DisplayCore

Generated by Doxygen 1.8.8

Sun Mar 27 2016 12:39:48

Contents

Chapter 1

BMPFile

Bitmap File Renderer (from SD card) for TFT library

Chapter 2

DisplayCore

http://DisplayCore.org

The DisplayCore system builds on the popular TFT library to create a fully modular and easily expandable video display system for chipKIT boards.

Instead of just one huge monolithic library the DisplayCore system is split into a number of much smaller libraries, each one handling a specific task. Each display has its own dedicated library, as do the different touch input systems. The idea of an abstracted connector system has been dropped in favour of embedding the connectivity into the screen driver.

All this means:

- Smaller code - you only include the parts you need
- Faster compilation - you don't need to compile code you won't be using
- Faster display access - removing the communication abstraction layer makes communication much faster
- Easy to support more displays - not just TFT screens, but any display technology can now be supported

Display drivers and tested devices:

- BD663474
  - WaveShare LCD22
- Goldelox
  - uOLED-128-G2
- HX8347D
  - WaveShare 2.8" Touchscreen TFT (SPI)
- ILI9163
  - Generic 1.44" SPI 128∗128 V1.1
- ILI9340
  - Adafruit 2.2" TFT
- ILI9481
  - HY-3.2TFT
- KS0108
  - Monochrome graphical LCDs

- LM6800
  - 4-chip based KS0108
- NativeFB
  - Linux framebuffer device (Pi, Armadillo, etc)
- PG25664CG
  - 256x64 grey-scale OLED
- Picadillo
  - Picadillo-35T
- SDL
  - Simple DirectMedia Layer (Linux X windows interface)
- SSD1289
  - TFT_320QVT
- SSD1306
  - Basic IO Shield OLED and PmodOLED
- SSD1963_7
  - Unbranded 7" TFT
- ST7735
  - Adafruit 1.8" TFT
- VGA
  - Emulated VGA device using SPI and DMA
- VLCD
  - UECIDE VirtualLCD

Framebuffer drivers:

- Framebuffer332 - RGB332 (8 bit) direct colour mapped framebuffer
- Framebuffer565 - RGB565 (16 bit) direct colour mapped framebuffer

Touch screen drivers:

- AnalogTouch
  - Uses the PIC's internal ADC to read a 4-wire resistive touch panel.
- LinuxEvent
  - Read mouse and keyboard events from the Linux input system
- XPT2046
  - Common SPI touchscreen controller

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 5

Data Structure Documentation

5.1 __attribute__ Struct Reference

Public Member Functions

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

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

- DisplayCore/DisplayCore.h

5.2 AdjustHSV Class Reference

Inheritance diagram for AdjustHSV:

Collaboration diagram for AdjustHSV:

Public Member Functions

- color_t **function** (color_t)
- void **adjustHue** (int16_t h)
- void **adjustSaturation** (int16_t s)
- void **adjustValue** (int16_t v)

Additional Inherited Members

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

- Filters/AdjustHSV.h
- Filters/AdjustHSV.cpp

5.3 AnalogTouch Class Reference

Inheritance diagram for AnalogTouch:

Collaboration diagram for AnalogTouch:

Public Member Functions

- **AnalogTouch** (uint8_t xl, uint8_t xr, uint8_t yu, uint8_t yd, int w, int h)
- void sample ()
- int **getSample** (uint8_t)
- int x ()
- int y ()
- int rawX ()
- int **rawY** ()
- void **offsetX** (int ox)
- void **offsetY** (int oy)
- void **scaleX** (float sx)
- void **scaleY** (float sy)
- boolean **isPressed** ()
- void initializeDevice ()
- int pressure ()
- void setRotation (int r)

Additional Inherited Members

5.3.1 Member Function Documentation

5.3.1.1 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.

5.3.1.2 int 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.

5.3.1.3 int AnalogTouch::rawX ( ) [inline], [virtual]

Get pressed status

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

Reimplemented from Touch.

5.3.1.4 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.

5.3.1.5 void AnalogTouch::setRotation (int *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.

Implements Touch.

5.3.1.6 int AnalogTouch::x ( ) [virtual]

Get X coordinate

This returns the X coordinate of the current touch position.

Implements Touch.

5.3.1.7 int 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:

- Drivers/AnalogTouch/AnalogTouch.h
- Drivers/AnalogTouch/AnalogTouch.cpp

5.4 BD663474 Class Reference

Inheritance diagram for BD663474:

Collaboration diagram for BD663474:

Public Member Functions

- **BD663474** (DSPI ∗dspi, uint8_t cs, uint8_t rs, uint8_t reset)
- **BD663474** (DSPI &dspi, uint8_t cs, uint8_t rs, uint8_t reset)
- void **setAddrWindow** (int x0, int y0, int x1, int y1)
- void fillScreen (color_t color)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)

- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- void **startDisplay** ()
- virtual void initializeDevice ()
- virtual void **command** (uint16_t)
- virtual void **data** (uint16_t)

Static Public Attributes

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

Additional Inherited Members

5.4.1 Member Function Documentation

5.4.1.1 void BD663474::displayOff ( ) [inline], [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.4.1.2 void BD663474::displayOn ( ) [inline], [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.4.1.3 void BD663474::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.4.1.4 void BD663474::drawVerticalLine (int *x,* int *y,* int *h,* color_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);
```

Reimplemented from DisplayCore.

5.4.1.5 void BD663474::fillRectangle (int *x,* int *y,* int *w,* int *h,* color_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 DisplayCore.

5.4.1.6 void BD663474::fillScreen (color_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 DisplayCore.

5.4.1.7 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 DisplayCore.

5.4.1.8 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 DisplayCore.

5.4.1.9 void BD663474::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.4.1.10 void BD663474::setRotation (int *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 DisplayCore.

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

- Drivers/BD663474/BD663474.h
- Drivers/BD663474/BD663474.cpp

5.5 BinaryVector Class Reference

