

chipKIT TFT Library

Generated by Doxygen 1.8.4

Fri Jun 27 2014 18:58:47

Contents

1	chipKIT-TFT	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Class Documentation	7
4.1	__attribute__ Struct Reference	7
4.2	Aggregator Class Reference	7
4.2.1	Detailed Description	8
4.2.2	Member Function Documentation	8
4.2.2.1	addDisplay	8
4.2.2.2	displayOff	8
4.2.2.3	displayOn	8
4.2.2.4	drawHorizontalLine	9
4.2.2.5	drawVerticalLine	9
4.2.2.6	fillScreen	9
4.2.2.7	getHeight	9
4.2.2.8	getWidth	9
4.2.2.9	initializeDevice	10
4.2.2.10	invertDisplay	10
4.2.2.11	setPixel	10
4.2.2.12	setRotation	10
4.3	AggregatorList Struct Reference	10
4.4	AnalogTouch Class Reference	11
4.4.1	Member Function Documentation	11
4.4.1.1	initializeDevice	11
4.4.1.2	isPressed	11
4.4.1.3	pressure	12
4.4.1.4	sample	12

4.4.1.5	setRotation	12
4.4.1.6	x	12
4.4.1.7	y	12
4.5	BD663474 Class Reference	12
4.5.1	Member Function Documentation	13
4.5.1.1	displayOff	13
4.5.1.2	displayOn	13
4.5.1.3	drawHorizontalLine	14
4.5.1.4	drawVerticalLine	14
4.5.1.5	fillRectangle	14
4.5.1.6	fillScreen	14
4.5.1.7	initializeDevice	14
4.5.1.8	invertDisplay	15
4.5.1.9	setPixel	15
4.5.1.10	setRotation	15
4.6	BitmapFileHeader Struct Reference	15
4.7	BitmapInfoHeader Struct Reference	16
4.8	BitmapPixel24 Struct Reference	16
4.9	BitmapPixel32 Struct Reference	16
4.10	BMP Class Reference	17
4.11	Color Class Reference	17
4.12	coord Struct Reference	22
4.13	CoreIO Class Reference	22
4.14	DataBlock Class Reference	22
4.15	DataStore Class Reference	23
4.16	DOGMe Class Reference	23
4.16.1	Member Function Documentation	24
4.16.1.1	initializeDevice	24
4.17	FontHeader Struct Reference	24
4.18	Framebuffer Class Reference	25
4.18.1	Member Function Documentation	26
4.18.1.1	bgColorAt	26
4.18.1.2	colorAt	27
4.18.1.3	displayOff	27
4.18.1.4	displayOn	27
4.18.1.5	drawHorizontalLine	27
4.18.1.6	drawVerticalLine	27
4.18.1.7	fillScreen	28
4.18.1.8	getHeight	28
4.18.1.9	getWidth	28

4.18.1.10 initializeDevice	28
4.18.1.11 invertDisplay	29
4.18.1.12 setPixel	29
4.18.1.13 setRotation	29
4.19 Framebuffer1 Class Reference	29
4.19.1 Member Function Documentation	30
4.19.1.1 colorAt	30
4.19.1.2 fillScreen	30
4.19.1.3 initializeDevice	30
4.19.1.4 setPixel	31
4.20 Framebuffer332 Class Reference	31
4.20.1 Member Function Documentation	31
4.20.1.1 bgColorAt	31
4.20.1.2 colorAt	32
4.20.1.3 fillScreen	32
4.20.1.4 initializeDevice	32
4.20.1.5 setPixel	32
4.21 Framebuffer332Fast Class Reference	33
4.21.1 Member Function Documentation	33
4.21.1.1 bgColorAt	33
4.21.1.2 colorAt	34
4.21.1.3 drawHorizontalLine	34
4.21.1.4 fillScreen	34
4.21.1.5 initializeDevice	34
4.21.1.6 setPixel	34
4.22 Framebuffer565 Class Reference	35
4.22.1 Member Function Documentation	35
4.22.1.1 bgColorAt	35
4.22.1.2 colorAt	36
4.22.1.3 fillScreen	36
4.22.1.4 initializeDevice	36
4.22.1.5 setPixel	36
4.23 HD44780 Class Reference	36
4.23.1 Member Function Documentation	37
4.23.1.1 displayOff	37
4.23.1.2 displayOn	38
4.23.1.3 drawHorizontalLine	38
4.23.1.4 drawVerticalLine	38
4.23.1.5 fillRectangle	38
4.23.1.6 fillScreen	38

4.23.1.7	initializeDevice	39
4.23.1.8	invertDisplay	39
4.23.1.9	setPixel	39
4.23.1.10	setRotation	39
4.24	HX8357 Class Reference	40
4.24.1	Member Function Documentation	40
4.24.1.1	closeWindow	40
4.24.1.2	displayOff	41
4.24.1.3	displayOn	41
4.24.1.4	drawHorizontalLine	41
4.24.1.5	drawVerticalLine	41
4.24.1.6	fillRectangle	41
4.24.1.7	fillScreen	42
4.24.1.8	initializeDevice	42
4.24.1.9	invertDisplay	42
4.24.1.10	openWindow	42
4.24.1.11	setPixel	43
4.24.1.12	setRotation	43
4.24.1.13	windowData	43
4.24.1.14	windowData	43
4.25	ILI9340 Class Reference	43
4.25.1	Member Function Documentation	44
4.25.1.1	displayOff	44
4.25.1.2	displayOn	44
4.25.1.3	drawHorizontalLine	45
4.25.1.4	drawVerticalLine	45
4.25.1.5	fillRectangle	45
4.25.1.6	fillScreen	45
4.25.1.7	initializeDevice	45
4.25.1.8	invertDisplay	46
4.25.1.9	setPixel	46
4.25.1.10	setRotation	46
4.26	Image Class Reference	46
4.27	KS0108 Class Reference	47
4.27.1	Member Function Documentation	48
4.27.1.1	displayOff	48
4.27.1.2	displayOn	48
4.27.1.3	drawHorizontalLine	48
4.27.1.4	drawVerticalLine	49
4.27.1.5	fillRectangle	49

4.27.1.6	fillScreen	49
4.27.1.7	initializeDevice	49
4.27.1.8	invertDisplay	49
4.27.1.9	setPixel	50
4.27.1.10	setRotation	50
4.28	LEDMatrix Class Reference	50
4.28.1	Member Function Documentation	51
4.28.1.1	displayOff	51
4.28.1.2	displayOn	51
4.28.1.3	fillScreen	51
4.28.1.4	initializeDevice	52
4.28.1.5	invertDisplay	52
4.28.1.6	setPixel	52
4.28.1.7	setRotation	52
4.29	MatrixISRLIST Struct Reference	52
4.30	MCP23S17 Class Reference	53
4.31	ParallelIO Class Reference	53
4.32	PICadillo35t Class Reference	54
4.32.1	Member Function Documentation	55
4.32.1.1	closeWindow	55
4.32.1.2	colorAt	55
4.32.1.3	displayOff	55
4.32.1.4	displayOn	56
4.32.1.5	drawHorizontalLine	56
4.32.1.6	drawVerticalLine	56
4.32.1.7	fillRectangle	56
4.32.1.8	fillScreen	56
4.32.1.9	initializeDevice	57
4.32.1.10	invertDisplay	57
4.32.1.11	openWindow	57
4.32.1.12	setPixel	57
4.32.1.13	setRotation	58
4.32.1.14	windowData	58
4.32.1.15	windowData	58
4.33	point3d Struct Reference	58
4.34	Raw565 Class Reference	59
4.35	Raw8 Class Reference	59
4.36	RawPar Class Reference	60
4.36.1	Member Function Documentation	60
4.36.1.1	nativeWidth	60

4.36.1.2	streamCommand16	60
4.36.1.3	streamCommand32	60
4.36.1.4	streamCommand8	61
4.36.1.5	streamData16	61
4.36.1.6	streamData32	61
4.36.1.7	streamData8	61
4.36.1.8	streamEnd	61
4.36.1.9	streamStart	61
4.36.1.10	writeCommand16	61
4.36.1.11	writeCommand32	61
4.36.1.12	writeCommand8	61
4.36.1.13	writeData16	62
4.36.1.14	writeData32	62
4.36.1.15	writeData8	62
4.37	RLE Class Reference	62
4.38	S6D0164 Class Reference	62
4.38.1	Member Function Documentation	63
4.38.1.1	closeWindow	63
4.38.1.2	displayOff	64
4.38.1.3	displayOn	64
4.38.1.4	drawHorizontalLine	64
4.38.1.5	drawVerticalLine	64
4.38.1.6	fillRectangle	64
4.38.1.7	fillScreen	65
4.38.1.8	initializeDevice	65
4.38.1.9	invertDisplay	65
4.38.1.10	openWindow	65
4.38.1.11	setPixel	65
4.38.1.12	setRotation	66
4.38.1.13	windowData	66
4.38.1.14	windowData	66
4.39	SPISRAM Class Reference	66
4.40	sprite Struct Reference	67
4.40.1	Detailed Description	67
4.40.2	Member Data Documentation	68
4.40.2.1	animdir	68
4.40.2.2	currentframe	68
4.40.2.3	data	68
4.40.2.4	frames	68
4.40.2.5	height	68

4.40.2.6	next	68
4.40.2.7	store	68
4.40.2.8	transparent	68
4.40.2.9	width	68
4.40.2.10	xpos	68
4.40.2.11	ypos	68
4.41	SRAM Class Reference	69
4.42	SSD1289 Class Reference	69
4.42.1	Member Function Documentation	70
4.42.1.1	closeWindow	70
4.42.1.2	displayOff	70
4.42.1.3	displayOn	71
4.42.1.4	drawHorizontalLine	71
4.42.1.5	drawVerticalLine	71
4.42.1.6	fillRectangle	71
4.42.1.7	fillScreen	71
4.42.1.8	initializeDevice	72
4.42.1.9	invertDisplay	72
4.42.1.10	openWindow	72
4.42.1.11	setPixel	72
4.42.1.12	setRotation	73
4.42.1.13	windowData	73
4.42.1.14	windowData	73
4.43	SSD1963 Class Reference	73
4.43.1	Member Function Documentation	74
4.43.1.1	displayOff	74
4.43.1.2	displayOn	74
4.43.1.3	drawHorizontalLine	74
4.43.1.4	drawVerticalLine	74
4.43.1.5	fillRectangle	75
4.43.1.6	fillScreen	75
4.43.1.7	initializeDevice	75
4.43.1.8	invertDisplay	75
4.43.1.9	setPixel	76
4.43.1.10	setRotation	76
4.43.2	Member Data Documentation	76
4.43.2.1	Height	76
4.43.2.2	Width	76
4.44	ST7735 Class Reference	76
4.44.1	Constructor & Destructor Documentation	77

4.44.1.1	ST7735	77
4.44.2	Member Function Documentation	77
4.44.2.1	displayOff	77
4.44.2.2	displayOn	78
4.44.2.3	drawHorizontalLine	78
4.44.2.4	drawVerticalLine	78
4.44.2.5	fillRectangle	78
4.44.2.6	fillScreen	78
4.44.2.7	initializeDevice	79
4.44.2.8	invertDisplay	79
4.44.2.9	setPixel	79
4.44.2.10	setRotation	79
4.44.3	Member Data Documentation	79
4.44.3.1	BlackTab	79
4.44.3.2	GreenTab	80
4.44.3.3	Height	80
4.44.3.4	RedTab	80
4.44.3.5	TypeB	80
4.44.3.6	Width	80
4.45	TFT Class Reference	80
4.45.1	Detailed Description	83
4.45.2	Constructor & Destructor Documentation	83
4.45.2.1	TFT	83
4.45.2.2	TFT	83
4.45.2.3	TFT	83
4.45.3	Member Function Documentation	83
4.45.3.1	bgColorAt	83
4.45.3.2	closeWindow	84
4.45.3.3	color565	84
4.45.3.4	colorAt	84
4.45.3.5	deltaE	84
4.45.3.6	deltaOrth	84
4.45.3.7	displayOff	85
4.45.3.8	displayOn	85
4.45.3.9	drawBitmap	85
4.45.3.10	drawChar	85
4.45.3.11	drawCircle	86
4.45.3.12	drawCircleHelper	86
4.45.3.13	drawHorizontalLine	86
4.45.3.14	drawLine	86

4.45.3.15 drawRectangle	86
4.45.3.16 drawRGB	86
4.45.3.17 drawRGBA	87
4.45.3.18 drawRoundRect	87
4.45.3.19 drawTriangle	87
4.45.3.20 drawVerticalLine	87
4.45.3.21 fatalError	87
4.45.3.22 fillCircle	88
4.45.3.23 fillCircleHelper	88
4.45.3.24 fillRectangle	88
4.45.3.25 fillRoundRect	88
4.45.3.26 fillScreen	89
4.45.3.27 fillTriangle	89
4.45.3.28 getCursor	89
4.45.3.29 getCursorX	89
4.45.3.30 getCursorY	89
4.45.3.31 getHeight	90
4.45.3.32 getTextColor	90
4.45.3.33 getWidth	90
4.45.3.34 initializeDevice	90
4.45.3.35 invertDisplay	90
4.45.3.36 invertTextColor	91
4.45.3.37 mix	91
4.45.3.38 openWindow	91
4.45.3.39 rgb2hsv	91
4.45.3.40 rgb2xyz	91
4.45.3.41 setCursor	92
4.45.3.42 setFont	92
4.45.3.43 setFontScaleX	92
4.45.3.44 setFontScaleY	92
4.45.3.45 setPixel	92
4.45.3.46 setRotation	93
4.45.3.47 setTextColor	93
4.45.3.48 setTextColor	93
4.45.3.49 setTextWrap	93
4.45.3.50 stringHeight	93
4.45.3.51 stringWidth	94
4.45.3.52 windowData	94
4.45.3.53 windowData	94
4.45.3.54 write	94

4.45.3.55 xyz2lab	94
4.45.4 Member Data Documentation	95
4.45.4.1 _comm	95
4.45.4.2 _height	95
4.45.4.3 _width	95
4.45.4.4 cursor_x	95
4.45.4.5 cursor_y	95
4.45.4.6 font	95
4.45.4.7 font_scale_x	95
4.45.4.8 font_scale_y	95
4.45.4.9 rotation	95
4.45.4.10 textbgcolor	95
4.45.4.11 textcolor	95
4.45.4.12 wrap	96
4.46 TFTCommunicator Class Reference	96
4.46.1 Detailed Description	97
4.46.2 Member Function Documentation	97
4.46.2.1 blockData	97
4.46.2.2 blockData	97
4.46.2.3 blockData	97
4.46.2.4 initializeDevice	97
4.46.2.5 nativeWidth	97
4.46.2.6 readCommand16	97
4.46.2.7 readCommand32	97
4.46.2.8 readCommand8	98
4.46.2.9 readData16	98
4.46.2.10 readData32	98
4.46.2.11 readData8	98
4.46.2.12 streamCommand16	98
4.46.2.13 streamCommand16	98
4.46.2.14 streamCommand32	98
4.46.2.15 streamCommand32	98
4.46.2.16 streamCommand8	98
4.46.2.17 streamCommand8	99
4.46.2.18 streamData16	99
4.46.2.19 streamData16	99
4.46.2.20 streamData32	99
4.46.2.21 streamData32	99
4.46.2.22 streamData8	99
4.46.2.23 streamData8	99

4.46.2.24 streamEnd	99
4.46.2.25 streamStart	99
4.46.2.26 writeCommand16	100
4.46.2.27 writeCommand32	100
4.46.2.28 writeCommand8	100
4.46.2.29 writeData16	100
4.46.2.30 writeData32	100
4.46.2.31 writeData8	100
4.47 TFTDSPI Class Reference	100
4.47.1 Detailed Description	101
4.47.2 Constructor & Destructor Documentation	101
4.47.2.1 TFTDSPI	101
4.47.3 Member Function Documentation	101
4.47.3.1 blockData	101
4.47.3.2 blockData	102
4.47.3.3 blockData	102
4.47.3.4 initializeDevice	102
4.47.3.5 nativeWidth	102
4.47.3.6 readCommand16	102
4.47.3.7 readCommand32	102
4.47.3.8 readCommand8	102
4.47.3.9 readData16	102
4.47.3.10 readData32	102
4.47.3.11 readData8	103
4.47.3.12 streamCommand16	103
4.47.3.13 streamCommand16	103
4.47.3.14 streamCommand32	103
4.47.3.15 streamCommand32	103
4.47.3.16 streamCommand8	103
4.47.3.17 streamCommand8	103
4.47.3.18 streamData16	103
4.47.3.19 streamData16	103
4.47.3.20 streamData32	104
4.47.3.21 streamData32	104
4.47.3.22 streamData8	104
4.47.3.23 streamData8	104
4.47.3.24 streamEnd	104
4.47.3.25 streamStart	104
4.47.3.26 writeCommand16	104
4.47.3.27 writeCommand32	104

4.47.3.28 writeCommand8	104
4.47.3.29 writeData16	105
4.47.3.30 writeData32	105
4.47.3.31 writeData8	105
4.48 TFTPar16 Class Reference	105
4.48.1 Detailed Description	106
4.48.2 Constructor & Destructor Documentation	106
4.48.2.1 TFTPar16	106
4.48.2.2 TFTPar16	106
4.48.3 Member Function Documentation	106
4.48.3.1 blockData	106
4.48.3.2 blockData	106
4.48.3.3 blockData	107
4.48.3.4 initializeDevice	107
4.48.3.5 nativeWidth	107
4.48.3.6 readCommand16	107
4.48.3.7 readCommand32	107
4.48.3.8 readCommand8	107
4.48.3.9 readData16	107
4.48.3.10 readData32	107
4.48.3.11 readData8	107
4.48.3.12 streamCommand16	108
4.48.3.13 streamCommand16	108
4.48.3.14 streamCommand32	108
4.48.3.15 streamCommand32	108
4.48.3.16 streamCommand8	108
4.48.3.17 streamCommand8	108
4.48.3.18 streamData16	108
4.48.3.19 streamData16	108
4.48.3.20 streamData32	108
4.48.3.21 streamData32	109
4.48.3.22 streamData8	109
4.48.3.23 streamData8	109
4.48.3.24 streamEnd	109
4.48.3.25 streamStart	109
4.48.3.26 writeCommand16	109
4.48.3.27 writeCommand32	109
4.48.3.28 writeCommand8	109
4.48.3.29 writeData16	109
4.48.3.30 writeData32	110

4.48.3.31	writeData8	110
4.48.4	Member Data Documentation	110
4.48.4.1	IleadAdapter	110
4.49	TFTPar4 Class Reference	110
4.49.1	Detailed Description	111
4.49.2	Constructor & Destructor Documentation	111
4.49.2.1	TFTPar4	111
4.49.2.2	TFTPar4	111
4.49.3	Member Function Documentation	111
4.49.3.1	nativeWidth	111
4.49.3.2	streamCommand16	111
4.49.3.3	streamCommand32	111
4.49.3.4	streamCommand8	111
4.49.3.5	streamData16	112
4.49.3.6	streamData32	112
4.49.3.7	streamData8	112
4.49.3.8	streamEnd	112
4.49.3.9	streamStart	112
4.49.3.10	writeCommand16	112
4.49.3.11	writeCommand32	112
4.49.3.12	writeCommand8	112
4.49.3.13	writeData16	112
4.49.3.14	writeData32	113
4.49.3.15	writeData8	113
4.50	TFTPar8 Class Reference	113
4.50.1	Detailed Description	114
4.50.2	Constructor & Destructor Documentation	114
4.50.2.1	TFTPar8	114
4.50.3	Member Function Documentation	114
4.50.3.1	blockData	114
4.50.3.2	blockData	114
4.50.3.3	blockData	114
4.50.3.4	initializeDevice	114
4.50.3.5	nativeWidth	115
4.50.3.6	readCommand16	115
4.50.3.7	readCommand32	115
4.50.3.8	readCommand8	115
4.50.3.9	readData16	115
4.50.3.10	readData32	115
4.50.3.11	readData8	115

4.50.3.12 streamCommand16	115
4.50.3.13 streamCommand16	115
4.50.3.14 streamCommand32	116
4.50.3.15 streamCommand32	116
4.50.3.16 streamCommand8	116
4.50.3.17 streamCommand8	116
4.50.3.18 streamData16	116
4.50.3.19 streamData16	116
4.50.3.20 streamData32	116
4.50.3.21 streamData32	116
4.50.3.22 streamData8	116
4.50.3.23 streamData8	117
4.50.3.24 streamEnd	117
4.50.3.25 streamStart	117
4.50.3.26 writeCommand16	117
4.50.3.27 writeCommand32	117
4.50.3.28 writeCommand8	117
4.50.3.29 writeData16	117
4.50.3.30 writeData32	117
4.50.3.31 writeData8	117
4.51 TFTPMP Class Reference	118
4.51.1 Member Function Documentation	118
4.51.1.1 blockData	118
4.51.1.2 blockData	119
4.51.1.3 blockData	119
4.51.1.4 initializeDevice	119
4.51.1.5 nativeWidth	119
4.51.1.6 readCommand16	119
4.51.1.7 readCommand32	119
4.51.1.8 readCommand8	119
4.51.1.9 readData16	119
4.51.1.10 readData32	119
4.51.1.11 readData8	120
4.51.1.12 streamCommand16	120
4.51.1.13 streamCommand16	120
4.51.1.14 streamCommand32	120
4.51.1.15 streamCommand32	120
4.51.1.16 streamCommand8	120
4.51.1.17 streamCommand8	120
4.51.1.18 streamData16	120

4.51.1.19 streamData16	120
4.51.1.20 streamData32	121
4.51.1.21 streamData32	121
4.51.1.22 streamData8	121
4.51.1.23 streamData8	121
4.51.1.24 streamEnd	121
4.51.1.25 streamStart	121
4.51.1.26 writeCommand16	121
4.51.1.27 writeCommand32	121
4.51.1.28 writeCommand8	121
4.51.1.29 writeData16	122
4.51.1.30 writeData32	122
4.51.1.31 writeData8	122
4.52 TFTSoftSPI Class Reference	122
4.52.1 Detailed Description	123
4.52.2 Constructor & Destructor Documentation	123
4.52.2.1 TFTSoftSPI	123
4.52.3 Member Function Documentation	123
4.52.3.1 blockData	123
4.52.3.2 blockData	123
4.52.3.3 blockData	123
4.52.3.4 initializeDevice	124
4.52.3.5 nativeWidth	124
4.52.3.6 readCommand16	124
4.52.3.7 readCommand32	124
4.52.3.8 readCommand8	124
4.52.3.9 readData16	124
4.52.3.10 readData32	124
4.52.3.11 readData8	124
4.52.3.12 streamCommand16	124
4.52.3.13 streamCommand16	125
4.52.3.14 streamCommand32	125
4.52.3.15 streamCommand32	125
4.52.3.16 streamCommand8	125
4.52.3.17 streamCommand8	125
4.52.3.18 streamData16	125
4.52.3.19 streamData16	125
4.52.3.20 streamData32	125
4.52.3.21 streamData32	125
4.52.3.22 streamData8	126

4.52.3.23	streamData8	126
4.52.3.24	streamEnd	126
4.52.3.25	streamStart	126
4.52.3.26	writeCommand16	126
4.52.3.27	writeCommand32	126
4.52.3.28	writeCommand8	126
4.52.3.29	writeData16	126
4.52.3.30	writeData32	126
4.52.3.31	writeData8	127
4.53	Touch Class Reference	127
4.53.1	Constructor & Destructor Documentation	127
4.53.1.1	Touch	127
4.53.1.2	Touch	127
4.53.1.3	Touch	128
4.53.2	Member Function Documentation	128
4.53.2.1	initializeDevice	128
4.53.2.2	isPressed	128
4.53.2.3	pressure	128
4.53.2.4	sample	128
4.53.2.5	setRotation	128
4.53.2.6	x	129
4.53.2.7	y	129
4.53.3	Member Data Documentation	129
4.53.3.1	_comm	129
4.53.3.2	_height	129
4.53.3.3	_width	129
4.54	XPT2046 Class Reference	129
4.54.1	Constructor & Destructor Documentation	130
4.54.1.1	XPT2046	130
4.54.2	Member Function Documentation	130
4.54.2.1	initializeDevice	130
4.54.2.2	isPressed	130
4.54.2.3	sample	130
4.54.2.4	setRotation	130
4.54.2.5	x	131
4.54.2.6	y	131

Chapter 1

chipKIT-TFT

Universal [TFT](#) and other display device library for the chipKIT and PIC32 based boards.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

__attribute__	7
AggregatorList	10
BitmapFileHeader	15
BitmapInfoHeader	16
BitmapPixel24	16
BitmapPixel32	16
Color	17
coord	22
DataBlock	22
DataStore	23
SPISRAM	66
SRAM	69
FontHeader	24
Image	46
BMP	17
Raw565	59
Raw8	59
RLE	62
MatrixISRList	52
ParallelIO	53
CoreIO	22
MCP23S17	53
point3d	58
Print	
TFT	80
Aggregator	7
BD663474	12
Framebuffer	25
Framebuffer1	29
Framebuffer332	31
Framebuffer332Fast	33
Framebuffer565	35
HD44780	36
DOGMe	23
HX8357	40
ILI9340	43
KS0108	47

LEDMatrix	50
PICadillo35t	54
S6D0164	62
SSD1289	69
SSD1963	73
ST7735	76
sprite	67
TFTCommunicator	96
RawPar	60
TFTDSPI	100
TFTPar16	105
TFTPar4	110
TFTPar8	113
TFTPMP	118
TFTSoftSPI	122
Touch	127
AnalogTouch	11
XPT2046	129

Chapter 3

Class Index

3.1 Class List

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

__attribute__	7
Aggregator	7
AggregatorList	10
AnalogTouch	11
BD663474	12
BitmapFileHeader	15
BitmapInfoHeader	16
BitmapPixel24	16
BitmapPixel32	16
BMP	17
Color	17
coord	22
CoreIO	22
DataBlock	22
DataStore	23
DOGMe	23
FontHeader	24
Framebuffer	25
Framebuffer1	29
Framebuffer332	31
Framebuffer332Fast	33
Framebuffer565	35
HD44780	36
HX8357	40
ILI9340	43
Image	46
KS0108	47
LEDMatrix	50
MatrixISRList	52
MCP23S17	53
ParallelIO	53
PICadillo35t	54
point3d	58
Raw565	59
Raw8	59
RawPar	60
RLE	62
S6D0164	62

SPISRAM	66
sprite	67
SRAM	69
SSD1289	69
SSD1963	73
ST7735	76
TFT	80
TFTCommunicator	96
TFTDSPI	100
TFTPar16	105
TFTPar4	110
TFTPar8	113
TFTPMP	118
TFTSoftSPI	122
Touch	127
XPT2046	129

Chapter 4

Class Documentation

4.1 `__attribute__` Struct Reference

Public Member Functions

- union {
 uint16_t value
} `__attribute__` ((packed))

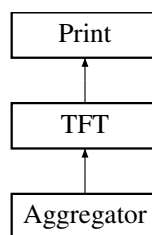
The documentation for this struct was generated from the following file:

- TFT.h

4.2 Aggregator Class Reference

```
#include <Aggregator.h>
```

Inheritance diagram for Aggregator:



Public Member Functions

- void `fillScreen` (uint16_t color)
- void `setPixel` (int16_t x, int16_t y, uint16_t color)
- void `setRotation` (uint8_t r)
- void `invertDisplay` (boolean i)
- void `displayOn` ()
- void `displayOff` ()
- void `initializeDevice` ()
- void `addDisplay` (TFT *d, int16_t x, int16_t y)

- void [drawHorizontalLine](#) (int16_t x, int16_t y, int16_t w, uint16_t c)
- void [drawVerticalLine](#) (int16_t x, int16_t y, int16_t h, uint16_t c)
- uint16_t [getWidth](#) ()
- uint16_t [getHeight](#) ()

Protected Attributes

- struct [AggregatorList](#) * **_displays**
- uint16_t **_width**
- uint16_t **_height**

Additional Inherited Members

4.2.1 Detailed Description

The aggregator takes one or more physical screens and combines them into a single virtual display. Screens can occupy any location within the virtual display space and do not need to be contiguous.

