	is a pointer to the font to be used.

This function sets the font to be used for string drawing operations in the specified drawing context.

Returns

None.

6.166.1.163 #define GrContextForegroundSet(*pContext*, *ulValue*)

Value:

```
do                                \
{                                \
    tContext *pC = pContext;      \
    pC->ulForeground = DpyColorTranslate(pC->pDisplay, ulValue); \
}                                \
while(0)
```

Sets the foreground color to be used.

Parameters

<i>pContext</i>	is a pointer to the drawing context to modify.
<i>ulValue</i>	is the 24-bit RGB color to be used.

This function sets the color to be used for drawing operations in the specified drawing context.

Returns

None.

6.166.1.164 #define GrContextForegroundSetTranslated(*pContext*, *ulValue*)

Value:

```
do \n{ \n    tContext *pC = pContext; \n    pC->ulForeground = ulValue; \n} \nwhile(0) \n
```

Sets the foreground color to be used.

Parameters

<i>pContext</i>	is a pointer to the drawing context to modify.
<i>ulValue</i>	is the display driver-specific color to be used.

This function sets the foreground color to be used for drawing operations in the specified drawing context, using a color that has been previously translated to a driver-specific color (for example, via [DpyColorTranslate\(\)](#)).

Returns

None.

6.166.1.165 #define GrFlush(*pContext*)

Value:

```
do \n{ \n    const tContext *pC = pContext; \n    DpyFlush(pC->pDisplay); \n} \nwhile(0) \n
```

Flushes any cached drawing operations.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
-----------------	---

This function flushes any cached drawing operations. For display drivers that draw into a local frame buffer before writing to the actual display, calling this function will cause the display to be updated to match the contents of the local frame buffer.

Returns

None.

6.166.1.166 #define GrFontBaselineGet(*pFont*) ((*pFont*)->ucBaseline)

Gets the baseline of a font.

Parameters

<i>pFont</i>	is a pointer to the font to query.
--------------	------------------------------------

This function determines the baseline position of a font. The baseline is the offset between the top of the font and the bottom of the capital letters. The only font data that exists below the baseline are the descenders on some lower-case letters (such as "y").

Returns

Returns the baseline of the font, in pixels.

6.166.1.167 #define GrFontHeightGet(*pFont*) ((*pFont*)->ucHeight)

Gets the height of a font.

Parameters

<i>pFont</i>	is a pointer to the font to query.
--------------	------------------------------------

This function determines the height of a font. The height is the offset between the top of the font and the bottom of the font, including any ascenders and descenders.

Returns

Returns the height of the font, in pixels.

6.166.1.168 #define GrFontMaxWidthGet(*pFont*) ((*pFont*)->ucMaxWidth)

Gets the maximum width of a font.

Parameters

<i>pFont</i>	is a pointer to the font to query.
--------------	------------------------------------

This function determines the maximum width of a font. The maximum width is the width of the widest individual character in the font.

Returns

Returns the maximum width of the font, in pixels.

6.166.1.169 #define GrImageColorsGet(*puclImage*) (((unsigned char *)*puclImage*)[5] + 1)

Gets the number of colors in an image.

Parameters

<i>puclImage</i>	is a pointer to the image to query.
------------------	-------------------------------------

This function determines the number of colors in the palette of an image. This is only valid for 4bpp and 8bpp images; 1bpp images do not contain a palette.

Returns

Returns the number of colors in the image.

6.166.1.170 #define GrImageHeightGet(*puclImage*) (*(unsigned short *)(*puclImage* + 3))

Gets the height of an image.

Parameters

<i>puclImage</i>	is a pointer to the image to query.
------------------	-------------------------------------

This function determines the height of an image in pixels.

Returns

Returns the height of the image in pixels.

6.166.1.171 #define GrImageWidthGet(*puclImage*) (*(unsigned short *)(*puclImage* + 1))

Gets the width of an image.

Parameters

<i>puclImage</i>	is a pointer to the image to query.
------------------	-------------------------------------

This function determines the width of an image in pixels.

Returns

Returns the width of the image in pixels.

```
6.166.1.172 #define GrLangDe 0x0407  
6.166.1.173 #define GrLangEnAUS 0x0C09  
6.166.1.174 #define GrLangEnCA 0x1009  
6.166.1.175 #define GrLangEnNZ 0x1409  
6.166.1.176 #define GrLangEnUK 0x0809  
6.166.1.177 #define GrLangEnUS 0x0409  
6.166.1.178 #define GrLangEsMX 0x080A  
6.166.1.179 #define GrLangEsSP 0x0C0A  
6.166.1.180 #define GrLangFr 0x040C  
6.166.1.181 #define GrLangHi 0x0439  
6.166.1.182 #define GrLangIt 0x0410  
6.166.1.183 #define GrLangJp 0x0411  
6.166.1.184 #define GrLangKo 0x0412  
6.166.1.185 #define GrLangSwKE 0x0441  
6.166.1.186 #define GrLangUrIN 0x0820  
6.166.1.187 #define GrLangUrPK 0x0420  
6.166.1.188 #define GrLangZhPRC 0x0804  
6.166.1.189 #define GrLangZhTW 0x0404  
6.166.1.190 #define GrOffScreen1BPPSize( IWidth, IHeight ) (5 + (((IWidth + 7) / 8) *  
                  IHeight))
```

Determines the size of the buffer for a 1 BPP off-screen image.

Parameters

<i>IWidth</i>	is the width of the image in pixels.
<i>IHeight</i>	is the height of the image in pixels.

This function determines the size of the memory buffer required to hold a 1 BPP off-screen image of the specified geometry.

Returns

Returns the number of bytes required by the image.

```
6.166.1.191 #define GrOffScreen1BPPSize( IWidth, IHeight ) (6 + (16 * 3) + (((IWidth + 1) / 2) * IHeight))
```

Determines the size of the buffer for a 1 BPP off-screen image.

Parameters

<i>IWidth</i>	is the width of the image in pixels.
<i>IHeight</i>	is the height of the image in pixels.

This function determines the size of the memory buffer required to hold a 4 BPP off-screen image of the specified geometry.

Returns

Returns the number of bytes required by the image.

```
6.166.1.192 #define GrOffScreen4BPPSize( IWidth, IHeight ) (6 + (16 * 3) + ((IWidth * IHeight)))
```

Determines the size of the buffer for a 4 BPP off-screen image.

Parameters

<i>IWidth</i>	is the width of the image in pixels.
<i>IHeight</i>	is the height of the image in pixels.

This function determines the size of the memory buffer required to hold an 8 BPP off-screen image of the specified geometry.

Returns

Returns the number of bytes required by the image.

```
6.166.1.193 #define GrPixelDraw( pContext, IX, IY )
```

Value:

do

{

\

\

```

const tContext *pC = pContext;
if((lX >= pC->sClipRegion.sXMin) &&
   (lX <= pC->sClipRegion.sXMax) &&
   (lY >= pC->sClipRegion.sYMin) &&
   (lY <= pC->sClipRegion.sYMax))
{
    DpyPixelDraw(pC->pDisplay, lX, lY, pC->ulForeground);
}
while(0)

```

Draws a pixel.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>lX</i>	is the X coordinate of the pixel.
<i>lY</i>	is the Y coordinate of the pixel.

This function draws a pixel if it resides within the clipping region.

Returns

None.

6.166.1.194 #define GrRectContainsPoint(*pRect*, *lX*, *lY*)

Value:

```

(((lX) >= (pRect)->sXMin) && ((lX) <= (pRect)->sXMax) &&
 ((lY) >= (pRect)->sYMin) && ((lY) <= (pRect)->sYMax)) ? 1 : 0

```

Determines if a point lies within a given rectangle.

Parameters

<i>pRect</i>	is a pointer to the rectangle which the point is to be checked against.
<i>lX</i>	is the X coordinate of the point to be checked.
<i>lY</i>	is the Y coordinate of the point to be checked.

This function determines whether point (*lX*, *lY*) lies within the rectangle described by *pRect*.

Returns

Returns 1 if the point is within the rectangle or 0 otherwise.

6.166.1.195 #define GrStringBaselineGet(*pContext*) ((pContext)->pFont->ucBaseline)

Gets the baseline of a string.

Parameters

<i>pContext</i>	is a pointer to the drawing context to query.
-----------------	---

This function determines the baseline position of a string. The baseline is the offset between the top of the string and the bottom of the capital letters. The only string data that exists below the baseline are the descenders on some lower-case letters (such as "y").

Returns

Returns the baseline of the string, in pixels.

```
6.166.1.196 #define GrStringDrawCentered( pContext, pcString, lLength, IX, IY,
bOpaque )
```

Value:

```
do
{
    const tContext *pC = pContext;
    const char *pcStr = pcString;

    GrStringDraw(pC, pcStr, lLength,
                 (IX) - (GrStringWidthGet(pC, pcStr, lLength) / 2),
                 (IY) - (pC->pFont->ucBaseline / 2), bOpaque);
}
while(0)
```

Draws a centered string.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>pcString</i>	is a pointer to the string to be drawn.
<i>lLength</i>	is the number of characters from the string that should be drawn on the screen.
<i>IX</i>	is the X coordinate of the center of the string position on the screen.
<i>IY</i>	is the Y coordinate of the center of the string position on the screen.
<i>bOpaque</i>	is true if the background of each character should be drawn and false if it should not (leaving the background as is).

This function draws a string of test on the screen centered upon the provided position. The *lLength* parameter allows a portion of the string to be examined without having to insert a NULL character at the stopping point (which would not be possible if the string was located in flash); specifying a length of -1 will cause the entire string to be rendered (subject to clipping).

Returns

None.

6.166.1.197 #define GrStringHeightGet(*pContext*) ((*pContext*)->*pFont*->*ucHeight*)

Gets the height of a string.

Parameters

<i>pContext</i>	is a pointer to the drawing context to query.
-----------------	---

This function determines the height of a string. The height is the offset between the top of the string and the bottom of the string, including any ascenders and descenders. Note that this will not account for the case where the string in question does not have any characters that use descenders but the font in the drawing context does contain characters with descenders.

Returns

Returns the height of the string, in pixels.

6.166.1.198 #define GrStringMaxWidthGet(*pContext*) ((*pContext*)->*pFont*->*ucMaxWidth*)

Gets the maximum width of a character in a string.

Parameters

<i>pContext</i>	is a pointer to the drawing context to query.
-----------------	---

This function determines the maximum width of a character in a string. The maximum width is the width of the widest individual character in the font used to render the string, which may be wider than the widest character that is used to render a particular string.

Returns

Returns the maximum width of a character in a string, in pixels.

6.166.1.199 #define IMAGE_FMT_16BPP_UNCOMP 0x10

Indicates that the image data is not compressed and represents each pixel with sixteen bits.

6.166.1.200 #define IMAGE_FMT_1BPP_COMP 0x81

Indicates that the image data is compressed and represents each pixel with a single bit.

6.166.1.201 #define IMAGE_FMT_1BPP_UNCOMP 0x01

Indicates that the image data is not compressed and represents each pixel with a single bit.

6.166.1.202 #define IMAGE_FMT_4BPP_COMP 0x84

Indicates that the image data is compressed and represents each pixel with four bits.

6.166.1.203 #define IMAGE_FMT_4BPP_UNCOMP 0x04

Indicates that the image data is not compressed and represents each pixel with four bits.

6.166.1.204 #define IMAGE_FMT_8BPP_COMP 0x88

Indicates that the image data is compressed and represents each pixel with eight bits.

6.166.1.205 #define IMAGE_FMT_8BPP_UNCOMP 0x08

Indicates that the image data is not compressed and represents each pixel with eight bits.

6.166.2 Function Documentation

6.166.2.1 void GrCircleDraw (const tContext * pContext, unsigned IX, unsigned IY, unsigned IRadius)

Draws a circle.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX</i>	is the X coordinate of the center of the circle.
<i>IY</i>	is the Y coordinate of the center of the circle.
<i>IRadius</i>	is the radius of the circle.

This function draws a circle, utilizing the Bresenham circle drawing algorithm. The extent of the circle is from *IX* - *IRadius* to *IX* + *IRadius* and *IY* - *IRadius* to *IY* + *IRadius*, inclusive.

Returns

None.

6.166.2.2 void GrCircleFill (const tContext * pContext, unsigned IX, unsigned IY, unsigned IRadius)

Draws a filled circle.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX</i>	is the X coordinate of the center of the circle.
<i>IY</i>	is the Y coordinate of the center of the circle.
<i>IRadius</i>	is the radius of the circle.

This function draws a filled circle, utilizing the Bresenham circle drawing algorithm. The extent of the circle is from *IX* - *IRadius* to *IX* + *IRadius* and *IY* - *IRadius* to *IY* + *IRadius*, inclusive.

Returns

None.

6.166.2.3 void GrContextClipRegionSet (tContext * pContext, tRectangle * pRect)

Sets the extents of the clipping region.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>pRect</i>	is a pointer to the structure containing the extents of the clipping region.

This function sets the extents of the clipping region. The clipping region is not allowed to exceed the extents of the screen, but may be a portion of the screen.

The supplied coordinate are inclusive; *sXMin* of 1 and *sXMax* of 1 will define a clipping region that will display only the pixels in the *X* = 1 column. A consequence of this is that the clipping region must contain at least one row and one column.

Returns

None.

6.166.2.4 void GrContextInit (tContext * pContext, const tDisplay * pDisplay)

Initializes a drawing context.

Parameters

<i>pContext</i>	is a pointer to the drawing context to initialize.
<i>pDisplay</i>	is a pointer to the <i>tDisplayInfo</i> structure that describes the display driver to use.

