

# GLCD

Generated by Doxygen 1.7.6.1

Wed Oct 24 2012 22:02:30



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Data Structure Index</b>	<b>3</b>
2.1	Data Structures . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Module Documentation</b>	<b>9</b>
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
<b>5</b>	<b>Data Structure Documentation</b>	<b>69</b>
5.1	__Widget Struct Reference . . . . .	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	tCanvasWidget Struct Reference . . . . .	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	tImageButtonWidget 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
<b>6</b>	<b>File Documentation</b>	<b>93</b>
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 . . . . .	101
6.6.1.6	GLCD_InitializePorts . . . . .	101
6.6.1.7	GLCD_SetCursorAddress . . . . .	101
6.6.1.8	GLCD_SetPixel . . . . .	101
6.6.1.9	GLCD_TextGoTo . . . . .	101
6.6.1.10	GLCD_WriteText . . . . .	101
6.6.1.11	GLCD_WriteTextP . . . . .	101
6.7	grlib/canvas.c File Reference . . . . .	101
6.7.1	Function Documentation . . . . .	101
6.7.1.1	CanvasInit . . . . .	102
6.7.1.2	CanvasMsgProc . . . . .	102
6.8	grlib/checkbox.c File Reference . . . . .	102
6.8.1	Function Documentation . . . . .	103
6.8.1.1	CheckBoxInit . . . . .	103
6.8.1.2	CheckBoxMsgProc . . . . .	103
6.9	grlib/circle.c File Reference . . . . .	104
6.9.1	Function Documentation . . . . .	104
6.9.1.1	GrCircleDraw . . . . .	104
6.9.1.2	GrCircleFill . . . . .	104
6.10	grlib/container.c File Reference . . . . .	105
6.10.1	Function Documentation . . . . .	105
6.10.1.1	ContainerInit . . . . .	105
6.10.1.2	ContainerMsgProc . . . . .	105
6.11	grlib/context.c File Reference . . . . .	106

6.11.1	Function Documentation . . . . .	106
6.11.1.1	GrContextClipRegionSet . . . . .	106
6.11.1.2	GrContextInit . . . . .	107
6.12	grlib/fonts/fontcm12.c File Reference . . . . .	107
6.12.1	Variable Documentation . . . . .	107
6.12.1.1	g_sFontCm12 . . . . .	107
6.13	grlib/fonts/fontcm12b.c File Reference . . . . .	107
6.13.1	Variable Documentation . . . . .	107
6.13.1.1	g_sFontCm12b . . . . .	107
6.14	grlib/fonts/fontcm12i.c File Reference . . . . .	108
6.14.1	Variable Documentation . . . . .	108
6.14.1.1	g_sFontCm12i . . . . .	108
6.15	grlib/fonts/fontcm14.c File Reference . . . . .	108
6.15.1	Variable Documentation . . . . .	108
6.15.1.1	g_sFontCm14 . . . . .	108
6.16	grlib/fonts/fontcm14b.c File Reference . . . . .	108
6.16.1	Variable Documentation . . . . .	108
6.16.1.1	g_sFontCm14b . . . . .	108
6.17	grlib/fonts/fontcm14i.c File Reference . . . . .	108
6.17.1	Variable Documentation . . . . .	109
6.17.1.1	g_sFontCm14i . . . . .	109
6.18	grlib/fonts/fontcm16.c File Reference . . . . .	109
6.18.1	Variable Documentation . . . . .	109
6.18.1.1	g_sFontCm16 . . . . .	109
6.19	grlib/fonts/fontcm16b.c File Reference . . . . .	109
6.19.1	Variable Documentation . . . . .	109
6.19.1.1	g_sFontCm16b . . . . .	109
6.20	grlib/fonts/fontcm16i.c File Reference . . . . .	109
6.20.1	Variable Documentation . . . . .	110
6.20.1.1	g_sFontCm16i . . . . .	110
6.21	grlib/fonts/fontcm18.c File Reference . . . . .	110
6.21.1	Variable Documentation . . . . .	110
6.21.1.1	g_sFontCm18 . . . . .	110
6.22	grlib/fonts/fontcm18b.c File Reference . . . . .	110

6.22.1	Variable Documentation	110
6.22.1.1	g_sFontCm18b	110
6.23	grlib/fonts/fontcm18i.c File Reference	110
6.23.1	Variable Documentation	110
6.23.1.1	g_sFontCm18i	110
6.24	grlib/fonts/fontcm20.c File Reference	111
6.24.1	Variable Documentation	111
6.24.1.1	g_sFontCm20	111
6.25	grlib/fonts/fontcm20b.c File Reference	111
6.25.1	Variable Documentation	111
6.25.1.1	g_sFontCm20b	111
6.26	grlib/fonts/fontcm20i.c File Reference	111
6.26.1	Variable Documentation	111
6.26.1.1	g_sFontCm20i	111
6.27	grlib/fonts/fontcm22.c File Reference	111
6.27.1	Variable Documentation	112
6.27.1.1	g_sFontCm22	112
6.28	grlib/fonts/fontcm22b.c File Reference	112
6.28.1	Variable Documentation	112
6.28.1.1	g_sFontCm22b	112
6.29	grlib/fonts/fontcm22i.c File Reference	112
6.29.1	Variable Documentation	112
6.29.1.1	g_sFontCm22i	112
6.30	grlib/fonts/fontcm24.c File Reference	112
6.30.1	Variable Documentation	113
6.30.1.1	g_sFontCm24	113
6.31	grlib/fonts/fontcm24b.c File Reference	113
6.31.1	Variable Documentation	113
6.31.1.1	g_sFontCm24b	113
6.32	grlib/fonts/fontcm24i.c File Reference	113
6.32.1	Variable Documentation	113
6.32.1.1	g_sFontCm24i	113
6.33	grlib/fonts/fontcm26.c File Reference	113
6.33.1	Variable Documentation	113

6.33.1.1	g_sFontCm26	113
6.34	grlib/fonts/fontcm26b.c File Reference	114
6.34.1	Variable Documentation	114
6.34.1.1	g_sFontCm26b	114
6.35	grlib/fonts/fontcm26i.c File Reference	114
6.35.1	Variable Documentation	114
6.35.1.1	g_sFontCm26i	114
6.36	grlib/fonts/fontcm28.c File Reference	114
6.36.1	Variable Documentation	114
6.36.1.1	g_sFontCm28	114
6.37	grlib/fonts/fontcm28b.c File Reference	114
6.37.1	Variable Documentation	115
6.37.1.1	g_sFontCm28b	115
6.38	grlib/fonts/fontcm28i.c File Reference	115
6.38.1	Variable Documentation	115
6.38.1.1	g_sFontCm28i	115
6.39	grlib/fonts/fontcm30.c File Reference	115
6.39.1	Variable Documentation	115
6.39.1.1	g_sFontCm30	115
6.40	grlib/fonts/fontcm30b.c File Reference	115
6.40.1	Variable Documentation	116
6.40.1.1	g_sFontCm30b	116
6.41	grlib/fonts/fontcm30i.c File Reference	116
6.41.1	Variable Documentation	116
6.41.1.1	g_sFontCm30i	116
6.42	grlib/fonts/fontcm32.c File Reference	116
6.42.1	Variable Documentation	116
6.42.1.1	g_sFontCm32	116
6.43	grlib/fonts/fontcm32b.c File Reference	116
6.43.1	Variable Documentation	116
6.43.1.1	g_sFontCm32b	116
6.44	grlib/fonts/fontcm32i.c File Reference	117
6.44.1	Variable Documentation	117
6.44.1.1	g_sFontCm32i	117

6.45 grlib/fonts/fontcm34.c File Reference . . . . .	117
6.45.1 Variable Documentation . . . . .	117
6.45.1.1 g_sFontCm34 . . . . .	117
6.46 grlib/fonts/fontcm34b.c File Reference . . . . .	117
6.46.1 Variable Documentation . . . . .	117
6.46.1.1 g_sFontCm34b . . . . .	117
6.47 grlib/fonts/fontcm34i.c File Reference . . . . .	117
6.47.1 Variable Documentation . . . . .	118
6.47.1.1 g_sFontCm34i . . . . .	118
6.48 grlib/fonts/fontcm36.c File Reference . . . . .	118
6.48.1 Variable Documentation . . . . .	118
6.48.1.1 g_sFontCm36 . . . . .	118
6.49 grlib/fonts/fontcm36b.c File Reference . . . . .	118
6.49.1 Variable Documentation . . . . .	118
6.49.1.1 g_sFontCm36b . . . . .	118
6.50 grlib/fonts/fontcm36i.c File Reference . . . . .	118
6.50.1 Variable Documentation . . . . .	119
6.50.1.1 g_sFontCm36i . . . . .	119
6.51 grlib/fonts/fontcm38.c File Reference . . . . .	119
6.51.1 Variable Documentation . . . . .	119
6.51.1.1 g_sFontCm38 . . . . .	119
6.52 grlib/fonts/fontcm38b.c File Reference . . . . .	119
6.52.1 Variable Documentation . . . . .	119
6.52.1.1 g_sFontCm38b . . . . .	119
6.53 grlib/fonts/fontcm38i.c File Reference . . . . .	119
6.53.1 Variable Documentation . . . . .	119
6.53.1.1 g_sFontCm38i . . . . .	119
6.54 grlib/fonts/fontcm40.c File Reference . . . . .	120
6.54.1 Variable Documentation . . . . .	120
6.54.1.1 g_sFontCm40 . . . . .	120
6.55 grlib/fonts/fontcm40b.c File Reference . . . . .	120
6.55.1 Variable Documentation . . . . .	120
6.55.1.1 g_sFontCm40b . . . . .	120
6.56 grlib/fonts/fontcm40i.c File Reference . . . . .	120

6.56.1	Variable Documentation	120
6.56.1.1	g_sFontCm40i	120
6.57	grlib/fonts/fontcm42.c File Reference	120
6.57.1	Variable Documentation	121
6.57.1.1	g_sFontCm42	121
6.58	grlib/fonts/fontcm42b.c File Reference	121
6.58.1	Variable Documentation	121
6.58.1.1	g_sFontCm42b	121
6.59	grlib/fonts/fontcm42i.c File Reference	121
6.59.1	Variable Documentation	121
6.59.1.1	g_sFontCm42i	121
6.60	grlib/fonts/fontcm44.c File Reference	121
6.60.1	Variable Documentation	122
6.60.1.1	g_sFontCm44	122
6.61	grlib/fonts/fontcm44b.c File Reference	122
6.61.1	Variable Documentation	122
6.61.1.1	g_sFontCm44b	122
6.62	grlib/fonts/fontcm44i.c File Reference	122
6.62.1	Variable Documentation	122
6.62.1.1	g_sFontCm44i	122
6.63	grlib/fonts/fontcm46.c File Reference	122
6.63.1	Variable Documentation	122
6.63.1.1	g_sFontCm46	122
6.64	grlib/fonts/fontcm46b.c File Reference	123
6.64.1	Variable Documentation	123
6.64.1.1	g_sFontCm46b	123
6.65	grlib/fonts/fontcm46i.c File Reference	123
6.65.1	Variable Documentation	123
6.65.1.1	g_sFontCm46i	123
6.66	grlib/fonts/fontcm48.c File Reference	123
6.66.1	Variable Documentation	123
6.66.1.1	g_sFontCm48	123
6.67	grlib/fonts/fontcm48b.c File Reference	123
6.67.1	Variable Documentation	124

6.67.1.1	<i>g_sFontCm48b</i>	124
6.68	grlib/fonts/fontcm48i.c File Reference	124
6.68.1	Variable Documentation	124
6.68.1.1	<i>g_sFontCm48i</i>	124
6.69	grlib/fonts/fontcmsc12.c File Reference	124
6.69.1	Variable Documentation	124
6.69.1.1	<i>g_sFontCmsc12</i>	124
6.70	grlib/fonts/fontcmsc14.c File Reference	124
6.70.1	Variable Documentation	125
6.70.1.1	<i>g_sFontCmsc14</i>	125
6.71	grlib/fonts/fontcmsc16.c File Reference	125
6.71.1	Variable Documentation	125
6.71.1.1	<i>g_sFontCmsc16</i>	125
6.72	grlib/fonts/fontcmsc18.c File Reference	125
6.72.1	Variable Documentation	125
6.72.1.1	<i>g_sFontCmsc18</i>	125
6.73	grlib/fonts/fontcmsc20.c File Reference	125
6.73.1	Variable Documentation	125
6.73.1.1	<i>g_sFontCmsc20</i>	125
6.74	grlib/fonts/fontcmsc22.c File Reference	126
6.74.1	Variable Documentation	126
6.74.1.1	<i>g_sFontCmsc22</i>	126
6.75	grlib/fonts/fontcmsc24.c File Reference	126
6.75.1	Variable Documentation	126
6.75.1.1	<i>g_sFontCmsc24</i>	126
6.76	grlib/fonts/fontcmsc26.c File Reference	126
6.76.1	Variable Documentation	126
6.76.1.1	<i>g_sFontCmsc26</i>	126
6.77	grlib/fonts/fontcmsc28.c File Reference	126
6.77.1	Variable Documentation	127
6.77.1.1	<i>g_sFontCmsc28</i>	127
6.78	grlib/fonts/fontcmsc30.c File Reference	127
6.78.1	Variable Documentation	127
6.78.1.1	<i>g_sFontCmsc30</i>	127

6.79 grlib/fonts/fontcmsc32.c File Reference . . . . .	127
6.79.1 Variable Documentation . . . . .	127
6.79.1.1 g_sFontCmsc32 . . . . .	127
6.80 grlib/fonts/fontcmsc34.c File Reference . . . . .	127
6.80.1 Variable Documentation . . . . .	128
6.80.1.1 g_sFontCmsc34 . . . . .	128
6.81 grlib/fonts/fontcmsc36.c File Reference . . . . .	128
6.81.1 Variable Documentation . . . . .	128
6.81.1.1 g_sFontCmsc36 . . . . .	128
6.82 grlib/fonts/fontcmsc38.c File Reference . . . . .	128
6.82.1 Variable Documentation . . . . .	128
6.82.1.1 g_sFontCmsc38 . . . . .	128
6.83 grlib/fonts/fontcmsc40.c File Reference . . . . .	128
6.83.1 Variable Documentation . . . . .	128
6.83.1.1 g_sFontCmsc40 . . . . .	128
6.84 grlib/fonts/fontcmsc42.c File Reference . . . . .	129
6.84.1 Variable Documentation . . . . .	129
6.84.1.1 g_sFontCmsc42 . . . . .	129
6.85 grlib/fonts/fontcmsc44.c File Reference . . . . .	129
6.85.1 Variable Documentation . . . . .	129
6.85.1.1 g_sFontCmsc44 . . . . .	129
6.86 grlib/fonts/fontcmsc46.c File Reference . . . . .	129
6.86.1 Variable Documentation . . . . .	129
6.86.1.1 g_sFontCmsc46 . . . . .	129
6.87 grlib/fonts/fontcmsc48.c File Reference . . . . .	129
6.87.1 Variable Documentation . . . . .	130
6.87.1.1 g_sFontCmsc48 . . . . .	130
6.88 grlib/fonts/fontcmss12.c File Reference . . . . .	130
6.88.1 Variable Documentation . . . . .	130
6.88.1.1 g_sFontCmss12 . . . . .	130
6.89 grlib/fonts/fontcmss12b.c File Reference . . . . .	130
6.89.1 Variable Documentation . . . . .	130
6.89.1.1 g_sFontCmss12b . . . . .	130
6.90 grlib/fonts/fontcmss12i.c File Reference . . . . .	130

6.90.1	Variable Documentation	131
6.90.1.1	g_sFontCmss12i	131
6.91	grlib/fonts/fontcmss14.c File Reference	131
6.91.1	Variable Documentation	131
6.91.1.1	g_sFontCmss14	131
6.92	grlib/fonts/fontcmss14b.c File Reference	131
6.92.1	Variable Documentation	131
6.92.1.1	g_sFontCmss14b	131
6.93	grlib/fonts/fontcmss14i.c File Reference	131
6.93.1	Variable Documentation	131
6.93.1.1	g_sFontCmss14i	131
6.94	grlib/fonts/fontcmss16.c File Reference	132
6.94.1	Variable Documentation	132
6.94.1.1	g_sFontCmss16	132
6.95	grlib/fonts/fontcmss16b.c File Reference	132
6.95.1	Variable Documentation	132
6.95.1.1	g_sFontCmss16b	132
6.96	grlib/fonts/fontcmss16i.c File Reference	132
6.96.1	Variable Documentation	132
6.96.1.1	g_sFontCmss16i	132
6.97	grlib/fonts/fontcmss18.c File Reference	132
6.97.1	Variable Documentation	133
6.97.1.1	g_sFontCmss18	133
6.98	grlib/fonts/fontcmss18b.c File Reference	133
6.98.1	Variable Documentation	133
6.98.1.1	g_sFontCmss18b	133
6.99	grlib/fonts/fontcmss18i.c File Reference	133
6.99.1	Variable Documentation	133
6.99.1.1	g_sFontCmss18i	133
6.100	grlib/fonts/fontcmss20.c File Reference	133
6.100.1	Variable Documentation	134
6.100.1.1	g_sFontCmss20	134
6.101	grlib/fonts/fontcmss20b.c File Reference	134
6.101.1	Variable Documentation	134

6.101.1.1 g_sFontCmss20b . . . . .	134
6.102grlib/fonts/fontcmss20i.c File Reference . . . . .	134
6.102.1 Variable Documentation . . . . .	134
6.102.1.1 g_sFontCmss20i . . . . .	134
6.103grlib/fonts/fontcmss22.c File Reference . . . . .	134
6.103.1 Variable Documentation . . . . .	134
6.103.1.1 g_sFontCmss22 . . . . .	134
6.104grlib/fonts/fontcmss22b.c File Reference . . . . .	135
6.104.1 Variable Documentation . . . . .	135
6.104.1.1 g_sFontCmss22b . . . . .	135
6.105grlib/fonts/fontcmss22i.c File Reference . . . . .	135
6.105.1 Variable Documentation . . . . .	135
6.105.1.1 g_sFontCmss22i . . . . .	135
6.106grlib/fonts/fontcmss24.c File Reference . . . . .	135
6.106.1 Variable Documentation . . . . .	135
6.106.1.1 g_sFontCmss24 . . . . .	135
6.107grlib/fonts/fontcmss24b.c File Reference . . . . .	135
6.107.1 Variable Documentation . . . . .	136
6.107.1.1 g_sFontCmss24b . . . . .	136
6.108grlib/fonts/fontcmss24i.c File Reference . . . . .	136
6.108.1 Variable Documentation . . . . .	136
6.108.1.1 g_sFontCmss24i . . . . .	136
6.109grlib/fonts/fontcmss26.c File Reference . . . . .	136
6.109.1 Variable Documentation . . . . .	136
6.109.1.1 g_sFontCmss26 . . . . .	136
6.110grlib/fonts/fontcmss26b.c File Reference . . . . .	136
6.110.1 Variable Documentation . . . . .	137
6.110.1.1 g_sFontCmss26b . . . . .	137
6.111grlib/fonts/fontcmss26i.c File Reference . . . . .	137
6.111.1 Variable Documentation . . . . .	137
6.111.1.1 g_sFontCmss26i . . . . .	137
6.112grlib/fonts/fontcmss28.c File Reference . . . . .	137
6.112.1 Variable Documentation . . . . .	137
6.112.1.1 g_sFontCmss28 . . . . .	137

6.113grlib/fonts/fontcmss28b.c File Reference . . . . .	137
6.113.1 Variable Documentation . . . . .	137
6.113.1.1 g_sFontCmss28b . . . . .	137
6.114grlib/fonts/fontcmss28i.c File Reference . . . . .	138
6.114.1 Variable Documentation . . . . .	138
6.114.1.1 g_sFontCmss28i . . . . .	138
6.115grlib/fonts/fontcmss30.c File Reference . . . . .	138
6.115.1 Variable Documentation . . . . .	138
6.115.1.1 g_sFontCmss30 . . . . .	138
6.116grlib/fonts/fontcmss30b.c File Reference . . . . .	138
6.116.1 Variable Documentation . . . . .	138
6.116.1.1 g_sFontCmss30b . . . . .	138
6.117grlib/fonts/fontcmss30i.c File Reference . . . . .	138
6.117.1 Variable Documentation . . . . .	139
6.117.1.1 g_sFontCmss30i . . . . .	139
6.118grlib/fonts/fontcmss32.c File Reference . . . . .	139
6.118.1 Variable Documentation . . . . .	139
6.118.1.1 g_sFontCmss32 . . . . .	139
6.119grlib/fonts/fontcmss32b.c File Reference . . . . .	139
6.119.1 Variable Documentation . . . . .	139
6.119.1.1 g_sFontCmss32b . . . . .	139
6.120grlib/fonts/fontcmss32i.c File Reference . . . . .	139
6.120.1 Variable Documentation . . . . .	140
6.120.1.1 g_sFontCmss32i . . . . .	140
6.121grlib/fonts/fontcmss34.c File Reference . . . . .	140
6.121.1 Variable Documentation . . . . .	140
6.121.1.1 g_sFontCmss34 . . . . .	140
6.122grlib/fonts/fontcmss34b.c File Reference . . . . .	140
6.122.1 Variable Documentation . . . . .	140
6.122.1.1 g_sFontCmss34b . . . . .	140
6.123grlib/fonts/fontcmss34i.c File Reference . . . . .	140
6.123.1 Variable Documentation . . . . .	140
6.123.1.1 g_sFontCmss34i . . . . .	140
6.124grlib/fonts/fontcmss36.c File Reference . . . . .	141

6.124.1 Variable Documentation . . . . .	141
6.124.1.1 g_sFontCmss36 . . . . .	141
6.125grlib/fonts/fontcmss36b.c File Reference . . . . .	141
6.125.1 Variable Documentation . . . . .	141
6.125.1.1 g_sFontCmss36b . . . . .	141
6.126grlib/fonts/fontcmss36i.c File Reference . . . . .	141
6.126.1 Variable Documentation . . . . .	141
6.126.1.1 g_sFontCmss36i . . . . .	141
6.127grlib/fonts/fontcmss38.c File Reference . . . . .	141
6.127.1 Variable Documentation . . . . .	142
6.127.1.1 g_sFontCmss38 . . . . .	142
6.128grlib/fonts/fontcmss38b.c File Reference . . . . .	142
6.128.1 Variable Documentation . . . . .	142
6.128.1.1 g_sFontCmss38b . . . . .	142
6.129grlib/fonts/fontcmss38i.c File Reference . . . . .	142
6.129.1 Variable Documentation . . . . .	142
6.129.1.1 g_sFontCmss38i . . . . .	142
6.130grlib/fonts/fontcmss40.c File Reference . . . . .	142
6.130.1 Variable Documentation . . . . .	143
6.130.1.1 g_sFontCmss40 . . . . .	143
6.131grlib/fonts/fontcmss40b.c File Reference . . . . .	143
6.131.1 Variable Documentation . . . . .	143
6.131.1.1 g_sFontCmss40b . . . . .	143
6.132grlib/fonts/fontcmss40i.c File Reference . . . . .	143
6.132.1 Variable Documentation . . . . .	143
6.132.1.1 g_sFontCmss40i . . . . .	143
6.133grlib/fonts/fontcmss42.c File Reference . . . . .	143
6.133.1 Variable Documentation . . . . .	143
6.133.1.1 g_sFontCmss42 . . . . .	143
6.134grlib/fonts/fontcmss42b.c File Reference . . . . .	144
6.134.1 Variable Documentation . . . . .	144
6.134.1.1 g_sFontCmss42b . . . . .	144
6.135grlib/fonts/fontcmss42i.c File Reference . . . . .	144
6.135.1 Variable Documentation . . . . .	144

