

## chipKIT TFT Library

Generated by Doxygen 1.8.4

Mon Jul 28 2014 10:17:39



# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>chipKIT-TFT</b>                        | <b>1</b> |
| <b>2</b> | <b>Hierarchical Index</b>                 | <b>3</b> |
| 2.1      | Class Hierarchy . . . . .                 | 3        |
| <b>3</b> | <b>Class Index</b>                        | <b>5</b> |
| 3.1      | Class List . . . . .                      | 5        |
| <b>4</b> | <b>Class Documentation</b>                | <b>7</b> |
| 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  | <a href="#">initializeDevice</a>              | 39 |
| 4.23.1.8  | <a href="#">invertDisplay</a>                 | 39 |
| 4.23.1.9  | <a href="#">setPixel</a>                      | 39 |
| 4.23.1.10 | <a href="#">setRotation</a>                   | 39 |
| 4.24      | <a href="#">HX8357 Class Reference</a>        | 40 |
| 4.24.1    | <a href="#">Member Function Documentation</a> | 40 |
| 4.24.1.1  | <a href="#">closeWindow</a>                   | 40 |
| 4.24.1.2  | <a href="#">displayOff</a>                    | 41 |
| 4.24.1.3  | <a href="#">displayOn</a>                     | 41 |
| 4.24.1.4  | <a href="#">drawHorizontalLine</a>            | 41 |
| 4.24.1.5  | <a href="#">drawVerticalLine</a>              | 41 |
| 4.24.1.6  | <a href="#">fillRectangle</a>                 | 41 |
| 4.24.1.7  | <a href="#">fillScreen</a>                    | 42 |
| 4.24.1.8  | <a href="#">initializeDevice</a>              | 42 |
| 4.24.1.9  | <a href="#">invertDisplay</a>                 | 42 |
| 4.24.1.10 | <a href="#">openWindow</a>                    | 42 |
| 4.24.1.11 | <a href="#">setPixel</a>                      | 43 |
| 4.24.1.12 | <a href="#">setRotation</a>                   | 43 |
| 4.24.1.13 | <a href="#">windowData</a>                    | 43 |
| 4.24.1.14 | <a href="#">windowData</a>                    | 43 |
| 4.25      | <a href="#">ILI9340 Class Reference</a>       | 43 |
| 4.25.1    | <a href="#">Member Function Documentation</a> | 44 |
| 4.25.1.1  | <a href="#">displayOff</a>                    | 44 |
| 4.25.1.2  | <a href="#">displayOn</a>                     | 44 |
| 4.25.1.3  | <a href="#">drawHorizontalLine</a>            | 45 |
| 4.25.1.4  | <a href="#">drawVerticalLine</a>              | 45 |
| 4.25.1.5  | <a href="#">fillRectangle</a>                 | 45 |
| 4.25.1.6  | <a href="#">fillScreen</a>                    | 45 |
| 4.25.1.7  | <a href="#">initializeDevice</a>              | 45 |
| 4.25.1.8  | <a href="#">invertDisplay</a>                 | 46 |
| 4.25.1.9  | <a href="#">setPixel</a>                      | 46 |
| 4.25.1.10 | <a href="#">setRotation</a>                   | 46 |
| 4.26      | <a href="#">Image Class Reference</a>         | 46 |
| 4.27      | <a href="#">KS0108 Class Reference</a>        | 47 |
| 4.27.1    | <a href="#">Member Function Documentation</a> | 48 |
| 4.27.1.1  | <a href="#">displayOff</a>                    | 48 |
| 4.27.1.2  | <a href="#">displayOn</a>                     | 48 |
| 4.27.1.3  | <a href="#">drawHorizontalLine</a>            | 48 |
| 4.27.1.4  | <a href="#">drawVerticalLine</a>              | 49 |
| 4.27.1.5  | <a href="#">fillRectangle</a>                 | 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      | point3d Struct Reference       | 54 |
| 4.33      | Raw565 Class Reference         | 54 |
| 4.34      | Raw8 Class Reference           | 54 |
| 4.35      | RawPar Class Reference         | 55 |
| 4.35.1    | Member Function Documentation  | 56 |
| 4.35.1.1  | nativeWidth                    | 56 |
| 4.35.1.2  | streamCommand16                | 56 |
| 4.35.1.3  | streamCommand32                | 56 |
| 4.35.1.4  | streamCommand8                 | 56 |
| 4.35.1.5  | streamData16                   | 56 |
| 4.35.1.6  | streamData32                   | 56 |
| 4.35.1.7  | streamData8                    | 56 |
| 4.35.1.8  | streamEnd                      | 56 |
| 4.35.1.9  | streamStart                    | 56 |
| 4.35.1.10 | writeCommand16                 | 57 |
| 4.35.1.11 | writeCommand32                 | 57 |
| 4.35.1.12 | writeCommand8                  | 57 |
| 4.35.1.13 | writeData16                    | 57 |
| 4.35.1.14 | writeData32                    | 57 |
| 4.35.1.15 | writeData8                     | 57 |
| 4.36      | RLE Class Reference            | 57 |
| 4.37      | S6D0164 Class Reference        | 58 |
| 4.37.1    | Member Function Documentation  | 59 |

|           |   |    |
|-----------|---|----|
| 4.37.1.1  | <a href="#">closeWindow</a>                   | 59 |
| 4.37.1.2  | <a href="#">displayOff</a>                    | 59 |
| 4.37.1.3  | <a href="#">displayOn</a>                     | 59 |
| 4.37.1.4  | <a href="#">drawHorizontalLine</a>            | 59 |
| 4.37.1.5  | <a href="#">drawVerticalLine</a>              | 59 |
| 4.37.1.6  | <a href="#">fillRectangle</a>                 | 60 |
| 4.37.1.7  | <a href="#">fillScreen</a>                    | 60 |
| 4.37.1.8  | <a href="#">initializeDevice</a>              | 60 |
| 4.37.1.9  | <a href="#">invertDisplay</a>                 | 60 |
| 4.37.1.10 | <a href="#">openWindow</a>                    | 61 |
| 4.37.1.11 | <a href="#">setPixel</a>                      | 61 |
| 4.37.1.12 | <a href="#">setRotation</a>                   | 61 |
| 4.37.1.13 | <a href="#">windowData</a>                    | 61 |
| 4.37.1.14 | <a href="#">windowData</a>                    | 61 |
| 4.38      | <a href="#">SPISRAM Class Reference</a>       | 62 |
| 4.39      | <a href="#">sprite Struct Reference</a>       | 62 |
| 4.39.1    | <a href="#">Detailed Description</a>          | 63 |
| 4.39.2    | <a href="#">Member Data Documentation</a>     | 63 |
| 4.39.2.1  | <a href="#">animdir</a>                       | 63 |
| 4.39.2.2  | <a href="#">currentframe</a>                  | 63 |
| 4.39.2.3  | <a href="#">data</a>                          | 63 |
| 4.39.2.4  | <a href="#">frames</a>                        | 63 |
| 4.39.2.5  | <a href="#">height</a>                        | 63 |
| 4.39.2.6  | <a href="#">next</a>                          | 63 |
| 4.39.2.7  | <a href="#">store</a>                         | 63 |
| 4.39.2.8  | <a href="#">transparent</a>                   | 63 |
| 4.39.2.9  | <a href="#">width</a>                         | 64 |
| 4.39.2.10 | <a href="#">xpos</a>                          | 64 |
| 4.39.2.11 | <a href="#">ypos</a>                          | 64 |
| 4.40      | <a href="#">SRAM Class Reference</a>          | 64 |
| 4.41      | <a href="#">SSD1289 Class Reference</a>       | 65 |
| 4.41.1    | <a href="#">Member Function Documentation</a> | 65 |
| 4.41.1.1  | <a href="#">closeWindow</a>                   | 65 |
| 4.41.1.2  | <a href="#">displayOff</a>                    | 66 |
| 4.41.1.3  | <a href="#">displayOn</a>                     | 66 |
| 4.41.1.4  | <a href="#">drawHorizontalLine</a>            | 66 |
| 4.41.1.5  | <a href="#">drawVerticalLine</a>              | 66 |
| 4.41.1.6  | <a href="#">fillRectangle</a>                 | 66 |
| 4.41.1.7  | <a href="#">fillScreen</a>                    | 67 |
| 4.41.1.8  | <a href="#">initializeDevice</a>              | 67 |



|           |  |    |
|-----------|--|----|
| 4.41.1.9  | invertDisplay                          | 67 |
| 4.41.1.10 | openWindow                             | 67 |
| 4.41.1.11 | setPixel                               | 67 |
| 4.41.1.12 | setRotation                            | 68 |
| 4.41.1.13 | windowData                             | 68 |
| 4.41.1.14 | windowData                             | 68 |
| 4.42      | SSD1963 Class Reference                | 68 |
| 4.42.1    | Member Function Documentation          | 69 |
| 4.42.1.1  | displayOff                             | 69 |
| 4.42.1.2  | displayOn                              | 69 |
| 4.42.1.3  | drawHorizontalLine                     | 69 |
| 4.42.1.4  | drawVerticalLine                       | 69 |
| 4.42.1.5  | fillRectangle                          | 70 |
| 4.42.1.6  | fillScreen                             | 70 |
| 4.42.1.7  | initializeDevice                       | 70 |
| 4.42.1.8  | invertDisplay                          | 70 |
| 4.42.1.9  | setPixel                               | 71 |
| 4.42.1.10 | setRotation                            | 71 |
| 4.42.2    | Member Data Documentation              | 71 |
| 4.42.2.1  | Height                                 | 71 |
| 4.42.2.2  | Width                                  | 71 |
| 4.43      | ST7735 Class Reference                 | 71 |
| 4.43.1    | Constructor & Destructor Documentation | 72 |
| 4.43.1.1  | ST7735                                 | 72 |
| 4.43.2    | Member Function Documentation          | 72 |
| 4.43.2.1  | displayOff                             | 72 |
| 4.43.2.2  | displayOn                              | 73 |
| 4.43.2.3  | drawHorizontalLine                     | 73 |
| 4.43.2.4  | drawVerticalLine                       | 73 |
| 4.43.2.5  | fillRectangle                          | 73 |
| 4.43.2.6  | fillScreen                             | 73 |
| 4.43.2.7  | initializeDevice                       | 74 |
| 4.43.2.8  | invertDisplay                          | 74 |
| 4.43.2.9  | setPixel                               | 74 |
| 4.43.2.10 | setRotation                            | 74 |
| 4.43.3    | Member Data Documentation              | 74 |
| 4.43.3.1  | BlackTab                               | 74 |
| 4.43.3.2  | GreenTab                               | 75 |
| 4.43.3.3  | Height                                 | 75 |
| 4.43.3.4  | RedTab                                 | 75 |

|           |  |    |
|-----------|--|----|
| 4.43.3.5  | TypeB                                  | 75 |
| 4.43.3.6  | Width                                  | 75 |
| 4.44      | TFT Class Reference                    | 75 |
| 4.44.1    | Detailed Description                   | 78 |
| 4.44.2    | Constructor & Destructor Documentation | 78 |
| 4.44.2.1  | TFT                                    | 78 |
| 4.44.2.2  | TFT                                    | 78 |
| 4.44.2.3  | TFT                                    | 78 |
| 4.44.3    | Member Function Documentation          | 78 |
| 4.44.3.1  | bgColorAt                              | 78 |
| 4.44.3.2  | clearClipping                          | 79 |
| 4.44.3.3  | closeWindow                            | 79 |
| 4.44.3.4  | color565                               | 79 |
| 4.44.3.5  | colorAt                                | 79 |
| 4.44.3.6  | deltaE                                 | 79 |
| 4.44.3.7  | deltaOrth                              | 80 |
| 4.44.3.8  | displayOff                             | 80 |
| 4.44.3.9  | displayOn                              | 80 |
| 4.44.3.10 | drawBitmap                             | 80 |
| 4.44.3.11 | drawChar                               | 81 |
| 4.44.3.12 | drawCircle                             | 81 |
| 4.44.3.13 | drawCircleHelper                       | 81 |
| 4.44.3.14 | drawHorizontalLine                     | 81 |
| 4.44.3.15 | drawLine                               | 81 |
| 4.44.3.16 | drawRectangle                          | 81 |
| 4.44.3.17 | drawRGB                                | 82 |
| 4.44.3.18 | drawRGBA                               | 82 |
| 4.44.3.19 | drawRoundRect                          | 82 |
| 4.44.3.20 | drawTriangle                           | 82 |
| 4.44.3.21 | drawVerticalLine                       | 83 |
| 4.44.3.22 | fatalError                             | 83 |
| 4.44.3.23 | fillCircle                             | 83 |
| 4.44.3.24 | fillCircleHelper                       | 83 |
| 4.44.3.25 | fillRectangle                          | 83 |
| 4.44.3.26 | fillRoundRect                          | 84 |
| 4.44.3.27 | fillScreen                             | 84 |
| 4.44.3.28 | fillTriangle                           | 84 |
| 4.44.3.29 | getCursor                              | 84 |
| 4.44.3.30 | getCursorX                             | 84 |
| 4.44.3.31 | getCursorY                             | 85 |