This function initializes a drawing context, preparing it for use. The provided display driver will be used for all subsequent graphics operations, and the default clipping region will be set to the extent of the screen.

Returns

None.

6.166.2.5 void GrImageDraw (const tContext * pContext, const unsigned char * puImage, unsigned IX, unsigned IY)

Draws a bitmap image.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>puImage</i>	is a pointer to the image to draw.
<i>IX</i>	is the X coordinate of the upper left corner of the image.
<i>IY</i>	is the Y coordinate of the upper left corner of the image.

This function draws a bitmap image. The image may be 1 bit per pixel (using the foreground and background color from the drawing context), 4 bits per pixel (using a palette supplied in the image data), or 8 bits per pixel (using a palette supplied in the image data). It can be uncompressed data, or it can be compressed using the Lempel-Ziv--Storer-Szymanski algorithm (as published in the Journal of the ACM, 29(4):928-951, October 1982).

Returns

None.

6.166.2.6 void GrLineDraw (const tContext * pContext, unsigned IX1, unsigned IY1, unsigned IX2, unsigned IY2)

Draws a line.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX1</i>	is the X coordinate of the start of the line.
<i>IY1</i>	is the Y coordinate of the start of the line.
<i>IX2</i>	is the X coordinate of the end of the line.
<i>IY2</i>	is the Y coordinate of the end of the line.

This function draws a line, utilizing [GrLineDrawH\(\)](#) and [GrLineDrawV\(\)](#) to draw the line as efficiently as possible. The line is clipped to the clippping rectangle using the Cohen-Sutherland clipping algorithm, and then scan converted using Bresenham's line drawing algorithm.

Returns

None.

6.166.2.7 void GrLineDrawH (const tContext * pContext, unsigned IX1, unsigned IX2, unsigned IY)

Draws a horizontal line.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX1</i>	is the X coordinate of one end of the line.
<i>IX2</i>	is the X coordinate of the other end of the line.
<i>IY</i>	is the Y coordinate of the line.

This function draws a horizontal line, taking advantage of the fact that the line is horizontal to draw it more efficiently. The clipping of the horizontal line to the clipping rectangle is performed within this routine; the display driver's horizontal line routine is used to perform the actual line drawing.

Returns

None.

6.166.2.8 void GrLineDrawV (const tContext * pContext, unsigned IX, unsigned IY1, unsigned IY2)

Draws a vertical line.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>IX</i>	is the X coordinate of the line.
<i>IY1</i>	is the Y coordinate of one end of the line.
<i>IY2</i>	is the Y coordinate of the other end of the line.

This function draws a vertical line, taking advantage of the fact that the line is vertical to draw it more efficiently. The clipping of the vertical line to the clipping rectangle is performed within this routine; the display driver's vertical line routine is used to perform the actual line drawing.

Returns

None.

6.166.2.9 void GrRectDraw (const tContext * pContext, const tRectangle * pRect)

Draws a rectangle.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>pRect</i>	is a pointer to the structure containing the extents of the rectangle.

This function draws a rectangle. The rectangle will extend from *IXMin* to *IXMax* and *IYMin* to *IYMax*, inclusive.

Returns

None.

6.166.2.10 void GrRectFill (const tContext * pContext, const tRectangle * pRect)

Draws a filled rectangle.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>pRect</i>	is a pointer to the structure containing the extents of the rectangle.

This function draws a filled rectangle. The rectangle will extend from *IXMin* to *IXMax* and *IYMin* to *IYMax*, inclusive. The clipping of the rectangle to the clipping rectangle is performed within this routine; the display driver's rectangle fill routine is used to perform the actual rectangle fill.

Returns

None.

6.166.2.11 int GrRectIntersectGet (tRectangle * psRect1, tRectangle * psRect2, tRectangle * psIntersect)

Determines the intersection of two rectangles.

Parameters

<i>psRect1</i>	is a pointer to the first rectangle.
<i>psRect2</i>	is a pointer to the second rectangle.
<i>psIntersect</i>	is a pointer to a rectangle which will be written with the intersection of <i>psRect1</i> and <i>psRect2</i> .

This function determines if two rectangles overlap and, if they do, calculates the rectangle representing their intersection. If the rectangles do not overlap, 0 is returned and

psIntersect is not written.

Returns

Returns 1 if there is an overlap or 0 if not.

6.166.2.12 int GrRectOverlapCheck (tRectangle * *psRect1*, tRectangle * *psRect2*)

Determines if two rectangles overlap.

Parameters

<i>psRect1</i>	is a pointer to the first rectangle.
<i>psRect2</i>	is a pointer to the second rectangle.

This function determines whether two rectangles overlap. It assumes that rectangles *psRect1* and *psRect2* are valid with *sXMin* < *sXMax* and *sYMin* < *sYMax*.

Returns

Returns 1 if there is an overlap or 0 if not.

6.166.2.13 void GrStringDraw (const tContext * *pContext*, const char * *pcString*, int *lLength*, unsigned *IX*, unsigned *IY*, int *bOpaque*)

Draws a string.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>pcString</i>	is a pointer to the string to be drawn.
<i>lLength</i>	is the number of characters from the string that should be drawn on the screen.
<i>IX</i>	is the X coordinate of the upper left corner of the string position on the screen.
<i>IY</i>	is the Y coordinate of the upper left corner of the string position on the screen.
<i>bOpaque</i>	is true of the background of each character should be drawn and false if it should not (leaving the background as is).

This function draws a string of test on the screen. The *lLength* parameter allows a portion of the string to be examined without having to insert a NULL character at the stopping point (which would not be possible if the string was located in flash); specifying a length of -1 will cause the entire string to be rendered (subject to clipping).

Returns

None.

6.166.2.14 unsigned GrStringGet (int *iIndex*, char * *pcData*, unsigned *ulSize*)

This function returns a string from the current string table.

Parameters

<i>iIndex</i>	is the index of the string to retrieve.
<i>pcData</i>	is the pointer to the buffer to store the string into.
<i>ulSize</i>	is the size of the buffer provided by <i>pcData</i> .

This function will return a string from the string table in the language set by the [GrStringLanguageSet\(\)](#) function. The value passed in *iIndex* parameter is the string that is being requested and will be returned in the buffer provided in the *pcData* parameter. The amount of data returned will be limited by the *ulSize* parameter.

Returns

Returns the number of valid bytes returned in the *pcData* buffer.

6.166.2.15 int GrStringLanguageSet (unsigned short *usLangID*)

This function sets the current language for strings returned by the [GrStringGet\(\)](#) function.

Parameters

<i>usLangID</i>	is one of the language identifiers provided in the string table.
-----------------	--

This function is used to set the language identifier for the strings returned by the [GrStringGet\(\)](#) function. The *usLangID* parameter should match one of the identifiers that was included in the string table. These are provided in a header file in the graphics library and must match the values that were passed through the sting compression utility.

Returns

This function returns 0 if the language was not found and a non-zero value if the lagauge was found.

6.166.2.16 void GrStringTableSet (const void * *pvTable*)

This function sets the location of the current string table.

Parameters

<i>pvTable</i>	is a pointer to a string table that was generated by the string compression utility.
----------------	--

This function is used to set the string table to use for strings in an application. This string table is created by the string compression utility. This function is used to swap out multiple string tables if the application requires more than one table. It does not allow using more than one string table at a time.

Returns

None.

6.166.2.17 int GrStringWidthGet (const tContext * *pContext*, const char * *pcString*, int *lLength*)

Determines the width of a string.

Parameters

<i>pContext</i>	is a pointer to the drawing context to use.
<i>pcString</i>	is the string in question.
<i>lLength</i>	is the length of the string.

This function determines the width of a string (or portion of the string) when drawn with a particular font. The *lLength* parameter allows a portion of the string to be examined without having to insert a NULL character at the stopping point (would not be possible if the string was located in flash); specifying a length of -1 will cause the width of the entire string to be computed.

Returns

Returns the width of the string in pixels.

6.166.3 Variable Documentation

6.166.3.1 const tFont g_sFontCm12

6.166.3.2 const tFont g_sFontCm12b

6.166.3.3 const tFont g_sFontCm12i

6.166.3.4 const tFont g_sFontCm14

6.166.3.5 const tFont g_sFontCm14b

6.166.3.6 const tFont g_sFontCm14i

6.166.3.7 const tFont g_sFontCm16
6.166.3.8 const tFont g_sFontCm16b
6.166.3.9 const tFont g_sFontCm16i
6.166.3.10 const tFont g_sFontCm18
6.166.3.11 const tFont g_sFontCm18b
6.166.3.12 const tFont g_sFontCm18i
6.166.3.13 const tFont g_sFontCm20
6.166.3.14 const tFont g_sFontCm20b
6.166.3.15 const tFont g_sFontCm20i
6.166.3.16 const tFont g_sFontCm22
6.166.3.17 const tFont g_sFontCm22b
6.166.3.18 const tFont g_sFontCm22i
6.166.3.19 const tFont g_sFontCm24
6.166.3.20 const tFont g_sFontCm24b
6.166.3.21 const tFont g_sFontCm24i
6.166.3.22 const tFont g_sFontCm26
6.166.3.23 const tFont g_sFontCm26b
6.166.3.24 const tFont g_sFontCm26i
6.166.3.25 const tFont g_sFontCm28
6.166.3.26 const tFont g_sFontCm28b
6.166.3.27 const tFont g_sFontCm28i
6.166.3.28 const tFont g_sFontCm30
6.166.3.29 const tFont g_sFontCm30b
6.166.3.30 const tFont g_sFontCm30i

6.166.3.31 const tFont g_sFontCm32
6.166.3.32 const tFont g_sFontCm32b
6.166.3.33 const tFont g_sFontCm32i
6.166.3.34 const tFont g_sFontCm34
6.166.3.35 const tFont g_sFontCm34b
6.166.3.36 const tFont g_sFontCm34i
6.166.3.37 const tFont g_sFontCm36
6.166.3.38 const tFont g_sFontCm36b
6.166.3.39 const tFont g_sFontCm36i
6.166.3.40 const tFont g_sFontCm38
6.166.3.41 const tFont g_sFontCm38b
6.166.3.42 const tFont g_sFontCm38i
6.166.3.43 const tFont g_sFontCm40
6.166.3.44 const tFont g_sFontCm40b
6.166.3.45 const tFont g_sFontCm40i
6.166.3.46 const tFont g_sFontCm42
6.166.3.47 const tFont g_sFontCm42b
6.166.3.48 const tFont g_sFontCm42i
6.166.3.49 const tFont g_sFontCm44
6.166.3.50 const tFont g_sFontCm44b
6.166.3.51 const tFont g_sFontCm44i
6.166.3.52 const tFont g_sFontCm46
6.166.3.53 const tFont g_sFontCm46b
6.166.3.54 const tFont g_sFontCm46i

6.166.3.55 const tFont g_sFontCm48
6.166.3.56 const tFont g_sFontCm48b
6.166.3.57 const tFont g_sFontCm48i
6.166.3.58 const tFont g_sFontCmsc12
6.166.3.59 const tFont g_sFontCmsc14
6.166.3.60 const tFont g_sFontCmsc16
6.166.3.61 const tFont g_sFontCmsc18
6.166.3.62 const tFont g_sFontCmsc20
6.166.3.63 const tFont g_sFontCmsc22
6.166.3.64 const tFont g_sFontCmsc24
6.166.3.65 const tFont g_sFontCmsc26
6.166.3.66 const tFont g_sFontCmsc28
6.166.3.67 const tFont g_sFontCmsc30
6.166.3.68 const tFont g_sFontCmsc32
6.166.3.69 const tFont g_sFontCmsc34
6.166.3.70 const tFont g_sFontCmsc36
6.166.3.71 const tFont g_sFontCmsc38
6.166.3.72 const tFont g_sFontCmsc40
6.166.3.73 const tFont g_sFontCmsc42
6.166.3.74 const tFont g_sFontCmsc44
6.166.3.75 const tFont g_sFontCmsc46
6.166.3.76 const tFont g_sFontCmsc48
6.166.3.77 const tFont g_sFontCmss12
6.166.3.78 const tFont g_sFontCmss12b

6.166.3.79 const tFont g_sFontCmss12i
6.166.3.80 const tFont g_sFontCmss14
6.166.3.81 const tFont g_sFontCmss14b
6.166.3.82 const tFont g_sFontCmss14i
6.166.3.83 const tFont g_sFontCmss16
6.166.3.84 const tFont g_sFontCmss16b
6.166.3.85 const tFont g_sFontCmss16i
6.166.3.86 const tFont g_sFontCmss18
6.166.3.87 const tFont g_sFontCmss18b
6.166.3.88 const tFont g_sFontCmss18i
6.166.3.89 const tFont g_sFontCmss20
6.166.3.90 const tFont g_sFontCmss20b
6.166.3.91 const tFont g_sFontCmss20i
6.166.3.92 const tFont g_sFontCmss22
6.166.3.93 const tFont g_sFontCmss22b
6.166.3.94 const tFont g_sFontCmss22i
6.166.3.95 const tFont g_sFontCmss24
6.166.3.96 const tFont g_sFontCmss24b
6.166.3.97 const tFont g_sFontCmss24i
6.166.3.98 const tFont g_sFontCmss26
6.166.3.99 const tFont g_sFontCmss26b
6.166.3.100 const tFont g_sFontCmss26i
6.166.3.101 const tFont g_sFontCmss28
6.166.3.102 const tFont g_sFontCmss28b

6.166.3.103 const tFont g_sFontCmss28i
6.166.3.104 const tFont g_sFontCmss30
6.166.3.105 const tFont g_sFontCmss30b
6.166.3.106 const tFont g_sFontCmss30i
6.166.3.107 const tFont g_sFontCmss32
6.166.3.108 const tFont g_sFontCmss32b
6.166.3.109 const tFont g_sFontCmss32i
6.166.3.110 const tFont g_sFontCmss34
6.166.3.111 const tFont g_sFontCmss34b
6.166.3.112 const tFont g_sFontCmss34i
6.166.3.113 const tFont g_sFontCmss36
6.166.3.114 const tFont g_sFontCmss36b
6.166.3.115 const tFont g_sFontCmss36i
6.166.3.116 const tFont g_sFontCmss38
6.166.3.117 const tFont g_sFontCmss38b
6.166.3.118 const tFont g_sFontCmss38i
6.166.3.119 const tFont g_sFontCmss40
6.166.3.120 const tFont g_sFontCmss40b
6.166.3.121 const tFont g_sFontCmss40i
6.166.3.122 const tFont g_sFontCmss42
6.166.3.123 const tFont g_sFontCmss42b
6.166.3.124 const tFont g_sFontCmss42i
6.166.3.125 const tFont g_sFontCmss44
6.166.3.126 const tFont g_sFontCmss44b

6.166.3.127 const tFont g_sFontCmss44i
6.166.3.128 const tFont g_sFontCmss46
6.166.3.129 const tFont g_sFontCmss46b
6.166.3.130 const tFont g_sFontCmss46i
6.166.3.131 const tFont g_sFontCmss48
6.166.3.132 const tFont g_sFontCmss48b
6.166.3.133 const tFont g_sFontCmss48i
6.166.3.134 const tFont g_sFontFixed6x8

6.167 include/imgbutton.h File Reference

Data Structures

- struct [tImageButtonWidget](#)
The structure that describes a image button widget.