---

6.135.1.1 <code>g_sFontCmss42i</code>	144
6.136grlib/fonts/fontcmss44.c File Reference	144
6.136.1 Variable Documentation	144
6.136.1.1 <code>g_sFontCmss44</code>	144
6.137grlib/fonts/fontcmss44b.c File Reference	144
6.137.1 Variable Documentation	145
6.137.1.1 <code>g_sFontCmss44b</code>	145
6.138grlib/fonts/fontcmss44i.c File Reference	145
6.138.1 Variable Documentation	145
6.138.1.1 <code>g_sFontCmss44i</code>	145
6.139grlib/fonts/fontcmss46.c File Reference	145
6.139.1 Variable Documentation	145
6.139.1.1 <code>g_sFontCmss46</code>	145
6.140grlib/fonts/fontcmss46b.c File Reference	145
6.140.1 Variable Documentation	146
6.140.1.1 <code>g_sFontCmss46b</code>	146
6.141grlib/fonts/fontcmss46i.c File Reference	146
6.141.1 Variable Documentation	146
6.141.1.1 <code>g_sFontCmss46i</code>	146
6.142grlib/fonts/fontcmss48.c File Reference	146
6.142.1 Variable Documentation	146
6.142.1.1 <code>g_sFontCmss48</code>	146
6.143grlib/fonts/fontcmss48b.c File Reference	146
6.143.1 Variable Documentation	146
6.143.1.1 <code>g_sFontCmss48b</code>	146
6.144grlib/fonts/fontcmss48i.c File Reference	147
6.144.1 Variable Documentation	147
6.144.1.1 <code>g_sFontCmss48i</code>	147
6.145grlib/fonts/fontfixed6x8.c File Reference	147
6.145.1 Variable Documentation	147
6.145.1.1 <code>g_sFontFixed6x8</code>	147
6.146grlib/image.c File Reference	147
6.146.1 Function Documentation	147
6.146.1.1 <code>GrlImageDraw</code>	147

6.147grlib/imgbutton.c File Reference . . . . .	148
6.147.1 Function Documentation . . . . .	148
6.147.1.1 ImageButtonInit . . . . .	148
6.147.1.2 ImageButtonMsgProc . . . . .	149
6.148grlib/line.c File Reference . . . . .	149
6.148.1 Function Documentation . . . . .	149
6.148.1.1 GrLineDraw . . . . .	149
6.148.1.2 GrLineDrawH . . . . .	150
6.148.1.3 GrLineDrawV . . . . .	150
6.149grlib/listbox.c File Reference . . . . .	151
6.149.1 Define Documentation . . . . .	151
6.149.1.1 abs . . . . .	151
6.149.1.2 max . . . . .	151
6.149.1.3 min . . . . .	151
6.150grlib/offscr1bpp.c File Reference . . . . .	151
6.151grlib/offscr4bpp.c File Reference . . . . .	152
6.152grlib/offscr8bpp.c File Reference . . . . .	152
6.153grlib/pushbutton.c File Reference . . . . .	152
6.154grlib/radiobutton.c File Reference . . . . .	153
6.155grlib/rectangle.c File Reference . . . . .	153
6.155.1 Define Documentation . . . . .	153
6.155.1.1 max . . . . .	153
6.155.1.2 min . . . . .	153
6.155.2 Function Documentation . . . . .	153
6.155.2.1 GrRectDraw . . . . .	153
6.155.2.2 GrRectFill . . . . .	154
6.155.2.3 GrRectIntersectGet . . . . .	154
6.155.2.4 GrRectOverlapCheck . . . . .	155
6.156grlib/slider.c File Reference . . . . .	155
6.156.1 Define Documentation . . . . .	155
6.156.1.1 max . . . . .	155
6.156.1.2 min . . . . .	155
6.156.2 Function Documentation . . . . .	155
6.156.2.1 SliderInit . . . . .	155

6.156.2.2 SliderMsgProc . . . . .	156
6.157 grlib/string.c File Reference . . . . .	156
6.157.1 Define Documentation . . . . .	157
6.157.1.1 ABSENT_CHAR_REPLACEMENT . . . . .	157
6.157.1.2 SC_FLAG_COMPRESSED . . . . .	157
6.157.1.3 SC_GET_INDEX . . . . .	157
6.157.1.4 SC_GET_LEN . . . . .	157
6.157.1.5 SC_GET_OFF . . . . .	157
6.157.1.6 SC_IS_NULL . . . . .	157
6.157.1.7 SC_MAX_INDEX . . . . .	157
6.157.1.8 SC_OFFSET_M . . . . .	157
6.157.2 Function Documentation . . . . .	157
6.157.2.1 GrStringDraw . . . . .	157
6.157.2.2 GrStringGet . . . . .	158
6.157.2.3 GrStringLanguageSet . . . . .	158
6.157.2.4 GrStringTableSet . . . . .	159
6.157.2.5 GrStringWidthGet . . . . .	159
6.157.2.6 NumLeadingZeros . . . . .	160
6.158 grlib/widget.c File Reference . . . . .	160
6.158.1 Define Documentation . . . . .	161
6.158.1.1 MQ_FLAG_POST_ORDER . . . . .	161
6.158.1.2 MQ_FLAG_STOP_ON_SUCCESS . . . . .	161
6.158.1.3 QUEUE_SIZE . . . . .	161
6.158.2 Function Documentation . . . . .	161
6.158.2.1 WidgetAdd . . . . .	161
6.158.2.2 WidgetDefaultMsgProc . . . . .	161
6.158.2.3 WidgetMessageQueueAdd . . . . .	162
6.158.2.4 WidgetMessageQueueProcess . . . . .	162
6.158.2.5 WidgetMessageSendPostOrder . . . . .	163
6.158.2.6 WidgetMessageSendPreOrder . . . . .	163
6.158.2.7 WidgetMutexGet . . . . .	164
6.158.2.8 WidgetMutexInit . . . . .	164
6.158.2.9 WidgetMutexPut . . . . .	165
6.158.2.10 WidgetPointerMessage . . . . .	165

6.158.2.11WidgetRemove . . . . .	166
6.158.3 Variable Documentation . . . . .	166
6.158.3.1 g_sRoot . . . . .	166
6.159 include/build/lpc210x.h File Reference . . . . .	166
6.159.1 Define Documentation . . . . .	171
6.159.1.1 ALDOM . . . . .	171
6.159.1.2 ALDOW . . . . .	171
6.159.1.3 ALDOY . . . . .	171
6.159.1.4 ALHOUR . . . . .	171
6.159.1.5 ALMIN . . . . .	171
6.159.1.6 ALMON . . . . .	171
6.159.1.7 ALSEC . . . . .	171
6.159.1.8 ALYEAR . . . . .	171
6.159.1.9 AMR . . . . .	171
6.159.1.10CCR . . . . .	171
6.159.1.11CIIR . . . . .	171
6.159.1.12CTC . . . . .	171
6.159.1.13CTIME0 . . . . .	171
6.159.1.14CTIME1 . . . . .	171
6.159.1.15CTIME2 . . . . .	171
6.159.1.16DOM . . . . .	171
6.159.1.17DOW . . . . .	171
6.159.1.18DOY . . . . .	171
6.159.1.19EXTINT . . . . .	171
6.159.1.20EXTWAKE . . . . .	171
6.159.1.21HOUR . . . . .	171
6.159.1.22I2C_I2ADR . . . . .	171
6.159.1.23I2C_I2CONCLR . . . . .	171
6.159.1.24I2C_I2CONSET . . . . .	172
6.159.1.25I2C_I2DAT . . . . .	172
6.159.1.26I2C_I2SCLH . . . . .	172
6.159.1.27I2C_I2SCLL . . . . .	172
6.159.1.28I2C_I2STAT . . . . .	172
6.159.1.29LR . . . . .	172

6.159.1.30OCLR . . . . .	172
6.159.1.31ODIR . . . . .	172
6.159.1.32OPIN . . . . .	172
6.159.1.33OSET . . . . .	172
6.159.1.34MAMCR . . . . .	172
6.159.1.35MAMMAP . . . . .	172
6.159.1.36MAMTIM . . . . .	172
6.159.1.37MIN . . . . .	172
6.159.1.38MONTH . . . . .	172
6.159.1.39PCON . . . . .	172
6.159.1.40PCONP . . . . .	172
6.159.1.41PINSEL0 . . . . .	172
6.159.1.42PINSEL1 . . . . .	172
6.159.1.43PLLCFG . . . . .	172
6.159.1.44PLLCON . . . . .	172
6.159.1.45PLLFEED . . . . .	172
6.159.1.46PLLSTAT . . . . .	172
6.159.1.47PREFRAC . . . . .	172
6.159.1.48PREINT . . . . .	173
6.159.1.49PWM_CCR . . . . .	173
6.159.1.50PWM_CR0 . . . . .	173
6.159.1.51PWM_CR1 . . . . .	173
6.159.1.52PWM_CR2 . . . . .	173
6.159.1.53PWM_CR3 . . . . .	173
6.159.1.54PWM_EMR . . . . .	173
6.159.1.55PWM_IR . . . . .	173
6.159.1.56PWM_LER . . . . .	173
6.159.1.57PWM_MCR . . . . .	173
6.159.1.58PWM_MR0 . . . . .	173
6.159.1.59PWM_MR1 . . . . .	173
6.159.1.60PWM_MR2 . . . . .	173
6.159.1.61PWM_MR3 . . . . .	173
6.159.1.62PWM_MR4 . . . . .	173
6.159.1.63PWM_MR5 . . . . .	173

6.159.1.64PWM_MR6 . . . . .	173
6.159.1.65PWM_PC . . . . .	173
6.159.1.66PWM_PCR . . . . .	173
6.159.1.67PWM_PR . . . . .	173
6.159.1.68PWM_TC . . . . .	173
6.159.1.69PWM_TCR . . . . .	173
6.159.1.70SEC . . . . .	173
6.159.1.71SPI_SPCCR . . . . .	173
6.159.1.72SPI_SPCR . . . . .	174
6.159.1.73SPI_SPDR . . . . .	174
6.159.1.74SPI_SPIINT . . . . .	174
6.159.1.75SPI_SPSR . . . . .	174
6.159.1.76SPI_SPTCR . . . . .	174
6.159.1.77SPI_SPTOR . . . . .	174
6.159.1.78SPI_SPTSR . . . . .	174
6.159.1.79T0CCR . . . . .	174
6.159.1.80T0CR0 . . . . .	174
6.159.1.81T0CR1 . . . . .	174
6.159.1.82T0CR2 . . . . .	174
6.159.1.83T0CR3 . . . . .	174
6.159.1.84T0EMR . . . . .	174
6.159.1.85T0IR . . . . .	174
6.159.1.86T0MCR . . . . .	174
6.159.1.87T0MR0 . . . . .	174
6.159.1.88T0MR1 . . . . .	174
6.159.1.89T0MR2 . . . . .	174
6.159.1.90T0MR3 . . . . .	174
6.159.1.91T0PC . . . . .	174
6.159.1.92T0PR . . . . .	174
6.159.1.93T0TC . . . . .	174
6.159.1.94T0TCR . . . . .	174
6.159.1.95T1CCR . . . . .	174
6.159.1.96T1CR0 . . . . .	175
6.159.1.97T1CR1 . . . . .	175

6.159.1.98T1CR2 . . . . .	175
6.159.1.99T1CR3 . . . . .	175
6.159.1.1001EMR . . . . .	175
6.159.1.1011IR . . . . .	175
6.159.1.1021MCR . . . . .	175
6.159.1.1031MR0 . . . . .	175
6.159.1.1041MR1 . . . . .	175
6.159.1.1051MR2 . . . . .	175
6.159.1.1061MR3 . . . . .	175
6.159.1.1071PC . . . . .	175
6.159.1.1081PR . . . . .	175
6.159.1.1091TC . . . . .	175
6.159.1.1101TCR . . . . .	175
6.159.1.1110DLL . . . . .	175
6.159.1.1120DLM . . . . .	175
6.159.1.1130FCR . . . . .	175
6.159.1.1140IER . . . . .	175
6.159.1.1150IIR . . . . .	175
6.159.1.1160LCR . . . . .	175
6.159.1.1170LSR . . . . .	175
6.159.1.1180MCR . . . . .	175
6.159.1.1190MSR . . . . .	175
6.159.1.1200RBR . . . . .	176
6.159.1.1210SCR . . . . .	176
6.159.1.1220THR . . . . .	176
6.159.1.1231DLL . . . . .	176
6.159.1.1241DLM . . . . .	176
6.159.1.1251FCR . . . . .	176
6.159.1.1261IER . . . . .	176
6.159.1.1271IIR . . . . .	176
6.159.1.1281LCR . . . . .	176
6.159.1.1291LSR . . . . .	176
6.159.1.1301MCR . . . . .	176
6.159.1.1311MSR . . . . .	176

6.159.1.1301RBR . . . . .	176
6.159.1.1301SCR . . . . .	176
6.159.1.1301THR . . . . .	176
6.159.1.1301ICDefVectAddr . . . . .	176
6.159.1.1301ICFIQStatus . . . . .	176
6.159.1.1301ICIntEnable . . . . .	176
6.159.1.1301ICIntEnClr . . . . .	176
6.159.1.1301ICIntSelect . . . . .	176
6.159.1.1401CIRQStatus . . . . .	176
6.159.1.1401VICProtection . . . . .	176
6.159.1.1401ICRawIntr . . . . .	176
6.159.1.1401ICSoflnt . . . . .	176
6.159.1.1401ICSoftIntClr . . . . .	177
6.159.1.1401ICVectAddr . . . . .	177
6.159.1.1401ICVectAddr0 . . . . .	177
6.159.1.1401ICVectAddr1 . . . . .	177
6.159.1.1401ICVectAddr10 . . . . .	177
6.159.1.1401ICVectAddr11 . . . . .	177
6.159.1.1501ICVectAddr12 . . . . .	177
6.159.1.1501ICVectAddr13 . . . . .	177
6.159.1.1501ICVectAddr14 . . . . .	177
6.159.1.1501ICVectAddr15 . . . . .	177
6.159.1.1501ICVectAddr2 . . . . .	177
6.159.1.1501ICVectAddr3 . . . . .	177
6.159.1.1501ICVectAddr4 . . . . .	177
6.159.1.1501ICVectAddr5 . . . . .	177
6.159.1.1501ICVectAddr6 . . . . .	177
6.159.1.1501ICVectAddr7 . . . . .	177
6.159.1.1601ICVectAddr8 . . . . .	177
6.159.1.1601ICVectAddr9 . . . . .	177
6.159.1.1601ICVectCtl0 . . . . .	177
6.159.1.1601ICVectCtl1 . . . . .	177
6.159.1.1601ICVectCtl10 . . . . .	177
6.159.1.1601ICVectCtl11 . . . . .	177

6.159.1.16 <sup>8</sup> ICVectCntl12 . . . . .	177
6.159.1.16 <sup>7</sup> ICVectCntl13 . . . . .	177
6.159.1.16 <sup>8</sup> ICVectCntl14 . . . . .	178
6.159.1.16 <sup>0</sup> ICVectCntl15 . . . . .	178
6.159.1.17 <sup>0</sup> ICVectCntl2 . . . . .	178
6.159.1.17 <sup>1</sup> ICVectCntl3 . . . . .	178
6.159.1.17 <sup>2</sup> ICVectCntl4 . . . . .	178
6.159.1.17 <sup>3</sup> ICVectCntl5 . . . . .	178
6.159.1.17 <sup>4</sup> ICVectCntl6 . . . . .	178
6.159.1.17 <sup>5</sup> ICVectCntl7 . . . . .	178
6.159.1.17 <sup>6</sup> ICVectCntl8 . . . . .	178
6.159.1.17 <sup>7</sup> ICVectCntl9 . . . . .	178
6.159.1.17 <sup>8</sup> PBDIV . . . . .	178
6.159.1.17 <sup>0</sup> DFEED . . . . .	178
6.159.1.18 <sup>0</sup> DMOD . . . . .	178
6.159.1.18 <sup>1</sup> WDTC . . . . .	178
6.159.1.18 <sup>2</sup> WDTV . . . . .	178
6.159.1.18 <sup>3</sup> EAR . . . . .	178
6.160include/build/lpc213x.h File Reference . . . . .	178
6.160.1 Define Documentation . . . . .	183
6.160.1.1 AD0CR . . . . .	183
6.160.1.2 AD0DR . . . . .	183
6.160.1.3 AD1CR . . . . .	183
6.160.1.4 AD1DR . . . . .	183
6.160.1.5 ALDOM . . . . .	183
6.160.1.6 ALDOW . . . . .	183
6.160.1.7 ALDOY . . . . .	183
6.160.1.8 ALHOUR . . . . .	183
6.160.1.9 ALMIN . . . . .	184
6.160.1.10ALMON . . . . .	184
6.160.1.11ALSEC . . . . .	184
6.160.1.12ALYEAR . . . . .	184
6.160.1.13AMR . . . . .	184
6.160.1.14CCR . . . . .	184

6.160.1.15CIIR . . . . .	184
6.160.1.16CSPR . . . . .	184
6.160.1.17CTC . . . . .	184
6.160.1.18CTIME0 . . . . .	184
6.160.1.19CTIME1 . . . . .	184
6.160.1.20CTIME2 . . . . .	184
6.160.1.21DACR . . . . .	184
6.160.1.22DOM . . . . .	184
6.160.1.23DOW . . . . .	184
6.160.1.24DOY . . . . .	184
6.160.1.25EXINT . . . . .	184
6.160.1.26EXTINT . . . . .	184
6.160.1.27EXTMODE . . . . .	184
6.160.1.28EXTPOLAR . . . . .	184
6.160.1.29FIO0CLR . . . . .	184
6.160.1.30FIO0DIR . . . . .	184
6.160.1.31FIO0MASK . . . . .	184
6.160.1.32FIO0PIN . . . . .	184
6.160.1.33FIO0SET . . . . .	185
6.160.1.34FIO1CLR . . . . .	185
6.160.1.35FIO1DIR . . . . .	185
6.160.1.36FIO1MASK . . . . .	185
6.160.1.37FIO1PIN . . . . .	185
6.160.1.38FIO1SET . . . . .	185
6.160.1.39HOUR . . . . .	185
6.160.1.40I20ADR . . . . .	185
6.160.1.41I20CONCLR . . . . .	185
6.160.1.42I20CONSET . . . . .	185
6.160.1.43I20DAT . . . . .	185
6.160.1.44I20SCLH . . . . .	185
6.160.1.45I20SCLL . . . . .	185
6.160.1.46I20STAT . . . . .	185
6.160.1.47I21ADR . . . . .	185
6.160.1.48I21CONCLR . . . . .	185

6.160.1.49	CONSET	185
6.160.1.50	21DAT	185
6.160.1.51	121SCLH	185
6.160.1.52	21SCLL	185
6.160.1.53	21STAT	185
6.160.1.54	LR	185
6.160.1.55	NTWAKE	185
6.160.1.56	OCLR0	185
6.160.1.57	OCLR1	186
6.160.1.58	ODIR0	186
6.160.1.59	ODIR1	186
6.160.1.60	OPIN0	186
6.160.1.61	OPIN1	186
6.160.1.62	OSET0	186
6.160.1.63	OSET1	186
6.160.1.64	MAMCR	186
6.160.1.65	MAMTIM	186
6.160.1.66	MEMMAP	186
6.160.1.67	MIN	186
6.160.1.68	MONTH	186
6.160.1.69	PCON	186
6.160.1.70	PCONP	186
6.160.1.71	PINSEL0	186
6.160.1.72	PINSEL1	186
6.160.1.73	PINSEL2	186
6.160.1.74	PLLCFG	186
6.160.1.75	PLLCON	186
6.160.1.76	PLLFEED	186
6.160.1.77	PLLSTAT	186
6.160.1.78	PREFRAC	186
6.160.1.79	PREINT	186
6.160.1.80	PWMEMR	186
6.160.1.81	PWMIR	187
6.160.1.82	PWMLER	187

6.160.1.83PWMMCR . . . . .	187
6.160.1.84PWMMR0 . . . . .	187
6.160.1.85PWMMR1 . . . . .	187
6.160.1.86PWMMR2 . . . . .	187
6.160.1.87PWMMR3 . . . . .	187
6.160.1.88PWMMR4 . . . . .	187
6.160.1.89PWMMR5 . . . . .	187
6.160.1.90PWMMR6 . . . . .	187
6.160.1.91PWMPC . . . . .	187
6.160.1.92PWMPCR . . . . .	187
6.160.1.93PWMPR . . . . .	187
6.160.1.94PWMTC . . . . .	187
6.160.1.95PWMTCR . . . . .	187
6.160.1.96RSID . . . . .	187
6.160.1.97S0SPCCR . . . . .	187
6.160.1.98S0SPCR . . . . .	187
6.160.1.99S0SPDR . . . . .	187
6.160.1.10S0SPINT . . . . .	187
6.160.1.10S0SPSR . . . . .	187
6.160.1.10S0SPTCR . . . . .	187
6.160.1.10S0SPTOR . . . . .	187
6.160.1.10S0SPTSR . . . . .	187
6.160.1.10SEC . . . . .	188
6.160.1.10SPCPSR . . . . .	188
6.160.1.10SSPCR0 . . . . .	188
6.160.1.10SSPCR1 . . . . .	188
6.160.1.10SSPDMACR . . . . .	188
6.160.1.11SSPDR . . . . .	188
6.160.1.11SSPICR . . . . .	188
6.160.1.11SSPIMSC . . . . .	188
6.160.1.11SSPMIS . . . . .	188
6.160.1.11SSPRIS . . . . .	188
6.160.1.11SSPSR . . . . .	188
6.160.1.11T0CCR . . . . .	188