```
#include <BinaryVector.h>
```

Inheritance diagram for BinaryVector:

Collaboration diagram for BinaryVector:

Public Member Functions

- **BinaryVector** (const uint16_t *program)
- void **draw** (DisplayCore *dc, int x, int y)
- void **draw** (DisplayCore *dc, int x, int y, color_t)
- void **drawTransformed** (DisplayCore *dc, int x, int y, int)
- void **drawTransformed** (DisplayCore *dc, int x, int y, int, color_t)

Static Public Attributes

- static const uint16_t **END_PROG** = 0x0000
- static const uint16_t **SET_PIXEL** = 0x0103
- static const uint16_t **DRAW_LINE** = 0x0205
- static const uint16_t **DRAW_BOX** = 0x0305
- static const uint16_t **FILL_BOX** = 0x0405
- static const uint16_t **DRAW_BEZIER** = 0x050A

Additional Inherited Members

5.5.1 Detailed Description

Binary Vector Drawing Class

The Binary Vector is a simple Vector Graphics drawing class. It takes a "program" of instructions on how to draw a shape. The program is a simple linear list of drawing operations to perform.

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

- ImageReaders/BinaryVector/BinaryVector.h
- ImageReaders/BinaryVector/BinaryVector.cpp

5.6 BitmapFileHeader Struct Reference

Data Fields

- 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 files:

- ImageReaders/BMP/BMP.h
- Utilities/ScreenDump/ScreenDump.cpp

5.7 BitmapInfoHeader Struct Reference

Data Fields

- 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**
- uint16_t **biCSType**
- uint32_t **biRedX**
- uint32_t **biRedY**
- uint32_t **biRedZ**
- uint32_t **biGreenX**

- uint32_t **biGreenY**
- uint32_t **biGreenZ**
- uint32_t **biBlueX**
- uint32_t **biBlueY**
- uint32_t **biBlueZ**
- uint32_t **biGammaRed**
- uint32_t **biGammaGreen**
- uint32_t **biGammaBlue**

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

- ImageReaders/BMP/BMP.h
- Utilities/ScreenDump/ScreenDump.cpp

5.8 BitmapPixel24 Struct Reference

Data Fields

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

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

- ImageReaders/BMP/BMP.h

5.9 BitmapPixel32 Struct Reference

Public Member Functions

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

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

- ImageReaders/BMP/BMP.h

5.10 LCARS::Block Class Reference

Inheritance diagram for LCARS::Block:

Collaboration diagram for LCARS::Block:

Public Member Functions

- **Block** (Touch &ts, DisplayCore &dev, int x, int y, int w, int h, color_t col, const char *txt)
- void setPixel (int x, int y, color_t c)
- void **draw** (DisplayCore *dev, int x, int y)
- void initializeDevice ()

Additional Inherited Members

5.10.1 Member Function Documentation

5.10.1.1 void LCARS::Block::initializeDevice ( ) [inline], [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 Image.

5.10.1.2 void LCARS::Block::setPixel (int *x*, int *y*, color_t *color*) [inline], [virtual]

Draw a pixel

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

Example:

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

Reimplemented from Image.

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.11 BMP Class Reference

Inheritance diagram for BMP:

Collaboration diagram for BMP:

Public Member Functions

- **BMP** (const char *data)
- void **draw** (DisplayCore *dev, int x, int y)
- void **draw** (DisplayCore *dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore *dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore *dev, int x, int y, int transform, color_t t)

Data Fields

- 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:

- ImageReaders/BMP/BMP.h
- ImageReaders/BMP/BMP.cpp

5.12 BMPFile Class Reference

Inheritance diagram for BMPFile:

Collaboration diagram for BMPFile:

Public Member Functions

- **BMPFile** (File &file)
- void **draw** (DisplayCore ∗dev, int x, int y)
- void **draw** (DisplayCore ∗dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore ∗dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore ∗dev, int x, int y, int transform, color_t t)
- virtual int getWidth ()
- virtual int getHeight ()

Data Fields

- struct BitmapFileHeader **_header**
- struct BitmapInfoHeader **_info**
- File ∗ **_file**
- struct BitmapPixel32 **_palette** [256]
- uint16_t **_paletteSize**
- uint32_t **_chunkSize**
- uint32_t **_spos**

Additional Inherited Members

5.12.1 Member Function Documentation

5.12.1.1 int BMPFile::getHeight ( ) [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from Image.

5.12.1.2 int BMPFile::getWidth ( ) [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from Image.

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

- ImageReaders/BMPFile/BMPFile.h
- ImageReaders/BMPFile/BMPFile.cpp

5.13 Brightness Class Reference

Inheritance diagram for Brightness:

Collaboration diagram for Brightness:

Public Member Functions

- color_t **function** (color_t)
- void **adjustBrightness** (int16_t b)

Additional Inherited Members

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

- Filters/Brightness.h
- Filters/Brightness.cpp

5.14 Color Class Reference

Static Public Attributes

- static const color_t **Snow** = RGB(255,250,250)
- static const color_t **GhostWhite** = RGB(248,248,255)
- static const color_t **WhiteSmoke** = RGB(245,245,245)
- static const color_t **Gainsboro** = RGB(220,220,220)
- static const color_t **FloralWhite** = RGB(255,250,240)
- static const color_t **OldLace** = RGB(253,245,230)
- static const color_t **Linen** = RGB(250,240,230)
- static const color_t **AntiqueWhite** = RGB(250,235,215)
- static const color_t **PapayaWhip** = RGB(255,239,213)
- static const color_t **BlanchedAlmond** = RGB(255,235,205)
- static const color_t **Bisque** = RGB(255,228,196)
- static const color_t **PeachPuff** = RGB(255,218,185)
- static const color_t **NavajoWhite** = RGB(255,222,173)
- static const color_t **Moccasin** = RGB(255,228,181)
- static const color_t **Cornsilk** = RGB(255,248,220)

- static const color_t **Ivory** = RGB(255,255,240)
- static const color_t **LemonChiffon** = RGB(255,250,205)
- static const color_t **Seashell** = RGB(255,245,238)
- static const color_t **Honeydew** = RGB(240,255,240)
- static const color_t **MintCream** = RGB(245,255,250)
- static const color_t **Azure** = RGB(240,255,255)
- static const color_t **AliceBlue** = RGB(240,248,255)
- static const color_t **Lavender** = RGB(230,230,250)
- static const color_t **LavenderBlush** = RGB(255,240,245)
- static const color_t **MistyRose** = RGB(255,228,225)
- static const color_t **White** = RGB(255,255,255)
- static const color_t **Black** = RGB(0, 0, 0)
- static const color_t **DarkSlateGray** = RGB(47, 79, 79)
- static const color_t **DimGray** = RGB(105,105,105)
- static const color_t **SlateGray** = RGB(112,128,144)
- static const color_t **LightSlateGray** = RGB(119,136,153)
- static const color_t **Gray** = RGB(190,190,190)
- static const color_t **LightGray** = RGB(211,211,211)
- static const color_t **MidnightBlue** = RGB(25, 25,112)
- static const color_t **Navy** = RGB(0, 0,128)
- static const color_t **NavyBlue** = RGB(0, 0,128)
- static const color_t **CornflowerBlue** = RGB(100,149,237)
- static const color_t **DarkSlateBlue** = RGB(72, 61,139)
- static const color_t **SlateBlue** = RGB(106, 90,205)
- static const color_t **MediumSlateBlue** = RGB(123,104,238)
- static const color_t **LightSlateBlue** = RGB(132,112,255)
- static const color_t **MediumBlue** = RGB(0, 0,205)
- static const color_t **RoyalBlue** = RGB(65,105,225)
- static const color_t **Blue** = RGB(0, 0,255)
- static const color_t **DodgerBlue** = RGB(30,144,255)
- static const color_t **DeepSkyBlue** = RGB(0,191,255)
- static const color_t **SkyBlue** = RGB(135,206,235)
- static const color_t **LightSkyBlue** = RGB(135,206,250)
- static const color_t **SteelBlue** = RGB(70,130,180)
- static const color_t **LightSteelBlue** = RGB(176,196,222)
- static const color_t **LightBlue** = RGB(173,216,230)
- static const color_t **PowderBlue** = RGB(176,224,230)
- static const color_t **PaleTurquoise** = RGB(175,238,238)
- static const color_t **DarkTurquoise** = RGB(0,206,209)
- static const color_t **MediumTurquoise** = RGB(72,209,204)
- static const color_t **Turquoise** = RGB(64,224,208)
- static const color_t **Cyan** = RGB(0,255,255)
- static const color_t **LightCyan** = RGB(224,255,255)
- static const color_t **CadetBlue** = RGB(95,158,160)
- static const color_t **MediumAquamarine** = RGB(102,205,170)
- static const color_t **Aquamarine** = RGB(127,255,212)
- static const color_t **DarkGreen** = RGB(0,100, 0)
- static const color_t **DarkOliveGreen** = RGB(85,107, 47)
- static const color_t **DarkSeaGreen** = RGB(143,188,143)
- static const color_t **SeaGreen** = RGB(46,139, 87)
- static const color_t **MediumSeaGreen** = RGB(60,179,113)
- static const color_t **LightSeaGreen** = RGB(32,178,170)
- static const color_t **PaleGreen** = RGB(152,251,152)
- static const color_t **SpringGreen** = RGB(0,255,127)
- static const color_t **LawnGreen** = RGB(124,252, 0)

- static const color_t **Green** = RGB(0,255, 0)
- static const color_t **Chartreuse** = RGB(127,255, 0)
- static const color_t **MediumSpringGreen** = RGB(0,250,154)
- static const color_t **GreenYellow** = RGB(173,255, 47)
- static const color_t **LimeGreen** = RGB(50,205, 50)
- static const color_t **YellowGreen** = RGB(154,205, 50)
- static const color_t **ForestGreen** = RGB(34,139, 34)
- static const color_t **OliveDrab** = RGB(107,142, 35)
- static const color_t **DarkKhaki** = RGB(189,183,107)
- static const color_t **Khaki** = RGB(240,230,140)
- static const color_t **PaleGoldenrod** = RGB(238,232,170)
- static const color_t **LightGoldenrodYellow** = RGB(250,250,210)
- static const color_t **LightYellow** = RGB(255,255,224)
- static const color_t **Yellow** = RGB(255,255, 0)
- static const color_t **Gold** = RGB(255,215, 0)
- static const color_t **LightGoldenrod** = RGB(238,221,130)
- static const color_t **Goldenrod** = RGB(218,165, 32)
- static const color_t **DarkGoldenrod** = RGB(184,134, 11)
- static const color_t **RosyBrown** = RGB(188,143,143)
- static const color_t **IndianRed** = RGB(205, 92, 92)
- static const color_t **SaddleBrown** = RGB(139, 69, 19)
- static const color_t **Sienna** = RGB(160, 82, 45)
- static const color_t **Peru** = RGB(205,133, 63)
- static const color_t **Burlywood** = RGB(222,184,135)
- static const color_t **Beige** = RGB(245,245,220)
- static const color_t **Wheat** = RGB(245,222,179)
- static const color_t **SandyBrown** = RGB(244,164, 96)
- static const color_t **Tan** = RGB(210,180,140)
- static const color_t **Chocolate** = RGB(210,105, 30)
- static const color_t **Firebrick** = RGB(178, 34, 34)
- static const color_t **Brown** = RGB(165, 42, 42)
- static const color_t **DarkSalmon** = RGB(233,150,122)
- static const color_t **Salmon** = RGB(250,128,114)
- static const color_t **LightSalmon** = RGB(255,160,122)
- static const color_t **Orange** = RGB(255,165, 0)
- static const color_t **DarkOrange** = RGB(255,140, 0)
- static const color_t **Coral** = RGB(255,127, 80)
- static const color_t **LightCoral** = RGB(240,128,128)
- static const color_t **Tomato** = RGB(255, 99, 71)
- static const color_t **OrangeRed** = RGB(255, 69, 0)
- static const color_t **Red** = RGB(255, 0, 0)
- static const color_t **HotPink** = RGB(255,105,180)
- static const color_t **DeepPink** = RGB(255, 20,147)
- static const color_t **Pink** = RGB(255,192,203)
- static const color_t **LightPink** = RGB(255,182,193)
- static const color_t **PaleVioletRed** = RGB(219,112,147)
- static const color_t **Maroon** = RGB(176, 48, 96)
- static const color_t **MediumVioletRed** = RGB(199, 21,133)
- static const color_t **VioletRed** = RGB(208, 32,144)
- static const color_t **Magenta** = RGB(255, 0,255)
- static const color_t **Violet** = RGB(238,130,238)
- static const color_t **Plum** = RGB(221,160,221)
- static const color_t **Orchid** = RGB(218,112,214)
- static const color_t **MediumOrchid** = RGB(186, 85,211)
- static const color_t **DarkOrchid** = RGB(153, 50,204)

- static const color_t **DarkViolet** = RGB(148, 0,211)
- static const color_t **BlueViolet** = RGB(138, 43,226)
- static const color_t **Purple** = RGB(160, 32,240)
- static const color_t **MediumPurple** = RGB(147,112,219)
- static const color_t **Thistle** = RGB(216,191,216)
- static const color_t **Gray0** = RGB(0, 0, 0)
- static const color_t **Gray1** = RGB(3, 3, 3)
- static const color_t **Gray2** = RGB(5, 5, 5)
- static const color_t **Gray3** = RGB(8, 8, 8)
- static const color_t **Gray4** = RGB(10, 10, 10)
- static const color_t **Gray5** = RGB(13, 13, 13)
- static const color_t **Gray6** = RGB(15, 15, 15)
- static const color_t **Gray7** = RGB(18, 18, 18)
- static const color_t **Gray8** = RGB(20, 20, 20)
- static const color_t **Gray9** = RGB(23, 23, 23)
- static const color_t **Gray10** = RGB(26, 26, 26)
- static const color_t **Gray11** = RGB(28, 28, 28)
- static const color_t **Gray12** = RGB(31, 31, 31)
- static const color_t **Gray13** = RGB(33, 33, 33)
- static const color_t **Gray14** = RGB(36, 36, 36)
- static const color_t **Gray15** = RGB(38, 38, 38)
- static const color_t **Gray16** = RGB(41, 41, 41)
- static const color_t **Gray17** = RGB(43, 43, 43)
- static const color_t **Gray18** = RGB(46, 46, 46)
- static const color_t **Gray19** = RGB(48, 48, 48)
- static const color_t **Gray20** = RGB(51, 51, 51)
- static const color_t **Gray21** = RGB(54, 54, 54)
- static const color_t **Gray22** = RGB(56, 56, 56)
- static const color_t **Gray23** = RGB(59, 59, 59)
- static const color_t **Gray24** = RGB(61, 61, 61)
- static const color_t **Gray25** = RGB(64, 64, 64)
- static const color_t **Gray26** = RGB(66, 66, 66)
- static const color_t **Gray27** = RGB(69, 69, 69)
- static const color_t **Gray28** = RGB(71, 71, 71)
- static const color_t **Gray29** = RGB(74, 74, 74)
- static const color_t **Gray30** = RGB(77, 77, 77)
- static const color_t **Gray31** = RGB(79, 79, 79)
- static const color_t **Gray32** = RGB(82, 82, 82)
- static const color_t **Gray33** = RGB(84, 84, 84)
- static const color_t **Gray34** = RGB(87, 87, 87)
- static const color_t **Gray35** = RGB(89, 89, 89)
- static const color_t **Gray36** = RGB(92, 92, 92)
- static const color_t **Gray37** = RGB(94, 94, 94)
- static const color_t **Gray38** = RGB(97, 97, 97)
- static const color_t **Gray39** = RGB(99, 99, 99)
- static const color_t **Gray40** = RGB(102,102,102)
- static const color_t **Gray41** = RGB(105,105,105)
- static const color_t **Gray42** = RGB(107,107,107)
- static const color_t **Gray43** = RGB(110,110,110)
- static const color_t **Gray44** = RGB(112,112,112)
- static const color_t **Gray45** = RGB(115,115,115)
- static const color_t **Gray46** = RGB(117,117,117)
- static const color_t **Gray47** = RGB(120,120,120)
- static const color_t **Gray48** = RGB(122,122,122)
- static const color_t **Gray49** = RGB(125,125,125)

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

- static const color_t **DarkRed** = RGB(139, 0, 0)
- static const color_t **LightGreen** = RGB(144,238,144)
- static const color_t **Sepia** = RGB(112, 66, 20)

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

- DisplayCore/Color.h

5.15 Contrast Class Reference

Inheritance diagram for Contrast:

Collaboration diagram for Contrast:

Public Member Functions

- color_t **function** (color_t)
- void **adjustContrast** (int16_t c)

Additional Inherited Members

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

- Filters/Contrast.h
- Filters/Contrast.cpp

5.16 coord Struct Reference

Data Fields

- int **x**
- int **y**

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

- DisplayCore/DisplayCore.h

5.17 DisplayCore Class Reference

Inheritance diagram for DisplayCore:

Collaboration diagram for DisplayCore:

Public Member Functions

- DisplayCore ()
- virtual void **translateCoordinates** (int *x, int *y)
- virtual void **drawBezier** (int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3, int resolution, color_t color)
- virtual void **fillBezier** (int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3, int resolution, color_t color)
- virtual void startBuffer ()

- virtual void endBuffer ()
- virtual void enableBacklight ()
- virtual void disableBacklight ()
- virtual void setBacklight (int b)
- virtual int getWidth ()
- virtual int getHeight ()
- uint32_t **color2rgb** (color_t c)

Drawing Functions

These functions draw pretty shapes on the screen.

- virtual void drawCircle (int x0, int y0, int r, color_t color)
- virtual void fillCircle (int x0, int y0, int r, color_t color)
- virtual void drawLine (int x0, int y0, int x1, int y1, color_t color)
- virtual void drawLine (int x0, int y0, int x1, int y1, int width, color_t color)
- virtual void drawRectangle (int x, int y, int w, int h, color_t color)
- virtual void drawRoundRect (int x, int y, int w, int h, int r, color_t color)
- virtual void fillRoundRect (int x, int y, int w, int h, int r, color_t color)
- virtual void drawTriangle (int x0, int y0, int x1, int y1, int x2, int y2, color_t color)
- virtual void fillTriangle (int x0, int y0, int x1, int y1, int x2, int y2, color_t color)
- virtual void fillScreen (color_t color)
- virtual void fillRectangle (int x, int y, int w, int h, color_t color)
- void setClipping (int x0, int y0, int x1, int y1)
- void clearClipping ()

Image drawing

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

- virtual void drawBitmap (int x, int y, const uint8_t *bitmap, int w, int h, color_t color)
- virtual void drawRGB (int x, int y, const color_t *bitmap, int w, int h)
- virtual void drawRGBA (int x, int y, const color_t *bitmap, int w, int h, color_t trans)

Text handing functions

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

- virtual void setCursor (int x, int y)
- virtual void setCursorX (int x)
- virtual void setCursorY (int y)
- virtual int getCursorX ()
- virtual int getCursorY ()
- virtual int getCursor (boolean x)
- virtual void setTextColor (color_t c)
- virtual void setTextColor (color_t fg, color_t bg)
- virtual color_t getTextColor ()
- virtual void invertTextColor ()
- virtual void setTextWrap (boolean w)
- virtual void setFont (const uint8_t *f)
- virtual int stringWidth (const char *text)
- virtual int stringHeight (const char *text)
- virtual void write (uint8_t c)
- virtual void **write** (const uint8_t *buffer, size_t size)
- int drawChar (int x, int y, unsigned char c, color_t color, color_t bg)

Pure virtual functions

These are all functions that must be implemented in a TFT driver in order for it to function.

- virtual void setRotation (int rotation)=0
- virtual void setPixel (int x, int y, color_t color)=0
- virtual void drawHorizontalLine (int x, int y, int w, color_t color)
- virtual void drawVerticalLine (int x, int y, int h, color_t color)

- 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 (int x0, int y0, int x1, int y1)
- virtual void windowData (color_t d)
- virtual void windowData (color_t *d, int 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 (int x0, int y0, int r, int cornername, color_t color)
- void fillCircleHelper (int x0, int y0, int r, int cornername, int delta, color_t color)
- boolean **clipToScreen** (int &x, int &y, int &w, int &h)
- void fatalError (const char *title, const char *message)

Data Fields

- int cursor_x
- int cursor_y
- boolean wrap
- color_t textcolor
- color_t textbgcolor
- int _width
- int _height
- int rotation
- int **_clip_x0**
- int **_clip_x1**
- int **_clip_y0**
- int **_clip_y1**

Protected Attributes

- const uint8_t * font
- int **winx0**
- int **winy0**
- int **winx1**
- int **winy1**
- int **winpx**
- int **winpy**
- color_t **bgColor**

Colour handling

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

- virtual color_t color565 (uint8_t r, uint8_t g, uint8_t b)
- virtual color_t colorAt (int x, int y)
- virtual void **getRectangle** (int x, int y, int w, int h, color_t *buf)
- point3d rgb2xyz (color_t c)
- point3d xyz2lab (point3d c)
- float deltaE (point3d labA, point3d labB)
- uint32_t deltaOrth (color_t c1, color_t c2)
- color_t mix (color_t a, color_t b, int pct)
- static uint32_t rgb2hsv (color_t rgb)
- static color_t **hsv2rgb** (uint32_t hsv)

5.17.1 Constructor & Destructor Documentation

5.17.1.1 DisplayCore::DisplayCore ( )

The default constructor takes no parameters. It creates a blank screen class with no communication abilities.

5.17.2 Member Function Documentation

5.17.2.1 void DisplayCore::clearClipping ( )

Clear clipping boundaries

Remove the clipping boundary imposed by setClipping().
Example:

```
clearClipping();
```

5.17.2.2 void DisplayCore::closeWindow ( ) [virtual]

Close the window

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

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

Reimplemented in SDL, SSD1289, ILI9340, ILI9481, Goldelox, HX8347D, and ILI9163.

5.17.2.3 color_t DisplayCore::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);
```

5.17.2.4 color_t DisplayCore::colorAt (int *x*, int *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 SSD1963, ILI9481, and Framebuffer332.

5.17.2.5 float DisplayCore::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);
```

5.17.2.6 uint32_t DisplayCore::deltaOrth (color_t *c1*, color_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);
```

5.17.2.7 virtual void DisplayCore::disableBacklight ( ) [inline], [virtual]

Disable Back Light

For devices with their own backlight control this function will turn the backlight off.

Reimplemented in SSD1963.