4.2.2 Member Function Documentation

4.2.2.1 void Aggregator::addDisplay (TFT * d, int16_t x, int16_t y)

Add a display

You add displays to the aggregator virtual display with this function. A display is passed as a pointer, along with the X/Y coordinates of the upper-left point of the screen in virtual display space.

Example:

```
agg.addDisplay(&tft, 0, 320);
```

4.2.2.2 void Aggregator::displayOff () [inline],[virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.2.2.3 void Aggregator::displayOn () [inline],[virtual]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.2.2.4 void Aggregator::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color) [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.2.2.5 void Aggregator::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.2.2.6 void Aggregator::fillScreen (uint16_t color) [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.2.2.7 uint16_t Aggregator::getHeight () [inline],[virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

```
int height = tft.getHeight();
```

Reimplemented from [TFT](#).

4.2.2.8 uint16_t Aggregator::getWidth () [inline],[virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

```
int width = tft.getWidth();
```

Reimplemented from [TFT](#).

4.2.2.9 void Aggregator::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.2.2.10 void Aggregator::invertDisplay (boolean i) [virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.2.2.11 void Aggregator::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.2.2.12 void Aggregator::setRotation (uint8_t rotation) [inline],[virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

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

- Aggregator.h
- Aggregator.cpp

4.3 AggregatorList Struct Reference

Public Attributes

- [TFT](#) * display

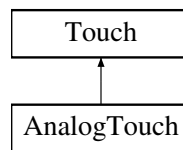
- int16_t **x**
- int16_t **y**
- struct [AggregatorList](#) * **next**

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

- Aggregator.h

4.4 AnalogTouch Class Reference

Inheritance diagram for AnalogTouch:



Public Member Functions

- **AnalogTouch** (uint8_t xl, uint8_t xr, uint8_t yu, uint8_t yd, uint16_t w, uint16_t h)
- void [sample](#) ()
- int **getSample** (uint8_t)
- uint16_t [x](#) ()
- uint16_t [y](#) ()
- boolean [isPressed](#) ()
- void [initializeDevice](#) ()
- void **scaleX** (float v)
- void **scaleY** (float v)
- void **offsetX** (int16_t v)
- void **offsetY** (int16_t v)
- uint16_t [pressure](#) ()
- void [setRotation](#) (uint8_t r)

Additional Inherited Members

4.4.1 Member Function Documentation

4.4.1.1 void AnalogTouch::initializeDevice () [virtual]

Initialize the device

This configures and enables the touch screen device. It should be called before any other touch screen functions.

Implements [Touch](#).

4.4.1.2 boolean AnalogTouch::isPressed () [virtual]

Get pressed status

Returns true if the touch screen is pressed, false otherwise.

Implements [Touch](#).

4.4.1.3 uint16_t AnalogTouch::pressure () [virtual]

Calculate the touch pressure

For touch screens that can calculate how hard you are pressing them, this returns the pressure value. For others it returns 0.

Example:

```
int pressure = ts.pressure();
```

Reimplemented from [Touch](#).

4.4.1.4 void AnalogTouch::sample () [virtual]

Sample the touch screen

This performs a sampling of the touch screen to get the current coordinates and touch status. It should be called regularly to update the current touch screen data.

Implements [Touch](#).

4.4.1.5 void AnalogTouch::setRotation (uint8_t r) [virtual]

Set rotation

This sets the screen orientation of the touch screen. It should be set to the same as the rotation used for the screen.

Reimplemented from [Touch](#).

4.4.1.6 uint16_t AnalogTouch::x () [virtual]

Get X coordinate

This returns the X coordinate of the current touch position.

Implements [Touch](#).

4.4.1.7 uint16_t AnalogTouch::y () [virtual]

Get Y coordinate

This returns the Y coordinate of the current touch position.

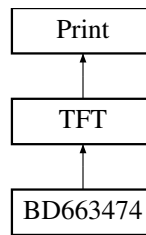
Implements [Touch](#).

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

- AnalogTouch.h
- AnalogTouch.cpp

4.5 BD663474 Class Reference

Inheritance diagram for BD663474:



Public Member Functions

- **BD663474** ([TFTCommunicator](#) *comms)
- **BD663474** ([TFTCommunicator](#) &comms)
- void **setAddrWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **fillScreen** (uint16_t color)
- void **setPixel** (int16_t x, int16_t y, uint16_t color)
- void **drawVerticalLine** (int16_t x, int16_t y, int16_t h, uint16_t color)
- void **drawHorizontalLine** (int16_t x, int16_t y, int16_t w, uint16_t color)
- void **fillRectangle** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void **setRotation** (uint8_t r)
- void **invertDisplay** (boolean i)
- void **displayOn** ()
- void **displayOff** ()
- void **initializeDevice** ()

Static Public Attributes

- static const uint16_t **Width** = 240
- static const uint16_t **Height** = 320

Additional Inherited Members

4.5.1 Member Function Documentation

4.5.1.1 void **BD663474::displayOff** () [[inline](#)],[[virtual](#)]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.5.1.2 void **BD663474::displayOn** () [[inline](#)],[[virtual](#)]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.5.1.3 void BD663474::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color) [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.5.1.4 void BD663474::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.5.1.5 void BD663474::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRectangle(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.5.1.6 void BD663474::fillScreen (uint16_t color) [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.5.1.7 void BD663474::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:


```
tft.initializeDevice();
```

Implements [TFT](#).

4.5.1.8 void BD663474::invertDisplay (boolean *i*) [virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.5.1.9 void BD663474::setPixel (int16_t *x*, int16_t *y*, uint16_t *color*) [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.5.1.10 void BD663474::setRotation (uint8_t *rotation*) [virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

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

- BD663474.h
- BD663474.cpp

4.6 BitmapFileHeader Struct Reference

Public Attributes

- uint16_t **bfType**
- uint32_t **bfSize**
- uint16_t **bfReserved1**
- uint16_t **bfReserved2**
- uint32_t **bfBitmapOffset**

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

- BMP.h

4.7 BitmapInfoHeader Struct Reference

Public Attributes

- uint32_t **biSize**
- int32_t **biWidth**
- int32_t **biHeight**
- uint16_t **biPlanes**
- uint16_t **biBitCount**
- uint32_t **biCompression**
- uint32_t **biSizeImage**
- int32_t **biXPelsPerMeter**
- int32_t **biYPelsPerMeter**
- uint32_t **biClrUsed**
- uint32_t **biClrImportant**
- uint32_t **biMaskRed**
- uint32_t **biMaskGreen**
- uint32_t **biMaskBlue**
- uint32_t **biMaskAlpha**

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

- BMP.h

4.8 BitmapPixel24 Struct Reference

Public Attributes

- uint8_t **b**
- uint8_t **g**
- uint8_t **r**

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

- BMP.h

4.9 BitmapPixel32 Struct Reference

Public Member Functions

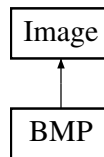
- union {
 uint32_t **value**
} **__attribute__((packed))**

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

- BMP.h

4.10 BMP Class Reference

Inheritance diagram for BMP:



Public Member Functions

- **BMP** (const char *data)
- void **draw** (TFT *dev, int16_t x, int16_t y)
- void **draw** (TFT *dev, int16_t x, int16_t y, uint16_t t)
- void **drawTransformed** (TFT *dev, int16_t x, int16_t y, uint8_t transform)
- void **drawTransformed** (TFT *dev, int16_t x, int16_t y, uint8_t transform, uint16_t t)

Public Attributes

- const char * **_data**
- const char * **_image**
- struct [BitmapFileHeader](#) * **_header**
- struct [BitmapInfoHeader](#) * **_info**
- struct [BitmapPixel32](#) * **_palette**
- uint16_t **_paletteSize**

Additional Inherited Members

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

- BMP.h
- BMP.cpp

4.11 Color Class Reference

Static Public Attributes

- static const uint16_t **Snow** = RGB(255,250,250)
- static const uint16_t **GhostWhite** = RGB(248,248,255)
- static const uint16_t **WhiteSmoke** = RGB(245,245,245)
- static const uint16_t **Gainsboro** = RGB(220,220,220)
- static const uint16_t **FloralWhite** = RGB(255,250,240)
- static const uint16_t **OldLace** = RGB(253,245,230)
- static const uint16_t **Linen** = RGB(250,240,230)
- static const uint16_t **AntiqueWhite** = RGB(250,235,215)
- static const uint16_t **PapayaWhip** = RGB(255,239,213)
- static const uint16_t **BlanchedAlmond** = RGB(255,235,205)
- static const uint16_t **Bisque** = RGB(255,228,196)
- static const uint16_t **PeachPuff** = RGB(255,218,185)
- static const uint16_t **NavajoWhite** = RGB(255,222,173)

- static const uint16_t **Moccasin** = RGB(255,228,181)
- static const uint16_t **Cornsilk** = RGB(255,248,220)
- static const uint16_t **Ivory** = RGB(255,255,240)
- static const uint16_t **LemonChiffon** = RGB(255,250,205)
- static const uint16_t **Seashell** = RGB(255,245,238)
- static const uint16_t **Honeydew** = RGB(240,255,240)
- static const uint16_t **MintCream** = RGB(245,255,250)
- static const uint16_t **Azure** = RGB(240,255,255)
- static const uint16_t **AliceBlue** = RGB(240,248,255)
- static const uint16_t **Lavender** = RGB(230,230,250)
- static const uint16_t **LavenderBlush** = RGB(255,240,245)
- static const uint16_t **MistyRose** = RGB(255,228,225)
- static const uint16_t **White** = RGB(255,255,255)
- static const uint16_t **Black** = RGB(0, 0, 0)
- static const uint16_t **DarkSlateGray** = RGB(47, 79, 79)
- static const uint16_t **DimGray** = RGB(105,105,105)
- static const uint16_t **SlateGray** = RGB(112,128,144)
- static const uint16_t **LightSlateGray** = RGB(119,136,153)
- static const uint16_t **Gray** = RGB(190,190,190)
- static const uint16_t **LightGray** = RGB(211,211,211)
- static const uint16_t **MidnightBlue** = RGB(25, 25,112)
- static const uint16_t **Navy** = RGB(0, 0,128)
- static const uint16_t **NavyBlue** = RGB(0, 0,128)
- static const uint16_t **CornflowerBlue** = RGB(100,149,237)
- static const uint16_t **DarkSlateBlue** = RGB(72, 61,139)
- static const uint16_t **SlateBlue** = RGB(106, 90,205)
- static const uint16_t **MediumSlateBlue** = RGB(123,104,238)
- static const uint16_t **LightSlateBlue** = RGB(132,112,255)
- static const uint16_t **MediumBlue** = RGB(0, 0,205)
- static const uint16_t **RoyalBlue** = RGB(65,105,225)
- static const uint16_t **Blue** = RGB(0, 0,255)
- static const uint16_t **DodgerBlue** = RGB(30,144,255)
- static const uint16_t **DeepSkyBlue** = RGB(0,191,255)
- static const uint16_t **SkyBlue** = RGB(135,206,235)
- static const uint16_t **LightSkyBlue** = RGB(135,206,250)
- static const uint16_t **SteelBlue** = RGB(70,130,180)
- static const uint16_t **LightSteelBlue** = RGB(176,196,222)
- static const uint16_t **LightBlue** = RGB(173,216,230)
- static const uint16_t **PowderBlue** = RGB(176,224,230)
- static const uint16_t **PaleTurquoise** = RGB(175,238,238)
- static const uint16_t **DarkTurquoise** = RGB(0,206,209)
- static const uint16_t **MediumTurquoise** = RGB(72,209,204)
- static const uint16_t **Turquoise** = RGB(64,224,208)
- static const uint16_t **Cyan** = RGB(0,255,255)
- static const uint16_t **LightCyan** = RGB(224,255,255)
- static const uint16_t **CadetBlue** = RGB(95,158,160)
- static const uint16_t **MediumAquamarine** = RGB(102,205,170)
- static const uint16_t **Aquamarine** = RGB(127,255,212)
- static const uint16_t **DarkGreen** = RGB(0,100, 0)
- static const uint16_t **DarkOliveGreen** = RGB(85,107, 47)
- static const uint16_t **DarkSeaGreen** = RGB(143,188,143)
- static const uint16_t **SeaGreen** = RGB(46,139, 87)
- static const uint16_t **MediumSeaGreen** = RGB(60,179,113)
- static const uint16_t **LightSeaGreen** = RGB(32,178,170)
- static const uint16_t **PaleGreen** = RGB(152,251,152)

- static const uint16_t **SpringGreen** = RGB(0,255,127)
- static const uint16_t **LawnGreen** = RGB(124,252, 0)
- static const uint16_t **Green** = RGB(0,255, 0)
- static const uint16_t **Chartreuse** = RGB(127,255, 0)
- static const uint16_t **MediumSpringGreen** = RGB(0,250,154)
- static const uint16_t **GreenYellow** = RGB(173,255, 47)
- static const uint16_t **LimeGreen** = RGB(50,205, 50)
- static const uint16_t **YellowGreen** = RGB(154,205, 50)
- static const uint16_t **ForestGreen** = RGB(34,139, 34)
- static const uint16_t **OliveDrab** = RGB(107,142, 35)
- static const uint16_t **DarkKhaki** = RGB(189,183,107)
- static const uint16_t **Khaki** = RGB(240,230,140)
- static const uint16_t **PaleGoldenrod** = RGB(238,232,170)
- static const uint16_t **LightGoldenrodYellow** = RGB(250,250,210)
- static const uint16_t **LightYellow** = RGB(255,255,224)
- static const uint16_t **Yellow** = RGB(255,255, 0)
- static const uint16_t **Gold** = RGB(255,215, 0)
- static const uint16_t **LightGoldenrod** = RGB(238,221,130)
- static const uint16_t **Goldenrod** = RGB(218,165, 32)
- static const uint16_t **DarkGoldenrod** = RGB(184,134, 11)
- static const uint16_t **RosyBrown** = RGB(188,143,143)
- static const uint16_t **IndianRed** = RGB(205, 92, 92)
- static const uint16_t **SaddleBrown** = RGB(139, 69, 19)
- static const uint16_t **Sienna** = RGB(160, 82, 45)
- static const uint16_t **Peru** = RGB(205,133, 63)
- static const uint16_t **Burlywood** = RGB(222,184,135)
- static const uint16_t **Beige** = RGB(245,245,220)
- static const uint16_t **Wheat** = RGB(245,222,179)
- static const uint16_t **SandyBrown** = RGB(244,164, 96)
- static const uint16_t **Tan** = RGB(210,180,140)
- static const uint16_t **Chocolate** = RGB(210,105, 30)
- static const uint16_t **Firebrick** = RGB(178, 34, 34)
- static const uint16_t **Brown** = RGB(165, 42, 42)
- static const uint16_t **DarkSalmon** = RGB(233,150,122)
- static const uint16_t **Salmon** = RGB(250,128,114)
- static const uint16_t **LightSalmon** = RGB(255,160,122)
- static const uint16_t **Orange** = RGB(255,165, 0)
- static const uint16_t **DarkOrange** = RGB(255,140, 0)
- static const uint16_t **Coral** = RGB(255,127, 80)
- static const uint16_t **LightCoral** = RGB(240,128,128)
- static const uint16_t **Tomato** = RGB(255, 99, 71)
- static const uint16_t **OrangeRed** = RGB(255, 69, 0)
- static const uint16_t **Red** = RGB(255, 0, 0)
- static const uint16_t **HotPink** = RGB(255,105,180)
- static const uint16_t **DeepPink** = RGB(255, 20,147)
- static const uint16_t **Pink** = RGB(255,192,203)
- static const uint16_t **LightPink** = RGB(255,182,193)
- static const uint16_t **PaleVioletRed** = RGB(219,112,147)
- static const uint16_t **Maroon** = RGB(176, 48, 96)
- static const uint16_t **MediumVioletRed** = RGB(199, 21,133)
- static const uint16_t **VioletRed** = RGB(208, 32,144)
- static const uint16_t **Magenta** = RGB(255, 0,255)
- static const uint16_t **Violet** = RGB(238,130,238)
- static const uint16_t **Plum** = RGB(221,160,221)
- static const uint16_t **Orchid** = RGB(218,112,214)

- static const uint16_t **MediumOrchid** = RGB(186, 85,211)
- static const uint16_t **DarkOrchid** = RGB(153, 50,204)
- static const uint16_t **DarkViolet** = RGB(148, 0,211)
- static const uint16_t **BlueViolet** = RGB(138, 43,226)
- static const uint16_t **Purple** = RGB(160, 32,240)
- static const uint16_t **MediumPurple** = RGB(147,112,219)
- static const uint16_t **Thistle** = RGB(216,191,216)
- static const uint16_t **Gray0** = RGB(0, 0, 0)
- static const uint16_t **Gray1** = RGB(3, 3, 3)
- static const uint16_t **Gray2** = RGB(5, 5, 5)
- static const uint16_t **Gray3** = RGB(8, 8, 8)
- static const uint16_t **Gray4** = RGB(10, 10, 10)
- static const uint16_t **Gray5** = RGB(13, 13, 13)
- static const uint16_t **Gray6** = RGB(15, 15, 15)
- static const uint16_t **Gray7** = RGB(18, 18, 18)
- static const uint16_t **Gray8** = RGB(20, 20, 20)
- static const uint16_t **Gray9** = RGB(23, 23, 23)
- static const uint16_t **Gray10** = RGB(26, 26, 26)
- static const uint16_t **Gray11** = RGB(28, 28, 28)
- static const uint16_t **Gray12** = RGB(31, 31, 31)
- static const uint16_t **Gray13** = RGB(33, 33, 33)
- static const uint16_t **Gray14** = RGB(36, 36, 36)
- static const uint16_t **Gray15** = RGB(38, 38, 38)
- static const uint16_t **Gray16** = RGB(41, 41, 41)
- static const uint16_t **Gray17** = RGB(43, 43, 43)
- static const uint16_t **Gray18** = RGB(46, 46, 46)
- static const uint16_t **Gray19** = RGB(48, 48, 48)
- static const uint16_t **Gray20** = RGB(51, 51, 51)
- static const uint16_t **Gray21** = RGB(54, 54, 54)
- static const uint16_t **Gray22** = RGB(56, 56, 56)
- static const uint16_t **Gray23** = RGB(59, 59, 59)
- static const uint16_t **Gray24** = RGB(61, 61, 61)
- static const uint16_t **Gray25** = RGB(64, 64, 64)
- static const uint16_t **Gray26** = RGB(66, 66, 66)
- static const uint16_t **Gray27** = RGB(69, 69, 69)
- static const uint16_t **Gray28** = RGB(71, 71, 71)
- static const uint16_t **Gray29** = RGB(74, 74, 74)
- static const uint16_t **Gray30** = RGB(77, 77, 77)
- static const uint16_t **Gray31** = RGB(79, 79, 79)
- static const uint16_t **Gray32** = RGB(82, 82, 82)
- static const uint16_t **Gray33** = RGB(84, 84, 84)
- static const uint16_t **Gray34** = RGB(87, 87, 87)
- static const uint16_t **Gray35** = RGB(89, 89, 89)
- static const uint16_t **Gray36** = RGB(92, 92, 92)
- static const uint16_t **Gray37** = RGB(94, 94, 94)
- static const uint16_t **Gray38** = RGB(97, 97, 97)
- static const uint16_t **Gray39** = RGB(99, 99, 99)
- static const uint16_t **Gray40** = RGB(102,102,102)
- static const uint16_t **Gray41** = RGB(105,105,105)
- static const uint16_t **Gray42** = RGB(107,107,107)
- static const uint16_t **Gray43** = RGB(110,110,110)
- static const uint16_t **Gray44** = RGB(112,112,112)
- static const uint16_t **Gray45** = RGB(115,115,115)
- static const uint16_t **Gray46** = RGB(117,117,117)
- static const uint16_t **Gray47** = RGB(120,120,120)

- static const uint16_t **Gray48** = RGB(122,122,122)
- static const uint16_t **Gray49** = RGB(125,125,125)
- static const uint16_t **Gray50** = RGB(127,127,127)
- static const uint16_t **Gray51** = RGB(130,130,130)
- static const uint16_t **Gray52** = RGB(133,133,133)
- static const uint16_t **Gray53** = RGB(135,135,135)
- static const uint16_t **Gray54** = RGB(138,138,138)
- static const uint16_t **Gray55** = RGB(140,140,140)
- static const uint16_t **Gray56** = RGB(143,143,143)
- static const uint16_t **Gray57** = RGB(145,145,145)
- static const uint16_t **Gray58** = RGB(148,148,148)
- static const uint16_t **Gray59** = RGB(150,150,150)
- static const uint16_t **Gray60** = RGB(153,153,153)
- static const uint16_t **Gray61** = RGB(156,156,156)
- static const uint16_t **Gray62** = RGB(158,158,158)
- static const uint16_t **Gray63** = RGB(161,161,161)
- static const uint16_t **Gray64** = RGB(163,163,163)
- static const uint16_t **Gray65** = RGB(166,166,166)
- static const uint16_t **Gray66** = RGB(168,168,168)
- static const uint16_t **Gray67** = RGB(171,171,171)
- static const uint16_t **Gray68** = RGB(173,173,173)
- static const uint16_t **Gray69** = RGB(176,176,176)
- static const uint16_t **Gray70** = RGB(179,179,179)
- static const uint16_t **Gray71** = RGB(181,181,181)
- static const uint16_t **Gray72** = RGB(184,184,184)
- static const uint16_t **Gray73** = RGB(186,186,186)
- static const uint16_t **Gray74** = RGB(189,189,189)
- static const uint16_t **Gray75** = RGB(191,191,191)
- static const uint16_t **Gray76** = RGB(194,194,194)
- static const uint16_t **Gray77** = RGB(196,196,196)
- static const uint16_t **Gray78** = RGB(199,199,199)
- static const uint16_t **Gray79** = RGB(201,201,201)
- static const uint16_t **Gray80** = RGB(204,204,204)
- static const uint16_t **Gray81** = RGB(207,207,207)
- static const uint16_t **Gray82** = RGB(209,209,209)
- static const uint16_t **Gray83** = RGB(212,212,212)
- static const uint16_t **Gray84** = RGB(214,214,214)
- static const uint16_t **Gray85** = RGB(217,217,217)
- static const uint16_t **Gray86** = RGB(219,219,219)
- static const uint16_t **Gray87** = RGB(222,222,222)
- static const uint16_t **Gray88** = RGB(224,224,224)
- static const uint16_t **Gray89** = RGB(227,227,227)
- static const uint16_t **Gray90** = RGB(229,229,229)
- static const uint16_t **Gray91** = RGB(232,232,232)
- static const uint16_t **Gray92** = RGB(235,235,235)
- static const uint16_t **Gray93** = RGB(237,237,237)
- static const uint16_t **Gray94** = RGB(240,240,240)
- static const uint16_t **Gray95** = RGB(242,242,242)
- static const uint16_t **Gray96** = RGB(245,245,245)
- static const uint16_t **Gray97** = RGB(247,247,247)
- static const uint16_t **Gray98** = RGB(250,250,250)
- static const uint16_t **Gray99** = RGB(252,252,252)
- static const uint16_t **Gray100** = RGB(255,255,255)
- static const uint16_t **DarkGray** = RGB(169,169,169)
- static const uint16_t **DarkBlue** = RGB(0 , 0,139)

- static const uint16_t **DarkCyan** = RGB(0 ,139,139)
- static const uint16_t **DarkMagenta** = RGB(139, 0,139)
- static const uint16_t **DarkRed** = RGB(139, 0, 0)
- static const uint16_t **LightGreen** = RGB(144,238,144)

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

- Color.h

4.12 coord Struct Reference

Public Attributes

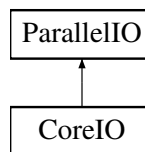
- uint16_t **x**
- uint16_t **y**

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

- TFT.h

4.13 CoreIO Class Reference

Inheritance diagram for CoreIO:



Public Member Functions

- uint16_t **pins** ()
- void **digitalWrite** (uint16_t pin, uint8_t val)
- uint8_t **digitalRead** (uint16_t pin)
- void **pinMode** (uint16_t pin, uint8_t mode)
- void **startBuffer** ()
- void **endBuffer** ()

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

- CoreIO.h

4.14 DataBlock Class Reference

Public Member Functions

- **DataBlock** (uint32_t start, uint32_t len, [DataStore](#) *store)
- uint8_t **operator[]** (uint32_t a)
- void **set** (uint32_t a, uint8_t v)

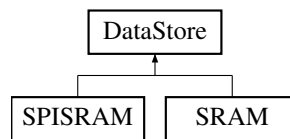
- uint8_t **get** (uint32_t a)

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

- DataStore.h
- DataStore.cpp

4.15 DataStore Class Reference

Inheritance diagram for DataStore:



Public Member Functions

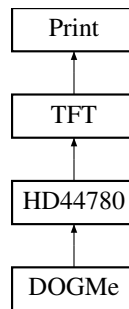
- virtual uint8_t **read8** (uint32_t addr)=0
- virtual uint16_t **read16** (uint32_t addr)=0
- virtual uint32_t **read32** (uint32_t addr)=0
- virtual void **read8** (uint32_t addr, uint8_t *data, uint32_t len)=0
- virtual void **read16** (uint32_t addr, uint16_t *data, uint32_t len)=0
- virtual void **read32** (uint32_t addr, uint32_t *data, uint32_t len)=0
- virtual void **write8** (uint32_t addr, uint8_t data)=0
- virtual void **write16** (uint32_t addr, uint16_t data)=0
- virtual void **write32** (uint32_t addr, uint32_t data)=0
- virtual void **write8** (uint32_t addr, uint8_t *data, uint32_t len)=0
- virtual void **write16** (uint32_t addr, uint16_t *data, uint32_t len)=0
- virtual void **write32** (uint32_t addr, uint32_t *data, uint32_t len)=0
- virtual void **setAll8** (uint8_t data)
- virtual void **setAll16** (uint16_t data)
- virtual void **setAll32** (uint32_t data)
- virtual void **initializeDevice** ()=0
- virtual uint32_t **size** ()=0
- [DataBlock](#) & **allocate** (uint32_t s)
- void **free** ([DataBlock](#) &b)