6.160.1.1170CR0 . . . . .	188
6.160.1.1180CR1 . . . . .	188
6.160.1.1190CR2 . . . . .	188
6.160.1.1200CR3 . . . . .	188
6.160.1.1210CTCR . . . . .	188
6.160.1.1220EMR . . . . .	188
6.160.1.1230IR . . . . .	188
6.160.1.1240MCR . . . . .	188
6.160.1.1250MR0 . . . . .	188
6.160.1.1260MR1 . . . . .	188
6.160.1.1270MR2 . . . . .	188
6.160.1.1280MR3 . . . . .	188
6.160.1.1290PC . . . . .	189
6.160.1.1300PR . . . . .	189
6.160.1.1310TC . . . . .	189
6.160.1.1320TCR . . . . .	189
6.160.1.1331CCR . . . . .	189
6.160.1.1341CR0 . . . . .	189
6.160.1.1351CR1 . . . . .	189
6.160.1.1361CR2 . . . . .	189
6.160.1.1371CR3 . . . . .	189
6.160.1.1381CTCR . . . . .	189
6.160.1.1391EMR . . . . .	189
6.160.1.1401IR . . . . .	189
6.160.1.1411MCR . . . . .	189
6.160.1.1421MR0 . . . . .	189
6.160.1.1431MR1 . . . . .	189
6.160.1.1441MR2 . . . . .	189
6.160.1.1451MR3 . . . . .	189
6.160.1.1461PC . . . . .	189
6.160.1.1471PR . . . . .	189
6.160.1.1481TC . . . . .	189
6.160.1.1491TCR . . . . .	189
6.160.1.1500DLL . . . . .	189

6.160.1.150DLM . . . . .	189
6.160.1.150FCR . . . . .	189
6.160.1.150IER . . . . .	190
6.160.1.150IIR . . . . .	190
6.160.1.150LCR . . . . .	190
6.160.1.150LSR . . . . .	190
6.160.1.150RBR . . . . .	190
6.160.1.150SCR . . . . .	190
6.160.1.150TER . . . . .	190
6.160.1.1600THR . . . . .	190
6.160.1.1601DLL . . . . .	190
6.160.1.1601DLM . . . . .	190
6.160.1.1601FCR . . . . .	190
6.160.1.1601IER . . . . .	190
6.160.1.1601IIR . . . . .	190
6.160.1.1601LCR . . . . .	190
6.160.1.1601LSR . . . . .	190
6.160.1.1601MCR . . . . .	190
6.160.1.1601MSR . . . . .	190
6.160.1.1701RBR . . . . .	190
6.160.1.1701SCR . . . . .	190
6.160.1.1701TER . . . . .	190
6.160.1.1701THR . . . . .	190
6.160.1.170ICDefVectAddr . . . . .	190
6.160.1.170ICFIQStatus . . . . .	190
6.160.1.170ICIntEnable . . . . .	190
6.160.1.170ICIntEnClr . . . . .	191
6.160.1.170ICIntSelect . . . . .	191
6.160.1.170ICIRQStatus . . . . .	191
6.160.1.180ICProtection . . . . .	191
6.160.1.180ICRawIntr . . . . .	191
6.160.1.180ICSoftInt . . . . .	191
6.160.1.180ICSoftIntClr . . . . .	191
6.160.1.180ICVectAddr . . . . .	191

6.160.1.18 <del>0</del> ICVectAddr0 . . . . .	191
6.160.1.18 <del>0</del> ICVectAddr1 . . . . .	191
6.160.1.18 <del>7</del> ICVectAddr10 . . . . .	191
6.160.1.18 <del>0</del> ICVectAddr11 . . . . .	191
6.160.1.18 <del>0</del> ICVectAddr12 . . . . .	191
6.160.1.19 <del>0</del> ICVectAddr13 . . . . .	191
6.160.1.19 <del>7</del> ICVectAddr14 . . . . .	191
6.160.1.19 <del>2</del> ICVectAddr15 . . . . .	191
6.160.1.19 <del>0</del> ICVectAddr2 . . . . .	191
6.160.1.19 <del>4</del> ICVectAddr3 . . . . .	191
6.160.1.19 <del>0</del> ICVectAddr4 . . . . .	191
6.160.1.19 <del>0</del> ICVectAddr5 . . . . .	191
6.160.1.19 <del>7</del> ICVectAddr6 . . . . .	191
6.160.1.19 <del>0</del> ICVectAddr7 . . . . .	191
6.160.1.19 <del>0</del> ICVectAddr8 . . . . .	191
6.160.1.20 <del>0</del> ICVectAddr9 . . . . .	191
6.160.1.20 <del>7</del> ICVectCtl0 . . . . .	192
6.160.1.20 <del>2</del> ICVectCtl1 . . . . .	192
6.160.1.20 <del>0</del> ICVectCtl10 . . . . .	192
6.160.1.20 <del>4</del> ICVectCtl11 . . . . .	192
6.160.1.20 <del>0</del> ICVectCtl12 . . . . .	192
6.160.1.20 <del>6</del> ICVectCtl13 . . . . .	192
6.160.1.20 <del>7</del> ICVectCtl14 . . . . .	192
6.160.1.20 <del>8</del> ICVectCtl15 . . . . .	192
6.160.1.20 <del>0</del> ICVectCtl2 . . . . .	192
6.160.1.21 <del>0</del> ICVectCtl3 . . . . .	192
6.160.1.21 <del>7</del> ICVectCtl4 . . . . .	192
6.160.1.21 <del>2</del> ICVectCtl5 . . . . .	192
6.160.1.21 <del>0</del> ICVectCtl6 . . . . .	192
6.160.1.21 <del>4</del> ICVectCtl7 . . . . .	192
6.160.1.21 <del>8</del> ICVectCtl8 . . . . .	192
6.160.1.21 <del>6</del> ICVectCtl9 . . . . .	192
6.160.1.21 <del>7</del> PBDIV . . . . .	192
6.160.1.21 <del>0</del> DFEED . . . . .	192

6.160.1.21WDMOD . . . . .	192
6.160.1.22WDTC . . . . .	192
6.160.1.22WDTV . . . . .	192
6.160.1.22WEAR . . . . .	192
6.161include/build/lpc23xx.h File Reference . . . . .	192
6.161.1 Define Documentation . . . . .	217
6.161.1.1 __LPC23xx_H . . . . .	217
6.161.1.2 AD0_BASE_ADDR . . . . .	217
6.161.1.3 AD0CR . . . . .	217
6.161.1.4 AD0DR0 . . . . .	217
6.161.1.5 AD0DR1 . . . . .	217
6.161.1.6 AD0DR2 . . . . .	217
6.161.1.7 AD0DR3 . . . . .	217
6.161.1.8 AD0DR4 . . . . .	217
6.161.1.9 AD0DR5 . . . . .	217
6.161.1.10AD0DR6 . . . . .	217
6.161.1.11AD0DR7 . . . . .	217
6.161.1.12AD0GDR . . . . .	217
6.161.1.13AD0INTEN . . . . .	217
6.161.1.14AD0STAT . . . . .	217
6.161.1.15AHBCFG1 . . . . .	217
6.161.1.16AHBCFG2 . . . . .	217
6.161.1.17CAN1_BASE_ADDR . . . . .	217
6.161.1.18CAN1BTR . . . . .	217
6.161.1.19CAN1CMR . . . . .	217
6.161.1.20CAN1EWL . . . . .	217
6.161.1.21CAN1GSR . . . . .	217
6.161.1.22CAN1ICR . . . . .	217
6.161.1.23CAN1IER . . . . .	217
6.161.1.24CAN1MOD . . . . .	218
6.161.1.25CAN1RDA . . . . .	218
6.161.1.26CAN1RDB . . . . .	218
6.161.1.27CAN1RFS . . . . .	218
6.161.1.28CAN1RID . . . . .	218

6.161.1.29CAN1SR . . . . .	218
6.161.1.30CAN1TDA1 . . . . .	218
6.161.1.31CAN1TDA2 . . . . .	218
6.161.1.32CAN1TDA3 . . . . .	218
6.161.1.33CAN1TDB1 . . . . .	218
6.161.1.34CAN1TDB2 . . . . .	218
6.161.1.35CAN1TDB3 . . . . .	218
6.161.1.36CAN1TFI1 . . . . .	218
6.161.1.37CAN1TFI2 . . . . .	218
6.161.1.38CAN1TFI3 . . . . .	218
6.161.1.39CAN1TID1 . . . . .	218
6.161.1.40CAN1TID2 . . . . .	218
6.161.1.41CAN1TID3 . . . . .	218
6.161.1.42CAN2_BASE_ADDR . . . . .	218
6.161.1.43CAN2BTR . . . . .	218
6.161.1.44CAN2CMR . . . . .	218
6.161.1.45CAN2EWL . . . . .	218
6.161.1.46CAN2GSR . . . . .	218
6.161.1.47CAN2ICR . . . . .	218
6.161.1.48CAN2IER . . . . .	219
6.161.1.49CAN2MOD . . . . .	219
6.161.1.50CAN2RDA . . . . .	219
6.161.1.51CAN2RDB . . . . .	219
6.161.1.52CAN2RFS . . . . .	219
6.161.1.53CAN2RID . . . . .	219
6.161.1.54CAN2SR . . . . .	219
6.161.1.55CAN2TDA1 . . . . .	219
6.161.1.56CAN2TDA2 . . . . .	219
6.161.1.57CAN2TDA3 . . . . .	219
6.161.1.58CAN2TDB1 . . . . .	219
6.161.1.59CAN2TDB2 . . . . .	219
6.161.1.60CAN2TDB3 . . . . .	219
6.161.1.61CAN2TFI1 . . . . .	219
6.161.1.62CAN2TFI2 . . . . .	219

6.161.1.63CAN2TFI3 . . . . .	219
6.161.1.64CAN2TID1 . . . . .	219
6.161.1.65CAN2TID2 . . . . .	219
6.161.1.66CAN2TID3 . . . . .	219
6.161.1.67CAN_ACCEPT_BASE_ADDR . . . . .	219
6.161.1.68CAN_AFMR . . . . .	219
6.161.1.69CAN_CENTRAL_BASE_ADDR . . . . .	219
6.161.1.70CAN_EFF_GRP_SA . . . . .	219
6.161.1.71CAN_EFF_SA . . . . .	220
6.161.1.72CAN_EOT . . . . .	220
6.161.1.73CAN_LUT_ERR . . . . .	220
6.161.1.74CAN_LUT_ERR_ADR . . . . .	220
6.161.1.75CAN_MSR . . . . .	220
6.161.1.76CAN_RX_SR . . . . .	220
6.161.1.77CAN_SFF_GRP_SA . . . . .	220
6.161.1.78CAN_SFF_SA . . . . .	220
6.161.1.79CAN_TX_SR . . . . .	220
6.161.1.80CCLKCFG . . . . .	220
6.161.1.81CLKSRCSEL . . . . .	220
6.161.1.82CMD_CODE . . . . .	220
6.161.1.83CMD_DATA . . . . .	220
6.161.1.84CSPR . . . . .	220
6.161.1.85DAC_BASE_ADDR . . . . .	220
6.161.1.86DPCR . . . . .	220
6.161.1.87DEV_INT_CLR . . . . .	220
6.161.1.88DEV_INT_EN . . . . .	220
6.161.1.89DEV_INT_PRIO . . . . .	220
6.161.1.90DEV_INT_SET . . . . .	221
6.161.1.91DEV_INT_STAT . . . . .	221
6.161.1.92DMA_BASE_ADDR . . . . .	221
6.161.1.93DMA_INT_EN . . . . .	221
6.161.1.94DMA_INT_STAT . . . . .	221
6.161.1.95DMA_REQ_CLR . . . . .	221
6.161.1.96DMA_REQ_SET . . . . .	221

6.161.1.97DMA_REQ_STAT . . . . .	221
6.161.1.98DYNAMIC_MEM0_BASE . . . . .	221
6.161.1.99DYNAMIC_MEM1_BASE . . . . .	221
6.161.1.100DYNAMIC_MEM2_BASE . . . . .	221
6.161.1.101DYNAMIC_MEM3_BASE . . . . .	221
6.161.1.102BMC_BASE_ADDR . . . . .	221
6.161.1.103BMC_CONFIG . . . . .	221
6.161.1.104BMC_CTRL . . . . .	221
6.161.1.105BMC_DYN_APP . . . . .	221
6.161.1.106BMC_DYN_CFG0 . . . . .	221
6.161.1.107BMC_DYN_CFG1 . . . . .	221
6.161.1.108BMC_DYN_CFG2 . . . . .	221
6.161.1.109BMC_DYN_CFG3 . . . . .	222
6.161.1.110BMC_DYN_CTRL . . . . .	222
6.161.1.111BMC_DYN_DAL . . . . .	222
6.161.1.112BMC_DYN_MRД . . . . .	222
6.161.1.113BMC_DYN_RAS . . . . .	222
6.161.1.114BMC_DYN_RASCAS0 . . . . .	222
6.161.1.115BMC_DYN_RASCAS1 . . . . .	222
6.161.1.116BMC_DYN_RASCAS2 . . . . .	222
6.161.1.117BMC_DYN_RASCAS3 . . . . .	222
6.161.1.118BMC_DYN_RC . . . . .	222
6.161.1.119BMC_DYN_RD_CFG . . . . .	222
6.161.1.120BMC_DYN_RFC . . . . .	222
6.161.1.121BMC_DYN_RFSH . . . . .	222
6.161.1.122BMC_DYN_RP . . . . .	222
6.161.1.123BMC_DYN_RRD . . . . .	222
6.161.1.124BMC_DYN_SREX . . . . .	222
6.161.1.125BMC_DYN_WR . . . . .	223
6.161.1.126BMC_DYN_XSR . . . . .	223
6.161.1.127BMC_STA_CFG0 . . . . .	223
6.161.1.128BMC_STA_CFG1 . . . . .	223
6.161.1.129BMC_STA_CFG2 . . . . .	223
6.161.1.130BMC_STA_CFG3 . . . . .	223

6.161.1.13 <sup>1</sup> MC_STA_EXT_WAIT . . . . .	223
6.161.1.13 <sup>2</sup> MC_STA_WAITOEN0 . . . . .	223
6.161.1.13 <sup>3</sup> MC_STA_WAITOEN1 . . . . .	223
6.161.1.13 <sup>4</sup> MC_STA_WAITOEN2 . . . . .	223
6.161.1.13 <sup>5</sup> MC_STA_WAITOEN3 . . . . .	223
6.161.1.13 <sup>6</sup> MC_STA_WAITPAGE0 . . . . .	223
6.161.1.13 <sup>7</sup> MC_STA_WAITPAGE1 . . . . .	223
6.161.1.13 <sup>8</sup> MC_STA_WAITPAGE2 . . . . .	223
6.161.1.13 <sup>9</sup> MC_STA_WAITPAGE3 . . . . .	223
6.161.1.14 <sup>10</sup> MC_STA_WAITRD0 . . . . .	223
6.161.1.14 <sup>11</sup> MC_STA_WAITRD1 . . . . .	224
6.161.1.14 <sup>12</sup> MC_STA_WAITRD2 . . . . .	224
6.161.1.14 <sup>13</sup> MC_STA_WAITRD3 . . . . .	224
6.161.1.14 <sup>14</sup> MC_STA_WAITTURN0 . . . . .	224
6.161.1.14 <sup>15</sup> MC_STA_WAITTURN1 . . . . .	224
6.161.1.14 <sup>16</sup> MC_STA_WAITTURN2 . . . . .	224
6.161.1.14 <sup>17</sup> MC_STA_WAITTURN3 . . . . .	224
6.161.1.14 <sup>18</sup> MC_STA_WAITWEN0 . . . . .	224
6.161.1.14 <sup>19</sup> MC_STA_WAITWEN1 . . . . .	224
6.161.1.15 <sup>20</sup> MC_STA_WAITWEN2 . . . . .	224
6.161.1.15 <sup>21</sup> MC_STA_WAITWEN3 . . . . .	224
6.161.1.15 <sup>22</sup> MC_STA_WAITWR0 . . . . .	224
6.161.1.15 <sup>23</sup> MC_STA_WAITWR1 . . . . .	224
6.161.1.15 <sup>24</sup> MC_STA_WAITWR2 . . . . .	224
6.161.1.15 <sup>25</sup> MC_STA_WAITWR3 . . . . .	224
6.161.1.15 <sup>26</sup> MC_STAT . . . . .	224
6.161.1.15 <sup>27</sup> OT_INT_CLR . . . . .	225
6.161.1.15 <sup>28</sup> OT_INT_SET . . . . .	225
6.161.1.15 <sup>29</sup> OT_INT_STAT . . . . .	225
6.161.1.16 <sup>30</sup> P_DMA_DIS . . . . .	225
6.161.1.16 <sup>31</sup> P_DMA_EN . . . . .	225
6.161.1.16 <sup>32</sup> P_DMA_STAT . . . . .	225
6.161.1.16 <sup>33</sup> P_INDEX . . . . .	225
6.161.1.16 <sup>34</sup> P_INT_CLR . . . . .	225

6.161.1.16 <sup>5</sup> P_INT_EN . . . . .	225
6.161.1.16 <sup>6</sup> P_INT_PRIO . . . . .	225
6.161.1.16 <sup>7</sup> P_INT_SET . . . . .	225
6.161.1.16 <sup>8</sup> P_INT_STAT . . . . .	225
6.161.1.16 <sup>9</sup> XTINT . . . . .	225
6.161.1.17 <sup>0</sup> XTMODE . . . . .	225
6.161.1.17 <sup>1</sup> XTPOLAR . . . . .	225
6.161.1.17 <sup>2</sup> IO0CLR . . . . .	225
6.161.1.17 <sup>3</sup> IO0CLR0 . . . . .	225
6.161.1.17 <sup>4</sup> IO0CLR1 . . . . .	225
6.161.1.17 <sup>5</sup> IO0CLR2 . . . . .	225
6.161.1.17 <sup>6</sup> IO0CLR3 . . . . .	225
6.161.1.17 <sup>7</sup> IO0CLRL . . . . .	225
6.161.1.17 <sup>8</sup> IO0CLRU . . . . .	226
6.161.1.17 <sup>9</sup> IO0DIR . . . . .	226
6.161.1.18 <sup>0</sup> IO0DIR0 . . . . .	226
6.161.1.18 <sup>1</sup> IO0DIR1 . . . . .	226
6.161.1.18 <sup>2</sup> IO0DIR2 . . . . .	226
6.161.1.18 <sup>3</sup> IO0DIR3 . . . . .	226
6.161.1.18 <sup>4</sup> IO0DIRL . . . . .	226
6.161.1.18 <sup>5</sup> IO0DIRU . . . . .	226
6.161.1.18 <sup>6</sup> IO0MASK . . . . .	226
6.161.1.18 <sup>7</sup> IO0MASK0 . . . . .	226
6.161.1.18 <sup>8</sup> IO0MASK1 . . . . .	226
6.161.1.18 <sup>9</sup> IO0MASK2 . . . . .	226
6.161.1.19 <sup>0</sup> IO0MASK3 . . . . .	226
6.161.1.19 <sup>1</sup> IO0MASKL . . . . .	226
6.161.1.19 <sup>2</sup> IO0MASKU . . . . .	226
6.161.1.19 <sup>3</sup> IO0PIN . . . . .	226
6.161.1.19 <sup>4</sup> IO0PINO . . . . .	226
6.161.1.19 <sup>5</sup> IO0PIN1 . . . . .	226
6.161.1.19 <sup>6</sup> IO0PIN2 . . . . .	226
6.161.1.19 <sup>7</sup> IO0PIN3 . . . . .	226
6.161.1.19 <sup>8</sup> IO0PINL . . . . .	226

6.161.1.19IO0PINU . . . . .	226
6.161.1.20IO0SET . . . . .	226
6.161.1.20IO0SET0 . . . . .	226
6.161.1.20IO0SET1 . . . . .	227
6.161.1.20IO0SET2 . . . . .	227
6.161.1.20IO0SET3 . . . . .	227
6.161.1.20IO0SETL . . . . .	227
6.161.1.20IO0SETU . . . . .	227
6.161.1.20IO1CLR . . . . .	227
6.161.1.20IO1CLR0 . . . . .	227
6.161.1.20IO1CLR1 . . . . .	227
6.161.1.21IO1CLR2 . . . . .	227
6.161.1.21IO1CLR3 . . . . .	227
6.161.1.21IO1CLRL . . . . .	227
6.161.1.21IO1CLRU . . . . .	227
6.161.1.21IO1DIR . . . . .	227
6.161.1.21IO1DIR0 . . . . .	227
6.161.1.21IO1DIR1 . . . . .	227
6.161.1.21IO1DIR2 . . . . .	227
6.161.1.21IO1DIR3 . . . . .	227
6.161.1.21IO1DIRL . . . . .	227
6.161.1.22IO1DIRU . . . . .	227
6.161.1.22IO1MASK . . . . .	227
6.161.1.22IO1MASK0 . . . . .	227
6.161.1.22IO1MASK1 . . . . .	227
6.161.1.22IO1MASK2 . . . . .	227
6.161.1.22IO1MASK3 . . . . .	227
6.161.1.22IO1MASKL . . . . .	228
6.161.1.22IO1MASKU . . . . .	228
6.161.1.22IO1PIN . . . . .	228
6.161.1.22IO1PIN0 . . . . .	228
6.161.1.23IO1PIN1 . . . . .	228
6.161.1.23IO1PIN2 . . . . .	228
6.161.1.23IO1PIN3 . . . . .	228

6.161.1.23 <del>5</del> IO1PINL . . . . .	228
6.161.1.23 <del>4</del> IO1PINU . . . . .	228
6.161.1.23 <del>5</del> IO1SET . . . . .	228
6.161.1.23 <del>6</del> IO1SET0 . . . . .	228
6.161.1.23 <del>7</del> IO1SET1 . . . . .	228
6.161.1.23 <del>8</del> IO1SET2 . . . . .	228
6.161.1.23 <del>9</del> IO1SET3 . . . . .	228
6.161.1.24 <del>0</del> IO1SETL . . . . .	228
6.161.1.24 <del>1</del> IO1SETU . . . . .	228
6.161.1.24 <del>2</del> IO2CLR . . . . .	228
6.161.1.24 <del>3</del> IO2CLR0 . . . . .	228
6.161.1.24 <del>4</del> IO2CLR1 . . . . .	228
6.161.1.24 <del>5</del> IO2CLR2 . . . . .	228
6.161.1.24 <del>6</del> IO2CLR3 . . . . .	228
6.161.1.24 <del>7</del> IO2CLRL . . . . .	228
6.161.1.24 <del>8</del> IO2CLRU . . . . .	228
6.161.1.24 <del>9</del> IO2DIR . . . . .	228
6.161.1.25 <del>0</del> IO2DIR0 . . . . .	229
6.161.1.25 <del>1</del> IO2DIR1 . . . . .	229
6.161.1.25 <del>2</del> IO2DIR2 . . . . .	229
6.161.1.25 <del>3</del> IO2DIR3 . . . . .	229
6.161.1.25 <del>4</del> IO2DIRL . . . . .	229
6.161.1.25 <del>5</del> IO2DIRU . . . . .	229
6.161.1.25 <del>6</del> IO2MASK . . . . .	229
6.161.1.25 <del>7</del> IO2MASK0 . . . . .	229
6.161.1.25 <del>8</del> IO2MASK1 . . . . .	229
6.161.1.25 <del>9</del> IO2MASK2 . . . . .	229
6.161.1.26 <del>0</del> IO2MASK3 . . . . .	229
6.161.1.26 <del>1</del> IO2MASKL . . . . .	229
6.161.1.26 <del>2</del> IO2MASKU . . . . .	229
6.161.1.26 <del>3</del> IO2PIN . . . . .	229
6.161.1.26 <del>4</del> IO2PIN0 . . . . .	229
6.161.1.26 <del>5</del> IO2PIN1 . . . . .	229
6.161.1.26 <del>6</del> IO2PIN2 . . . . .	229