5.17.2.8 virtual void DisplayCore::displayOff ( ) [pure virtual]

Turn off the display

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

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

Implemented in Image, SSD1963, KS0108_BB2, LM6800, SSD1289, ST7735, ILI9340, ILI9481, PG25664CG, SSD1306, SDL, Goldelox, VGA, KS0108_2, NativeFB, VLCD, KS0108, BD663474, HX8347D, and ILI9163.

5.17.2.9 virtual void DisplayCore::displayOn ( ) [pure virtual]

Turn on the display

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

Example:

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

Implemented in Image, SSD1963, KS0108_BB2, LM6800, SSD1289, ST7735, ILI9340, ILI9481, PG25664CG, SSD1306, SDL, Goldelox, VGA, KS0108_2, NativeFB, VLCD, KS0108, BD663474, HX8347D, and ILI9163.

5.17.2.10 void DisplayCore::drawBitmap (int *x,* int *y,* const uint8_t * *bitmap,* int *w,* int *h,* color_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,
    0b00000000};
tft.drawBitmap(100, 100, letterA, 8, 8, Color::Red);
```

5.17.2.11 int DisplayCore::drawChar (int *x,* int *y,* unsigned char *c,* color_t *color,* color_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);
```

5.17.2.12 void DisplayCore::drawCircle (int *x0,* int *y0,* int *r,* color_t *color*) [virtual]

Draw a circle

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

Example:

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

5.17.2.13 void DisplayCore::drawCircleHelper (int *x0,* int *y0,* int *r,* int *cornername,* color_t *color*)

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

5.17.2.14 void DisplayCore::drawHorizontalLine (int *x,* int *y,* int *w,* color_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 in SSD1963, LM6800, SSD1289, ST7735, ILI9340, ILI9481, Goldelox, BD663474, HX8347D, and ILI9163.

5.17.2.15 void DisplayCore::drawLine (int *x0,* int *y0,* int *x1,* int *y1,* color_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);
```

Reimplemented in Goldelox, and VLCD.

5.17.2.16 void DisplayCore::drawLine (int *x0,* int *y0,* int *x1,* int *y1,* int *width,* color_t *color*) [virtual]

Draw a thick 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).

Thickness is added using the highly inefficient "cheating" method of drawing circles instead of pixels.
Example:

```
tft.drawLine(10, 10, 40, 60, 4, Color::Green);
```

5.17.2.17 void DisplayCore::drawRectangle (int *x,* int *y,* int *w,* int *h,* color_t *color*) [virtual]

Draw a rectangle

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

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

Reimplemented in Goldelox.

5.17.2.18 void DisplayCore::drawRGB (int *x,* int *y,* const color_t * *bitmap,* int *w,* int *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);

5.17.2.19 void DisplayCore::drawRGBA (int *x,* int *y,* const color_t * *bitmap,* int *w,* int *h,* color_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);

5.17.2.20 void DisplayCore::drawRoundRect (int *x,* int *y,* int *w,* int *h,* int *r,* color_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);
```

5.17.2.21 void DisplayCore::drawTriangle (int *x0,* int *y0,* int *x1,* int *y1,* int *x2,* int *y2,* color_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);
```

5.17.2.22 void DisplayCore::drawVerticalLine (int *x,* int *y,* int *h,* color_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);
```

Reimplemented in SSD1963, LM6800, SSD1289, ST7735, ILI9340, ILI9481, Goldelox, BD663474, HX8347D, and ILI9163.

5.17.2.23 virtual void DisplayCore::enableBacklight ( ) [inline], [virtual]

Enable Back Light

For devices with their own backlight control this function will turn the backlight on. The brightness should be either the default brightness (typically full on) or the last brightness set with `setBacklight()`.

Reimplemented in SSD1963.

5.17.2.24 virtual void DisplayCore::endBuffer ( ) [inline], [virtual]

End buffered mode

Any changes that are pending will be pushed out to the screen. See `startBuffer()` for more information.

Reimplemented in KS0108_BB2, LM6800, PG25664CG, SDL, SSD1306, KS0108_2, and KS0108.

5.17.2.25 void DisplayCore::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");
```

5.17.2.26 void DisplayCore::fillCircle (int *x0*, int *y0*, int *radius*, color_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);
```

5.17.2.27 void DisplayCore::fillCircleHelper (int *x0*, int *y0*, int *r*, int *cornername*, int *delta*, color_t *color*)

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

5.17.2.28 void DisplayCore::fillRectangle (int *x*, int *y*, int *w*, int *h*, color_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, LM6800, SSD1289, ST7735, ILI9340, ILI9481, Goldelox, BD663474, HX8347D, and ILI9163.

5.17.2.29 void DisplayCore::fillRoundRect (int *x,* int *y,* int *w,* int *h,* int *r,* color_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);
```

5.17.2.30 void DisplayCore::fillScreen (color_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, LM6800, SSD1306, PG25664CG, SSD1289, ST7735, ILI9340, SDL, VGA, ILI9481, Goldelox, BD663474, HX8347D, Framebuffer332, and Framebuffer565.

5.17.2.31 void DisplayCore::fillTriangle (int *x0,* int *y0,* int *x1,* int *y1,* int *x2,* int *y2,* color_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);
```

5.17.2.32 int DisplayCore::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);
```

5.17.2.33 int DisplayCore::getCursorX ( ) [virtual]

Get X Cursor

Returns the current X position of the text cursor.

Example:

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

5.17.2.34 int DisplayCore::getCursorY ( ) [virtual]

Get Y Cursor

Returns the current Y position of the text cursor.

Example:

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

5.17.2.35 int DisplayCore::getHeight ( ) [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented in Image, KS0108_BB2, BMPFile, PG25664CG, SSD1306, SDL, Goldelox, VGA, KS0108_2, NativeFB, VLCD, KS0108, twAnimIcon, and gciWidget.

5.17.2.36 color_t DisplayCore::getTextColor ( ) [virtual]

Get the current foreground colour

Returns the currently selected foreground colour.

Example:

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

5.17.2.37 int DisplayCore::getWidth ( ) [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented in Image, KS0108_BB2, BMPFile, PG25664CG, SSD1306, SDL, Goldelox, VGA, KS0108_2, NativeFB, VLCD, KS0108, twAnimIcon, and gciWidget.

5.17.2.38 virtual void DisplayCore::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 Image, SSD1963, KS0108_BB2, PG25664CG_PMP, LM6800, LCARS::MiniScope, KS0108_BB, LCARS::StaticText, PG25664CG_PORTB, SSD1306_BB, SSD1289_PMP, ILI9481_PMP, ILI9340, ILI9481, SSD1289, ST7735, LCARS::Block, SSD1306, SDL, Goldelox, LCARS::HBarBend, VGA, KS0108_2, PG25664CG, NativeFB, ILI9163, VLCD, LCARS::HBar, BD663474, KS0108, HX8347D, twText, Framebuffer332, and Framebuffer565.

5.17.2.39 virtual void DisplayCore::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 Image, SSD1963, KS0108_BB2, LM6800, SSD1289, ST7735, ILI9340, PG25664CG, SSD1306, Goldelox, ILI9481, SDL, VGA, KS0108_2, NativeFB, VLCD, KS0108, BD663474, HX8347D, and ILI9163.

5.17.2.40 void DisplayCore::invertTextColor ( ) [virtual]

Invert the text colours

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

Example:

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

5.17.2.41 color_t DisplayCore::mix (color_t *a,* color_t *b,* int *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);
```

5.17.2.42 void DisplayCore::openWindow (int *x0,* int *y0,* int *x1,* int *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 SSD1963, SSD1289, ILI9340, Goldelox, ILI9481, HX8347D, and ILI9163.

5.17.2.43 uint32_t DisplayCore::rgb2hsv (color_t *rgb*) [static]

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);
```

5.17.2.44 point3d DisplayCore::rgb2xyz (color_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);
```

5.17.2.45 virtual void DisplayCore::setBacklight (int *b*) [inline], [virtual]

Set Back Light Brightness

For devices with their own backlight control this function will set the brightness of the backlight.

Reimplemented in SSD1963.

5.17.2.46 void DisplayCore::setClipping (int *x0*, int *y0*, int *x1*, int *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);
```

5.17.2.47 void DisplayCore::setCursor (int *x*, int *y*) [virtual]

Set the text cursor

All future printing will happen from the pixel (x,y).
Example:

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

5.17.2.48 void DisplayCore::setCursorX (int *x*) [virtual]

Set the text X cursor

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

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

5.17.2.49 void DisplayCore::setCursorY (int *y*) [virtual]

Set the text Y cursor

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

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

5.17.2.50 void DisplayCore::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);
```

Reimplemented in twButton, twAnimIcon, and twIcon.

5.17.2.51 virtual void DisplayCore::setPixel (int *x*, int *y*, color_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 Image, SSD1963, KS0108_BB2, LCARS::MiniScope, LM6800, LCARS::StaticText, LCARS::Block, PG25664CG, SSD1289, ST7735, ILI9340, SSD1306, ILI9481, Goldelox, VGA, LCARS::HBarBend, KS0108_2, VLCD, KS0108, LCARS::HBar, BD663474, HX8347D, ILI9163, twText, Framebuffer332, and Framebuffer565.

5.17.2.52 virtual void DisplayCore::setRotation (int *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 Image, SSD1963, LM6800, SSD1289, ST7735, ILI9340, PG25664CG, SSD1306, ILI9481, VGA, KS0108_2, VLCD, KS0108, BD663474, HX8347D, and ILI9163.

5.17.2.53 void DisplayCore::setTextColor (color_t *c*) [virtual]

Set the text foreground colour

Sets the foreground colour of all future printing to (c).

Example:

```
tft.setTextColor(Color::Magenta);
```

Reimplemented in twAnimIcon, and twIcon.

5.17.2.54 void DisplayCore::setTextColor (color_t *fg*, color_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);
```

Reimplemented in twButton.

5.17.2.55 void DisplayCore::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 alows you to enable (true) or disable (false) this functionality. By default text wrapping is enabled.

Example:

```
tft.setTextWrap(false);
```

5.17.2.56 virtual void DisplayCore::startBuffer ( ) [inline], [virtual]

Start buffered mode

In buffered mode, where applicable, any data that would be sent to the screen is delayed until buffered mode is ended. This generally has no effect on most screens, but those that use their own driver level may use this to delay pushing out of the buffer to the screen.

Reimplemented in KS0108_BB2, LM6800, PG25664CG, SDL, SSD1306, KS0108_2, and KS0108.

5.17.2.57 int DisplayCore::stringHeight (const 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");
```

5.17.2.58 int DisplayCore::stringWidth (const 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");
```

5.17.2.59 void DisplayCore::windowData (color_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 SSD1963, SDL, SSD1289, ILI9340, Goldelox, ILI9481, HX8347D, and ILI9163.

5.17.2.60 void DisplayCore::windowData (color_t * *d,* int *l*) [virtual]

Send a block of pixel data to the window

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

Example:

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

Reimplemented in SSD1289, ILI9340, Goldelox, ILI9481, and ILI9163.

5.17.2.61 void DisplayCore::write (uint8_t *c*) [virtual]

Get Port Data

Utility function toget the information about an IO port for high speed access.

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');
```

Reimplemented in LCARS::MessageLog.