Defines

- #define [IB_STYLE_FILL](#) 0x0002
This flag indicates that the image button should be filled.
- #define [IB_STYLE_TEXT](#) 0x0004
This flag indicates that the image button should have text drawn on it.
- #define [IB_STYLE_IMAGE_OFF](#) 0x0008
This flag indicates that the background image is to be disabled.
- #define [IB_STYLE_KEYCAP_OFF](#) 0x0010
This flag indicates that the keycap image is to be disabled.
- #define [IB_STYLE_AUTO_REPEAT](#) 0x0020
- #define [IB_STYLE_PRESSED](#) 0x0040
This flag indicates that the image button is pressed.
- #define [IB_STYLE_RELEASE_NOTIFY](#) 0x0080
- #define [ImageButtonStruct](#)(pParent, pNext, pChild, pDisplay, IX, IY,IWidth, IHeight, ulStyle, ulForeColor,ulPressColor, ulBackColor, pFont, pcText, puclImage, pucPressImage, pucKeycapImage, sXOff, sYOff, usAutoRepeatDelay, usAutoRepeatRate, pfnOnClick)
- #define [ImageButton](#)(sName, pParent, pNext, pChild, pDisplay, IX, IY,IWidth, IHeight, ulStyle, ulForeColor, ulPressColor,ulBackColor, pFont, pcText, puclImage, pucPressImage, pucKeycapImage, sXOff, sYOff, usAutoRepeatDelay, usAutoRepeatRate, pfnOnClick)

- #define ImageButtonAutoRepeatDelaySet(pWidget, usDelay)
- #define ImageButtonAutoRepeatOff(pWidget)
- #define ImageButtonAutoRepeatOn(pWidget)
- #define ImageButtonAutoRepeatRateSet(pWidget, usRate)
- #define ImageButtonCallbackSet(pWidget, pfnOnClik)
- #define ImageButtonFillColorSet(pWidget, ulColor)
- #define ImageButtonFillOff(pWidget)
- #define ImageButtonFillOn(pWidget)
- #define ImageButtonImageOn(pWidget)
- #define ImageButtonImageOff(pWidget)
- #define ImageButtonKeycapOn(pWidget)
- #define ImageButtonKeycapOff(pWidget)
- #define ImageButtonImageSet(pWidget, plImg)
- #define ImageButtonImagePressedSet(pWidget, plImg)
- #define ImageButtonImageKeycapSet(pWidget, plImg)
- #define ImageButtonKeycapOffsetSet(pWidget, sX, sY)
- #define ImageButtonBackgroundColorSet(pWidget, ulColor)
- #define ImageButtonForegroundColorSet(pWidget, ulColor)
- #define ImageButtonPressedColorSet(pWidget, ulColor)
- #define ImageButtonTextSet(pWidget, pcTxt)
- #define ImageButtonTextOff(pWidget)
- #define ImageButtonTextOn(pWidget)

Functions

- int ImageButtonMsgProc (tWidget *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void ImageButtonInit (tImageButtonWidget *pWidget, const tDisplay *pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

6.167.1 Define Documentation

6.167.1.1 #define IB_STYLE_AUTO_REPEAT 0x0020

This flag indicates that the image button should auto-repeat, generating repeated click events while it is pressed.

6.167.1.2 #define IB_STYLE_FILL 0x0002

This flag indicates that the image button should be filled.

6.167.1.3 #define IB_STYLE_IMAGE_OFF 0x0008

This flag indicates that the background image is to be disabled.

6.167.1.4 #define IB_STYLE_KEYCAP_OFF 0x0010

This flag indicates that the keycap image is to be disabled.

6.167.1.5 #define IB_STYLE_PRESSED 0x0040

This flag indicates that the image button is pressed.

6.167.1.6 #define IB_STYLE_RELEASE_NOTIFY 0x0080

This flag indicates that the image button callback should be made when the button is released rather than when it is pressed. This does not affect the operation of auto repeat buttons.

6.167.1.7 #define IB_STYLE_TEXT 0x0004

This flag indicates that the image button should have text drawn on it.

6.167.1.8 #define ImageButton(sName, pParent, pNext, pChild, pDisplay, IX, IY, lWidth, lHeight, ulStyle, ulForeColor, ulPressColor, ulBackColor, pFont, pcText, pucImage, pucPressImage, pucKeycapImage, sXOff, sYOff, usAutoRepeatDelay, usAutoRepeatRate, pfnOnClick)
Value:

```
tImageButtonWidget sName =
    ImageButtonStruct(pParent, pNext, pChild, pDisplay, lX, lY,
                      lWidth, lHeight, ulStyle, ulForeColor,
                      ulPressColor, ulBackColor, pFont, pcText,
                      pucImage, pucPressImage, pucKeycapImage,
                      sXOff, sYOff, usAutoRepeatDelay,
                      usAutoRepeatRate, pfnOnClick)
```

Declares an initialized variable containing a image button widget data structure.

Parameters

<i>sName</i>	is the name of the variable to be declared.
<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the upper left corner of the image button.
<i>IY</i>	is the Y coordinate of the upper left corner of the image button.
<i>lWidth</i>	is the width of the image button.
<i>lHeight</i>	is the height of the image button.
<i>ulStyle</i>	is the style to be applied to the image button.

<i>ulForeColor</i>	is the color to be used for foreground pixels when a 1bpp image is being drawn. It is ignored for all other image bit depths.
<i>ulPressColor</i>	is the color to be used for foreground pixels when the button is pressed and a 1bpp image is being drawn. It is ignored for all other image bit depths.
<i>ulBackColor</i>	is the color to be used for background pixels when the button is released and a 1bpp image is being drawn. It is ignored for all other image bit depths.
<i>pFont</i>	is a pointer to the font to be used to draw text on the button.
<i>pcText</i>	is a pointer to the text to draw on this button.
<i>puImage</i>	is a pointer to the image to draw on the background of this image button when it is in the released state.
<i>pucPress-Image</i>	is a pointer to the image to draw on the background of this image button when it is in the pressed state.
<i>pucKeycap-Image</i>	is a pointer to the image to draw as the keycap of the on top of the image button, on top of the background image.
<i>sXOff</i>	is the horizontal offset to apply when drawing the keycap image on the button when in the pressed state.
<i>sYOff</i>	is the vertical offset to apply when drawing the keycap image on the button when in the pressed state.
<i>usAuto-Repeat-Delay</i>	is the delay before starting auto-repeat.
<i>usAuto-RepeatRate</i>	is the rate at which auto-repeat events are generated.
<i>pfnOnClick</i>	is a pointer to the function that is called when the push button is pressed.

This macro provides an initialized image button widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

ulStyle is the logical OR of the following:

- **IB_STYLE_TEXT** to indicate that text should be drawn on the button.
- **IB_STYLE_FILL** to indicate that the background of the button should be filled with color.
- **IB_STYLE_KEYCAP_OFF** to indicate that the keycap image should not be drawn.
- **IB_STYLE_IMAGE_OFF** to indicate that the background image should not be drawn.
- **IB_STYLE_AUTO_REPEAT** to indicate that auto-repeat should be used.
- **IB_STYLE_RELEASE_NOTIFY** to indicate that the callback should be made when the button is released. If absent, the callback is called when the button is initially pressed.

Returns

Nothing; this is not a function.

6.167.1.9 #define ImageButtonAutoRepeatDelaySet(pWidget, usDelay)**Value:**

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget;      \
    pW->usAutoRepeatDelay = usDelay;      \
}                                \
while(0)                            \
```

Sets the auto-repeat delay for a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
<i>usDelay</i>	is the number of pointer events before auto-repeat starts.

This function sets the delay before auto-repeat begins. Unpredictable behavior will occur if this is called while the image button is pressed.

Returns

None.

6.167.1.10 #define ImageButtonAutoRepeatOff(pWidget)**Value:**

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget;      \
    pW->ulStyle &= ~(IB_STYLE_AUTO_REPEAT); \
}                                \
while(0)                            \
```

Disables auto-repeat for a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function disables the auto-repeat behavior of a image button.

Returns

None.

6.167.1.11 #define ImageButtonAutoRepeatOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget;   \
    pW->ulStyle |= IB_STYLE_AUTO_REPEAT; \
}                                \
while(0)
```

Enables auto-repeat for a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function enables the auto-repeat behavior of a image button. Unpredictable behavior will occur if this is called while the image button is pressed.

Returns

None.

6.167.1.12 #define ImageButtonAutoRepeatRateSet(*pWidget*, *usRate*)**Value:**

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget;   \
    pW->usAutoRepeatRate = usRate;    \
}                                \
while(0)
```

Sets the auto-repeat rate for a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
<i>usRate</i>	is the number of pointer events between auto-repeat events.

This function sets the rate at which auto-repeat events occur. Unpredictable behavior will occur if this is called while the image button is pressed.

Returns

None.

6.167.1.13 #define ImageButtonBackgroundColorSet(*pWidget*, *ulColor*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = (pWidget); \
    pW->ulBackgroundColor = (ulColor); \
}                                \
while(0)
```

Sets the color of background pixels when using 1bpp images.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>ulColor</i>	is the background color to use.

This function changes the color that is used to draw background pixels when a 1bpp image is rendered on the button and the button is in the released state. The value is ignored for all other image bit depths. The display is not updated until the next paint request.

Returns

None.

6.167.1.14 #define ImageButtonCallbackSet(*pWidget*, *pfnOnClik*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    pW->pfnOnClick = pfnOnClick; \
}                                \
while(0)
```

Sets the function to call when this image button widget is pressed.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
<i>pfnOnClick</i>	is a pointer to the function to call.

This function sets the function to be called when this image button is pressed. The supplied function is called when the image button is first pressed, and then repeated while the image button is pressed if auto-repeat is enabled.

Returns

None.

6.167.1.15 #define ImageButtonFillColorSet(pWidget, ulColor)**Value:**

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget;   \
    pW->ulBackgroundColor = ulColor;   \
}                                \
while(0)
```

Sets the fill color of a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the image button.

This function changes the color used to fill the background of the image button on the display. This is a duplicate of ImageButtonBackgroundColorSet which is left for backwards compatibility. The display is not updated until the next paint request.

Returns

None.

6.167.1.16 #define ImageButtonFillOff(pWidget)**Value:**

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget;   \
    pW->ulStyle &= ~(IB_STYLE_FILL);   \
}                                \
while(0)
```

Disables filling of a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function disables the filling of a image button widget. The display is not updated until the next paint request.

Returns

None.

6.167.1.17 #define ImageButtonFillOn(*pWidget*)

Value:

```
do                                \
{                               \
    tImageButtonWidget *pW = pWidget; \
    pW->ulStyle |= IB_STYLE_FILL; \
}                               \
while(0)
```

Enables filling of a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function enables the filling of a image button widget. The display is not updated until the next paint request.

Returns

None.

6.167.1.18 #define ImageButtonForegroundColorSet(*pWidget*, *ulColor*)

Value:

```
do                                \
{                               \
    tImageButtonWidget *pW = (pWidget); \
    pW->ulForegroundColor = (ulColor); \
}                               \
while(0)
```

Sets the color of foreground pixels when using 1bpp images.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>ulColor</i>	is the foreground color to use.

This function changes the color that is used to draw foreground pixels when a 1bpp image or text string is rendered on the button. The value is ignored for all other image bit depths. The display is not updated until the next paint request.

Returns

None.

6.167.1.19 #define ImageButtonImageKeycapSet(*pWidget*, *pImg*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = (pWidget); \
    const unsigned char *pI = (pImg);   \
    pW->pucKeycapImage = pI;        \
}                                \
while(0)
```

Changes the keycap image drawn on a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>pImg</i>	is a pointer to the image to draw onto the image button.

This function changes the image that is drawn onto the top of the push button. The display is not updated until the next paint request.

Returns

None.

6.167.1.20 #define ImageButtonImageOff(*pWidget*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    pW->ulStyle |= IB_STYLE_IMAGE_OFF; \
}                                \
while(0)
```

Disables the background image for an image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function disables the drawing of the background image on an image button widget. The display is not updated until the next paint request.

