SH1106 Display Driver - Kala, Jaraczewski

Generated by Doxygen 1.9.2

1 Hierarchical Index

1.1 Class Hierarchy

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

2 Class Index

2.1 Class List

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

3 File Index

3.1 File List

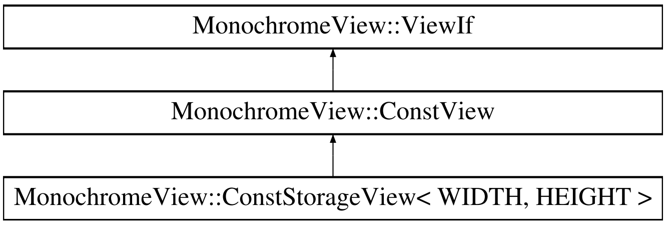
Here is a list of all documented files with brief descriptions:

4 Class Documentation

4.1 MonochromeView::ConstStorageView< WIDTH, HEIGHT > Class Template Reference

```
#include <ConstStorageView.hpp>
```

Inheritance diagram for MonochromeView::ConstStorageView< WIDTH, HEIGHT >:


MonochromeView::ViewIf
MonochromeView::ConstView
MonochromeView::ConstStorageView< WIDTH, HEIGHT >

Public Member Functions

- template<typename... ViewBytesTypes>
  ConstStorageView (ViewBytesTypes... viewBytes)

Additional Inherited Members

4.1.1 Detailed Description

template<size_t WIDTH, size_t HEIGHT>
class MonochromeView::ConstStorageView< WIDTH, HEIGHT >

Constant view with internal storage.

Template Parameters

WIDTH	Width of a view.
HEIGHT	Height of a view.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 ConstStorageView()

```
template<size_t WIDTH, size_t HEIGHT>
template<typename... ViewBytesTypes>
MonochromeView::ConstStorageView< WIDTH, HEIGHT >::ConstStorageView (
            ViewBytesTypes... viewBytes ) [inline]
```

Construct a new constant storage view object.

Template Parameters

ViewBytesTypes	Types of view bytes.

Parameters

viewBytes	List of bytes to store for a view. The first element is a left-top cell (1 column and 8 rows pixels). The first column is repeated for each next 'width' pixels.

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

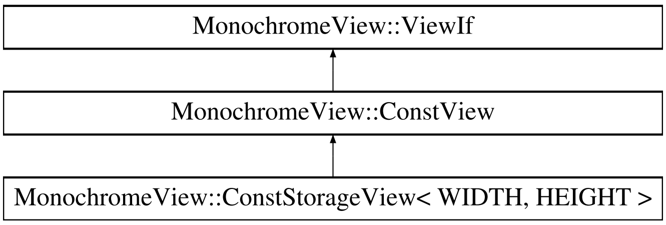
- DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstStorageView.hpp

4.2 MonochromeView::ConstView Class Reference

Constant view.

```
#include <ConstView.hpp>
```

Inheritance diagram for MonochromeView::ConstView:


MonochromeView::ViewIf
MonochromeView::ConstView
MonochromeView::ConstStorageView< WIDTH, HEIGHT >

Public Member Functions

- ConstView (const uint8_t ∗const pBuffer, const size_t width, const size_t height)
- size_t Width () const override

  Width of an view in pixels.
- size_t Height () const override

  Height of an view in pixels.
- bool GetPixelColor (const size_t x, const size_t y) const override
- bool IfViewChanged () const

Additional Inherited Members

4.2.1 Detailed Description

Constant view.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 ConstView()

```
MonochromeView::ConstView::ConstView (
            const uint8_t *const pBuffer,
            const size_t width,
            const size_t height )
```

Construct a new constant view object.

Parameters

pBuffer	Pointer to the view data buffer. The first element is a left-top cell (1 column and 8 rows pixels). The first column is repeated for each next 'width' pixels.
width	Width of a view.
height	Height of a view.

4.2.3 Member Function Documentation

4.2.3.1 GetPixelColor()

```
bool MonochromeView::ConstView::GetPixelColor (
        const size_t x,
        const size_t y ) const  [override], [virtual]
```

Get color of the pixel.

Parameters

x	Horizontal coordinate.
y	Vertical coordinate.

Returns

Color of the pixel. [0 - black, 1 - default color].

Implements MonochromeView::ViewIf.

4.2.3.2 Height()

```
size_t MonochromeView::ConstView::Height ( ) const  [override], [virtual]
```

Height of an view in pixels.