5.17.2.62 point3d DisplayCore::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);
```

5.17.3 Field Documentation

5.17.3.1 int DisplayCore::_height

Height of the TFT screen

5.17.3.2 int DisplayCore::_width

Width of the TFT screen

5.17.3.3 int DisplayCore::cursor_x

The text cursor X position

5.17.3.4 int DisplayCore::cursor_y

The text cursor Y position

5.17.3.5 const uint8_t* DisplayCore::font [protected]

A pointer to the currently selected font table

5.17.3.6 int DisplayCore::rotation

Current rotation

5.17.3.7 color_t DisplayCore::textbgcolor

Text background colour

5.17.3.8 color_t DisplayCore::textcolor

Text foreground colour

5.17.3.9 boolean DisplayCore::wrap

Whether or not text wrapping is enabled

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

- DisplayCore/DisplayCore.h
- DisplayCore/DisplayCore.cpp

5.18 Event Struct Reference

Collaboration diagram for Event:

Data Fields

- Widget * **source**
- int **x**
- int **y**
- int **dx**
- int **dy**
- uint32_t **type**

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

- DisplayCore/DisplayCore.h

5.19 event Struct Reference

Data Fields

- struct timeval **tv**
- unsigned short **type**
- unsigned short **code**
- unsigned int **value**

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

- Drivers/LinuxEvent/LinuxEvent.h

5.20 LCARS::ExpandedOvalButton Class Reference

Inheritance diagram for LCARS::ExpandedOvalButton:

Collaboration diagram for LCARS::ExpandedOvalButton:

Public Member Functions

- **ExpandedOvalButton** (Touch &ts, DisplayCore &dev, int x, int y, int w, color_t off, color_t on, color_t hi, color_t st, const char *text, const char *title, const char *offtext, const char *ontext)
- void **draw** (DisplayCore *dev, int x, int y)

Additional Inherited Members

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.21 Filter Class Reference

Inheritance diagram for Filter:

Collaboration diagram for Filter:

Public Member Functions

- virtual color_t **function** (color_t)=0
- void **chain** (Filter &f)
- void **chain** (Filter *f)
- void **endChain** ()

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.

- color_t **process** (color_t)

Protected Attributes

- Filter * **_next**

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

- DisplayCore/DisplayCore.h
- DisplayCore/DisplayCore.cpp

5.22 FontHeader Struct Reference

Data Fields

- 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:

- DisplayCore/DisplayCore.h

5.23 Form Class Reference

Public Member Functions

- **Form** (int num...)
- void **render** ()
- void **redraw** ()
- void **onPress** (void(*func)(Event *))
- void **onRelease** (void(*func)(Event *))

- void **onDrag** (void(∗func)(Event ∗))
- void **onTap** (void(∗func)(Event ∗))
- void **onRepeat** (void(∗func)(Event ∗))

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

- DisplayCore/DisplayCore.h
- DisplayCore/DisplayCore.cpp

5.24 Framebuffer332 Class Reference

Inheritance diagram for Framebuffer332:

Collaboration diagram for Framebuffer332:

Public Member Functions

- **Framebuffer332** (int w, int h, uint8_t ∗b)
- void initializeDevice ()
- void setPixel (int x, int y, color_t c)
- color_t colorAt (int x, int y)
- void fillScreen (color_t c)
- void **draw** (DisplayCore ∗dev, int x, int y)
- void **draw** (DisplayCore ∗dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore ∗dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore ∗dev, int x, int y, int transform, color_t t)
- void **draw** (DisplayCore &dev, int x, int y)
- void **draw** (DisplayCore &dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform, color_t t)

Additional Inherited Members

5.24.1 Member Function Documentation

5.24.1.1 color_t Framebuffer332::colorAt (int *x*, int *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 DisplayCore.

5.24.1.2 void Framebuffer332::fillScreen (color_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 DisplayCore.

5.24.1.3 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 Image.

5.24.1.4 void Framebuffer332::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Reimplemented from Image.

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

- Drivers/Framebuffer332/Framebuffer332.h
- Drivers/Framebuffer332/Framebuffer332.cpp

5.25 Framebuffer565 Class Reference

Inheritance diagram for Framebuffer565:

Collaboration diagram for Framebuffer565:

Public Member Functions

- **Framebuffer565** (int w, int h, color_t *b)
- void initializeDevice ()
- void setPixel (int x, int y, color_t c)
- void fillScreen (color_t c)
- void **draw** (DisplayCore *dev, int x, int y)
- void **draw** (DisplayCore *dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore *dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore *dev, int x, int y, int transform, color_t t)
- void **draw** (DisplayCore &dev, int x, int y)
- void **draw** (DisplayCore &dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform, color_t t)

Additional Inherited Members

5.25.1 Member Function Documentation

5.25.1.1 void Framebuffer565::fillScreen (color_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 DisplayCore.

5.25.1.2 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 Image.

5.25.1.3 void Framebuffer565::setPixel (int *x,* int *y,* color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Reimplemented from Image.

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

- Drivers/Framebuffer565/Framebuffer565.h
- Drivers/Framebuffer565/Framebuffer565.cpp

5.26 gcihdr Struct Reference

Data Fields

- uint16_t **width**
- uint16_t **height**
- uint8_t **mode**
- uint8_t **fdel**
- uint16_t **frames**

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

- Toolkits/gciWidget/gciWidget.h

5.27 gciWidget Class Reference

Inheritance diagram for gciWidget:

Collaboration diagram for gciWidget:

Public Member Functions

- **gciWidget** (Touch &ts, DisplayCore &dev, const char *fn, const char *wn)
- void **draw** (DisplayCore *dev, int x, int y)
- int **getFrames** ()
- int getWidth ()
- int getHeight ()
- void **setValue** (int x)

Data Fields

- bool **_loaded**
- File * **_file**
- gcihdr **_header**
- const char * **_filename**
- const char * **_widgetname**
- uint32_t **_offset**

Additional Inherited Members

5.27.1 Member Function Documentation

5.27.1.1 int gciWidget::getHeight ( ) [inline], [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from Image.

5.27.1.2 int gciWidget::getWidth ( ) [inline], [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from Image.

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

- Toolkits/gciWidget/gciWidget.h
- Toolkits/gciWidget/gciWidget.cpp

5.28 Goldelox Class Reference

Inheritance diagram for Goldelox:

Collaboration diagram for Goldelox:

Public Member Functions

- **Goldelox** (Stream *dev, int w, int h)
- **Goldelox** (Stream &dev, int w, int h)
- **Goldelox** (Stream *dev, int w, int h, uint8_t reset)
- **Goldelox** (Stream &dev, int w, int h, uint8_t reset)
- void initializeDevice ()
- void displayOn ()
- void displayOff ()
- void fillScreen (color_t color)
- void setPixel (int x, int y, color_t color)
- void drawLine (int x0, int y0, int x1, int y1, color_t color)
- int getWidth ()
- int getHeight ()
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void drawRectangle (int x, int y, int w, int h, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void **setRotation** (uint8_t r)
- void invertDisplay (boolean i)
- void **changeBaudRate** (uint32_t baud)
- void openWindow (int x, int y, int w, int h)
- void closeWindow ()
- void windowData (color_t d)
- void windowData (color_t *d, int l)

Additional Inherited Members

5.28.1 Member Function Documentation

5.28.1.1 void Goldelox::closeWindow ( ) [virtual]

Close the window

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

Example:

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

Reimplemented from DisplayCore.

5.28.1.2 void Goldelox::displayOff ( ) [virtual]

Turn off the display

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

Example:

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

Implements DisplayCore.

5.28.1.3 void Goldelox::displayOn ( ) [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.28.1.4 void Goldelox::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.28.1.5 void Goldelox::drawLine (int *x0*, int *y0*, int *x1*, int *y1*, color_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);
```

Reimplemented from DisplayCore.

5.28.1.6 void Goldelox::drawRectangle (int *x*, int *y*, int *w*, int *h*, color_t *color*) [virtual]

Draw a rectangle

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

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

Reimplemented from DisplayCore.

5.28.1.7 void Goldelox::drawVerticalLine (int *x,* int *y,* int *h,* color_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);
```

Reimplemented from DisplayCore.

5.28.1.8 void Goldelox::fillRectangle (int *x,* int *y,* int *w,* int *h,* color_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 DisplayCore.

5.28.1.9 void Goldelox::fillScreen (color_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 DisplayCore.

5.28.1.10 int Goldelox::getHeight ( ) [inline], [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.28.1.11 int Goldelox::getWidth ( ) [inline], [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.28.1.12 void Goldelox::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 DisplayCore.

5.28.1.13 void Goldelox::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 DisplayCore.

5.28.1.14 void Goldelox::openWindow (int *x0*, int *y0*, int *x1*, int *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 DisplayCore.

5.28.1.15 void Goldelox::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Implements DisplayCore.

5.28.1.16 void Goldelox::windowData (color_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 DisplayCore.

5.28.1.17 void Goldelox::windowData (color_t * *d,* int *l*) [virtual]

Send a block of pixel data to the window

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

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

Reimplemented from DisplayCore.

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

- Drivers/Goldelox/Goldelox.h
- Drivers/Goldelox/Goldelox.cpp

5.29 LCARS::HBar Class Reference

Inheritance diagram for LCARS::HBar:

Collaboration diagram for LCARS::HBar:

Public Member Functions

- **HBar** (Touch &ts, DisplayCore &dev, int y, color_t lc, color_t mc, color_t rc, const char *lt, const char *mt, const char *rt)
- void setPixel (int x, int y, color_t c)
- void **draw** (DisplayCore *dev, int x, int y)
- void initializeDevice ()

Additional Inherited Members

5.29.1 Member Function Documentation

5.29.1.1 void LCARS::HBar::initializeDevice ( ) [inline], [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 Image.

5.29.1.2 void LCARS::HBar::setPixel (int *x,* int *y,* color_t *color*) [inline], [virtual]

Draw a pixel

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

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

Reimplemented from Image.

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.30 LCARS::HBarBend Class Reference

Inheritance diagram for LCARS::HBarBend:

Collaboration diagram for LCARS::HBarBend:

Public Member Functions

- **HBarBend** (Touch &ts, DisplayCore &dev, int x, int y, int bt, color_t lc, color_t mc, color_t rc, color_t ec, int mp, int ms, int bs, const char *t)
- void setPixel (int x, int y, color_t c)
- void **setValue** (int x)
- void **draw** (DisplayCore *dev, int x, int y)
- void initializeDevice ()
- void **render** ()

Static Public Attributes

- static const int **BendDown** = 0x01
- static const int **BendUp** = 0x02
- static const int **BendLeft** = 0x10
- static const int **BendRight** = 0x20

Additional Inherited Members

5.30.1 Member Function Documentation

5.30.1.1 void LCARS::HBarBend::initializeDevice ( ) [inline], [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 Image.

5.30.1.2 void LCARS::HBarBend::setPixel (int *x*, int *y*, color_t *color*) [inline], [virtual]

Draw a pixel

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

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

Reimplemented from Image.

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.31 HX8347D Class Reference

Inheritance diagram for HX8347D:

Collaboration diagram for HX8347D:

Public Member Functions

- **HX8347D** (DSPI &spi, int dc, int cs)
- void **setAddrWindow** (int x0, int y0, int x1, int y1)
- void **setAddrWindowRead** (int x0, int y0, int x1, int y1)
- void fillScreen (color_t color)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- void initializeDevice ()
- void openWindow (int x0, int y0, int x1, int y1)
- void windowData (color_t d)
- void **windowData** (color_t *d, uint32_t l)
- void closeWindow ()
- void **writeCommand** (uint8_t)
- void **writeData** (uint8_t)
- void **setRegister** (uint8_t reg, uint8_t val)

Static Public Attributes

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

Additional Inherited Members

5.31.1 Member Function Documentation

5.31.1.1 void HX8347D::closeWindow ( ) [virtual]

Close the window

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

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

Reimplemented from DisplayCore.

5.31.1.2 void HX8347D::displayOff ( ) [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.31.1.3 void HX8347D::displayOn ( ) [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.31.1.4 void HX8347D::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.31.1.5 void HX8347D::drawVerticalLine (int *x*, int *y*, int *h*, color_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);
```

Reimplemented from DisplayCore.

5.31.1.6 void HX8347D::fillRectangle (int *x,* int *y,* int *w,* int *h,* color_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 DisplayCore.

5.31.1.7 void HX8347D::fillScreen (color_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 DisplayCore.

5.31.1.8 void HX8347D::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 DisplayCore.

5.31.1.9 void HX8347D::invertDisplay (boolean *i*) [virtual]

Invert the display colours

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

Example:

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

Implements DisplayCore.

5.31.1.10 void HX8347D::openWindow (int *x0,* int *y0,* int *x1,* int *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 DisplayCore.

5.31.1.11 void HX8347D::setPixel (int *x,* int *y,* color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.31.1.12 void HX8347D::setRotation (int *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 DisplayCore.

5.31.1.13 void HX8347D::windowData (color_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 DisplayCore.

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

- Drivers/HX8347D/HX8347D.h
- Drivers/HX8347D/HX8347D.cpp

5.32 ILI9163 Class Reference

Inheritance diagram for ILI9163:

Collaboration diagram for ILI9163:

Public Member Functions

- **ILI9163** (DSPI *dspi, uint8_t cs, uint8_t rs, uint8_t reset)
- void **setAddrWindow** (int x0, int y0, int x1, int y1)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- void openWindow (int, int, int, int)
- void windowData (color_t)
- void windowData (color_t *, int)
- void closeWindow ()
- void **startDisplay** ()
- virtual void initializeDevice ()
- virtual void **data** (uint16_t)
- virtual void **command** (uint16_t)

Static Public Attributes

- static const int **Width** = 128
- static const int **Height** = 128

Additional Inherited Members

5.32.1 Member Function Documentation

5.32.1.1 void ILI9163::closeWindow ( ) [virtual]

Close the window

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

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

Reimplemented from DisplayCore.

5.32.1.2 void ILI9163::displayOff ( ) [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.32.1.3 void ILI9163::displayOn ( ) [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.32.1.4 void ILI9163::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.32.1.5 void ILI9163::drawVerticalLine (int *x*, int *y*, int *h*, color_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);
```

Reimplemented from DisplayCore.

5.32.1.6 void ILI9163::fillRectangle (int *x*, int *y*, int *w*, int *h*, color_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 DisplayCore.

5.32.1.7 void ILI9163::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 DisplayCore.

5.32.1.8 void ILI9163::invertDisplay (boolean *i*) [virtual]

Invert the display colours

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

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

Implements DisplayCore.

5.32.1.9 void ILI9163::openWindow (int *x0,* int *y0,* int *x1,* int *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 DisplayCore.

5.32.1.10 void ILI9163::setPixel (int *x,* int *y,* color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.32.1.11 void ILI9163::setRotation (int *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 DisplayCore.

5.32.1.12 void ILI9163::windowData (color_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 DisplayCore.

5.32.1.13 void ILI9163::windowData (color_t * d, int l) [virtual]

Send a block of pixel data to the window

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

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

Reimplemented from DisplayCore.

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

- Drivers/ILI9163/ILI9163.h
- Drivers/ILI9163/ILI9163.cpp

5.33 ILI9340 Class Reference

Inheritance diagram for ILI9340:

Collaboration diagram for ILI9340:

Public Member Functions

- **ILI9340** (DSPI *spi, uint8_t cs, uint8_t dc, uint8_t reset)
- **ILI9340** (DSPI &spi, uint8_t cs, uint8_t dc, uint8_t reset)
- void **setAddrWindow** (int x0, int y0, int x1, int y1)
- void fillScreen (color_t color)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- void openWindow (int, int, int, int)
- void windowData (color_t)
- void windowData (color_t *, int)
- void closeWindow ()
- void initializeDevice ()
- void **data** (uint8_t)
- void **command** (uint8_t)

Static Public Attributes

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

Additional Inherited Members

5.33.1 Member Function Documentation

5.33.1.1 void ILI9340::closeWindow ( ) [virtual]

Close the window

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

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

Reimplemented from DisplayCore.

5.33.1.2 void ILI9340::displayOff ( ) [inline], [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.33.1.3 void ILI9340::displayOn ( ) [inline], [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.33.1.4 void ILI9340::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.33.1.5 void ILI9340::drawVerticalLine (int *x*, int *y*, int *h*, color_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);
```

Reimplemented from DisplayCore.

5.33.1.6 void ILI9340::fillRectangle (int *x*, int *y*, int *w*, int *h*, color_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 DisplayCore.

5.33.1.7 void ILI9340::fillScreen (color_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 DisplayCore.

5.33.1.8 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 DisplayCore.

5.33.1.9 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 DisplayCore.

5.33.1.10 void ILI9340::openWindow (int *x0,* int *y0,* int *x1,* int *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 DisplayCore.

5.33.1.11 void ILI9340::setPixel (int *x,* int *y,* color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.33.1.12 void ILI9340::setRotation (int *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 DisplayCore.

5.33.1.13 void ILI9340::windowData (color_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 DisplayCore.

5.33.1.14 void ILI9340::windowData (color_t * *d,* int *l*) [virtual]

Send a block of pixel data to the window

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

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

Reimplemented from DisplayCore.