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

- DataStore.h
- DataStore.cpp

4.16 DOGMe Class Reference

Inheritance diagram for DOGMe:



Public Member Functions

- **DOGMe** ([TFTCommunicator](#) *comm, uint8_t w, uint8_t h)
- **DOGMe** ([TFTCommunicator](#) &comm, uint8_t w, uint8_t h)
- void [initializeDevice](#) ()
- void **table** (uint8_t tab)
- void **setBits** (uint8_t b)
- void **setLines** (uint8_t l)
- void **setFunction** ()
- void **setContrast** (uint8_t c)
- void **setBias** (uint8_t b)
- void **setFollower** (uint8_t f)
- void **singleHeight** ()
- void **doubleHeight** ()
- void **write** (uint8_t c)

Additional Inherited Members

4.16.1 Member Function Documentation

4.16.1.1 void **DOGMe::initializeDevice** () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

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

- DOGMe.h
- DOGMe.cpp

4.17 FontHeader Struct Reference

Public Attributes

- uint8_t **linesPerCharacter**

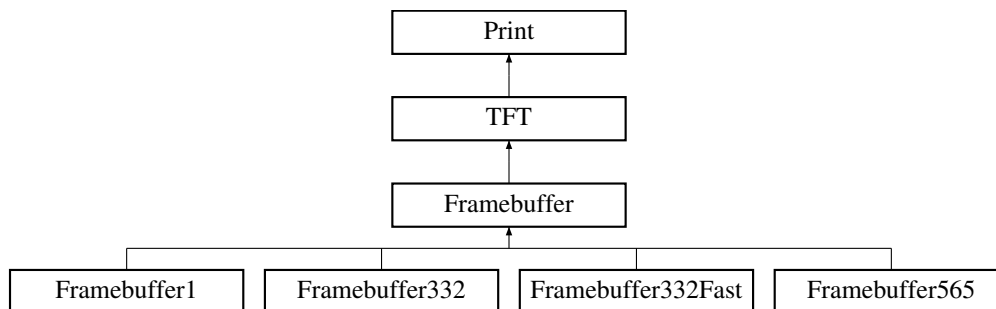
- `uint8_t bytesPerLine`
- `uint8_t startGlyph`
- `uint8_t endGlyph`
- `uint8_t bitsPerPixel`

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

- `TFT.h`

4.18 Framebuffer Class Reference

Inheritance diagram for Framebuffer:



Public Member Functions

- **Framebuffer** (`int16_t w`, `int16_t h`, `DataStore *b`)
- virtual void **setPixel** (`int16_t x`, `int16_t y`, `uint16_t c`)
- virtual void **drawVerticalLine** (`int16_t x`, `int16_t y`, `int16_t h`, `uint16_t color`)
- virtual void **drawHorizontalLine** (`int16_t x`, `int16_t y`, `int16_t w`, `uint16_t color`)
- virtual void **fillRect** (`int16_t x`, `int16_t y`, `int16_t w`, `int16_t h`, `uint16_t color`)
- virtual void **fillScreen** (`uint16_t`)
- virtual void **setAntiAlias** (`uint8_t aa`)
- virtual void **copyRect** (`int16_t dx`, `int16_t dy`, `int16_t sx`, `int16_t sy`, `uint16_t w`, `uint16_t h`)
- virtual void **setColor** (`uint8_t color`, `uint16_t rgb`)
- virtual void **setColor** (`uint8_t color`, `uint8_t r`, `uint8_t g`, `uint8_t b`)
- virtual void **loadPalette** (`const uint16_t *p`)
- virtual void **loadPalette** (`const uint8_t p[256][3]`)
- virtual void **loadPalette** (`Framebuffer *fb`)
- virtual `uint16_t` **colorAt** (`int16_t x`, `int16_t y`)
- virtual `uint16_t` **bgColorAt** (`int16_t x`, `int16_t y`)
- virtual void **getScanLine** (`uint16_t y`, `uint16_t *data`)
- virtual void **getScanLine** (`uint16_t y`, `uint16_t x`, `uint16_t w`, `uint16_t *data`)
- virtual `struct sprite *` **addSprite** (`const uint8_t *data`, `uint16_t w`, `uint16_t h`, `uint8_t t`, `uint8_t f`)
- virtual void **removeSprite** (`struct sprite *s`)
- virtual void **moveTo** (`struct sprite *s`, `int16_t x`, `int16_t y`)
- virtual void **moveBy** (`struct sprite *s`, `int16_t dx`, `int16_t dy`)
- virtual `struct sprite *` **spriteAt** (`int16_t x`, `int16_t y`)
- virtual void **animate** (`struct sprite *s`)
- virtual void **animatePingPong** (`struct sprite *s`)
- virtual `struct sprite *` **collidesWith** (`struct sprite *s`)
- virtual `struct sprite *` **firstSprite** ()
- virtual `struct sprite *` **nextSprite** ()

- virtual int8_t **getSprite** (struct [sprite](#) *s, uint8_t n)
- virtual void **setSprite** (struct [sprite](#) *s, uint8_t n, int8_t v)
- virtual void [initializeDevice](#) ()
- virtual void [displayOn](#) ()
- virtual void [displayOff](#) ()
- virtual void [invertDisplay](#) (boolean i)
- virtual uint8_t **bufferRead** (uint32_t addr)
- virtual void **bufferWrite** (uint32_t addr, uint8_t value)
- virtual void [setRotation](#) (uint8_t rot)
- virtual uint16_t [getWidth](#) ()
- virtual uint16_t [getHeight](#) ()
- virtual uint8_t **getClosestColor** (uint16_t c)
- void **translateCoordinates** (int16_t *x, int16_t *y)
- virtual void **scroll** (int16_t dx, int16_t dy)
- virtual void **update** ([TFT](#) *tft)
- virtual void **update** ([TFT](#) *tft, int16_t x0, int16_t y0)
- virtual void **update** ([TFT](#) &tft)
- virtual void **update** ([TFT](#) &tft, int16_t x0, int16_t y0)

Public Attributes

- [DataStore](#) * **buffer**
- uint16_t **palette** [256]
- struct [sprite](#) * **sprites**
- struct [sprite](#) * **selectedSprite**
- int32_t **_minX**
- int32_t **_minY**
- int32_t **_maxX**
- int32_t **_maxY**

Static Public Attributes

- static const uint8_t **MirrorH** = 0x01
- static const uint8_t **MirrorV** = 0x02
- static const uint8_t **Rotate180** = 0x03

Protected Attributes

- uint16_t **_masterWidth**
- uint16_t **_masterHeight**
- uint8_t **_antiAlias**

4.18.1 Member Function Documentation

4.18.1.1 uint16_t Framebuffer::bgColorAt (int16_t x, int16_t y) [virtual]

Get the raw colour at a location

Returns the base image colour at (x,y) before any further layers or post processing effects are performed.

Example:

```
unsigned int color = tft.bgColorAt(100, 100);
```

Reimplemented from [TFT](#).

Reimplemented in [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.18.1.2 `uint16_t Framebuffer::colorAt(int16_t x, int16_t y)` [virtual]

Get the colour at a location

Returns the colour at (x,y) as seen by the screen.

Example:

```
unsigned int color = tft.colorAt(100, 100);
```

Reimplemented from [TFT](#).

Reimplemented in [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.18.1.3 `virtual void Framebuffer::displayOff()` [inline],[virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.18.1.4 `virtual void Framebuffer::displayOn()` [inline],[virtual]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.18.1.5 `void Framebuffer::drawHorizontalLine(int16_t x, int16_t y, int16_t w, uint16_t color)` [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

Reimplemented in [Framebuffer332Fast](#).

4.18.1.6 `void Framebuffer::drawVerticalLine(int16_t x, int16_t y, int16_t h, uint16_t color)` [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.18.1.7 `void Framebuffer::fillScreen (uint16_t color)` [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

Reimplemented in [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.18.1.8 `uint16_t Framebuffer::getHeight ()` [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

```
int height = tft.getHeight();
```

Reimplemented from [TFT](#).

4.18.1.9 `uint16_t Framebuffer::getWidth ()` [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

```
int width = tft.getWidth();
```

Reimplemented from [TFT](#).

4.18.1.10 `void Framebuffer::initializeDevice ()` [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

Reimplemented in [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.18.1.11 `virtual void Framebuffer::invertDisplay (boolean i)` `[inline],[virtual]`

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.18.1.12 `void Framebuffer::setPixel (int16_t x, int16_t y, uint16_t color)` `[virtual]`

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

Reimplemented in [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.18.1.13 `void Framebuffer::setRotation (uint8_t rotation)` `[virtual]`

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

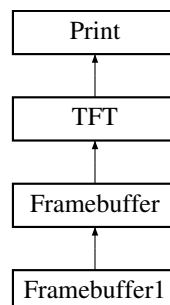
Implements [TFT](#).

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

- [Framebuffer.h](#)
- [Framebuffer.cpp](#)

4.19 Framebuffer1 Class Reference

Inheritance diagram for Framebuffer1:



Public Member Functions

- **Framebuffer1** (int16_t w, int16_t h, [DataStore](#) *b)
- void [fillScreen](#) (uint16_t c)
- void [setPixel](#) (int16_t x, int16_t y, uint16_t c)
- void **setColor** (uint8_t color, uint16_t rgb)
- void **setColor** (uint8_t color, uint8_t r, uint8_t g, uint8_t b)
- void **loadPalette** (const uint16_t *p)
- void **loadPalette** (const uint8_t p[256][3])
- void **loadPalette** ([Framebuffer](#) *fb)
- uint16_t [colorAt](#) (int16_t x, int16_t y)
- void **getScanLine** (uint16_t y, uint16_t x, uint16_t w, uint16_t *data)
- void [initializeDevice](#) ()

Public Attributes

- uint16_t **palette** [2]

Additional Inherited Members

4.19.1 Member Function Documentation

4.19.1.1 `uint16_t Framebuffer1::colorAt (int16_t x, int16_t y)` `[virtual]`

Get the colour at a location

Returns the colour at (x,y) as seen by the screen.

Example:

```
unsigned int color = tft.colorAt(100, 100);
```

Reimplemented from [Framebuffer](#).

4.19.1.2 `void Framebuffer1::fillScreen (uint16_t color)` `[virtual]`

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [Framebuffer](#).

4.19.1.3 `void Framebuffer1::initializeDevice ()` `[virtual]`

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Reimplemented from [Framebuffer](#).

4.19.1.4 `void Framebuffer1::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]`

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

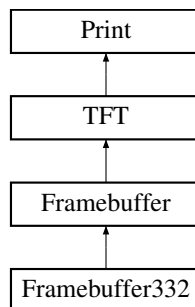
Reimplemented from [Framebuffer](#).

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

- `Framebuffer1.h`
- `Framebuffer1.cpp`

4.20 Framebuffer332 Class Reference

Inheritance diagram for Framebuffer332:



Public Member Functions

- **Framebuffer332** (int16_t w, int16_t h, [DataStore](#) *b)
- void [fillScreen](#) (uint16_t c)
- void [setPixel](#) (int16_t x, int16_t y, uint16_t c)
- void **setColor** (uint8_t color, uint16_t rgb)
- void **setColor** (uint8_t color, uint8_t r, uint8_t g, uint8_t b)
- void **loadPalette** (const uint16_t *p)
- void **loadPalette** (const uint8_t p[256][3])
- void **loadPalette** ([Framebuffer](#) *fb)
- uint16_t [colorAt](#) (int16_t x, int16_t y)
- uint16_t [bgColorAt](#) (int16_t x, int16_t y)
- void **getScanLine** (uint16_t y, uint16_t x, uint16_t w, uint16_t *data)
- void [initializeDevice](#) ()
- void **update** ([TFT](#) *)
- void **update** ([TFT](#) *, int16_t x0, int16_t y0)

Additional Inherited Members

4.20.1 Member Function Documentation

4.20.1.1 `uint16_t Framebuffer332::bgColorAt (int16_t x, int16_t y) [virtual]`

Get the raw colour at a location

Returns the base image colour at (x,y) before any further layers or post processing effects are performed.

Example:

```
unsigned int color = tft.bgColorAt(100, 100);
```

Reimplemented from [Framebuffer](#).

4.20.1.2 `uint16_t Framebuffer332::colorAt(int16_t x, int16_t y)` [virtual]

Get the colour at a location

Returns the colour at (x,y) as seen by the screen.

Example:

```
unsigned int color = tft.colorAt(100, 100);
```

Reimplemented from [Framebuffer](#).

4.20.1.3 `void Framebuffer332::fillScreen(uint16_t color)` [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [Framebuffer](#).

4.20.1.4 `void Framebuffer332::initializeDevice()` [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Reimplemented from [Framebuffer](#).

4.20.1.5 `void Framebuffer332::setPixel(int16_t x, int16_t y, uint16_t color)` [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

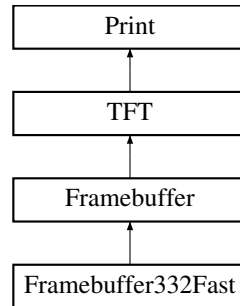
Reimplemented from [Framebuffer](#).

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

- [Framebuffer332.h](#)
- [Framebuffer332.cpp](#)

4.21 Framebuffer332Fast Class Reference

Inheritance diagram for Framebuffer332Fast:



Public Member Functions

- **Framebuffer332Fast** (int16_t w, int16_t h, uint8_t *b)
- void [fillScreen](#) (uint16_t c)
- void [setPixel](#) (int16_t x, int16_t y, uint16_t c)
- void **setColor** (uint8_t color, uint16_t rgb)
- void **setColor** (uint8_t color, uint8_t r, uint8_t g, uint8_t b)
- void **loadPalette** (const uint16_t *p)
- void **loadPalette** (const uint8_t p[256][3])
- void **loadPalette** ([Framebuffer](#) *fb)
- uint16_t [colorAt](#) (int16_t x, int16_t y)
- uint16_t [bgColorAt](#) (int16_t x, int16_t y)
- void **getScanLine** (uint16_t y, uint16_t x, uint16_t w, uint16_t *data)
- void [initializeDevice](#) ()
- void [drawHorizontalLine](#) (int16_t x, int16_t y, int16_t w, uint16_t color)
- void **update** ([TFT](#) *tft)

Additional Inherited Members

4.21.1 Member Function Documentation

4.21.1.1 uint16_t Framebuffer332Fast::bgColorAt (int16_t x, int16_t y) [virtual]

Get the raw colour at a location

Returns the base image colour at (x,y) before any further layers or post processing effects are performed.

Example:

```
unsigned int color = tft.bgColorAt(100, 100);
```

Reimplemented from [Framebuffer](#).

4.21.1.2 `uint16_t Framebuffer332Fast::colorAt (int16_t x, int16_t y)` [virtual]

Get the colour at a location

Returns the colour at (x,y) as seen by the screen.

Example:

```
unsigned int color = tft.colorAt(100, 100);
```

Reimplemented from [Framebuffer](#).

4.21.1.3 `void Framebuffer332Fast::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color)` [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Reimplemented from [Framebuffer](#).

4.21.1.4 `void Framebuffer332Fast::fillScreen (uint16_t color)` [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [Framebuffer](#).

4.21.1.5 `void Framebuffer332Fast::initializeDevice ()` [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Reimplemented from [Framebuffer](#).

4.21.1.6 `void Framebuffer332Fast::setPixel (int16_t x, int16_t y, uint16_t color)` [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

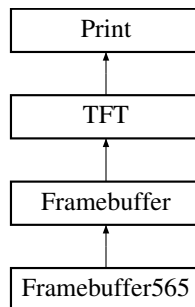
Reimplemented from [Framebuffer](#).

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

- Framebuffer332Fast.h
- Framebuffer332Fast.cpp

4.22 Framebuffer565 Class Reference

Inheritance diagram for Framebuffer565:



Public Member Functions

- **Framebuffer565** (int16_t w, int16_t h, [DataStore](#) *b)
- void [fillScreen](#) (uint16_t c)
- void [setPixel](#) (int16_t x, int16_t y, uint16_t c)
- void **setColor** (uint8_t color, uint16_t rgb)
- void **setColor** (uint8_t color, uint8_t r, uint8_t g, uint8_t b)
- void **loadPalette** (const uint16_t *p)
- void **loadPalette** (const uint8_t p[256][3])
- void **loadPalette** ([Framebuffer](#) *fb)
- uint16_t [colorAt](#) (int16_t x, int16_t y)
- uint16_t [bgColorAt](#) (int16_t x, int16_t y)
- void **getScanLine** (uint16_t y, uint16_t x, uint16_t w, uint16_t *data)
- void [initializeDevice](#) ()

Additional Inherited Members

4.22.1 Member Function Documentation

4.22.1.1 uint16_t Framebuffer565::bgColorAt (int16_t x, int16_t y) [virtual]

Get the raw colour at a location

Returns the base image colour at (x,y) before any further layers or post processing effects are performed.

Example:

```
unsigned int color = tft.bgColorAt(100, 100);
```

Reimplemented from [Framebuffer](#).

4.22.1.2 `uint16_t Framebuffer565::colorAt (int16_t x, int16_t y)` [virtual]

Get the colour at a location

Returns the colour at (x,y) as seen by the screen.

Example:

```
unsigned int color = tft.colorAt(100, 100);
```

Reimplemented from [Framebuffer](#).

4.22.1.3 `void Framebuffer565::fillScreen (uint16_t color)` [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [Framebuffer](#).

4.22.1.4 `void Framebuffer565::initializeDevice ()` [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Reimplemented from [Framebuffer](#).

4.22.1.5 `void Framebuffer565::setPixel (int16_t x, int16_t y, uint16_t color)` [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

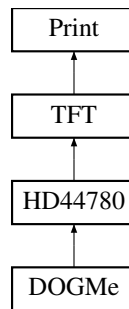
Reimplemented from [Framebuffer](#).

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

- `Framebuffer565.h`
- `Framebuffer565.cpp`

4.23 HD44780 Class Reference

Inheritance diagram for HD44780:



Public Member Functions

- **HD44780** ([TFTCommunicator](#) *comm, uint8_t w, uint8_t h)
- **HD44780** ([TFTCommunicator](#) &comm, uint8_t w, uint8_t h)
- void [fillScreen](#) (uint16_t color)
- void [setPixel](#) (int16_t x, int16_t y, uint16_t color)
- void [drawVerticalLine](#) (int16_t x, int16_t y, int16_t h, uint16_t color)
- void [drawHorizontalLine](#) (int16_t x, int16_t y, int16_t w, uint16_t color)
- void [fillRectangle](#) (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void [setRotation](#) (uint8_t r)
- void [invertDisplay](#) (boolean i)
- void [displayOn](#) ()
- void [displayOff](#) ()
- void [initializeDevice](#) ()
- void [clearScreen](#) ()
- void [home](#) ()
- void [write](#) (char c)
- void [command](#) (uint8_t cmd)
- void [data](#) (uint8_t d)

Public Attributes

- uint8_t [_bits](#)
- boolean [_cursor](#)
- boolean [_blink](#)

Additional Inherited Members

4.23.1 Member Function Documentation

4.23.1.1 void HD44780::displayOff () [virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.23.1.2 void HD44780::displayOn () [virtual]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.23.1.3 void HD44780::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color) [inline],[virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.23.1.4 void HD44780::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [inline],[virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.23.1.5 void HD44780::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [inline],[virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRectangle(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.23.1.6 void HD44780::fillScreen (uint16_t color) [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:


```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.23.1.7 void HD44780::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.23.1.8 void HD44780::invertDisplay (boolean i) [inline],[virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.23.1.9 void HD44780::setPixel (int16_t x, int16_t y, uint16_t color) [inline],[virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.23.1.10 void HD44780::setRotation (uint8_t rotation) [inline],[virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

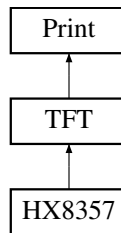
Implements [TFT](#).

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

- HD44780.h
- HD44780.cpp

4.24 HX8357 Class Reference

Inheritance diagram for HX8357:



Public Member Functions

- **HX8357** ([TFTCommunicator](#) *comms)
- **HX8357** ([TFTCommunicator](#) &comms)
- void **setAddrWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **fillScreen** (uint16_t color)
- void **setPixel** (int16_t x, int16_t y, uint16_t color)
- void **drawVerticalLine** (int16_t x, int16_t y, int16_t h, uint16_t color)
- void **drawHorizontalLine** (int16_t x, int16_t y, int16_t w, uint16_t color)
- void **fillRectangle** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void **setRotation** (uint8_t r)
- void **invertDisplay** (boolean i)
- void **displayOn** ()
- void **displayOff** ()
- void **initializeDevice** ()
- void **openWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **windowData** (uint16_t d)
- void **windowData** (uint16_t *d, uint32_t l)
- void **closeWindow** ()

Static Public Attributes

- static const uint16_t **Width** = 320
- static const uint16_t **Height** = 480

Protected Attributes

- uint8_t **colstart**
- uint8_t **rowstart**

Additional Inherited Members

4.24.1 Member Function Documentation

4.24.1.1 void HX8357::closeWindow () [virtual]

Close the window

Close the currently opened window and return to normal drawing operations.

Example:

```
tft.closeWindow();
```

Reimplemented from [TFT](#).

4.24.1.2 `void HX8357::displayOff ()` [virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.24.1.3 `void HX8357::displayOn ()` [virtual]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.24.1.4 `void HX8357::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color)` [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.24.1.5 `void HX8357::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color)` [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.24.1.6 `void HX8357::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)` [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRect(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.24.1.7 void HX8357::fillScreen (uint16_t *color*) [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.24.1.8 void HX8357::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.24.1.9 void HX8357::invertDisplay (boolean *i*) [virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.24.1.10 void HX8357::openWindow (uint16_t *x0*, uint16_t *y0*, uint16_t *x1*, uint16_t *y1*) [virtual]

Open a window

Opens the rectangle defined by (x0,y0) to (x1,y1) as a raw data window.

Example:

```
tft.openWindow(0, 0, 100, 100);
```

Reimplemented from [TFT](#).

4.24.1.11 `void HX8357::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]`

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.24.1.12 `void HX8357::setRotation (uint8_t rotation) [virtual]`

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

4.24.1.13 `void HX8357::windowData (uint16_t d) [virtual]`

Send pixel data to the window

Sends the raw pixel data for one pixel to the currently opened window.

Example:

```
tft.windowData(Color::Red);
```

Reimplemented from [TFT](#).

4.24.1.14 `void HX8357::windowData (uint16_t * d, uint32_t l) [virtual]`

Send a block of pixel data to the window

The array of pixel data (*d) and size (l) is dumped verbatim to the currently opened window.

Example:

```
tft.windowData(myData, 1000);
```

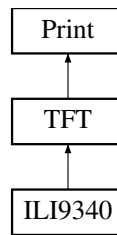
Reimplemented from [TFT](#).

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

- HX8357.h
- HX8357.cpp

4.25 ILI9340 Class Reference

Inheritance diagram for ILI9340:



Public Member Functions

- **ILI9340** ([TFTCommunicator](#) *comms)
- **ILI9340** ([TFTCommunicator](#) &comms)
- void **setAddrWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **fillScreen** (uint16_t color)
- void **setPixel** (int16_t x, int16_t y, uint16_t color)
- void **drawVerticalLine** (int16_t x, int16_t y, int16_t h, uint16_t color)
- void **drawHorizontalLine** (int16_t x, int16_t y, int16_t w, uint16_t color)
- void **fillRectangle** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void **setRotation** (uint8_t r)
- void **invertDisplay** (boolean i)
- void **displayOn** ()
- void **displayOff** ()
- void **initializeDevice** ()

Static Public Attributes

- static const uint16_t **Width** = 240
- static const uint16_t **Height** = 320

Additional Inherited Members

4.25.1 Member Function Documentation

4.25.1.1 void **ILI9340::displayOff** () [[inline](#)], [[virtual](#)]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.25.1.2 void **ILI9340::displayOn** () [[inline](#)], [[virtual](#)]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.25.1.3 void ILI9340::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color) [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.25.1.4 void ILI9340::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.25.1.5 void ILI9340::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRect(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.25.1.6 void ILI9340::fillScreen (uint16_t color) [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.25.1.7 void ILI9340::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.25.1.8 `void ILI9340::invertDisplay (boolean i) [virtual]`

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.25.1.9 `void ILI9340::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]`

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.25.1.10 `void ILI9340::setRotation (uint8_t rotation) [virtual]`

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

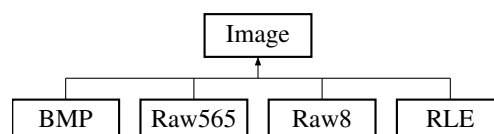
Implements [TFT](#).

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

- ILI9340.h
- ILI9340.cpp

4.26 Image Class Reference

Inheritance diagram for Image:



Public Member Functions

- virtual uint16_t **getWidth** ()
- virtual uint16_t **getHeight** ()
- virtual void **draw** (TFT *dev, int16_t x, int16_t y)=0
- virtual void **draw** (TFT *dev, int16_t x, int16_t y, uint16_t t)=0
- virtual void **drawTransformed** (TFT *dev, int16_t x, int16_t y, uint8_t transform)=0
- virtual void **drawTransformed** (TFT *dev, int16_t x, int16_t y, uint8_t transform, uint16_t t)=0

Public Attributes

- uint16_t **_width**
- uint16_t **_height**

Static Public Attributes

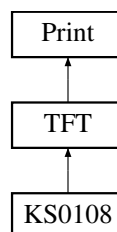
- static const uint8_t **MirrorH** = 0x01
- static const uint8_t **MirrorV** = 0x02
- static const uint8_t **Rotate180** = 0x03

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

- Image.h
- Image.cpp

4.27 KS0108 Class Reference

Inheritance diagram for KS0108:



Public Member Functions

- **KS0108** (TFTCommunicator *chip)
- **KS0108** (TFTCommunicator &chip)
- void **setAddrWindow** (uint8_t x0, uint8_t y0, uint8_t x1, uint8_t y1)
- void **fillScreen** (uint16_t color)
- void **doSetPixel** (int16_t x, int16_t y, uint16_t color)
- void **setPixel** (int16_t x, int16_t y, uint16_t color)
- void **drawVerticalLine** (int16_t x, int16_t y, int16_t h, uint16_t color)
- void **drawHorizontalLine** (int16_t x, int16_t y, int16_t w, uint16_t color)
- void **fillRectangle** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void **setRotation** (uint8_t r)
- void **invertDisplay** (boolean i)
- void **displayOn** ()
- void **displayOff** ()
- void **initializeDevice** ()
- void **updateScreen** ()

Protected Member Functions

- void **streamCommands** (uint8_t *cmdlist)
- void **setPage** (uint8_t page)
- void **setY** (uint8_t y)

Protected Attributes

- uint8_t **colstart**
- uint8_t **rowstart**
- uint8_t **_variant**
- uint8_t **buffer** [64 *8]

Additional Inherited Members

4.27.1 Member Function Documentation

4.27.1.1 void KS0108::displayOff() [inline],[virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.27.1.2 void KS0108::displayOn() [inline],[virtual]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.27.1.3 void KS0108::drawHorizontalLine(int16_t x, int16_t y, int16_t w, uint16_t color) [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.27.1.4 void KS0108::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.27.1.5 void KS0108::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRect(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.27.1.6 void KS0108::fillScreen (uint16_t color) [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.27.1.7 void KS0108::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.27.1.8 void KS0108::invertDisplay (boolean i) [virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.27.1.9 void KS0108::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.27.1.10 void KS0108::setRotation (uint8_t rotation) [inline],[virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

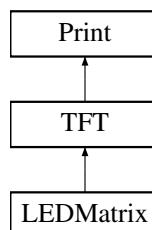
Implements [TFT](#).

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

- KS0108.h
- KS0108.cpp

4.28 LEDMatrix Class Reference

Inheritance diagram for LEDMatrix:



Public Member Functions

- **LEDMatrix** ([TFTCommunicator](#) *row, [TFTCommunicator](#) *col)
- **LEDMatrix** ([TFTCommunicator](#) &row, [TFTCommunicator](#) &col)
- void [fillScreen](#) (uint16_t color)
- void [setPixel](#) (int16_t x, int16_t y, uint16_t color)
- void [setRotation](#) (uint8_t r)
- void [invertDisplay](#) (boolean i)
- void [displayOn](#) ()

- void [displayOff](#) ()
- void [initializeDevice](#) ()
- void **UpdateISR** ()

Static Public Attributes

- static const int16_t **Width** = 32
- static const int16_t **Height** = 32

Protected Attributes

- [TFTCommunicator](#) * **_row**
- [TFTCommunicator](#) * **_col**
- uint32_t **buffer** [32]
- uint8_t **currentRow**

Additional Inherited Members

4.28.1 Member Function Documentation

4.28.1.1 void LEDMatrix::displayOff () [\[inline\]](#), [\[virtual\]](#)

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.28.1.2 void LEDMatrix::displayOn () [\[inline\]](#), [\[virtual\]](#)

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.28.1.3 void LEDMatrix::fillScreen (uint16_t *color*) [\[virtual\]](#)

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.28.1.4 void LEDMatrix::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.28.1.5 void LEDMatrix::invertDisplay (boolean i) [inline],[virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.28.1.6 void LEDMatrix::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.28.1.7 void LEDMatrix::setRotation (uint8_t rotation) [inline],[virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

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

- LEDMatrix.h
- LEDMatrix.cpp

4.29 MatrixISRList Struct Reference

Public Attributes

- [LEDMatrix](#) * matrix

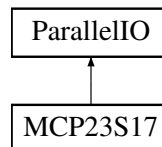
- struct [MatrixISRList](#) * **next**

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

- LEDMatrix.h

4.30 MCP23S17 Class Reference

Inheritance diagram for MCP23S17:



Public Member Functions

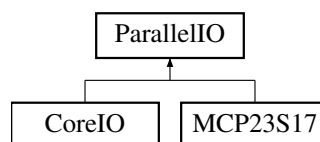
- **MCP23S17** (DSPI *spi, uint8_t cs, uint8_t addr)
- **MCP23S17** (DSPI &spi, uint8_t cs, uint8_t addr)
- void **pinMode** (uint16_t pin, uint8_t mode)
- void **digitalWrite** (uint16_t pin, uint8_t value)
- uint8_t **digitalRead** (uint16_t pin)
- uint16_t **pins** ()
- void **startBuffer** ()
- void **endBuffer** ()

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

- MCP23S17.h
- MCP23S17.cpp

4.31 ParallelIO Class Reference

Inheritance diagram for ParallelIO:



Public Member Functions

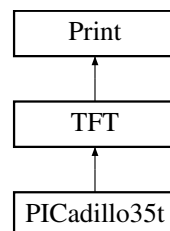
- virtual void **digitalWrite** (uint16_t pin, uint8_t value)=0
- virtual uint8_t **digitalRead** (uint16_t pin)=0
- virtual void **pinMode** (uint16_t pin, uint8_t mode)=0
- virtual uint16_t **pins** ()=0
- virtual void **startBuffer** ()=0
- virtual void **endBuffer** ()=0

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

- ParallelIO.h

4.32 PICadillo35t Class Reference

Inheritance diagram for PICadillo35t:



Public Member Functions

- void **loadCacheBlock** (int16_t x, int16_t y)
- void **flushCacheBlock** ()
- void **setAddrWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **setAddrWindowRead** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **fillScreen** (uint16_t color)
- void **setPixel** (int16_t x, int16_t y, uint16_t color)
- void **drawVerticalLine** (int16_t x, int16_t y, int16_t h, uint16_t color)
- void **drawHorizontalLine** (int16_t x, int16_t y, int16_t w, uint16_t color)
- void **fillRectangle** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void **setRotation** (uint8_t r)
- void **invertDisplay** (boolean i)
- void **displayOn** ()
- void **displayOff** ()
- uint16_t **colorAt** (int16_t x, int16_t y)
- void **initializeDevice** ()
- void **openWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **windowData** (uint16_t d)
- void **windowData** (uint16_t *d, uint32_t l)
- void **closeWindow** ()
- void **__attribute__** ((alwaysinline)) writeCommand(uint16_t)
- void **__attribute__** ((alwaysinline)) writeData(uint16_t)

Public Attributes

- uint16_t **_cacheData** [(1<< cacheDimension)*(1<< cacheDimension)]
- uint8_t **_cacheState**
- int16_t **_cacheX**
- int16_t **_cacheY**
- uint8_t **_lastOp**

Static Public Attributes

- static const uint8_t **opWrite** = 0
- static const uint8_t **opRead** = 1
- static const uint8_t **cacheInvalid** = 0
- static const uint8_t **cacheClean** = 1
- static const uint8_t **cacheDirty** = 2
- static const uint8_t **cacheDimension** = 4
- static const uint16_t **Width** = 320
- static const uint16_t **Height** = 480

Protected Attributes

- uint8_t **colstart**
- uint8_t **rowstart**

4.32.1 Member Function Documentation

4.32.1.1 void PICadillo35t::closeWindow () [virtual]

Close the window

Close the currently opened window and return to normal drawing operations.

Example:

```
tft.closeWindow();
```

Reimplemented from [TFT](#).

4.32.1.2 uint16_t PICadillo35t::colorAt (int16_t x, int16_t y) [virtual]

Get the colour at a location

Returns the colour at (x,y) as seen by the screen.

Example:

```
unsigned int color = tft.colorAt(100, 100);
```

Reimplemented from [TFT](#).

4.32.1.3 void PICadillo35t::displayOff () [virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.32.1.4 void PICadillo35t::displayOn () [virtual]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.32.1.5 void PICadillo35t::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color) [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.32.1.6 void PICadillo35t::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.32.1.7 void PICadillo35t::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRectangle(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.32.1.8 void PICadillo35t::fillScreen (uint16_t color) [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.32.1.9 void PICadillo35t::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.32.1.10 void PICadillo35t::invertDisplay (boolean i) [virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.32.1.11 void PICadillo35t::openWindow (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1) [virtual]

Open a window

Opens the rectangle defined by (x0,y0) to (x1,y1) as a raw data window.

Example:

```
tft.openWindow(0, 0, 100, 100);
```

Reimplemented from [TFT](#).

4.32.1.12 void PICadillo35t::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.32.1.13 void PICadillo35t::setRotation (uint8_t *rotation*) [virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

4.32.1.14 void PICadillo35t::windowData (uint16_t *d*) [virtual]

Send pixel data to the window

Sends the raw pixel data for one pixel to the currently opened window.

Example:

```
tft.windowData(Color::Red);
```

Reimplemented from [TFT](#).

4.32.1.15 void PICadillo35t::windowData (uint16_t* *d*, uint32_t *l*) [virtual]

Send a block of pixel data to the window

The array of pixel data (*d) and size (l) is dumped verbatim to the currently opened window.

Example:

```
tft.windowData(myData, 1000);
```

Reimplemented from [TFT](#).

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

- PICadillo35t.h
- PICadillo35t.cpp

4.33 point3d Struct Reference

Public Attributes

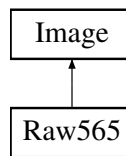
- float **x**
- float **y**
- float **z**

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

- TFT.h

4.34 Raw565 Class Reference

Inheritance diagram for Raw565:



Public Member Functions

- **Raw565** (const uint16_t *data, uint16_t w, uint16_t h)
- void **draw** (TFT *dev, int16_t x, int16_t y)
- void **draw** (TFT *dev, int16_t x, int16_t y, uint16_t t)
- void **drawTransformed** (TFT *dev, int16_t x, int16_t y, uint8_t transform)
- void **drawTransformed** (TFT *dev, int16_t x, int16_t y, uint8_t transform, uint16_t t)

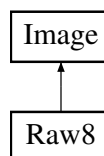
Additional Inherited Members

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

- Raw565.h
- Raw565.cpp

4.35 Raw8 Class Reference

Inheritance diagram for Raw8:



Public Member Functions

- **Raw8** (const uint8_t *data, uint16_t w, uint16_t h)
- void **draw** (TFT *dev, int16_t x, int16_t y)
- void **draw** (TFT *dev, int16_t x, int16_t y, uint16_t t)
- void **drawTransformed** (TFT *dev, int16_t x, int16_t y, uint8_t transform)
- void **drawTransformed** (TFT *dev, int16_t x, int16_t y, uint8_t transform, uint16_t t)

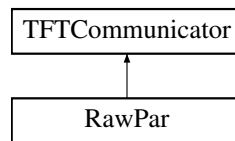
Additional Inherited Members

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

- Raw8.h
- Raw8.cpp

4.36 RawPar Class Reference

Inheritance diagram for RawPar:



Public Member Functions

- **RawPar** (uint8_t d0, uint8_t d1=255, uint8_t d2=255, uint8_t d3=255, uint8_t d4=255, uint8_t d5=255, uint8_t d6=255, uint8_t d7=255, uint8_t d8=255, uint8_t d9=255, uint8_t d10=255, uint8_t d11=255, uint8_t d12=255, uint8_t d13=255, uint8_t d14=255, uint8_t d15=255, uint8_t d16=255, uint8_t d17=255, uint8_t d18=255, uint8_t d19=255, uint8_t d20=255, uint8_t d21=255, uint8_t d22=255, uint8_t d23=255, uint8_t d24=255, uint8_t d25=255, uint8_t d26=255, uint8_t d27=255, uint8_t d28=255, uint8_t d29=255, uint8_t d30=255, uint8_t d31=255)
- void [writeCommand8](#) (uint8_t command)
- void [writeCommand16](#) (uint16_t command)
- void [writeCommand32](#) (uint32_t command)
- void [writeData8](#) (uint8_t data)
- void [writeData16](#) (uint16_t data)
- void [writeData32](#) (uint32_t data)
- void [streamStart](#) ()
- void [streamEnd](#) ()
- void [streamCommand8](#) (uint8_t)
- void [streamCommand16](#) (uint16_t)
- void [streamCommand32](#) (uint32_t)
- void [streamData8](#) (uint8_t)
- void [streamData16](#) (uint16_t)
- void [streamData32](#) (uint32_t)
- void [setBus](#) (uint32_t)
- uint8_t [nativeWidth](#) ()

4.36.1 Member Function Documentation

4.36.1.1 uint8_t RawPar::nativeWidth () [virtual]

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.36.1.2 void RawPar::streamCommand16 (uint16_t data) [virtual]

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.36.1.3 void RawPar::streamCommand32 (uint32_t data) [virtual]

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.36.1.4 `void RawPar::streamCommand8 (uint8_t data) [virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.36.1.5 `void RawPar::streamData16 (uint16_t data) [virtual]`

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.36.1.6 `void RawPar::streamData32 (uint32_t data) [virtual]`

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.36.1.7 `void RawPar::streamData8 (uint8_t data) [virtual]`

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.36.1.8 `void RawPar::streamEnd () [virtual]`

Close the currently open stream

Implements [TFTCommunicator](#).

4.36.1.9 `void RawPar::streamStart () [virtual]`

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.36.1.10 `void RawPar::writeCommand16 (uint16_t command) [virtual]`

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.36.1.11 `void RawPar::writeCommand32 (uint32_t command) [virtual]`

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.36.1.12 `void RawPar::writeCommand8 (uint8_t command) [virtual]`

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.36.1.13 `void RawPar::writeData16 (uint16_t data) [virtual]`

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.36.1.14 `void RawPar::writeData32 (uint32_t data) [virtual]`

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.36.1.15 `void RawPar::writeData8 (uint8_t data) [virtual]`

Write 8 bits of data to the device

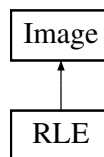
Implements [TFTCommunicator](#).

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

- RawPar.h
- RawPar.cpp

4.37 RLE Class Reference

Inheritance diagram for RLE:



Public Member Functions

- **RLE** (const uint8_t *data)
- void **draw** ([TFT](#) *dev, int16_t x, int16_t y)
- void **draw** ([TFT](#) *dev, int16_t x, int16_t y, uint16_t t)
- void **drawTransformed** ([TFT](#) *dev, int16_t x, int16_t y, uint8_t transform)
- void **drawTransformed** ([TFT](#) *dev, int16_t x, int16_t y, uint8_t transform, uint16_t t)

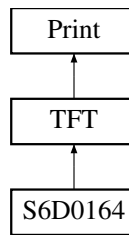
Additional Inherited Members

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

- RLE.h
- RLE.cpp

4.38 S6D0164 Class Reference

Inheritance diagram for S6D0164:



Public Member Functions

- **S6D0164** ([TFTCommunicator](#) *comms)
- **S6D0164** ([TFTCommunicator](#) &comms)
- void **setAddrWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **fillScreen** (uint16_t color)
- void **setPixel** (int16_t x, int16_t y, uint16_t color)
- void **drawVerticalLine** (int16_t x, int16_t y, int16_t h, uint16_t color)
- void **drawHorizontalLine** (int16_t x, int16_t y, int16_t w, uint16_t color)
- void **fillRectangle** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void **setRotation** (uint8_t r)
- void **invertDisplay** (boolean i)
- void **displayOn** ()
- void **displayOff** ()
- void **initializeDevice** ()
- void **openWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **windowData** (uint16_t d)
- void **windowData** (uint16_t *d, uint32_t l)
- void **closeWindow** ()

Static Public Attributes

- static const uint16_t **Width** = 176
- static const uint16_t **Height** = 220

Additional Inherited Members

4.38.1 Member Function Documentation

4.38.1.1 void S6D0164::closeWindow () [[virtual](#)]

Close the window

Close the currently opened window and return to normal drawing operations.

Example:

```
tft.closeWindow();
```

Reimplemented from [TFT](#).

4.38.1.2 void S6D0164::displayOff () [virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.38.1.3 void S6D0164::displayOn () [virtual]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.38.1.4 void S6D0164::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color) [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.38.1.5 void S6D0164::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.38.1.6 void S6D0164::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRectangle(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.38.1.7 `void S6D0164::fillScreen (uint16_t color) [virtual]`

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.38.1.8 `void S6D0164::initializeDevice () [virtual]`

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.38.1.9 `void S6D0164::invertDisplay (boolean i) [virtual]`

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.38.1.10 `void S6D0164::openWindow (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1) [virtual]`

Open a window

Opens the rectangle defined by (x0,y0) to (x1,y1) as a raw data window.

Example:

```
tft.openWindow(0, 0, 100, 100);
```

Reimplemented from [TFT](#).

4.38.1.11 `void S6D0164::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]`

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.38.1.12 void S6D0164::setRotation (uint8_t *rotation*) [virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

4.38.1.13 void S6D0164::windowData (uint16_t *d*) [virtual]

Send pixel data to the window

Sends the raw pixel data for one pixel to the currently opened window.

Example:

```
tft.windowData(Color::Red);
```

Reimplemented from [TFT](#).

4.38.1.14 void S6D0164::windowData (uint16_t* *d*, uint32_t *l*) [virtual]

Send a block of pixel data to the window

The array of pixel data (*d) and size (l) is dumped verbatim to the currently opened window.

Example:

```
tft.windowData(myData, 1000);
```

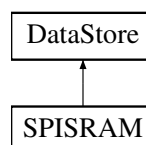
Reimplemented from [TFT](#).

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

- S6D0164.h
- S6D0164.cpp

4.39 SPIRAM Class Reference

Inheritance diagram for SPIRAM:



Public Member Functions

- **SPISRAM** (DSPI *spi, uint8_t cs, uint32_t s)
- **SPISRAM** (DSPI &spi, uint8_t cs, uint32_t s)
- uint8_t **read8** (uint32_t address)
- uint16_t **read16** (uint32_t address)
- uint32_t **read32** (uint32_t address)
- void **read8** (uint32_t address, uint8_t *data, uint32_t len)
- void **read16** (uint32_t address, uint16_t *data, uint32_t len)
- void **read32** (uint32_t address, uint32_t *data, uint32_t len)
- void **write8** (uint32_t address, uint8_t data)
- void **write16** (uint32_t address, uint16_t data)
- void **write32** (uint32_t address, uint32_t data)
- void **write8** (uint32_t address, uint8_t *data, uint32_t len)
- void **write16** (uint32_t address, uint16_t *data, uint32_t len)
- void **write32** (uint32_t address, uint32_t *data, uint32_t len)
- void **setAll8** (uint8_t data)
- void **setAll16** (uint16_t data)
- void **setAll32** (uint32_t data)
- void **initializeDevice** ()
- uint32_t **size** ()

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

- SPISRAM.h
- SPISRAM.cpp

4.40 sprite Struct Reference

```
#include <Framebuffer.h>
```

Public Attributes

- int16_t [xpos](#)
- int16_t [ypos](#)
- uint16_t [width](#)
- uint16_t [height](#)
- uint8_t [transparent](#)
- int8_t [frames](#)
- int8_t [currentframe](#)
- int8_t [animdir](#)
- int8_t [store](#) [8]
- const uint8_t * [data](#)
- struct [sprite](#) * [next](#)

4.40.1 Detailed Description

A sprite is a small 8-bit indexed image overlaid on the screen at rendering time

4.40.2 Member Data Documentation

4.40.2.1 `int8_t sprite::animdir`

Direction the animation is running

4.40.2.2 `int8_t sprite::currentframe`

Currently displayed frame number

4.40.2.3 `const uint8_t* sprite::data`

Pointer to graphical data for sprite

4.40.2.4 `int8_t sprite::frames`

Number of frames in the sprite

4.40.2.5 `uint16_t sprite::height`

Height of the sprite

4.40.2.6 `struct sprite* sprite::next`

Pointer to next sprite in the list

4.40.2.7 `int8_t sprite::store[8]`

Internal data store for sprite specific information

4.40.2.8 `uint8_t sprite::transparent`

Transparent colour index

4.40.2.9 `uint16_t sprite::width`

Width of the sprite

4.40.2.10 `int16_t sprite::xpos`

X Position of the sprite

4.40.2.11 `int16_t sprite::ypos`

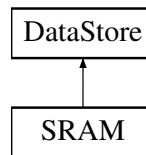
Y Position of the sprite

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

- Framebuffer.h

4.41 SRAM Class Reference

Inheritance diagram for SRAM:



Public Member Functions

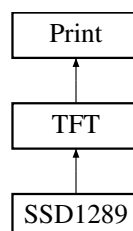
- **SRAM** (uint8_t *buf, uint32_t s)
- uint8_t **read8** (uint32_t address)
- uint16_t **read16** (uint32_t address)
- uint32_t **read32** (uint32_t address)
- void **read8** (uint32_t address, uint8_t *data, uint32_t len)
- void **read16** (uint32_t address, uint16_t *data, uint32_t len)
- void **read32** (uint32_t address, uint32_t *data, uint32_t len)
- void **write8** (uint32_t address, uint8_t data)
- void **write16** (uint32_t address, uint16_t data)
- void **write32** (uint32_t address, uint32_t data)
- void **write8** (uint32_t address, uint8_t *data, uint32_t len)
- void **write16** (uint32_t address, uint16_t *data, uint32_t len)
- void **write32** (uint32_t address, uint32_t *data, uint32_t len)
- void **setAll8** (uint8_t data)
- void **setAll16** (uint16_t data)
- void **setAll32** (uint32_t data)
- void **initializeDevice** ()
- uint32_t **size** ()

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

- SRAM.h
- SRAM.cpp

4.42 SSD1289 Class Reference

Inheritance diagram for SSD1289:



Public Member Functions

- **SSD1289** ([TFTCommunicator](#) *comms)
- **SSD1289** ([TFTCommunicator](#) &comms)
- void **setAddrWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- void **fillScreen** (uint16_t color)
- void **setPixel** (int16_t x, int16_t y, uint16_t color)
- void **drawVerticalLine** (int16_t x, int16_t y, int16_t h, uint16_t color)
- void **drawHorizontalLine** (int16_t x, int16_t y, int16_t w, uint16_t color)
- void **fillRectangle** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void **setRotation** (uint8_t r)
- void **invertDisplay** (boolean i)
- void **displayOn** ()
- void **displayOff** ()
- void **initializeDevice** ()
- virtual void **openWindow** (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- virtual void **windowData** (uint16_t d)
- virtual void **windowData** (uint16_t *d, uint32_t l)
- virtual void **closeWindow** ()

Static Public Attributes

- static const uint16_t **Width** = 240
- static const uint16_t **Height** = 320

Additional Inherited Members

4.42.1 Member Function Documentation

4.42.1.1 void SSD1289::closeWindow () [virtual]

Close the window

Close the currently opened window and return to normal drawing operations.

Example:

```
tft.closeWindow();
```

Reimplemented from [TFT](#).

4.42.1.2 void SSD1289::displayOff () [inline],[virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.42.1.3 `void SSD1289::displayOn () [inline],[virtual]`

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.42.1.4 `void SSD1289::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color) [virtual]`

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.42.1.5 `void SSD1289::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [virtual]`

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.42.1.6 `void SSD1289::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [virtual]`

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRect(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.42.1.7 `void SSD1289::fillScreen (uint16_t color) [virtual]`

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.42.1.8 void SSD1289::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.42.1.9 void SSD1289::invertDisplay (boolean i) [virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.42.1.10 void SSD1289::openWindow (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1) [virtual]

Open a window

Opens the rectangle defined by (x0,y0) to (x1,y1) as a raw data window.

Example:

```
tft.openWindow(0, 0, 100, 100);
```

Reimplemented from [TFT](#).

4.42.1.11 void SSD1289::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.42.1.12 `void SSD1289::setRotation (uint8_t rotation) [virtual]`

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

4.42.1.13 `void SSD1289::windowData (uint16_t d) [virtual]`

Send pixel data to the window

Sends the raw pixel data for one pixel to the currently opened window.

Example:

```
tft.windowData(Color::Red);
```

Reimplemented from [TFT](#).

4.42.1.14 `void SSD1289::windowData (uint16_t* d, uint32_t l) [virtual]`

Send a block of pixel data to the window

The array of pixel data (*d) and size (l) is dumped verbatim to the currently opened window.

Example:

```
tft.windowData(myData, 1000);
```

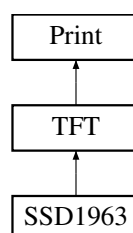
Reimplemented from [TFT](#).

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

- SSD1289.h
- SSD1289.cpp

4.43 SSD1963 Class Reference

Inheritance diagram for SSD1963:



Public Member Functions

- **SSD1963** ([TFTCommunicator](#) *comms)
- **SSD1963** ([TFTCommunicator](#) &comms)
- void [fillScreen](#) (uint16_t color)
- void [setPixel](#) (int16_t x, int16_t y, uint16_t color)
- void [drawVerticalLine](#) (int16_t x, int16_t y, int16_t h, uint16_t color)
- void [drawHorizontalLine](#) (int16_t x, int16_t y, int16_t w, uint16_t color)
- void [fillRectangle](#) (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- void [setRotation](#) (uint8_t r)
- void [invertDisplay](#) (boolean i)
- void [displayOn](#) ()
- void [displayOff](#) ()
- void [initializeDevice](#) ()

Static Public Attributes

- static const uint16_t [Width](#) = 800
- static const uint16_t [Height](#) = 480

Additional Inherited Members

4.43.1 Member Function Documentation

4.43.1.1 void [SSD1963::displayOff](#) () [[inline](#)], [[virtual](#)]

Not currently implemented

Implements [TFT](#).

4.43.1.2 void [SSD1963::displayOn](#) () [[inline](#)], [[virtual](#)]

Not currently implemented

Implements [TFT](#).

4.43.1.3 void [SSD1963::drawHorizontalLine](#) (int16_t x, int16_t y, int16_t w, uint16_t *color*) [[virtual](#)]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.43.1.4 void [SSD1963::drawVerticalLine](#) (int16_t x, int16_t y, int16_t h, uint16_t *color*) [[virtual](#)]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.43.1.5 `void SSD1963::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)` [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRect(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.43.1.6 `void SSD1963::fillScreen (uint16_t color)` [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.43.1.7 `void SSD1963::initializeDevice ()` [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.43.1.8 `void SSD1963::invertDisplay (boolean i)` [virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.43.1.9 `void SSD1963::setPixel (int16_t x, int16_t y, uint16_t color)` [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.43.1.10 `void SSD1963::setRotation (uint8_t rotation)` [virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

4.43.2 Member Data Documentation

4.43.2.1 `const uint16_t SSD1963::Height = 480` [static]

The height of the screen is 480 pixels

4.43.2.2 `const uint16_t SSD1963::Width = 800` [static]

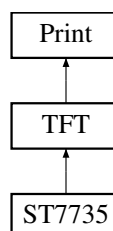
The width of the screen is 800 pixels

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

- SSD1963.h
- SSD1963.cpp

4.44 ST7735 Class Reference

Inheritance diagram for ST7735:



Public Member Functions

- void [fillScreen](#) (uint16_t color)
 - void [setPixel](#) (int16_t x, int16_t y, uint16_t color)
 - void [drawVerticalLine](#) (int16_t x, int16_t y, int16_t h, uint16_t color)
 - void [drawHorizontalLine](#) (int16_t x, int16_t y, int16_t w, uint16_t color)
 - void [fillRectangle](#) (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
 - void [setRotation](#) (uint8_t r)
 - void [invertDisplay](#) (boolean i)
 - void [displayOn](#) ()
 - void [displayOff](#) ()
 - void [initializeDevice](#) ()
-
- [ST7735](#) ([TFTCommunicator](#) *comms, uint8_t variant)
 - [ST7735](#) ([TFTCommunicator](#) &comms, uint8_t variant)

Static Public Attributes

- static const uint8_t [GreenTab](#) = 0x00
- static const uint8_t [RedTab](#) = 0x01
- static const uint8_t [BlackTab](#) = 0x02
- static const uint8_t [TypeB](#) = 0x03
- static const uint8_t [Width](#) = 128
- static const uint8_t [Height](#) = 160

Additional Inherited Members

4.44.1 Constructor & Destructor Documentation

4.44.1.1 ST7735::ST7735 ([TFTCommunicator](#) * *comms*, uint8_t *variant*) [inline]

The constructor takes an SPI compatible communicator class. Also, as there are multiple screens available with the same chip, each working slightly differently, a "variant" value must be provided:

- ST7736::GreenTab
- ST7736::RedTab
- ST7736::BlackTab
- ST7736::TypeB

4.44.2 Member Function Documentation

4.44.2.1 void ST7735::displayOff () [inline], [virtual]

Turn off the display

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implements [TFT](#).

4.44.2.2 void ST7735::displayOn () [inline],[virtual]

Turn on the display

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implements [TFT](#).

4.44.2.3 void ST7735::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color) [virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.44.2.4 void ST7735::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color) [virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implements [TFT](#).

4.44.2.5 void ST7735::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRectangle(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented from [TFT](#).

4.44.2.6 void ST7735::fillScreen (uint16_t color) [virtual]

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:


```
tft.fillScreen(Color::Black);
```

Reimplemented from [TFT](#).

4.44.2.7 void ST7735::initializeDevice () [virtual]

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implements [TFT](#).

4.44.2.8 void ST7735::invertDisplay (boolean i) [virtual]

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implements [TFT](#).

4.44.2.9 void ST7735::setPixel (int16_t x, int16_t y, uint16_t color) [virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implements [TFT](#).

4.44.2.10 void ST7735::setRotation (uint8_t rotation) [virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implements [TFT](#).

4.44.3 Member Data Documentation

4.44.3.1 const uint8_t ST7735::BlackTab = 0x02 [static]

Adafruit screen with a black tab

4.44.3.2 `const uint8_t ST7735::GreenTab = 0x00` `[static]`

Adafruit screen with a green tab

4.44.3.3 `const uint8_t ST7735::Height = 160` `[static]`

The native size of the screen is 160 pixels high

4.44.3.4 `const uint8_t ST7735::RedTab = 0x01` `[static]`

Adafruit screen with a red tab

4.44.3.5 `const uint8_t ST7735::TypeB = 0x03` `[static]`

Adafruit "Type B" screen

4.44.3.6 `const uint8_t ST7735::Width = 128` `[static]`

The native size of the screen is 128 pixels wide

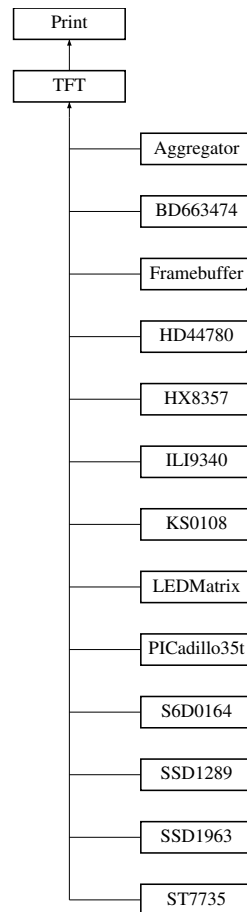
The documentation for this class was generated from the following files:

- ST7735.h
- ST7735.cpp

4.45 TFT Class Reference

```
#include <TFT.h>
```

Inheritance diagram for TFT:



Public Member Functions

- [TFT](#) ()
- [TFT](#) ([TFTCommunicator](#) *comms)
- [TFT](#) ([TFTCommunicator](#) &comms)
- virtual uint16_t [getWidth](#) ()
- virtual uint16_t [getHeight](#) ()

Drawing Functions

These functions draw pretty shapes on the screen.

- virtual void [drawCircle](#) (int16_t x0, int16_t y0, int16_t r, uint16_t color)
- virtual void [fillCircle](#) (int16_t x0, int16_t y0, int16_t r, uint16_t color)
- virtual void [drawLine](#) (int16_t x0, int16_t y0, int16_t x1, int16_t y1, uint16_t color)
- virtual void [drawRectangle](#) (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- virtual void [drawRoundRect](#) (int16_t x, int16_t y, int16_t w, int16_t h, int16_t r, uint16_t color)
- virtual void [fillRoundRect](#) (int16_t x, int16_t y, int16_t w, int16_t h, int16_t r, uint16_t color)
- virtual void [drawTriangle](#) (int16_t x0, int16_t y0, int16_t x1, int16_t y1, int16_t x2, int16_t y2, uint16_t color)
- virtual void [fillTriangle](#) (int16_t x0, int16_t y0, int16_t x1, int16_t y1, int16_t x2, int16_t y2, uint16_t color)
- virtual void [fillScreen](#) (uint16_t color)
- virtual void [fillRectangle](#) (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)

Image drawing

These routines are used for drawing basic bitmap images to the screen.

- virtual void [drawBitmap](#) (int16_t x, int16_t y, const uint8_t *bitmap, int16_t w, int16_t h, uint16_t color)
- virtual void [drawRGB](#) (int16_t x, int16_t y, const uint16_t *bitmap, int16_t w, int16_t h)

- virtual void [drawRGBA](#) (int16_t x, int16_t y, const uint16_t *bitmap, int16_t w, int16_t h, uint16_t trans)

Text handing functions

These are functions used for dealing with text and printing of strings to the screen.

- virtual void [setCursor](#) (int16_t x, int16_t y)
- virtual int16_t [getCursorX](#) ()
- virtual int16_t [getCursorY](#) ()
- virtual int16_t [getCursor](#) (boolean x)
- virtual void [setTextColor](#) (uint16_t c)
- virtual void [setTextColor](#) (uint16_t fg, uint16_t bg)
- virtual uint16_t [getTextColor](#) ()
- virtual void [invertTextColor](#) ()
- virtual void [setTextWrap](#) (boolean w)
- virtual void [setFont](#) (const uint8_t *f)
- virtual uint16_t [stringWidth](#) (char *text)
- virtual uint16_t [stringHeight](#) (char *text)
- void [write](#) (uint8_t c)
- uint8_t [drawChar](#) (int16_t x, int16_t y, unsigned char c, uint16_t color, uint16_t bg)
- void [setFontScaleX](#) (uint8_t sx)
- void [setFontScaleY](#) (uint8_t sy)

Colour handling

These functions are all related to manipulating colours in one way or another.

- virtual uint16_t [color565](#) (uint8_t r, uint8_t g, uint8_t b)
- virtual uint16_t [bgColorAt](#) (int16_t x, int16_t y)
- virtual uint16_t [colorAt](#) (int16_t x, int16_t y)
- [point3d rgb2xyz](#) (uint16_t c)
- [point3d xyz2lab](#) (point3d c)
- float [deltaE](#) (point3d labA, point3d labB)
- uint32_t [deltaOrth](#) (uint16_t c1, uint16_t c2)
- uint32_t [rgb2hsv](#) (uint16_t rgb)
- uint16_t [mix](#) (uint16_t a, uint16_t b, uint8_t pct)

Pure virtual functions

These are all functions that must be implemented in a [TFT](#) driver in order for it to function.

- virtual void [setRotation](#) (uint8_t rotation)=0
- virtual void [setPixel](#) (int16_t x, int16_t y, uint16_t color)=0
- virtual void [drawHorizontalLine](#) (int16_t x, int16_t y, int16_t w, uint16_t color)=0
- virtual void [drawVerticalLine](#) (int16_t x, int16_t y, int16_t h, uint16_t color)=0
- virtual void [initializeDevice](#) ()=0
- virtual void [displayOn](#) ()=0
- virtual void [displayOff](#) ()=0
- virtual void [invertDisplay](#) (boolean i)=0

Window operations

The window system is what makes some of the fastest operations available.

- virtual void [openWindow](#) (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
- virtual void [windowData](#) (uint16_t d)
- virtual void [windowData](#) (uint16_t *d, uint32_t l)
- virtual void [closeWindow](#) ()

Helper Functions

These are functions used by other functions to do their work. They may be useful in other situations as well, but they won't be as fully documented.

- void [drawCircleHelper](#) (int16_t x0, int16_t y0, int16_t r, uint8_t cornername, uint16_t color)
- void [fillCircleHelper](#) (int16_t x0, int16_t y0, int16_t r, uint8_t cornername, int16_t delta, uint16_t color)
- boolean [clipToScreen](#) (int16_t &x, int16_t &y, int16_t &w, int16_t &h)
- void [fatalError](#) (const char *title, const char *message)

Public Attributes

- [TFTCommunicator](#) * [_comm](#)
- [int16_t](#) [cursor_x](#)
- [int16_t](#) [cursor_y](#)
- [boolean](#) [wrap](#)
- [uint16_t](#) [textcolor](#)
- [uint16_t](#) [textbgcolor](#)
- [uint16_t](#) [_width](#)
- [uint16_t](#) [_height](#)
- [uint8_t](#) [rotation](#)

Protected Attributes

- [const uint8_t](#) * [font](#)
- [uint8_t](#) [font_scale_x](#)
- [uint8_t](#) [font_scale_y](#)

4.45.1 Detailed Description

The [TFT](#) class describes and controls all the [TFT](#) screens. It acts as a polymorphic parent class for the other screen drivers, and also contains the generic primitive drawing routines.

It is expected that a [TFT](#) screen driver will override some functions from this class (some are pure virtual and must be overridden).

4.45.2 Constructor & Destructor Documentation

4.45.2.1 [TFT::TFT \(\)](#)

The default constructor takes no parameters. It creates a blank [TFT](#) screen class with no communication abilities.

4.45.2.2 [TFT::TFT \(\[TFTCommunicator\]\(#\) * *comm* \)](#)

In general, when constructing a [TFT](#) screen, you just need to pass the communication object to it. This can be as a pointer, or as a reference.

4.45.2.3 [TFT::TFT \(\[TFTCommunicator\]\(#\) & *comm* \)](#)

In general, when constructing a [TFT](#) screen, you just need to pass the communication object to it. This can be as a pointer, or as a reference.

4.45.3 Member Function Documentation

4.45.3.1 [uint16_t](#) [TFT::bgColorAt \(\[int16_t\]\(#\) *x*, \[int16_t\]\(#\) *y* \)](#) [[virtual](#)]

Get the raw colour at a location

Returns the base image colour at (x,y) before any further layers or post processing effects are performed.

Example:

```
unsigned int color = tft.bgColorAt(100, 100);
```

Reimplemented in [Framebuffer](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.45.3.2 void TFT::closeWindow () [virtual]

Close the window

Close the currently opened window and return to normal drawing operations.

Example:

```
tft.closeWindow();
```

Reimplemented in [PICadillo35t](#), [S6D0164](#), [HX8357](#), and [SSD1289](#).

4.45.3.3 uint16_t TFT::color565 (uint8_t r, uint8_t g, uint8_t b) [virtual]

Convert RGB to 565 colour

This function takes an RGB triplet (r, g, b) and converts it into a 16-bit 565 colour.

Example:

```
unsigned int yellow = tft.color565(255, 255, 0);
```

4.45.3.4 uint16_t TFT::colorAt (int16_t x, int16_t y) [virtual]

Get the colour at a location

Returns the colour at (x,y) as seen by the screen.

Example:

```
unsigned int color = tft.colorAt(100, 100);
```

Reimplemented in [Framebuffer](#), [PICadillo35t](#), [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.45.3.5 float TFT::deltaE (point3d labA, point3d labB)

Calculate the DeltaE between two LAB colours

This function takes two LAB colours and calculates the difference (delta) between them.

Example:

```
float delta = tft.deltaE(colorA, colorB);
```

4.45.3.6 uint32_t TFT::deltaOrth (uint16_t c1, uint16_t c2)

Calculate the orthogonal difference between colours

Two RGB 565 colours are compared and the orthogonal distance between them (as HSV colours) is calculated.

Example:

```
unsigned long delta = tft.deltaOrth(Color::Yellow, Color::Orange);
```

4.45.3.7 `virtual void TFT::displayOff ()` [pure virtual]**Turn off the display**

Disable the video output of the display (if supported).

Example:

```
tft.displayOff();
```

Implemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), and [LEDMatrix](#).

4.45.3.8 `virtual void TFT::displayOn ()` [pure virtual]**Turn on the display**

Enable the video output of the display (if supported).

Example:

```
tft.displayOn();
```

Implemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), and [LEDMatrix](#).

4.45.3.9 `void TFT::drawBitmap (int16_t x, int16_t y, const uint8_t* bitmap, int16_t w, int16_t h, uint16_t color)` [virtual]**Draw a 1-bit bitmap image**

A 1-bit bitmap image is a byte array where each byte represents 8 contiguous pixels. The image is rendered to the screen as naturally transparent, with set bits rendered in (color) and unset bits skipped. The image is rendered with the upper left corner at (x,y) and the image is (w,h) in size.

Example:

```
const byte letterA[] = {
    0b00000000,
    0b00111100,
    0b01000010,
    0b01000010,
    0b01111110,
    0b01000010,
    0b01000010,
    0b01000010,
    0b00000000};
tft.drawBitmap(100, 100, letterA, 8, 8, Color::Red);
```

4.45.3.10 `uint8_t TFT::drawChar (int16_t x, int16_t y, unsigned char c, uint16_t color, uint16_t bg)`**Draw a character**

This is the heart of the text handling. It takes the current font, locates the right character (c) data, and renders it to the screen at the specified (x,y) location. It is drawn in colour (color), and the background is filled in (bg). If (bg) and (color) are equal then the background pixels are skipped.

Example:

```
tft.drawChar(30, 30, 'Q', Color::Red, Color::Blue);
```

4.45.3.11 `void TFT::drawCircle (int16_t x0, int16_t y0, int16_t r, uint16_t color)` [virtual]

Draw a circle

This function draws the outline of a circle. Its center is at (x0, y0), it has radius (r) and is drawn in colour (color).

Example:

```
tft.drawCircle(50, 50, 20, Color::Red);
```

4.45.3.12 `void TFT::drawCircleHelper (int16_t x0, int16_t y0, int16_t r, uint8_t cornername, uint16_t color)`

This is a helper function. It is used to draw portions of a circle.

4.45.3.13 `virtual void TFT::drawHorizontalLine (int16_t x, int16_t y, int16_t w, uint16_t color)` [pure virtual]

Draw a horizontal line

A horizontal line of width (w) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawHorizontalLine(10, 10, 50, Color::Blue);
```

Implemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [Aggregator](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Framebuffer332Fast](#), [KS0108](#), [BD663474](#), [HX8357](#), and [SSD1289](#).

4.45.3.14 `void TFT::drawLine (int16_t x0, int16_t y0, int16_t x1, int16_t y1, uint16_t color)` [virtual]

Draw a straight line

This function uses Bresenham's algorithm to draw a straight line. The line starts at coordinates (x0, y0) and extends to coordinates (x1, y1). The line is drawn in color (color).

Example:

```
tft.drawLine(10, 10, 40, 60, Color::Green);
```

4.45.3.15 `void TFT::drawRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)` [virtual]

Draw a rectangle

This function uses accelerated line drawing routines if available. It draws a rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.drawRectangle(10, 10, 200, 300, Color::Blue);
```

4.45.3.16 `void TFT::drawRGB (int16_t x, int16_t y, const uint16_t* bitmap, int16_t w, int16_t h)` [virtual]

Draw an RGB (565) image

A 565 raw RGB image is rendered to the screen at (x,y). The image data is stored as an array of 16-bit values, and is (w,h) pixels in size.

Example:

```
tft.drawRGB(10, 30, myImage, 16, 16);
```

4.45.3.17 `void TFT::drawRGBA (int16_t x, int16_t y, const uint16_t * bitmap, int16_t w, int16_t h, uint16_t trans)`
[virtual]

Draw a transparent RGB (565) image

A 565 raw RGB image is rendered to the screen at (x,y). The image data is stored as an array of 16-bit values, and is (w,h) pixels in size. Any pixels with colour (trans) are skipped.

Example:

```
tft.drawRGBA(10, 30, myImage, 16, 16, Color::Black);
```

4.45.3.18 `void TFT::drawRoundRect (int16_t x, int16_t y, int16_t w, int16_t h, int16_t r, uint16_t color)` [virtual]

Draw a rounded rectangle

A rounded rectangle is a normal rectangle but with the corners rounded off. It is drawn with the upper-left corner at (x,y) and a width of (w) and height of (h). The corners are rounded off at a radius of (r) pixels, and it is drawn in colour (color).

Example:

```
tft.drawRoundRect(10, 10, 100, 50, 4, Color::Yellow);
```

4.45.3.19 `void TFT::drawTriangle (int16_t x0, int16_t y0, int16_t x1, int16_t y1, int16_t x2, int16_t y2, uint16_t color)`
[virtual]

Draw a triangle

A simple three lines joined together to form a triangle. The three points of the triangle are defined as (x0, y0), (x1, y1) and (x2, y2). It is drawn in colour (color).

Example:

```
tft.drawTriangle(40, 10, 60, 30, 20, 30, Color::Cyan);
```

4.45.3.20 `virtual void TFT::drawVerticalLine (int16_t x, int16_t y, int16_t h, uint16_t color)` [pure virtual]

Draw a vertical line

A vertical line of height (h) is drawn from point (x,y) in colour (color);

Example:

```
tft.drawVerticalLine(10, 10, 50, Color::Blue);
```

Implemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [Aggregator](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [KS0108](#), [BD663474](#), [HX8357](#), and [SSD1289](#).

4.45.3.21 `void TFT::fatalError (const char * title, const char * message)`

Display a fatal error

Used internally by various functions and libraries to display a fatal error message. This is an error that cannot be recovered from, so the program stops here.

Example:

```
tft.fatalError("MEMORY ERROR", "Unable to allocate space for objects");
```

4.45.3.22 void TFT::fillCircle (int16_t x0, int16_t y0, int16_t radius, uint16_t color) [virtual]

Draw a filled circle

This function draws a filled circle. It is highly optimised to get the maximum possible speed out of it.

Like the drawCircle function it centers the circle at (x0, y0), has radius (radius) and is drawn in (color).

Example:

```
tft.fillCircle(50, 50, 20, Color::Red);
```

4.45.3.23 void TFT::fillCircleHelper (int16_t x0, int16_t y0, int16_t r, uint8_t cornername, int16_t delta, uint16_t color)

This is a helper function. It is used to draw segments of a filled circle.

4.45.3.24 void TFT::fillRectangle (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color) [virtual]

Draw a rectangle

This function draws a filled rectangle on the screen. The upper-left corner of the rectangle is at (x, y), and it extends to the right and down for a distance of (w) and (h) pixels respectively. It is drawn in colour (color).

Example:

```
tft.fillRectangle(10, 10, 200, 300, Color::Blue);
```

It is expected that actual screen drivers will override this function with a high speed optimized function.

Reimplemented in [SSD1963](#), [ST7735](#), [PICadillo35t](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [KS0108](#), [BD663474](#), [HX8357](#), and [SSD1289](#).

4.45.3.25 void TFT::fillRoundRect (int16_t x, int16_t y, int16_t w, int16_t h, int16_t r, uint16_t color) [virtual]

Draw a filled rounded rectangle

A rounded rectangle is a normal rectangle but with the corners rounded off. It is drawn with the upper-left corner at (x,y) and a width of (w) and height of (h). The corners are rounded off at a radius of (r) pixels, and it is drawn (and filled) in colour (color).

Example:

```
tft.fillRoundRect(10, 10, 100, 50, 4, Color::Yellow);
```

4.45.3.26 `void TFT::fillScreen (uint16_t color) [virtual]`

Fill the screen with a colour

This function fills the entire screen with a solid colour.

Example:

```
tft.fillScreen(Color::Black);
```

Reimplemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [HD44780](#), [Aggregator](#), [S6D0164](#), [ILI9340](#), [K-S0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), [LEDMatrix](#), [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.45.3.27 `void TFT::fillTriangle (int16_t x0, int16_t y0, int16_t x1, int16_t y1, int16_t x2, int16_t y2, uint16_t color) [virtual]`

Draw a filled triangle

A simple three lines joined together to form a triangle. The three points of the triangle are defined as (x0, y0), (x1, y1) and (x2, y2). It is drawn in colour (color).

Example:

```
tft.fillTriangle(40, 10, 60, 30, 20, 30, Color::Cyan);
```

4.45.3.28 `int16_t TFT::getCursor (boolean x) [virtual]`

Get Text Cursor

Returns the either the current X or Y position of the text cursor. A parameter of `true` requests the X coordinate, otherwise the Y coordinate is returned.

Example:

```
int x = tft.getCursor(true);  
int y = tft.getCursor(false);
```

4.45.3.29 `int16_t TFT::getCursorX () [virtual]`

Get X Cursor

Returns the current X position of the text cursor.

Example:

```
int x = tft.getCursorX();
```

4.45.3.30 `int16_t TFT::getCursorY () [virtual]`

Get Y Cursor

Returns the current Y position of the text cursor.

Example:

```
int y = tft.getCursorY();
```

4.45.3.31 `virtual uint16_t TFT::getHeight () [inline],[virtual]`

Get screen height

Returns the height (in pixels) of the screen.

Example:

```
int height = tft.getHeight();
```

Reimplemented in [Framebuffer](#), and [Aggregator](#).

4.45.3.32 `uint16_t TFT::getTextColor () [virtual]`

Get the current foreground colour

Returns the currently selected foreground colour.

Example:

```
unsigned int color = tft.getTextColor();
```

4.45.3.33 `virtual uint16_t TFT::getWidth () [inline],[virtual]`

Get screen width

Returns the width (in pixels) of the screen.

Example:

```
int width = tft.getWidth();
```

Reimplemented in [Framebuffer](#), and [Aggregator](#).

4.45.3.34 `virtual void TFT::initializeDevice () [pure virtual]`

Initialize the display

The display is configured and made ready to work. This function *must* be called before anything can happen on the screen, and it *should* be called before any other function.

Example:

```
tft.initializeDevice();
```

Implemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), [Framebuffer565](#), [LEDMatrix](#), and [DOGM](#).

4.45.3.35 `virtual void TFT::invertDisplay (boolean i) [pure virtual]`

Invert the display colours

All colours become reversed. Black becomes white, red becomes cyan, etc.

Example:

```
tft.invertDisplay(true);
```

Implemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), and [LEDMatrix](#).

4.45.3.36 void TFT::invertTextColor () [virtual]

Invert the text colours

The foreground becomes the background, and the background becomes the foreground.

Example:

```
tft.invertTextColor();
```

4.45.3.37 uint16_t TFT::mix (uint16_t a, uint16_t b, uint8_t pct)

Mix two colours together

Returns a new colour that is the mixing of the two provided colours.

Example:

```
unsigned int yellow = tft.mix(Color::Red, Color::Green);
```

4.45.3.38 void TFT::openWindow (uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1) [virtual]

Open a window

Opens the rectangle defined by (x0,y0) to (x1,y1) as a raw data window.

Example:

```
tft.openWindow(0, 0, 100, 100);
```

Reimplemented in [PICadillo35t](#), [S6D0164](#), [HX8357](#), and [SSD1289](#).

4.45.3.39 uint32_t TFT::rgb2hsv (uint16_t rgb)

Convert a 565 RGB colour to HSV

Calculate the HSV values for a 565 16-bit RGB colour.

Example:

```
unsigned long hsv = tft.rgb2hsv(Color::Green);
```

4.45.3.40 point3d TFT::rgb2xyz (uint16_t rgb)

Get the 3D colour space of a colour

This function converts a 565 colour into a 3D coordinate in RGB colour space (X, Y, Z).

Example:

```
point3d color = tft.rgb2xyz(Color::Cyan);
```

4.45.3.41 void TFT::setCursor (int16_t x, int16_t y) [virtual]

Set the text cursor

All future printing will happen from the pixel (x,y).

Example:

```
tft.setCursor(0, 100);
```

4.45.3.42 void TFT::setFont (const uint8_t* f) [virtual]

Set the current font

The current font is set to the font provided. A font is a byte array of data with metric information embedded in it.

Example:

```
tft.setFont(Fonts::Ubuntu12);
```

4.45.3.43 void TFT::setFontScaleX (uint8_t sx)

Set the X scale of the font

A font can be stretched in either of the X or Y coordinates to make it bigger than normal.

Example:

```
tft.setFontScaleX(2);
```

4.45.3.44 void TFT::setFontScaleY (uint8_t sy)

Set the Y scale of the font

A font can be stretched in either of the X or Y coordinates to make it bigger than normal.

Example:

```
tft.setFontScaleY(2);
```

4.45.3.45 virtual void TFT::setPixel (int16_t x, int16_t y, uint16_t color) [pure virtual]

Draw a pixel

A pixel, coloured (color) is drawn at (x,y).

Example:

```
tft.drawPixel(100, 100, Color::Green);
```

Implemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [HD44780](#), [Aggregator](#), [S6D0164](#), [ILI9340](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), [LEDMatrix](#), [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.45.3.46 `virtual void TFT::setRotation (uint8_t rotation)` [pure virtual]

Set screen rotation

This rotates the screen. Value is between 0 and 3, for 0°, 90°, 180° or 270°.

Example:

```
tft.setRotation(1);
```

Implemented in [SSD1963](#), [Framebuffer](#), [ST7735](#), [PICadillo35t](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), and [LEDMatrix](#).

4.45.3.47 `void TFT::setTextColor (uint16_t c)` [virtual]

Set the text foreground colour

Sets the foreground colour of all future printing to (c).

Example:

```
tft.setTextColor(Color::Magenta);
```

4.45.3.48 `void TFT::setTextColor (uint16_t fg, uint16_t bg)` [virtual]

Sets both foreground and background colour

Sets both the foreground and the background colours of all future printing. If the foreground and background colours match the background will be transparent.

Example:

```
tft.setTextColor(Color::Red, Color::Blue);
```

4.45.3.49 `void TFT::setTextWrap (boolean w)` [virtual]

Enable or disable text wrapping

With text wrapping enabled, when text reaches the right-hand edge of the screen it wraps around back to the left on the next line down. This function allows you to enable (true) or disable (false) this functionality. By default text wrapping is enabled.

Example:

```
tft.setTextWrap(false);
```

4.45.3.50 `uint16_t TFT::stringHeight (char * text)` [virtual]

Calculate the height of a string

As fonts are all fixed height, this just returns the height of the currently selected font in pixels.

Example:

```
int height = stringHeight("The quick brown fox jumped over the lazy dog");
```

4.45.3.51 uint16_t TFT::stringWidth (char * text) [virtual]

Calculate the width of a string

The total width of a string of characters is calculated by examining the width of each character using the current font in turn and accumulating the total width.

Example:

```
int width = tft.stringWidth("The quick brown fox jumped over the lazy dog");
```

4.45.3.52 void TFT::windowData (uint16_t d) [virtual]

Send pixel data to the window

Sends the raw pixel data for one pixel to the currently opened window.

Example:

```
tft.windowData(Color::Red);
```

Reimplemented in [PICadillo35t](#), [S6D0164](#), [HX8357](#), and [SSD1289](#).

4.45.3.53 void TFT::windowData (uint16_t* d, uint32_t l) [virtual]

Send a block of pixel data to the window

The array of pixel data (*d) and size (l) is dumped verbatim to the currently opened window.

Example:

```
tft.windowData(myData, 1000);
```

Reimplemented in [PICadillo35t](#), [S6D0164](#), [HX8357](#), and [SSD1289](#).

4.45.3.54 void TFT::write (uint8_t c)

Write a character to the screen

This writes a single character to the screen at the current cursor position. It is used by (among other things) the print routines for rendering strings.

Example:

```
tft.write('Q');
```

4.45.3.55 point3d TFT::xyz2lab (point3d xyz)

Convert a 3D colour space point to LAB

Calculate the LAB colour space value of a 3D point in RGB colour space.

Example:

```
point3d labcolor = tft.xyz2lab(color3d);
```


4.45.4 Member Data Documentation

4.45.4.1 TFTCommunicator* TFT::_comm

The device used to communicate with the [TFT](#) screen

4.45.4.2 uint16_t TFT::_height

Height of the [TFT](#) screen

4.45.4.3 uint16_t TFT::_width

Width of the [TFT](#) screen

4.45.4.4 int16_t TFT::cursor_x

The text cursor X position

4.45.4.5 int16_t TFT::cursor_y

The text cursor Y position

4.45.4.6 const uint8_t* TFT::font [protected]

A pointer to the currently selected font table

4.45.4.7 uint8_t TFT::font_scale_x [protected]

The current X scaling factor of the font

4.45.4.8 uint8_t TFT::font_scale_y [protected]

The current Y scaling factor of the font

4.45.4.9 uint8_t TFT::rotation

Current rotation

4.45.4.10 uint16_t TFT::textbgcolor

Text background colour

4.45.4.11 uint16_t TFT::textcolor

Text foreground colour

4.45.4.12 boolean TFT::wrap

Whether or not text wrapping is enabled

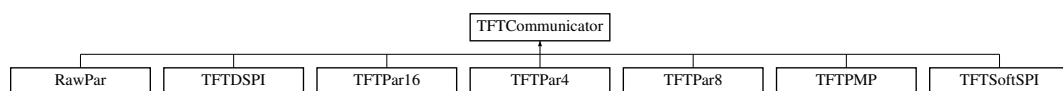
The documentation for this class was generated from the following files:

- TFT.h
- TFT.cpp

4.46 TFTCommunicator Class Reference

```
#include <TFTCommunicator.h>
```

Inheritance diagram for TFTCommunicator:



Public Member Functions

Single functions

Functions to read and write single commands or items of data

- virtual uint8_t [readCommand8](#) ()=0
- virtual uint16_t [readCommand16](#) ()=0
- virtual uint32_t [readCommand32](#) ()=0
- virtual void [writeCommand8](#) (uint8_t command)=0
- virtual void [writeCommand16](#) (uint16_t command)=0
- virtual void [writeCommand32](#) (uint32_t command)=0
- virtual uint8_t [readData8](#) ()=0
- virtual uint16_t [readData16](#) ()=0
- virtual uint32_t [readData32](#) ()=0
- virtual void [writeData8](#) (uint8_t data)=0
- virtual void [writeData16](#) (uint16_t data)=0
- virtual void [writeData32](#) (uint32_t data)=0

Streaming functions

Functions to read and write streams of mixed data and commands

- virtual void [streamStart](#) ()=0
- virtual void [streamEnd](#) ()=0
- virtual void [streamCommand8](#) (uint8_t data)=0
- virtual void [streamCommand16](#) (uint16_t data)=0
- virtual void [streamCommand32](#) (uint32_t data)=0
- virtual uint8_t [streamCommand8](#) ()=0
- virtual uint16_t [streamCommand16](#) ()=0
- virtual uint32_t [streamCommand32](#) ()=0
- virtual void [streamData8](#) (uint8_t data)=0
- virtual void [streamData16](#) (uint16_t data)=0
- virtual void [streamData32](#) (uint32_t data)=0
- virtual uint8_t [streamData8](#) ()=0
- virtual uint16_t [streamData16](#) ()=0
- virtual uint32_t [streamData32](#) ()=0

Block data functions

Functions to write large blocks of data

- virtual void [blockData](#) (uint8_t *data, uint32_t len)=0
- virtual void [blockData](#) (uint16_t *data, uint32_t len)=0
- virtual void [blockData](#) (uint32_t *data, uint32_t len)=0

Device control functions

Functions to manage the device

- virtual void [initializeDevice](#) ()=0
- virtual uint8_t [nativeWidth](#) ()=0

4.46.1 Detailed Description

A [TFTCommunicator](#) device forms the bridge between the [TFT](#) class and the physical screen.

4.46.2 Member Function Documentation

4.46.2.1 virtual void [TFTCommunicator::blockData](#) (uint8_t * *data*, uint32_t *len*) [pure virtual]

Transfer a block of 8-bit data to the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTDSPI](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.46.2.2 virtual void [TFTCommunicator::blockData](#) (uint16_t * *data*, uint32_t *len*) [pure virtual]

Transfer a block of 16-bit data to the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTDSPI](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.46.2.3 virtual void [TFTCommunicator::blockData](#) (uint32_t * *data*, uint32_t *len*) [pure virtual]

Transfer a block of 32-bit data to the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTDSPI](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.46.2.4 virtual void [TFTCommunicator::initializeDevice](#) () [pure virtual]

Initialize the communication device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), [TFTPMP](#), and [TFTDSPI](#).

4.46.2.5 virtual uint8_t [TFTCommunicator::nativeWidth](#) () [pure virtual]

Returns the real physical width of the data channel

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPar8](#), [TFTDSPI](#), [TFTPMP](#), [TFTPar4](#), and [RawPar](#).

4.46.2.6 virtual uint16_t [TFTCommunicator::readCommand16](#) () [pure virtual]

Read a 16-bit command from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), [TFTDSPI](#), and [TFTPMP](#).

4.46.2.7 virtual uint32_t [TFTCommunicator::readCommand32](#) () [pure virtual]

Read a 32-bit command from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), [TFTDSPI](#), and [TFTPMP](#).

4.46.2.8 `virtual uint8_t TFTCommunicator::readCommand8 () [pure virtual]`

Read an 8-bit command from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), [TFTDSPI](#), and [TFTPMP](#).

4.46.2.9 `virtual uint16_t TFTCommunicator::readData16 () [pure virtual]`

Read 16 bits of data from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), [TFTDSPI](#), and [TFTPMP](#).

4.46.2.10 `virtual uint32_t TFTCommunicator::readData32 () [pure virtual]`

Read 32 bits of data from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), [TFTDSPI](#), and [TFTPMP](#).

4.46.2.11 `virtual uint8_t TFTCommunicator::readData8 () [pure virtual]`

Read 8 bits of data from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), [TFTDSPI](#), and [TFTPMP](#).

4.46.2.12 `virtual void TFTCommunicator::streamCommand16 (uint16_t data) [pure virtual]`

Send a 16-bit command through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.13 `virtual uint16_t TFTCommunicator::streamCommand16 () [pure virtual]`

Read a 16-bit command through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTDSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.46.2.14 `virtual void TFTCommunicator::streamCommand32 (uint32_t data) [pure virtual]`

Send a 32-bit command through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.15 `virtual uint32_t TFTCommunicator::streamCommand32 () [pure virtual]`

Read a 32-bit command through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTDSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.46.2.16 `virtual void TFTCommunicator::streamCommand8 (uint8_t data) [pure virtual]`

Send an 8-bit command through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.17 `virtual uint8_t TFTCommunicator::streamCommand8 () [pure virtual]`

Read an 8-bit command through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTDSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.46.2.18 `virtual void TFTCommunicator::streamData16 (uint16_t data) [pure virtual]`

Send 16-bits of data through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.19 `virtual uint16_t TFTCommunicator::streamData16 () [pure virtual]`

Read 16 bits of data through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.46.2.20 `virtual void TFTCommunicator::streamData32 (uint32_t data) [pure virtual]`

Send 32-bits of data through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.21 `virtual uint32_t TFTCommunicator::streamData32 () [pure virtual]`

Read 32 bits of data through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.46.2.22 `virtual void TFTCommunicator::streamData8 (uint8_t data) [pure virtual]`

Send 8-bits of data through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.23 `virtual uint8_t TFTCommunicator::streamData8 () [pure virtual]`

Read 8 bits of data through the stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.46.2.24 `virtual void TFTCommunicator::streamEnd () [pure virtual]`

Close the currently open stream

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTSoftSPI](#), [TFTPMP](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.25 `virtual void TFTCommunicator::streamStart () [pure virtual]`

Open a stream to the device endpoint

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTSoftSPI](#), [TFTPMP](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.26 `virtual void TFTCommunicator::writeCommand16 (uint16_t command) [pure virtual]`

Write a 16-bit command to the device

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.27 `virtual void TFTCommunicator::writeCommand32 (uint32_t command) [pure virtual]`

Write a 32-bit command to the device

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.28 `virtual void TFTCommunicator::writeCommand8 (uint8_t command) [pure virtual]`

Write an 8-bit command to the device

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.29 `virtual void TFTCommunicator::writeData16 (uint16_t data) [pure virtual]`

Write 16 bits of data to the device

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.30 `virtual void TFTCommunicator::writeData32 (uint32_t data) [pure virtual]`

Write 32 bits of data to the device

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.46.2.31 `virtual void TFTCommunicator::writeData8 (uint8_t data) [pure virtual]`

Write 8 bits of data to the device

Implemented in [TFTPar16](#), [TFTDSPI](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

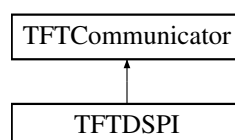
The documentation for this class was generated from the following file:

- [TFTCommunicator.h](#)

4.47 TFTDSPI Class Reference

```
#include <TFTDSPI.h>
```

Inheritance diagram for TFTDSPI:



Public Member Functions

- void [initializeDevice](#) ()

- uint8_t readCommand8 ()
- uint16_t readCommand16 ()
- uint32_t readCommand32 ()
- uint8_t readData8 ()
- uint16_t readData16 ()
- uint32_t readData32 ()
- void writeCommand8 (uint8_t command)
- void writeCommand16 (uint16_t command)
- void writeCommand32 (uint32_t command)
- void writeData8 (uint8_t data)
- void writeData16 (uint16_t data)
- void writeData32 (uint32_t data)
- void streamStart ()
- void streamEnd ()
- void streamCommand8 (uint8_t)
- void streamCommand16 (uint16_t)
- void streamCommand32 (uint32_t)
- uint8_t streamCommand8 ()
- uint16_t streamCommand16 ()
- uint32_t streamCommand32 ()
- void streamData8 (uint8_t)
- void streamData16 (uint16_t)
- void streamData32 (uint32_t)
- uint8_t streamData8 ()
- uint16_t streamData16 ()
- uint32_t streamData32 ()
- uint8_t nativeWidth ()
- void blockData (uint8_t *data, uint32_t len)
- void blockData (uint16_t *data, uint32_t len)
- void blockData (uint32_t *data, uint32_t len)
- **TFTDSPI** (DSPI *spi, uint8_t cs, uint8_t dc=255, uint32_t sp=40000000UL)
- **TFTDSPI** (DSPI &spi, uint8_t cs, uint8_t dc=255, uint32_t sp=40000000UL)

4.47.1 Detailed Description

The **TFTDSPI** class creates a new SPI interface using the chipKIT DSPI library.

4.47.2 Constructor & Destructor Documentation

4.47.2.1 **TFTDSPI::TFTDSPI (DSPI * spi, uint8_t cs, uint8_t dc = 255, uint32_t sp = 40000000UL)** [inline]

Construct a new SPI communication object. Pass either a pointer or reference to a DSPI object, a Chip Select pin, a Data/Command pin and (optionally) a communication speed.

Example:

```
DSPI0 spi;
TFTDSPI mySPI(spi, 10, 8);
```

4.47.3 Member Function Documentation

4.47.3.1 **void TFTDSPI::blockData (uint8_t * data, uint32_t len)** [virtual]

Transfer a block of 8-bit data to the device

Implements **TFTCommunicator**.

4.47.3.2 void TFTDSPI::blockData (uint16_t * data, uint32_t len) [virtual]

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

4.47.3.3 void TFTDSPI::blockData (uint32_t * data, uint32_t len) [virtual]

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.47.3.4 void TFTDSPI::initializeDevice () [virtual]

Initialize the communication device

Implements [TFTCommunicator](#).

4.47.3.5 uint8_t TFTDSPI::nativeWidth () [inline],[virtual]

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.47.3.6 uint16_t TFTDSPI::readCommand16 () [virtual]

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.47.3.7 uint32_t TFTDSPI::readCommand32 () [virtual]

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.47.3.8 uint8_t TFTDSPI::readCommand8 () [virtual]

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.47.3.9 uint16_t TFTDSPI::readData16 () [virtual]

Read 16 bits of data from the device

Implements [TFTCommunicator](#).

4.47.3.10 uint32_t TFTDSPI::readData32 () [virtual]

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.47.3.11 `uint8_t TFTDSPI::readData8 () [virtual]`

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.47.3.12 `void TFTDSPI::streamCommand16 (uint16_t data) [virtual]`

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.47.3.13 `uint16_t TFTDSPI::streamCommand16 () [virtual]`

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.47.3.14 `void TFTDSPI::streamCommand32 (uint32_t data) [virtual]`

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.47.3.15 `uint32_t TFTDSPI::streamCommand32 () [virtual]`

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.47.3.16 `void TFTDSPI::streamCommand8 (uint8_t data) [virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.47.3.17 `uint8_t TFTDSPI::streamCommand8 () [virtual]`

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.47.3.18 `void TFTDSPI::streamData16 (uint16_t data) [virtual]`

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.19 `uint16_t TFTDSPI::streamData16 () [virtual]`

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.20 void TFTDSPI::streamData32 (uint32_t *data*) [virtual]

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.21 uint32_t TFTDSPI::streamData32 () [virtual]

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.22 void TFTDSPI::streamData8 (uint8_t *data*) [virtual]

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.23 uint8_t TFTDSPI::streamData8 () [virtual]

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.24 void TFTDSPI::streamEnd () [virtual]

Close the currently open stream

Implements [TFTCommunicator](#).

4.47.3.25 void TFTDSPI::streamStart () [virtual]

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.47.3.26 void TFTDSPI::writeCommand16 (uint16_t *command*) [virtual]

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.47.3.27 void TFTDSPI::writeCommand32 (uint32_t *command*) [virtual]

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.47.3.28 void TFTDSPI::writeCommand8 (uint8_t *command*) [virtual]

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.47.3.29 void TFTDSPI::writeData16 (uint16_t data) [virtual]

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.47.3.30 void TFTDSPI::writeData32 (uint32_t data) [virtual]

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.47.3.31 void TFTDSPI::writeData8 (uint8_t data) [virtual]

Write 8 bits of data to the device

Implements [TFTCommunicator](#).

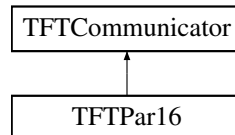
The documentation for this class was generated from the following file:

- TFTDSPI.h

4.48 TFTPar16 Class Reference

```
#include <TFTPar16.h>
```

Inheritance diagram for TFTPar16:



Public Member Functions

- [TFTPar16](#) (uint8_t cs, uint8_t dc, uint8_t clk, uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3, uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7, uint8_t d8, uint8_t d9, uint8_t d10, uint8_t d11, uint8_t d12, uint8_t d13, uint8_t d14, uint8_t d15)
- [TFTPar16](#) (const uint8_t *profile)
- void [writeCommand8](#) (uint8_t command)
- void [writeCommand16](#) (uint16_t command)
- void [writeCommand32](#) (uint32_t command)
- void [writeData8](#) (uint8_t data)
- void [writeData16](#) (uint16_t data)
- void [writeData32](#) (uint32_t data)
- void [streamStart](#) ()
- void [streamEnd](#) ()
- void [streamCommand8](#) (uint8_t)
- void [streamCommand16](#) (uint16_t)
- void [streamCommand32](#) (uint32_t)
- void [streamData8](#) (uint8_t)
- void [streamData16](#) (uint16_t)
- void [streamData32](#) (uint32_t)
- uint8_t [streamCommand8](#) ()

- uint16_t [streamCommand16](#) ()
- uint32_t [streamCommand32](#) ()
- uint8_t [streamData8](#) ()
- uint16_t [streamData16](#) ()
- uint32_t [streamData32](#) ()
- uint8_t [readCommand8](#) ()
- uint16_t [readCommand16](#) ()
- uint32_t [readCommand32](#) ()
- uint8_t [readData8](#) ()
- uint16_t [readData16](#) ()
- uint32_t [readData32](#) ()
- uint8_t [nativeWidth](#) ()
- void [initializeDevice](#) ()
- void [blockData](#) (uint8_t *d, uint32_t)
- void [blockData](#) (uint16_t *d, uint32_t)
- void [blockData](#) (uint32_t *d, uint32_t)

Static Public Attributes

- static const uint8_t [ltheadAdapter](#) []

4.48.1 Detailed Description

The [TFTPar16](#) class creates a full 16-bit parallel interface to a [TFT](#) device

4.48.2 Constructor & Destructor Documentation

4.48.2.1 [TFTPar16::TFTPar16](#) (uint8_t *cs*, uint8_t *dc*, uint8_t *clk*, uint8_t *d0*, uint8_t *d1*, uint8_t *d2*, uint8_t *d3*, uint8_t *d4*, uint8_t *d5*, uint8_t *d6*, uint8_t *d7*, uint8_t *d8*, uint8_t *d9*, uint8_t *d10*, uint8_t *d11*, uint8_t *d12*, uint8_t *d13*, uint8_t *d14*, uint8_t *d15*) [inline]

Construct a new 16-bit parallel device using individual pins

4.48.2.2 [TFTPar16::TFTPar16](#) (const uint8_t* *profile*) [inline]

Construct a new 16-bit parallel device using a pre-programmed profile

4.48.3 Member Function Documentation

4.48.3.1 void [TFTPar16::blockData](#) (uint8_t* *data*, uint32_t *len*) [virtual]

Transfer a block of 8-bit data to the device

Implements [TFTCommunicator](#).

4.48.3.2 void [TFTPar16::blockData](#) (uint16_t* *data*, uint32_t *len*) [virtual]

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

4.48.3.3 void TFTPar16::blockData (uint32_t* *data*, uint32_t *len*) [virtual]

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.48.3.4 void TFTPar16::initializeDevice () [inline],[virtual]

Initialize the communication device

Implements [TFTCommunicator](#).

4.48.3.5 uint8_t TFTPar16::nativeWidth () [inline],[virtual]

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.48.3.6 uint16_t TFTPar16::readCommand16 () [inline],[virtual]

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.48.3.7 uint32_t TFTPar16::readCommand32 () [inline],[virtual]

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.48.3.8 uint8_t TFTPar16::readCommand8 () [inline],[virtual]

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.48.3.9 uint16_t TFTPar16::readData16 () [inline],[virtual]

Read 16 bits of data from the device

Implements [TFTCommunicator](#).

4.48.3.10 uint32_t TFTPar16::readData32 () [inline],[virtual]

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.48.3.11 uint8_t TFTPar16::readData8 () [inline],[virtual]

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.48.3.12 void TFTPar16::streamCommand16 (uint16_t *data*) [virtual]

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.13 uint16_t TFTPar16::streamCommand16 () [inline],[virtual]

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.14 void TFTPar16::streamCommand32 (uint32_t *data*) [virtual]

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.15 uint32_t TFTPar16::streamCommand32 () [inline],[virtual]

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.16 void TFTPar16::streamCommand8 (uint8_t *data*) [virtual]

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.17 uint8_t TFTPar16::streamCommand8 () [inline],[virtual]

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.18 void TFTPar16::streamData16 (uint16_t *data*) [virtual]

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.19 uint16_t TFTPar16::streamData16 () [inline],[virtual]

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.20 void TFTPar16::streamData32 (uint32_t *data*) [virtual]

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.21 `uint32_t TFTPar16::streamData32 () [inline],[virtual]`

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.22 `void TFTPar16::streamData8 (uint8_t data) [virtual]`

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.23 `uint8_t TFTPar16::streamData8 () [inline],[virtual]`

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.24 `void TFTPar16::streamEnd () [virtual]`

Close the currently open stream

Implements [TFTCommunicator](#).

4.48.3.25 `void TFTPar16::streamStart () [virtual]`

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.48.3.26 `void TFTPar16::writeCommand16 (uint16_t command) [virtual]`

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.48.3.27 `void TFTPar16::writeCommand32 (uint32_t command) [virtual]`

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.48.3.28 `void TFTPar16::writeCommand8 (uint8_t command) [virtual]`

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.48.3.29 `void TFTPar16::writeData16 (uint16_t data) [virtual]`

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.48.3.30 void TFTPar16::writeData32 (uint32_t data) [virtual]

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.48.3.31 void TFTPar16::writeData8 (uint8_t data) [virtual]

Write 8 bits of data to the device

Implements [TFTCommunicator](#).

4.48.4 Member Data Documentation

4.48.4.1 const uint8_t TFTPar16::lteadAdapter [static]

Initial value:

```
= {
0, 1, 2,
3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18
}
```

Interface profile for the ITeard Studios Arduino Mega [TFT](#) Interface Adapter

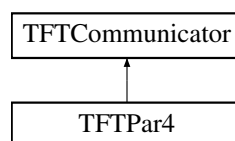
The documentation for this class was generated from the following files:

- TFTPar16.h
- TFTPar16.cpp

4.49 TFTPar4 Class Reference

```
#include <TFTPar4.h>
```

Inheritance diagram for TFTPar4:



Public Member Functions

- [TFTPar4](#) (uint8_t dc, uint8_t clk, uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3)
- [TFTPar4](#) (uint8_t cs, uint8_t dc, uint8_t clk, uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3)
- void [writeCommand8](#) (uint8_t command)
- void [writeCommand16](#) (uint16_t command)
- void [writeCommand32](#) (uint32_t command)
- void [writeData8](#) (uint8_t data)
- void [writeData16](#) (uint16_t data)
- void [writeData32](#) (uint32_t data)
- void [streamStart](#) ()
- void [streamEnd](#) ()

- void [streamCommand8](#) (uint8_t)
- void [streamCommand16](#) (uint16_t)
- void [streamCommand32](#) (uint32_t)
- void [streamData8](#) (uint8_t)
- void [streamData16](#) (uint16_t)
- void [streamData32](#) (uint32_t)
- uint8_t [nativeWidth](#) ()

4.49.1 Detailed Description

The [TFTPar4](#) class creates a new 4-bit interface compatible with popular text only LCD screens

4.49.2 Constructor & Destructor Documentation

4.49.2.1 [TFTPar4::TFTPar4](#) (uint8_t *dc*, uint8_t *clk*, uint8_t *d0*, uint8_t *d1*, uint8_t *d2*, uint8_t *d3*)

Create a new [TFTPar4](#) object

This creates a new 4-bit [TFT](#) interface, including the Data/Command (dc) pin, Clock (clk) (sometimes called E) and data bits d0 - d3.

4.49.2.2 [TFTPar4::TFTPar4](#) (uint8_t *cs*, uint8_t *dc*, uint8_t *clk*, uint8_t *d0*, uint8_t *d1*, uint8_t *d2*, uint8_t *d3*)

Create a new [TFTPar4](#) object

This creates a new 4-bit full [TFT](#) interface, including the Chip Select (cs) pin, Data/Command (dc) pin, Clock (clk) (sometimes called E) and data bits d0 - d3.

4.49.3 Member Function Documentation

4.49.3.1 [uint8_t TFTPar4::nativeWidth](#) () [\[inline\]](#), [\[virtual\]](#)

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.49.3.2 [void TFTPar4::streamCommand16](#) (uint16_t *data*) [\[virtual\]](#)

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.49.3.3 [void TFTPar4::streamCommand32](#) (uint32_t *data*) [\[virtual\]](#)

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.49.3.4 [void TFTPar4::streamCommand8](#) (uint8_t *data*) [\[virtual\]](#)

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.49.3.5 void TFTPar4::streamData16 (uint16_t *data*) [virtual]

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.49.3.6 void TFTPar4::streamData32 (uint32_t *data*) [virtual]

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.49.3.7 void TFTPar4::streamData8 (uint8_t *data*) [virtual]

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.49.3.8 void TFTPar4::streamEnd () [virtual]

Close the currently open stream

Implements [TFTCommunicator](#).

4.49.3.9 void TFTPar4::streamStart () [virtual]

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.49.3.10 void TFTPar4::writeCommand16 (uint16_t *command*) [virtual]

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.49.3.11 void TFTPar4::writeCommand32 (uint32_t *command*) [virtual]

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.49.3.12 void TFTPar4::writeCommand8 (uint8_t *command*) [virtual]

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.49.3.13 void TFTPar4::writeData16 (uint16_t *data*) [virtual]

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.49.3.14 void TFTPar4::writeData32 (uint32_t data) [virtual]

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.49.3.15 void TFTPar4::writeData8 (uint8_t data) [virtual]

Write 8 bits of data to the device

Implements [TFTCommunicator](#).

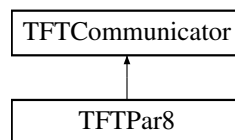
The documentation for this class was generated from the following files:

- TFTPar4.h
- TFTPar4.cpp

4.50 TFTPar8 Class Reference