Returns

None.

6.167.1.21 #define ImageButtonImageOn(*pWidget*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    pW->ulStyle &= ~(IB_STYLE_IMAGE_OFF); \
}                                \
while(0)
```

Enables the background image for an image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function enables the drawing of the background image on an image button widget. The display is not updated until the next paint request.

Returns

None.

6.167.1.22 #define ImageButtonImagePressedSet(*pWidget*, *pImg*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    const unsigned char *pI = pImg;   \
    pW->pucPressImage = pI;        \
}                                \
while(0)
```

Changes the image drawn on a image button widget when it is pressed.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>pImg</i>	is a pointer to the image to draw onto the image button when it is pressed.

This function changes the image that is drawn onto the background of the image button in its pressed state. The display is not updated until the next paint request.

Returns

None.

6.167.1.23 #define ImageButtonImageSet(*pWidget*, *pImg*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    const unsigned char *pI = pImg;   \
    pW->pucImage = pI;             \
}                                \
while(0)
```

Changes the image drawn on a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>pImg</i>	is a pointer to the image to draw onto the image button.

This function changes the image that is drawn onto the background of the image button in its unpressed state. The display is not updated until the next paint request.

Returns

None.

6.167.1.24 #define ImageButtonKeycapOff(*pWidget*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    pW->ulStyle |= IB_STYLE_KEYCAP_OFF; \
}                                \
while(0)
```

Disables the keycap image for an image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function disables the drawing of the keycap image on an image button widget. The display is not updated until the next paint request.

Returns

None.

6.167.1.25 #define ImageButtonKeycapOffsetSet(*pWidget*, *sX*, *sY*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = (pWidget); \
    pW->sXOffset = (short) (sX);      \
    pW->sYOffset = (short) (sY);      \
}                                \
while (0)
```

Changes the keycap image offset position on an image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>sX</i>	is the signed horizontal position offset for the keycap image when the image button is pressed. Positive values move the image right.
<i>sY</i>	is the signed vertical position offset for the keycap image when the image button is pressed. Positive values move the image down.

This function changes the position that the keycap image is drawn at when the image button is pressed. The keycap image is moved *iX* pixels right and *iY* pixels down from the center position if the image button is pressed. This feature can be used to support 3D buttons. The display is not updated until the next paint request.

Returns

None.

6.167.1.26 #define ImageButtonKeycapOn(*pWidget*)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    pW->ulStyle &= ~(IB_STYLE_KEYCAP_OFF); \
}                                \
while (0)
```

Enables the keycap image for an image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function enables the drawing of the keycap image on an image button widget. The display is not updated until the next paint request.

Returns

None.

6.167.1.27 #define ImageButtonPressedColorSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                \
    tImageButtonWidget *pW = (pWidget);      \
    pW->ulPressedColor = (ulColor);        \
}                                \
while(0)
```

Sets the color of foreground pixels when the button is pressed and when using 1bpp images.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>ulColor</i>	is the pressed foreground color to use.

This function changes the color that is used to draw foreground pixels when a 1bpp image is rendered on the button and the button is in the pressed state. The value is ignored for all other image bit depths. The display is not updated until the next paint request.

Returns

None.

**6.167.1.28 #define ImageButtonStruct(*pParent*, *pNext*, *pChild*, *pDisplay*, *IX*, *IY*,
IWidth, *IHeight*, *ulStyle*, *ulForeColor*, *ulPressColor*, *ulBackColor*, *pFont*,
pcText, *puImage*, *pucPressImage*, *pucKeycapImage*, *sXOff*, *sYOff*,
usAutoRepeatDelay, *usAutoRepeatRate*, *pfnOnClick*)**

Declares an initialized image button widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the push button.
<i>IX</i>	is the X coordinate of the upper left corner of the image button.
<i>IY</i>	is the Y coordinate of the upper left corner of the image button.
<i>IWidth</i>	is the width of the image button.
<i>IHeight</i>	is the height of the image button.
<i>ulStyle</i>	is the style to be applied to the image button.
<i>ulForeColor</i>	is the color to be used for foreground pixels when a 1bpp image or text is being drawn. It is ignored for all other image bit depths.
<i>ulPressColor</i>	is the color to be used for foreground pixels when the button is pressed and a 1bpp image is being drawn. It is ignored for all other image bit depths.
<i>ulBackColor</i>	is the color to be used for background pixels when the button is released and a 1bpp image is being drawn. It is ignored for all other image bit depths.
<i>pFont</i>	is a pointer to the font to be used to draw text on the button.
<i>pcText</i>	is a pointer to the text to draw on this button.
<i>puclImage</i>	is a pointer to the image to draw on the background of this image button when it is in the released state.
<i>pucPress-Image</i>	is a pointer to the image to draw on the background of this image button when it is in the pressed state.
<i>pucKeycap-Image</i>	is a pointer to the image to draw as the keycap of the on top of the image button, on top of the background image.
<i>sXOff</i>	is the horizontal offset to apply when drawing the keycap image on the button when in the pressed state.
<i>sYOff</i>	is the vertical offset to apply when drawing the keycap image on the button when in the pressed state.
<i>usAuto-Repeat-Delay</i>	is the delay before starting auto-repeat.
<i>usAuto-RepeatRate</i>	is the rate at which auto-repeat events are generated.
<i>pfnOnClick</i>	is a pointer to the function that is called when the push button is pressed.

This macro provides an initialized image button widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
///!     tImageButtonWidget g_sImageButton = ImageButtonStruct(...);
///!
```

Or, in an array of variables:

```
///!     tImageButtonWidget g_psImageButtons[] =
///!     {
```

```
//!      ImageButtonStruct(...),
//!      ImageButtonStruct(...),
//!  };
//!
```

ulStyle is the logical OR of the following:

- **IB_STYLE_TEXT** to indicate that text should be drawn on the button.
- **IB_STYLE_FILL** to indicate that the background of the button should be filled with color.
- **IB_STYLE_KEYCAP_OFF** to indicate that the keycap image should not be drawn.
- **IB_STYLE_IMAGE_OFF** to indicate that the background image should not be drawn.
- **IB_STYLE_AUTO_REPEAT** to indicate that auto-repeat should be used.
- **IB_STYLE_RELEASE_NOTIFY** to indicate that the callback should be made when the button is released. If absent, the callback is called when the button is initially pressed.

Returns

Nothing; this is not a function.

6.167.1.29 #define ImageButtonTextOff(pWidget)

Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    pW->ulStyle &= ~(IB_STYLE_TEXT); \
}                                \
while(0)
```

Disables the text on a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function disables the drawing of text on a image button widget. The display is not updated until the next paint request.

Returns

None.

6.167.1.30 #define ImageButtonTextOn(*pWidget*)
Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    pW->ulStyle |= IB_STYLE_TEXT;   \
}                                \
while(0)
```

Enables the text on a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to modify.
----------------	--

This function enables the drawing of text on a image button widget. The display is not updated until the next paint request.

Returns

None.

6.167.1.31 #define ImageButtonTextSet(*pWidget*, *pcTxt*)
Value:

```
do                                \
{                                \
    tImageButtonWidget *pW = pWidget; \
    const char *pcT = pcTxt;        \
    pW->pcText = pcT;             \
}                                \
while(0)
```

Changes the text drawn on a image button widget.

Parameters

<i>pWidget</i>	is a pointer to the image button widget to be modified.
<i>pcTxt</i>	is a pointer to the text to draw onto the image button.

This function changes the text that is drawn onto the image button. The display is not updated until the next paint request.

Returns

None.

6.167.2 Function Documentation

6.167.2.1 `void ImageButtonInit (tImageButtonWidget * pWidget, const tDisplay * pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)`

Initializes an image button widget.

Parameters

<code>pWidget</code>	is a pointer to the image button widget to initialize.
<code>pDisplay</code>	is a pointer to the display on which to draw the push button.
<code>IX</code>	is the X coordinate of the upper left corner of the image button.
<code>IY</code>	is the Y coordinate of the upper left corner of the image button.
<code>IWidth</code>	is the width of the image button.
<code>IHeight</code>	is the height of the image button.

This function initializes the provided image button widget.

Returns

None.

6.167.2.2 `int ImageButtonMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)`

Handles messages for an image button widget.

Parameters

<code>pWidget</code>	is a pointer to the image button widget.
<code>ulMsg</code>	is the message.
<code>ulParam1</code>	is the first parameter to the message.
<code>ulParam2</code>	is the second parameter to the message.

This function receives messages intended for this image button widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.168 include/listbox.h File Reference

Data Structures

- struct [tListBoxWidget](#)

The structure that describes a listbox widget.

Defines

- #define LISTBOX_STYLE_OUTLINE 0x0001
- #define LISTBOX_STYLE_LOCKED 0x0002
- #define LISTBOX_STYLE_WRAP 0x0004
- #define [ListBoxStruct](#)(pParent, pNext, pChild, pDisplay, IX, IY, IWidth,IHeight, ulStyle, ulBgColor, ulSelBgColor,ulTextColor, ulSelTextColor, ulOutlineColor, p-Font,ppcText, usMaxEntries, usPopulatedEntries,pfnOnChange)
- #define [ListBox](#)(sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth,IHeight, ulStyle, ulBgColor, ulSelBgColor, ulTextColor,ulSelTextColor, ulOutlineColor, p-Font, ppcText, usMaxEntries, usPopulatedEntries, pfnOnChange)
- #define [ListBoxCallbackSet](#)(pWidget, pfnCallback)
- #define [ListBoxBackgroundColorSet](#)(pWidget, ulColor)
- #define [ListBoxSelectedBackgroundColorSet](#)(pWidget, ulColor)
- #define [ListBoxFontSet](#)(pWidget, pFnt)
- #define [ListBoxOutlineColorSet](#)(pWidget, ulColor)
- #define [ListBoxOutlineOff](#)(pWidget)
- #define [ListBoxOutlineOn](#)(pWidget)
- #define [ListBoxTextColorSet](#)(pWidget, ulColor)
- #define [ListBoxSelectedTextColorSet](#)(pWidget, ulColor)
- #define [ListBoxTextSet](#)(pWidget, pcTxt, ullIndex)
- #define [ListBoxLock](#)(pWidget)
- #define [ListBoxUnlock](#)(pWidget)
- #define [ListBoxWrapEnable](#)(pWidget)
- #define [ListBoxWrapDisable](#)(pWidget)
- #define [ListBoxClear](#)(pWidget)
- #define [ListBoxSelectionSet](#)(pWidget, sSel)
- #define [ListBoxSelectionGet](#)(pWidget) ((([tListBoxWidget](#) *) (pWidget))->s-Selected)

Functions

- int [ListBoxMsgProc](#) ([tWidget](#) *pWidget, unsigned ulMsg, unsigned ulParam1, un- signed ulParam2)
- void [ListBoxInit](#) ([tListBoxWidget](#) *pWidget, const [tDisplay](#) *pDisplay, const char **ppcText, unsigned short usMaxEntries, unsigned short usPopulatedEntries, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)
- int [ListBoxTextAdd](#) ([tListBoxWidget](#) *pWidget, const char *pcTxt)

6.169 include/pushbutton.h File Reference

Data Structures

- struct [tPushButtonWidget](#)

The structure that describes a push button widget.

Defines

- `#define PB_STYLE_OUTLINE 0x0001`
This flag indicates that the push button should be outlined.
- `#define PB_STYLE_FILL 0x0002`
This flag indicates that the push button should be filled.
- `#define PB_STYLE_TEXT 0x0004`
This flag indicates that the push button should have text drawn on it.
- `#define PB_STYLE_IMG 0x0008`
This flag indicates that the push button should have an image drawn on it.
- `#define PB_STYLE_TEXT_OPAQUE 0x0010`
- `#define PB_STYLE_AUTO_REPEAT 0x0020`
- `#define PB_STYLE_PRESSED 0x0040`
This flag indicates that the push button is pressed.
- `#define PB_STYLE_RELEASE_NOTIFY 0x0080`
- `#define CircularButtonStruct(pParent, pNext, pChild, pDisplay, IX, IY, IR, ulStyle, ulFillColor, ulPressFillColor,ulOutlineColor, ulTextColor, pFont, pcText,puclImage, pucPressImage, usAutoRepeatDelay,usAutoRepeatRate, pfnOnClick)`
- `#define CircularButton(sName, pParent, pNext, pChild, pDisplay, IX, IY, IR,ulStyle, ulFillColor, ulPressFillColor,ulOutlineColor, ulTextColor, pFont, pcText, puclImage, pucPressImage, usAutoRepeatDelay, usAutoRepeatRate,pfnOnClick)`
- `#define RectangularButtonStruct(pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, ulStyle, ulFillColor,ulPressFillColor, ulOutlineColor,ulTextColor, pFont, pcText, puclImage, pucPressImage, usAutoRepeatDelay,usAutoRepeatRate, pfnOnClick)`
- `#define RectangularButton(sName, pParent, pNext, pChild, pDisplay, IX, IY,I-Width, IHeight, ulStyle, ulFillColor,ulPressFillColor, ulOutlineColor,ulTextColor, p-Font, pcText, puclImage, pucPressImage,usAutoRepeatDelay, usAutoRepeatRate, usAutoRepeatRate, pfnOnClick)`
- `#define PushButtonAutoRepeatDelaySet(pWidget, usDelay)`
- `#define PushButtonAutoRepeatOff(pWidget)`
- `#define PushButtonAutoRepeatOn(pWidget)`
- `#define PushButtonAutoRepeatRateSet(pWidget, usRate)`
- `#define PushButtonCallbackSet(pWidget, pfnOnClick)`
- `#define PushButtonFillColorSet(pWidget, ulColor)`
- `#define PushButtonFillColorPressedSet(pWidget, ulColor)`
- `#define PushButtonFillOff(pWidget)`
- `#define PushButtonFillOn(pWidget)`
- `#define PushButtonFontSet(pWidget, pFnt)`
- `#define PushButtonImageSet(pWidget, plImg)`
- `#define PushButtonImagePressedSet(pWidget, plImg)`
- `#define PushButtonImageOff(pWidget)`
- `#define PushButtonImageOn(pWidget)`
- `#define PushButtonOutlineColorSet(pWidget, ulColor)`
- `#define PushButtonOutlineOff(pWidget)`
- `#define PushButtonOutlineOn(pWidget)`

- #define PushButtonTextColorSet(pWidget, ulColor)
- #define PushButtonTextOff(pWidget)
- #define PushButtonTextOn(pWidget)
- #define PushButtonTextOpaqueOff(pWidget)
- #define PushButtonTextOpaqueOn(pWidget)
- #define PushButtonTextSet(pWidget, pcTxt)

Functions

- int `RectangularButtonMsgProc` (tWidget *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void `RectangularButtonInit` (tPushButtonWidget *pWidget, const tDisplay *pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)
- int `CircularButtonMsgProc` (tWidget *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void `CircularButtonInit` (tPushButtonWidget *pWidget, const tDisplay *pDisplay, unsigned IX, unsigned IY, unsigned IR)

6.170 include/radiobutton.h File Reference

Data Structures

- struct `tRadioButtonWidget`
The structure that describes a radio button widget.