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

- Drivers/ILI9340/ILI9340.h
- Drivers/ILI9340/ILI9340.cpp

5.34 ILI9481 Class Reference

Inheritance diagram for ILI9481:

Collaboration diagram for ILI9481:

Public Member Functions

- **ILI9481** (uint8_t rs, uint8_t wr, uint8_t rd, uint8_t cs, uint8_t reset, 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)
- void **setAddrWindow** (int x0, int y0, int x1, int y1)
- void fillScreen (color_t color)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- void openWindow (int, int, int, int)
- void windowData (color_t)
- void windowData (color_t *, int)
- void closeWindow ()
- virtual color_t colorAt (int x, int y)
- void **startDisplay** ()
- virtual void initializeDevice ()
- virtual void **data** (uint16_t)
- virtual void **command** (uint16_t)
- virtual uint16_t **read** (boolean cont=false)
- virtual void **getRectangle** (int x, int y, int w, int h, color_t *buf)

Static Public Attributes

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

Additional Inherited Members

5.34.1 Member Function Documentation

5.34.1.1 void ILI9481::closeWindow ( ) [virtual]

Close the window

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

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

Reimplemented from DisplayCore.

5.34.1.2 color_t ILI9481::colorAt (int *x*, int *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 DisplayCore.

5.34.1.3 void ILI9481::displayOff ( ) [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.34.1.4 void ILI9481::displayOn ( ) [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.34.1.5 void ILI9481::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.34.1.6 void ILI9481::drawVerticalLine (int *x*, int *y*, int *h*, color_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);
```

Reimplemented from DisplayCore.

5.34.1.7 void ILI9481::fillRectangle (int *x*, int *y*, int *w*, int *h*, color_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 DisplayCore.

5.34.1.8 void ILI9481::fillScreen (color_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 DisplayCore.

5.34.1.9 void ILI9481::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 DisplayCore.

Reimplemented in ILI9481_PMP.

5.34.1.10 void ILI9481::invertDisplay (boolean *i*) [virtual]

Invert the display colours

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

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

Implements DisplayCore.

5.34.1.11 void ILI9481::openWindow (int *x0*, int *y0*, int *x1*, int *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 DisplayCore.

5.34.1.12 void ILI9481::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.34.1.13 void ILI9481::setRotation (int *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 DisplayCore.

5.34.1.14 void ILI9481::windowData (color_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 DisplayCore.

5.34.1.15 void ILI9481::windowData (color_t * *d*, int *l*) [virtual]

Send a block of pixel data to the window

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

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

Reimplemented from DisplayCore.

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

- Drivers/ILI9481/ILI9481.h
- Drivers/ILI9481/ILI9481.cpp

5.35 ILI9481_PMP Class Reference

Inheritance diagram for ILI9481_PMP:

Collaboration diagram for ILI9481_PMP:

Public Member Functions

- **ILI9481_PMP** (uint8_t res)
- void initializeDevice ()
- void **data** (uint16_t)
- void **command** (uint16_t)
- uint16_t **read** (boolean cont=false)

Additional Inherited Members

5.35.1 Member Function Documentation

5.35.1.1 void ILI9481_PMP::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 ILI9481.

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

- Drivers/ILI9481/ILI9481.h
- Drivers/ILI9481/ILI9481.cpp

5.36 Image Class Reference

Inheritance diagram for Image:

Collaboration diagram for Image:

Public Member Functions

- virtual int getWidth ()
- virtual int getHeight ()
- virtual void **draw** (DisplayCore *dev, int x, int y)=0
- virtual void **draw** (DisplayCore *dev, int x, int y, color_t t)=0
- virtual void **drawTransformed** (DisplayCore *dev, int x, int y, int transform)=0
- virtual void **drawTransformed** (DisplayCore *dev, int x, int y, int transform, color_t t)=0
- void **draw** (DisplayCore &dev, int x, int y)
- void **draw** (DisplayCore &dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform, color_t t)
- void **setFilter** (Filter &f)
- void **removeFilter** ()
- void setRotation (int r)
- void setPixel (int x, int y, color_t c)
- void initializeDevice ()
- void displayOn ()
- void displayOff ()
- void invertDisplay (boolean i)

Data Fields

- int **_width**
- int **_height**

Static Public Attributes

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

Protected Attributes

- Filter * **_filter**

Additional Inherited Members

5.36.1 Member Function Documentation

5.36.1.1 void Image::displayOff ( ) [inline], [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.36.1.2 void Image::displayOn ( ) [inline], [virtual]

Turn on the display

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

Example:

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

Implements DisplayCore.

5.36.1.3 virtual int Image::getHeight ( ) [inline], [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

Reimplemented in BMPFile, twAnimIcon, and gciWidget.

5.36.1.4 virtual int Image::getWidth ( ) [inline], [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

Reimplemented in BMPFile, twAnimIcon, and gciWidget.

5.36.1.5 void Image::initializeDevice ( ) [inline], [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 DisplayCore.

Reimplemented in LCARS::MiniScope, LCARS::StaticText, LCARS::Block, LCARS::HBarBend, LCARS::HBar, twText, Framebuffer332, and Framebuffer565.

5.36.1.6 void Image::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 DisplayCore.

5.36.1.7 void Image::setPixel (int *x*, int *y*, color_t *color*) [inline], [virtual]

Draw a pixel

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

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

Implements DisplayCore.

Reimplemented in LCARS::MiniScope, LCARS::StaticText, LCARS::Block, LCARS::HBarBend, LCARS::HBar, tw↩ Text, Framebuffer332, and Framebuffer565.

5.36.1.8 void Image::setRotation (int *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 DisplayCore.

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

- DisplayCore/DisplayCore.h

5.37 Invert Class Reference

Inheritance diagram for Invert:

Collaboration diagram for Invert:

Public Member Functions

- color_t **function** (color_t)

Additional Inherited Members

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

- Filters/Invert.h
- Filters/Invert.cpp

5.38 KS0108 Class Reference

Inheritance diagram for KS0108:

Collaboration diagram for KS0108:

Public Member Functions

- **KS0108** (uint32_t reg, uint32_t data)
- virtual void initializeDevice ()
- virtual void **displayInit** ()
- void setPixel (int x, int y, color_t color)
- void startBuffer ()
- void endBuffer ()
- void setRotation (int r)
- void displayOn ()
- void displayOff ()
- void invertDisplay (boolean b)
- int getWidth ()
- int getHeight ()

Additional Inherited Members

5.38.1 Member Function Documentation

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

Turn off the display

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

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

Implements DisplayCore.

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

Turn on the display

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

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

Implements DisplayCore.

5.38.1.3 void KS0108::endBuffer ( ) [virtual]

End buffered mode

Any changes that are pending will be pushed out to the screen. See `startBuffer()` for more information.

Reimplemented from DisplayCore.

5.38.1.4 int KS0108::getHeight ( ) [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.38.1.5 int KS0108::getWidth ( ) [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.38.1.6 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 DisplayCore.

Reimplemented in KS0108_BB.

5.38.1.7 void KS0108::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 DisplayCore.

5.38.1.8 void KS0108::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Implements DisplayCore.

5.38.1.9 void KS0108::setRotation (int *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 DisplayCore.

5.38.1.10 void KS0108::startBuffer ( ) [virtual]

Start buffered mode

In buffered mode, where applicable, any data that would be sent to the screen is delayed until buffered mode is ended. This generally has no effect on most screens, but those that use their own driver level may use this to delay pushing out of the buffer to the screen.

Reimplemented from DisplayCore.

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

- Drivers/KS0108/KS0108.h
- Drivers/KS0108/KS0108.cpp

5.39 KS0108_2 Class Reference

Inheritance diagram for KS0108_2:

Collaboration diagram for KS0108_2:

Public Member Functions

- **KS0108_2** (uint32_t reg1, uint32_t data1, uint32_t reg2, uint32_t data2)
- void initializeDevice ()
- void setPixel (int x, int y, color_t color)
- void startBuffer ()
- void endBuffer ()
- void setRotation (int r)
- void displayOn ()
- void displayOff ()
- void invertDisplay (boolean b)
- int getWidth ()
- int getHeight ()

Additional Inherited Members

5.39.1 Member Function Documentation

5.39.1.1 void KS0108_2::displayOff ( ) [inline], [virtual]

Turn off the display

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

Example:

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

Implements DisplayCore.

5.39.1.2 void KS0108_2::displayOn ( ) [inline], [virtual]

Turn on the display

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

Example:

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

Implements DisplayCore.

5.39.1.3 void KS0108_2::endBuffer ( ) [virtual]

End buffered mode

Any changes that are pending will be pushed out to the screen. See `startBuffer()` for more information.

Reimplemented from DisplayCore.

5.39.1.4 int KS0108_2::getHeight ( ) [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.39.1.5 int KS0108_2::getWidth ( ) [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.39.1.6 void KS0108_2::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 DisplayCore.

5.39.1.7 void KS0108_2::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 DisplayCore.

5.39.1.8 void KS0108_2::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.39.1.9 void KS0108_2::setRotation (int *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 DisplayCore.

5.39.1.10 void KS0108_2::startBuffer ( ) [virtual]

Start buffered mode

In buffered mode, where applicable, any data that would be sent to the screen is delayed until buffered mode is ended. This generally has no effect on most screens, but those that use their own driver level may use this to delay pushing out of the buffer to the screen.

Reimplemented from DisplayCore.

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

- Drivers/KS0108/KS0108.h
- Drivers/KS0108/KS0108.cpp

5.40 KS0108_BB Class Reference

Inheritance diagram for KS0108_BB:

Collaboration diagram for KS0108_BB:

Public Member Functions

- **KS0108_BB** (uint8_t rs, uint8_t rw, uint8_t e, uint8_t cs, 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 initializeDevice ()

Additional Inherited Members

5.40.1 Member Function Documentation

5.40.1.1 void KS0108_BB::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 KS0108.

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

- Drivers/KS0108/KS0108.h
- Drivers/KS0108/KS0108.cpp

5.41 KS0108_BB2 Class Reference

Inheritance diagram for KS0108_BB2:

Collaboration diagram for KS0108_BB2:

Public Member Functions

- **KS0108_BB2** (uint8_t rs, uint8_t rw, uint8_t e, uint8_t cs1, uint8_t cs2, 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 initializeDevice ()
- void setPixel (int x, int y, color_t color)
- void startBuffer ()
- void endBuffer ()
- void **setRotation** (uint8_t r)
- void displayOn ()
- void displayOff ()
- void invertDisplay (boolean b)
- int getWidth ()
- int getHeight ()

Additional Inherited Members

5.41.1 Member Function Documentation

5.41.1.1 void KS0108_BB2::displayOff ( ) [inline], [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.41.1.2 void KS0108_BB2::displayOn ( ) [inline], [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.41.1.3 void KS0108_BB2::endBuffer ( ) [virtual]

End buffered mode

Any changes that are pending will be pushed out to the screen. See `startBuffer()` for more information.

Reimplemented from DisplayCore.

5.41.1.4 int KS0108_BB2::getHeight ( ) [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.41.1.5 int KS0108_BB2::getWidth ( ) [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.41.1.6 void KS0108_BB2::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 DisplayCore.

5.41.1.7 void KS0108_BB2::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 DisplayCore.

5.41.1.8 void KS0108_BB2::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Implements DisplayCore.

5.41.1.9 void KS0108_BB2::startBuffer ( ) [virtual]

Start buffered mode

In buffered mode, where applicable, any data that would be sent to the screen is delayed until buffered mode is ended. This generally has no effect on most screens, but those that use their own driver level may use this to delay pushing out of the buffer to the screen.

Reimplemented from DisplayCore.

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

- Drivers/KS0108/KS0108.h
- Drivers/KS0108/KS0108.cpp

5.42 LinuxEvent Class Reference

Inheritance diagram for LinuxEvent:

Collaboration diagram for LinuxEvent:

Public Member Functions

- **LinuxEvent** (uint16_t w, uint16_t h)
- void sample ()
- int **getSample** (uint8_t)
- uint16_t x ()
- uint16_t y ()
- uint16_t rawX ()
- uint16_t **rawY** ()
- boolean **isPressed** ()
- void initializeDevice ()
- uint16_t pressure ()
- void **setRotation** (uint8_t r)
- size_t **write** (uint8_t v)
- int **available** ()
- int **read** ()
- int **peek** ()
- void **flush** ()

Additional Inherited Members

5.42.1 Member Function Documentation

5.42.1.1 void LinuxEvent::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.

5.42.1.2 uint16_t LinuxEvent::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.

5.42.1.3 uint16_t LinuxEvent::rawX ( ) [virtual]

Get pressed status

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

Reimplemented from Touch.

5.42.1.4 void LinuxEvent::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.

5.42.1.5 uint16_t LinuxEvent::x ( ) [virtual]

Get X coordinate

This returns the X coordinate of the current touch position.

Implements Touch.

5.42.1.6 uint16_t LinuxEvent::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:

- Drivers/LinuxEvent/LinuxEvent.h
- Drivers/LinuxEvent/LinuxEvent.cpp

5.43 LM6800 Class Reference

Inheritance diagram for LM6800:

Collaboration diagram for LM6800:

Public Member Functions

- **LM6800** (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 csa, uint8_t csb, uint8_t csc, uint8_t e, uint8_t rs, uint8_t rw, uint8_t reset=NULL)
- void **setAddrWindow** (int x0, int y0, int x1, int y1)
- void fillScreen (color_t color)
- void **doSetPixel** (int x, int y, color_t color)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- void initializeDevice ()
- void **updateScreen** ()
- void startBuffer ()
- void endBuffer ()

Static Public Attributes

- static const int **Width** = 256
- static const int **Height** = 64

Protected Attributes

- uint8_t **colstart**
- uint8_t **rowstart**
- uint8_t **_variant**
- uint8_t **buffer** [2048]

Additional Inherited Members

5.43.1 Member Function Documentation

5.43.1.1 void LM6800::displayOff ( ) [inline], [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.43.1.2 void LM6800::displayOn ( ) [inline], [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.43.1.3 void LM6800::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.43.1.4 void LM6800::drawVerticalLine (int *x,* int *y,* int *h,* color_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);
```

Reimplemented from DisplayCore.