|           |                           |    |
|-----------|---------------------------|----|
| 4.44.3.32 | getHeight                 | 85 |
| 4.44.3.33 | getTextColor              | 85 |
| 4.44.3.34 | getWidth                  | 85 |
| 4.44.3.35 | initializeDevice          | 85 |
| 4.44.3.36 | invertDisplay             | 86 |
| 4.44.3.37 | invertTextColor           | 86 |
| 4.44.3.38 | mix                       | 86 |
| 4.44.3.39 | openWindow                | 86 |
| 4.44.3.40 | rgb2hsv                   | 86 |
| 4.44.3.41 | rgb2xyz                   | 87 |
| 4.44.3.42 | setClipping               | 87 |
| 4.44.3.43 | setCursor                 | 87 |
| 4.44.3.44 | setCursorX                | 87 |
| 4.44.3.45 | setCursorY                | 87 |
| 4.44.3.46 | setFont                   | 88 |
| 4.44.3.47 | setFontScaleX             | 88 |
| 4.44.3.48 | setFontScaleY             | 88 |
| 4.44.3.49 | setPixel                  | 88 |
| 4.44.3.50 | setRotation               | 88 |
| 4.44.3.51 | setTextColor              | 89 |
| 4.44.3.52 | setTextColor              | 89 |
| 4.44.3.53 | setTextWrap               | 89 |
| 4.44.3.54 | stringHeight              | 89 |
| 4.44.3.55 | stringWidth               | 89 |
| 4.44.3.56 | windowData                | 90 |
| 4.44.3.57 | windowData                | 90 |
| 4.44.3.58 | write                     | 90 |
| 4.44.3.59 | xyz2lab                   | 90 |
| 4.44.4    | Member Data Documentation | 90 |
| 4.44.4.1  | _comm                     | 90 |
| 4.44.4.2  | _height                   | 90 |
| 4.44.4.3  | _width                    | 91 |
| 4.44.4.4  | cursor_x                  | 91 |
| 4.44.4.5  | cursor_y                  | 91 |
| 4.44.4.6  | font                      | 91 |
| 4.44.4.7  | font_scale_x              | 91 |
| 4.44.4.8  | font_scale_y              | 91 |
| 4.44.4.9  | rotation                  | 91 |
| 4.44.4.10 | textbgcolor               | 91 |
| 4.44.4.11 | textcolor                 | 91 |

|   |    |
|---|----|
| 4.44.4.12 wrap . . . . .                                | 91 |
| 4.45 TFTCommunicator Class Reference . . . . .          | 91 |
| 4.45.1 Detailed Description . . . . .                   | 92 |
| 4.45.2 Member Function Documentation . . . . .          | 93 |
| 4.45.2.1 blockData . . . . .                            | 93 |
| 4.45.2.2 blockData . . . . .                            | 93 |
| 4.45.2.3 blockData . . . . .                            | 93 |
| 4.45.2.4 initializeDevice . . . . .                     | 93 |
| 4.45.2.5 nativeWidth . . . . .                          | 93 |
| 4.45.2.6 readCommand16 . . . . .                        | 93 |
| 4.45.2.7 readCommand32 . . . . .                        | 93 |
| 4.45.2.8 readCommand8 . . . . .                         | 93 |
| 4.45.2.9 readData16 . . . . .                           | 93 |
| 4.45.2.10 readData32 . . . . .                          | 94 |
| 4.45.2.11 readData8 . . . . .                           | 94 |
| 4.45.2.12 streamCommand16 . . . . .                     | 94 |
| 4.45.2.13 streamCommand16 . . . . .                     | 94 |
| 4.45.2.14 streamCommand32 . . . . .                     | 94 |
| 4.45.2.15 streamCommand32 . . . . .                     | 94 |
| 4.45.2.16 streamCommand8 . . . . .                      | 94 |
| 4.45.2.17 streamCommand8 . . . . .                      | 94 |
| 4.45.2.18 streamData16 . . . . .                        | 94 |
| 4.45.2.19 streamData16 . . . . .                        | 95 |
| 4.45.2.20 streamData32 . . . . .                        | 95 |
| 4.45.2.21 streamData32 . . . . .                        | 95 |
| 4.45.2.22 streamData8 . . . . .                         | 95 |
| 4.45.2.23 streamData8 . . . . .                         | 95 |
| 4.45.2.24 streamEnd . . . . .                           | 95 |
| 4.45.2.25 streamStart . . . . .                         | 95 |
| 4.45.2.26 writeCommand16 . . . . .                      | 95 |
| 4.45.2.27 writeCommand32 . . . . .                      | 95 |
| 4.45.2.28 writeCommand8 . . . . .                       | 96 |
| 4.45.2.29 writeData16 . . . . .                         | 96 |
| 4.45.2.30 writeData32 . . . . .                         | 96 |
| 4.45.2.31 writeData8 . . . . .                          | 96 |
| 4.46 TFTPar16 Class Reference . . . . .                 | 96 |
| 4.46.1 Detailed Description . . . . .                   | 97 |
| 4.46.2 Constructor & Destructor Documentation . . . . . | 97 |
| 4.46.2.1 TFTPar16 . . . . .                             | 97 |
| 4.46.2.2 TFTPar16 . . . . .                             | 97 |

|           |  |     |
|-----------|--|-----|
| 4.46.3    | Member Function Documentation          | 97  |
| 4.46.3.1  | blockData                              | 97  |
| 4.46.3.2  | blockData                              | 98  |
| 4.46.3.3  | blockData                              | 98  |
| 4.46.3.4  | initializeDevice                       | 98  |
| 4.46.3.5  | nativeWidth                            | 98  |
| 4.46.3.6  | readCommand16                          | 98  |
| 4.46.3.7  | readCommand32                          | 98  |
| 4.46.3.8  | readCommand8                           | 98  |
| 4.46.3.9  | readData16                             | 98  |
| 4.46.3.10 | readData32                             | 98  |
| 4.46.3.11 | readData8                              | 99  |
| 4.46.3.12 | streamCommand16                        | 99  |
| 4.46.3.13 | streamCommand16                        | 99  |
| 4.46.3.14 | streamCommand32                        | 99  |
| 4.46.3.15 | streamCommand32                        | 99  |
| 4.46.3.16 | streamCommand8                         | 99  |
| 4.46.3.17 | streamCommand8                         | 99  |
| 4.46.3.18 | streamData16                           | 99  |
| 4.46.3.19 | streamData16                           | 99  |
| 4.46.3.20 | streamData32                           | 100 |
| 4.46.3.21 | streamData32                           | 100 |
| 4.46.3.22 | streamData8                            | 100 |
| 4.46.3.23 | streamData8                            | 100 |
| 4.46.3.24 | streamEnd                              | 100 |
| 4.46.3.25 | streamStart                            | 100 |
| 4.46.3.26 | writeCommand16                         | 100 |
| 4.46.3.27 | writeCommand32                         | 100 |
| 4.46.3.28 | writeCommand8                          | 100 |
| 4.46.3.29 | writeData16                            | 101 |
| 4.46.3.30 | writeData32                            | 101 |
| 4.46.3.31 | writeData8                             | 101 |
| 4.46.4    | Member Data Documentation              | 101 |
| 4.46.4.1  | IteadAdapter                           | 101 |
| 4.47      | TFTPar4 Class Reference                | 101 |
| 4.47.1    | Detailed Description                   | 102 |
| 4.47.2    | Constructor & Destructor Documentation | 102 |
| 4.47.2.1  | TFTPar4                                | 102 |
| 4.47.2.2  | TFTPar4                                | 102 |
| 4.47.3    | Member Function Documentation          | 102 |

|           |  |     |
|-----------|--|-----|
| 4.47.3.1  | <a href="#">nativeWidth</a>                                | 102 |
| 4.47.3.2  | <a href="#">streamCommand16</a>                            | 102 |
| 4.47.3.3  | <a href="#">streamCommand32</a>                            | 102 |
| 4.47.3.4  | <a href="#">streamCommand8</a>                             | 103 |
| 4.47.3.5  | <a href="#">streamData16</a>                               | 103 |
| 4.47.3.6  | <a href="#">streamData32</a>                               | 103 |
| 4.47.3.7  | <a href="#">streamData8</a>                                | 103 |
| 4.47.3.8  | <a href="#">streamEnd</a>                                  | 103 |
| 4.47.3.9  | <a href="#">streamStart</a>                                | 103 |
| 4.47.3.10 | <a href="#">writeCommand16</a>                             | 103 |
| 4.47.3.11 | <a href="#">writeCommand32</a>                             | 103 |
| 4.47.3.12 | <a href="#">writeCommand8</a>                              | 103 |
| 4.47.3.13 | <a href="#">writeData16</a>                                | 104 |
| 4.47.3.14 | <a href="#">writeData32</a>                                | 104 |
| 4.47.3.15 | <a href="#">writeData8</a>                                 | 104 |
| 4.48      | <a href="#">TFTPar8 Class Reference</a>                    | 104 |
| 4.48.1    | <a href="#">Detailed Description</a>                       | 105 |
| 4.48.2    | <a href="#">Constructor &amp; Destructor Documentation</a> | 105 |
| 4.48.2.1  | <a href="#">TFTPar8</a>                                    | 105 |
| 4.48.3    | <a href="#">Member Function Documentation</a>              | 105 |
| 4.48.3.1  | <a href="#">blockData</a>                                  | 105 |
| 4.48.3.2  | <a href="#">blockData</a>                                  | 105 |
| 4.48.3.3  | <a href="#">blockData</a>                                  | 105 |
| 4.48.3.4  | <a href="#">initializeDevice</a>                           | 106 |
| 4.48.3.5  | <a href="#">nativeWidth</a>                                | 106 |
| 4.48.3.6  | <a href="#">readCommand16</a>                              | 106 |
| 4.48.3.7  | <a href="#">readCommand32</a>                              | 106 |
| 4.48.3.8  | <a href="#">readCommand8</a>                               | 106 |
| 4.48.3.9  | <a href="#">readData16</a>                                 | 106 |
| 4.48.3.10 | <a href="#">readData32</a>                                 | 106 |
| 4.48.3.11 | <a href="#">readData8</a>                                  | 106 |
| 4.48.3.12 | <a href="#">streamCommand16</a>                            | 106 |
| 4.48.3.13 | <a href="#">streamCommand16</a>                            | 107 |
| 4.48.3.14 | <a href="#">streamCommand32</a>                            | 107 |
| 4.48.3.15 | <a href="#">streamCommand32</a>                            | 107 |
| 4.48.3.16 | <a href="#">streamCommand8</a>                             | 107 |
| 4.48.3.17 | <a href="#">streamCommand8</a>                             | 107 |
| 4.48.3.18 | <a href="#">streamData16</a>                               | 107 |
| 4.48.3.19 | <a href="#">streamData16</a>                               | 107 |
| 4.48.3.20 | <a href="#">streamData32</a>                               | 107 |

|  |     |
|--|-----|
| 4.48.3.21 streamData32 . . . . .               | 107 |
| 4.48.3.22 streamData8 . . . . .                | 108 |
| 4.48.3.23 streamData8 . . . . .                | 108 |
| 4.48.3.24 streamEnd . . . . .                  | 108 |
| 4.48.3.25 streamStart . . . . .                | 108 |
| 4.48.3.26 writeCommand16 . . . . .             | 108 |
| 4.48.3.27 writeCommand32 . . . . .             | 108 |
| 4.48.3.28 writeCommand8 . . . . .              | 108 |
| 4.48.3.29 writeData16 . . . . .                | 108 |
| 4.48.3.30 writeData32 . . . . .                | 108 |
| 4.48.3.31 writeData8 . . . . .                 | 109 |
| 4.49 TFTPMP Class Reference . . . . .          | 109 |
| 4.49.1 Member Function Documentation . . . . . | 110 |
| 4.49.1.1 blockData . . . . .                   | 110 |
| 4.49.1.2 blockData . . . . .                   | 110 |
| 4.49.1.3 blockData . . . . .                   | 110 |
| 4.49.1.4 initializeDevice . . . . .            | 110 |
| 4.49.1.5 nativeWidth . . . . .                 | 110 |
| 4.49.1.6 readCommand16 . . . . .               | 110 |
| 4.49.1.7 readCommand32 . . . . .               | 110 |
| 4.49.1.8 readCommand8 . . . . .                | 110 |
| 4.49.1.9 readData16 . . . . .                  | 110 |
| 4.49.1.10 readData32 . . . . .                 | 111 |
| 4.49.1.11 readData8 . . . . .                  | 111 |
| 4.49.1.12 streamCommand16 . . . . .            | 111 |
| 4.49.1.13 streamCommand16 . . . . .            | 111 |
| 4.49.1.14 streamCommand32 . . . . .            | 111 |
| 4.49.1.15 streamCommand32 . . . . .            | 111 |
| 4.49.1.16 streamCommand8 . . . . .             | 111 |
| 4.49.1.17 streamCommand8 . . . . .             | 111 |
| 4.49.1.18 streamData16 . . . . .               | 111 |
| 4.49.1.19 streamData16 . . . . .               | 112 |
| 4.49.1.20 streamData32 . . . . .               | 112 |
| 4.49.1.21 streamData32 . . . . .               | 112 |
| 4.49.1.22 streamData8 . . . . .                | 112 |
| 4.49.1.23 streamData8 . . . . .                | 112 |
| 4.49.1.24 streamEnd . . . . .                  | 112 |
| 4.49.1.25 streamStart . . . . .                | 112 |
| 4.49.1.26 writeCommand16 . . . . .             | 112 |
| 4.49.1.27 writeCommand32 . . . . .             | 112 |