Defines

- #define `RB_STYLE_OUTLINE` 0x0001
This flag indicates that the radio button should be outlined.
- #define `RB_STYLE_FILL` 0x0002
This flag indicates that the radio button should be filled.
- #define `RB_STYLE_TEXT` 0x0004
This flag indicates that the radio button should have text drawn on it.
- #define `RB_STYLE_IMG` 0x0008
This flag indicates that the radio button should have an image drawn on it.
- #define `RB_STYLE_TEXT_OPAQUE` 0x0010
- #define `RB_STYLE_SELECTED` 0x0020
This flag indicates that the radio button is selected.
- #define `RadioButtonStruct`(pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, usStyle, usCircleSize, ulFillColor,ulOutlineColor, ulTextColor, pFont, pcText,puImage, pfnOnChange)
- #define `RadioButton`(sName, pParent, pNext, pChild, pDisplay, IX, IY, IWidth, IHeight, usStyle, usCircleSize, ulFillColor,ulOutlineColor, ulTextColor, pFont, pcText, puImage,pfnOnChange)

- #define RadioButtonCircleSizeSet(pWidget, usSize)
- #define RadioButtonCallbackSet(pWidget, pfnOnChg)
- #define RadioButtonFillColorSet(pWidget, ulColor)
- #define RadioButtonFillOff(pWidget)
- #define RadioButtonFillOn(pWidget)
- #define RadioButtonFontSet(pWidget, pFnt)
- #define RadioButtonImageSet(pWidget, plImg)
- #define RadioButtonImageOff(pWidget)
- #define RadioButtonImageOn(pWidget)
- #define RadioButtonOutlineColorSet(pWidget, ulColor)
- #define RadioButtonOutlineOff(pWidget)
- #define RadioButtonOutlineOn(pWidget)
- #define RadioButtonTextColorSet(pWidget, ulColor)
- #define RadioButtonTextOff(pWidget)
- #define RadioButtonTextOn(pWidget)
- #define RadioButtonTextOpaqueOff(pWidget)
- #define RadioButtonTextOpaqueOn(pWidget)
- #define RadioButtonTextSet(pWidget, pcTxt)

Functions

- int RadioButtonMsgProc (tWidget *pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)
- void RadioButtonInit (tRadioButtonWidget *pWidget, const tDisplay *pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

6.171 include/sed1335.h File Reference

Defines

- #define SED1335_SYSTEM_SET 0x40
- #define SED1335_SLEEP_IN 0x53
- #define SED1335_DISP_OFF 0x58
- #define SED1335_DISP_ON 0x59
- #define SED1335_SCROLL 0x44
- #define SED1335_CSRFORM 0x5d
- #define SED1335_CGRAM_ADR 0x5c
- #define SED1335_CSRDIR_U 0x4e
- #define SED1335_CSRDIR_D 0x4f
- #define SED1335_CSRDIR_L 0x4d
- #define SED1335_CSRDIR_R 0x4c
- #define SED1335_HDOT_SCR 0x5a
- #define SED1335_OVLAY 0x5b
- #define SED1335_CSRW 0x46
- #define SED1335_CSRR 0x47

```
• #define SED1335_MWRITE 0x42
• #define SED1335_MREAD 0x43
• #define SED1335_SCR_WIDTH 319
• #define SED1335_M0 0
• #define SED1335_M1 0
• #define SED1335_M2 0
• #define SED1335_WS 0
• #define SED1335_IV 1
• #define SED1335_FX 7
• #define SED1335_FY 7
• #define SED1335_WF 1
• #define SED1335_CR 39
• #define SED1335_TCR 43
• #define SED1335_LF 239
• #define SED1335_APL 40
• #define SED1335_APB 0
• #define SED1335_LINES 30
• #define SED1335_SAD1L 0
• #define SED1335_SAD1H 0
• #define SED1335_SL1 0xEF
• #define SED1335_SAD2L 0xB0
• #define SED1335_SAD2H 0x04
• #define SED1335_SL2 0xEF
• #define SED1335_SAD3L 0
• #define SED1335_SAD3H 0
• #define SED1335_SAD4L 0
• #define SED1335_SAD4H 0
• #define SED1335_CRX 0x04
• #define SED1335_CRY 0x07
• #define SED1335_CM 0
• #define SED1335_MX0 1
• #define SED1335_MX1 0
• #define SED1335_DM1 0
• #define SED1335_DM2 0
• #define SED1335_OV 0
• #define SED1335_SAGL 0
• #define SED1335_SAGH 0x70
• #define SED1335_SCRD 0
• #define SED1335_FLASH 0x16
• #define SED1335_TEXTSIZE ((SED1335_SAD2H << 8) + SED1335_SAD2L)
• #define SED1335_GRAPHICSTART ((SED1335_SAD2H << 8) + SED1335_SAD2L)
• #define SED1335_GRAPHICSIZE ((SED1335_SL2+1) * (SED1335_SCR_WIDTH+1))>>3
• #define SED1335_MEM_END 10800
• #define SED1335_SYS_P1 0x10 | (SED1335_IV << 5) | (SED1335_WS << 3)
| (SED1335_M2 << 2) | (SED1335_M1 << 1) | SED1335_M0
```

- #define **SED1335_SYS_P2** 0x00 | (SED1335_WF << 7) | SED1335_FX
- #define **SED1335_CSRF_P2** 0x00 | (SED1335_CM << 7) | SED1335_CRY
- #define **SED1335_OVLAY_P1** 0x00 | (SED1335_OV << 4) | (SED1335_DM2 << 3) | (SED1335_DM1 << 2) | (SED1335_MX1 << 1) | SED1335_MX0

Functions

- void **GLCD_Initialize** (void)
- void **GLCD_WriteCommand** (unsigned char)
- void **GLCD_WriteData** (unsigned char)
- unsigned char **GLCD_ReadData** (void)
- char **GLCD_ReadByteFromROMMemory** (char *)
- void **GLCD_ClearText** (void)
- void **GLCD_ClearGraphic** (void)
- void **GLCD_TextGoTo** (unsigned char, unsigned char)
- void **GLCD_WriteText** (char *)
- void **GLCD_SetPixel** (unsigned int x, unsigned int y, int **color**)
- void **GLCD_SetCursorAddress** (unsigned int address)
- void **GLCD_Bitmap** (char *bmp, int x, int y, int dx, int dy)

6.171.1 Define Documentation

6.171.1.1 #define **SED1335_APY** 0

6.171.1.2 #define **SED1335_APL** 40

6.171.1.3 #define **SED1335_CGRAM_ADR** 0x5c

6.171.1.4 #define **SED1335_CM** 0

6.171.1.5 #define **SED1335_CR** 39

6.171.1.6 #define **SED1335_CRX** 0x04

6.171.1.7 #define **SED1335_CRY** 0x07

6.171.1.8 #define **SED1335_CSRDIR_D** 0x4f

6.171.1.9 #define **SED1335_CSRDIR_L** 0x4d

6.171.1.10 #define **SED1335_CSRDIR_R** 0x4c

6.171.1.11 #define **SED1335_CSRDIR_U** 0x4e

6.171.1.12 #define **SED1335_CSRF_P2** 0x00 | (SED1335_CM << 7) | SED1335_CRY

```
6.171.1.13 #define SED1335_CSRFORM 0x5d
6.171.1.14 #define SED1335_CSRR 0x47
6.171.1.15 #define SED1335_CSRW 0x46
6.171.1.16 #define SED1335_DISP_OFF 0x58
6.171.1.17 #define SED1335_DISP_ON 0x59
6.171.1.18 #define SED1335_DM1 0
6.171.1.19 #define SED1335_DM2 0
6.171.1.20 #define SED1335_FLASH 0x16
6.171.1.21 #define SED1335_FX 7
6.171.1.22 #define SED1335_FY 7
6.171.1.23 #define SED1335_GRAPHICSIZE ((SED1335_SL2+1) *
(SED1335_SCR_WIDTH+1))>>3
6.171.1.24 #define SED1335_GRAPHICSTART ((SED1335_SAD2H << 8) +
SED1335_SAD2L)
6.171.1.25 #define SED1335_HDOT_SCR 0x5a
6.171.1.26 #define SED1335_IV 1
6.171.1.27 #define SED1335_LF 239
6.171.1.28 #define SED1335_LINES 30
6.171.1.29 #define SED1335_M0 0
6.171.1.30 #define SED1335_M1 0
6.171.1.31 #define SED1335_M2 0
6.171.1.32 #define SED1335_MEM_END 10800
6.171.1.33 #define SED1335_MREAD 0x43
6.171.1.34 #define SED1335_MWRITE 0x42
6.171.1.35 #define SED1335_MX0 1
```

```
6.171.1.36 #define SED1335_MX1 0
6.171.1.37 #define SED1335_OV 0
6.171.1.38 #define SED1335_OVLAY 0x5b
6.171.1.39 #define SED1335_OVLAY_P1 0x00 | (SED1335_OV << 4) | (SED1335_DM2 <<
3) | (SED1335_DM1 << 2) | (SED1335_MX1 << 1) | SED1335_MX0
6.171.1.40 #define SED1335_SAD1H 0
6.171.1.41 #define SED1335_SAD1L 0
6.171.1.42 #define SED1335_SAD2H 0x04
6.171.1.43 #define SED1335_SAD2L 0xB0
6.171.1.44 #define SED1335_SAD3H 0
6.171.1.45 #define SED1335_SAD3L 0
6.171.1.46 #define SED1335_SAD4H 0
6.171.1.47 #define SED1335_SAD4L 0
6.171.1.48 #define SED1335_SAGH 0x70
6.171.1.49 #define SED1335_SAGL 0
6.171.1.50 #define SED1335_SCR_WIDTH 319
6.171.1.51 #define SED1335_SCRD 0
6.171.1.52 #define SED1335_SCROLL 0x44
6.171.1.53 #define SED1335_SL1 0xEF
6.171.1.54 #define SED1335_SL2 0xEF
6.171.1.55 #define SED1335_SLEEP_IN 0x53
6.171.1.56 #define SED1335_SYS_P1 0x10 | (SED1335_IV << 5) | (SED1335_WS << 3) |
(SED1335_M2 << 2) | (SED1335_M1 << 1) | SED1335_M0
6.171.1.57 #define SED1335_SYS_P2 0x00 | (SED1335_WF << 7) | SED1335_FX
6.171.1.58 #define SED1335_SYSTEM_SET 0x40
```

```
6.171.1.59 #define SED1335_TCR 43  
6.171.1.60 #define SED1335_TEXTSIZE ((SED1335_SAD2H << 8) +  
    SED1335_SAD2L)  
6.171.1.61 #define SED1335_WF 1  
6.171.1.62 #define SED1335_WS 0
```

6.171.2 Function Documentation

6.171.2.1 void **GLCD_Bitmap** (char * *bmp*, int *x*, int *y*, int *width*, int *height*)

Draw a bit

Parameters

<i>bmp</i>	is the pointer that contains the values of the Bitmap
<i>x</i>	is the position where the bitmap will start in X
<i>y</i>	is the position where the bitmap will start in Y.
<i>width</i>	is the width of the Bitmap
<i>height</i>	is the height of the Bitmap

This function draw a Bitmap on GLCD, starting from (X,Y)

Returns

None.

6.171.2.2 void **GLCD_ClearGraphic** (void)

Clean the screen

This function clean the screen of the GLCD for graphics.

Returns

None.

6.171.2.3 void **GLCD_ClearText** (void)

Clean the screen

This function clean the screen of the GLCD for texts.

Returns

None.

6.171.2.4 void GLCD_Initialize(void)

Initialize the GLCD

This function initialize the ports and start up the GLCD with the default options.

Returns

None.

6.171.2.5 char GLCD_ReadByteFromROMMemory(char *)**6.171.2.6 unsigned char GLCD_ReadData(void)****6.171.2.7 void GLCD_SetCursorAddress(unsigned int address)**

Set the cursor on a address of the ROM

Parameters

<i>address</i>	is the local where the cursor will point.
----------------	---

This function addresses the GLCD's cursor on a specific address.