5.43.1.5 void LM6800::endBuffer ( ) [inline], [virtual]

End buffered mode

Any changes that are pending will be pushed out to the screen. See `startBuffer()` for more information.

Reimplemented from DisplayCore.

5.43.1.6 void LM6800::fillRectangle (int *x,* int *y,* int *w,* int *h,* color_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 DisplayCore.

5.43.1.7 void LM6800::fillScreen (color_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 DisplayCore.

5.43.1.8 void LM6800::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 DisplayCore.

5.43.1.9 void LM6800::invertDisplay (boolean *i*) [virtual]

Invert the display colours

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

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

Implements DisplayCore.

5.43.1.10 void LM6800::setPixel (int *x,* int *y,* color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.43.1.11 void LM6800::setRotation (int *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 DisplayCore.

5.43.1.12 void LM6800::startBuffer ( ) [inline], [virtual]

Start buffered mode

In buffered mode, where applicable, any data that would be sent to the screen is delayed until buffered mode is ended. This generally has no effect on most screens, but those that use their own driver level may use this to delay pushing out of the buffer to the screen.

Reimplemented from DisplayCore.

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

- Drivers/LM6800/LM6800.h
- Drivers/LM6800/LM6800.cpp

5.44 LCARS::MessageLog Class Reference

Inheritance diagram for LCARS::MessageLog:

Collaboration diagram for LCARS::MessageLog:

Public Member Functions

- **MessageLog** (Touch &ts, DisplayCore &dev, int x, int y)
- void **setValue** (int v)
- void **setValue** (const char *str)
- void **draw** (DisplayCore *dev, int x, int y)
- void **render** ()
- virtual void write (uint8_t v)

Additional Inherited Members

5.44.1 Member Function Documentation

5.44.1.1 void LCARS::MessageLog::write (uint8_t *c*) [virtual]

Get Port Data

Utility function toget the information about an IO port for high speed access.

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');
```

Reimplemented from DisplayCore.

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.45 LCARS::MiniScope Class Reference

Inheritance diagram for LCARS::MiniScope:

Collaboration diagram for LCARS::MiniScope:

Public Member Functions

- **MiniScope** (Touch &ts, DisplayCore &dev, int x, int y)
- void **setValue** (int v)
- int **getValue** ()
- int **getAverage** ()
- void setPixel (int x, int y, color_t c)
- void **draw** (DisplayCore *dev, int x, int y)
- void initializeDevice ()

Additional Inherited Members

5.45.1 Member Function Documentation

5.45.1.1 void LCARS::MiniScope::initializeDevice ( ) [inline], [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 Image.

5.45.1.2 void LCARS::MiniScope::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Reimplemented from Image.

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.46 Monochrome Class Reference

Inheritance diagram for Monochrome:

Collaboration diagram for Monochrome:

Public Member Functions

- color_t **function** (color_t)

Additional Inherited Members

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

- Filters/Monochrome.h
- Filters/Monochrome.cpp

5.47 MonoIcon Class Reference

Inheritance diagram for MonoIcon:

Collaboration diagram for MonoIcon:

Public Member Functions

- **MonoIcon** (Touch &ts, DisplayCore &dev, int x, int y, int w, int h, const color_t *bg, const uint8_t *icon, color_t color, const char *text, const uint8_t *font, color_t textcol)
- void **draw** (DisplayCore *dev, int x, int y)
- void **setColor** (color_t c)
- void **setIcon** (const uint8_t *i)

Static Public Attributes

- static const color_t **MonoIconBG** []

Additional Inherited Members

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

- Toolkits/MonoIcon/MonoIcon.h
- Toolkits/MonoIcon/MonoIcon.cpp
- Toolkits/MonoIcon/MonoIconBG.cpp

5.48 NativeFB Class Reference

Inheritance diagram for NativeFB:

Collaboration diagram for NativeFB:

Public Member Functions

- void initializeDevice ()
- void **setPixel** (int16_t x, int16_t y, uint16_t c)
- uint16_t **colorAt** (int16_t x, int16_t y)
- void **setRotation** (uint8_t r)
- void displayOn ()
- void displayOff ()
- void invertDisplay (boolean b)
- uint16_t getWidth ()
- uint16_t getHeight ()
- void **disableCursor** ()
- void **enableCursor** ()

Additional Inherited Members

5.48.1 Member Function Documentation

5.48.1.1 void NativeFB::displayOff ( ) [virtual]

Turn off the display

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

Example:

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

Implements DisplayCore.

5.48.1.2 void NativeFB::displayOn ( ) [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.48.1.3 uint16_t NativeFB::getHeight ( ) [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.48.1.4 uint16_t NativeFB::getWidth ( ) [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.48.1.5 void NativeFB::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 DisplayCore.

5.48.1.6 void NativeFB::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 DisplayCore.

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

- Drivers/NativeFB/NativeFB.h
- Drivers/NativeFB/NativeFB.cpp

5.49 Noise Class Reference

Inheritance diagram for Noise:

Collaboration diagram for Noise:

Public Member Functions

- **Noise** (int n)
- color_t **function** (color_t)
- void **setLevel** (int n)

Additional Inherited Members

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

- Filters/Noise.h
- Filters/Noise.cpp

5.50 LCARS::OvalButton Class Reference

Inheritance diagram for LCARS::OvalButton:

Collaboration diagram for LCARS::OvalButton:

Public Member Functions

- **OvalButton** (Touch &ts, DisplayCore &dev, int x, int y, color_t off, color_t on, color_t hi, const char *text)
- void **draw** (DisplayCore *dev, int x, int y)

Additional Inherited Members

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.51 PG25664CG Class Reference

Inheritance diagram for PG25664CG:

Collaboration diagram for PG25664CG:

Public Member Functions

- **PG25664CG** (uint8_t dc, uint8_t wr, uint8_t rd, uint8_t cs, uint8_t reset, 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 **startDisplay** ()
- virtual void initializeDevice ()
- void displayOn ()
- void displayOff ()
- void setPixel (int x, int y, color_t c)
- void fillScreen (color_t c)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void startBuffer ()
- void endBuffer ()
- int getWidth ()
- int getHeight ()

Data Fields

- int **_buffered**

Additional Inherited Members

5.51.1 Member Function Documentation

5.51.1.1 void PG25664CG::displayOff ( ) [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.51.1.2 void PG25664CG::displayOn ( ) [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.51.1.3 void PG25664CG::endBuffer ( ) [virtual]

End buffered mode

Any changes that are pending will be pushed out to the screen. See `startBuffer()` for more information.

Reimplemented from DisplayCore.

5.51.1.4 void PG25664CG::fillScreen (color_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 DisplayCore.

5.51.1.5 int PG25664CG::getHeight ( ) [inline], [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.51.1.6 int PG25664CG::getWidth ( ) [inline], [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.51.1.7 virtual void PG25664CG::initializeDevice ( ) [inline], [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 DisplayCore.

Reimplemented in PG25664CG_PMP, and PG25664CG_PORTB.

5.51.1.8 void PG25664CG::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 DisplayCore.

5.51.1.9 void PG25664CG::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.51.1.10 void PG25664CG::setRotation (int *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 DisplayCore.

5.51.1.11 void PG25664CG::startBuffer ( ) [virtual]

Start buffered mode

In buffered mode, where applicable, any data that would be sent to the screen is delayed until buffered mode is ended. This generally has no effect on most screens, but those that use their own driver level may use this to delay pushing out of the buffer to the screen.

Reimplemented from DisplayCore.

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

- Drivers/PG25664CG/PG25664CG.h
- Drivers/PG25664CG/PG25664CG.cpp

5.52 PG25664CG_PMP Class Reference

Inheritance diagram for PG25664CG_PMP:

Collaboration diagram for PG25664CG_PMP:

Public Member Functions

- **PG25664CG_PMP** (uint8_t res)
- void initializeDevice ()

Additional Inherited Members

5.52.1 Member Function Documentation

5.52.1.1 void PG25664CG_PMP::initializeDevice ( ) [inline], [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 PG25664CG.

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

- Drivers/PG25664CG/PG25664CG.h
- Drivers/PG25664CG/PG25664CG.cpp

5.53 PG25664CG_PORTB Class Reference

Inheritance diagram for PG25664CG_PORTB:

Collaboration diagram for PG25664CG_PORTB:

Public Member Functions

- **PG25664CG_PORTB** (uint8_t dc, uint8_t wr, uint8_t rd, uint8_t cs, uint8_t res, uint8_t doff)
- void initializeDevice ()

Additional Inherited Members

5.53.1 Member Function Documentation

5.53.1.1 void PG25664CG_PORTB::initializeDevice ( ) [inline], [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 PG25664CG.

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

- Drivers/PG25664CG/PG25664CG.h
- Drivers/PG25664CG/PG25664CG.cpp

5.54 point3d Struct Reference

Data Fields

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

- float **z**

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

- DisplayCore/DisplayCore.h

5.55 Raw565 Class Reference

Inheritance diagram for Raw565:

Collaboration diagram for Raw565:

Public Member Functions

- **Raw565** (const color_t *data, int w, int h)
- void **draw** (DisplayCore *dev, int x, int y)
- void **draw** (DisplayCore *dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore *dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore *dev, int x, int y, int transform, color_t t)
- void **draw** (DisplayCore &dev, int x, int y)
- void **draw** (DisplayCore &dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform, color_t t)

Additional Inherited Members

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

- ImageReaders/Raw565/Raw565.h
- ImageReaders/Raw565/Raw565.cpp

5.56 LCARS::RectButton Class Reference

Inheritance diagram for LCARS::RectButton:

Collaboration diagram for LCARS::RectButton:

Public Member Functions

- **RectButton** (Touch &ts, DisplayCore &dev, int x, int y, int w, int h, color_t off, color_t on, color_t hi, const uint8_t *f, const char *t)
- void **draw** (DisplayCore *dev, int x, int y)

Additional Inherited Members

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.57 ScreenDump Class Reference

Static Public Member Functions

- static void **dump565** (DisplayCore &dev, SDClass &sd, const char *filename)
- static void **dumpBMP** (DisplayCore &dev, SDClass &sd, const char *filename)

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

- Utilities/ScreenDump/ScreenDump.h
- Utilities/ScreenDump/ScreenDump.cpp

5.58 SDL Class Reference

Inheritance diagram for SDL:

Collaboration diagram for SDL:

Public Member Functions

- **SDL** (uint16_t w, uint16_t h, uint8_t t)
- **SDL** (uint8_t t)
- void initializeDevice ()
- void **setPixel** (int16_t x, int16_t y, uint16_t c)
- uint16_t **colorAt** (int16_t x, int16_t y)
- void **setRotation** (uint8_t r)
- void displayOn ()
- void displayOff ()
- void invertDisplay (boolean b)
- void fillScreen (uint16_t c)
- void **fillRectangle** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
- uint16_t getWidth ()
- uint16_t getHeight ()
- void startBuffer ()
- void endBuffer ()
- void **flip** ()
- void **hideCursor** ()
- void **showCursor** ()
- 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 uint8_t **Windowed** = 0
- static const uint8_t **Window** = 0
- static const uint8_t **Fullscreen** = 0x01
- static const uint8_t **DoubleBuffer** = 0x02
- static const uint8_t **Doublebuffer** = 0x02
- static const uint8_t **DoubleBuffered** = 0x02
- static const uint8_t **Doublebuffered** = 0x02

Additional Inherited Members

5.58.1 Member Function Documentation

5.58.1.1 void SDL::closeWindow ( ) [virtual]

Close the window

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

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

Reimplemented from DisplayCore.

5.58.1.2 void SDL::displayOff ( ) [inline], [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.58.1.3 void SDL::displayOn ( ) [inline], [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.58.1.4 void SDL::endBuffer ( ) [virtual]

End buffered mode

Any changes that are pending will be pushed out to the screen. See `startBuffer()` for more information.

Reimplemented from DisplayCore.

5.58.1.5 void SDL::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 DisplayCore.

5.58.1.6 uint16_t SDL::getHeight ( ) [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.58.1.7 uint16_t SDL::getWidth ( ) [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.58.1.8 void SDL::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 DisplayCore.

5.58.1.9 void SDL::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 DisplayCore.

5.58.1.10 void SDL::startBuffer ( ) [virtual]

Start buffered mode

In buffered mode, where applicable, any data that would be sent to the screen is delayed until buffered mode is ended. This generally has no effect on most screens, but those that use their own driver level may use this to delay pushing out of the buffer to the screen.

Reimplemented from DisplayCore.

5.58.1.11 void SDL::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 DisplayCore.

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

- Drivers/SDL/SDL.h
- Drivers/SDL/SDL.cpp

5.59 SDLTouch Class Reference

Inheritance diagram for SDLTouch:

Collaboration diagram for SDLTouch:

Public Member Functions

- **SDLTouch** (uint16_t w, uint16_t h)
- boolean **isPressed** ()
- uint16_t x ()
- uint16_t y ()
- uint16_t rawX ()
- uint16_t **rawY** ()
- void **setRotation** (uint8_t r)
- void sample ()
- void initializeDevice ()

Additional Inherited Members

5.59.1 Member Function Documentation

5.59.1.1 void SDLTouch::initializeDevice ( ) [inline], [virtual]

Initialize the device

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

Implements Touch.

5.59.1.2 uint16_t SDLTouch::rawX ( ) [inline], [virtual]

Get pressed status

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

Reimplemented from Touch.

5.59.1.3 void SDLTouch::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.

5.59.1.4 uint16_t SDLTouch::x ( ) [virtual]

Get X coordinate

This returns the X coordinate of the current touch position.

Implements Touch.

5.59.1.5 uint16_t SDLTouch::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:

- Drivers/SDL/SDL.h
- Drivers/SDL/SDL.cpp

5.60 SSD1289 Class Reference

Inheritance diagram for SSD1289:

Collaboration diagram for SSD1289:

Public Member Functions

- virtual void **command** (uint16_t)
- virtual void **data** (uint16_t)
- **SSD1289** (uint8_t rs, uint8_t wr, uint8_t rd, uint8_t cs, uint8_t reset, 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)
- void **startDisplay** ()
- void **setAddrWindow** (int x0, int y0, int x1, int y1)
- void fillScreen (color_t color)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- virtual void initializeDevice ()

- virtual void openWindow (int x0, int y0, int x1, int y1)
- virtual void windowData (color_t d)
- virtual void windowData (color_t *d, int l)
- virtual void closeWindow ()

Static Public Attributes

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

Additional Inherited Members

5.60.1 Member Function Documentation

5.60.1.1 void SSD1289::closeWindow ( ) [virtual]

Close the window

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

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

Reimplemented from DisplayCore.

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

Turn off the display

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

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

Implements DisplayCore.

5.60.1.3 void SSD1289::displayOn ( ) [inline], [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.60.1.4 void SSD1289::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.60.1.5 void SSD1289::drawVerticalLine (int *x*, int *y*, int *h*, color_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);
```

Reimplemented from DisplayCore.