|   |     |
|---|-----|
| 4.49.1.28 writeCommand8 . . . . .                       | 113 |
| 4.49.1.29 writeData16 . . . . .                         | 113 |
| 4.49.1.30 writeData32 . . . . .                         | 113 |
| 4.49.1.31 writeData8 . . . . .                          | 113 |
| 4.50 TFTSoftSPI Class Reference . . . . .               | 113 |
| 4.50.1 Detailed Description . . . . .                   | 114 |
| 4.50.2 Constructor & Destructor Documentation . . . . . | 114 |
| 4.50.2.1 TFTSoftSPI . . . . .                           | 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 . . . . .                            | 115 |
| 4.50.3.4 initializeDevice . . . . .                     | 115 |
| 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 . . . . .                     | 116 |
| 4.50.3.13 streamCommand16 . . . . .                     | 116 |
| 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 . . . . .                        | 117 |
| 4.50.3.22 streamData8 . . . . .                         | 117 |
| 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 . . . . .                         | 118 |
| 4.50.3.31 writeData8 . . . . .                          | 118 |



|   |     |
|---|-----|
| 4.51 Touch Class Reference . . . . .                    | 118 |
| 4.51.1 Constructor & Destructor Documentation . . . . . | 118 |
| 4.51.1.1 Touch . . . . .                                | 118 |
| 4.51.1.2 Touch . . . . .                                | 119 |
| 4.51.1.3 Touch . . . . .                                | 119 |
| 4.51.2 Member Function Documentation . . . . .          | 119 |
| 4.51.2.1 initializeDevice . . . . .                     | 119 |
| 4.51.2.2 isPressed . . . . .                            | 119 |
| 4.51.2.3 pressure . . . . .                             | 119 |
| 4.51.2.4 sample . . . . .                               | 119 |
| 4.51.2.5 setRotation . . . . .                          | 120 |
| 4.51.2.6 x . . . . .                                    | 120 |
| 4.51.2.7 y . . . . .                                    | 120 |
| 4.51.3 Member Data Documentation . . . . .              | 120 |
| 4.51.3.1 _comm . . . . .                                | 120 |
| 4.51.3.2 _height . . . . .                              | 120 |
| 4.51.3.3 _width . . . . .                               | 120 |
| 4.52 XPT2046 Class Reference . . . . .                  | 120 |
| 4.52.1 Constructor & Destructor Documentation . . . . . | 121 |
| 4.52.1.1 XPT2046 . . . . .                              | 121 |
| 4.52.2 Member Function Documentation . . . . .          | 121 |
| 4.52.2.1 initializeDevice . . . . .                     | 121 |
| 4.52.2.2 isPressed . . . . .                            | 121 |
| 4.52.2.3 sample . . . . .                               | 121 |
| 4.52.2.4 setRotation . . . . .                          | 122 |
| 4.52.2.5 x . . . . .                                    | 122 |
| 4.52.2.6 y . . . . .                                    | 122 |



## 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 . . . . .            | 62 |
| SRAM . . . . .               | 64 |
| FontHeader . . . . .         | 24 |
| Image . . . . .              | 46 |
| BMP . . . . .                | 17 |
| Raw565 . . . . .             | 54 |
| Raw8 . . . . .               | 54 |
| RLE . . . . .                | 57 |
| MatrixISRList . . . . .      | 52 |
| ParallelIO . . . . .         | 53 |
| CoreIO . . . . .             | 22 |
| MCP23S17 . . . . .           | 53 |
| point3d . . . . .            | 54 |
| Print . . . . .              |    |
| TFT . . . . .                | 75 |
| 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  |
| S6D0164 . . . . .         | 58  |
| SSD1289 . . . . .         | 65  |
| SSD1963 . . . . .         | 68  |
| ST7735 . . . . .          | 71  |
| sprite . . . . .          | 62  |
| TFTCommunicator . . . . . | 91  |
| RawPar . . . . .          | 55  |
| TFTPar16 . . . . .        | 96  |
| TFTPar4 . . . . .         | 101 |
| TFTPar8 . . . . .         | 104 |
| TFTPMP . . . . .          | 109 |
| TFTSoftSPI . . . . .      | 113 |
| Touch . . . . .           | 118 |
| AnalogTouch . . . . .     | 11  |
| XPT2046 . . . . .         | 120 |

## Chapter 3

# Class Index

### 3.1 Class List

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

|                                    |    |
|------------------------------------|----|
| <a href="#">__attribute__</a>      | 7  |
| <a href="#">Aggregator</a>         | 7  |
| <a href="#">AggregatorList</a>     | 10 |
| <a href="#">AnalogTouch</a>        | 11 |
| <a href="#">BD663474</a>           | 12 |
| <a href="#">BitmapFileHeader</a>   | 15 |
| <a href="#">BitmapInfoHeader</a>   | 16 |
| <a href="#">BitmapPixel24</a>      | 16 |
| <a href="#">BitmapPixel32</a>      | 16 |
| <a href="#">BMP</a>                | 17 |
| <a href="#">Color</a>              | 17 |
| <a href="#">coord</a>              | 22 |
| <a href="#">CoreIO</a>             | 22 |
| <a href="#">DataBlock</a>          | 22 |
| <a href="#">DataStore</a>          | 23 |
| <a href="#">DOGMe</a>              | 23 |
| <a href="#">FontHeader</a>         | 24 |
| <a href="#">Framebuffer</a>        | 25 |
| <a href="#">Framebuffer1</a>       | 29 |
| <a href="#">Framebuffer332</a>     | 31 |
| <a href="#">Framebuffer332Fast</a> | 33 |
| <a href="#">Framebuffer565</a>     | 35 |
| <a href="#">HD44780</a>            | 36 |
| <a href="#">HX8357</a>             | 40 |
| <a href="#">ILI9340</a>            | 43 |
| <a href="#">Image</a>              | 46 |
| <a href="#">KS0108</a>             | 47 |
| <a href="#">LEDMatrix</a>          | 50 |
| <a href="#">MatrixISRList</a>      | 52 |
| <a href="#">MCP23S17</a>           | 53 |
| <a href="#">ParallelIO</a>         | 53 |
| <a href="#">point3d</a>            | 54 |
| <a href="#">Raw565</a>             | 54 |
| <a href="#">Raw8</a>               | 54 |
| <a href="#">RawPar</a>             | 55 |
| <a href="#">RLE</a>                | 57 |
| <a href="#">S6D0164</a>            | 58 |
| <a href="#">SPISRAM</a>            | 62 |

|                                 |     |
|---------------------------------|-----|
| <a href="#">sprite</a>          | 62  |
| <a href="#">SRAM</a>            | 64  |
| <a href="#">SSD1289</a>         | 65  |
| <a href="#">SSD1963</a>         | 68  |
| <a href="#">ST7735</a>          | 71  |
| <a href="#">TFT</a>             | 75  |
| <a href="#">TFTCommunicator</a> | 91  |
| <a href="#">TFTPar16</a>        | 96  |
| <a href="#">TFTPar4</a>         | 101 |
| <a href="#">TFTPar8</a>         | 104 |
| <a href="#">TFTPMP</a>          | 109 |
| <a href="#">TFTSoftSPI</a>      | 113 |
| <a href="#">Touch</a>           | 118 |
| <a href="#">XPT2046</a>         | 120 |



## Chapter 4

# Class Documentation

### 4.1 `__attribute__` Struct Reference

#### Public Member Functions

- union {  
    uint16\_t value  
} `__attribute__` ((packed))

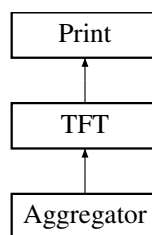
The documentation for this struct was generated from the following file:

- 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

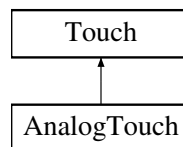
- `int16_t x`
- `int16_t y`
- `struct AggregatorList * next`

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.

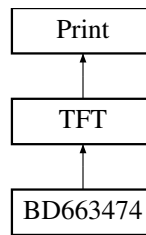
Implements [Touch](#).

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

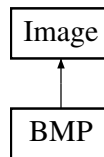
- union {  
    uint32\_t **value**  
} **\_\_attribute\_\_((packed))**

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

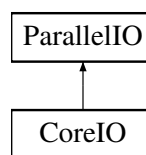
- uint16\_t **x**
- uint16\_t **y**

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)



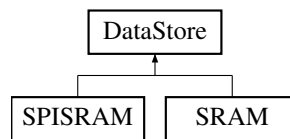
- `uint8_t get (uint32_t a)`

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

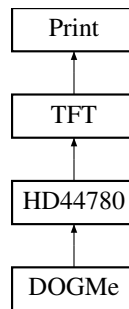
- virtual `uint8_t read8 (uint32_t addr)=0`
- virtual `uint16_t read16 (uint32_t addr)=0`
- virtual `uint32_t read32 (uint32_t addr)=0`
- virtual `void read8 (uint32_t addr, uint8_t *data, uint32_t len)=0`
- virtual `void read16 (uint32_t addr, uint16_t *data, uint32_t len)=0`
- virtual `void read32 (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** ()
- size\_t **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**

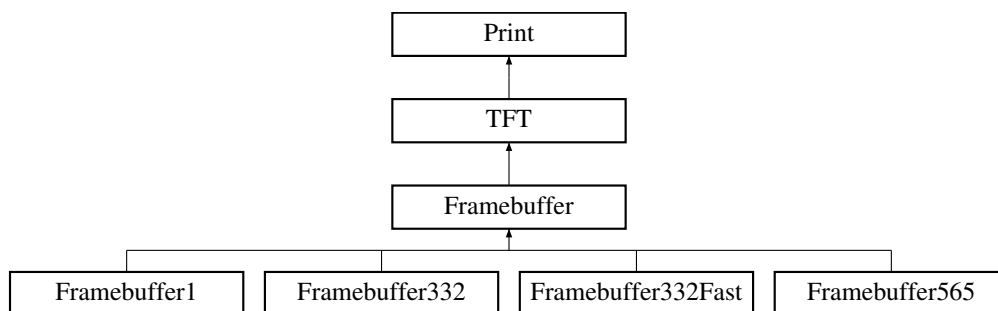
- `uint8_t bytesPerLine`
- `uint8_t startGlyph`
- `uint8_t endGlyph`
- `uint8_t bitsPerPixel`

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);
```

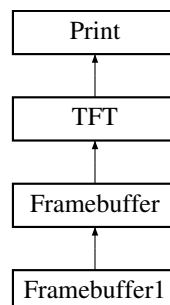
Implements [TFT](#).

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);
```

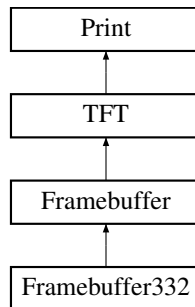
Reimplemented from [Framebuffer](#).

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);
```

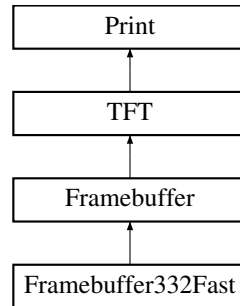
Reimplemented from [Framebuffer](#).

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);
```

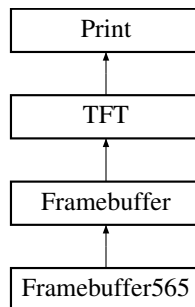
Reimplemented from [Framebuffer](#).

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);
```

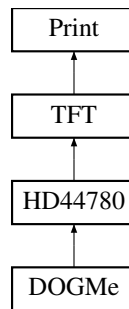
Reimplemented from [Framebuffer](#).

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);
```

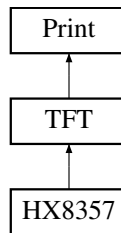
Implements [TFT](#).

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);
```

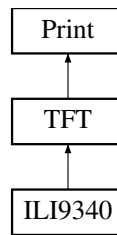
Reimplemented from [TFT](#).

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);
```

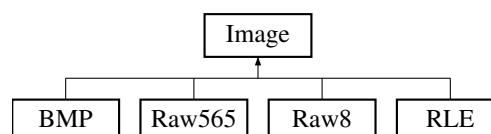
Implements [TFT](#).

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

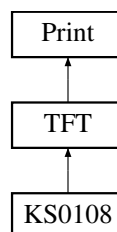
- static const uint8\_t **MirrorH** = 0x01
- static const uint8\_t **MirrorV** = 0x02
- static const uint8\_t **Rotate180** = 0x03

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.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](#).