6.161.1.267IO2PIN3 . . . . .	229
6.161.1.268IO2PINL . . . . .	229
6.161.1.269IO2PINU . . . . .	229
6.161.1.270IO2SET . . . . .	229
6.161.1.271IO2SET0 . . . . .	229
6.161.1.272IO2SET1 . . . . .	229
6.161.1.273IO2SET2 . . . . .	229
6.161.1.274IO2SET3 . . . . .	230
6.161.1.275IO2SETL . . . . .	230
6.161.1.276IO2SETU . . . . .	230
6.161.1.277IO3CLR . . . . .	230
6.161.1.278IO3CLR0 . . . . .	230
6.161.1.279IO3CLR1 . . . . .	230
6.161.1.280IO3CLR2 . . . . .	230
6.161.1.281IO3CLR3 . . . . .	230
6.161.1.282IO3CLRL . . . . .	230
6.161.1.283IO3CLRU . . . . .	230
6.161.1.284IO3DIR . . . . .	230
6.161.1.285IO3DIR0 . . . . .	230
6.161.1.286IO3DIR1 . . . . .	230
6.161.1.287IO3DIR2 . . . . .	230
6.161.1.288IO3DIR3 . . . . .	230
6.161.1.289IO3DIRL . . . . .	230
6.161.1.290IO3DIRU . . . . .	230
6.161.1.291IO3MASK . . . . .	230
6.161.1.292IO3MASK0 . . . . .	230
6.161.1.293IO3MASK1 . . . . .	230
6.161.1.294IO3MASK2 . . . . .	230
6.161.1.295IO3MASK3 . . . . .	230
6.161.1.296IO3MASKL . . . . .	230
6.161.1.297IO3MASKU . . . . .	230
6.161.1.298IO3PIN . . . . .	231
6.161.1.299IO3PIN0 . . . . .	231
6.161.1.300IO3PIN1 . . . . .	231

6.161.1.30 <del>I</del> IO3PIN2	231
6.161.1.30 <del>I</del> IO3PIN3	231
6.161.1.30 <del>I</del> IO3PINL	231
6.161.1.30 <del>I</del> IO3PINU	231
6.161.1.30 <del>I</del> IO3SET	231
6.161.1.30 <del>I</del> IO3SET0	231
6.161.1.30 <del>I</del> IO3SET1	231
6.161.1.30 <del>I</del> IO3SET2	231
6.161.1.30 <del>I</del> IO3SET3	231
6.161.1.31 <del>I</del> IO3SETL	231
6.161.1.31 <del>I</del> IO3SETU	231
6.161.1.31 <del>I</del> IO4CLR	231
6.161.1.31 <del>I</del> IO4CLR0	231
6.161.1.31 <del>I</del> IO4CLR1	231
6.161.1.31 <del>I</del> IO4CLR2	231
6.161.1.31 <del>I</del> IO4CLR3	231
6.161.1.31 <del>I</del> IO4CLRL	231
6.161.1.31 <del>I</del> IO4CLRU	231
6.161.1.31 <del>I</del> IO4DIR	231
6.161.1.32 <del>I</del> IO4DIR0	231
6.161.1.32 <del>I</del> IO4DIR1	231
6.161.1.32 <del>I</del> IO4DIR2	232
6.161.1.32 <del>I</del> IO4DIR3	232
6.161.1.32 <del>I</del> IO4DIRL	232
6.161.1.32 <del>I</del> IO4DIRU	232
6.161.1.32 <del>I</del> IO4MASK	232
6.161.1.32 <del>I</del> IO4MASK0	232
6.161.1.32 <del>I</del> IO4MASK1	232
6.161.1.32 <del>I</del> IO4MASK2	232
6.161.1.32 <del>I</del> IO4MASK3	232
6.161.1.33 <del>I</del> IO4MASKL	232
6.161.1.33 <del>I</del> IO4MASKU	232
6.161.1.33 <del>I</del> IO4PIN	232
6.161.1.33 <del>I</del> IO4PIN0	232

---

**CONTENTS**

6.161.1.335IO4PIN1 . . . . .	232
6.161.1.336IO4PIN2 . . . . .	232
6.161.1.337IO4PIN3 . . . . .	232
6.161.1.338IO4PINL . . . . .	232
6.161.1.339IO4PINU . . . . .	232
6.161.1.340IO4SET . . . . .	232
6.161.1.341IO4SET0 . . . . .	232
6.161.1.342IO4SET1 . . . . .	232
6.161.1.343IO4SET2 . . . . .	232
6.161.1.344IO4SET3 . . . . .	232
6.161.1.345IO4SETL . . . . .	232
6.161.1.346IO4SETU . . . . .	233
6.161.1.347IO_BASE_ADDR . . . . .	233
6.161.1.348PDMA_CH0_CFG . . . . .	233
6.161.1.349PDMA_CH0_CTRL . . . . .	233
6.161.1.350PDMA_CH0_DEST . . . . .	233
6.161.1.350PDMA_CH0_LLI . . . . .	233
6.161.1.350PDMA_CH0_SRC . . . . .	233
6.161.1.353PDMA_CH1_CFG . . . . .	233
6.161.1.353PDMA_CH1_CTRL . . . . .	233
6.161.1.355PDMA_CH1_DEST . . . . .	233
6.161.1.356PDMA_CH1_LLI . . . . .	233
6.161.1.356PDMA_CH1_SRC . . . . .	233
6.161.1.358PDMA_CONFIG . . . . .	233
6.161.1.359PDMA_ENABLED_CHNS . . . . .	233
6.161.1.360PDMA_INT_ERR_CLR . . . . .	233
6.161.1.360PDMA_INT_ERR_STAT . . . . .	233
6.161.1.360PDMA_INT_STAT . . . . .	233
6.161.1.363PDMA_INT_TCCLR . . . . .	234
6.161.1.363PDMA_INT_TCSTAT . . . . .	234
6.161.1.365PDMA_RAW_INT_ERR_STAT . . . . .	234
6.161.1.366PDMA_RAW_INT_TCSTAT . . . . .	234
6.161.1.367PDMA_SOFT_BREQ . . . . .	234
6.161.1.368PDMA_SOFT_LBREQ . . . . .	234

6.161.1.369PDMA_SOFT_LSREQ . . . . .	234
6.161.1.370PDMA_SOFT_SREQ . . . . .	234
6.161.1.371PDMA_SYNC . . . . .	234
6.161.1.372PIO_BASE_ADDR . . . . .	234
6.161.1.373BC_BULK_CUR_ED . . . . .	234
6.161.1.374BC_BULK_HEAD_ED . . . . .	234
6.161.1.375BC_CMD_STAT . . . . .	234
6.161.1.376BC_CONTROL . . . . .	234
6.161.1.377BC_CTRL_CUR_ED . . . . .	234
6.161.1.378BC_CTRL_HEAD_ED . . . . .	234
6.161.1.379BC_DONE_HEAD . . . . .	235
6.161.1.380BC_FM_INTERVAL . . . . .	235
6.161.1.381BC_FM_NUMBER . . . . .	235
6.161.1.382BC_FM_REMAINING . . . . .	235
6.161.1.383BC_HCCA . . . . .	235
6.161.1.384BC_INT_DIS . . . . .	235
6.161.1.385BC_INT_EN . . . . .	235
6.161.1.386BC_INT_STAT . . . . .	235
6.161.1.387BC_LS THRHL . . . . .	235
6.161.1.388BC_PERIOD_CUR_ED . . . . .	235
6.161.1.389BC_PERIOD_START . . . . .	235
6.161.1.390BC_REVISION . . . . .	235
6.161.1.391BC_RH_DESCA . . . . .	235
6.161.1.392BC_RH_DESCB . . . . .	235
6.161.1.393BC_RH_PORT_STAT1 . . . . .	235
6.161.1.394BC_RH_PORT_STAT2 . . . . .	235
6.161.1.395BC_RH_STAT . . . . .	236
6.161.1.396ADR . . . . .	236
6.161.1.397CONCLR . . . . .	236
6.161.1.398CONSET . . . . .	236
6.161.1.399DAT . . . . .	236
6.161.1.400SCLH . . . . .	236
6.161.1.401SCLL . . . . .	236
6.161.1.402STAT . . . . .	236

6.161.1.40 <del>1</del> 1ADR . . . . .	236
6.161.1.40 <del>1</del> 1CONCLR . . . . .	236
6.161.1.40 <del>1</del> 1CONSET . . . . .	236
6.161.1.40 <del>1</del> 1DAT . . . . .	236
6.161.1.40 <del>1</del> 1SCLH . . . . .	236
6.161.1.40 <del>1</del> 1SCLL . . . . .	236
6.161.1.40 <del>1</del> 1STAT . . . . .	236
6.161.1.41 <del>1</del> 2ADR . . . . .	236
6.161.1.41 <del>1</del> 2CONCLR . . . . .	236
6.161.1.41 <del>1</del> 2CONSET . . . . .	236
6.161.1.41 <del>1</del> 2DAT . . . . .	236
6.161.1.41 <del>1</del> 2SCLH . . . . .	236
6.161.1.41 <del>1</del> 2SCLL . . . . .	236
6.161.1.41 <del>1</del> 2STAT . . . . .	236
6.161.1.41 <del>1</del> C0_BASE_ADDR . . . . .	236
6.161.1.41 <del>1</del> C1_BASE_ADDR . . . . .	237
6.161.1.41 <del>1</del> C2_BASE_ADDR . . . . .	237
6.161.1.42 <del>1</del> S_BASE_ADDR . . . . .	237
6.161.1.42 <del>1</del> S_DAI . . . . .	237
6.161.1.42 <del>1</del> S.DAO . . . . .	237
6.161.1.42 <del>1</del> S.DMA1 . . . . .	237
6.161.1.42 <del>1</del> S.DMA2 . . . . .	237
6.161.1.42 <del>1</del> S.IRQ . . . . .	237
6.161.1.42 <del>1</del> S.RX_FIFO . . . . .	237
6.161.1.42 <del>1</del> S.RXRATE . . . . .	237
6.161.1.42 <del>1</del> S.STATE . . . . .	237
6.161.1.42 <del>1</del> S.TX_FIFO . . . . .	237
6.161.1.43 <del>1</del> S.TXRATE . . . . .	237
6.161.1.43 <del>1</del> NTWAKE . . . . .	237
6.161.1.43 <del>1</del> 00_INT_CLR . . . . .	237
6.161.1.43 <del>1</del> 00_INT_EN_F . . . . .	237
6.161.1.43 <del>1</del> 00_INT_EN_R . . . . .	237
6.161.1.43 <del>1</del> 00_INT_STAT_F . . . . .	237
6.161.1.43 <del>1</del> 00_INT_STAT_R . . . . .	237

6.161.1.43 <del>02</del> _INT_CLR . . . . .	237
6.161.1.43 <del>02</del> _INT_EN_F . . . . .	237
6.161.1.43 <del>02</del> _INT_EN_R . . . . .	238
6.161.1.44 <del>02</del> _INT_STAT_F . . . . .	238
6.161.1.44 <del>02</del> _INT_STAT_R . . . . .	238
6.161.1.44 <del>02</del> _INT_STAT . . . . .	238
6.161.1.44 <del>02</del> _CLR0 . . . . .	238
6.161.1.44 <del>02</del> _CLR1 . . . . .	238
6.161.1.44 <del>02</del> _DIR0 . . . . .	238
6.161.1.44 <del>02</del> _DIR1 . . . . .	238
6.161.1.44 <del>02</del> _PIN0 . . . . .	238
6.161.1.44 <del>02</del> _PIN1 . . . . .	238
6.161.1.44 <del>02</del> _SET0 . . . . .	238
6.161.1.45 <del>02</del> _SET1 . . . . .	238
6.161.1.45 <del>02</del> _MAC_BASE_ADDR . . . . .	238
6.161.1.45 <del>02</del> _MAC_CLRT . . . . .	238
6.161.1.45 <del>02</del> _MAC_COMMAND . . . . .	238
6.161.1.45 <del>02</del> _MAC_FLOWCONTROLCNT . . . . .	238
6.161.1.45 <del>02</del> _MAC_FLOWCONTROLSTS . . . . .	238
6.161.1.45 <del>02</del> _MAC_HASHFILTERH . . . . .	238
6.161.1.45 <del>02</del> _MAC_HASHFILTERL . . . . .	238
6.161.1.45 <del>02</del> _MAC_INTCLEAR . . . . .	239
6.161.1.45 <del>02</del> _MAC_INTENABLE . . . . .	239
6.161.1.46 <del>02</del> _MAC_INTSET . . . . .	239
6.161.1.46 <del>02</del> _MAC_INTSTATUS . . . . .	239
6.161.1.46 <del>02</del> _MAC_IPGR . . . . .	239
6.161.1.46 <del>02</del> _MAC_IPGT . . . . .	239
6.161.1.46 <del>02</del> _MAC_MAC1 . . . . .	239
6.161.1.46 <del>02</del> _MAC_MAC2 . . . . .	239
6.161.1.46 <del>02</del> _MAC_MADR . . . . .	239
6.161.1.46 <del>02</del> _MAC_MAXF . . . . .	239
6.161.1.46 <del>02</del> _MAC_MCFG . . . . .	239
6.161.1.46 <del>02</del> _MAC_MCMD . . . . .	239
6.161.1.47 <del>02</del> _MAC_MIND . . . . .	239

6.161.1.47MAC_MODULEID . . . . .	239
6.161.1.47MAC_MRDD . . . . .	239
6.161.1.47MAC_MWTD . . . . .	239
6.161.1.47MAC_POWERDOWN . . . . .	240
6.161.1.47MAC_RSV . . . . .	240
6.161.1.47MAC_RXCONSUMEINDEX . . . . .	240
6.161.1.47MAC_RXDESCRIPTOR . . . . .	240
6.161.1.47MAC_RXDESCRIPTORMNUM . . . . .	240
6.161.1.47MAC_RXFILTERCTRL . . . . .	240
6.161.1.48MAC_RXFILTERWOLCLR . . . . .	240
6.161.1.48MAC_RXFILTERWOLSTS . . . . .	240
6.161.1.48MAC_RXPRODUCEINDEX . . . . .	240
6.161.1.48MAC_RXSTATUS . . . . .	240
6.161.1.48MAC_SA0 . . . . .	240
6.161.1.48MAC_SA1 . . . . .	240
6.161.1.48MAC_SA2 . . . . .	240
6.161.1.48MAC_STATUS . . . . .	240
6.161.1.48MAC_SUPP . . . . .	240
6.161.1.48MAC_TEST . . . . .	240
6.161.1.49MAC_TSV0 . . . . .	241
6.161.1.49MAC_TSV1 . . . . .	241
6.161.1.49MAC_TXCONSUMEINDEX . . . . .	241
6.161.1.49MAC_TXDESCRIPTOR . . . . .	241
6.161.1.49MAC_TXDESCRIPTORMNUM . . . . .	241
6.161.1.49MAC_TXPRODUCEINDEX . . . . .	241
6.161.1.49MAC_TXSTATUS . . . . .	241
6.161.1.49MAMCR . . . . .	241
6.161.1.49MAMTIM . . . . .	241
6.161.1.49MAXPACKET_SIZE . . . . .	241
6.161.1.50MCI_ARGUMENT . . . . .	241
6.161.1.50MCI_BASE_ADDR . . . . .	241
6.161.1.50MCI_CLEAR . . . . .	241
6.161.1.50MCI_CLOCK . . . . .	241
6.161.1.50MCI_COMMAND . . . . .	241

6.161.1.50 <u>MCI_DATA_CNT</u>	241
6.161.1.50 <u>MCI_DATA_CTRL</u>	241
6.161.1.50 <u>MCI_DATA_LEN</u>	241
6.161.1.50 <u>MCI_DATA_TMR</u>	242
6.161.1.50 <u>MCI_FIFO</u>	242
6.161.1.51 <u>MCI_FIFO_CNT</u>	242
6.161.1.51 <u>MCI_MASK0</u>	242
6.161.1.51 <u>MCI_MASK1</u>	242
6.161.1.51 <u>MCI_POWER</u>	242
6.161.1.51 <u>MCI_RESP0</u>	242
6.161.1.51 <u>MCI_RESP1</u>	242
6.161.1.51 <u>MCI_RESP2</u>	242
6.161.1.51 <u>MCI_RESP3</u>	242
6.161.1.51 <u>MCI_RESP_CMD</u>	242
6.161.1.51 <u>MCI_STATUS</u>	242
6.161.1.52 <u>MEMMAP</u>	242
6.161.1.52 <u>REQ_INT_CLR</u>	242
6.161.1.52 <u>REQ_INT_SET</u>	242
6.161.1.52 <u>REQ_INT_STAT</u>	242
6.161.1.52 <u>TG_CLK_CTRL</u>	242
6.161.1.52 <u>TG_CLK_STAT</u>	242
6.161.1.52 <u>TG_I2C_CLKHI</u>	242
6.161.1.52 <u>TG_I2C_CLKLO</u>	243
6.161.1.52 <u>TG_I2C_CTL</u>	243
6.161.1.52 <u>TG_I2C_RX</u>	243
6.161.1.53 <u>TG_I2C_STS</u>	243
6.161.1.53 <u>TG_I2C_TX</u>	243
6.161.1.53 <u>TG_INT_CLR</u>	243
6.161.1.53 <u>TG_INT_EN</u>	243
6.161.1.53 <u>TG_INT_SET</u>	243
6.161.1.53 <u>TG_INT_STAT</u>	243
6.161.1.53 <u>TG_STAT_CTRL</u>	243
6.161.1.53 <u>TG_TIMER</u>	243
6.161.1.53 <u>BARTCFG</u>	243

6.161.1.539ARTCFG_BASE_ADDR . . . . .	243
6.161.1.540CLKSEL0 . . . . .	243
6.161.1.541CLKSEL1 . . . . .	243
6.161.1.542CON . . . . .	243
6.161.1.543CONP . . . . .	243
6.161.1.544INMODE0 . . . . .	243
6.161.1.545INMODE1 . . . . .	244
6.161.1.546INMODE2 . . . . .	244
6.161.1.547INMODE3 . . . . .	244
6.161.1.548INMODE4 . . . . .	244
6.161.1.549INMODE5 . . . . .	244
6.161.1.550INMODE6 . . . . .	244
6.161.1.551INMODE7 . . . . .	244
6.161.1.552INMODE8 . . . . .	244
6.161.1.553INMODE9 . . . . .	244
6.161.1.554INSEL0 . . . . .	244
6.161.1.555INSEL1 . . . . .	244
6.161.1.556INSEL10 . . . . .	244
6.161.1.557INSEL2 . . . . .	244
6.161.1.558INSEL3 . . . . .	244
6.161.1.559INSEL4 . . . . .	244
6.161.1.560INSEL5 . . . . .	244
6.161.1.561INSEL6 . . . . .	244
6.161.1.562INSEL7 . . . . .	244
6.161.1.563INSEL8 . . . . .	244
6.161.1.564INSEL9 . . . . .	244
6.161.1.565INSEL_BASE_ADDR . . . . .	245
6.161.1.566LLCFG . . . . .	245
6.161.1.567LLCON . . . . .	245
6.161.1.568LLFEED . . . . .	245
6.161.1.569LLSTAT . . . . .	245
6.161.1.570WM0_BASE_ADDR . . . . .	245
6.161.1.571WM0CCR . . . . .	245
6.161.1.572WM0CR0 . . . . .	245

6.161.1.57 <sup>B</sup> WM0CR1 . . . . .	245
6.161.1.57 <sup>D</sup> WM0CR2 . . . . .	245
6.161.1.57 <sup>E</sup> WM0CR3 . . . . .	245
6.161.1.57 <sup>F</sup> WM0CTCR . . . . .	245
6.161.1.57 <sup>F</sup> WM0EMR . . . . .	245
6.161.1.57 <sup>G</sup> WM0IR . . . . .	245
6.161.1.57 <sup>H</sup> WM0LER . . . . .	245
6.161.1.58 <sup>B</sup> WM0MCR . . . . .	245
6.161.1.58 <sup>B</sup> WM0MR0 . . . . .	245
6.161.1.58 <sup>C</sup> WM0MR1 . . . . .	245
6.161.1.58 <sup>B</sup> WM0MR2 . . . . .	245
6.161.1.58 <sup>C</sup> WM0MR3 . . . . .	245
6.161.1.58 <sup>B</sup> WM0MR4 . . . . .	245
6.161.1.58 <sup>B</sup> WM0MR5 . . . . .	245
6.161.1.58 <sup>F</sup> WM0MR6 . . . . .	245
6.161.1.58 <sup>B</sup> WM0PC . . . . .	246
6.161.1.58 <sup>H</sup> WM0PCR . . . . .	246
6.161.1.59 <sup>B</sup> WM0PR . . . . .	246
6.161.1.59 <sup>F</sup> WM0TC . . . . .	246
6.161.1.59 <sup>C</sup> WM0TCR . . . . .	246
6.161.1.59 <sup>B</sup> WM1_BASE_ADDR . . . . .	246
6.161.1.59 <sup>C</sup> WM1CCR . . . . .	246
6.161.1.59 <sup>B</sup> WM1CR0 . . . . .	246
6.161.1.59 <sup>B</sup> WM1CR1 . . . . .	246
6.161.1.59 <sup>F</sup> WM1CR2 . . . . .	246
6.161.1.59 <sup>B</sup> WM1CR3 . . . . .	246
6.161.1.59 <sup>H</sup> WM1CTCR . . . . .	246
6.161.1.60 <sup>B</sup> WM1EMR . . . . .	246
6.161.1.60 <sup>F</sup> WM1IR . . . . .	246
6.161.1.60 <sup>C</sup> WM1LER . . . . .	246
6.161.1.60 <sup>B</sup> WM1MCR . . . . .	246
6.161.1.60 <sup>C</sup> WM1MR0 . . . . .	246
6.161.1.60 <sup>B</sup> WM1MR1 . . . . .	246
6.161.1.60 <sup>B</sup> WM1MR2 . . . . .	246