5.60.1.6 void SSD1289::fillRectangle (int *x*, int *y*, int *w*, int *h*, color_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 DisplayCore.

5.60.1.7 void SSD1289::fillScreen (color_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 DisplayCore.

5.60.1.8 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 DisplayCore.

Reimplemented in SSD1289_PMP.

5.60.1.9 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 DisplayCore.

5.60.1.10 void SSD1289::openWindow (int *x0*, int *y0*, int *x1*, int *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 DisplayCore.

5.60.1.11 void SSD1289::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.60.1.12 void SSD1289::setRotation (int *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 DisplayCore.

5.60.1.13 void SSD1289::windowData (color_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 DisplayCore.

5.60.1.14 void SSD1289::windowData (color_t * d, int l) [virtual]

Send a block of pixel data to the window

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

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

Reimplemented from DisplayCore.

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

- Drivers/SSD1289/SSD1289.h
- Drivers/SSD1289/SSD1289.cpp

5.61 SSD1289_PMP Class Reference

Inheritance diagram for SSD1289_PMP:

Collaboration diagram for SSD1289_PMP:

Public Member Functions

- void **command** (uint16_t)
- void **data** (uint16_t)
- **SSD1289_PMP** (uint8_t res)
- void initializeDevice ()

Additional Inherited Members

5.61.1 Member Function Documentation

5.61.1.1 void SSD1289_PMP::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 SSD1289.

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

- Drivers/SSD1289/SSD1289.h
- Drivers/SSD1289/SSD1289.cpp

5.62 SSD1306 Class Reference

Inheritance diagram for SSD1306:

Collaboration diagram for SSD1306:

Public Member Functions

- **SSD1306** (DSPI &spi, int cs, int dc, int vdd, int vbat, int reset=-1)
- **SSD1306** (DSPI *spi, int cs, int dc, int vdd, int vbat, int reset=-1)
- **SSD1306** (int cs, int dc, int vdd, int vbat, int reset=-1)
- virtual void initializeDevice ()
- void setPixel (int x, int y, color_t color)
- void displayOn ()
- void displayOff ()
- void setRotation (int r)
- void invertDisplay (boolean i)
- void startBuffer ()
- void endBuffer ()
- int getWidth ()
- int getHeight ()
- void fillScreen (color_t color)

Protected Member Functions

- void **updateDisplay** ()
- virtual void **command** (uint8_t c)
- virtual void **command** (uint8_t c1, uint8_t c2)
- virtual void **command** (uint8_t c1, uint8_t c2, uint8_t c3)
- virtual void **data** (uint8_t d)
- void **setPage** (int p)
- void **setY** (int y)

Protected Attributes

- const int **DC_DATA** = HIGH
- const int **DC_COMMAND** = LOW
- const int **CMD_SEG_REMAP** = 0xA1
- const int **CMD_COM_DIR** = 0xC8
- const int **CMD_COM_CONFIG** = 0xDA
- const int **CMD_DISP_ON** = 0xAF
- const int **CMD_DISP_OFF** = 0xAE
- DSPI * **_spi**
- int **_cs**
- int **_dc**
- int **_vdd**
- int **_vbat**
- int **_reset**
- int **_buffered**
- uint8_t **_buffer** [8 *128]

Additional Inherited Members

5.62.1 Member Function Documentation

5.62.1.1 void SSD1306::displayOff ( ) [virtual]

Turn off the display

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

Example:

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

Implements DisplayCore.

5.62.1.2 void SSD1306::displayOn ( ) [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.62.1.3 void SSD1306::endBuffer ( ) [inline], [virtual]

End buffered mode

Any changes that are pending will be pushed out to the screen. See `startBuffer()` for more information.

Reimplemented from DisplayCore.

5.62.1.4 void SSD1306::fillScreen (color_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 DisplayCore.

5.62.1.5 int SSD1306::getHeight ( ) [inline], [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.62.1.6 int SSD1306::getWidth ( ) [inline], [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.62.1.7 void SSD1306::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 DisplayCore.

Reimplemented in SSD1306_BB.

5.62.1.8 void SSD1306::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 DisplayCore.

5.62.1.9 void SSD1306::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Implements DisplayCore.

5.62.1.10 void SSD1306::setRotation (int *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 DisplayCore.

5.62.1.11 void SSD1306::startBuffer ( ) [inline], [virtual]

Start buffered mode

In buffered mode, where applicable, any data that would be sent to the screen is delayed until buffered mode is ended. This generally has no effect on most screens, but those that use their own driver level may use this to delay pushing out of the buffer to the screen.

Reimplemented from DisplayCore.

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

- Drivers/SSD1306/SSD1306.h
- Drivers/SSD1306/SSD1306.cpp

5.63 SSD1306_BB Class Reference

Inheritance diagram for SSD1306_BB:

Collaboration diagram for SSD1306_BB:

Public Member Functions

- **SSD1306_BB** (uint32_t mosi, uint32_t sck, int cs, int dc, int vdd, int vbat, int reset=-1)
- virtual void initializeDevice ()

Additional Inherited Members

5.63.1 Member Function Documentation

5.63.1.1 void SSD1306_BB::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 SSD1306.

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

- Drivers/SSD1306/SSD1306.h
- Drivers/SSD1306/SSD1306.cpp

5.64 SSD1306_IOSHIELD Class Reference

Inheritance diagram for SSD1306_IOSHIELD:

Collaboration diagram for SSD1306_IOSHIELD:

Additional Inherited Members

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

- Drivers/SSD1306/SSD1306.h

5.65 SSD1306_UMOD_JA Class Reference

Inheritance diagram for SSD1306_UMOD_JA:

Collaboration diagram for SSD1306_UMOD_JA:

Additional Inherited Members

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

- Drivers/SSD1306/SSD1306.h

5.66 SSD1306_UMOD_JB Class Reference

Inheritance diagram for SSD1306_UMOD_JB:

Collaboration diagram for SSD1306_UMOD_JB:

Additional Inherited Members

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

- Drivers/SSD1306/SSD1306.h

5.67 SSD1306_UMOD_JD Class Reference

Inheritance diagram for SSD1306_UMOD_JD:

Collaboration diagram for SSD1306_UMOD_JD:

Additional Inherited Members

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

- Drivers/SSD1306/SSD1306.h

5.68 SSD1306_UMOD_JE Class Reference

Inheritance diagram for SSD1306_UMOD_JE:

Collaboration diagram for SSD1306_UMOD_JE:

Additional Inherited Members

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

- Drivers/SSD1306/SSD1306.h

5.69 SSD1963 Class Reference

Inheritance diagram for SSD1963:

Collaboration diagram for SSD1963:

Public Member Functions

- **SSD1963** (uint8_t rs, uint8_t wr, uint8_t rd, uint8_t cs, uint8_t reset, 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, uint8_t tft_bus_width=TFTBUS18)
- **SSD1963** (uint8_t rs, uint8_t wr, uint8_t rd, uint8_t cs, uint8_t reset, 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 tft_bus_width=TFTBUS18)
- void fillScreen (color_t color)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- color_t colorAt (int x, int y)
- void initializeDevice ()
- void windowData (color_t d)
- void openWindow (int, int, int, int)
- void enableBacklight ()
- void disableBacklight ()
- void setBacklight (int b)

Static Public Attributes

- static const int Width = 800
- static const int Height = 480

Additional Inherited Members

5.69.1 Member Function Documentation

5.69.1.1 color_t SSD1963::colorAt (int *x*, int *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 DisplayCore.

5.69.1.2 void SSD1963::disableBacklight ( ) [virtual]

Disable Back Light

For devices with their own backlight control this function will turn the backlight off.

Reimplemented from DisplayCore.

5.69.1.3 void SSD1963::displayOff ( ) [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.69.1.4 void SSD1963::displayOn ( ) [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.69.1.5 void SSD1963::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.69.1.6 void SSD1963::drawVerticalLine (int *x*, int *y*, int *h*, color_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);
```

Reimplemented from DisplayCore.

5.69.1.7 void SSD1963::enableBacklight ( ) [virtual]

Enable Back Light

For devices with their own backlight control this function will turn the backlight on. The brightness should be either the default brightness (typically full on) or the last brightness set with `setBacklight()`.

Reimplemented from DisplayCore.

5.69.1.8 void SSD1963::fillRectangle (int *x*, int *y*, int *w*, int *h*, color_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 DisplayCore.

5.69.1.9 void SSD1963::fillScreen (color_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 DisplayCore.

5.69.1.10 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 DisplayCore.

5.69.1.11 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 DisplayCore.

5.69.1.12 void SSD1963::openWindow (int *x0,* int *y0,* int *x1,* int *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 DisplayCore.

5.69.1.13 void SSD1963::setBacklight (int *b*) [virtual]

Set Back Light Brightness

For devices with their own backlight control this function will set the brightness of the backlight.

Reimplemented from DisplayCore.

5.69.1.14 void SSD1963::setPixel (int *x,* int *y,* color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.69.1.15 void SSD1963::setRotation (int *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 DisplayCore.

5.69.1.16 void SSD1963::windowData (color_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 DisplayCore.

5.69.2 Field Documentation

5.69.2.1 const int SSD1963::Height = 480 [static]

The height of the screen is 480 pixels

5.69.2.2 const int SSD1963::Width = 800 [static]

The width of the screen is 800 pixels

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

- Drivers/SSD1963/SSD1963.h
- Drivers/SSD1963/SSD1963.cpp

5.70 ST7735 Class Reference

Inheritance diagram for ST7735:

Collaboration diagram for ST7735:

Public Member Functions

- **ST7735** (DSPI *spi, uint8_t cs, uint8_t dc, uint8_t variant)
- **ST7735** (DSPI &spi, uint8_t cs, uint8_t dc, uint8_t variant)
- void fillScreen (color_t color)
- void setPixel (int x, int y, color_t color)
- void drawVerticalLine (int x, int y, int h, color_t color)
- void drawHorizontalLine (int x, int y, int w, color_t color)
- void fillRectangle (int x, int y, int w, int h, color_t color)
- void setRotation (int r)
- void invertDisplay (boolean i)
- void displayOn ()
- void displayOff ()
- void initializeDevice ()

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

5.70.1 Member Function Documentation

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

Turn off the display

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

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

Implements DisplayCore.

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

Turn on the display

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

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

Implements DisplayCore.

5.70.1.3 void ST7735::drawHorizontalLine (int *x*, int *y*, int *w*, color_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 DisplayCore.

5.70.1.4 void ST7735::drawVerticalLine (int *x*, int *y*, int *h*, color_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);
```

Reimplemented from DisplayCore.

5.70.1.5 void ST7735::fillRectangle (int *x*, int *y*, int *w*, int *h*, color_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 DisplayCore.

5.70.1.6 void ST7735::fillScreen (color_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 DisplayCore.

5.70.1.7 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 DisplayCore.

5.70.1.8 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 DisplayCore.

5.70.1.9 void ST7735::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Implements DisplayCore.

5.70.1.10 void ST7735::setRotation (int *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 DisplayCore.

5.70.2 Field Documentation

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

Adafruit screen with a black tab

5.70.2.2 const uint8_t ST7735::GreenTab = 0x00 [static]

Adafruit screen with a green tab

5.70.2.3 const uint8_t ST7735::Height = 160 [static]

The native size of the screen is 160 pixels high

5.70.2.4 const uint8_t ST7735::RedTab = 0x01 [static]

Adafruit screen with a red tab

5.70.2.5 const uint8_t ST7735::TypeB = 0x03 [static]

Adafruit "Type B" screen

5.70.2.6 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:

- Drivers/ST7735/ST7735.h
- Drivers/ST7735/ST7735.cpp

5.71 LCARS::StaticText Class Reference

Inheritance diagram for LCARS::StaticText:

Collaboration diagram for LCARS::StaticText:

Public Member Functions

- **StaticText** (Touch &ts, DisplayCore &dev, int x, int y, color_t col, const uint8_t *f, const char *txt)
- void setPixel (int x, int y, color_t c)
- void **draw** (DisplayCore *dev, int x, int y)
- void initializeDevice ()
- void **setAlign** (int align)
- void **setText** (const char *txt)

Static Public Attributes

- static const int **AlignLeft** = 0
- static const int **AlignRight** = 1
- static const int **AlignCenter** = 2

Additional Inherited Members

5.71.1 Member Function Documentation

5.71.1.1 void LCARS::StaticText::initializeDevice ( ) [inline], [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 Image.

5.71.1.2 void LCARS::StaticText::setPixel (int *x*, int *y*, color_t *color*) [inline], [virtual]

Draw a pixel

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

Example:

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

Reimplemented from Image.

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.72 Tint Class Reference

Inheritance diagram for Tint:

Collaboration diagram for Tint:

Public Member Functions

- **Tint** (color_t c)
- color_t **function** (color_t)
- void **setTint** (color_t c)

Additional Inherited Members

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

- Filters/Tint.h
- Filters/Tint.cpp

5.73 Touch Class Reference

Inheritance diagram for Touch:

Public Member Functions

- Touch (int w, int h)
- virtual void initializeDevice ()=0
- virtual int x ()=0
- virtual int y ()=0
- virtual int rawX ()
- virtual int **rawY** ()
- virtual boolean **isPressed** ()=0
- virtual int pressure ()
- virtual void setRotation (int r)=0
- virtual void sample ()=0
- virtual void **scaleX** (float x)
- virtual void **scaleY** (float y)
- virtual void **offsetX** (float x)
- virtual void **offsetY** (float y)

Data Fields

- float **_scale_x**
- float **_scale_y**
- int **_offset_x**
- int **_offset_y**

Protected Attributes

- int _width
- int _height

5.73.1 Constructor & Destructor Documentation

5.73.1.1 Touch::Touch (int *w*, int *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.

5.73.2 Member Function Documentation

5.73.2.1 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 LinuxEvent, SDLTouch, XPT2046, and AnalogTouch.