4.27.1.6 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);
```

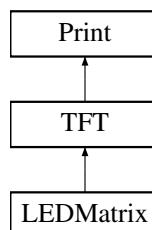
Implements [TFT](#).

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

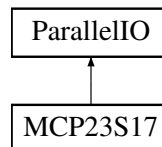
- struct [MatrixISRList](#) \* **next**

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

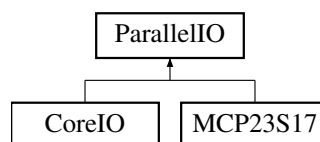
- **MCP23S17** (SPIClass \*spi, uint8\_t cs, uint8\_t addr)
- **MCP23S17** (SPIClass &spi, uint8\_t cs, uint8\_t addr)
- void **pinMode** (uint16\_t pin, uint8\_t mode)
- void **digitalWrite** (uint16\_t pin, uint8\_t value)
- uint8\_t **digitalRead** (uint16\_t pin)
- uint16\_t **pins** ()
- void **startBuffer** ()
- 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 point3d Struct Reference

### Public Attributes

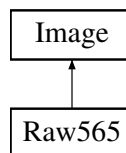
- float **x**
- float **y**
- float **z**

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

- TFT.h

## 4.33 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)

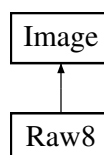
### Additional Inherited Members

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

- Raw565.h
- Raw565.cpp

## 4.34 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)

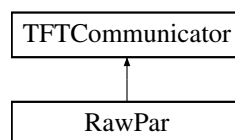
## Additional Inherited Members

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

- Raw8.h
- Raw8.cpp

## 4.35 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** ()

### 4.35.1 Member Function Documentation

#### 4.35.1.1 `uint8_t RawPar::nativeWidth ( )` [virtual]

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

#### 4.35.1.2 `void RawPar::streamCommand16 ( uint16_t data )` [virtual]

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

#### 4.35.1.3 `void RawPar::streamCommand32 ( uint32_t data )` [virtual]

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

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

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

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

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

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

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

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

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

#### 4.35.1.8 `void RawPar::streamEnd ( )` [virtual]

Close the currently open stream

Implements [TFTCommunicator](#).

#### 4.35.1.9 `void RawPar::streamStart ( )` [virtual]

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

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

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

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

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

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

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

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

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

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

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

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

Write 8 bits of data to the device

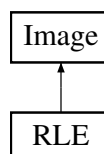
Implements [TFTCommunicator](#).

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

- RawPar.h
- RawPar.cpp

## 4.36 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)

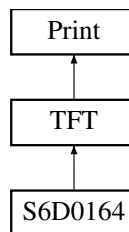
### Additional Inherited Members

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

- RLE.h
- RLE.cpp

## 4.37 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

### 4.37.1 Member Function Documentation

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

##### Close the window

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

Example:

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

Reimplemented from [TFT](#).

#### 4.37.1.2 void S6D0164::displayOff ( ) [virtual]

##### Turn off the display

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

Example:

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

Implements [TFT](#).

#### 4.37.1.3 void S6D0164::displayOn ( ) [virtual]

##### Turn on the display

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

Example:

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

Implements [TFT](#).

#### 4.37.1.4 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](#).

#### 4.37.1.5 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](#).

**4.37.1.6** `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.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](#).

**4.37.1.7** `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](#).

**4.37.1.8** `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](#).

**4.37.1.9** `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](#).

4.37.1.10 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](#).

4.37.1.11 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](#).

4.37.1.12 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](#).

4.37.1.13 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](#).

4.37.1.14 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);
```

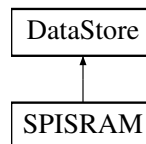
Reimplemented from [TFT](#).

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

- S6D0164.h

## 4.38 SPIRAM Class Reference

Inheritance diagram for SPIRAM:



### Public Member Functions

- **SPIRAM** (SPIClass \*spi, uint8\_t cs, uint32\_t s)
- **SPIRAM** (SPIClass &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:

- SPIRAM.h
- SPIRAM.cpp

## 4.39 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`

### 4.39.1 Detailed Description

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

### 4.39.2 Member Data Documentation

#### 4.39.2.1 `int8_t sprite::animdir`

Direction the animation is running

#### 4.39.2.2 `int8_t sprite::currentframe`

Currently displayed frame number

#### 4.39.2.3 `const uint8_t* sprite::data`

Pointer to graphical data for sprite

#### 4.39.2.4 `int8_t sprite::frames`

Number of frames in the sprite

#### 4.39.2.5 `uint16_t sprite::height`

Height of the sprite

#### 4.39.2.6 `struct sprite* sprite::next`

Pointer to next sprite in the list

#### 4.39.2.7 `int8_t sprite::store[8]`

Internal data store for sprite specific information

#### 4.39.2.8 `uint8_t sprite::transparent`

Transparent colour index

#### 4.39.2.9 uint16\_t sprite::width

Width of the sprite

#### 4.39.2.10 int16\_t sprite::xpos

X Position of the sprite

#### 4.39.2.11 int16\_t sprite::ypos

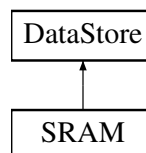
Y Position of the sprite

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

- Framebuffer.h

## 4.40 SRAM Class Reference

Inheritance diagram for SRAM:



### Public Member Functions

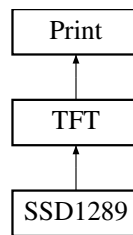
- **SRAM** (uint8\_t \*buf, uint32\_t s)
- uint8\_t **read8** (uint32\_t address)
- uint16\_t **read16** (uint32\_t address)
- uint32\_t **read32** (uint32\_t address)
- void **read8** (uint32\_t address, uint8\_t \*data, uint32\_t len)
- void **read16** (uint32\_t address, uint16\_t \*data, uint32\_t 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

## 4.41 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

#### 4.41.1 Member Function Documentation

4.41.1.1 void **SSD1289::closeWindow** ( ) [[virtual](#)]

##### Close the window

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

Example:

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

Reimplemented from [TFT](#).

4.41.1.2 `void SSD1289::displayOff ( ) [inline],[virtual]`

### Turn off the display

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

Example:

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

Implements [TFT](#).

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

### Turn on the display

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

Example:

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

Implements [TFT](#).

4.41.1.4 `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](#).

4.41.1.5 `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](#).

4.41.1.6 `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.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](#).

**4.41.1.7** `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](#).

**4.41.1.8** `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](#).

**4.41.1.9** `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](#).

**4.41.1.10** `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](#).

**4.41.1.11** `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](#).

4.41.1.12 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](#).

4.41.1.13 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](#).

4.41.1.14 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);
```

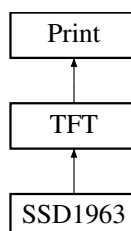
Reimplemented from [TFT](#).

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

- SSD1289.h
- SSD1289.cpp

## 4.42 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

### 4.42.1 Member Function Documentation

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

Not currently implemented

Implements [TFT](#).

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

Not currently implemented

Implements [TFT](#).

4.42.1.3 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](#).

4.42.1.4 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](#).

**4.42.1.5** `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](#).

**4.42.1.6** `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](#).

**4.42.1.7** `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](#).

**4.42.1.8** `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](#).



4.42.1.9 `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](#).

4.42.1.10 `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](#).

### 4.42.2 Member Data Documentation

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

The height of the screen is 480 pixels

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

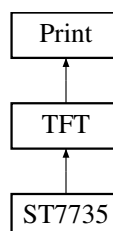
The width of the screen is 800 pixels

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

- SSD1963.h
- SSD1963.cpp

## 4.43 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

### 4.43.1 Constructor & Destructor Documentation

#### 4.43.1.1 [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](#)

### 4.43.2 Member Function Documentation

#### 4.43.2.1 [void \[ST7735::displayOff\]\(#\) \( \)](#) [\[inline\]](#), [\[virtual\]](#)

#### Turn off the display

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

Example:

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

Implements [TFT](#).

#### 4.43.2.2 void ST7735::displayOn ( ) [inline],[virtual]

##### Turn on the display

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

Example:

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

Implements [TFT](#).

#### 4.43.2.3 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](#).

#### 4.43.2.4 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](#).

#### 4.43.2.5 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.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](#).

#### 4.43.2.6 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](#).

#### 4.43.2.7 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](#).

#### 4.43.2.8 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](#).

#### 4.43.2.9 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](#).

#### 4.43.2.10 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](#).

### 4.43.3 Member Data Documentation

#### 4.43.3.1 const uint8\_t ST7735::BlackTab = 0x02 [static]

Adafruit screen with a black tab

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

Adafruit screen with a green tab

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

The native size of the screen is 160 pixels high

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

Adafruit screen with a red tab

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

Adafruit "Type B" screen

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

The native size of the screen is 128 pixels wide

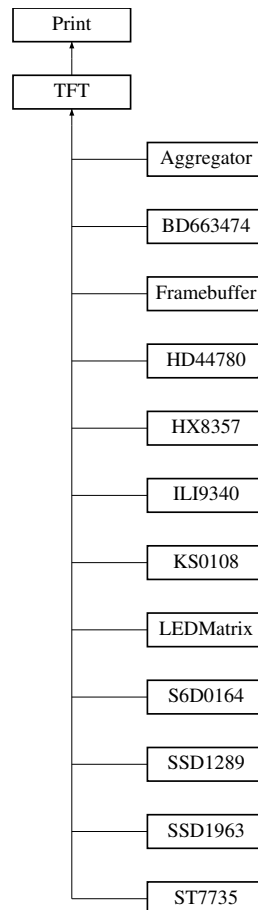
The documentation for this class was generated from the following files:

- ST7735.h
- ST7735.cpp

## 4.44 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)
- void [setClipping](#) (int16\_t x0, int16\_t y0, int16\_t x1, int16\_t y1)
- void [clearClipping](#) ()

## 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 void [setCursorX](#) (int16\_t x)
- virtual void [setCursorY](#) (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](#)
- [int16\\_t](#) [\\_clip\\_x0](#)
- [int16\\_t](#) [\\_clip\\_x1](#)
- [int16\\_t](#) [\\_clip\\_y0](#)
- [int16\\_t](#) [\\_clip\\_y1](#)

## Protected Attributes

- [const uint8\\_t](#) \* [font](#)
- [uint8\\_t](#) [font\\_scale\\_x](#)
- [uint8\\_t](#) [font\\_scale\\_y](#)

### 4.44.1 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).

### 4.44.2 Constructor & Destructor Documentation

#### 4.44.2.1 [TFT::TFT](#) ( )

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

#### 4.44.2.2 [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.

#### 4.44.2.3 [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.

### 4.44.3 Member Function Documentation

#### 4.44.3.1 [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](#).

#### 4.44.3.2 void TFT::clearClipping ( )

##### Clear clipping boundaries

Remove the clipping boundary imposed by [setClipping\(\)](#).

Example:

```
clearClipping();
```

#### 4.44.3.3 void TFT::closeWindow ( ) [virtual]

##### Close the window

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

Example:

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

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

#### 4.44.3.4 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);
```

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

#### 4.44.3.6 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);
```

#### 4.44.3.7 `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);
```

#### 4.44.3.8 `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](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), and [LEDMatrix](#).

#### 4.44.3.9 `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](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), and [LEDMatrix](#).

#### 4.44.3.10 `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);
```

4.44.3.11 `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);
```

4.44.3.12 `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);
```

4.44.3.13 `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.

4.44.3.14 `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](#), [Aggregator](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Framebuffer332Fast](#), [KS0108](#), [BD663474](#), [HX8357](#), and [SSD1289](#).

4.44.3.15 `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);
```

4.44.3.16 `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.drawRect(10, 10, 200, 300, Color::Blue);
```

**4.44.3.17** 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);
```

**4.44.3.18** 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);
```

**4.44.3.19** 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);
```

**4.44.3.20** 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);
```

4.44.3.21 `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](#), [Aggregator](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [KS0108](#), [BD663474](#), [HX8357](#), and [SSD1289](#).

4.44.3.22 `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");
```

4.44.3.23 `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);
```

4.44.3.24 `void TFT::fillCircleHelper ( int16_t x0, int16_t y0, int16_t r, uint8_t cornename, int16_t delta, uint16_t color )`

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

4.44.3.25 `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](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [KS0108](#), [BD663474](#), [HX8357](#), and [SSD1289](#).

**4.44.3.26** `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);
```

**4.44.3.27** `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](#), [HD44780](#), [Aggregator](#), [S6D0164](#), [ILI9340](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), [LEDMatrix](#), [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

**4.44.3.28** `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);
```

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

#### Get Text Cursor

Returns the ether 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);
```

**4.44.3.30** `int16_t TFT::getCursorX ( )` [virtual]

#### Get X Cursor

Returns the current X position of the text cursor.

Example:

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

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

#### Get Y Cursor

Returns the current Y position of the text cursor.

Example:

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

4.44.3.32 `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](#).

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

#### Get the current foreground colour

Returns the currently selected foreground colour.

Example:

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

4.44.3.34 `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](#).

4.44.3.35 `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](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), [Framebuffer565](#), [LEDMatrix](#), and [DOG-Me](#).

**4.44.3.36** `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](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), and [LEDMatrix](#).

**4.44.3.37** `void TFT::invertTextColor ( )` [virtual]

#### Invert the text colours

The foreground becomes the background, and the background becomes the foreground.

Example:

```
tft.invertTextColor();
```

**4.44.3.38** `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);
```

**4.44.3.39** `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 [S6D0164](#), [HX8357](#), and [SSD1289](#).

**4.44.3.40** `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);
```



#### 4.44.3.41 `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);
```

#### 4.44.3.42 `void TFT::setClipping ( int16_t x0, int16_t y0, int16_t x1, int16_t y1 )`

##### Set clipping boundaries

The clipping boundaries limit where a pixel can be drawn on the screen. It allows you to define an area where primitives will be drawn within and any portion outside the clipping area will be discarded.

Example:

```
setClipping(100, 100, 200, 200);
```

#### 4.44.3.43 `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);
```

#### 4.44.3.44 `void TFT::setCursorX ( int16_t x ) [virtual]`

##### Set the text X cursor

All future printing will happen from the X pixel (x).