Implements MonochromeView::ViewIf.

4.2.3.3 IfViewChanged()

```
bool MonochromeView::ConstView::IfViewChanged ( ) const  [virtual]
```

Implements MonochromeView::ViewIf.

4.2.3.4 Width()

```
size_t MonochromeView::ConstView::Width ( ) const  [override], [virtual]
```

Width of an view in pixels.

Implements MonochromeView::ViewIf.

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

- DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstView.hpp

4.3 DisplayComm::DisplayCommIf Class Reference

Interface for communication with display.

```
#include <DisplayCommIf.hpp>
```

Public Member Functions

- virtual bool WriteCmd (const uint8_t cmd) const =0
- virtual bool WriteData (const uint8_t ∗const pData, const size_t dataSize) const =0

4.3.1 Detailed Description

Interface for communication with display.

4.3.2 Member Function Documentation

4.3.2.1 WriteCmd()

```
virtual bool DisplayComm::DisplayCommIf::WriteCmd (
        const uint8_t cmd ) const [pure virtual]
```

Write command to the display.

Parameters

cmd	Command code.

Returns

Write result.

4.3.2.2 WriteData()

```
virtual bool DisplayComm::DisplayCommIf::WriteData (
        const uint8_t *const pData,
        const size_t dataSize ) const [pure virtual]
```

Write data to the display.

Parameters

pData	Pointer to the data buffer.
dataSize	Data size in bytes.

Returns

Write result.

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

- DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayCommIf.hpp

4.4 DisplayComm::DisplayDataCmdIf Class Reference

Interface for changing mode of the display.

```
#include <DisplayDataCmdIf.hpp>
```

Public Member Functions

- virtual void **EnableCmdMode** () const =0

  Enable command mode in the display.
- virtual void **EnableDataMode** () const =0

  Enable data mode in the display.

4.4.1 Detailed Description

Interface for changing mode of the display.

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

- DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayDataCmdIf.hpp

4.5 MonochromeGraphicDisplay::DisplayDriverIf Class Reference

Diver interface of graphical screens.

```
#include <DisplayDriverIf.hpp>
```

Public Member Functions

- virtual bool TurnOnDisplay ()=0
- virtual bool TurnOffDisplay ()=0
- virtual bool InverseColor (bool inverse)=0
- virtual bool SetContrast (uint8_t value)=0
- virtual bool RefreshScreen ()=0
- virtual MonochromeView::DynamicView & GetView ()=0
- virtual size_t GetWidth ()=0
- virtual size_t GetHeight ()=0

4.5.1 Detailed Description

Diver interface of graphical screens.

4.5.2 Member Function Documentation

4.5.2.1 GetHeight() `virtual size_t MonochromeGraphicDisplay::DisplayDriverIf::GetHeight ( ) [pure virtual]`

Get the display height.

Returns

Height in pixels.

4.5.2.2 GetView() `virtual MonochromeView::DynamicView & MonochromeGraphicDisplay::DisplayDriverIf::GetView ( ) [pure virtual]`

Get display view object.

Returns

Reference to display view object.

4.5.2.3 GetWidth() `virtual size_t MonochromeGraphicDisplay::DisplayDriverIf::GetWidth ( ) [pure virtual]`

Get the display width.

Returns

Width in pixels.

4.5.2.4 InverseColor() `virtual bool MonochromeGraphicDisplay::DisplayDriverIf::InverseColor ( bool inverse ) [pure virtual]`

Inverse display colors.

Parameters

color	[0 - normal, 1 - inversed]

Returns

Result.

4.5.2.5 RefreshScreen() `virtual bool MonochromeGraphicDisplay::DisplayDriverIf::RefreshScreen ( ) [pure virtual]`

Refresh screen.

Returns

Result.

4.5.2.6 SetContrast() `virtual bool MonochromeGraphicDisplay::DisplayDriverIf::SetContrast ( uint8_t value ) [pure virtual]`

Set the contrast.

Parameters

value	Contrast [0u..255u].

Returns

Result.

4.5.2.7 TurnOffDisplay() `virtual bool MonochromeGraphicDisplay::DisplayDriverIf::TurnOffDisplay ( ) [pure virtual]`

Turns off display.

Returns

Result.

4.5.2.8 TurnOnDisplay() `virtual bool MonochromeGraphicDisplay::DisplayDriverIf::TurnOnDisplay ( ) [pure virtual]`

Turns on display.