6.161.1.60 <sup>7</sup> WM1MR3 . . . . .	246
6.161.1.60 <sup>8</sup> WM1MR4 . . . . .	246
6.161.1.60 <sup>9</sup> WM1MR5 . . . . .	246
6.161.1.61 <sup>0</sup> WM1MR6 . . . . .	246
6.161.1.61 <sup>1</sup> WM1PC . . . . .	247
6.161.1.61 <sup>2</sup> WM1PCR . . . . .	247
6.161.1.61 <sup>3</sup> WM1PR . . . . .	247
6.161.1.61 <sup>4</sup> WM1TC . . . . .	247
6.161.1.61 <sup>5</sup> WM1TCR . . . . .	247
6.161.1.61 <sup>6</sup> REALIZE_EP . . . . .	247
6.161.1.61 <sup>7</sup> SIR . . . . .	247
6.161.1.61 <sup>8</sup> TC_ALDOM . . . . .	247
6.161.1.61 <sup>9</sup> TC_ALDOW . . . . .	247
6.161.1.62 <sup>0</sup> TC_ALDOY . . . . .	247
6.161.1.62 <sup>1</sup> TC_ALHOUR . . . . .	247
6.161.1.62 <sup>2</sup> TC_ALMIN . . . . .	247
6.161.1.62 <sup>3</sup> TC_ALMON . . . . .	247
6.161.1.62 <sup>4</sup> TC_ALSEC . . . . .	247
6.161.1.62 <sup>5</sup> TC_ALYEAR . . . . .	247
6.161.1.62 <sup>6</sup> TC_AMR . . . . .	247
6.161.1.62 <sup>7</sup> TC_BASE_ADDR . . . . .	247
6.161.1.62 <sup>8</sup> TC_CCR . . . . .	247
6.161.1.62 <sup>9</sup> TC_CIIR . . . . .	247
6.161.1.63 <sup>0</sup> TC_CISS . . . . .	247
6.161.1.63 <sup>1</sup> TC_CTC . . . . .	247
6.161.1.63 <sup>2</sup> TC_CTIME0 . . . . .	247
6.161.1.63 <sup>3</sup> TC_CTIME1 . . . . .	247
6.161.1.63 <sup>4</sup> TC_CTIME2 . . . . .	248
6.161.1.63 <sup>5</sup> TC_DOM . . . . .	248
6.161.1.63 <sup>6</sup> TC_DOW . . . . .	248
6.161.1.63 <sup>7</sup> TC_DOY . . . . .	248
6.161.1.63 <sup>8</sup> TC_HOUR . . . . .	248
6.161.1.63 <sup>9</sup> TC_ILR . . . . .	248
6.161.1.64 <sup>0</sup> TC_MIN . . . . .	248

6.161.1.64RTC_MONTH . . . . .	248
6.161.1.64RTC_PREFRAC . . . . .	248
6.161.1.64BTC_PREINT . . . . .	248
6.161.1.64RTC_SEC . . . . .	248
6.161.1.64BTC_YEAR . . . . .	248
6.161.1.64BX_DATA . . . . .	248
6.161.1.64RX_PLENGTH . . . . .	248
6.161.1.64SPCCR . . . . .	248
6.161.1.64SPPCR . . . . .	248
6.161.1.65SPDR . . . . .	248
6.161.1.65SPINT . . . . .	248
6.161.1.65SPSR . . . . .	248
6.161.1.65SCB_BASE_ADDR . . . . .	248
6.161.1.65CS . . . . .	248
6.161.1.65SPI0_BASE_ADDR . . . . .	248
6.161.1.65SP0_BASE_ADDR . . . . .	248
6.161.1.65SP0CPSR . . . . .	249
6.161.1.65SP0CR0 . . . . .	249
6.161.1.65SP0CR1 . . . . .	249
6.161.1.66SP0DMACR . . . . .	249
6.161.1.66SP0DR . . . . .	249
6.161.1.66SP0ICR . . . . .	249
6.161.1.66SP0IMSC . . . . .	249
6.161.1.66SP0MIS . . . . .	249
6.161.1.66SP0RIS . . . . .	249
6.161.1.66SP0SR . . . . .	249
6.161.1.67SP1_BASE_ADDR . . . . .	249
6.161.1.67SP1CPSR . . . . .	249
6.161.1.67SP1CR0 . . . . .	249
6.161.1.67SP1CR1 . . . . .	249
6.161.1.67SP1DMACR . . . . .	249
6.161.1.67SP1DR . . . . .	249
6.161.1.67SP1ICR . . . . .	249
6.161.1.67SP1IMSC . . . . .	249

6.161.1.67\$SP1MIS . . . . .	249
6.161.1.67\$SP1RIS . . . . .	249
6.161.1.67\$SP1SR . . . . .	249
6.161.1.67\$STATIC_MEM0_BASE . . . . .	249
6.161.1.67\$STATIC_MEM1_BASE . . . . .	249
6.161.1.68\$STATIC_MEM2_BASE . . . . .	250
6.161.1.68\$STATIC_MEM3_BASE . . . . .	250
6.161.1.68\$SYS_ERR_INT_CLR . . . . .	250
6.161.1.68\$SYS_ERR_INT_SET . . . . .	250
6.161.1.68\$SYS_ERR_INT_STAT . . . . .	250
6.161.1.68\$CCR . . . . .	250
6.161.1.68\$CR0 . . . . .	250
6.161.1.68\$CR1 . . . . .	250
6.161.1.68\$CR2 . . . . .	250
6.161.1.68\$CR3 . . . . .	250
6.161.1.69\$CTCR . . . . .	250
6.161.1.69\$EMR . . . . .	250
6.161.1.69\$IR . . . . .	250
6.161.1.69\$MCR . . . . .	250
6.161.1.69\$MR0 . . . . .	250
6.161.1.69\$MR1 . . . . .	250
6.161.1.69\$MR2 . . . . .	250
6.161.1.69\$MR3 . . . . .	250
6.161.1.69\$PC . . . . .	250
6.161.1.69\$PR . . . . .	250
6.161.1.70\$TC . . . . .	250
6.161.1.70\$TCR . . . . .	250
6.161.1.70\$CCR . . . . .	251
6.161.1.70\$B1CR0 . . . . .	251
6.161.1.70\$B1CR1 . . . . .	251
6.161.1.70\$B1CR2 . . . . .	251
6.161.1.70\$B1CR3 . . . . .	251
6.161.1.70\$V1CTCR . . . . .	251
6.161.1.70\$B1EMR . . . . .	251

6.161.1.70 <del>0</del> 1IR . . . . .	251
6.161.1.71 <del>0</del> 1MCR . . . . .	251
6.161.1.71 <del>1</del> 1MR0 . . . . .	251
6.161.1.71 <del>1</del> 1MR1 . . . . .	251
6.161.1.71 <del>1</del> 1MR2 . . . . .	251
6.161.1.71 <del>1</del> 1MR3 . . . . .	251
6.161.1.71 <del>1</del> 1PC . . . . .	251
6.161.1.71 <del>1</del> 1PR . . . . .	251
6.161.1.71 <del>1</del> 1TC . . . . .	251
6.161.1.71 <del>1</del> 1TCR . . . . .	251
6.161.1.71 <del>2</del> 2CCR . . . . .	251
6.161.1.72 <del>0</del> 2CR0 . . . . .	251
6.161.1.72 <del>0</del> 2CR1 . . . . .	251
6.161.1.72 <del>0</del> 2CR2 . . . . .	251
6.161.1.72 <del>0</del> 2CR3 . . . . .	251
6.161.1.72 <del>1</del> 2CTCR . . . . .	251
6.161.1.72 <del>5</del> 2EMR . . . . .	251
6.161.1.72 <del>6</del> 2IR . . . . .	252
6.161.1.72 <del>7</del> 2MCR . . . . .	252
6.161.1.72 <del>8</del> 2MR0 . . . . .	252
6.161.1.72 <del>8</del> 2MR1 . . . . .	252
6.161.1.73 <del>0</del> 2MR2 . . . . .	252
6.161.1.73 <del>1</del> 2MR3 . . . . .	252
6.161.1.73 <del>2</del> 2PC . . . . .	252
6.161.1.73 <del>3</del> 2PR . . . . .	252
6.161.1.73 <del>4</del> 2TC . . . . .	252
6.161.1.73 <del>5</del> 2TCR . . . . .	252
6.161.1.73 <del>6</del> 3CCR . . . . .	252
6.161.1.73 <del>7</del> 3CR0 . . . . .	252
6.161.1.73 <del>8</del> 3CR1 . . . . .	252
6.161.1.73 <del>9</del> 3CR2 . . . . .	252
6.161.1.74 <del>0</del> 3CR3 . . . . .	252
6.161.1.74 <del>1</del> 3CTCR . . . . .	252
6.161.1.74 <del>2</del> 3EMR . . . . .	252

6.161.1.74B3IR . . . . .	252
6.161.1.74T3MCR . . . . .	252
6.161.1.74B3MR0 . . . . .	252
6.161.1.74B3MR1 . . . . .	252
6.161.1.74T3MR2 . . . . .	252
6.161.1.74B3MR3 . . . . .	252
6.161.1.74B3PC . . . . .	252
6.161.1.75T3PR . . . . .	253
6.161.1.75T3TC . . . . .	253
6.161.1.75T3TCR . . . . .	253
6.161.1.75BMR0_BASE_ADDR . . . . .	253
6.161.1.75BMR1_BASE_ADDR . . . . .	253
6.161.1.75BMR2_BASE_ADDR . . . . .	253
6.161.1.75BMR3_BASE_ADDR . . . . .	253
6.161.1.75TX_DATA . . . . .	253
6.161.1.75BX_PLENGTH . . . . .	253
6.161.1.7590ACR . . . . .	253
6.161.1.7600DLL . . . . .	253
6.161.1.7610DLM . . . . .	253
6.161.1.7620FCR . . . . .	253
6.161.1.7630FDR . . . . .	253
6.161.1.7640ICR . . . . .	253
6.161.1.7650IER . . . . .	253
6.161.1.7660IIR . . . . .	253
6.161.1.7670LCR . . . . .	253
6.161.1.7680LSR . . . . .	253
6.161.1.7690RBR . . . . .	253
6.161.1.7700SCR . . . . .	253
6.161.1.7710TER . . . . .	253
6.161.1.7720THR . . . . .	253
6.161.1.7731ACR . . . . .	254
6.161.1.7741DLL . . . . .	254
6.161.1.7751DLM . . . . .	254
6.161.1.7761FCR . . . . .	254

6.161.1.7701FDR . . . . .	254
6.161.1.7701IER . . . . .	254
6.161.1.7701IIR . . . . .	254
6.161.1.7801LCR . . . . .	254
6.161.1.7801LSR . . . . .	254
6.161.1.7801MCR . . . . .	254
6.161.1.7801MSR . . . . .	254
6.161.1.7801RBR . . . . .	254
6.161.1.7801SCR . . . . .	254
6.161.1.7801TER . . . . .	254
6.161.1.7801THR . . . . .	254
6.161.1.7802ACR . . . . .	254
6.161.1.7802DLL . . . . .	254
6.161.1.7902DLM . . . . .	254
6.161.1.7902FCR . . . . .	254
6.161.1.7902FDR . . . . .	254
6.161.1.7902ICR . . . . .	254
6.161.1.7902IER . . . . .	254
6.161.1.7902IIR . . . . .	254
6.161.1.7902LCR . . . . .	254
6.161.1.7902LSR . . . . .	255
6.161.1.7902RBR . . . . .	255
6.161.1.7902SCR . . . . .	255
6.161.1.8002TER . . . . .	255
6.161.1.8002THR . . . . .	255
6.161.1.8003ACR . . . . .	255
6.161.1.8003DLL . . . . .	255
6.161.1.8003DLM . . . . .	255
6.161.1.8003FCR . . . . .	255
6.161.1.8003FDR . . . . .	255
6.161.1.8003ICR . . . . .	255
6.161.1.8003IER . . . . .	255
6.161.1.8003IIR . . . . .	255
6.161.1.8103LCR . . . . .	255

6.161.1.81 <u>3</u> LSR . . . . .	255
6.161.1.81 <u>3</u> RBR . . . . .	255
6.161.1.81 <u>3</u> SCR . . . . .	255
6.161.1.81 <u>3</u> TER . . . . .	255
6.161.1.81 <u>3</u> THR . . . . .	255
6.161.1.81 <u>ART0</u> _BASE_ADDR . . . . .	255
6.161.1.81 <u>ART1</u> _BASE_ADDR . . . . .	255
6.161.1.81 <u>ART2</u> _BASE_ADDR . . . . .	255
6.161.1.81 <u>ART3</u> _BASE_ADDR . . . . .	255
6.161.1.82 <u>DCA</u> _HEAD . . . . .	255
6.161.1.82 <u>USB</u> _BASE_ADDR . . . . .	256
6.161.1.82 <u>SB</u> _CTRL . . . . .	256
6.161.1.82 <u>SB</u> _INT_BASE_ADDR . . . . .	256
6.161.1.82 <u>SB</u> _INT_STAT . . . . .	256
6.161.1.82 <u>SBCLKCFG</u> . . . . .	256
6.161.1.82 <u>SBClkCtrl</u> . . . . .	256
6.161.1.82 <u>SBClkSt</u> . . . . .	256
6.161.1.82 <u>SBHC</u> _BASE_ADDR . . . . .	256
6.161.1.82 <u>SBOTG</u> _BASE_ADDR . . . . .	256
6.161.1.83 <u>SBOTG</u> _CLK_BASE_ADDR . . . . .	256
6.161.1.83 <u>SBOTG</u> _I2C_BASE_ADDR . . . . .	256
6.161.1.83 <u>SBPortSel</u> . . . . .	256
6.161.1.83 <u>IIC</u> _BASE_ADDR . . . . .	256
6.161.1.83 <u>IICFIQStatus</u> . . . . .	256
6.161.1.83 <u>IICIntEnable</u> . . . . .	256
6.161.1.83 <u>IICIntEnClr</u> . . . . .	256
6.161.1.83 <u>IICIntSelect</u> . . . . .	256
6.161.1.83 <u>IICIRQStatus</u> . . . . .	256
6.161.1.83 <u>IICProtection</u> . . . . .	256
6.161.1.84 <u>IICRawIntr</u> . . . . .	256
6.161.1.84 <u>IICSoftInt</u> . . . . .	256
6.161.1.84 <u>IICSoftIntClr</u> . . . . .	257
6.161.1.84 <u>ICSWPrioMask</u> . . . . .	257
6.161.1.84 <u>IICVectAddr</u> . . . . .	257

6.161.1.84 <sup>8</sup> ICVectAddr0 . . . . .	257
6.161.1.84 <sup>8</sup> ICVectAddr1 . . . . .	257
6.161.1.84 <sup>7</sup> ICVectAddr10 . . . . .	257
6.161.1.84 <sup>8</sup> ICVectAddr11 . . . . .	257
6.161.1.84 <sup>9</sup> ICVectAddr12 . . . . .	257
6.161.1.85 <sup>0</sup> ICVectAddr13 . . . . .	257
6.161.1.85 <sup>1</sup> ICVectAddr14 . . . . .	257
6.161.1.85 <sup>2</sup> ICVectAddr15 . . . . .	257
6.161.1.85 <sup>3</sup> ICVectAddr16 . . . . .	257
6.161.1.85 <sup>4</sup> ICVectAddr17 . . . . .	257
6.161.1.85 <sup>5</sup> ICVectAddr18 . . . . .	257
6.161.1.85 <sup>6</sup> ICVectAddr19 . . . . .	257
6.161.1.85 <sup>7</sup> ICVectAddr2 . . . . .	257
6.161.1.85 <sup>8</sup> ICVectAddr20 . . . . .	257
6.161.1.85 <sup>9</sup> ICVectAddr21 . . . . .	258
6.161.1.86 <sup>0</sup> ICVectAddr22 . . . . .	258
6.161.1.86 <sup>1</sup> ICVectAddr23 . . . . .	258
6.161.1.86 <sup>2</sup> ICVectAddr24 . . . . .	258
6.161.1.86 <sup>3</sup> ICVectAddr25 . . . . .	258
6.161.1.86 <sup>4</sup> ICVectAddr26 . . . . .	258
6.161.1.86 <sup>5</sup> ICVectAddr27 . . . . .	258
6.161.1.86 <sup>6</sup> ICVectAddr28 . . . . .	258
6.161.1.86 <sup>7</sup> ICVectAddr29 . . . . .	258
6.161.1.86 <sup>8</sup> ICVectAddr3 . . . . .	258
6.161.1.86 <sup>9</sup> ICVectAddr30 . . . . .	258
6.161.1.87 <sup>0</sup> ICVectAddr31 . . . . .	258
6.161.1.87 <sup>1</sup> ICVectAddr4 . . . . .	258
6.161.1.87 <sup>2</sup> ICVectAddr5 . . . . .	258
6.161.1.87 <sup>3</sup> ICVectAddr6 . . . . .	258
6.161.1.87 <sup>4</sup> ICVectAddr7 . . . . .	258
6.161.1.87 <sup>5</sup> ICVectAddr8 . . . . .	259
6.161.1.87 <sup>6</sup> ICVectAddr9 . . . . .	259
6.161.1.87 <sup>7</sup> ICVectCntrl0 . . . . .	259
6.161.1.87 <sup>8</sup> ICVectCntrl1 . . . . .	259

6.161.1.87VICVectCtl10 . . . . .	259
6.161.1.88VICVectCtl11 . . . . .	259
6.161.1.88VICVectCtl12 . . . . .	259
6.161.1.88VICVectCtl13 . . . . .	259
6.161.1.88VICVectCtl14 . . . . .	259
6.161.1.88VICVectCtl15 . . . . .	259
6.161.1.88VICVectCtl16 . . . . .	259
6.161.1.88VICVectCtl17 . . . . .	259
6.161.1.88VICVectCtl18 . . . . .	259
6.161.1.88VICVectCtl19 . . . . .	259
6.161.1.88VICVectCtl20 . . . . .	259
6.161.1.89VICVectCtl21 . . . . .	259
6.161.1.89VICVectCtl22 . . . . .	260
6.161.1.89VICVectCtl23 . . . . .	260
6.161.1.89VICVectCtl24 . . . . .	260
6.161.1.89VICVectCtl25 . . . . .	260
6.161.1.89VICVectCtl26 . . . . .	260
6.161.1.89VICVectCtl27 . . . . .	260
6.161.1.89VICVectCtl28 . . . . .	260
6.161.1.89VICVectCtl29 . . . . .	260
6.161.1.90VICVectCtl3 . . . . .	260
6.161.1.90VICVectCtl30 . . . . .	260
6.161.1.90VICVectCtl31 . . . . .	260
6.161.1.90VICVectCtl4 . . . . .	260
6.161.1.90VICVectCtl5 . . . . .	260
6.161.1.90VICVectCtl6 . . . . .	260
6.161.1.90VICVectCtl7 . . . . .	260
6.161.1.90VICVectCtl8 . . . . .	260
6.161.1.90VICVectCtl9 . . . . .	260
6.161.1.90PBDIV . . . . .	260
6.161.1.91WDCLKSEL . . . . .	260
6.161.1.91WDFEED . . . . .	261
6.161.1.91WDG_BASE_ADDR . . . . .	261

6.161.1.91WDMOD . . . . .	261
6.161.1.91WDTC . . . . .	261
6.161.1.91WDTV . . . . .	261
6.162include/canvas.h File Reference . . . . .	261
6.162.1 Define Documentation . . . . .	262
6.162.1.1 Canvas . . . . .	262
6.162.1.2 CANVAS_STYLE_ALIGN_HMASK . . . . .	264
6.162.1.3 CANVAS_STYLE_ALIGN_MASK . . . . .	264
6.162.1.4 CANVAS_STYLE_ALIGN_VMASK . . . . .	264
6.162.1.5 CANVAS_STYLE_APP_DRAWN . . . . .	264
6.162.1.6 CANVAS_STYLE_FILL . . . . .	264
6.162.1.7 CANVAS_STYLE_IMG . . . . .	264
6.162.1.8 CANVAS_STYLE_OUTLINE . . . . .	265
6.162.1.9 CANVAS_STYLE_TEXT . . . . .	265
6.162.1.10CANVAS_STYLE_TEXT_BOTTOM . . . . .	265
6.162.1.11CANVAS_STYLE_TEXT_HCENTER . . . . .	265
6.162.1.12CANVAS_STYLE_TEXT_LEFT . . . . .	265
6.162.1.13CANVAS_STYLE_TEXT_OPAQUE . . . . .	265
6.162.1.14CANVAS_STYLE_TEXT_RIGHT . . . . .	265
6.162.1.15CANVAS_STYLE_TEXT_TOP . . . . .	265
6.162.1.16CANVAS_STYLE_TEXT_VCENTER . . . . .	265
6.162.1.17CanvasAppDrawnOff . . . . .	266
6.162.1.18CanvasAppDrawnOn . . . . .	266
6.162.1.19CanvasCallbackSet . . . . .	266
6.162.1.20CanvasFillColorSet . . . . .	267
6.162.1.21CanvasFillOff . . . . .	267
6.162.1.22CanvasFillOn . . . . .	268
6.162.1.23CanvasFontSet . . . . .	268
6.162.1.24CanvasImageOff . . . . .	269
6.162.1.25CanvasImageOn . . . . .	269
6.162.1.26CanvasImageSet . . . . .	270
6.162.1.27CanvasOutlineColorSet . . . . .	270
6.162.1.28CanvasOutlineOff . . . . .	271
6.162.1.29CanvasOutlineOn . . . . .	271

6.162.1.30CanvasStruct . . . . .	272
6.162.1.31CanvasTextAlignment . . . . .	274
6.162.1.32CanvasTextColorSet . . . . .	275
6.162.1.33CanvasTextOff . . . . .	275
6.162.1.34CanvasTextOn . . . . .	276
6.162.1.35CanvasTextOpaqueOff . . . . .	276
6.162.1.36CanvasTextOpaqueOn . . . . .	277
6.162.1.37CanvasTextSet . . . . .	277
6.162.2 Function Documentation . . . . .	278
6.162.2.1 CanvasInit . . . . .	278
6.162.2.2 CanvasMsgProc . . . . .	278
6.163include/checkbox.h File Reference . . . . .	278
6.163.1 Define Documentation . . . . .	280
6.163.1.1 CB_STYLE_FILL . . . . .	280
6.163.1.2 CB_STYLE_IMG . . . . .	280
6.163.1.3 CB_STYLE_OUTLINE . . . . .	280
6.163.1.4 CB_STYLE_SELECTED . . . . .	280
6.163.1.5 CB_STYLE_TEXT . . . . .	280
6.163.1.6 CB_STYLE_TEXT_OPAQUE . . . . .	280
6.163.1.7 CheckBox . . . . .	280
6.163.1.8 CheckBoxBoxSizeSet . . . . .	281
6.163.1.9 CheckBoxCallbackSet . . . . .	282
6.163.1.10CheckBoxFillColorSet . . . . .	282
6.163.1.11CheckBoxFillOff . . . . .	283
6.163.1.12CheckBoxFillOn . . . . .	283
6.163.1.13CheckBoxFontSet . . . . .	284
6.163.1.14CheckBoxImageOff . . . . .	284
6.163.1.15CheckBoxImageOn . . . . .	285
6.163.1.16CheckBoxImageSet . . . . .	285
6.163.1.17CheckBoxOutlineColorSet . . . . .	286
6.163.1.18CheckBoxOutlineOff . . . . .	286
6.163.1.19CheckBoxOutlineOn . . . . .	287
6.163.1.20CheckBoxStruct . . . . .	287
6.163.1.21CheckBoxTextColorSet . . . . .	289