Example:

```
tft.setCursorX(100);
```

#### 4.44.3.45 `void TFT::setCursorY ( int16_t y ) [virtual]`

##### Set the text Y cursor

All future printing will happen from the Y pixel (y).

Example:

```
tft.setCursorY(100);
```

4.44.3.46 `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);
```

4.44.3.47 `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);
```

4.44.3.48 `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);
```

4.44.3.49 `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](#), [HD44780](#), [Aggregator](#), [S6D0164](#), [ILI9340](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), [LEDMatrix](#), [Framebuffer1](#), [Framebuffer332](#), [Framebuffer332Fast](#), and [Framebuffer565](#).

4.44.3.50 `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](#), [HD44780](#), [S6D0164](#), [ILI9340](#), [Aggregator](#), [KS0108](#), [BD663474](#), [HX8357](#), [SSD1289](#), and [LEDMatrix](#).

4.44.3.51 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);
```

4.44.3.52 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);
```

4.44.3.53 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);
```

4.44.3.54 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");
```

4.44.3.55 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");
```

4.44.3.56 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 [S6D0164](#), [HX8357](#), and [SSD1289](#).

4.44.3.57 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 [S6D0164](#), [HX8357](#), and [SSD1289](#).

4.44.3.58 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');
```

4.44.3.59 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);
```

### 4.44.4 Member Data Documentation

4.44.4.1 TFTCommunicator\* TFT::\_comm

The device used to communicate with the [TFT](#) screen

4.44.4.2 uint16\_t TFT::\_height

Height of the [TFT](#) screen

## 4.44.4.3 uint16\_t TFT::\_width

Width of the [TFT](#) screen

## 4.44.4.4 int16\_t TFT::cursor\_x

The text cursor X position

## 4.44.4.5 int16\_t TFT::cursor\_y

The text cursor Y position

## 4.44.4.6 const uint8\_t\* TFT::font [protected]

A pointer to the currently selected font table

## 4.44.4.7 uint8\_t TFT::font\_scale\_x [protected]

The current X scaling factor of the font

## 4.44.4.8 uint8\_t TFT::font\_scale\_y [protected]

The current Y scaling factor of the font

## 4.44.4.9 uint8\_t TFT::rotation

Current rotation

## 4.44.4.10 uint16\_t TFT::textbgcolor

Text background colour

## 4.44.4.11 uint16\_t TFT::textcolor

Text foreground colour

## 4.44.4.12 boolean TFT::wrap

Whether or not text wrapping is enabled

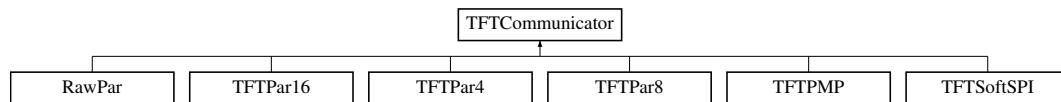
The documentation for this class was generated from the following files:

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

## 4.45 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

## 4.45.1 Detailed Description

A [TFTCommunicator](#) device forms the bridge between the [TFT](#) class and the physical screen.

## 4.45.2 Member Function Documentation

4.45.2.1 `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](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.2 `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](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.3 `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](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.4 `virtual void TFTCommunicator::initializeDevice ( ) [pure virtual]`

Initialize the communication device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.5 `virtual uint8_t TFTCommunicator::nativeWidth ( ) [pure virtual]`

Returns the real physical width of the data channel

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPar8](#), [TFTPMP](#), [TFTPar4](#), and [RawPar](#).

4.45.2.6 `virtual uint16_t TFTCommunicator::readCommand16 ( ) [pure virtual]`

Read a 16-bit command from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.7 `virtual uint32_t TFTCommunicator::readCommand32 ( ) [pure virtual]`

Read a 32-bit command from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.8 `virtual uint8_t TFTCommunicator::readCommand8 ( ) [pure virtual]`

Read an 8-bit command from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.9 `virtual uint16_t TFTCommunicator::readData16 ( ) [pure virtual]`

Read 16 bits of data from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.10 `virtual uint32_t TFTCommunicator::readData32 ( ) [pure virtual]`

Read 32 bits of data from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.11 `virtual uint8_t TFTCommunicator::readData8 ( ) [pure virtual]`

Read 8 bits of data from the device

Implemented in [TFTPar16](#), [TFTPar8](#), [TFTSoftSPI](#), and [TFTPMP](#).

4.45.2.12 `virtual void TFTCommunicator::streamCommand16 ( uint16_t data ) [pure virtual]`

Send a 16-bit command through the stream

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.13 `virtual uint16_t TFTCommunicator::streamCommand16 ( ) [pure virtual]`

Read a 16-bit command through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.45.2.14 `virtual void TFTCommunicator::streamCommand32 ( uint32_t data ) [pure virtual]`

Send a 32-bit command through the stream

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.15 `virtual uint32_t TFTCommunicator::streamCommand32 ( ) [pure virtual]`

Read a 32-bit command through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.45.2.16 `virtual void TFTCommunicator::streamCommand8 ( uint8_t data ) [pure virtual]`

Send an 8-bit command through the stream

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.17 `virtual uint8_t TFTCommunicator::streamCommand8 ( ) [pure virtual]`

Read an 8-bit command through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.45.2.18 `virtual void TFTCommunicator::streamData16 ( uint16_t data ) [pure virtual]`

Send 16-bits of data through the stream

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).



4.45.2.19 `virtual uint16_t TFTCommunicator::streamData16 ( ) [pure virtual]`

Read 16 bits of data through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.45.2.20 `virtual void TFTCommunicator::streamData32 ( uint32_t data ) [pure virtual]`

Send 32-bits of data through the stream

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.21 `virtual uint32_t TFTCommunicator::streamData32 ( ) [pure virtual]`

Read 32 bits of data through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.45.2.22 `virtual void TFTCommunicator::streamData8 ( uint8_t data ) [pure virtual]`

Send 8-bits of data through the stream

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.23 `virtual uint8_t TFTCommunicator::streamData8 ( ) [pure virtual]`

Read 8 bits of data through the stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPar8](#), and [TFTPMP](#).

4.45.2.24 `virtual void TFTCommunicator::streamEnd ( ) [pure virtual]`

Close the currently open stream

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPMP](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.25 `virtual void TFTCommunicator::streamStart ( ) [pure virtual]`

Open a stream to the device endpoint

Implemented in [TFTPar16](#), [TFTSoftSPI](#), [TFTPMP](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.26 `virtual void TFTCommunicator::writeCommand16 ( uint16_t command ) [pure virtual]`

Write a 16-bit command to the device

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.27 `virtual void TFTCommunicator::writeCommand32 ( uint32_t command ) [pure virtual]`

Write a 32-bit command to the device

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.28 `virtual void TFTCommunicator::writeCommand8 ( uint8_t command ) [pure virtual]`

Write an 8-bit command to the device

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.29 `virtual void TFTCommunicator::writeData16 ( uint16_t data ) [pure virtual]`

Write 16 bits of data to the device

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.30 `virtual void TFTCommunicator::writeData32 ( uint32_t data ) [pure virtual]`

Write 32 bits of data to the device

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

4.45.2.31 `virtual void TFTCommunicator::writeData8 ( uint8_t data ) [pure virtual]`

Write 8 bits of data to the device

Implemented in [TFTPar16](#), [TFTPMP](#), [TFTSoftSPI](#), [TFTPar4](#), [TFTPar8](#), and [RawPar](#).

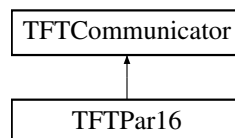
The documentation for this class was generated from the following file:

- [TFTCommunicator.h](#)

## 4.46 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](#) []

#### 4.46.1 Detailed Description

The [TFTPar16](#) class creates a full 16-bit parallel interface to a [TFT](#) device

#### 4.46.2 Constructor & Destructor Documentation

4.46.2.1 [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

4.46.2.2 [TFTPar16::TFTPar16 \( const uint8\\_t \\* profile \)](#) [\[inline\]](#)

Construct a new 16-bit parallel device using a pre-programmed profile

#### 4.46.3 Member Function Documentation

4.46.3.1 [void TFTPar16::blockData \( uint8\\_t \\* data, uint32\\_t len \)](#) [\[virtual\]](#)

Transfer a block of 8-bit data to the device

Implements [TFTCommunicator](#).

4.46.3.2 void TFTPar16::blockData ( uint16\_t \* *data*, uint32\_t *len* ) [virtual]

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

4.46.3.3 void TFTPar16::blockData ( uint32\_t \* *data*, uint32\_t *len* ) [virtual]

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.46.3.4 void TFTPar16::initializeDevice ( ) [inline],[virtual]

Initialize the communication device

Implements [TFTCommunicator](#).

4.46.3.5 uint8\_t TFTPar16::nativeWidth ( ) [inline],[virtual]

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.46.3.6 uint16\_t TFTPar16::readCommand16 ( ) [inline],[virtual]

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.46.3.7 uint32\_t TFTPar16::readCommand32 ( ) [inline],[virtual]

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.46.3.8 uint8\_t TFTPar16::readCommand8 ( ) [inline],[virtual]

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.46.3.9 uint16\_t TFTPar16::readData16 ( ) [inline],[virtual]

Read 16 bits of data from the device

Implements [TFTCommunicator](#).

4.46.3.10 uint32\_t TFTPar16::readData32 ( ) [inline],[virtual]

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.46.3.11 `uint8_t TFTPar16::readData8 ( ) [inline],[virtual]`

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.46.3.12 `void TFTPar16::streamCommand16 ( uint16_t data ) [virtual]`

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.46.3.13 `uint16_t TFTPar16::streamCommand16 ( ) [inline],[virtual]`

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.46.3.14 `void TFTPar16::streamCommand32 ( uint32_t data ) [virtual]`

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.46.3.15 `uint32_t TFTPar16::streamCommand32 ( ) [inline],[virtual]`

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.46.3.16 `void TFTPar16::streamCommand8 ( uint8_t data ) [virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.46.3.17 `uint8_t TFTPar16::streamCommand8 ( ) [inline],[virtual]`

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.46.3.18 `void TFTPar16::streamData16 ( uint16_t data ) [virtual]`

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.46.3.19 `uint16_t TFTPar16::streamData16 ( ) [inline],[virtual]`

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.46.3.20 void TFTPar16::streamData32 ( uint32\_t *data* ) [virtual]

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.46.3.21 uint32\_t TFTPar16::streamData32 ( ) [inline],[virtual]

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.46.3.22 void TFTPar16::streamData8 ( uint8\_t *data* ) [virtual]

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.46.3.23 uint8\_t TFTPar16::streamData8 ( ) [inline],[virtual]

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.46.3.24 void TFTPar16::streamEnd ( ) [virtual]

Close the currently open stream

Implements [TFTCommunicator](#).

4.46.3.25 void TFTPar16::streamStart ( ) [virtual]

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.46.3.26 void TFTPar16::writeCommand16 ( uint16\_t *command* ) [virtual]

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.46.3.27 void TFTPar16::writeCommand32 ( uint32\_t *command* ) [virtual]

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.46.3.28 void TFTPar16::writeCommand8 ( uint8\_t *command* ) [virtual]

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.46.3.29 `void TFTPar16::writeData16 ( uint16_t data ) [virtual]`

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.46.3.30 `void TFTPar16::writeData32 ( uint32_t data ) [virtual]`

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.46.3.31 `void TFTPar16::writeData8 ( uint8_t data ) [virtual]`

Write 8 bits of data to the device

Implements [TFTCommunicator](#).

## 4.46.4 Member Data Documentation

4.46.4.1 `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 ITead Studios Arduino Mega [TFT](#) Interface Adapter

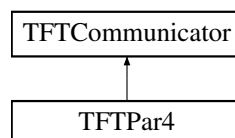
The documentation for this class was generated from the following files:

- TFTPar16.h
- TFTPar16.cpp

## 4.47 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](#) ()

#### 4.47.1 Detailed Description

The [TFTPar4](#) class creates a new 4-bit interface compatible with popular text only LCD screens

#### 4.47.2 Constructor & Destructor Documentation

4.47.2.1 [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*.

4.47.2.2 [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*.

#### 4.47.3 Member Function Documentation

4.47.3.1 [uint8\\_t TFTPar4::nativeWidth](#) ( ) [\[inline\]](#), [\[virtual\]](#)

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.47.3.2 [void TFTPar4::streamCommand16](#) ( uint16\_t *data* ) [\[virtual\]](#)

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.47.3.3 [void TFTPar4::streamCommand32](#) ( uint32\_t *data* ) [\[virtual\]](#)

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).



4.47.3.4 void TFTPar4::streamCommand8 ( uint8\_t *data* ) [virtual]

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.47.3.5 void TFTPar4::streamData16 ( uint16\_t *data* ) [virtual]

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.6 void TFTPar4::streamData32 ( uint32\_t *data* ) [virtual]

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.7 void TFTPar4::streamData8 ( uint8\_t *data* ) [virtual]

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.47.3.8 void TFTPar4::streamEnd ( ) [virtual]

Close the currently open stream

Implements [TFTCommunicator](#).