```
#include <TFTPar8.h>
```

Inheritance diagram for TFTPar8:



Public Member Functions

- [TFTPar8](#) ([Parallelio](#) *dev, uint8_t cs, uint8_t dc, uint8_t clk, uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3, uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7)
- void [writeCommand8](#) (uint8_t command)
- void [writeCommand16](#) (uint16_t command)
- void [writeCommand32](#) (uint32_t command)
- void [writeData8](#) (uint8_t data)
- void [writeData16](#) (uint16_t data)
- void [writeData32](#) (uint32_t data)
- void [streamStart](#) ()
- void [streamEnd](#) ()
- void [streamCommand8](#) (uint8_t)
- void [streamCommand16](#) (uint16_t)
- void [streamCommand32](#) (uint32_t)
- void [streamData8](#) (uint8_t)
- void [streamData16](#) (uint16_t)
- void [streamData32](#) (uint32_t)
- uint8_t [streamCommand8](#) ()
- uint16_t [streamCommand16](#) ()
- uint32_t [streamCommand32](#) ()
- uint8_t [streamData8](#) ()
- uint16_t [streamData16](#) ()
- uint32_t [streamData32](#) ()
- uint8_t [readCommand8](#) ()

- uint16_t [readCommand16](#) ()
- uint32_t [readCommand32](#) ()
- uint8_t [readData8](#) ()
- uint16_t [readData16](#) ()
- uint32_t [readData32](#) ()
- uint8_t [nativeWidth](#) ()
- void [initializeDevice](#) ()
- void [blockData](#) (uint8_t *d, uint32_t)
- void [blockData](#) (uint16_t *d, uint32_t)
- void [blockData](#) (uint32_t *d, uint32_t)

4.50.1 Detailed Description

The [TFTPar8](#) class defines an 8-bit parallel interface incorporating the normal [TFT](#) control signals.

4.50.2 Constructor & Destructor Documentation

4.50.2.1 [TFTPar8::TFTPar8](#) ([ParallelIO](#) * *dev*, uint8_t *cs*, uint8_t *dc*, uint8_t *clk*, uint8_t *d0*, uint8_t *d1*, uint8_t *d2*, uint8_t *d3*, uint8_t *d4*, uint8_t *d5*, uint8_t *d6*, uint8_t *d7*)

Construct a new [TFTPar8](#) device

This class constructor creates a new 8-bit parallel interface. It requires a [ParallelIO](#) device pointer (*dev) to access the IO pins. Also required are the Chip Select (cs), Data/Command (dc) Clock (clk) and data pins 0-7 (d0-d7);

Example:

```
TFTPar8 pardev(&core, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 11);
```

4.50.3 Member Function Documentation

4.50.3.1 void [TFTPar8::blockData](#) (uint8_t * *data*, uint32_t *len*) [virtual]

Transfer a block of 8-bit data to the device

Implements [TFTCommunicator](#).

4.50.3.2 void [TFTPar8::blockData](#) (uint16_t * *data*, uint32_t *len*) [virtual]

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

4.50.3.3 void [TFTPar8::blockData](#) (uint32_t * *data*, uint32_t *len*) [virtual]

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.50.3.4 void [TFTPar8::initializeDevice](#) () [inline],[virtual]

Initialize the communication device

Implements [TFTCommunicator](#).

4.50.3.5 `uint8_t TFTPar8::nativeWidth () [inline],[virtual]`

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.50.3.6 `uint16_t TFTPar8::readCommand16 () [inline],[virtual]`

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.50.3.7 `uint32_t TFTPar8::readCommand32 () [inline],[virtual]`

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.50.3.8 `uint8_t TFTPar8::readCommand8 () [inline],[virtual]`

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.50.3.9 `uint16_t TFTPar8::readData16 () [inline],[virtual]`

Read 16 bits of data from the device

Implements [TFTCommunicator](#).

4.50.3.10 `uint32_t TFTPar8::readData32 () [inline],[virtual]`

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.50.3.11 `uint8_t TFTPar8::readData8 () [inline],[virtual]`

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.50.3.12 `void TFTPar8::streamCommand16 (uint16_t data) [virtual]`

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.13 `uint16_t TFTPar8::streamCommand16 () [inline],[virtual]`

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.14 `void TFTPar8::streamCommand32 (uint32_t data) [virtual]`

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.15 `uint32_t TFTPar8::streamCommand32 () [inline],[virtual]`

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.16 `void TFTPar8::streamCommand8 (uint8_t data) [virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.17 `uint8_t TFTPar8::streamCommand8 () [inline],[virtual]`

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.18 `void TFTPar8::streamData16 (uint16_t data) [virtual]`

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.19 `uint16_t TFTPar8::streamData16 () [inline],[virtual]`

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.20 `void TFTPar8::streamData32 (uint32_t data) [virtual]`

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.21 `uint32_t TFTPar8::streamData32 () [inline],[virtual]`

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.22 `void TFTPar8::streamData8 (uint8_t data) [virtual]`

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.23 `uint8_t TFTPar8::streamData8 () [inline],[virtual]`

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.24 `void TFTPar8::streamEnd () [virtual]`

Close the currently open stream

Implements [TFTCommunicator](#).

4.50.3.25 `void TFTPar8::streamStart () [virtual]`

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.50.3.26 `void TFTPar8::writeCommand16 (uint16_t command) [virtual]`

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.50.3.27 `void TFTPar8::writeCommand32 (uint32_t command) [virtual]`

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.50.3.28 `void TFTPar8::writeCommand8 (uint8_t command) [virtual]`

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.50.3.29 `void TFTPar8::writeData16 (uint16_t data) [virtual]`

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.50.3.30 `void TFTPar8::writeData32 (uint32_t data) [virtual]`

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.50.3.31 `void TFTPar8::writeData8 (uint8_t data) [virtual]`

Write 8 bits of data to the device

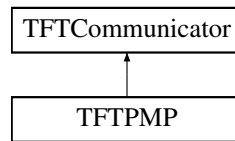
Implements [TFTCommunicator](#).

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

- TFTPar8.h
- TFTPar8.cpp

4.51 TFTPMP Class Reference

Inheritance diagram for TFTPMP:



Public Member Functions

- uint8_t [readCommand8](#) ()
- uint16_t [readCommand16](#) ()
- uint32_t [readCommand32](#) ()
- uint8_t [readData8](#) ()
- uint16_t [readData16](#) ()
- uint32_t [readData32](#) ()
- void [writeCommand8](#) (uint8_t command)
- void [writeCommand16](#) (uint16_t command)
- void [writeCommand32](#) (uint32_t command)
- void [writeData8](#) (uint8_t data)
- void [writeData16](#) (uint16_t data)
- void [writeData32](#) (uint32_t data)
- void [streamStart](#) ()
- void [streamEnd](#) ()
- uint8_t [streamCommand8](#) ()
- uint16_t [streamCommand16](#) ()
- uint32_t [streamCommand32](#) ()
- uint8_t [streamData8](#) ()
- uint16_t [streamData16](#) ()
- uint32_t [streamData32](#) ()
- void [streamCommand8](#) (uint8_t)
- void [streamCommand16](#) (uint16_t)
- void [streamCommand32](#) (uint32_t)
- void [streamData8](#) (uint8_t)
- void [streamData16](#) (uint16_t)
- void [streamData32](#) (uint32_t)
- void [blockData](#) (uint8_t *, uint32_t)
- void [blockData](#) (uint16_t *, uint32_t)
- void [blockData](#) (uint32_t *, uint32_t)
- uint8_t [nativeWidth](#) ()
- void [initializeDevice](#) ()

4.51.1 Member Function Documentation

4.51.1.1 void [TFTPMP::blockData](#) (uint8_t * *data*, uint32_t *len*) [inline],[virtual]

Transfer a block of 8-bit data to the device

Implements [TFTCommunicator](#).

4.51.1.2 void TFTPMP::blockData (uint16_t * *data*, uint32_t *len*) [inline],[virtual]

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

4.51.1.3 void TFTPMP::blockData (uint32_t * *data*, uint32_t *len*) [inline],[virtual]

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.51.1.4 void TFTPMP::initializeDevice () [virtual]

Initialize the communication device

Implements [TFTCommunicator](#).

4.51.1.5 uint8_t TFTPMP::nativeWidth () [inline],[virtual]

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.51.1.6 uint16_t TFTPMP::readCommand16 () [inline],[virtual]

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.51.1.7 uint32_t TFTPMP::readCommand32 () [inline],[virtual]

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.51.1.8 uint8_t TFTPMP::readCommand8 () [inline],[virtual]

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.51.1.9 uint16_t TFTPMP::readData16 () [inline],[virtual]

Read 16 bits of data from the device

Implements [TFTCommunicator](#).

4.51.1.10 uint32_t TFTPMP::readData32 () [inline],[virtual]

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.51.1.11 `uint8_t TFTMPMP::readData8 () [inline],[virtual]`

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.51.1.12 `uint16_t TFTMPMP::streamCommand16 () [inline],[virtual]`

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.51.1.13 `void TFTMPMP::streamCommand16 (uint16_t data) [inline],[virtual]`

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.51.1.14 `uint32_t TFTMPMP::streamCommand32 () [inline],[virtual]`

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.51.1.15 `void TFTMPMP::streamCommand32 (uint32_t data) [inline],[virtual]`

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.51.1.16 `uint8_t TFTMPMP::streamCommand8 () [inline],[virtual]`

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.51.1.17 `void TFTMPMP::streamCommand8 (uint8_t data) [inline],[virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.51.1.18 `uint16_t TFTMPMP::streamData16 () [inline],[virtual]`

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.51.1.19 `void TFTMPMP::streamData16 (uint16_t data) [inline],[virtual]`

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.51.1.20 `uint32_t TFTPMP::streamData32() [inline],[virtual]`

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.51.1.21 `void TFTPMP::streamData32(uint32_t data) [inline],[virtual]`

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.51.1.22 `uint8_t TFTPMP::streamData8() [inline],[virtual]`

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.51.1.23 `void TFTPMP::streamData8(uint8_t data) [inline],[virtual]`

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.51.1.24 `void TFTPMP::streamEnd() [inline],[virtual]`

Close the currently open stream

Implements [TFTCommunicator](#).

4.51.1.25 `void TFTPMP::streamStart() [inline],[virtual]`

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.51.1.26 `void TFTPMP::writeCommand16(uint16_t command) [inline],[virtual]`

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.51.1.27 `void TFTPMP::writeCommand32(uint32_t command) [inline],[virtual]`

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.51.1.28 `void TFTPMP::writeCommand8(uint8_t command) [inline],[virtual]`

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.51.1.29 `void TFTPMP::writeData16 (uint16_t data) [inline],[virtual]`

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.51.1.30 `void TFTPMP::writeData32 (uint32_t data) [inline],[virtual]`

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.51.1.31 `void TFTPMP::writeData8 (uint8_t data) [inline],[virtual]`

Write 8 bits of data to the device

Implements [TFTCommunicator](#).

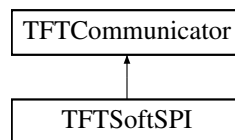
The documentation for this class was generated from the following files:

- TFTPMP.h
- TFTPMP.cpp

4.52 TFTSoftSPI Class Reference

```
#include <TFTSoftSPI.h>
```

Inheritance diagram for TFTSoftSPI:



Public Member Functions

- [TFTSoftSPI](#) (uint8_t sdo, uint8_t sck, uint8_t cs, uint8_t dc)
- void [writeCommand8](#) (uint8_t command)
- void [writeCommand16](#) (uint16_t command)
- void [writeCommand32](#) (uint32_t command)
- void [writeData8](#) (uint8_t data)
- void [writeData16](#) (uint16_t data)
- void [writeData32](#) (uint32_t data)
- uint8_t [readCommand8](#) ()
- uint16_t [readCommand16](#) ()
- uint32_t [readCommand32](#) ()
- uint8_t [readData8](#) ()
- uint16_t [readData16](#) ()
- uint32_t [readData32](#) ()
- void [streamStart](#) ()
- void [streamEnd](#) ()
- void [streamCommand8](#) (uint8_t)
- void [streamCommand16](#) (uint16_t)
- void [streamCommand32](#) (uint32_t)

- void [streamData8](#) (uint8_t)
- void [streamData16](#) (uint16_t)
- void [streamData32](#) (uint32_t)
- uint8_t [streamCommand8](#) ()
- uint16_t [streamCommand16](#) ()
- uint32_t [streamCommand32](#) ()
- uint8_t [streamData8](#) ()
- uint16_t [streamData16](#) ()
- uint32_t [streamData32](#) ()
- void [blockData](#) (uint8_t *d, uint32_t c)
- void [blockData](#) (uint16_t *d, uint32_t c)
- void [blockData](#) (uint32_t *d, uint32_t c)
- void [initializeDevice](#) ()
- uint8_t [nativeWidth](#) ()

4.52.1 Detailed Description

The [TFTSoftSPI](#) communicator creates an SPI channel on any IO pins. Not as fast as hardware SPI, but allows extra flexibility.

4.52.2 Constructor & Destructor Documentation

4.52.2.1 [TFTSoftSPI::TFTSoftSPI](#) (uint8_t *sdo*, uint8_t *sck*, uint8_t *cs*, uint8_t *dc*)

Create a new software SPI communicator

This constructor takes 4 IO pins and creates a uni-directional (write-only) software SPI channel. The pins required are Serial Data Out (sdo), Serial Clock (sck), Chip Select (cs) and Data/Command (dc).

Example:

```
TFTSoftSPI mySPI(4, 5, 6, 7);
```

4.52.3 Member Function Documentation

4.52.3.1 void [TFTSoftSPI::blockData](#) (uint8_t * *data*, uint32_t *len*) [inline],[virtual]

Transfer a block of 8-bit data to the device

Implements [TFTCommunicator](#).

4.52.3.2 void [TFTSoftSPI::blockData](#) (uint16_t * *data*, uint32_t *len*) [inline],[virtual]

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

4.52.3.3 void [TFTSoftSPI::blockData](#) (uint32_t * *data*, uint32_t *len*) [inline],[virtual]

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.52.3.4 `void TFTSoftSPI::initializeDevice () [inline],[virtual]`

Initialize the communication device

Implements [TFTCommunicator](#).

4.52.3.5 `uint8_t TFTSoftSPI::nativeWidth () [inline],[virtual]`

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.52.3.6 `uint16_t TFTSoftSPI::readCommand16 () [inline],[virtual]`

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.52.3.7 `uint32_t TFTSoftSPI::readCommand32 () [inline],[virtual]`

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.52.3.8 `uint8_t TFTSoftSPI::readCommand8 () [inline],[virtual]`

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.52.3.9 `uint16_t TFTSoftSPI::readData16 () [inline],[virtual]`

Read 16 bits of data from the device

Implements [TFTCommunicator](#).

4.52.3.10 `uint32_t TFTSoftSPI::readData32 () [inline],[virtual]`

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.52.3.11 `uint8_t TFTSoftSPI::readData8 () [inline],[virtual]`

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.52.3.12 `void TFTSoftSPI::streamCommand16 (uint16_t data) [virtual]`

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.52.3.13 `uint16_t TFTSoftSPI::streamCommand16 () [inline],[virtual]`

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.52.3.14 `void TFTSoftSPI::streamCommand32 (uint32_t data) [virtual]`

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.52.3.15 `uint32_t TFTSoftSPI::streamCommand32 () [inline],[virtual]`

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.52.3.16 `void TFTSoftSPI::streamCommand8 (uint8_t data) [virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.52.3.17 `uint8_t TFTSoftSPI::streamCommand8 () [inline],[virtual]`

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.52.3.18 `void TFTSoftSPI::streamData16 (uint16_t data) [virtual]`

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.52.3.19 `uint16_t TFTSoftSPI::streamData16 () [inline],[virtual]`

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.52.3.20 `void TFTSoftSPI::streamData32 (uint32_t data) [virtual]`

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.52.3.21 `uint32_t TFTSoftSPI::streamData32 () [inline],[virtual]`

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.52.3.22 void TFTSoftSPI::streamData8 (uint8_t *data*) [virtual]

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.52.3.23 uint8_t TFTSoftSPI::streamData8 () [inline],[virtual]

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.52.3.24 void TFTSoftSPI::streamEnd () [virtual]

Close the currently open stream

Implements [TFTCommunicator](#).

4.52.3.25 void TFTSoftSPI::streamStart () [virtual]

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.52.3.26 void TFTSoftSPI::writeCommand16 (uint16_t *command*) [virtual]

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.52.3.27 void TFTSoftSPI::writeCommand32 (uint32_t *command*) [virtual]

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.52.3.28 void TFTSoftSPI::writeCommand8 (uint8_t *command*) [virtual]

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.52.3.29 void TFTSoftSPI::writeData16 (uint16_t *data*) [virtual]

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.52.3.30 void TFTSoftSPI::writeData32 (uint32_t *data*) [virtual]

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.52.3.31 `void TFTSoftSPI::writeData8 (uint8_t data) [virtual]`

Write 8 bits of data to the device

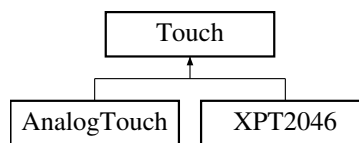
Implements [TFTCommunicator](#).

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

- TFTSoftSPI.h
- TFTSoftSPI.cpp

4.53 Touch Class Reference

Inheritance diagram for Touch:



Public Member Functions

- [Touch](#) ([TFTCommunicator](#) *comm, uint16_t w, uint16_t h)
- [Touch](#) ([TFTCommunicator](#) &comm, uint16_t w, uint16_t h)
- [Touch](#) (uint16_t w, uint16_t h)
- virtual void [initializeDevice](#) ()=0
- virtual uint16_t [x](#) ()=0
- virtual uint16_t [y](#) ()=0
- virtual boolean [isPressed](#) ()=0
- virtual uint16_t [pressure](#) ()
- virtual void [setRotation](#) (uint8_t r)
- virtual void [sample](#) ()=0

Protected Attributes

- [TFTCommunicator](#) * [_comm](#)
- uint16_t [_width](#)
- uint16_t [_height](#)

4.53.1 Constructor & Destructor Documentation

4.53.1.1 `Touch::Touch (TFTCommunicator * comm, uint16_t w, uint16_t h) [inline]`

Create a new touch screen object

This takes a pointer to a communication device, and the width and height of the touch screen.

4.53.1.2 `Touch::Touch (TFTCommunicator & comm, uint16_t w, uint16_t h) [inline]`

Create a new touch screen object

This takes a reference to a communication device, and the width and height of the touch screen.

4.53.1.3 Touch::Touch (uint16_t w, uint16_t h) [inline]

Create a new touch screen object

This creates a new controller-less touch device.

4.53.2 Member Function Documentation

4.53.2.1 virtual void Touch::initializeDevice () [pure virtual]

Initialize the device

This configures and enables the touch screen device. It should be called before any other touch screen functions.

Implemented in [XPT2046](#), and [AnalogTouch](#).

4.53.2.2 virtual boolean Touch::isPressed () [pure virtual]

Get pressed status

Returns true if the touch screen is pressed, false otherwise.

Implemented in [XPT2046](#), and [AnalogTouch](#).

4.53.2.3 virtual uint16_t Touch::pressure () [inline],[virtual]

Calculate the touch pressure

For touch screens that can calculate how hard you are pressing them, this returns the pressure value. For others it returns 0.

Example:

```
int pressure = ts.pressure();
```

Reimplemented in [AnalogTouch](#).

4.53.2.4 virtual void Touch::sample () [pure virtual]

Sample the touch screen

This performs a sampling of the touch screen to get the current coordinates and touch status. It should be called regularly to update the current touch screen data.

Implemented in [XPT2046](#), and [AnalogTouch](#).

4.53.2.5 virtual void Touch::setRotation (uint8_t r) [virtual]

Set rotation

This sets the screen orientation of the touch screen. It should be set to the same as the rotation used for the screen.

Reimplemented in [XPT2046](#), and [AnalogTouch](#).

4.53.2.6 `virtual uint16_t Touch::x () [pure virtual]`

Get X coordinate

This returns the X coordinate of the current touch position.

Implemented in [XPT2046](#), and [AnalogTouch](#).

4.53.2.7 `virtual uint16_t Touch::y () [pure virtual]`

Get Y coordinate

This returns the Y coordinate of the current touch position.

Implemented in [XPT2046](#), and [AnalogTouch](#).

4.53.3 Member Data Documentation

4.53.3.1 `TFTCommunicator* Touch::_comm [protected]`

The communication device used to communicate with the touch screen controller (if any)

4.53.3.2 `uint16_t Touch::_height [protected]`

The height of the touch screen in pixels

4.53.3.3 `uint16_t Touch::_width [protected]`

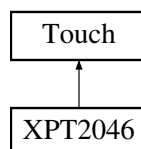
The width of the touch screen in pixels

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

- Touch.h

4.54 XPT2046 Class Reference

Inheritance diagram for XPT2046:



Public Member Functions

- void [sample](#) ()
- uint16_t [x](#) ()
- uint16_t [y](#) ()
- boolean [isPressed](#) ()
- void [initializeDevice](#) ()
- void [setRotation](#) (uint8_t r)

- [XPT2046](#) ([TFTCommunicator](#) *comm, uint16_t w, uint16_t h)
- [XPT2046](#) ([TFTCommunicator](#) &comm, uint16_t w, uint16_t h)

Additional Inherited Members

4.54.1 Constructor & Destructor Documentation

4.54.1.1 `XPT2046::XPT2046 (TFTCommunicator * comm, uint16_t w, uint16_t h)` `[inline]`

Create a new [XPT2046](#) object

This creates a new [XPT2046](#) touchscreen object. It requires an SPI compatible [TFTCommunicator](#) driver to be passed either as a pointer or as a reference. It also requires the width and height (natural orientation) of the touch screen.

Example:

```
XPT2046 ts(spiDev, 240, 320);
```

4.54.2 Member Function Documentation

4.54.2.1 `void XPT2046::initializeDevice ()` `[virtual]`

Initialize the device

This configures and enables the touch screen device. It should be called before any other touch screen functions.

Implements [Touch](#).

4.54.2.2 `boolean XPT2046::isPressed ()` `[virtual]`

Get pressed status

Returns true if the touch screen is pressed, false otherwise.

Implements [Touch](#).

4.54.2.3 `void XPT2046::sample ()` `[virtual]`

Sample the touch screen

This performs a sampling of the touch screen to get the current coordinates and touch status. It should be called regularly to update the current touch screen data.

Implements [Touch](#).

4.54.2.4 `void XPT2046::setRotation (uint8_t r)` `[virtual]`

Set rotation

This sets the screen orientation of the touch screen. It should be set to the same as the rotation used for the screen.

Reimplemented from [Touch](#).

4.54.2.5 `uint16_t XPT2046::x () [virtual]`

Get X coordinate

This returns the X coordinate of the current touch position.

Implements [Touch](#).

4.54.2.6 `uint16_t XPT2046::y () [virtual]`

Get Y coordinate

This returns the Y coordinate of the current touch position.

Implements [Touch](#).

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

- XPT2046.h
- XPT2046.cpp

Index

- [__attribute__](#), [7](#)
 - [_comm](#)
 - [TFT](#), [95](#)
 - [Touch](#), [129](#)
 - [_height](#)
 - [TFT](#), [95](#)
 - [Touch](#), [129](#)
 - [_width](#)
 - [TFT](#), [95](#)
 - [Touch](#), [129](#)
- [addDisplay](#)
 - [Aggregator](#), [8](#)
- [Aggregator](#), [7](#)
 - [addDisplay](#), [8](#)
 - [displayOff](#), [8](#)
 - [displayOn](#), [8](#)
 - [drawHorizontalLine](#), [8](#)
 - [drawVerticalLine](#), [9](#)
 - [fillScreen](#), [9](#)
 - [getHeight](#), [9](#)
 - [getWidth](#), [9](#)
 - [initializeDevice](#), [9](#)
 - [invertDisplay](#), [10](#)
 - [setPixel](#), [10](#)
 - [setRotation](#), [10](#)
- [AggregatorList](#), [10](#)
- [AnalogTouch](#), [11](#)
 - [initializeDevice](#), [11](#)
 - [isPressed](#), [11](#)
 - [pressure](#), [11](#)
 - [sample](#), [12](#)
 - [setRotation](#), [12](#)
 - [x](#), [12](#)
 - [y](#), [12](#)
- [animdir](#)
 - [sprite](#), [68](#)
- [BD663474](#), [12](#)
 - [displayOff](#), [13](#)
 - [displayOn](#), [13](#)
 - [drawHorizontalLine](#), [13](#)
 - [drawVerticalLine](#), [14](#)
 - [fillRectangle](#), [14](#)
 - [fillScreen](#), [14](#)
 - [initializeDevice](#), [14](#)
 - [invertDisplay](#), [15](#)
 - [setPixel](#), [15](#)
 - [setRotation](#), [15](#)
- [BMP](#), [17](#)
- [bgColorAt](#)
 - [Framebuffer](#), [26](#)
 - [Framebuffer332](#), [31](#)
 - [Framebuffer332Fast](#), [33](#)
 - [Framebuffer565](#), [35](#)
 - [TFT](#), [83](#)
- [BitmapFileHeader](#), [15](#)
- [BitmapInfoHeader](#), [16](#)
- [BitmapPixel24](#), [16](#)
- [BitmapPixel32](#), [16](#)
- [BlackTab](#)
 - [ST7735](#), [79](#)
- [blockData](#)
 - [TFTCommunicator](#), [97](#)
 - [TFTDSPI](#), [101](#), [102](#)
 - [TFTPar16](#), [106](#)
 - [TFTPar8](#), [114](#)
 - [TFTPMP](#), [118](#), [119](#)
 - [TFTSoftSPI](#), [123](#)
- [closeWindow](#)
 - [HX8357](#), [40](#)
 - [PICadillo35t](#), [55](#)
 - [S6D0164](#), [63](#)
 - [SSD1289](#), [70](#)
 - [TFT](#), [83](#)
- [Color](#), [17](#)
- [color565](#)
 - [TFT](#), [84](#)
- [colorAt](#)
 - [Framebuffer](#), [26](#)
 - [Framebuffer1](#), [30](#)
 - [Framebuffer332](#), [32](#)
 - [Framebuffer332Fast](#), [33](#)
 - [Framebuffer565](#), [35](#)
 - [PICadillo35t](#), [55](#)
 - [TFT](#), [84](#)
- [coord](#), [22](#)
- [CoreIO](#), [22](#)
- [currentframe](#)
 - [sprite](#), [68](#)
- [cursor_x](#)
 - [TFT](#), [95](#)
- [cursor_y](#)
 - [TFT](#), [95](#)
- [DOGMe](#), [23](#)
 - [initializeDevice](#), [24](#)
- [data](#)
 - [sprite](#), [68](#)

- DataBlock, [22](#)
- DataStore, [23](#)
- deltaE
 - TFT, [84](#)
- deltaOrth
 - TFT, [84](#)
- displayOff
 - Aggregator, [8](#)
 - BD663474, [13](#)
 - Framebuffer, [27](#)
 - HD44780, [37](#)
 - HX8357, [41](#)
 - ILI9340, [44](#)
 - KS0108, [48](#)
 - LEDMatrix, [51](#)
 - PICadillo35t, [55](#)
 - S6D0164, [63](#)
 - SSD1289, [70](#)
 - SSD1963, [74](#)
 - ST7735, [77](#)
 - TFT, [84](#)
- displayOn
 - Aggregator, [8](#)
 - BD663474, [13](#)
 - Framebuffer, [27](#)
 - HD44780, [37](#)
 - HX8357, [41](#)
 - ILI9340, [44](#)
 - KS0108, [48](#)
 - LEDMatrix, [51](#)
 - PICadillo35t, [55](#)
 - S6D0164, [64](#)
 - SSD1289, [70](#)
 - SSD1963, [74](#)
 - ST7735, [77](#)
 - TFT, [85](#)
- drawBitmap
 - TFT, [85](#)
- drawChar
 - TFT, [85](#)
- drawCircle
 - TFT, [85](#)
- drawCircleHelper
 - TFT, [86](#)
- drawHorizontalLine
 - Aggregator, [8](#)
 - BD663474, [13](#)
 - Framebuffer, [27](#)
 - Framebuffer332Fast, [34](#)
 - HD44780, [38](#)
 - HX8357, [41](#)
 - ILI9340, [44](#)
 - KS0108, [48](#)
 - PICadillo35t, [56](#)
 - S6D0164, [64](#)
 - SSD1289, [71](#)
 - SSD1963, [74](#)
 - ST7735, [78](#)
- TFT, [86](#)
- drawLine
 - TFT, [86](#)
- drawRGB
 - TFT, [86](#)
- drawRGBA
 - TFT, [87](#)
- drawRectangle
 - TFT, [86](#)
- drawRoundRect
 - TFT, [87](#)
- drawTriangle
 - TFT, [87](#)
- drawVerticalLine
 - Aggregator, [9](#)
 - BD663474, [14](#)
 - Framebuffer, [27](#)
 - HD44780, [38](#)
 - HX8357, [41](#)
 - ILI9340, [45](#)
 - KS0108, [48](#)
 - PICadillo35t, [56](#)
 - S6D0164, [64](#)
 - SSD1289, [71](#)
 - SSD1963, [74](#)
 - ST7735, [78](#)
 - TFT, [87](#)
- fatalError
 - TFT, [87](#)
- fillCircle
 - TFT, [88](#)
- fillCircleHelper
 - TFT, [88](#)
- fillRectangle
 - BD663474, [14](#)
 - HD44780, [38](#)
 - HX8357, [41](#)
 - ILI9340, [45](#)
 - KS0108, [49](#)
 - PICadillo35t, [56](#)
 - S6D0164, [64](#)
 - SSD1289, [71](#)
 - SSD1963, [75](#)
 - ST7735, [78](#)
 - TFT, [88](#)
- fillRoundRect
 - TFT, [88](#)
- fillScreen
 - Aggregator, [9](#)
 - BD663474, [14](#)
 - Framebuffer, [28](#)
 - Framebuffer1, [30](#)
 - Framebuffer332, [32](#)
 - Framebuffer332Fast, [34](#)
 - Framebuffer565, [36](#)
 - HD44780, [38](#)
 - HX8357, [42](#)
 - ILI9340, [45](#)

- KS0108, [49](#)
- LEDMatrix, [51](#)
- PICadillo35t, [56](#)
- S6D0164, [65](#)
- SSD1289, [71](#)
- SSD1963, [75](#)
- ST7735, [78](#)
- TFT, [88](#)
- fillTriangle
 - TFT, [89](#)
- font
 - TFT, [95](#)
- font_scale_x
 - TFT, [95](#)
- font_scale_y
 - TFT, [95](#)
- FontHeader, [24](#)
- Framebuffer, [25](#)
 - bgColorAt, [26](#)
 - colorAt, [26](#)
 - displayOff, [27](#)
 - displayOn, [27](#)
 - drawHorizontalLine, [27](#)
 - drawVerticalLine, [27](#)
 - fillScreen, [28](#)
 - getHeight, [28](#)
 - getWidth, [28](#)
 - initializeDevice, [28](#)
 - invertDisplay, [28](#)
 - setPixel, [29](#)
 - setRotation, [29](#)
- Framebuffer1, [29](#)
 - colorAt, [30](#)
 - fillScreen, [30](#)
 - initializeDevice, [30](#)
 - setPixel, [30](#)
- Framebuffer332, [31](#)
 - bgColorAt, [31](#)
 - colorAt, [32](#)
 - fillScreen, [32](#)
 - initializeDevice, [32](#)
 - setPixel, [32](#)
- Framebuffer332Fast, [33](#)
 - bgColorAt, [33](#)
 - colorAt, [33](#)
 - drawHorizontalLine, [34](#)
 - fillScreen, [34](#)
 - initializeDevice, [34](#)
 - setPixel, [34](#)
- Framebuffer565, [35](#)
 - bgColorAt, [35](#)
 - colorAt, [35](#)
 - fillScreen, [36](#)
 - initializeDevice, [36](#)
 - setPixel, [36](#)
- frames
 - sprite, [68](#)
- getCursor
 - TFT, [89](#)
- getCursorX
 - TFT, [89](#)
- getCursorY
 - TFT, [89](#)
- getHeight
 - Aggregator, [9](#)
 - Framebuffer, [28](#)
 - TFT, [89](#)
- getTextColor
 - TFT, [90](#)
- getWidth
 - Aggregator, [9](#)
 - Framebuffer, [28](#)
 - TFT, [90](#)
- GreenTab
 - ST7735, [79](#)
- HD44780, [36](#)
 - displayOff, [37](#)
 - displayOn, [37](#)
 - drawHorizontalLine, [38](#)
 - drawVerticalLine, [38](#)
 - fillRectangle, [38](#)
 - fillScreen, [38](#)
 - initializeDevice, [39](#)
 - invertDisplay, [39](#)
 - setPixel, [39](#)
 - setRotation, [39](#)
- HX8357, [40](#)
 - closeWindow, [40](#)
 - displayOff, [41](#)
 - displayOn, [41](#)
 - drawHorizontalLine, [41](#)
 - drawVerticalLine, [41](#)
 - fillRectangle, [41](#)
 - fillScreen, [42](#)
 - initializeDevice, [42](#)
 - invertDisplay, [42](#)
 - openWindow, [42](#)
 - setPixel, [42](#)
 - setRotation, [43](#)
 - windowData, [43](#)
- Height
 - SSD1963, [76](#)
 - ST7735, [80](#)
- height
 - sprite, [68](#)
- ILI9340, [43](#)
 - displayOff, [44](#)
 - displayOn, [44](#)
 - drawHorizontalLine, [44](#)
 - drawVerticalLine, [45](#)
 - fillRectangle, [45](#)
 - fillScreen, [45](#)
 - initializeDevice, [45](#)
 - invertDisplay, [46](#)
 - setPixel, [46](#)

- setRotation, 46
- Image, 46
- initializeDevice
 - Aggregator, 9
 - AnalogTouch, 11
 - BD663474, 14
 - DOGMe, 24
 - Framebuffer, 28
 - Framebuffer1, 30
 - Framebuffer332, 32
 - Framebuffer332Fast, 34
 - Framebuffer565, 36
 - HD44780, 39
 - HX8357, 42
 - ILI9340, 45
 - KS0108, 49
 - LEDMatrix, 51
 - PICadillo35t, 57
 - S6D0164, 65
 - SSD1289, 72
 - SSD1963, 75
 - ST7735, 79
 - TFT, 90
 - TFTCommunicator, 97
 - TFTDSPI, 102
 - TFTPar16, 107
 - TFTPar8, 114
 - TFTPMP, 119
 - TFTSoftSPI, 123
 - Touch, 128
 - XPT2046, 130
- invertDisplay
 - Aggregator, 10
 - BD663474, 15
 - Framebuffer, 28
 - HD44780, 39
 - HX8357, 42
 - ILI9340, 46
 - KS0108, 49
 - LEDMatrix, 52
 - PICadillo35t, 57
 - S6D0164, 65
 - SSD1289, 72
 - SSD1963, 75
 - ST7735, 79
 - TFT, 90
- invertTextColor
 - TFT, 90
- isPressed
 - AnalogTouch, 11
 - Touch, 128
 - XPT2046, 130
- IteadAdapter
 - TFTPar16, 110
- KS0108, 47
 - displayOff, 48
 - displayOn, 48
 - drawHorizontalLine, 48
 - drawVerticalLine, 48
 - fillRectangle, 49
 - fillScreen, 49
 - initializeDevice, 49
 - invertDisplay, 49
 - setPixel, 50
 - setRotation, 50
- LEDMatrix, 50
 - displayOff, 51
 - displayOn, 51
 - fillScreen, 51
 - initializeDevice, 51
 - invertDisplay, 52
 - setPixel, 52
 - setRotation, 52
- MCP23S17, 53
- MatrixISRList, 52
- mix
 - TFT, 91
- nativeWidth
 - RawPar, 60
 - TFTCommunicator, 97
 - TFTDSPI, 102
 - TFTPar16, 107
 - TFTPar4, 111
 - TFTPar8, 114
 - TFTPMP, 119
 - TFTSoftSPI, 124
- next
 - sprite, 68
- openWindow
 - HX8357, 42
 - PICadillo35t, 57
 - S6D0164, 65
 - SSD1289, 72
 - TFT, 91
- PICadillo35t, 54
 - closeWindow, 55
 - colorAt, 55
 - displayOff, 55
 - displayOn, 55
 - drawHorizontalLine, 56
 - drawVerticalLine, 56
 - fillRectangle, 56
 - fillScreen, 56
 - initializeDevice, 57
 - invertDisplay, 57
 - openWindow, 57
 - setPixel, 57
 - setRotation, 57
 - windowData, 58
- ParallelIO, 53
- point3d, 58
- pressure

- AnalogTouch, 11
- Touch, 128
- RLE, 62
- Raw565, 59
- Raw8, 59
- RawPar, 60
 - nativeWidth, 60
 - streamCommand16, 60
 - streamCommand32, 60
 - streamCommand8, 60
 - streamData16, 61
 - streamData32, 61
 - streamData8, 61
 - streamEnd, 61
 - streamStart, 61
 - writeCommand16, 61
 - writeCommand32, 61
 - writeCommand8, 61
 - writeData16, 61
 - writeData32, 62
 - writeData8, 62
- readCommand16
 - TFTCommunicator, 97
 - TFTDSPI, 102
 - TFTPar16, 107
 - TFTPar8, 115
 - TFTPMP, 119
 - TFTSoftSPI, 124
- readCommand32
 - TFTCommunicator, 97
 - TFTDSPI, 102
 - TFTPar16, 107
 - TFTPar8, 115
 - TFTPMP, 119
 - TFTSoftSPI, 124
- readCommand8
 - TFTCommunicator, 97
 - TFTDSPI, 102
 - TFTPar16, 107
 - TFTPar8, 115
 - TFTPMP, 119
 - TFTSoftSPI, 124
- readData16
 - TFTCommunicator, 98
 - TFTDSPI, 102
 - TFTPar16, 107
 - TFTPar8, 115
 - TFTPMP, 119
 - TFTSoftSPI, 124
- readData32
 - TFTCommunicator, 98
 - TFTDSPI, 102
 - TFTPar16, 107
 - TFTPar8, 115
 - TFTPMP, 119
 - TFTSoftSPI, 124
- readData8
 - TFTCommunicator, 98
- TFTDSPI, 102
- TFTPar16, 107
- TFTPar8, 115
- TFTPMP, 119
- TFTSoftSPI, 124
- RedTab
 - ST7735, 80
- rgb2hsv
 - TFT, 91
- rgb2xyz
 - TFT, 91
- rotation
 - TFT, 95
- S6D0164, 62
 - closeWindow, 63
 - displayOff, 63
 - displayOn, 64
 - drawHorizontalLine, 64
 - drawVerticalLine, 64
 - fillRectangle, 64
 - fillScreen, 65
 - initializeDevice, 65
 - invertDisplay, 65
 - openWindow, 65
 - setPixel, 65
 - setRotation, 66
 - windowData, 66
- SPISRAM, 66
- SRAM, 69
- SSD1289, 69
 - closeWindow, 70
 - displayOff, 70
 - displayOn, 70
 - drawHorizontalLine, 71
 - drawVerticalLine, 71
 - fillRectangle, 71
 - fillScreen, 71
 - initializeDevice, 72
 - invertDisplay, 72
 - openWindow, 72
 - setPixel, 72
 - setRotation, 72
 - windowData, 73
- SSD1963, 73
 - displayOff, 74
 - displayOn, 74
 - drawHorizontalLine, 74
 - drawVerticalLine, 74
 - fillRectangle, 75
 - fillScreen, 75
 - Height, 76
 - initializeDevice, 75
 - invertDisplay, 75
 - setPixel, 75
 - setRotation, 76
 - Width, 76
- ST7735, 76
 - BlackTab, 79

- displayOff, [77](#)
- displayOn, [77](#)
- drawHorizontalLine, [78](#)
- drawVerticalLine, [78](#)
- fillRectangle, [78](#)
- fillScreen, [78](#)
- GreenTab, [79](#)
- Height, [80](#)
- initializeDevice, [79](#)
- invertDisplay, [79](#)
- RedTab, [80](#)
- ST7735, [77](#)
- setPixel, [79](#)
- setRotation, [79](#)
- ST7735, [77](#)
- TypeB, [80](#)
- Width, [80](#)
- sample
 - AnalogTouch, [12](#)
 - Touch, [128](#)
 - XPT2046, [130](#)
- setCursor
 - TFT, [91](#)
- setFont
 - TFT, [92](#)
- setFontScaleX
 - TFT, [92](#)
- setFontScaleY
 - TFT, [92](#)
- setPixel
 - Aggregator, [10](#)
 - BD663474, [15](#)
 - Framebuffer, [29](#)
 - Framebuffer1, [30](#)
 - Framebuffer332, [32](#)
 - Framebuffer332Fast, [34](#)
 - Framebuffer565, [36](#)
 - HD44780, [39](#)
 - HX8357, [42](#)
 - ILI9340, [46](#)
 - KS0108, [50](#)
 - LEDMatrix, [52](#)
 - PICadillo35t, [57](#)
 - S6D0164, [65](#)
 - SSD1289, [72](#)
 - SSD1963, [75](#)
 - ST7735, [79](#)
 - TFT, [92](#)
- setRotation
 - Aggregator, [10](#)
 - AnalogTouch, [12](#)
 - BD663474, [15](#)
 - Framebuffer, [29](#)
 - HD44780, [39](#)
 - HX8357, [43](#)
 - ILI9340, [46](#)
 - KS0108, [50](#)
 - LEDMatrix, [52](#)
 - PICadillo35t, [57](#)
 - S6D0164, [66](#)
 - SSD1289, [72](#)
 - SSD1963, [76](#)
 - ST7735, [79](#)
 - TFT, [92](#)
 - Touch, [128](#)
 - XPT2046, [130](#)
- setTextColor
 - TFT, [93](#)
- setTextWrap
 - TFT, [93](#)
- sprite, [67](#)
 - animdir, [68](#)
 - currentframe, [68](#)
 - data, [68](#)
 - frames, [68](#)
 - height, [68](#)
 - next, [68](#)
 - store, [68](#)
 - transparent, [68](#)
 - width, [68](#)
 - xpos, [68](#)
 - ypos, [68](#)
- store
 - sprite, [68](#)
- streamCommand16
 - RawPar, [60](#)
 - TFTCommunicator, [98](#)
 - TFTDSPI, [103](#)
 - TFTPar16, [107](#), [108](#)
 - TFTPar4, [111](#)
 - TFTPar8, [115](#)
 - TFTPMP, [120](#)
 - TFTSoftSPI, [124](#)
- streamCommand32
 - RawPar, [60](#)
 - TFTCommunicator, [98](#)
 - TFTDSPI, [103](#)
 - TFTPar16, [108](#)
 - TFTPar4, [111](#)
 - TFTPar8, [115](#), [116](#)
 - TFTPMP, [120](#)
 - TFTSoftSPI, [125](#)
- streamCommand8
 - RawPar, [60](#)
 - TFTCommunicator, [98](#)
 - TFTDSPI, [103](#)
 - TFTPar16, [108](#)
 - TFTPar4, [111](#)
 - TFTPar8, [116](#)
 - TFTPMP, [120](#)
 - TFTSoftSPI, [125](#)
- streamData16
 - RawPar, [61](#)
 - TFTCommunicator, [99](#)
 - TFTDSPI, [103](#)
 - TFTPar16, [108](#)

- TFTPar4, 111
- TFTPar8, 116
- TFTPMP, 120
- TFTSoftSPI, 125
- streamData32
 - RawPar, 61
 - TFTCommunicator, 99
 - TFTDSPI, 103, 104
 - TFTPar16, 108
 - TFTPar4, 112
 - TFTPar8, 116
 - TFTPMP, 120, 121
 - TFTSoftSPI, 125
- streamData8
 - RawPar, 61
 - TFTCommunicator, 99
 - TFTDSPI, 104
 - TFTPar16, 109
 - TFTPar4, 112
 - TFTPar8, 116
 - TFTPMP, 121
 - TFTSoftSPI, 125, 126
- streamEnd
 - RawPar, 61
 - TFTCommunicator, 99
 - TFTDSPI, 104
 - TFTPar16, 109
 - TFTPar4, 112
 - TFTPar8, 117
 - TFTPMP, 121
 - TFTSoftSPI, 126
- streamStart
 - RawPar, 61
 - TFTCommunicator, 99
 - TFTDSPI, 104
 - TFTPar16, 109
 - TFTPar4, 112
 - TFTPar8, 117
 - TFTPMP, 121
 - TFTSoftSPI, 126
- stringHeight
 - TFT, 93
- stringWidth
 - TFT, 93
- TFT, 80
 - _comm, 95
 - _height, 95
 - _width, 95
 - bgColorAt, 83
 - closeWindow, 83
 - color565, 84
 - colorAt, 84
 - cursor_x, 95
 - cursor_y, 95
 - deltaE, 84
 - deltaOrth, 84
 - displayOff, 84
 - displayOn, 85
 - drawBitmap, 85
 - drawChar, 85
 - drawCircle, 85
 - drawCircleHelper, 86
 - drawHorizontalLine, 86
 - drawLine, 86
 - drawRGB, 86
 - drawRGBA, 87
 - drawRectangle, 86
 - drawRoundRect, 87
 - drawTriangle, 87
 - drawVerticalLine, 87
 - fatalError, 87
 - fillCircle, 88
 - fillCircleHelper, 88
 - fillRectangle, 88
 - fillRoundRect, 88
 - fillScreen, 88
 - fillTriangle, 89
 - font, 95
 - font_scale_x, 95
 - font_scale_y, 95
 - getCursor, 89
 - getCursorX, 89
 - getCursorY, 89
 - getHeight, 89
 - getTextColor, 90
 - getWidth, 90
 - initializeDevice, 90
 - invertDisplay, 90
 - invertTextColor, 90
 - mix, 91
 - openWindow, 91
 - rgb2hsv, 91
 - rgb2xyz, 91
 - rotation, 95
 - setCursor, 91
 - setFont, 92
 - setFontScaleX, 92
 - setFontScaleY, 92
 - setPixel, 92
 - setRotation, 92
 - setTextColor, 93
 - setTextWrap, 93
 - stringHeight, 93
 - stringWidth, 93
 - TFT, 83
 - textbgcolor, 95
 - textcolor, 95
 - TFT, 83
 - windowData, 94
 - wrap, 95
 - write, 94
 - xyz2lab, 94
- TFTCommunicator, 96
 - blockData, 97
 - initializeDevice, 97
 - nativeWidth, 97

- readCommand16, [97](#)
- readCommand32, [97](#)
- readCommand8, [97](#)
- readData16, [98](#)
- readData32, [98](#)
- readData8, [98](#)
- streamCommand16, [98](#)
- streamCommand32, [98](#)
- streamCommand8, [98](#)
- streamData16, [99](#)
- streamData32, [99](#)
- streamData8, [99](#)
- streamEnd, [99](#)
- streamStart, [99](#)
- writeCommand16, [99](#)
- writeCommand32, [100](#)
- writeCommand8, [100](#)
- writeData16, [100](#)
- writeData32, [100](#)
- writeData8, [100](#)
- TFTDSPI, [100](#)
 - blockData, [101](#), [102](#)
 - initializeDevice, [102](#)
 - nativeWidth, [102](#)
 - readCommand16, [102](#)
 - readCommand32, [102](#)
 - readCommand8, [102](#)
 - readData16, [102](#)
 - readData32, [102](#)
 - readData8, [102](#)
 - streamCommand16, [103](#)
 - streamCommand32, [103](#)
 - streamCommand8, [103](#)
 - streamData16, [103](#)
 - streamData32, [103](#), [104](#)
 - streamData8, [104](#)
 - streamEnd, [104](#)
 - streamStart, [104](#)
 - TFTDSPI, [101](#)
 - TFTDSPI, [101](#)
 - writeCommand16, [104](#)
 - writeCommand32, [104](#)
 - writeCommand8, [104](#)
 - writeData16, [104](#)
 - writeData32, [105](#)
 - writeData8, [105](#)
- TFTPMP, [118](#)
 - blockData, [118](#), [119](#)
 - initializeDevice, [119](#)
 - nativeWidth, [119](#)
 - readCommand16, [119](#)
 - readCommand32, [119](#)
 - readCommand8, [119](#)
 - readData16, [119](#)
 - readData32, [119](#)
 - readData8, [119](#)
 - streamCommand16, [120](#)
 - streamCommand32, [120](#)
 - streamCommand8, [120](#)
 - streamData16, [120](#)
 - streamData32, [120](#), [121](#)
 - streamData8, [121](#)
 - streamEnd, [121](#)
 - streamStart, [121](#)
 - writeCommand16, [121](#)
 - writeCommand32, [121](#)
 - writeCommand8, [121](#)
 - writeData16, [121](#)
 - writeData32, [122](#)
 - writeData8, [122](#)
- TFTPar16, [105](#)
 - blockData, [106](#)
 - initializeDevice, [107](#)
 - ltheadAdapter, [110](#)
 - nativeWidth, [107](#)
 - readCommand16, [107](#)
 - readCommand32, [107](#)
 - readCommand8, [107](#)
 - readData16, [107](#)
 - readData32, [107](#)
 - readData8, [107](#)
 - streamCommand16, [107](#), [108](#)
 - streamCommand32, [108](#)
 - streamCommand8, [108](#)
 - streamData16, [108](#)
 - streamData32, [108](#)
 - streamData8, [109](#)
 - streamEnd, [109](#)
 - streamStart, [109](#)
 - TFTPar16, [106](#)
 - TFTPar16, [106](#)
 - writeCommand16, [109](#)
 - writeCommand32, [109](#)
 - writeCommand8, [109](#)
 - writeData16, [109](#)
 - writeData32, [109](#)
 - writeData8, [110](#)
- TFTPar4, [110](#)
 - nativeWidth, [111](#)
 - streamCommand16, [111](#)
 - streamCommand32, [111](#)
 - streamCommand8, [111](#)
 - streamData16, [111](#)
 - streamData32, [112](#)
 - streamData8, [112](#)
 - streamEnd, [112](#)
 - streamStart, [112](#)
 - TFTPar4, [111](#)
 - TFTPar4, [111](#)
 - writeCommand16, [112](#)
 - writeCommand32, [112](#)
 - writeCommand8, [112](#)
 - writeData16, [112](#)
 - writeData32, [112](#)
 - writeData8, [113](#)
- TFTPar8, [113](#)

- blockData, 114
- initializeDevice, 114
- nativeWidth, 114
- readCommand16, 115
- readCommand32, 115
- readCommand8, 115
- readData16, 115
- readData32, 115
- readData8, 115
- streamCommand16, 115
- streamCommand32, 115, 116
- streamCommand8, 116
- streamData16, 116
- streamData32, 116
- streamData8, 116
- streamEnd, 117
- streamStart, 117
- TFTPar8, 114
- TFTPar8, 114
- writeCommand16, 117
- writeCommand32, 117
- writeCommand8, 117
- writeData16, 117
- writeData32, 117
- writeData8, 117
- TFTSoftSPI, 122
 - blockData, 123
 - initializeDevice, 123
 - nativeWidth, 124
 - readCommand16, 124
 - readCommand32, 124
 - readCommand8, 124
 - readData16, 124
 - readData32, 124
 - readData8, 124
 - streamCommand16, 124
 - streamCommand32, 125
 - streamCommand8, 125
 - streamData16, 125
 - streamData32, 125
 - streamData8, 125, 126
 - streamEnd, 126
 - streamStart, 126
 - TFTSoftSPI, 123
 - TFTSoftSPI, 123
 - writeCommand16, 126
 - writeCommand32, 126
 - writeCommand8, 126
 - writeData16, 126
 - writeData32, 126
 - writeData8, 126
- textbgcolor
 - TFT, 95
- textcolor
 - TFT, 95
- Touch, 127
 - _comm, 129
 - _height, 129
 - _width, 129
 - initializeDevice, 128
 - isPressed, 128
 - pressure, 128
 - sample, 128
 - setRotation, 128
 - Touch, 127
 - x, 128
 - y, 129
- transparent
 - sprite, 68
- TypeB
 - ST7735, 80
- Width
 - SSD1963, 76
 - ST7735, 80
- width
 - sprite, 68
- windowData
 - HX8357, 43
 - PICadillo35t, 58
 - S6D0164, 66
 - SSD1289, 73
 - TFT, 94
- wrap
 - TFT, 95
- write
 - TFT, 94
- writeCommand16
 - RawPar, 61
 - TFTCommunicator, 99
 - TFTDSPI, 104
 - TFTPar16, 109
 - TFTPar4, 112
 - TFTPar8, 117
 - TFTPMP, 121
 - TFTSoftSPI, 126
- writeCommand32
 - RawPar, 61
 - TFTCommunicator, 100
 - TFTDSPI, 104
 - TFTPar16, 109
 - TFTPar4, 112
 - TFTPar8, 117
 - TFTPMP, 121
 - TFTSoftSPI, 126
- writeCommand8
 - RawPar, 61
 - TFTCommunicator, 100
 - TFTDSPI, 104
 - TFTPar16, 109
 - TFTPar4, 112
 - TFTPar8, 117
 - TFTPMP, 121
 - TFTSoftSPI, 126
- writeData16
 - RawPar, 61
 - TFTCommunicator, 100

- TFTDSPI, [104](#)
- TFTPar16, [109](#)
- TFTPar4, [112](#)
- TFTPar8, [117](#)
- TFTPMP, [121](#)
- TFTSoftSPI, [126](#)
- writeData32
 - RawPar, [62](#)
 - TFTCommunicator, [100](#)
 - TFTDSPI, [105](#)
 - TFTPar16, [109](#)
 - TFTPar4, [112](#)
 - TFTPar8, [117](#)
 - TFTPMP, [122](#)
 - TFTSoftSPI, [126](#)
- writeData8
 - RawPar, [62](#)
 - TFTCommunicator, [100](#)
 - TFTDSPI, [105](#)
 - TFTPar16, [110](#)
 - TFTPar4, [113](#)
 - TFTPar8, [117](#)
 - TFTPMP, [122](#)
 - TFTSoftSPI, [126](#)
- x
 - AnalogTouch, [12](#)
 - Touch, [128](#)
 - XPT2046, [130](#)
- XPT2046, [129](#)
 - initializeDevice, [130](#)
 - isPressed, [130](#)
 - sample, [130](#)
 - setRotation, [130](#)
 - x, [130](#)
 - XPT2046, [130](#)
 - XPT2046, [130](#)
 - y, [131](#)
- xpos
 - sprite, [68](#)
- xyz2lab
 - TFT, [94](#)
- y
 - AnalogTouch, [12](#)
 - Touch, [129](#)
 - XPT2046, [131](#)
- ypos
 - sprite, [68](#)