6.163.1.22CheckBoxTextOff . . . . .	289
6.163.1.23CheckBoxTextOn . . . . .	290
6.163.1.24CheckBoxTextOpaqueOff . . . . .	290
6.163.1.25CheckBoxTextOpaqueOn . . . . .	291
6.163.1.26CheckBoxTextSet . . . . .	291
6.163.2 Function Documentation . . . . .	292
6.163.2.1 CheckBoxInit . . . . .	292
6.163.2.2 CheckBoxMsgProc . . . . .	292
6.164 include/container.h File Reference . . . . .	293
6.164.1 Define Documentation . . . . .	294
6.164.1.1 Container . . . . .	294
6.164.1.2 ContainerFillColorSet . . . . .	295
6.164.1.3 ContainerFillOff . . . . .	295
6.164.1.4 ContainerFillOn . . . . .	296
6.164.1.5 ContainerFontSet . . . . .	296
6.164.1.6 ContainerOutlineColorSet . . . . .	297
6.164.1.7 ContainerOutlineOff . . . . .	297
6.164.1.8 ContainerOutlineOn . . . . .	298
6.164.1.9 ContainerStruct . . . . .	298
6.164.1.10ContainerTextCenterOff . . . . .	300
6.164.1.11ContainerTextCenterOn . . . . .	300
6.164.1.12ContainerTextColorSet . . . . .	301
6.164.1.13ContainerTextOff . . . . .	301
6.164.1.14ContainerTextOn . . . . .	302
6.164.1.15ContainerTextOpaqueOff . . . . .	302
6.164.1.16ContainerTextOpaqueOn . . . . .	303
6.164.1.17ContainerTextSet . . . . .	303
6.164.1.18CTR_STYLE_FILL . . . . .	304
6.164.1.19CTR_STYLE_OUTLINE . . . . .	304
6.164.1.20CTR_STYLE_TEXT . . . . .	304
6.164.1.21CTR_STYLE_TEXT_CENTER . . . . .	304
6.164.1.22CTR_STYLE_TEXT_OPAQUE . . . . .	304
6.164.2 Function Documentation . . . . .	304
6.164.2.1 ContainerInit . . . . .	305

6.164.2.2 ContainerMsgProc . . . . .	305
6.165include/graphic.h File Reference . . . . .	305
6.165.1 Function Documentation . . . . .	306
6.165.1.1 GLCD_Circle . . . . .	306
6.165.1.2 GLCD_Line . . . . .	306
6.165.1.3 GLCD_Rectangle . . . . .	306
6.166include/grlib.h File Reference . . . . .	306
6.166.1 Define Documentation . . . . .	315
6.166.1.1 ClrAliceBlue . . . . .	315
6.166.1.2 ClrAntiqueWhite . . . . .	315
6.166.1.3 ClrAqua . . . . .	315
6.166.1.4 ClrAquamarine . . . . .	315
6.166.1.5 ClrAzure . . . . .	315
6.166.1.6 ClrBeige . . . . .	315
6.166.1.7 ClrBisque . . . . .	315
6.166.1.8 ClrBlack . . . . .	315
6.166.1.9 ClrBlanchedAlmond . . . . .	315
6.166.1.10ClrBlue . . . . .	315
6.166.1.11ClrBlueMask . . . . .	315
6.166.1.12ClrBlueShift . . . . .	315
6.166.1.13ClrBlueViolet . . . . .	315
6.166.1.14ClrBrown . . . . .	315
6.166.1.15ClrBurlyWood . . . . .	315
6.166.1.16ClrCadetBlue . . . . .	315
6.166.1.17ClrChartreuse . . . . .	315
6.166.1.18ClrChocolate . . . . .	315
6.166.1.19ClrCoral . . . . .	315
6.166.1.20ClrCornflowerBlue . . . . .	315
6.166.1.21ClrCornsilk . . . . .	315
6.166.1.22ClrCrimson . . . . .	315
6.166.1.23ClrCyan . . . . .	315
6.166.1.24ClrDarkBlue . . . . .	316
6.166.1.25ClrDarkCyan . . . . .	316
6.166.1.26ClrDarkGoldenrod . . . . .	316

6.166.1.27ClrDarkGray . . . . .	316
6.166.1.28ClrDarkGreen . . . . .	316
6.166.1.29ClrDarkKhaki . . . . .	316
6.166.1.30ClrDarkMagenta . . . . .	316
6.166.1.31ClrDarkOliveGreen . . . . .	316
6.166.1.32ClrDarkOrange . . . . .	316
6.166.1.33ClrDarkOrchid . . . . .	316
6.166.1.34ClrDarkRed . . . . .	316
6.166.1.35ClrDarkSalmon . . . . .	316
6.166.1.36ClrDarkSeaGreen . . . . .	316
6.166.1.37ClrDarkSlateBlue . . . . .	316
6.166.1.38ClrDarkSlateGray . . . . .	316
6.166.1.39ClrDarkTurquoise . . . . .	316
6.166.1.40ClrDarkViolet . . . . .	316
6.166.1.41ClrDeepPink . . . . .	316
6.166.1.42ClrDeepSkyBlue . . . . .	316
6.166.1.43ClrDimGray . . . . .	316
6.166.1.44ClrDodgerBlue . . . . .	316
6.166.1.45ClrFireBrick . . . . .	316
6.166.1.46ClrFloralWhite . . . . .	316
6.166.1.47ClrForestGreen . . . . .	316
6.166.1.48ClrFuchsia . . . . .	317
6.166.1.49ClrGainsboro . . . . .	317
6.166.1.50ClrGhostWhite . . . . .	317
6.166.1.51ClrGold . . . . .	317
6.166.1.52ClrGoldenrod . . . . .	317
6.166.1.53ClrGray . . . . .	317
6.166.1.54ClrGreen . . . . .	317
6.166.1.55ClrGreenMask . . . . .	317
6.166.1.56ClrGreenShift . . . . .	317
6.166.1.57ClrGreenYellow . . . . .	317
6.166.1.58ClrHoneydew . . . . .	317
6.166.1.59ClrHotPink . . . . .	317
6.166.1.60ClrIndianRed . . . . .	317

6.166.1.61ClrIndigo . . . . .	317
6.166.1.62ClrIvory . . . . .	317
6.166.1.63ClrKhaki . . . . .	317
6.166.1.64ClrLavender . . . . .	317
6.166.1.65ClrLavenderBlush . . . . .	317
6.166.1.66ClrLawnGreen . . . . .	317
6.166.1.67ClrLemonChiffon . . . . .	317
6.166.1.68ClrLightBlue . . . . .	317
6.166.1.69ClrLightCoral . . . . .	317
6.166.1.70ClrLightCyan . . . . .	317
6.166.1.71ClrLightGoldenrodYellow . . . . .	317
6.166.1.72ClrLightGreen . . . . .	318
6.166.1.73ClrLightGrey . . . . .	318
6.166.1.74ClrLightPink . . . . .	318
6.166.1.75ClrLightSalmon . . . . .	318
6.166.1.76ClrLightSeaGreen . . . . .	318
6.166.1.77ClrLightSkyBlue . . . . .	318
6.166.1.78ClrLightSlateGray . . . . .	318
6.166.1.79ClrLightSteelBlue . . . . .	318
6.166.1.80ClrLightYellow . . . . .	318
6.166.1.81ClrLime . . . . .	318
6.166.1.82ClrLimeGreen . . . . .	318
6.166.1.83ClrLinen . . . . .	318
6.166.1.84ClrMagenta . . . . .	318
6.166.1.85ClrMaroon . . . . .	318
6.166.1.86ClrMediumAquamarine . . . . .	318
6.166.1.87ClrMediumBlue . . . . .	318
6.166.1.88ClrMediumOrchid . . . . .	318
6.166.1.89ClrMediumPurple . . . . .	318
6.166.1.90ClrMediumSeaGreen . . . . .	318
6.166.1.91ClrMediumSlateBlue . . . . .	318
6.166.1.92ClrMediumSpringGreen . . . . .	318
6.166.1.93ClrMediumTurquoise . . . . .	318
6.166.1.94ClrMediumVioletRed . . . . .	318

6.166.1.95ClrMidnightBlue . . . . .	318
6.166.1.96ClrMintCream . . . . .	319
6.166.1.97ClrMistyRose . . . . .	319
6.166.1.98ClrMoccasin . . . . .	319
6.166.1.99ClrNavajoWhite . . . . .	319
6.166.1.100IrNavy . . . . .	319
6.166.1.103IrOldLace . . . . .	319
6.166.1.102IrOlive . . . . .	319
6.166.1.103IrOliveDrab . . . . .	319
6.166.1.104IrOrange . . . . .	319
6.166.1.105IrOrangeRed . . . . .	319
6.166.1.106IrOrchid . . . . .	319
6.166.1.107IrPaleGoldenrod . . . . .	319
6.166.1.108IrPaleGreen . . . . .	319
6.166.1.109IrPaleTurquoise . . . . .	319
6.166.1.110IrPaleVioletRed . . . . .	319
6.166.1.110IrPapayaWhip . . . . .	319
6.166.1.112IrPeachPuff . . . . .	319
6.166.1.113IrPeru . . . . .	319
6.166.1.114IrPink . . . . .	319
6.166.1.115IrPlum . . . . .	319
6.166.1.116IrPowderBlue . . . . .	319
6.166.1.117IrPurple . . . . .	319
6.166.1.118IrRed . . . . .	319
6.166.1.119IrRedMask . . . . .	319
6.166.1.120IrRedShift . . . . .	320
6.166.1.121IrRosyBrown . . . . .	320
6.166.1.122IrRoyalBlue . . . . .	320
6.166.1.123IrSaddleBrown . . . . .	320
6.166.1.124IrSalmon . . . . .	320
6.166.1.125IrSandyBrown . . . . .	320
6.166.1.126IrSeaGreen . . . . .	320
6.166.1.127IrSeashell . . . . .	320
6.166.1.128IrSienna . . . . .	320

6.166.1.129lrSilver . . . . .	320
6.166.1.130lrSkyBlue . . . . .	320
6.166.1.131lrSlateBlue . . . . .	320
6.166.1.132lrSlateGray . . . . .	320
6.166.1.133lrSnow . . . . .	320
6.166.1.134lrSpringGreen . . . . .	320
6.166.1.135lrSteelBlue . . . . .	320
6.166.1.136lrTan . . . . .	320
6.166.1.137lrTeal . . . . .	320
6.166.1.138lrThistle . . . . .	320
6.166.1.139lrTomato . . . . .	320
6.166.1.140lrTurquoise . . . . .	320
6.166.1.141lrViolet . . . . .	320
6.166.1.142lrWheat . . . . .	320
6.166.1.143lrWhite . . . . .	320
6.166.1.144lrWhiteSmoke . . . . .	321
6.166.1.145lrYellow . . . . .	321
6.166.1.146lrYellowGreen . . . . .	321
6.166.1.147DpyColorTranslate . . . . .	321
6.166.1.148DpyFlush . . . . .	321
6.166.1.149DpyHeightGet . . . . .	322
6.166.1.150DpyLineDrawH . . . . .	322
6.166.1.151DpyLineDrawV . . . . .	322
6.166.1.152DpyPixelDraw . . . . .	323
6.166.1.153DpyPixelDrawMultiple . . . . .	323
6.166.1.154DpyRectFill . . . . .	324
6.166.1.155DpyWidthGet . . . . .	325
6.166.1.156FONT_FMT_PIXEL_RLE . . . . .	325
6.166.1.157FONT_FMT_UNCOMPRESSED . . . . .	325
6.166.1.158rContextBackgroundSet . . . . .	325
6.166.1.159rContextBackgroundSetTranslated . . . . .	326
6.166.1.160rContextDpyHeightGet . . . . .	326
6.166.1.161rContextDpyWidthGet . . . . .	327
6.166.1.162rContextFontSet . . . . .	327

6.166.1.163rContextForegroundSet . . . . .	327
6.166.1.164rContextForegroundSetTranslated . . . . .	328
6.166.1.165rFlush . . . . .	328
6.166.1.166rFontBaselineGet . . . . .	329
6.166.1.167rFontHeightGet . . . . .	329
6.166.1.168rFontMaxWidthGet . . . . .	329
6.166.1.169rImageColorsGet . . . . .	330
6.166.1.170rImageHeightGet . . . . .	330
6.166.1.171rImageWidthGet . . . . .	330
6.166.1.172rLangDe . . . . .	331
6.166.1.173rLangEnAUS . . . . .	331
6.166.1.174rLangEnCA . . . . .	331
6.166.1.175rLangEnNZ . . . . .	331
6.166.1.176rLangEnUK . . . . .	331
6.166.1.177rLangEnUS . . . . .	331
6.166.1.178rLangEsMX . . . . .	331
6.166.1.179rLangEsSP . . . . .	331
6.166.1.180rLangFr . . . . .	331
6.166.1.181rLangHi . . . . .	331
6.166.1.182rLangIt . . . . .	331
6.166.1.183rLangJp . . . . .	331
6.166.1.184rLangKo . . . . .	331
6.166.1.185rLangSwKE . . . . .	331
6.166.1.186rLangUrIN . . . . .	331
6.166.1.187rLangUrPK . . . . .	331
6.166.1.188rLangZhPRC . . . . .	331
6.166.1.189rLangZhTW . . . . .	331
6.166.1.190rOffScreen1BPPSize . . . . .	331
6.166.1.191rOffScreen4BPPSize . . . . .	332
6.166.1.192rOffScreen8BPPSize . . . . .	332
6.166.1.193rPixelDraw . . . . .	333
6.166.1.194rRectContainsPoint . . . . .	333
6.166.1.195rStringBaselineGet . . . . .	334
6.166.1.196rStringDrawCentered . . . . .	334

6.166.1.19GrStringHeightGet . . . . .	335
6.166.1.19GrStringMaxWidthGet . . . . .	335
6.166.1.19IMAGE_FMT_16BPP_UNCOMP . . . . .	336
6.166.1.20IMAGE_FMT_1BPP_COMP . . . . .	336
6.166.1.20IMAGE_FMT_1BPP_UNCOMP . . . . .	336
6.166.1.20IMAGE_FMT_4BPP_COMP . . . . .	336
6.166.1.20IMAGE_FMT_4BPP_UNCOMP . . . . .	336
6.166.1.20IMAGE_FMT_8BPP_COMP . . . . .	336
6.166.1.20IMAGE_FMT_8BPP_UNCOMP . . . . .	336
6.166.2 Function Documentation . . . . .	336
6.166.2.1 GrCircleDraw . . . . .	336
6.166.2.2 GrCircleFill . . . . .	337
6.166.2.3 GrContextClipRegionSet . . . . .	337
6.166.2.4 GrContextInit . . . . .	338
6.166.2.5 GrImageDraw . . . . .	338
6.166.2.6 GrLineDraw . . . . .	339
6.166.2.7 GrLineDrawH . . . . .	339
6.166.2.8 GrLineDrawV . . . . .	339
6.166.2.9 GrRectDraw . . . . .	340
6.166.2.10GrRectFill . . . . .	340
6.166.2.11GrRectIntersectGet . . . . .	341
6.166.2.12GrRectOverlapCheck . . . . .	341
6.166.2.13GrStringDraw . . . . .	341
6.166.2.14GrStringGet . . . . .	342
6.166.2.15GrStringLanguageSet . . . . .	342
6.166.2.16GrStringTableSet . . . . .	343
6.166.2.17GrStringWidthGet . . . . .	343
6.166.3 Variable Documentation . . . . .	344
6.166.3.1 g_sFontCm12 . . . . .	344
6.166.3.2 g_sFontCm12b . . . . .	344
6.166.3.3 g_sFontCm12i . . . . .	344
6.166.3.4 g_sFontCm14 . . . . .	344
6.166.3.5 g_sFontCm14b . . . . .	344
6.166.3.6 g_sFontCm14i . . . . .	344

6.166.3.7 g_sFontCm16 . . . . .	344
6.166.3.8 g_sFontCm16b . . . . .	344
6.166.3.9 g_sFontCm16i . . . . .	344
6.166.3.10g_sFontCm18 . . . . .	344
6.166.3.11g_sFontCm18b . . . . .	344
6.166.3.12g_sFontCm18i . . . . .	344
6.166.3.13g_sFontCm20 . . . . .	344
6.166.3.14g_sFontCm20b . . . . .	344
6.166.3.15g_sFontCm20i . . . . .	344
6.166.3.16g_sFontCm22 . . . . .	344
6.166.3.17g_sFontCm22b . . . . .	344
6.166.3.18g_sFontCm22i . . . . .	344
6.166.3.19g_sFontCm24 . . . . .	344
6.166.3.20g_sFontCm24b . . . . .	344
6.166.3.21g_sFontCm24i . . . . .	345
6.166.3.22g_sFontCm26 . . . . .	345
6.166.3.23g_sFontCm26b . . . . .	345
6.166.3.24g_sFontCm26i . . . . .	345
6.166.3.25g_sFontCm28 . . . . .	345
6.166.3.26g_sFontCm28b . . . . .	345
6.166.3.27g_sFontCm28i . . . . .	345
6.166.3.28g_sFontCm30 . . . . .	345
6.166.3.29g_sFontCm30b . . . . .	345
6.166.3.30g_sFontCm30i . . . . .	345
6.166.3.31g_sFontCm32 . . . . .	345
6.166.3.32g_sFontCm32b . . . . .	345
6.166.3.33g_sFontCm32i . . . . .	345
6.166.3.34g_sFontCm34 . . . . .	345
6.166.3.35g_sFontCm34b . . . . .	345
6.166.3.36g_sFontCm34i . . . . .	345
6.166.3.37g_sFontCm36 . . . . .	345
6.166.3.38g_sFontCm36b . . . . .	345
6.166.3.39g_sFontCm36i . . . . .	345
6.166.3.40g_sFontCm38 . . . . .	345

6.166.3.41g_sFontCm38b . . . . .	345
6.166.3.42g_sFontCm38i . . . . .	345
6.166.3.43g_sFontCm40 . . . . .	345
6.166.3.44g_sFontCm40b . . . . .	345
6.166.3.45g_sFontCm40i . . . . .	346
6.166.3.46g_sFontCm42 . . . . .	346
6.166.3.47g_sFontCm42b . . . . .	346
6.166.3.48g_sFontCm42i . . . . .	346
6.166.3.49g_sFontCm44 . . . . .	346
6.166.3.50g_sFontCm44b . . . . .	346
6.166.3.51g_sFontCm44i . . . . .	346
6.166.3.52g_sFontCm46 . . . . .	346
6.166.3.53g_sFontCm46b . . . . .	346
6.166.3.54g_sFontCm46i . . . . .	346
6.166.3.55g_sFontCm48 . . . . .	346
6.166.3.56g_sFontCm48b . . . . .	346
6.166.3.57g_sFontCm48i . . . . .	346
6.166.3.58g_sFontCmsc12 . . . . .	346
6.166.3.59g_sFontCmsc14 . . . . .	346
6.166.3.60g_sFontCmsc16 . . . . .	346
6.166.3.61g_sFontCmsc18 . . . . .	346
6.166.3.62g_sFontCmsc20 . . . . .	346
6.166.3.63g_sFontCmsc22 . . . . .	346
6.166.3.64g_sFontCmsc24 . . . . .	346
6.166.3.65g_sFontCmsc26 . . . . .	346
6.166.3.66g_sFontCmsc28 . . . . .	346
6.166.3.67g_sFontCmsc30 . . . . .	346
6.166.3.68g_sFontCmsc32 . . . . .	346
6.166.3.69g_sFontCmsc34 . . . . .	347
6.166.3.70g_sFontCmsc36 . . . . .	347
6.166.3.71g_sFontCmsc38 . . . . .	347
6.166.3.72g_sFontCmsc40 . . . . .	347
6.166.3.73g_sFontCmsc42 . . . . .	347
6.166.3.74g_sFontCmsc44 . . . . .	347

6.166.3.75g_sFontCmsc46 . . . . .	347
6.166.3.76g_sFontCmsc48 . . . . .	347
6.166.3.77g_sFontCmss12 . . . . .	347
6.166.3.78g_sFontCmss12b . . . . .	347
6.166.3.79g_sFontCmss12i . . . . .	347
6.166.3.80g_sFontCmss14 . . . . .	347
6.166.3.81g_sFontCmss14b . . . . .	347
6.166.3.82g_sFontCmss14i . . . . .	347
6.166.3.83g_sFontCmss16 . . . . .	347
6.166.3.84g_sFontCmss16b . . . . .	347
6.166.3.85g_sFontCmss16i . . . . .	347
6.166.3.86g_sFontCmss18 . . . . .	347
6.166.3.87g_sFontCmss18b . . . . .	347
6.166.3.88g_sFontCmss18i . . . . .	347
6.166.3.89g_sFontCmss20 . . . . .	347
6.166.3.90g_sFontCmss20b . . . . .	347
6.166.3.91g_sFontCmss20i . . . . .	347
6.166.3.92g_sFontCmss22 . . . . .	347
6.166.3.93g_sFontCmss22b . . . . .	348
6.166.3.94g_sFontCmss22i . . . . .	348
6.166.3.95g_sFontCmss24 . . . . .	348
6.166.3.96g_sFontCmss24b . . . . .	348
6.166.3.97g_sFontCmss24i . . . . .	348
6.166.3.98g_sFontCmss26 . . . . .	348
6.166.3.99g_sFontCmss26b . . . . .	348
6.166.3.100g_sFontCmss26i . . . . .	348
6.166.3.101g_sFontCmss28 . . . . .	348
6.166.3.102g_sFontCmss28b . . . . .	348
6.166.3.103g_sFontCmss28i . . . . .	348
6.166.3.104g_sFontCmss30 . . . . .	348
6.166.3.105g_sFontCmss30b . . . . .	348
6.166.3.106g_sFontCmss30i . . . . .	348
6.166.3.107g_sFontCmss32 . . . . .	348
6.166.3.108g_sFontCmss32b . . . . .	348