4.47.3.9 void TFTPar4::streamStart ( ) [virtual]

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.47.3.10 void TFTPar4::writeCommand16 ( uint16\_t *command* ) [virtual]

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.47.3.11 void TFTPar4::writeCommand32 ( uint32\_t *command* ) [virtual]

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.47.3.12 void TFTPar4::writeCommand8 ( uint8\_t *command* ) [virtual]

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.47.3.13 void TFTPar4::writeData16 ( uint16\_t data ) [virtual]

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.47.3.14 void TFTPar4::writeData32 ( uint32\_t data ) [virtual]

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.47.3.15 void TFTPar4::writeData8 ( uint8\_t data ) [virtual]

Write 8 bits of data to the device

Implements [TFTCommunicator](#).

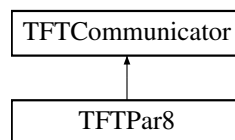
The documentation for this class was generated from the following files:

- TFTPar4.h
- TFTPar4.cpp

## 4.48 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)

#### 4.48.1 Detailed Description

The [TFTPar8](#) class defines an 8-bit parallel interface incorporating the normal [TFT](#) control signals.

#### 4.48.2 Constructor & Destructor Documentation

- 4.48.2.1 [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);
```

#### 4.48.3 Member Function Documentation

- 4.48.3.1 void [TFTPar8::blockData](#) ( uint8\_t \* *data*, uint32\_t *len* ) [virtual]

Transfer a block of 8-bit data to the device

Implements [TFTCommunicator](#).

- 4.48.3.2 void [TFTPar8::blockData](#) ( uint16\_t \* *data*, uint32\_t *len* ) [virtual]

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

- 4.48.3.3 void [TFTPar8::blockData](#) ( uint32\_t \* *data*, uint32\_t *len* ) [virtual]

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.48.3.4 `void TFTPar8::initializeDevice( ) [inline],[virtual]`

Initialize the communication device

Implements [TFTCommunicator](#).

4.48.3.5 `uint8_t TFTPar8::nativeWidth( ) [inline],[virtual]`

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.48.3.6 `uint16_t TFTPar8::readCommand16( ) [inline],[virtual]`

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.48.3.7 `uint32_t TFTPar8::readCommand32( ) [inline],[virtual]`

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.48.3.8 `uint8_t TFTPar8::readCommand8( ) [inline],[virtual]`

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.48.3.9 `uint16_t TFTPar8::readData16( ) [inline],[virtual]`

Read 16 bits of data from the device

Implements [TFTCommunicator](#).

4.48.3.10 `uint32_t TFTPar8::readData32( ) [inline],[virtual]`

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.48.3.11 `uint8_t TFTPar8::readData8( ) [inline],[virtual]`

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.48.3.12 `void TFTPar8::streamCommand16( uint16_t data ) [virtual]`

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.13 `uint16_t TFTPar8::streamCommand16 ( ) [inline],[virtual]`

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.14 `void TFTPar8::streamCommand32 ( uint32_t data ) [virtual]`

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.15 `uint32_t TFTPar8::streamCommand32 ( ) [inline],[virtual]`

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.16 `void TFTPar8::streamCommand8 ( uint8_t data ) [virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.17 `uint8_t TFTPar8::streamCommand8 ( ) [inline],[virtual]`

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.48.3.18 `void TFTPar8::streamData16 ( uint16_t data ) [virtual]`

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.19 `uint16_t TFTPar8::streamData16 ( ) [inline],[virtual]`

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.20 `void TFTPar8::streamData32 ( uint32_t data ) [virtual]`

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.21 `uint32_t TFTPar8::streamData32 ( ) [inline],[virtual]`

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.22 void TFTPar8::streamData8 ( uint8\_t *data* ) [virtual]

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.23 uint8\_t TFTPar8::streamData8 ( ) [inline],[virtual]

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.48.3.24 void TFTPar8::streamEnd ( ) [virtual]

Close the currently open stream

Implements [TFTCommunicator](#).

4.48.3.25 void TFTPar8::streamStart ( ) [virtual]

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.48.3.26 void TFTPar8::writeCommand16 ( uint16\_t *command* ) [virtual]

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.48.3.27 void TFTPar8::writeCommand32 ( uint32\_t *command* ) [virtual]

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.48.3.28 void TFTPar8::writeCommand8 ( uint8\_t *command* ) [virtual]

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.48.3.29 void TFTPar8::writeData16 ( uint16\_t *data* ) [virtual]

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.48.3.30 void TFTPar8::writeData32 ( uint32\_t *data* ) [virtual]

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.48.3.31 void TFTPPar8::writeData8 ( uint8\_t data ) [virtual]

Write 8 bits of data to the device

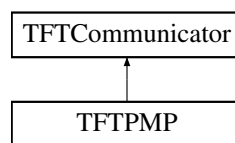
Implements [TFTCommunicator](#).

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

- TFTPPar8.h
- TFTPPar8.cpp

## 4.49 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 ()

## 4.49.1 Member Function Documentation

4.49.1.1 `void TFTPMP::blockData ( uint8_t * data, uint32_t len ) [inline],[virtual]`

Transfer a block of 8-bit data to the device

Implements [TFTCommunicator](#).

4.49.1.2 `void TFTPMP::blockData ( uint16_t * data, uint32_t len ) [inline],[virtual]`

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

4.49.1.3 `void TFTPMP::blockData ( uint32_t * data, uint32_t len ) [inline],[virtual]`

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.49.1.4 `void TFTPMP::initializeDevice ( ) [virtual]`

Initialize the communication device

Implements [TFTCommunicator](#).

4.49.1.5 `uint8_t TFTPMP::nativeWidth ( ) [inline],[virtual]`

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.49.1.6 `uint16_t TFTPMP::readCommand16 ( ) [inline],[virtual]`

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.49.1.7 `uint32_t TFTPMP::readCommand32 ( ) [inline],[virtual]`

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.49.1.8 `uint8_t TFTPMP::readCommand8 ( ) [inline],[virtual]`

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.49.1.9 `uint16_t TFTPMP::readData16 ( ) [inline],[virtual]`

Read 16 bits of data from the device

Implements [TFTCommunicator](#).



4.49.1.10 `uint32_t TFTPMP::readData32 ( ) [inline],[virtual]`

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.49.1.11 `uint8_t TFTPMP::readData8 ( ) [inline],[virtual]`

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.49.1.12 `uint16_t TFTPMP::streamCommand16 ( ) [inline],[virtual]`

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.49.1.13 `void TFTPMP::streamCommand16 ( uint16_t data ) [inline],[virtual]`

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.49.1.14 `uint32_t TFTPMP::streamCommand32 ( ) [inline],[virtual]`

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.49.1.15 `void TFTPMP::streamCommand32 ( uint32_t data ) [inline],[virtual]`

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.49.1.16 `uint8_t TFTPMP::streamCommand8 ( ) [inline],[virtual]`

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.49.1.17 `void TFTPMP::streamCommand8 ( uint8_t data ) [inline],[virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.49.1.18 `uint16_t TFTPMP::streamData16 ( ) [inline],[virtual]`

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.49.1.19 void TFTPMP::streamData16 ( uint16\_t *data* ) [inline],[virtual]

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.49.1.20 uint32\_t TFTPMP::streamData32 ( ) [inline],[virtual]

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.49.1.21 void TFTPMP::streamData32 ( uint32\_t *data* ) [inline],[virtual]

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.49.1.22 uint8\_t TFTPMP::streamData8 ( ) [inline],[virtual]

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.49.1.23 void TFTPMP::streamData8 ( uint8\_t *data* ) [inline],[virtual]

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.49.1.24 void TFTPMP::streamEnd ( ) [inline],[virtual]

Close the currently open stream

Implements [TFTCommunicator](#).

4.49.1.25 void TFTPMP::streamStart ( ) [inline],[virtual]

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.49.1.26 void TFTPMP::writeCommand16 ( uint16\_t *command* ) [inline],[virtual]

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.49.1.27 void TFTPMP::writeCommand32 ( uint32\_t *command* ) [inline],[virtual]

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.49.1.28 `void TFTPMP::writeCommand8 ( uint8_t command ) [inline],[virtual]`

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.49.1.29 `void TFTPMP::writeData16 ( uint16_t data ) [inline],[virtual]`

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.49.1.30 `void TFTPMP::writeData32 ( uint32_t data ) [inline],[virtual]`

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.49.1.31 `void TFTPMP::writeData8 ( uint8_t data ) [inline],[virtual]`

Write 8 bits of data to the device

Implements [TFTCommunicator](#).

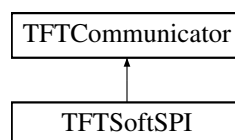
The documentation for this class was generated from the following file:

- TFTPMP.h

## 4.50 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](#) ()

#### 4.50.1 Detailed Description

The [TFTSoftSPI](#) communicator creates an SPI channel on any IO pins. Not as fast as hardware SPI, but allows extra flexibility.

#### 4.50.2 Constructor & Destructor Documentation

4.50.2.1 [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);
```

#### 4.50.3 Member Function Documentation

4.50.3.1 void [TFTSoftSPI::blockData](#) ( uint8\_t \* *data*, uint32\_t *len* ) [inline],[virtual]

Transfer a block of 8-bit data to the device

Implements [TFTCommunicator](#).

4.50.3.2 void [TFTSoftSPI::blockData](#) ( uint16\_t \* *data*, uint32\_t *len* ) [inline],[virtual]

Transfer a block of 16-bit data to the device

Implements [TFTCommunicator](#).

4.50.3.3 void TFTSoftSPI::blockData ( uint32\_t \* *data*, uint32\_t *len* ) [inline],[virtual]

Transfer a block of 32-bit data to the device

Implements [TFTCommunicator](#).

4.50.3.4 void TFTSoftSPI::initializeDevice ( ) [inline],[virtual]

Initialize the communication device

Implements [TFTCommunicator](#).

4.50.3.5 uint8\_t TFTSoftSPI::nativeWidth ( ) [inline],[virtual]

Returns the real physical width of the data channel

Implements [TFTCommunicator](#).

4.50.3.6 uint16\_t TFTSoftSPI::readCommand16 ( ) [inline],[virtual]

Read a 16-bit command from the device

Implements [TFTCommunicator](#).

4.50.3.7 uint32\_t TFTSoftSPI::readCommand32 ( ) [inline],[virtual]

Read a 32-bit command from the device

Implements [TFTCommunicator](#).

4.50.3.8 uint8\_t TFTSoftSPI::readCommand8 ( ) [inline],[virtual]

Read an 8-bit command from the device

Implements [TFTCommunicator](#).

4.50.3.9 uint16\_t TFTSoftSPI::readData16 ( ) [inline],[virtual]

Read 16 bits of data from the device

Implements [TFTCommunicator](#).

4.50.3.10 uint32\_t TFTSoftSPI::readData32 ( ) [inline],[virtual]

Read 32 bits of data from the device

Implements [TFTCommunicator](#).

4.50.3.11 uint8\_t TFTSoftSPI::readData8 ( ) [inline],[virtual]

Read 8 bits of data from the device

Implements [TFTCommunicator](#).

4.50.3.12 `void TFTSoftSPI::streamCommand16 ( uint16_t data ) [virtual]`

Send a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.13 `uint16_t TFTSoftSPI::streamCommand16 ( ) [inline],[virtual]`

Read a 16-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.14 `void TFTSoftSPI::streamCommand32 ( uint32_t data ) [virtual]`

Send a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.15 `uint32_t TFTSoftSPI::streamCommand32 ( ) [inline],[virtual]`

Read a 32-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.16 `void TFTSoftSPI::streamCommand8 ( uint8_t data ) [virtual]`

Send an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.17 `uint8_t TFTSoftSPI::streamCommand8 ( ) [inline],[virtual]`

Read an 8-bit command through the stream

Implements [TFTCommunicator](#).

4.50.3.18 `void TFTSoftSPI::streamData16 ( uint16_t data ) [virtual]`

Send 16-bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.19 `uint16_t TFTSoftSPI::streamData16 ( ) [inline],[virtual]`

Read 16 bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.20 `void TFTSoftSPI::streamData32 ( uint32_t data ) [virtual]`

Send 32-bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.21 `uint32_t TFTSoftSPI::streamData32( ) [inline],[virtual]`

Read 32 bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.22 `void TFTSoftSPI::streamData8( uint8_t data ) [virtual]`

Send 8-bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.23 `uint8_t TFTSoftSPI::streamData8( ) [inline],[virtual]`

Read 8 bits of data through the stream

Implements [TFTCommunicator](#).

4.50.3.24 `void TFTSoftSPI::streamEnd( ) [virtual]`

Close the currently open stream

Implements [TFTCommunicator](#).

4.50.3.25 `void TFTSoftSPI::streamStart( ) [virtual]`

Open a stream to the device endpoint

Implements [TFTCommunicator](#).

4.50.3.26 `void TFTSoftSPI::writeCommand16( uint16_t command ) [virtual]`

Write a 16-bit command to the device

Implements [TFTCommunicator](#).

4.50.3.27 `void TFTSoftSPI::writeCommand32( uint32_t command ) [virtual]`

Write a 32-bit command to the device

Implements [TFTCommunicator](#).

4.50.3.28 `void TFTSoftSPI::writeCommand8( uint8_t command ) [virtual]`

Write an 8-bit command to the device

Implements [TFTCommunicator](#).

4.50.3.29 `void TFTSoftSPI::writeData16( uint16_t data ) [virtual]`

Write 16 bits of data to the device

Implements [TFTCommunicator](#).

4.50.3.30 `void TFTSoftSPI::writeData32 ( uint32_t data ) [virtual]`

Write 32 bits of data to the device

Implements [TFTCommunicator](#).