Returns

None.

6.171.2.8 void GLCD_SetPixel(unsigned int x, unsigned int y, int color)

Draws a pixel point

Parameters

<i>x</i>	is the position where the point will be print in X.
<i>y</i>	is the position where the point will be print in Y.
<i>color</i>	is the color of the pixel printed in (X,Y).

This function draws pixel point in position (X,Y) with the color chosen.

Returns

None.

6.171.2.9 void GLCD_TextGoTo(unsigned char x, unsigned char y)

Set the text's cursor on a address

Parameters

<i>x</i>	is the position where the cursor will be send in X.
<i>y</i>	is the position where the cursor will be send in Y.

This function addresses the GLCD's text cursor on a specific address.

Returns

None.

6.171.2.10 void GLCD_WriteCommand (unsigned char)

6.171.2.11 void GLCD_WriteData (unsigned char)

6.171.2.12 void GLCD_WriteText (char * *tekst*)

Print a message on GLCD

Parameters

<i>tekst</i>	is a string that will be send to the GLCD.
--------------	--

This function prints a message on GLCD.

Returns

None.

6.172 include/slider.h File Reference

Data Structures

- struct [tSliderWidget](#)

The structure that describes a slider widget.

Defines

- #define [SL_STYLE_OUTLINE](#) 0x0001

This flag indicates that the slider should be outlined.

- #define [SL_STYLE_FILL](#) 0x0002

This flag indicates that the active portion of the slider should be filled.

- #define [SL_STYLE_BACKG_FILL](#) 0x0004

- #define [SL_STYLE_TEXT](#) 0x0008

- #define [SL_STYLE_BACKG_TEXT](#) 0x0010

- #define [SL_STYLE_IMG](#) 0x0020

This flag indicates that the slider should have an image drawn on it.

- #define `SL_STYLE_BACKG_IMG` 0x0040
- #define `SL_STYLE_TEXT_OPAQUE` 0x0080
- #define `SL_STYLE_BACKG_TEXT_OPAQUE` 0x0100
- #define `SL_STYLE_VERTICAL` 0x0200
- #define `SL_STYLE_LOCKED` 0x0400
- #define `SliderStruct`(pParent, pNext, pChild, pDisplay, IX, IY, IWidth,IHeight, IMin, IMax, IValue, ulStyle, ulFillColor,ulBackgroundFillColor, ulOutlineColor, ulTextColor,ulBackgroundTextColor, pFont, pcText, puImage,pucBackgroundImage, pfnOnChange)
- #define `Slider`(sName, pParent, pNext, pChild, pDisplay, IX, IY,IWidth, IHeight, IMin, IMax, IValue, ulStyle,ulFillColor, ulBackgroundFillColor, ulOutlineColor,ulTextColor, ulBackgroundTextColor, pFont, pcText,puImage, pucBackgroundImage, pfnOnChange)
- #define `SliderCallbackSet`(pWidget, pfnCallback)
- #define `SliderFillColorSet`(pWidget, ulColor)
- #define `SliderFillColorBackgroundedSet`(pWidget, ulColor)
- #define `SliderFillOff`(pWidget)
- #define `SliderFillOn`(pWidget)
- #define `SliderFontSet`(pWidget, pFnt)
- #define `SliderImageSet`(pWidget, plImg)
- #define `SliderBackgroundImageSet`(pWidget, plImg)
- #define `SliderImageOff`(pWidget)
- #define `SliderImageOn`(pWidget)
- #define `SliderBackgroundImageOff`(pWidget)
- #define `SliderBackgroundImageOn`(pWidget)
- #define `SliderOutlineColorSet`(pWidget, ulColor)
- #define `SliderOutlineOff`(pWidget)
- #define `SliderOutlineOn`(pWidget)
- #define `SliderTextColorSet`(pWidget, ulColor)
- #define `SliderBackgroundTextColorSet`(pWidget, ulColor)
- #define `SliderTextOff`(pWidget)
- #define `SliderTextOn`(pWidget)
- #define `SliderTextOpaqueOff`(pWidget)
- #define `SliderTextOpaqueOn`(pWidget)
- #define `SliderBackgroundTextOff`(pWidget)
- #define `SliderBackgroundTextOn`(pWidget)
- #define `SliderBackgroundTextOpaqueOff`(pWidget)
- #define `SliderBackgroundTextOpaqueOn`(pWidget)
- #define `SliderLock`(pWidget)
- #define `SliderUnlock`(pWidget)
- #define `SliderTextSet`(pWidget, pcTxt)
- #define `SliderRangeSet`(pWidget, IMinimum, IMaximum)
- #define `SliderValueSet`(pWidget, IVal)

Functions

- int `SliderMsgProc` (`tWidget` *`pWidget`, unsigned `ulMsg`, unsigned `ulParam1`, unsigned `ulParam2`)
- void `SliderInit` (`tSliderWidget` *`pWidget`, const `tDisplay` *`pDisplay`, unsigned `IX`, unsigned `IY`, unsigned `IWidth`, unsigned `IHeight`)

6.172.1 Define Documentation

6.172.1.1 `#define SL_STYLE_BACKG_FILL 0x0004`

This flag indicates that the background portion of the slider should be filled.

6.172.1.2 `#define SL_STYLE_BACKG_IMG 0x0040`

This flag indicates that the slider should have an image drawn on its background.

6.172.1.3 `#define SL_STYLE_BACKG_TEXT 0x0010`

This flag indicates that the slider should have text drawn on top of the background portion.

6.172.1.4 `#define SL_STYLE_BACKG_TEXT_OPAQUE 0x0100`

This flag indicates that the slider text should be drawn opaque (in other words, drawing the background pixels as well as the foreground pixels) in the background portion of the slider.

6.172.1.5 `#define SL_STYLE_FILL 0x0002`

This flag indicates that the active portion of the slider should be filled.

6.172.1.6 `#define SL_STYLE_IMG 0x0020`

This flag indicates that the slider should have an image drawn on it.

6.172.1.7 `#define SL_STYLE_LOCKED 0x0400`

This flag causes the slider to ignore pointer input and act as a passive indicator. An application may set its value and repaint it as normal but its value will not be changed in response to any touchscreen activity.

6.172.1.8 #define SL_STYLE_OUTLINE 0x0001

This flag indicates that the slider should be outlined.

6.172.1.9 #define SL_STYLE_TEXT 0x0008

This flag indicates that the slider should have text drawn on top of the active portion.

6.172.1.10 #define SL_STYLE_TEXT_OPAQUE 0x0080

This flag indicates that the slider text should be drawn opaque (in other words, drawing the background pixels as well as the foreground pixels) in the active portion of the slider.

6.172.1.11 #define SL_STYLE_VERTICAL 0x0200

This flag indicates that the slider is vertical rather than horizontal. If the flag is absent, the slider is assumed to operate horizontally with the reported value increasing from left to right. If set, the reported value increases from the bottom of the widget towards the top.

6.172.1.12 #define Slider(sName, pParent, pNext, pChild, pDisplay, lX, lY, lWidth, lHeight, lMin, lMax, lValue, ulStyle, ulFillColor, ulBackgroundFillColor, ulOutlineColor, ulTextColor, ulBackgroundTextColor, pFont, pcText, puImage, pucBackgroundImage, pfnOnChange)

Value:

```
tSliderWidget sName =
    SliderStruct(pParent, pNext, pChild, pDisplay, lX, lY, \
    lWidth, lHeight, lMin, lMax, lValue, \
    ulStyle, ulFillColor, ulBackgroundFillColor, \
    ulOutlineColor, ulTextColor, \
    ulBackgroundTextColor, pFont, pcText, \
    puImage, pucBackgroundImage, pfnOnChange)
```

Declares an initialized variable containing a slider widget data structure.

Parameters

<i>sName</i>	is the name of the variable to be declared.
<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the slider.
<i>lX</i>	is the X coordinate of the upper left corner of the slider.
<i>lY</i>	is the Y coordinate of the upper left corner of the slider.
<i>lWidth</i>	is the width of the slider.

<i>IHeight</i>	is the height of the slider.
<i>IMin</i>	is the minimum value for the slider (corresponding to the left or bottom position).
<i>IMax</i>	is the maximum value for the slider (corresponding to the right or top position).
<i>IValue</i>	is the initial value of the slider. This must lie in the range defined by <i>IMin</i> and <i>IMax</i> .
<i>ulStyle</i>	is the style to be applied to the slider.
<i>ulFillColor</i>	is the color used to fill in the slider.
<i>ul-Background-FillColor</i>	is the color used to fill in the background area of the slider.
<i>ulOutline-Color</i>	is the color used to outline the slider.
<i>ul-Background-TextColor</i>	is the color used to draw text on the slider.
<i>pFont</i>	is a pointer to the font to be used to draw text on the slider.
<i>pcText</i>	is a pointer to the text to draw on this slider.
<i>puclImage</i>	is a pointer to the image to draw on this slider.
<i>puc-Background-Image</i>	is a pointer to the image to draw on the slider background.
<i>pfnOn-Change</i>	is a pointer to the function that is called to notify the application of slider value changes.

This macro provides an initialized slider widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls).

ulStyle is the logical OR of the following:

- **SL_STYLE_OUTLINE** to indicate that the slider should be outlined.
- **SL_STYLE_FILL** to indicate that the slider should be filled.
- **SL_STYLE_BACKG_FILL** to indicate that the background portion of the slider should be filled.
- **SL_STYLE_TEXT** to indicate that the slider should have text drawn on its active portion (using *pFont* and *pcText*).
- **SL_STYLE_BACKG_TEXT** to indicate that the slider should have text drawn on its background portion (using *pFont* and *pcText*).
- **SL_STYLE_IMG** to indicate that the slider should have an image drawn on it (using *puclImage*).
- **SL_STYLE_BACKG_IMG** to indicate that the slider should have an image drawn on its background (using *pucBackgroundImage*).

- **SL_STYLE_TEXT_OPAQUE** to indicate that the slider text should be drawn opaque (in other words, drawing the background pixels).
- **SL_STYLE_BACKG_TEXT_OPAQUE** to indicate that the slider text should be drawn opaque in the background portion of the widget. (in other words, drawing the background pixels).
- **SL_STYLE_VERTICAL** to indicate that this is a vertical slider rather than a horizontal one (the default if this style flag is not set).
- **SL_STYLE_LOCKED** to indicate that the slider is being used as an indicator and should ignore user input.

Returns

Nothing; this is not a function.

6.172.1.13 #define SliderBackgroundImageOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle &= ~ (SL_STYLE_BACKG_IMG); \
}                                \
while(0)
```

Disables the image on the background area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function disables the drawing of an image on the background area of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.14 #define SliderBackgroundImageOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle |= SL_STYLE_BACKG_IMG; \
}                                \
while(0)
```

Enables the image on the background area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function enables the drawing of an image on the background area of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.15 #define SliderBackgroundImageSet(*pWidget*, *plmg*)

Value:

```
do                                \
{                                 \
    tSliderWidget *pW = pWidget;   \
    const unsigned char *pI = plmg; \
    pW->pucBackgroundImage = pI;  \
}                                 \
while (0)
```

Changes the image drawn on the background area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
----------------	---

<i>plmg</i>	is a pointer to the image to draw onto the background area of the slider.
-------------	---

This function changes the image that is drawn onto the background area of the slider. This image will be centered within the widget rectangle and the portion in the area not represented by the current slider value will be visible. The display is not updated until the next paint request.

Returns

None.

6.172.1.16 #define SliderBackgroundTextColorSet(*pWidget*, *ulColor*)

Value:

```
do                                \
{                                 \
    tSliderWidget *pW = pWidget;   \
    pW->ulBackgroundColor = ulColor; \
}                                 \
while (0)
```

Sets the background text color of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw background text on the slider.

This function changes the color used to draw text on the slider's background portion on the display. The display is not updated until the next paint request.

Returns

None.

6.172.1.17 #define SliderBackgroundTextOff(*pWidget*)

Value:

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle &= ~ (SL_STYLE_BACKG_TEXT); \
}                                \
while(0)
```

Disables the text on the background portion of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function disables the drawing of text on the background portion of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.18 #define SliderBackgroundTextOn(*pWidget*)

Value:

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle |= SL_STYLE_BACKG_TEXT; \
}                                \
while(0)
```

Enables the text on the background portion of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function enables the drawing of text on the background portion of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.19 #define SliderBackgroundTextOpaqueOff(*pWidget*)

Value:

```
do \\\n{ \\\n    tSliderWidget *pW = pWidget; \\\n    pW->ulStyle &= ~SL_STYLE_BACKG_TEXT_OPAQUE; \\\n} \\\nwhile(0) \\\n
```

Disables opaque background text on a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function disables the use of opaque text on the background portion of this slider. When not using opaque text, only the foreground pixels of the text are drawn on the screen, allowing the previously drawn pixels (such as the slider image) to show through the text. Note that SL_STYLE_BACKG_TEXT must also be cleared to disable text rendering on the slider background area.

Returns

None.

6.172.1.20 #define SliderBackgroundTextOpaqueOn(*pWidget*)

Value:

```
do \\\n{ \\\n    tSliderWidget *pW = pWidget; \\\n    pW->ulStyle |= SL_STYLE_BACKG_TEXT_OPAQUE; \\\n} \\\nwhile(0) \\\n
```

Enables opaque background text on a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function enables the use of opaque text on the background portion of this slider. When using opaque text, both the foreground and background pixels of the text are drawn on the screen, blocking out the previously drawn pixels. Note that SL_STYL-E_BACKG_TEXT must also be set to enable text rendering on the slider background area.