5.73.2.2 virtual int 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 LinuxEvent, and AnalogTouch.

5.73.2.3 virtual int Touch::rawX ( ) [inline], [virtual]

Get pressed status

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

Reimplemented in LinuxEvent, SDLTouch, and AnalogTouch.

5.73.2.4 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 LinuxEvent, SDLTouch, XPT2046, and AnalogTouch.

5.73.2.5 virtual void Touch::setRotation (int *r*) [pure 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.

Implemented in AnalogTouch, and XPT2046.

5.73.2.6 virtual int Touch::x ( ) [pure virtual]

Get X coordinate

This returns the X coordinate of the current touch position.

Implemented in LinuxEvent, SDLTouch, XPT2046, and AnalogTouch.

5.73.2.7 virtual int Touch::y ( ) [pure virtual]

Get Y coordinate

This returns the Y coordinate of the current touch position.

Implemented in LinuxEvent, SDLTouch, XPT2046, and AnalogTouch.

5.73.3 Field Documentation

5.73.3.1 int Touch::_height [protected]

The height of the touch screen in pixels

5.73.3.2 int Touch::_width [protected]

The width of the toush screen in pixels

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

- DisplayCore/DisplayCore.h

5.74 tsAnimIconData Struct Reference

Data Fields

- uint16_t **frames**
- uint16_t **width**
- uint16_t **height**
- uint16_t **data** []

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

- Toolkits/Widgets/twAnimIcon.h

5.75 twAnimIcon Class Reference

Inheritance diagram for twAnimIcon:

Collaboration diagram for twAnimIcon:

Public Member Functions

- **twAnimIcon** (Touch &ts, DisplayCore &dev, int x, int y, const char *txt, const uint16_t *data, const uint8_t *f)
- void **draw** (DisplayCore *dev, int x, int y)
- void **setBackgroundColor** (color_t c)
- void setTextColor (color_t c)
- void setFont (const uint8_t *f)
- void **setLabel** (const char *t)
- int getWidth ()
- int getHeight ()

Additional Inherited Members

5.75.1 Member Function Documentation

5.75.1.1 int twAnimIcon::getHeight ( ) [inline], [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from Image.

5.75.1.2 int twAnimIcon::getWidth ( ) [inline], [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from Image.

5.75.1.3 void twAnimIcon::setFont (const uint8_t * *f*) [inline], [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);
```

Reimplemented from DisplayCore.

5.75.1.4 void twAnimIcon::setTextColor (color_t *c*) [inline], [virtual]

Set the text foreground colour

Sets the foreground colour of all future printing to (c).
Example:

```
tft.setTextColor(Color::Magenta);
```

Reimplemented from DisplayCore.

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

- Toolkits/Widgets/twAnimIcon.h
- Toolkits/Widgets/twAnimIcon.cpp

5.76 twButton Class Reference

Inheritance diagram for twButton:

Collaboration diagram for twButton:

Public Member Functions

- **twButton** (Touch &ts, DisplayCore &dev, int x, int y, int w, int h, const char *txt)
- void **draw** (DisplayCore *dev, int x, int y)
- void **setBackgroundColor** (color_t c1, color_t c2)
- void setTextColor (color_t c1, color_t c2)
- void **setBevel** (int b)
- void **setBevelColor** (color_t hi, color_t low)
- void setFont (const uint8_t *f)
- void **setLabel** (char *t)

Additional Inherited Members

5.76.1 Member Function Documentation

5.76.1.1 void twButton::setFont (const uint8_t * *f*) [inline], [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);
```

Reimplemented from DisplayCore.

5.76.1.2 void twButton::setTextColor (color_t *fg*, color_t *bg*) [inline], [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);
```

Reimplemented from DisplayCore.

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

- Toolkits/Widgets/twButton.h
- Toolkits/Widgets/twButton.cpp

5.77 twHBar Class Reference

Inheritance diagram for twHBar:

Collaboration diagram for twHBar:

Public Member Functions

- **twHBar** (Touch &ts, DisplayCore &dev, int x, int y, int w, int h)
- void **draw** (DisplayCore *dev, int x, int y)

- void **setBorderColor** (color_t c)
- void **setScaleColor** (color_t c)
- void **setBackgroundColor** (color_t c)
- void **setMinimum** (int32_t m)
- void **setMaximum** (int32_t m)
- void **setValue** (int32_t m)

Additional Inherited Members

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

- Toolkits/Widgets/twHBar.h
- Toolkits/Widgets/twHBar.cpp

5.78 twIcon Class Reference

Inheritance diagram for twIcon:

Collaboration diagram for twIcon:

Public Member Functions

- **twIcon** (Touch &ts, DisplayCore &dev, int x, int y, int w, int h, const char *txt, const color_t *icon, const uint8_t *f)
- void **draw** (DisplayCore *dev, int x, int y)
- void **setBackgroundColor** (color_t c)
- void setTextColor (color_t c)
- void setFont (const uint8_t *f)
- void **setLabel** (const char *t)

Additional Inherited Members

5.78.1 Member Function Documentation

5.78.1.1 void twIcon::setFont (const uint8_t * *f*) [inline], [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);
```

Reimplemented from DisplayCore.

5.78.1.2 void twIcon::setTextColor (color_t *c*) [inline], [virtual]

Set the text foreground colour

Sets the foreground colour of all future printing to (c).
Example:

```
tft.setTextColor(Color::Magenta);
```

Reimplemented from DisplayCore.

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

- Toolkits/Widgets/twIcon.h
- Toolkits/Widgets/twIcon.cpp

5.79 twText Class Reference

Inheritance diagram for twText:

Collaboration diagram for twText:

Public Member Functions

- **twText** (Touch &ts, DisplayCore &dev, int x, int y, color_t col, const uint8_t *f, const char *txt)
- void setPixel (int x, int y, color_t c)
- void **draw** (DisplayCore *dev, int x, int y)
- void initializeDevice ()
- void **setAlign** (uint8_t align)
- void **setText** (const char *txt)

Static Public Attributes

- static const uint8_t **AlignLeft** = 0
- static const uint8_t **AlignRight** = 1
- static const uint8_t **AlignCenter** = 2

Additional Inherited Members

5.79.1 Member Function Documentation

5.79.1.1 void twText::initializeDevice ( ) [inline], [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 Image.

5.79.1.2 void twText::setPixel (int *x*, int *y*, color_t *color*) [inline], [virtual]

Draw a pixel

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

Example:

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

Reimplemented from Image.

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

- Toolkits/Widgets/twText.h
- Toolkits/Widgets/twText.cpp

5.80 VGA Class Reference

Inheritance diagram for VGA:

Collaboration diagram for VGA:

Public Member Functions

- **VGA** (uint8_t hsync, uint8_t vsync)
- void initializeDevice ()
- void setPixel (int x, int y, color_t c)
- void setRotation (int r)
- void displayOn ()
- void displayOff ()
- void invertDisplay (boolean i)
- int getWidth ()
- int getHeight ()
- void **vblank** ()
- void **flip** ()
- void fillScreen (color_t c)
- uint32_t **millis** ()

Static Public Attributes

- static const int **Width** = 768 / (VGA_USE_H_RES + 1)
- static const int **Height** = 480 / (VGA_USE_V_RES + 1)

Additional Inherited Members

5.80.1 Member Function Documentation

5.80.1.1 void VGA::displayOff ( ) [inline], [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.80.1.2 void VGA::displayOn ( ) [inline], [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.80.1.3 void VGA::fillScreen (color_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 DisplayCore.

5.80.1.4 int VGA::getHeight ( ) [inline], [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.80.1.5 int VGA::getWidth ( ) [inline], [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.80.1.6 void VGA::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 DisplayCore.

5.80.1.7 void VGA::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 DisplayCore.

5.80.1.8 void VGA::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

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

Implements DisplayCore.

5.80.1.9 void VGA::setRotation (int *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 DisplayCore.

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

- Drivers/VGA/VGA.h
- Drivers/VGA/VGA.cpp

5.81 VLCD Class Reference

Inheritance diagram for VLCD:

Collaboration diagram for VLCD:

Public Member Functions

- void initializeDevice ()
- void **initializeDevice** (Stream &s)
- void **setSize** (int x, int y)
- void **sendData** (uint8_t *data)
- void setPixel (int x, int y, color_t c)
- void drawLine (int x0, int y0, int x1, int y1, color_t c)
- void setRotation (int r)

- void displayOn ()
- void displayOff ()
- void invertDisplay (boolean b)
- void **setForeground** (color_t c)
- void **setBackground** (color_t c)
- void **setBaud** (uint32_t b)
- int getWidth ()
- int getHeight ()

Additional Inherited Members

5.81.1 Member Function Documentation

5.81.1.1 void VLCD::displayOff ( ) [inline], [virtual]

Turn off the display

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

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

Implements DisplayCore.

5.81.1.2 void VLCD::displayOn ( ) [inline], [virtual]

Turn on the display

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

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

Implements DisplayCore.

5.81.1.3 void VLCD::drawLine (int *x0*, int *y0*, int *x1*, int *y1*, color_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);
```

Reimplemented from DisplayCore.

5.81.1.4 int VLCD::getHeight ( ) [inline], [virtual]

Get screen height

Returns the height (in pixels) of the screen.

Example:

int height = tft.getHeight();

Reimplemented from DisplayCore.

5.81.1.5 int VLCD::getWidth ( ) [inline], [virtual]

Get screen width

Returns the width (in pixels) of the screen.

Example:

int width = tft.getWidth();

Reimplemented from DisplayCore.

5.81.1.6 void VLCD::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 DisplayCore.

5.81.1.7 void VLCD::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 DisplayCore.

5.81.1.8 void VLCD::setPixel (int *x*, int *y*, color_t *color*) [virtual]

Draw a pixel

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

Example:

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

Implements DisplayCore.

5.81.1.9 void VLCD::setRotation (int *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 DisplayCore.

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

- Drivers/VLCD/VLCD.h
- Drivers/VLCD/VLCD.cpp

5.82 LCARS::VScale Class Reference

Inheritance diagram for LCARS::VScale:

Collaboration diagram for LCARS::VScale:

Public Member Functions

- **VScale** (Touch &ts, DisplayCore &dev, int x, int y, color_t lowCol, color_t hiCol, color_t overCol)
- void **setValue** (int v)
- void **draw** (DisplayCore *dev, int x, int y)
- void **render** ()

Additional Inherited Members

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

- Toolkits/LCARSInterface/LCARSInterface.h
- Toolkits/LCARSInterface/LCARSInterface.cpp

5.83 Widget Class Reference

Inheritance diagram for Widget:

Collaboration diagram for Widget:

Public Member Functions

- **Widget** (Touch &t, DisplayCore &d, int x, int y)
- virtual void **setValue** (int v)
- virtual int **getValue** ()
- virtual void **render** ()
- void **setUserValue** (uint32_t v)
- uint32_t **getUserValue** ()
- void **onPress** (void(*func)(Event *))
- void **onRelease** (void(*func)(Event *))
- void **onDrag** (void(*func)(Event *))
- void **onTap** (void(*func)(Event *))

- void **onRepeat** (void(∗func)(Event ∗))
- void **handleTouch** ()
- virtual void **draw** (DisplayCore ∗dev, int x, int y)=0
- void **draw** (DisplayCore ∗dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore ∗dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore ∗dev, int x, int y, int transform, color_t t)
- void **draw** (DisplayCore &dev, int x, int y)
- void **draw** (DisplayCore &dev, int x, int y, color_t t)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform)
- void **drawTransformed** (DisplayCore &dev, int x, int y, int transform, color_t t)
- virtual void **redraw** ()
- virtual void **setEnabled** (boolean e)
- virtual boolean **isEnabled** ()
- virtual boolean **isActive** ()
- void **setLocation** (int x, int y)

Data Fields

- int **_sense_x**
- int **_sense_y**
- int **_sense_w**
- int **_sense_h**

Protected Attributes

- Touch ∗ **_ts**
- DisplayCore ∗ **_dev**
- int **_x**
- int **_y**
- int **_value**
- uint32_t **_user**
- boolean **_redraw**
- uint32_t **_dbStart**
- boolean **_dbPressed**
- int **_sx**
- int **_sy**
- int **_ex**
- int **_ey**
- int **_rx**
- int **_ry**
- uint32_t **_st**
- uint32_t **_rt**
- uint32_t **_et**
- int **_rc**
- int **_rp**
- int **_tx**
- int **_ty**
- boolean **_active**
- boolean **_touch**
- boolean **_enabled**
- void(∗ **_press**)(Event ∗)
- void(∗ **_release**)(Event ∗)
- void(∗ **_drag**)(Event ∗)
- void(∗ **_tap**)(Event ∗)
- void(∗ **_repeat**)(Event ∗)

Additional Inherited Members

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

- DisplayCore/DisplayCore.h
- DisplayCore/DisplayCore.cpp

5.84 XPT2046 Class Reference

Inheritance diagram for XPT2046:

Collaboration diagram for XPT2046:

Public Member Functions

- void sample ()
- int x ()
- int y ()
- boolean **isPressed** ()
- void initializeDevice ()
- void setRotation (int r)

- XPT2046 (DSPI *spi, uint8_t cs, int w, int h)
- **XPT2046** (DSPI &spi, uint8_t cs, int w, int h)

Additional Inherited Members

5.84.1 Constructor & Destructor Documentation

5.84.1.1 XPT2046::XPT2046 (DSPI * *spi,* uint8_t *cs,* int *w,* int *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);
```

5.84.2 Member Function Documentation

5.84.2.1 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.

5.84.2.2 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.

5.84.2.3 void XPT2046::setRotation (int *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.

Implements Touch.

5.84.2.4 int XPT2046::x ( ) [virtual]

Get X coordinate

This returns the X coordinate of the current touch position.

Implements Touch.

5.84.2.5 int 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:

- Drivers/XPT2046/XPT2046.h
- Drivers/XPT2046/XPT2046.cpp

Index