4.50.3.31 `void TFTSoftSPI::writeData8 ( uint8_t data ) [virtual]`

Write 8 bits of data to the device

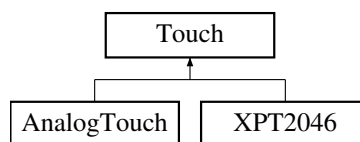
Implements [TFTCommunicator](#).

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

- TFTSoftSPI.h
- TFTSoftSPI.cpp

## 4.51 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](#)

### 4.51.1 Constructor & Destructor Documentation

4.51.1.1 `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.



4.51.1.2 `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.

4.51.1.3 `Touch::Touch ( uint16_t w, uint16_t h ) [inline]`

#### Create a new touch screen object

This creates a new controller-less touch device.

### 4.51.2 Member Function Documentation

4.51.2.1 `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](#).

4.51.2.2 `virtual boolean Touch::isPressed ( ) [pure virtual]`

#### Get pressed status

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

Implemented in [XPT2046](#), and [AnalogTouch](#).

4.51.2.3 `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](#).

4.51.2.4 `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](#).

4.51.2.5 `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](#).

4.51.2.6 `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](#).

4.51.2.7 `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](#).

### 4.51.3 Member Data Documentation

4.51.3.1 `TFTCommunicator* Touch::_comm [protected]`

The communication device used to communicate with the touch screen controller (if any)