6.166.3.109_sFontCmss32i . . . . .	348
6.166.3.110_sFontCmss34 . . . . .	348
6.166.3.111_sFontCmss34b . . . . .	348
6.166.3.112_sFontCmss34i . . . . .	348
6.166.3.113_sFontCmss36 . . . . .	348
6.166.3.114_sFontCmss36b . . . . .	348
6.166.3.115_sFontCmss36i . . . . .	348
6.166.3.116_sFontCmss38 . . . . .	348
6.166.3.117_sFontCmss38b . . . . .	349
6.166.3.118_sFontCmss38i . . . . .	349
6.166.3.119_sFontCmss40 . . . . .	349
6.166.3.120_sFontCmss40b . . . . .	349
6.166.3.121_sFontCmss40i . . . . .	349
6.166.3.122_sFontCmss42 . . . . .	349
6.166.3.123_sFontCmss42b . . . . .	349
6.166.3.124_sFontCmss42i . . . . .	349
6.166.3.125_sFontCmss44 . . . . .	349
6.166.3.126_sFontCmss44b . . . . .	349
6.166.3.127_sFontCmss44i . . . . .	349
6.166.3.128_sFontCmss46 . . . . .	349
6.166.3.129_sFontCmss46b . . . . .	349
6.166.3.130_sFontCmss46i . . . . .	349
6.166.3.131_sFontCmss48 . . . . .	349
6.166.3.132_sFontCmss48b . . . . .	349
6.166.3.133_sFontCmss48i . . . . .	349
6.166.3.134_sFontFixed6x8 . . . . .	349
6.167 include/imgbutton.h File Reference . . . . .	349
6.167.1 Define Documentation . . . . .	351
6.167.1.1 IB_STYLE_AUTO_REPEAT . . . . .	351
6.167.1.2 IB_STYLE_FILL . . . . .	351
6.167.1.3 IB_STYLE_IMAGE_OFF . . . . .	351
6.167.1.4 IB_STYLE_KEYCAP_OFF . . . . .	351
6.167.1.5 IB_STYLE_PRESSED . . . . .	351
6.167.1.6 IB_STYLE_RELEASE_NOTIFY . . . . .	351

6.167.1.7 IB_STYLE_TEXT . . . . .	351
6.167.1.8 ImageButton . . . . .	351
6.167.1.9 ImageButtonAutoRepeatDelaySet . . . . .	353
6.167.1.10 ImageButtonAutoRepeatOff . . . . .	353
6.167.1.11 ImageButtonAutoRepeatOn . . . . .	354
6.167.1.12 ImageButtonAutoRepeatRateSet . . . . .	354
6.167.1.13 ImageButtonBackgroundColorSet . . . . .	355
6.167.1.14 ImageButtonCallbackSet . . . . .	355
6.167.1.15 ImageButtonFillColorSet . . . . .	356
6.167.1.16 ImageButtonFillOff . . . . .	356
6.167.1.17 ImageButtonFillOn . . . . .	357
6.167.1.18 ImageButtonForegroundColorSet . . . . .	357
6.167.1.19 ImageButtonImageKeycapSet . . . . .	358
6.167.1.20 ImageButtonImageOff . . . . .	358
6.167.1.21 ImageButtonImageOn . . . . .	359
6.167.1.22 ImageButtonImagePressedSet . . . . .	359
6.167.1.23 ImageButtonImageSet . . . . .	360
6.167.1.24 ImageButtonKeycapOff . . . . .	360
6.167.1.25 ImageButtonKeycapOffsetSet . . . . .	361
6.167.1.26 ImageButtonKeycapOn . . . . .	362
6.167.1.27 ImageButtonPressedColorSet . . . . .	362
6.167.1.28 ImageButtonStruct . . . . .	363
6.167.1.29 ImageButtonTextOff . . . . .	364
6.167.1.30 ImageButtonTextOn . . . . .	365
6.167.1.31 ImageButtonTextSet . . . . .	365
6.167.2 Function Documentation . . . . .	366
6.167.2.1 ImageButtonInit . . . . .	366
6.167.2.2 ImageButtonMsgProc . . . . .	366
6.168 include/listbox.h File Reference . . . . .	367
6.169 include/pushbutton.h File Reference . . . . .	368
6.170 include/radiobutton.h File Reference . . . . .	369
6.171 include/sed1335.h File Reference . . . . .	371
6.171.1 Define Documentation . . . . .	373
6.171.1.1 SED1335_APH . . . . .	373

6.171.1.2 SED1335_APL . . . . .	373
6.171.1.3 SED1335_CGRAM_ADR . . . . .	373
6.171.1.4 SED1335_CM . . . . .	373
6.171.1.5 SED1335_CR . . . . .	373
6.171.1.6 SED1335_CRX . . . . .	373
6.171.1.7 SED1335_CRY . . . . .	373
6.171.1.8 SED1335_CSRDIR_D . . . . .	373
6.171.1.9 SED1335_CSRDIR_L . . . . .	373
6.171.1.10 SED1335_CSRDIR_R . . . . .	373
6.171.1.11 SED1335_CSRDIR_U . . . . .	373
6.171.1.12 SED1335_CSRF_P2 . . . . .	373
6.171.1.13 SED1335_CSRFORM . . . . .	373
6.171.1.14 SED1335_CSRR . . . . .	373
6.171.1.15 SED1335_CSRW . . . . .	373
6.171.1.16 SED1335_DISP_OFF . . . . .	373
6.171.1.17 SED1335_DISP_ON . . . . .	373
6.171.1.18 SED1335_DM1 . . . . .	373
6.171.1.19 SED1335_DM2 . . . . .	373
6.171.1.20 SED1335_FLASH . . . . .	373
6.171.1.21 SED1335_FX . . . . .	373
6.171.1.22 SED1335_FY . . . . .	373
6.171.1.23 SED1335_GRAPHICSIZE . . . . .	374
6.171.1.24 SED1335_GRAPHICSTART . . . . .	374
6.171.1.25 SED1335_HDOT_SCR . . . . .	374
6.171.1.26 SED1335_IV . . . . .	374
6.171.1.27 SED1335_LF . . . . .	374
6.171.1.28 SED1335_LINES . . . . .	374
6.171.1.29 SED1335_M0 . . . . .	374
6.171.1.30 SED1335_M1 . . . . .	374
6.171.1.31 SED1335_M2 . . . . .	374
6.171.1.32 SED1335_MEM_END . . . . .	374
6.171.1.33 SED1335_MREAD . . . . .	374
6.171.1.34 SED1335_MWRITE . . . . .	374
6.171.1.35 SED1335_MX0 . . . . .	374

6.171.1.36SED1335_MX1 . . . . .	374
6.171.1.37SED1335_OV . . . . .	374
6.171.1.38SED1335_OVLAY . . . . .	374
6.171.1.39SED1335_OVLAY_P1 . . . . .	374
6.171.1.40SED1335_SAD1H . . . . .	374
6.171.1.41SED1335_SAD1L . . . . .	374
6.171.1.42SED1335_SAD2H . . . . .	374
6.171.1.43SED1335_SAD2L . . . . .	374
6.171.1.44SED1335_SAD3H . . . . .	374
6.171.1.45SED1335_SAD3L . . . . .	375
6.171.1.46SED1335_SAD4H . . . . .	375
6.171.1.47SED1335_SAD4L . . . . .	375
6.171.1.48SED1335_SAGH . . . . .	375
6.171.1.49SED1335_SAGL . . . . .	375
6.171.1.50SED1335_SCR_WIDTH . . . . .	375
6.171.1.51SED1335_SCRD . . . . .	375
6.171.1.52SED1335_SCROLL . . . . .	375
6.171.1.53SED1335_SL1 . . . . .	375
6.171.1.54SED1335_SL2 . . . . .	375
6.171.1.55SED1335_SLEEP_IN . . . . .	375
6.171.1.56SED1335_SYS_P1 . . . . .	375
6.171.1.57SED1335_SYS_P2 . . . . .	375
6.171.1.58SED1335_SYSTEM_SET . . . . .	375
6.171.1.59SED1335_TCR . . . . .	375
6.171.1.60SED1335_TEXTSIZE . . . . .	375
6.171.1.61SED1335_WF . . . . .	375
6.171.1.62SED1335_WS . . . . .	375
6.171.2 Function Documentation . . . . .	375
6.171.2.1 GLCD_Bitmap . . . . .	375
6.171.2.2 GLCD_ClearGraphic . . . . .	375
6.171.2.3 GLCD_ClearText . . . . .	375
6.171.2.4 GLCD_Initialize . . . . .	375
6.171.2.5 GLCD_ReadByteFromROMMemory . . . . .	376
6.171.2.6 GLCD_ReadData . . . . .	376

6.171.2.7 GLCD_SetCursorAddress . . . . .	376
6.171.2.8 GLCD_SetPixel . . . . .	376
6.171.2.9 GLCD_TextGoTo . . . . .	376
6.171.2.10 GLCD_WriteCommand . . . . .	376
6.171.2.11 GLCD_WriteData . . . . .	376
6.171.2.12 GLCD_WriteText . . . . .	376
6.172 include/slider.h File Reference . . . . .	376
6.172.1 Define Documentation . . . . .	377
6.172.1.1 SL_STYLE_BACKG_FILL . . . . .	377
6.172.1.2 SL_STYLE_BACKG_IMG . . . . .	378
6.172.1.3 SL_STYLE_BACKG_TEXT . . . . .	378
6.172.1.4 SL_STYLE_BACKG_TEXT_OPAQUE . . . . .	378
6.172.1.5 SL_STYLE_FILL . . . . .	378
6.172.1.6 SL_STYLE_IMG . . . . .	378
6.172.1.7 SL_STYLE_LOCKED . . . . .	378
6.172.1.8 SL_STYLE_OUTLINE . . . . .	378
6.172.1.9 SL_STYLE_TEXT . . . . .	378
6.172.1.10 SL_STYLE_TEXT_OPAQUE . . . . .	378
6.172.1.11 SL_STYLE_VERTICAL . . . . .	379
6.172.1.12 Slider . . . . .	379
6.172.1.13 SliderBackgroundImageOff . . . . .	381
6.172.1.14 SliderBackgroundImageOn . . . . .	381
6.172.1.15 SliderBackgroundImageSet . . . . .	381
6.172.1.16 SliderBackgroundTextColorSet . . . . .	382
6.172.1.17 SliderBackgroundTextOff . . . . .	383
6.172.1.18 SliderBackgroundTextOn . . . . .	383
6.172.1.19 SliderBackgroundTextOpaqueOff . . . . .	383
6.172.1.20 SliderBackgroundTextOpaqueOn . . . . .	384
6.172.1.21 SliderCallbackSet . . . . .	385
6.172.1.22 SliderFillColorBackgroundedSet . . . . .	385
6.172.1.23 SliderFillColorSet . . . . .	386
6.172.1.24 SliderFillOff . . . . .	386
6.172.1.25 SliderFillOn . . . . .	387
6.172.1.26 SliderFontSet . . . . .	387

6.172.1.27SliderImageOff . . . . .	388
6.172.1.28SliderImageOn . . . . .	388
6.172.1.29SliderImageSet . . . . .	388
6.172.1.30SliderLock . . . . .	389
6.172.1.31SliderOutlineColorSet . . . . .	390
6.172.1.32SliderOutlineOff . . . . .	390
6.172.1.33SliderOutlineOn . . . . .	391
6.172.1.34SliderRangeSet . . . . .	391
6.172.1.35SliderStruct . . . . .	392
6.172.1.36SliderTextColorSet . . . . .	394
6.172.1.37SliderTextOff . . . . .	394
6.172.1.38SliderTextOn . . . . .	395
6.172.1.39SliderTextOpaqueOff . . . . .	395
6.172.1.40SliderTextOpaqueOn . . . . .	396
6.172.1.41SliderTextSet . . . . .	396
6.172.1.42SliderUnlock . . . . .	397
6.172.1.43SliderValueSet . . . . .	397
6.172.2 Function Documentation . . . . .	398
6.172.2.1 SliderInit . . . . .	398
6.172.2.2 SliderMsgProc . . . . .	398
6.173include/widget.h File Reference . . . . .	399
6.173.1 Define Documentation . . . . .	400
6.173.1.1 WIDGET_MSG_PAINT . . . . .	400
6.173.1.2 WIDGET_MSG_PTR_DOWN . . . . .	400
6.173.1.3 WIDGET_MSG_PTR_MOVE . . . . .	400
6.173.1.4 WIDGET_MSG_PTR_UP . . . . .	400
6.173.1.5 WIDGET_ROOT . . . . .	400
6.173.1.6 WidgetPaint . . . . .	400
6.173.2 Typedef Documentation . . . . .	401
6.173.2.1 tWidget . . . . .	401
6.173.3 Function Documentation . . . . .	401
6.173.3.1 WidgetAdd . . . . .	401
6.173.3.2 WidgetDefaultMsgProc . . . . .	402
6.173.3.3 WidgetMessageQueueAdd . . . . .	402

6.173.3.4 WidgetMessageQueueProcess . . . . .	403
6.173.3.5 WidgetMessageSendPostOrder . . . . .	403
6.173.3.6 WidgetMessageSendPreOrder . . . . .	404
6.173.3.7 WidgetMutexGet . . . . .	404
6.173.3.8 WidgetMutexInit . . . . .	405
6.173.3.9 WidgetMutexPut . . . . .	405
6.173.3.10WidgetPointerMessage . . . . .	406
6.173.3.11WidgetRemove . . . . .	406
6.173.4 Variable Documentation . . . . .	406
6.173.4.1 g_sRoot . . . . .	406
6.174OLED/graphic.c File Reference . . . . .	406
6.174.1 Function Documentation . . . . .	407
6.174.1.1 GLCD_Circle . . . . .	407
6.174.1.2 GLCD_Line . . . . .	407
6.174.1.3 GLCD_Rectangle . . . . .	407
6.174.1.4 GLCD_SetPixel . . . . .	407
6.174.2 Variable Documentation . . . . .	407
6.174.2.1 color . . . . .	407
6.175src/main.c File Reference . . . . .	407
6.175.1 Function Documentation . . . . .	407
6.175.1.1 main . . . . .	407
6.176src/msp430-main.c File Reference . . . . .	407
6.176.1 Function Documentation . . . . .	408
6.176.1.1 main . . . . .	408
6.177src/old_main.c File Reference . . . . .	408
6.177.1 Function Documentation . . . . .	408
6.177.1.1 main . . . . .	408

# 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:

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



# Chapter 3

## File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

driver/ <a href="#">sed1335-AT91.c</a>	93
driver/ <a href="#">sed1335-avr.c</a>	95
driver/ <a href="#">sed1335-LPC2100.c</a>	96
driver/ <a href="#">sed1335-MSP430.c</a>	98
driver/ <a href="#">sed1335-STM32.c</a>	99
driver/ <a href="#">sed1335.c</a>	100
grlib/ <a href="#">canvas.c</a>	101
grlib/ <a href="#">checkbox.c</a>	102
grlib/ <a href="#">circle.c</a>	104
grlib/ <a href="#">container.c</a>	105
grlib/ <a href="#">context.c</a>	106
grlib/ <a href="#">image.c</a>	147
grlib/ <a href="#">imgbutton.c</a>	148
grlib/ <a href="#">line.c</a>	149
grlib/ <a href="#">listbox.c</a>	151
grlib/ <a href="#">offscr1bpp.c</a>	151
grlib/ <a href="#">offscr4bpp.c</a>	152
grlib/ <a href="#">offscr8bpp.c</a>	152
grlib/ <a href="#">pushbutton.c</a>	152
grlib/ <a href="#">radiobutton.c</a>	153
grlib/ <a href="#">rectangle.c</a>	153
grlib/ <a href="#">slider.c</a>	155
grlib/ <a href="#">string.c</a>	156
grlib/ <a href="#">widget.c</a>	160
grlib/fonts/ <a href="#">fontcm12.c</a>	107
grlib/fonts/ <a href="#">fontcm12b.c</a>	107
grlib/fonts/ <a href="#">fontcm12i.c</a>	108
grlib/fonts/ <a href="#">fontcm14.c</a>	108
grlib/fonts/ <a href="#">fontcm14b.c</a>	108

gllib/fonts/fontcm14i.c . . . . .	108
gllib/fonts/fontcm16.c . . . . .	109
gllib/fonts/fontcm16b.c . . . . .	109
gllib/fonts/fontcm16i.c . . . . .	109
gllib/fonts/fontcm18.c . . . . .	110
gllib/fonts/fontcm18b.c . . . . .	110
gllib/fonts/fontcm18i.c . . . . .	110
gllib/fonts/fontcm20.c . . . . .	111
gllib/fonts/fontcm20b.c . . . . .	111
gllib/fonts/fontcm20i.c . . . . .	111
gllib/fonts/fontcm22.c . . . . .	111
gllib/fonts/fontcm22b.c . . . . .	112
gllib/fonts/fontcm22i.c . . . . .	112
gllib/fonts/fontcm24.c . . . . .	112
gllib/fonts/fontcm24b.c . . . . .	113
gllib/fonts/fontcm24i.c . . . . .	113
gllib/fonts/fontcm26.c . . . . .	113
gllib/fonts/fontcm26b.c . . . . .	114
gllib/fonts/fontcm26i.c . . . . .	114
gllib/fonts/fontcm28.c . . . . .	114
gllib/fonts/fontcm28b.c . . . . .	114
gllib/fonts/fontcm28i.c . . . . .	115
gllib/fonts/fontcm30.c . . . . .	115
gllib/fonts/fontcm30b.c . . . . .	115
gllib/fonts/fontcm30i.c . . . . .	116
gllib/fonts/fontcm32.c . . . . .	116
gllib/fonts/fontcm32b.c . . . . .	116
gllib/fonts/fontcm32i.c . . . . .	117
gllib/fonts/fontcm34.c . . . . .	117
gllib/fonts/fontcm34b.c . . . . .	117
gllib/fonts/fontcm34i.c . . . . .	117
gllib/fonts/fontcm36.c . . . . .	118
gllib/fonts/fontcm36b.c . . . . .	118
gllib/fonts/fontcm36i.c . . . . .	118
gllib/fonts/fontcm38.c . . . . .	119
gllib/fonts/fontcm38b.c . . . . .	119
gllib/fonts/fontcm38i.c . . . . .	119
gllib/fonts/fontcm40.c . . . . .	120
gllib/fonts/fontcm40b.c . . . . .	120
gllib/fonts/fontcm40i.c . . . . .	120
gllib/fonts/fontcm42.c . . . . .	120
gllib/fonts/fontcm42b.c . . . . .	121
gllib/fonts/fontcm42i.c . . . . .	121
gllib/fonts/fontcm44.c . . . . .	121
gllib/fonts/fontcm44b.c . . . . .	122
gllib/fonts/fontcm44i.c . . . . .	122
gllib/fonts/fontcm46.c . . . . .	122
gllib/fonts/fontcm46b.c . . . . .	123
gllib/fonts/fontcm46i.c . . . . .	123
gllib/fonts/fontcm48.c . . . . .	123

gllib/fonts/ <a href="#">fontcm48b.c</a>	123
gllib/fonts/ <a href="#">fontcm48i.c</a>	124
gllib/fonts/ <a href="#">fontcmsc12.c</a>	124
gllib/fonts/ <a href="#">fontcmsc14.c</a>	124
gllib/fonts/ <a href="#">fontcmsc16.c</a>	125
gllib/fonts/ <a href="#">fontcmsc18.c</a>	125
gllib/fonts/ <a href="#">fontcmsc20.c</a>	125
gllib/fonts/ <a href="#">fontcmsc22.c</a>	126
gllib/fonts/ <a href="#">fontcmsc24.c</a>	126
gllib/fonts/ <a href="#">fontcmsc26.c</a>	126
gllib/fonts/ <a href="#">fontcmsc28.c</a>	126
gllib/fonts/ <a href="#">fontcmsc30.c</a>	127
gllib/fonts/ <a href="#">fontcmsc32.c</a>	127
gllib/fonts/ <a href="#">fontcmsc34.c</a>	127
gllib/fonts/ <a href="#">fontcmsc36.c</a>	128
gllib/fonts/ <a href="#">fontcmsc38.c</a>	128
gllib/fonts/ <a href="#">fontcmsc40.c</a>	128
gllib/fonts/ <a href="#">fontcmsc42.c</a>	129
gllib/fonts/ <a href="#">fontcmsc44.c</a>	129
gllib/fonts/ <a href="#">fontcmsc46.c</a>	129
gllib/fonts/ <a href="#">fontcmsc48.c</a>	129
gllib/fonts/ <a href="#">fontcmss12.c</a>	130
gllib/fonts/ <a href="#">fontcmss12b.c</a>	130
gllib/fonts/ <a href="#">fontcmss12i.c</a>	130
gllib/fonts/ <a href="#">fontcmss14.c</a>	131
gllib/fonts/ <a href="#">fontcmss14b.c</a>	131
gllib/fonts/ <a href="#">fontcmss14i.c</a>	131
gllib/fonts/ <a href="#">fontcmss16.c</a>	132
gllib/fonts/ <a href="#">fontcmss16b.c</a>	132
gllib/fonts/ <a href="#">fontcmss16i.c</a>	132
gllib/fonts/ <a href="#">fontcmss18.c</a>	132
gllib/fonts/ <a href="#">fontcmss18b.c</a>	133
gllib/fonts/ <a href="#">fontcmss18i.c</a>	133
gllib/fonts/ <a href="#">fontcmss20.c</a>	133
gllib/fonts/ <a href="#">fontcmss20b.c</a>	134
gllib/fonts/ <a href="#">fontcmss20i.c</a>	134
gllib/fonts/ <a href="#">fontcmss22.c</a>	134
gllib/fonts/ <a href="#">fontcmss22b.c</a>	135
gllib/fonts/ <a href="#">fontcmss22i.c</a>	135
gllib/fonts/ <a href="#">fontcmss24.c</a>	135
gllib/fonts/ <a href="#">fontcmss24b.c</a>	135
gllib/fonts/ <a href="#">fontcmss24i.c</a>	136
gllib/fonts/ <a href="#">fontcmss26.c</a>	136
gllib/fonts/ <a href="#">fontcmss26b.c</a>	136
gllib/fonts/ <a href="#">fontcmss26i.c</a>	137
gllib/fonts/ <a href="#">fontcmss28.c</a>	137
gllib/fonts/ <a href="#">fontcmss28b.c</a>	137
gllib/fonts/ <a href="#">fontcmss28i.c</a>	138
gllib/fonts/ <a href="#">fontcmss30.c</a>	138
gllib/fonts/ <a href="#">fontcmss30b.c</a>	138

gllib/fonts/fontcmss30i.c	138
gllib/fonts/fontcmss32.c	139
gllib/fonts/fontcmss32b.c	139
gllib/fonts/fontcmss32i.c	139
gllib/fonts/fontcmss34.c	140
gllib/fonts/fontcmss34b.c	140
gllib/fonts/fontcmss34i.c	140
gllib/fonts/fontcmss36.c	141
gllib/fonts/fontcmss36b.c	141
gllib/fonts/fontcmss36i.c	141
gllib/fonts/fontcmss38.c	141
gllib/fonts/fontcmss38b.c	142
gllib/fonts/fontcmss38i.c	142
gllib/fonts/fontcmss40.c	142
gllib/fonts/fontcmss40b.c	143
gllib/fonts/fontcmss40i.c	143
gllib/fonts/fontcmss42.c	143
gllib/fonts/fontcmss42b.c	144
gllib/fonts/fontcmss42i.c	144
gllib/fonts/fontcmss44.c	144
gllib/fonts/fontcmss44b.c	144
gllib/fonts/fontcmss44i.c	145
gllib/fonts/fontcmss46.c	145
gllib/fonts/fontcmss46b.c	145
gllib/fonts/fontcmss46i.c	146
gllib/fonts/fontcmss48.c	146
gllib/fonts/fontcmss48b.c	146
gllib/fonts/fontcmss48i.c	147
gllib/fonts/fontfixed6x8.c	147
include/canvas.h	261
include/checkbox.h	278
include/container.h	293
include/graphic.h	305
include/gplib.h	306
include/imgbutton.h	349
include/listbox.h	367
include/pushbutton.h	368
include/radiobutton.h	369
include/sed1335.h	371
include/slider.h	376
include/widget.h	399
include/build/lpc210x.h	166
include/build/lpc213x.h	178
include/build/lpc23xx.h	192
OLED/graphic.c	406
src/main.c	407
src/msp430-main.c	407
src/old_main.c	408

## 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 lWidth, long lHeight` )

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>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.