Returns

None.

6.172.1.21 #define SliderCallbackSet(*pWidget*, *pfnCallback*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;    \
    pW->pfnOnChange = pfnCallback; \
}                                \
while(0)
```

Sets the function to call when this slider widget's value changes.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
<i>pfnCallback</i>	is a pointer to the function to call.

This function sets the function to be called when the value represented by the slider changes.

Returns

None.

6.172.1.22 #define SliderFillColorBackgroundedSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;    \
    pW->ulBackgroundFillColor = ulColor; \
}                                \
while(0)
```

Sets the fill color for the background area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the background area of the slider.

This function changes the color used to fill the background area of the slider on the display. The display is not updated until the next paint request.

Returns

None.

6.172.1.23 #define SliderFillColorSet(*pWidget*, *ulColor*)**Value:**

```
do                                \
{                                 \
    tSliderWidget *pW = pWidget;   \
    pW->ulFillColor = ulColor;   \
}                                 \
while(0)
```

Sets the fill color for the active area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to fill the slider.

This function changes the color used to fill the active are of the slider on the display. The display is not updated until the next paint request.

Returns

None.

6.172.1.24 #define SliderFillOff(*pWidget*)**Value:**

```
do                                \
{                                 \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle &= ~ (SL_STYLE_FILL); \
}                                 \
while(0)
```

Disables filling of the active area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function disables the filling of the active area of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.25 #define SliderFillOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle |= SL_STYLE_FILL; \
}                                \
while(0)
```

Enables filling of the active area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function enables the filling of the active area of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.26 #define SliderFontSet(*pWidget*, *pFnt*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    const tFont *pF = pFnt;       \
    pW->pFont = pF;             \
}                                \
while(0)
```

Sets the font for a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
<i>pFnt</i>	is a pointer to the font to use to draw text on the slider.

Generated on Wed Oct 24 2012 22:40:50 for GLCD by Doxygen

This function changes the font used to draw text on the slider. The display is not updated until the next paint request.

Returns

None.

6.172.1.27 #define SliderImageOff(*pWidget*)

Value:

```
do                                \
{                                 \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle &= ~(SL_STYLE_IMG); \
}                                 \
while(0)
```

Disables the image on the active area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function disables the drawing of an image on the active area of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.28 #define SliderImageOn(*pWidget*)

Value:

```
do                                \
{                                 \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle |= SL_STYLE_IMG; \
}                                 \
while(0)
```

Enables the image on the active area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function enables the drawing of an image on the active area of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.29 #define SliderImageSet(pWidget, plmg)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    const unsigned char *pI = plmg; \
    pW->pucImage = pI;           \
}                                \
while(0)
```

Changes the image drawn on the active area of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>plmg</i>	is a pointer to the image to draw onto the slider.

This function changes the image that is drawn on the active area of the slider. This image will be centered within the widget rectangle and the portion represented by the current slider value will be visible. The display is not updated until the next paint request.

Returns

None.

6.172.1.30 #define SliderLock(pWidget)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle |= SL_STYLE_LOCKED; \
}                                \
while(0)
```

Locks a slider making it ignore pointer input.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function locks a slider widget and makes it ignore all pointer input. When locked, a slider acts as a passive indicator. Its value may be changed using [SliderValueSet\(\)](#) and the value display updated using [WidgetPaint\(\)](#) but no user interaction via the pointer will

change the widget value.

Returns

None.

6.172.1.31 #define SliderOutlineColorSet(*pWidget*, *ulColor*)

Value:

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulOutlineColor = ulColor; \
}                                \
while(0)
```

Sets the outline color of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to outline the slider.

This function changes the color used to outline the slider on the display. The display is not updated until the next paint request.

Returns

None.

6.172.1.32 #define SliderOutlineOff(*pWidget*)

Value:

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle &= ~ (SL_STYLE_OUTLINE); \
}                                \
while(0)
```

Disables outlining of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function disables the outlining of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.33 #define SliderOutlineOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle |= SL_STYLE_OUTLINE; \
}                                \
while(0)
```

Enables outlining of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function enables the outlining of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.34 #define SliderRangeSet(*pWidget*, *lMinimum*, *lMaximum*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->lMin = (lMinimum);       \
    pW->lMax = (lMaximum);      \
}                                \
while(0)
```

Changes the value range for a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>lMinimum</i>	is the minimum value that the slider will report.
<i>lMaximum</i>	is the maximum value that the slider will report.

This function changes the range of a slider. Slider positions are reported in terms of this range with the current position of the slider on the display being scaled and translated into this range such that the minimum value represents the left position of a horizontal

slider or the bottom position of a vertical slider and the maximum value represents the other end of the slider range. Note that this function does not cause the slider to be redrawn. The caller must call [WidgetPaint\(\)](#) explicitly after this call to ensure that the widget is redrawn.

Returns

None.

```
6.172.1.35 #define SliderStruct( pParent, pNext, pChild, pDisplay, IX, IY, IWidth,
                           IHeight, IMin, IMax, IValue, ulStyle, ulFillColor, ulBackgroundFillColor,
                           ulOutlineColor, ulTextColor, ulBackgroundTextColor, pFont, pcText, puImage,
                           pucBackgroundImage, pfnOnChange )
```

Declares an initialized slider widget data structure.

Parameters

<i>pParent</i>	is a pointer to the parent widget.
<i>pNext</i>	is a pointer to the sibling widget.
<i>pChild</i>	is a pointer to the first child widget.
<i>pDisplay</i>	is a pointer to the display on which to draw the slider.
<i>IX</i>	is the X coordinate of the upper left corner of the slider.
<i>IY</i>	is the Y coordinate of the upper left corner of the slider.
<i>IWidth</i>	is the width of the slider.
<i>IHeight</i>	is the height of the slider.
<i>IMin</i>	is the minimum value for the slider (corresponding to the left or bottom position).
<i>IMax</i>	is the maximum value for the slider (corresponding to the right or top position).
<i>IValue</i>	is the initial value of the slider. This must lie in the range defined by <i>IMin</i> and <i>IMax</i> .
<i>ulStyle</i>	is the style to be applied to the slider.
<i>ulFillColor</i>	is the color used to fill in the slider.
<i>ul-Background-FillColor</i>	is the color used to fill the background area of the slider.
<i>ulOutline-Color</i>	is the color used to outline the slider.
<i>ulTextColor</i>	is the color used to draw text on the slider.
<i>ul-Background-TextColor</i>	is the color used to draw text on the background portion of the slider.
<i>pFont</i>	is a pointer to the font to be used to draw text on the slider.
<i>pcText</i>	is a pointer to the text to draw on this slider.
<i>puImage</i>	is a pointer to the image to draw on this slider.
<i>puc-Background-Image</i>	is a pointer to the image to draw on the slider background.

<i>pfnOn- Change</i>	is a pointer to the function that is called to notify the application of slider value changes.
--------------------------	---

This macro provides an initialized slider widget data structure, which can be used to construct the widget tree at compile time in global variables (as opposed to run-time via function calls). This must be assigned to a variable, such as:

```
///!      tSliderWidget g_sSlider = SliderStruct(...);  
///!
```

Or, in an array of variables:

```
///!      tSliderWidget g_psSliders[] =  
///!      {  
///!          SliderStruct(...),  
///!          SliderStruct(...)  
///!      };  
///!
```

ulStyle is the logical OR of the following:

- **SL_STYLE_OUTLINE** to indicate that the slider should be outlined.
- **SL_STYLE_FILL** to indicate that the slider should be filled.
- **SL_STYLE_BACKG_FILL** to indicate that the background portion of the slider should be filled.
- **SL_STYLE_TEXT** to indicate that the slider should have text drawn on its active portion (using *pFont* and *pcText*).
- **SL_STYLE_BACKG_TEXT** to indicate that the slider should have text drawn on its background portion (using *pFont* and *pcText*).
- **SL_STYLE_IMG** to indicate that the slider should have an image drawn on it (using *puclImage*).
- **SL_STYLE_BACKG_IMG** to indicate that the slider should have an image drawn on its background (using *pucBackgroundImage*).
- **SL_STYLE_TEXT_OPAQUE** to indicate that the slider text should be drawn opaque (in other words, drawing the background pixels).
- **SL_STYLE_BACKG_TEXT_OPAQUE** to indicate that the slider text should be drawn opaque in the background portion of the widget. (in other words, drawing the background pixels).
- **SL_STYLE_VERTICAL** to indicate that this is a vertical slider rather than a horizontal one (the default if this style flag is not set).
- **SL_STYLE_LOCKED** to indicate that the slider is being used as an indicator and should ignore user input.

Returns

Nothing; this is not a function.

6.172.1.36 #define SliderTextColorSet(*pWidget*, *ulColor*)

Value:

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulTextColor = ulColor;    \
}                                \
while(0)
```

Sets the text color of the active portion of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>ulColor</i>	is the 24-bit RGB color to use to draw text on the slider.

This function changes the color used to draw text on the active portion of the slider on the display. The display is not updated until the next paint request.

Returns

None.

6.172.1.37 #define SliderTextOff(*pWidget*)

Value:

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle &= ~SL_STYLE_TEXT; \
}                                \
while(0)
```

Disables the text on the active portion of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function disables the drawing of text on the active portion of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.38 #define SliderTextOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle |= SL_STYLE_TEXT; \
}                                \
while(0)
```

Enables the text on the active portion of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function enables the drawing of text on the active portion of a slider widget. The display is not updated until the next paint request.

Returns

None.

6.172.1.39 #define SliderTextOpaqueOff(*pWidget*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle &= ~ (SL_STYLE_TEXT_OPAQUE); \
}                                \
while(0)
```

Disables opaque text on the active portion of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function disables the use of opaque text on the active portion of this slider. When not using opaque text, only the foreground pixels of the text are drawn on the screen, allowing the previously drawn pixels (such as the slider image) to show through the text. Note that SL_STYLE_TEXT must also be cleared to disable text rendering on the slider active area.

Returns

None.

6.172.1.40 #define SliderTextOpaqueOn(*pWidget*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->ulStyle |= SL_STYLE_TEXT_OPAQUE; \
}                                \
while(0)
```

Enables opaque text on the active portion of a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function enables the use of opaque text on the active portion of this slider. When using opaque text, both the foreground and background pixels of the text are drawn on the screen, blocking out the previously drawn pixels. Note that SL_STYLE_TEXT must also be set to enable text rendering on the slider active area.

Returns

None.

6.172.1.41 #define SliderTextSet(*pWidget*, *pcTxt*)**Value:**

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    const char *pcT = pcTxt;      \
    pW->pcText = pcT;           \
}                                \
while(0)
```

Changes the text drawn on a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>pcTxt</i>	is a pointer to the text to draw onto the slider.

This function changes the text that is drawn onto the slider. The string is centered across the slider and straddles the active and background portions of the widget. The display

is not updated until the next paint request.

Returns

None.

6.172.1.42 #define SliderUnlock(*pWidget*)

Value:

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->uiStyle &= ~(SL_STYLE_LOCKED); \
}                                \
while(0)
```

Unlocks a slider making it pay attention to pointer input.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to modify.
----------------	--

This function unlocks a slider widget. When unlocked, a slider will respond to pointer input by setting its value appropriately and informing the application via callbacks.

Returns

None.

6.172.1.43 #define SliderValueSet(*pWidget*, *lVal*)

Value:

```
do                                \
{                                \
    tSliderWidget *pW = pWidget;   \
    pW->lValue = (lVal);        \
}                                \
while(0)
```

Changes the minimum value for a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to be modified.
<i>lVal</i>	is the new value to set for the slider. This is in terms of the value range currently set for the slider.

This function changes the value that the slider will display the next time the widget is painted. The caller is responsible for ensuring that the value passed is within the range specified for the target widget. The caller must call [WidgetPaint\(\)](#) explicitly after this call to ensure that the widget is redrawn.

Returns

None.

6.172.2 Function Documentation

6.172.2.1 void SliderInit (tSliderWidget * pWidget, const tDisplay * pDisplay, unsigned IX, unsigned IY, unsigned IWidth, unsigned IHeight)

Initializes a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget to initialize.
<i>pDisplay</i>	is a pointer to the display on which to draw the slider.
<i>IX</i>	is the X coordinate of the upper left corner of the slider.
<i>IY</i>	is the Y coordinate of the upper left corner of the slider.
<i>IWidth</i>	is the width of the slider.
<i>IHeight</i>	is the height of the slider.

This function initializes the provided slider widget.

Returns

None.

6.172.2.2 int SliderMsgProc (tWidget * pWidget, unsigned ulMsg, unsigned ulParam1, unsigned ulParam2)

Handles messages for a slider widget.

Parameters

<i>pWidget</i>	is a pointer to the slider widget.
<i>ulMsg</i>	is the message.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function receives messages intended for this slider widget and processes them accordingly. The processing of the message varies based on the message in question.

Unrecognized messages are handled by calling [WidgetDefaultMsgProc\(\)](#).

Returns

Returns a value appropriate to the supplied message.

6.173 include/widget.h File Reference

Data Structures

- struct [__Widget](#)

Defines

- #define [WIDGET_ROOT](#) &g_sRoot
- #define [WIDGET_MSG_PAINT](#) 0x0001
- #define [WIDGET_MSG_PTR_DOWN](#) 0x0002
- #define [WIDGET_MSG_PTR_MOVE](#) 0x0003
- #define [WIDGET_MSG_PTR_UP](#) 0x0004
- #define [WidgetPaint\(pWidget\)](#) [WidgetMessageQueueAdd\(pWidget, WIDGET_MSG_PAINT, 0, 0, 0, 0\)](#)

Typedefs

- typedef struct [__Widget](#) tWidget

Functions

- int [WidgetDefaultMsgProc](#) (tWidget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2)
- void [WidgetAdd](#) (tWidget *pParent, tWidget *pWidget)
- void [WidgetRemove](#) (tWidget *pWidget)
- int [WidgetMessageSendPreOrder](#) (tWidget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2, int bStopOnSuccess)
- int [WidgetMessageSendPostOrder](#) (tWidget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2, int bStopOnSuccess)
- int [WidgetMessageQueueAdd](#) (tWidget *pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2, unsigned bPostOrder, int bStopOnSuccess)
- void [WidgetMessageQueueProcess](#) (void)
- int [WidgetPointerMessage](#) (unsigned ulMessage, unsigned IX, unsigned IY)
- void [WidgetMutexInit](#) (int *pcMutex)
- int [WidgetMutexGet](#) (int *pcMutex)
- void [WidgetMutexPut](#) (int *pcMutex)

Variables

- [tWidget g_sRoot](#)

6.173.1 Define Documentation

6.173.1.1 #define WIDGET_MSG_PAINT 0x0001

This message is sent to indicate that the widget should draw itself on the display. - Neither *ulParam1* nor *ulParam2* are used by this message. This message is delivered in top-down order.

6.173.1.2 #define WIDGET_MSG_PTR_DOWN 0x0002

This message is sent to indicate that the pointer is now down. *ulParam1* is the X coordinate of the location where the pointer down event occurred, and *ulParam2* is the Y coordinate. This message is delivered in bottom-up order.

6.173.1.3 #define WIDGET_MSG_PTR_MOVE 0x0003

This message is sent to indicate that the pointer has moved while being down. *ulParam1* is the X coordinate of the new pointer location, and *ulParam2* is the Y coordinate. This message is delivered in bottom-up order.

6.173.1.4 #define WIDGET_MSG_PTR_UP 0x0004

This message is sent to indicate that the pointer is now up. *ulParam1* is the X coordinate of the location where the pointer up event occurred, and *ulParam2* is the Y coordinate. This message is delivered in bottom-up order.

6.173.1.5 #define WIDGET_ROOT &g_sRoot

The widget at the root of the widget tree. This can be used when constructing a widget tree at compile time (used as the *pParent* argument to a widget declaration) or as the *pWidget* argument to an API (such as [WidgetPaint\(\)](#) to paint the entire widget tree).

6.173.1.6 #define WidgetPaint(pWidget) WidgetMessageQueueAdd(pWidget, WIDGET_MSG_PAINT, 0, 0, 0, 0)

Requests a redraw of the widget tree.

Parameters

<i>pWidget</i>	is a pointer to the widget tree to paint.
----------------	---

This function sends a **WIDGET_MSG_PAINT** message to the given widgets, and all of the widget beneath it, so that they will draw or redraw themselves on the display. The actual drawing will occur when this message is retrieved from the message queue and processed.

Returns

Returns 1 if the message was added to the message queue and 0 if it could not be added (due to the queue being full).

6.173.2 Typedef Documentation

6.173.2.1 `typedef struct __Widget tWidget`

The structure that describes a generic widget. This structure is the base “class” for all other widgets.

6.173.3 Function Documentation

6.173.3.1 `void WidgetAdd (tWidget * pParent, tWidget * pWidget)`

Adds a widget to the widget tree.

Parameters

<i>pParent</i>	is the parent for the widget. To add to the root of the tree set this parameter to WIDGET_ROOT .
<i>pWidget</i>	is the widget to add.

This function adds a widget to the widget tree at the given position within the tree. The widget will become the last child of its parent, and will therefore be searched after the existing children.

The added widget can be a full widget tree, allowing addition of an entire hierarchy all at once (for example, adding an entire screen to the widget tree all at once). In this case, it is the responsibility of the caller to ensure that the *pParent* field of each widget in the added tree is correctly set (in other words, only the widget pointed to by *pWidget* is updated to properly reside in the tree).

It is the responsibility of the caller to initialize the *pNext* and *pChild* field of the added widget; either of these fields being non-zero results in a pre-defined tree of widgets being added instead of a single one.

Returns

None.

**6.173.3.2 int WidgetDefaultMsgProc (tWidget * pWidget, unsigned ulMessage,
unsigned ulParam1, unsigned ulParam2)**

Handles widget messages.

Parameters

<i>pWidget</i>	is a pointer to the widget.
<i>ulMessage</i>	is the message to be processed.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.

This function is a default handler for widget messages; it simply ignores all messages sent to it. This is used as the message handler for the root widget, and should be called by the message handler for other widgets when they do not explicitly handle the provided message (in case new messages are added that require some default but override-able processing).

Returns

Always returns 0.

**6.173.3.3 int WidgetMessageQueueAdd (tWidget * pWidget, unsigned ulMessage,
unsigned ulParam1, unsigned ulParam2, unsigned bPostOrder, int bStopOnSuccess
)**

Adds a message to the widget message queue.

Parameters

<i>pWidget</i>	is the widget to which the message should be sent.
<i>ulMessage</i>	is the message to be sent.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.
<i>bPostOrder</i>	is true if the message should be sent via a post-order search, and false if it should be sent via a pre-order search.
<i>bStopOn-Success</i>	is true if the message should be sent to widgets until one returns success, and false if it should be sent to all widgets.

This function places a widget message into the message queue for later processing. The messages are removed from the queue by [WidgetMessageQueueProcess\(\)](#) and sent to the appropriate place.

It is safe for code which interrupts [WidgetMessageQueueProcess\(\)](#) (or called by it) to call this function to send a message. It is not safe for code which interrupts this function to call this function as well; it is up to the caller to guarantee that the later sequence never occurs.

Returns

Returns 1 if the message was added to the queue, and 0 if it could not be added since either the queue is full or another context is currently adding a message to the queue.

6.173.3.4 void WidgetMessageQueueProcess (void)

Processes the messages in the widget message queue.

This function extracts messages from the widget message queue one at a time and processes them. If the processing of a widget message requires that a new message be sent, it is acceptable to call [WidgetMessageQueueAdd\(\)](#). It is also acceptable for code which interrupts this function to call [WidgetMessageQueueAdd\(\)](#) to send more messages. In both cases, the newly added message will also be processed before this function returns.

Returns

None.

6.173.3.5 int WidgetMessageSendPostOrder (tWidget * pWidget, unsigned ulMessage, unsigned ulParam1, unsigned ulParam2, int bStopOnSuccess)

Sends a message to a widget tree via a post-order, depth-first search.

Parameters

<i>pWidget</i>	is a pointer to the widget tree; if this is zero then the root of the widget tree will be used.
<i>ulMessage</i>	is the message to send.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.
<i>bStopOn- Success</i>	is true if the search should be stopped when the first widget is found that returns success in response to the message.

This function performs a post-order, depth-first search of the widget tree, sending a message to each widget encountered. In a depth-first search, the children of a widget are searched before its sibling (preferring to go deeper into the tree, hence the name depth-first). A post-order search means that the message is sent to a widget after all of its children are searched.

An example use of the post-order search is for pointer-related messages; those messages should be delivered to the lowest widget in the tree before its parents (in other words, the widget deepest in the tree that has a hit should get the message, not the higher up widgets that also include the hit location).

Special handling is performed for pointer-related messages. The widget that accepts [WIDGET_MSG_PTR_DOWN](#) is remembered and subsequent [WIDGET_MSG_PTR_MOVE](#) and [WIDGET_MSG_PTR_UP](#) messages are sent directly to that widget.

Returns

Returns 0 if *bStopOnSuccess* is false or no widget returned success in response to the message, or the value returned by the first widget to successfully process the message.

6.173.3.6 int WidgetMessageSendPreOrder (tWidget * *pWidget*, unsigned *ulMessage*, unsigned *ulParam1*, unsigned *ulParam2*, int *bStopOnSuccess*)

Sends a message to a widget tree via a pre-order, depth-first search.

Parameters

<i>pWidget</i>	is a pointer to the widget tree.
<i>ulMessage</i>	is the message to send.
<i>ulParam1</i>	is the first parameter to the message.
<i>ulParam2</i>	is the second parameter to the message.
<i>bStopOn- Success</i>	is true if the search should be stopped when the first widget is found that returns success in response to the message.

This function performs a pre-order, depth-first search of the widget tree, sending a message to each widget encountered. In a depth-first search, the children of a widget are searched before its siblings (preferring to go deeper into the tree, hence the name depth-first). A pre-order search means that the message is sent to a widget before any of its children are searched.

An example use of the pre-order search is for paint messages; the larger enclosing widgets should be drawn on the screen before the smaller widgets that reside within the parent widget (otherwise, the children would be overwritten by the parent).

Returns

Returns 0 if *bStopOnSuccess* is false or no widget returned success in response to the message, or the value returned by the first widget to successfully process the message.

6.173.3.7 int WidgetMutexGet (int * *pcMutex*)

Attempts to acquire a mutex.

Parameters

<i>pcMutex</i>	is a pointer to mutex that is to be acquired.
----------------	---

This function attempts to acquire a mutual exclusion semaphore (mutex) on behalf of the caller. If the mutex is not already held, 0 is returned to indicate that the caller may safely access whichever resource the mutex is protecting. If the mutex is already held, 1 is returned and the caller must not access the shared resource.

When access to the shared resource is complete, the mutex owner should call [WidgetMutexPut\(\)](#) to release the mutex and relinquish ownership of the shared resource.

Returns

Returns 0 if the mutex is acquired successfully or 1 if it is already held by another caller.

6.173.3.8 void WidgetMutexInit (int * pcMutex)

Initializes a mutex to the unowned state.

Parameters

<i>pcMutex</i>	is a pointer to mutex that is to be initialized.
----------------	--

This function initializes a mutual exclusion semaphore (mutex) to its unowned state in preparation for use with [WidgetMutexGet\(\)](#) and [WidgetMutexPut\(\)](#). A mutex is a two state object typically used to serialize access to a shared resource. An application will call [WidgetMutexGet\(\)](#) to request ownership of the mutex. If ownership is granted, the caller may safely access the resource then release the mutex using [WidgetMutexPut\(\)](#) once it is finished. If ownership is not granted, the caller knows that some other context is currently modifying the shared resource and it must not access the resource at that time.

Note that this function must not be called if the mutex passed in *pcMutex* is already in use since this will have the effect of releasing the lock even if some caller currently owns it.

Returns

None.

6.173.3.9 void WidgetMutexPut (int * pcMutex)

Release a mutex.

Parameters

<i>pcMutex</i>	is a pointer to mutex that is to be released.
----------------	---

This function releases a mutual exclusion semaphore (mutex), leaving it in the unowned state.

Returns

None.

6.173.3.10 int WidgetPointerMessage (unsigned *uIMessage*, unsigned *IX*, unsigned *IY*)

Sends a pointer message.

Parameters

<i>uIMessage</i>	is the pointer message to be sent.
<i>IX</i>	is the X coordinate associated with the message.
<i>IY</i>	is the Y coordinate associated with the message.

This function sends a pointer message to the root widget. A pointer driver (such as a touch screen driver) can use this function to deliver pointer activity to the widget tree without having to have direct knowledge of the structure of the widget framework.

Returns

Returns 1 if the message was added to the queue, and 0 if it could not be added since the queue is full.

6.173.3.11 void WidgetRemove (tWidget * *pWidget*)

Removes a widget from the widget tree.

Parameters

<i>pWidget</i>	is the widget to be removed.
----------------	------------------------------

This function removes a widget from the widget tree. The removed widget can be a full widget tree, allowing removal of an entire heirarchy all at once (for example, removing an entire screen from the widget tree).

Returns

None.

6.173.4 Variable Documentation

6.173.4.1 tWidget g_sRoot

6.174 OLED/graphic.c File Reference

Functions

- void [GLCD_SetPixel](#) (int x, int y, int *color*)
- void [GLCD_Rectangle](#) (unsigned int x, unsigned int y, unsigned int b, unsigned int a)
- void [GLCD_Circle](#) (unsigned int cx, unsigned int cy, unsigned int radius)
- void [GLCD_Line](#) (int X1, int Y1, int X2, int Y2)

Variables

- const unsigned char `color` = 1

6.174.1 Function Documentation

6.174.1.1 void `GLCD_Circle` (`unsigned int cx`, `unsigned int cy`, `unsigned int radius`)

6.174.1.2 void `GLCD_Line` (`int X1`, `int Y1`, `int X2`, `int Y2`)

6.174.1.3 void `GLCD_Rectangle` (`unsigned int x`, `unsigned int y`, `unsigned int b`, `unsigned int a`)

6.174.1.4 void `GLCD_SetPixel` (`int x`, `int y`, `int color`)

6.174.2 Variable Documentation

6.174.2.1 const unsigned char `color` = 1

6.175 src/main.c File Reference

```
#include <grlib.h> #include <sed1335.h>
```

Functions

- int `main` (void)

6.175.1 Function Documentation

6.175.1.1 int `main` (void)

6.176 src/msp430-main.c File Reference

```
#include <iomacros.h> #include <in430.h> #include <msp430.-  
h> #include <grlib.h> #include <sed1335.h> #include <isr-  
_compat.h>
```

Functions

- int `main` (void)

6.176.1 Function Documentation

6.176.1.1 int main(void)

6.177 src/old_main.c File Reference

```
#include <iomacros.h> #include <msp430.h>
```

Functions

- int **main** (void)

6.177.1 Function Documentation

6.177.1.1 int main(void)