4.51.3.2 `uint16_t Touch::_height [protected]`

The height of the touch screen in pixels

4.51.3.3 `uint16_t Touch::_width [protected]`

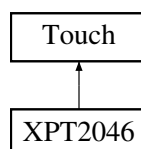
The width of the touch screen in pixels

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

- Touch.h

## 4.52 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

### 4.52.1 Constructor & Destructor Documentation

4.52.1.1 [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);
```

### 4.52.2 Member Function Documentation

4.52.2.1 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](#).

4.52.2.2 boolean [XPT2046::isPressed](#) ( ) [virtual]

#### Get pressed status

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

Implements [Touch](#).

4.52.2.3 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](#).

4.52.2.4 `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](#).

4.52.2.5 `uint16_t XPT2046::x ( ) [virtual]`

#### Get X coordinate

This returns the X coordinate of the current touch position.  
Implements [Touch](#).

4.52.2.6 `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](#), [90](#)
    - [Touch](#), [120](#)
  - [\\_height](#)
    - [TFT](#), [90](#)
    - [Touch](#), [120](#)
  - [\\_width](#)
    - [TFT](#), [90](#)
    - [Touch](#), [120](#)
- [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](#), [63](#)
- [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](#), [78](#)
- [BitmapFileHeader](#), [15](#)
- [BitmapInfoHeader](#), [16](#)
- [BitmapPixel24](#), [16](#)
- [BitmapPixel32](#), [16](#)
- [BlackTab](#)
  - [ST7735](#), [74](#)
- [blockData](#)
  - [TFTCommunicator](#), [93](#)
  - [TFTPar16](#), [97](#), [98](#)
  - [TFTPar8](#), [105](#)
  - [TFTPMP](#), [110](#)
  - [TFTSoftSPI](#), [114](#)
- [clearClipping](#)
  - [TFT](#), [79](#)
- [closeWindow](#)
  - [HX8357](#), [40](#)
  - [S6D0164](#), [59](#)
  - [SSD1289](#), [65](#)
  - [TFT](#), [79](#)
- [Color](#), [17](#)
- [color565](#)
  - [TFT](#), [79](#)
- [colorAt](#)
  - [Framebuffer](#), [26](#)
  - [Framebuffer1](#), [30](#)
  - [Framebuffer332](#), [32](#)
  - [Framebuffer332Fast](#), [33](#)
  - [Framebuffer565](#), [35](#)
  - [TFT](#), [79](#)
- [coord](#), [22](#)
- [CoreIO](#), [22](#)
- [currentframe](#)
  - [sprite](#), [63](#)
- [cursor\\_x](#)
  - [TFT](#), [91](#)
- [cursor\\_y](#)
  - [TFT](#), [91](#)
- [DOGMe](#), [23](#)
  - [initializeDevice](#), [24](#)
- [data](#)
  - [sprite](#), [63](#)
- [DataBlock](#), [22](#)

- Aggregator, [8](#)
  - BD663474, [13](#)
  - Framebuffer, [27](#)
  - HD44780, [37](#)
  - HX8357, [41](#)
  - ILI9340, [44](#)
  - KS0108, [48](#)
  - LEDMatrix, [51](#)
  - S6D0164, [59](#)
  - SSD1289, [65](#)
  - SSD1963, [69](#)
  - ST7735, [72](#)
  - TFT, [80](#)
- displayOn
  - Aggregator, [8](#)
  - BD663474, [13](#)
  - Framebuffer, [27](#)
  - HD44780, [37](#)
  - HX8357, [41](#)
  - ILI9340, [44](#)
  - KS0108, [48](#)
  - LEDMatrix, [51](#)
  - S6D0164, [59](#)
  - SSD1289, [66](#)
  - SSD1963, [69](#)
  - ST7735, [72](#)
  - TFT, [80](#)
- drawBitmap
  - TFT, [80](#)
- drawChar
  - TFT, [80](#)
- drawCircle
  - TFT, [81](#)
- drawCircleHelper
  - TFT, [81](#)
- drawHorizontalLine
  - Aggregator, [8](#)
  - BD663474, [13](#)
  - Framebuffer, [27](#)
  - Framebuffer332Fast, [34](#)
  - HD44780, [38](#)
  - HX8357, [41](#)
  - ILI9340, [44](#)
  - KS0108, [48](#)
  - S6D0164, [59](#)
  - SSD1289, [66](#)
  - SSD1963, [69](#)
  - ST7735, [73](#)
  - TFT, [81](#)
- drawLine
  - TFT, [81](#)
- drawRGB
  - TFT, [82](#)
- drawRGBA
  - TFT, [82](#)
- drawRectangle
  - TFT, [81](#)
- drawRoundRect
  - TFT, [82](#)
- drawTriangle
  - TFT, [82](#)
- drawVerticalLine
  - Aggregator, [9](#)
  - BD663474, [14](#)
  - Framebuffer, [27](#)
  - HD44780, [38](#)
  - HX8357, [41](#)
  - ILI9340, [45](#)
  - KS0108, [48](#)
  - S6D0164, [59](#)
  - SSD1289, [66](#)
  - SSD1963, [69](#)
  - ST7735, [73](#)
  - TFT, [82](#)
- fatalError
  - TFT, [83](#)
- fillCircle
  - TFT, [83](#)
- fillCircleHelper
  - TFT, [83](#)
- fillRectangle
  - BD663474, [14](#)
  - HD44780, [38](#)
  - HX8357, [41](#)
  - ILI9340, [45](#)
  - KS0108, [49](#)
  - S6D0164, [60](#)
  - SSD1289, [66](#)
  - SSD1963, [70](#)
  - ST7735, [73](#)
  - TFT, [83](#)
- fillRoundRect
  - TFT, [83](#)
- 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](#)
  - S6D0164, [60](#)
  - SSD1289, [67](#)
  - SSD1963, [70](#)
  - ST7735, [73](#)

- TFT, [84](#)
- fillTriangle
  - TFT, [84](#)
- font
  - TFT, [91](#)
- font\_scale\_x
  - TFT, [91](#)
- font\_scale\_y
  - TFT, [91](#)
- 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, [63](#)
- getCursor
  - TFT, [84](#)
- getCursorX
  - TFT, [84](#)
- getCursorY
  - TFT, [84](#)
- getHeight
  - Aggregator, [9](#)
- Framebuffer, [28](#)
- TFT, [85](#)
- getTextColor
  - TFT, [85](#)
- getWidth
  - Aggregator, [9](#)
  - Framebuffer, [28](#)
  - TFT, [85](#)
- GreenTab
  - ST7735, [74](#)
- 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, [71](#)
  - ST7735, [75](#)
- height
  - sprite, [63](#)
- 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](#)
  - DOGM, [24](#)

- Framebuffer, 28
- Framebuffer1, 30
- Framebuffer332, 32
- Framebuffer332Fast, 34
- Framebuffer565, 36
- HD44780, 39
- HX8357, 42
- ILI9340, 45
- KS0108, 49
- LEDMatrix, 51
- S6D0164, 60
- SSD1289, 67
- SSD1963, 70
- ST7735, 74
- TFT, 85
- TFTCommunicator, 93
- TFTPar16, 98
- TFTPar8, 105
- TFTPMP, 110
- TFTSoftSPI, 115
- Touch, 119
- XPT2046, 121
- invertDisplay
  - Aggregator, 10
  - BD663474, 15
  - Framebuffer, 28
  - HD44780, 39
  - HX8357, 42
  - ILI9340, 46
  - KS0108, 49
  - LEDMatrix, 52
  - S6D0164, 60
  - SSD1289, 67
  - SSD1963, 70
  - ST7735, 74
  - TFT, 85
- invertTextColor
  - TFT, 86
- isPressed
  - AnalogTouch, 11
  - Touch, 119
  - XPT2046, 121
- IteadAdapter
  - TFTPar16, 101
- 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, 86
- nativeWidth
  - RawPar, 56
  - TFTCommunicator, 93
  - TFTPar16, 98
  - TFTPar4, 102
  - TFTPar8, 106
  - TFTPMP, 110
  - TFTSoftSPI, 115
- next
  - sprite, 63
- openWindow
  - HX8357, 42
  - S6D0164, 60
  - SSD1289, 67
  - TFT, 86
- ParallelIO, 53
- point3d, 54
- pressure
  - AnalogTouch, 11
  - Touch, 119
- RLE, 57
- Raw565, 54
- Raw8, 54
- RawPar, 55
  - nativeWidth, 56
  - streamCommand16, 56
  - streamCommand32, 56
  - streamCommand8, 56
  - streamData16, 56
  - streamData32, 56
  - streamData8, 56
  - streamEnd, 56
  - streamStart, 56
  - writeCommand16, 56
  - writeCommand32, 57
  - writeCommand8, 57
  - writeData16, 57
  - writeData32, 57
  - writeData8, 57
- readCommand16
  - TFTCommunicator, 93
  - TFTPar16, 98
  - TFTPar8, 106
  - TFTPMP, 110



- TFTSoftSPI, 115
- readCommand32
  - TFTCommunicator, 93
  - TFTPar16, 98
  - TFTPar8, 106
  - TFTPMP, 110
  - TFTSoftSPI, 115
- readCommand8
  - TFTCommunicator, 93
  - TFTPar16, 98
  - TFTPar8, 106
  - TFTPMP, 110
  - TFTSoftSPI, 115
- readData16
  - TFTCommunicator, 93
  - TFTPar16, 98
  - TFTPar8, 106
  - TFTPMP, 110
  - TFTSoftSPI, 115
- readData32
  - TFTCommunicator, 93
  - TFTPar16, 98
  - TFTPar8, 106
  - TFTPMP, 110
  - TFTSoftSPI, 115
- readData8
  - TFTCommunicator, 94
  - TFTPar16, 98
  - TFTPar8, 106
  - TFTPMP, 111
  - TFTSoftSPI, 115
- RedTab
  - ST7735, 75
- rgb2hsv
  - TFT, 86
- rgb2xyz
  - TFT, 86
- rotation
  - TFT, 91
- S6D0164, 58
  - closeWindow, 59
  - displayOff, 59
  - displayOn, 59
  - drawHorizontalLine, 59
  - drawVerticalLine, 59
  - fillRectangle, 60
  - fillScreen, 60
  - initializeDevice, 60
  - invertDisplay, 60
  - openWindow, 60
  - setPixel, 61
  - setRotation, 61
  - windowData, 61
- SPISRAM, 62
- SRAM, 64
- SSD1289, 65
  - closeWindow, 65
  - displayOff, 65
  - displayOn, 66
  - drawHorizontalLine, 66
  - drawVerticalLine, 66
  - fillRectangle, 66
  - fillScreen, 67
  - initializeDevice, 67
  - invertDisplay, 67
  - openWindow, 67
  - setPixel, 67
  - setRotation, 68
  - windowData, 68
- SSD1963, 68
  - displayOff, 69
  - displayOn, 69
  - drawHorizontalLine, 69
  - drawVerticalLine, 69
  - fillRectangle, 70
  - fillScreen, 70
  - Height, 71
  - initializeDevice, 70
  - invertDisplay, 70
  - setPixel, 70
  - setRotation, 71
  - Width, 71
- ST7735, 71
  - BlackTab, 74
  - displayOff, 72
  - displayOn, 72
  - drawHorizontalLine, 73
  - drawVerticalLine, 73
  - fillRectangle, 73
  - fillScreen, 73
  - GreenTab, 74
  - Height, 75
  - initializeDevice, 74
  - invertDisplay, 74
  - RedTab, 75
  - ST7735, 72
  - setPixel, 74
  - setRotation, 74
  - ST7735, 72
  - TypeB, 75
  - Width, 75
- sample
  - AnalogTouch, 12
  - Touch, 119
  - XPT2046, 121
- setClipping
  - TFT, 87
- setCursor
  - TFT, 87
- setCursorX
  - TFT, 87
- setCursorY
  - TFT, 87
- setFont
  - TFT, 87
- setFontScaleX

- TFT, [88](#)
- setFontScaleY
  - TFT, [88](#)
- 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](#)
  - S6D0164, [61](#)
  - SSD1289, [67](#)
  - SSD1963, [70](#)
  - ST7735, [74](#)
  - TFT, [88](#)
- setRotation
  - Aggregator, [10](#)
  - AnalogTouch, [12](#)
  - BD663474, [15](#)
  - Framebuffer, [29](#)
  - HD44780, [39](#)
  - HX8357, [43](#)
  - ILI9340, [46](#)
  - KS0108, [50](#)
  - LEDMatrix, [52](#)
  - S6D0164, [61](#)
  - SSD1289, [68](#)
  - SSD1963, [71](#)
  - ST7735, [74](#)
  - TFT, [88](#)
  - Touch, [119](#)
  - XPT2046, [121](#)
- setTextColor
  - TFT, [88](#), [89](#)
- setTextWrap
  - TFT, [89](#)
- sprite, [62](#)
  - animdir, [63](#)
  - currentframe, [63](#)
  - data, [63](#)
  - frames, [63](#)
  - height, [63](#)
  - next, [63](#)
  - store, [63](#)
  - transparent, [63](#)
  - width, [63](#)
  - xpos, [64](#)
  - ypos, [64](#)
- store
  - sprite, [63](#)
- streamCommand16
  - RawPar, [56](#)
- TFTCommunicator, [94](#)
- TFTPar16, [99](#)
- TFTPar4, [102](#)
- TFTPar8, [106](#)
- TFTPMP, [111](#)
- TFTSoftSPI, [115](#), [116](#)
- streamCommand32
  - RawPar, [56](#)
  - TFTCommunicator, [94](#)
  - TFTPar16, [99](#)
  - TFTPar4, [102](#)
  - TFTPar8, [107](#)
  - TFTPMP, [111](#)
  - TFTSoftSPI, [116](#)
- streamCommand8
  - RawPar, [56](#)
  - TFTCommunicator, [94](#)
  - TFTPar16, [99](#)
  - TFTPar4, [102](#)
  - TFTPar8, [107](#)
  - TFTPMP, [111](#)
  - TFTSoftSPI, [116](#)
- streamData16
  - RawPar, [56](#)
  - TFTCommunicator, [94](#)
  - TFTPar16, [99](#)
  - TFTPar4, [103](#)
  - TFTPar8, [107](#)
  - TFTPMP, [111](#)
  - TFTSoftSPI, [116](#)
- streamData32
  - RawPar, [56](#)
  - TFTCommunicator, [95](#)
  - TFTPar16, [99](#), [100](#)
  - TFTPar4, [103](#)
  - TFTPar8, [107](#)
  - TFTPMP, [112](#)
  - TFTSoftSPI, [116](#)
- streamData8
  - RawPar, [56](#)
  - TFTCommunicator, [95](#)
  - TFTPar16, [100](#)
  - TFTPar4, [103](#)
  - TFTPar8, [107](#), [108](#)
  - TFTPMP, [112](#)
  - TFTSoftSPI, [117](#)
- streamEnd
  - RawPar, [56](#)
  - TFTCommunicator, [95](#)
  - TFTPar16, [100](#)
  - TFTPar4, [103](#)
  - TFTPar8, [108](#)
  - TFTPMP, [112](#)
  - TFTSoftSPI, [117](#)
- streamStart
  - RawPar, [56](#)
  - TFTCommunicator, [95](#)
  - TFTPar16, [100](#)

- TFTPar4, 103
- TFTPar8, 108
- TFTPMP, 112
- TFTSoftSPI, 117
- stringHeight
  - TFT, 89
- stringWidth
  - TFT, 89
- TFT, 75
  - \_comm, 90
  - \_height, 90
  - \_width, 90
  - bgColorAt, 78
  - clearClipping, 79
  - closeWindow, 79
  - color565, 79
  - colorAt, 79
  - cursor\_x, 91
  - cursor\_y, 91
  - deltaE, 79
  - deltaOrth, 79
  - displayOff, 80
  - displayOn, 80
  - drawBitmap, 80
  - drawChar, 80
  - drawCircle, 81
  - drawCircleHelper, 81
  - drawHorizontalLine, 81
  - drawLine, 81
  - drawRGB, 82
  - drawRGBA, 82
  - drawRectangle, 81
  - drawRoundRect, 82
  - drawTriangle, 82
  - drawVerticalLine, 82
  - fatalError, 83
  - fillCircle, 83
  - fillCircleHelper, 83
  - fillRectangle, 83
  - fillRoundRect, 83
  - fillScreen, 84
  - fillTriangle, 84
  - font, 91
  - font\_scale\_x, 91
  - font\_scale\_y, 91
  - getCursor, 84
  - getCursorX, 84
  - getCursorY, 84
  - getHeight, 85
  - getTextColor, 85
  - getWidth, 85
  - initializeDevice, 85
  - invertDisplay, 85
  - invertTextColor, 86
  - mix, 86
  - openWindow, 86
  - rgb2hsv, 86
  - rgb2xyz, 86
  - rotation, 91
  - setClipping, 87
  - setCursor, 87
  - setCursorX, 87
  - setCursorY, 87
  - setFont, 87
  - setFontScaleX, 88
  - setFontScaleY, 88
  - setPixel, 88
  - setRotation, 88
  - setTextColor, 88, 89
  - setTextWrap, 89
  - stringHeight, 89
  - stringWidth, 89
  - TFT, 78
  - textbgcolor, 91
  - textcolor, 91
  - TFT, 78
  - windowData, 89, 90
  - wrap, 91
  - write, 90
  - xyz2lab, 90
- TFTCommunicator, 91
  - blockData, 93
  - initializeDevice, 93
  - nativeWidth, 93
  - readCommand16, 93
  - readCommand32, 93
  - readCommand8, 93
  - readData16, 93
  - readData32, 93
  - readData8, 94
  - streamCommand16, 94
  - streamCommand32, 94
  - streamCommand8, 94
  - streamData16, 94
  - streamData32, 95
  - streamData8, 95
  - streamEnd, 95
  - streamStart, 95
  - writeCommand16, 95
  - writeCommand32, 95
  - writeCommand8, 95
  - writeData16, 96
  - writeData32, 96
  - writeData8, 96
- TFTPMP, 109
  - blockData, 110
  - initializeDevice, 110
  - nativeWidth, 110
  - readCommand16, 110
  - readCommand32, 110
  - readCommand8, 110
  - readData16, 110
  - readData32, 110
  - readData8, 111
  - streamCommand16, 111
  - streamCommand32, 111

- streamCommand8, 111
- streamData16, 111
- streamData32, 112
- streamData8, 112
- streamEnd, 112
- streamStart, 112
- writeCommand16, 112
- writeCommand32, 112
- writeCommand8, 112
- writeData16, 113
- writeData32, 113
- writeData8, 113
- TFTPar16, 96
  - blockData, 97, 98
  - initializeDevice, 98
  - ltheadAdapter, 101
  - nativeWidth, 98
  - readCommand16, 98
  - readCommand32, 98
  - readCommand8, 98
  - readData16, 98
  - readData32, 98
  - readData8, 98
  - streamCommand16, 99
  - streamCommand32, 99
  - streamCommand8, 99
  - streamData16, 99
  - streamData32, 99, 100
  - streamData8, 100
  - streamEnd, 100
  - streamStart, 100
  - TFTPar16, 97
  - TFTPar16, 97
  - writeCommand16, 100
  - writeCommand32, 100
  - writeCommand8, 100
  - writeData16, 100
  - writeData32, 101
  - writeData8, 101
- TFTPar4, 101
  - nativeWidth, 102
  - streamCommand16, 102
  - streamCommand32, 102
  - streamCommand8, 102
  - streamData16, 103
  - streamData32, 103
  - streamData8, 103
  - streamEnd, 103
  - streamStart, 103
  - TFTPar4, 102
  - TFTPar4, 102
  - writeCommand16, 103
  - writeCommand32, 103
  - writeCommand8, 103
  - writeData16, 103
  - writeData32, 104
  - writeData8, 104
- TFTPar8, 104
  - blockData, 105
  - initializeDevice, 105
  - nativeWidth, 106
  - readCommand16, 106
  - readCommand32, 106
  - readCommand8, 106
  - readData16, 106
  - readData32, 106
  - readData8, 106
  - streamCommand16, 106
  - streamCommand32, 107
  - streamCommand8, 107
  - streamData16, 107
  - streamData32, 107
  - streamData8, 107, 108
  - streamEnd, 108
  - streamStart, 108
  - TFTPar8, 105
  - TFTPar8, 105
  - writeCommand16, 108
  - writeCommand32, 108
  - writeCommand8, 108
  - writeData16, 108
  - writeData32, 108
  - writeData8, 108
- TFTSoftSPI, 113
  - blockData, 114
  - initializeDevice, 115
  - nativeWidth, 115
  - readCommand16, 115
  - readCommand32, 115
  - readCommand8, 115
  - readData16, 115
  - readData32, 115
  - readData8, 115
  - streamCommand16, 115, 116
  - streamCommand32, 116
  - streamCommand8, 116
  - streamData16, 116
  - streamData32, 116
  - streamData8, 117
  - streamEnd, 117
  - streamStart, 117
  - TFTSoftSPI, 114
  - TFTSoftSPI, 114
  - writeCommand16, 117
  - writeCommand32, 117
  - writeCommand8, 117
  - writeData16, 117
  - writeData32, 117
  - writeData8, 118
- textbgcolor
  - TFT, 91
- textcolor
  - TFT, 91
- Touch, 118
  - \_comm, 120
  - \_height, 120

- [\\_width](#), [120](#)
  - [initializeDevice](#), [119](#)
  - [isPressed](#), [119](#)
  - [pressure](#), [119](#)
  - [sample](#), [119](#)
  - [setRotation](#), [119](#)
  - [Touch](#), [118](#), [119](#)
  - [x](#), [120](#)
  - [y](#), [120](#)
- [transparent](#)
  - [sprite](#), [63](#)
- [TypeB](#)
  - [ST7735](#), [75](#)
- [Width](#)
  - [SSD1963](#), [71](#)
  - [ST7735](#), [75](#)
- [width](#)
  - [sprite](#), [63](#)
- [windowData](#)
  - [HX8357](#), [43](#)
  - [S6D0164](#), [61](#)
  - [SSD1289](#), [68](#)
  - [TFT](#), [89](#), [90](#)
- [wrap](#)
  - [TFT](#), [91](#)
- [write](#)
  - [TFT](#), [90](#)
- [writeCommand16](#)
  - [RawPar](#), [56](#)
  - [TFTCommunicator](#), [95](#)
  - [TFTPar16](#), [100](#)
  - [TFTPar4](#), [103](#)
  - [TFTPar8](#), [108](#)
  - [TFTPMP](#), [112](#)
  - [TFTSoftSPI](#), [117](#)
- [writeCommand32](#)
  - [RawPar](#), [57](#)
  - [TFTCommunicator](#), [95](#)
  - [TFTPar16](#), [100](#)
  - [TFTPar4](#), [103](#)
  - [TFTPar8](#), [108](#)
  - [TFTPMP](#), [112](#)
  - [TFTSoftSPI](#), [117](#)
- [writeCommand8](#)
  - [RawPar](#), [57](#)
  - [TFTCommunicator](#), [95](#)
  - [TFTPar16](#), [100](#)
  - [TFTPar4](#), [103](#)
  - [TFTPar8](#), [108](#)
  - [TFTPMP](#), [112](#)
  - [TFTSoftSPI](#), [117](#)
- [writeData16](#)
  - [RawPar](#), [57](#)
  - [TFTCommunicator](#), [96](#)
  - [TFTPar16](#), [100](#)
  - [TFTPar4](#), [103](#)
  - [TFTPar8](#), [108](#)
  - [TFTPMP](#), [113](#)
  - [TFTSoftSPI](#), [117](#)
- [writeData32](#)
  - [RawPar](#), [57](#)
  - [TFTCommunicator](#), [96](#)
  - [TFTPar16](#), [101](#)
  - [TFTPar4](#), [104](#)
  - [TFTPar8](#), [108](#)
  - [TFTPMP](#), [113](#)
  - [TFTSoftSPI](#), [117](#)
- [writeData8](#)
  - [RawPar](#), [57](#)
  - [TFTCommunicator](#), [96](#)
  - [TFTPar16](#), [101](#)
  - [TFTPar4](#), [104](#)
  - [TFTPar8](#), [108](#)
  - [TFTPMP](#), [113](#)
  - [TFTSoftSPI](#), [118](#)
- [x](#)
  - [AnalogTouch](#), [12](#)
  - [Touch](#), [120](#)
  - [XPT2046](#), [122](#)
- [XPT2046](#), [120](#)
  - [initializeDevice](#), [121](#)
  - [isPressed](#), [121](#)
  - [sample](#), [121](#)
  - [setRotation](#), [121](#)
  - [x](#), [122](#)
  - [XPT2046](#), [121](#)
  - [XPT2046](#), [121](#)
  - [y](#), [122](#)
- [xpos](#)
  - [sprite](#), [64](#)
- [xyz2lab](#)
  - [TFT](#), [90](#)
- [y](#)
  - [AnalogTouch](#), [12](#)
  - [Touch](#), [120](#)
  - [XPT2046](#), [122](#)
- [ypos](#)
  - [sprite](#), [64](#)