#### 4.1.2.5 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, 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 \*\* ppcText**
- **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* )

6.6.1.2 void `GLCD_ClearGraphic` ( void )

6.6.1.3 void `GLCD_ClearText` ( void )

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

6.6.1.5 void `GLCD_Initialize` ( void )

6.6.1.6 void `GLCD_InitializePorts` ( void )

6.6.1.7 void `GLCD_SetCursorPosition` ( unsigned int *address* )

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

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

6.6.1.10 void `GLCD_WriteText` ( char \* *tekst* )

6.6.1.11 void `GLCD_WriteTextP` ( char \* *tekst* )

## 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 lX, unsigned lY, unsigned lWidth, unsigned lHeight )**

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>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.

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 *IRadius*)
- void `GrCircleFill` (const `tContext` \**pContext*, unsigned *IX*, unsigned *IY*, unsigned *IRadius*)

#### 6.9.1 Function Documentation

##### 6.9.1.1 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.

##### 6.9.1.2 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  $IX - IRadiu$ s to  $IX + IRadiu$ s and  $IY - IRadiu$ s to  $IY + IRadiu$ s, 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**

<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.

**6.11.1.2 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 tDisplayInfo 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/fontcmsg12.c File Reference**

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

**Variables**

- const tFont g\_sFontCmsg12

**6.69.1 Variable Documentation**

6.69.1.1 const tFont g\_sFontCmsg12

**6.70 grlib/fonts/fontcmsg14.c File Reference**

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

**Variables**

- const tFont g\_sFontCmsg14

### 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>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.

6.148.1.3 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.

## 6.149 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, 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 *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` `lWidth`, `long` `lHeight`)

## 6.151 grlib/offscr4bpp.c File Reference

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

## Functions

- void `GrOffScreen4BPPInit` (`tDisplay` \*`pDisplay`, `unsigned char` \*`puImage`, `long` `lWidth`, `long` `lHeight`)
- void `GrOffScreen4BPPPPaletteSet` (`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` \*`puImage`, `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 `IWidth`, unsigned `IHeight`)

## 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

<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.

#### 6.155.2.3 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 **ILength**)
- void **GrStringDraw** (const **tContext** \*pContext, const char \*pcString, int **ILength**, 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 *ILength*, 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>ILength</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 text on the screen. The *iLength* 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 laguage 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 <b>WIDGET_ROOT</b> .
<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.

6.158.2.2 `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 <b>true</b> if the message should be sent via a post-order search, and <b>false</b> if it should be sent via a pre-order search.
<i>bStopOn-Success</i>	is <b>true</b> if the message should be sent to widgets until one returns success, and <b>false</b> 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

*pWidget* 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 **T0TCR** (\*((volatile unsigned long \*) 0xE0004004))
- #define **T0TC** (\*((volatile unsigned long \*) 0xE0004008))
- #define **T0PR** (\*((volatile unsigned long \*) 0xE000400C))
- #define **T0PC** (\*((volatile unsigned long \*) 0xE0004010))
- #define **T0MCR** (\*((volatile unsigned long \*) 0xE0004014))
- #define **T0MRO** (\*((volatile unsigned long \*) 0xE0004018))
- #define **T0MR1** (\*((volatile unsigned long \*) 0xE000401C))
- #define **T0MR2** (\*((volatile unsigned long \*) 0xE0004020))
- #define **T0MR3** (\*((volatile unsigned long \*) 0xE0004024))
- #define **T0CCR** (\*((volatile unsigned long \*) 0xE0004028))
- #define **T0CR0** (\*((volatile unsigned long \*) 0xE000402C))
- #define **T0CR1** (\*((volatile unsigned long \*) 0xE0004030))
- #define **T0CR2** (\*((volatile unsigned long \*) 0xE0004034))
- #define **T0CR3** (\*((volatile unsigned long \*) 0xE0004038))
- #define **T0EMR** (\*((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 **T1MRO** (\*((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\_I2ADDR** (\*((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 **CIRR** (\*((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 PCONP (*((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_CR0 (*((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 T0MRO (*((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 T1MR0 (*((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 *) 0x3FFFC014))
• #define FIO0SET (*((volatile unsigned long *) 0x3FFFC018))
• #define FIO0DIR (*((volatile unsigned long *) 0x3FFFC000))
```

- #define **FIO0CLR** (\*((volatile unsigned long \*) 0x3FFFC01C))
- #define **FIO0MASK** (\*((volatile unsigned long \*) 0x3FFFC010))
- #define **FIO1PIN** (\*((volatile unsigned long \*) 0x3FFFC034))
- #define **FIO1SET** (\*((volatile unsigned long \*) 0x3FFFC038))
- #define **FIO1DIR** (\*((volatile unsigned long \*) 0x3FFFC020))
- #define **FIO1CLR** (\*((volatile unsigned long \*) 0x3FFFC03C))
- #define **FIO1MASK** (\*((volatile unsigned long \*) 0x3FFFC030))
- #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 **T0IR** (\*((volatile unsigned long \*) 0xE0004000))
- #define **T0TCR** (\*((volatile unsigned long \*) 0xE0004004))
- #define **T0TC** (\*((volatile unsigned long \*) 0xE0004008))
- #define **T0PR** (\*((volatile unsigned long \*) 0xE000400C))
- #define **T0PC** (\*((volatile unsigned long \*) 0xE0004010))
- #define **T0MCR** (\*((volatile unsigned long \*) 0xE0004014))
- #define **T0MR0** (\*((volatile unsigned long \*) 0xE0004018))
- #define **T0MR1** (\*((volatile unsigned long \*) 0xE000401C))
- #define **T0MR2** (\*((volatile unsigned long \*) 0xE0004020))
- #define **T0MR3** (\*((volatile unsigned long \*) 0xE0004024))
- #define **T0CCR** (\*((volatile unsigned long \*) 0xE0004028))
- #define **T0CR0** (\*((volatile unsigned long \*) 0xE000402C))
- #define **T0CR1** (\*((volatile unsigned long \*) 0xE0004030))
- #define **T0CR2** (\*((volatile unsigned long \*) 0xE0004034))
- #define **T0CR3** (\*((volatile unsigned long \*) 0xE0004038))
- #define **T0EMR** (\*((volatile unsigned long \*) 0xE000403C))
- #define **T0CTCR** (\*((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 **PWMMPR** (\*((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 **SSPDMDACR** (\*((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 IOPINO (*((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 S0SPPCR (*((volatile unsigned char *) 0xE0020000))

6.160.1.99 #define S0SPDR (*((volatile unsigned char *) 0xE0020008))

6.160.1.100 #define S0SPINT (*((volatile unsigned char *) 0xE002001C))

6.160.1.101 #define S0SPSR (*((volatile unsigned char *) 0xE0020004))

6.160.1.102 #define S0SPTCR (*((volatile unsigned char *) 0xE0020010))

6.160.1.103 #define S0SPTOR (*((volatile unsigned char *) 0xE0020018))

6.160.1.104 #define S0SPTSR (*((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 SSPDR (*((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 TOEMR (*((volatile unsigned long *) 0xE000403C))

6.160.1.123 #define TOIR (*((volatile unsigned long *) 0xE0004000))

6.160.1.124 #define TOMCR (*((volatile unsigned long *) 0xE0004014))

6.160.1.125 #define TOMR0 (*((volatile unsigned long *) 0xE0004018))

6.160.1.126 #define TOMR1 (*((volatile unsigned long *) 0xE000401C))

6.160.1.127 #define TOMR2 (*((volatile unsigned long *) 0xE0004020))

6.160.1.128 #define TOMR3 (*((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 T1MR0 (*((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\_APRA (\*(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\_MRD (\*(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 I20CONSET (\*(volatile unsigned long \*) (I2C0\_BASE\_ADDR + 0x00))
- #define I20STAT (\*(volatile unsigned long \*) (I2C0\_BASE\_ADDR + 0x04))
- #define I20DAT (\*(volatile unsigned long \*) (I2C0\_BASE\_ADDR + 0x08))
- #define I20ADR (\*(volatile unsigned long \*) (I2C0\_BASE\_ADDR + 0x0C))
- #define I20SCLH (\*(volatile unsigned long \*) (I2C0\_BASE\_ADDR + 0x10))
- #define I20SCLL (\*(volatile unsigned long \*) (I2C0\_BASE\_ADDR + 0x14))
- #define I20CONCLR (\*(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 I22ADR (\*(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\_THRHLD (\*(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)) /\* CoLlision 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\_RXDESCRIPTORNUM** (\*(volatile unsigned long \*)(**MAC\_BASE\_ADDR** + 0x110)) /\* Rx number of descriptors reg \*/
- #define **MAC\_RXPRODUCEINDEX** (\*(volatile unsigned long \*)(**MAC\_BASE\_A-  
DDR** + 0x114)) /\* Rx produce index reg (RO) \*/
- #define **MAC\_RXCONSUMEINDEX** (\*(volatile unsigned long \*)(**MAC\_BASE\_A-  
DDR** + 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\_TXDESCRIPTORNUM** (\*(volatile unsigned long \*)(**MAC\_BASE\_A-  
DDR** + 0x124)) /\* Tx number of descriptors reg \*/
- #define **MAC\_TXPRODUCEINDEX** (\*(volatile unsigned long \*)(**MAC\_BASE\_A-  
DDR** + 0x128)) /\* Tx produce index reg \*/
- #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-  
DDR** + 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\_A-  
DDR** + 0x204)) /\* Rx filter WoL status reg (RO) \*/
- #define **MAC\_RXFILTERWOLCLR** (\*(volatile unsigned long \*)(**MAC\_BASE\_A-  
DDR** + 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 CAN1EWL (*(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_APB (*(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_MRД (*(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_WAITOEN0 (*(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_WAITWENO (*(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 *) (DMA_BASE_ADDR + 0x9A))  
6.161.1.347 #define FIO_BASE_ADDR 0x3FFFC000  
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 THRHL (*(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 IODIRO0 (*(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 IOPINO (*(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 PWM1MR0 (*(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 SSP1CRO (*(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 TOCTCR (*(volatile unsigned long *)(TMR0_BASE_ADDR + 0x70))
6.161.1.691 #define TOEMR (*(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 TOMCR (*(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 T1MR0 (*(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 T2CR0 (*(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 T3CRO (*(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 T3MRO (*(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 USBCCLKCFG (*(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 VICSoftInt (*(volatile unsigned long *) (VIC_BASE_ADDR + 0x018))
```

```
6.161.1.842 #define VICSoftIntClr (*(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 VICVectAddr2 (*(volatile unsigned long *)(VIC_BASE_ADDR +
0x108))

6.161.1.858 #define VICVectAddr20 (*(volatile unsigned long *)(VIC_BASE_ADDR +
0x150))
```

```
6.161.1.859 #define VICVectAddr21 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x154))  
  
6.161.1.860 #define VICVectAddr22 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x158))  
  
6.161.1.861 #define VICVectAddr23 (*(volatile unsigned long *)(VIC_BASE_ADDR +  
0x15C))  
  
6.161.1.862 #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 VICVectAddr8 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x120))  
  
6.161.1.876 #define VICVectAddr9 (*(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 VICVectCntl10 (*(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 VICVectCntl20 (*(volatile unsigned long *) (VIC_BASE_ADDR +  
0x250))  
  
6.161.1.890 #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, plImg)`
- #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**

<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 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.
<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 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)
```

```

    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*, *pImg* )

#### Value:

```

do                                \
{
    tCanvasWidget *pW = pWidget;      \
    const unsigned char *pI = pImg;   \
    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>pImg</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;           \
}                                           \
while(0)
```

```

        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, pfOnPaint )
```

#### 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,
pfOnPaint
}
```

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>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.
<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 <b>CANVAS_STYLE_TEXT_LEFT</b> , <b>CANVAS_STYLE_TEXT_RIGHT</b> , <b>CANVAS_STYLE_TEXT_HCENTER</b> , <b>CANVAS_STYLE_TEXT_VCENTER</b> , <b>CANVAS_STYLE_TEXT_TOP</b> and <b>CANVAS_STYLE_TEXT_BOTTOM</b> .

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.

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, lWidth,lHeight, usStyle, usBoxSize, ulFillColor,ulOutlineColor, ulTextColor, pFont, pcText, puclImage, pfnOnChange)
- #define `CheckBox`(sName, pParent, pNext, pChild, pDisplay, IX, IY, lWidth,lHeight, usStyle, usBoxSize, ulFillColor, ulOutlineColor,ulTextColor, 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 ulParam1, unsigned ulParam2)
- void `CheckBoxInit` (tCheckBoxWidget \*pWidget, const tDisplay \*pDisplay, unsigned IX, unsigned IY, unsigned lWidth, unsigned lHeight)

### 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, puImage, pfnOnChange )`

#### Value:

```
tCheckBoxWidget sName =
    CheckBoxStruct(pParent, pNext, pChild, pDisplay, lX, lY, lWidth, \
    lHeight, usStyle, usBoxSize, ulFillColor, \
    ulOutlineColor, ulTextColor, pFont, pcText, \
    puImage, pfnOnChange)
```

Declares an initialized variable containing a check box 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.

<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).

*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*, *uiColor* )

#### 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>plImg</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

*pWidget* 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

*pWidget* 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, IX, IY, IWidth,  
    IHeight, usStyle, usBoxSize, ulFillColor, ulOutlineColor, ulTextColor, pFont,  
    pcText, puclImage, pfnOnChange )
```

## Value:

```
{\n{\n    sizeof(tCheckBoxWidget),\n    (tWidget *) (pParent),\n    (tWidget *) (pNext),\n    (tWidget *) (pChild),\n    pDisplay,\n{\n    lX,\n    lY,\n    (lX) + (lWidth) - 1,
```

```

        (lY) + (lHeight) - 1
    },
    CheckBoxMsgProc
},
usStyle,
usBoxSize,
ulFillColor,
ulOutlineColor,
ulTextColor,
pFont,
pcText,
pucImage,
pfnOnChange
}

```

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>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>pucImage</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 previously 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

6.164.1.1 #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, \
                    ulTextColor, pFont, pcText)
```

Declares an initialized variable containing a container 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 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 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.

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*, *IX*, *IY*, *IWidth*, *IHeight*, *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>lX</i>	is the X coordinate of the upper left corner of the container widget.
<i>lY</i>	is the Y coordinate of the upper left corner of the container widget.
<i>lWidth</i>	is the width of the container widget.
<i>lHeight</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 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). 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** 0x007FFFD4
- #define **ClrAzure** 0x00F0FFFF
- #define **ClrBeige** 0x00F5F5DC
- #define **ClrBisque** 0x00FFE4C4
- #define **ClrBlack** 0x00000000
- #define **ClrBlanchedAlmond** 0x00FFEBCD
- #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 0x00FFFFAF0
- #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 0x00FFFFFF0
- #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** 0x00FFFFFFE0
- #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** 0x00F5FFFA
- #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** 0x00FFEF05
- #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 0x00000FF00
- #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 **GrImageDraw** (const **tContext** \*pContext, const unsigned char \*puclImage, 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 \*puclImage, long IWidth, long IHeight)
- void **GrOffScreen4BPPInit** (**tDisplay** \*pDisplay, unsigned char \*puclImage, long IWidth, long IHeight)
- void **GrOffScreen4BPPPaletteSet** (**tDisplay** \*pDisplay, unsigned long \*pulPalette, unsigned long ulOffset, unsigned long ulCount)
- void **GrOffScreen8BPPInit** (**tDisplay** \*pDisplay, unsigned char \*puclImage, 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 0x007FFFD4  
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 0x00FFEBCD  
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 0x00FFFAF0  
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 0x00F0FFF0
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 0x00FFFFE0
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 0x00FFEF05  
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:

```
{
    \n
    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>IX</i>	is the X coordinate of the line.
<i>IY1</i>	is the starting Y coordinate of the line.
<i>IY2</i>	is the ending Y coordinate of the line.
<i>ulValue</i>	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, IX, IY, 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*, *IX0*, *ICount*, *IBPP*, *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>lX</i>	is the X coordinate of the first pixel.
<i>lY</i>	is the Y coordinate of the first pixel.
<i>lX0</i>	is sub-pixel offset within the pixel data, which is valid for 1 or 4 bit per pixel formats.
<i>lCount</i>	is the number of pixels to draw.
<i>lBPP</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
{
    tContext *pC = pContext;
    pC->ulBackground = DpyColorTranslate(pC->pDisplay, ulValue);
}
while(0)
```

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 \\\
{ \\\
    tContext *pC = pContext; \\\
    pC->ulBackground = ulValue; \\\
} \\\
while (0) \\\
```

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    } \n    while(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    } \n    while(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 + (256 * 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*, *lX*, *lY* )**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

*pContext* 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, ILength, IX, IY,  
bOpaque )
```

**Value:**

```
do                                \
{                                     \
    const tContext *pC = pContext;   \
    const char *pcStr = pcString;   \
    GrStringDraw(pC, pcStr, lLength, \
                 (lX) - (GrStringWidthGet(pC, pcStr, lLength) / 2), \
                 (lY) - (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>lX</i>	is the X coordinate of the center of the string position on the screen.
<i>lY</i>	is the Y coordinate of the center of the string position on the screen.
<i>bOpaque</i>	is <b>true</b> if the background of each character should be drawn and <b>false</b> if it should not (leaving the background as is).

This function draws a string of text 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 tDisplayInfo 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 \* puclImage, unsigned /X, unsigned /Y )

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>/X</i>	is the X coordinate of the upper left corner of the image.
<i>/Y</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 clipping 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>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.

**6.166.2.9 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>IX</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.

**6.166.2.14 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 laguge 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

- `#define IB_STYLE_TEXT 0x0004`

*This flag indicates that the image button should be filled.*
- `#define IB_STYLE_IMAGE_OFF 0x0008`

*This flag indicates that the image button should have text drawn on it.*
- `#define IB_STYLE_KEYCAP_OFF 0x0010`

*This flag indicates that the background image is to be disabled.*
- `#define IB_STYLE_AUTO_REPEAT 0x0020`

*This flag indicates that the keycap image is to be disabled.*
- `#define IB_STYLE_PRESSED 0x0040`

*This flag indicates that the image button is pressed.*
- `#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, lX, lY, 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 ) \\\
```

```
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>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 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>pucImage</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)
```

```

}
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 = pfnOnClik;
}
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>pfnOnClik</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*, *plImg* )

#### Value:

```

do                                \
{
    tImageButtonWidget *pW = (pWidget); \
    const unsigned char *pI = (plImg); \
    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>plImg</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*, *plmg* )

#### 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>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). 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

<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.

#### 6.167.2.2 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.

## 6.168 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, 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 \*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, IWidth, IHeight, ulStyle, ulFillColor,ulPressFillColor, ulOutlineColor, ulTextColor, pFont, pcText, puclImage, pucPressImage, usAutoRepeatDelay, 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, 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, pfOnChange)
- #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\_ADDR 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\_APH 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\_WID-TH+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_APB 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_TEXTCODE ((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 dx, int dy )
6.171.2.2 void GLCD_ClearGraphic( void )
6.171.2.3 void GLCD_ClearText( void )
6.171.2.4 void GLCD_Initialize( void )
```

- 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 )`
- 6.171.2.8 `void GLCD_SetPixel( unsigned int x, unsigned int y, int color )`
- 6.171.2.9 `void GLCD_TextGoTo( unsigned char, unsigned char )`
- 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 * )`

## 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,puBackgroundImage, 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, plmg)
- #define `SliderBackgroundImageSet`(pWidget, plmg)
- #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>lHeight</i>	is the height of the slider.
<i>lMin</i>	is the minimum value for the slider (corresponding to the left or bottom position).
<i>lMax</i>	is the maximum value for the slider (corresponding to the right or top position).
<i>lValue</i>	is the initial value of the slider. This must lie in the range defined by <i>lMin</i> and <i>lMax</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>ulOutlineColor</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>puclImage</i>	is a pointer to the image to draw on this slider.
<i>pucBackgroundImage</i>	is a pointer to the image to draw on the slider background.
<i>pfnOnChange</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 = pImg;    \
    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>pImg</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 SliderBackgroundColorSet( *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
{
    tSliderWidget *pW = pWidget;
    pW->ulStyle &= ~(SL_STYLE_BACKG_TEXT_OPAQUE);
}
while(0)

```

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
{
    tSliderWidget *pW = pWidget;
    pW->ulStyle |= SL_STYLE_BACKG_TEXT_OPAQUE;
}
while(0)

```

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\_STYLE\_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->ulBackgroundColor = 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.

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 = pImg; \
    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>pImg</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>pfnOnChange</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->ulStyle &= ~ (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*, *IVal* )****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>IVal</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 <b>WIDGET_ROOT</b> .
<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 <b>true</b> if the message should be sent via a post-order search, and <b>false</b> if it should be sent via a pre-order search.
<i>bStopOn-Success</i>	is <b>true</b> if the message should be sent to widgets until one returns success, and <b>false</b> 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

<code>pcMutex</code>	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

<code>pcMutex</code>	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 )