Returns

Result.

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

- DisplayApp/App/MonochromeGraphicDisplay/Inc/MonochromeGraphicDisplay/DisplayDriverIf.hpp

4.6 DisplayComm::DisplayResetIf Class Reference

Interface for the reseting display.

`#include <DisplayResetIf.hpp>`

Public Member Functions

- virtual void **Reset** () const =0

  Reset display.

4.6.1 Detailed Description

Interface for the reseting display.

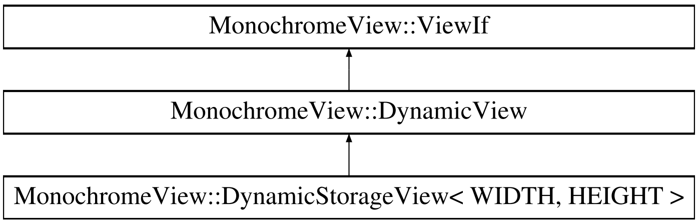
The documentation for this class was generated from the following file:

- DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayResetIf.hpp

4.7 MonochromeView::DynamicStorageView< WIDTH, HEIGHT > Class Template Reference

`#include <DynamicStorageView.hpp>`

Inheritance diagram for MonochromeView::DynamicStorageView< WIDTH, HEIGHT >:


MonochromeView::ViewIf
MonochromeView::DynamicView
MonochromeView::DynamicStorageView< WIDTH, HEIGHT >

Public Member Functions

- template<typename... ViewBytesTypes>
  DynamicStorageView (ViewBytesTypes... viewBytes)

Additional Inherited Members

4.7.1 Detailed Description

template<size_t WIDTH, size_t HEIGHT>
class MonochromeView::DynamicStorageView< WIDTH, HEIGHT >

Dynamic view with internal storage.

Template Parameters

WIDTH	Width of a view.
HEIGHT	Height of a view.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 DynamicStorageView()

```
template<size_t WIDTH, size_t HEIGHT>
template<typename... ViewBytesTypes>
MonochromeView::DynamicStorageView< WIDTH, HEIGHT >::DynamicStorageView (
              ViewBytesTypes...  viewBytes )  [inline]
```

Construct a new dynamic storage view object.

Template Parameters

ViewBytesTypes	Types of view bytes.

Parameters

viewBytes	List of bytes to store for a view. The first element is a left-top cell (1 column and 8 rows pixels). The first column is repeated for each next 'width' pixels.

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

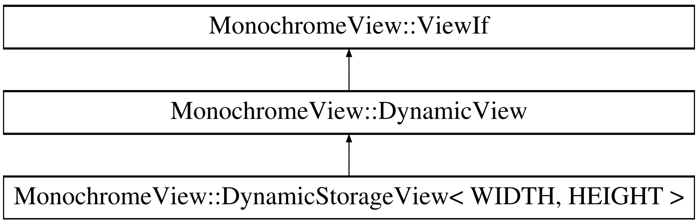
- DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicStorageView.hpp

4.8 MonochromeView::DynamicView Class Reference

Dynamic view.

```
#include <DynamicView.hpp>
```

Inheritance diagram for MonochromeView::DynamicView:


MonochromeView::ViewIf
MonochromeView::DynamicView
MonochromeView::DynamicStorageView< WIDTH, HEIGHT >

Public Member Functions

- DynamicView (uint8_t ∗const pBuffer, const size_t width, const size_t height)
- size_t Width () const override

  Width of an view in pixels.
- size_t Height () const override

  Height of an view in pixels.
- bool GetPixelColor (const size_t x, const size_t y) const override
- bool IfViewChanged () const
- void DrawAt (const int32_t x, const int32_t y, const ViewIf &rAnotherView, const uint8_t drawOption=DRAW↩ _OPT_NONE)
- void DrawLine (const int32_t x1, const int32_t y1, const int32_t x2, const int32_t y2, const bool color)
- void SetPixelColor (const size_t x, const size_t y, const bool color)
- void Fill (const bool color)

Additional Inherited Members

4.8.1 Detailed Description

Dynamic view.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 DynamicView()

```
MonochromeView::DynamicView::DynamicView (
        uint8_t ∗const pBuffer,
        const size_t width,
        const size_t height )
```

Construct a new constant view object.

Parameters

pBuffer	Pointer to the view data buffer. The first element is a left-top cell (1 column and 8 rows pixels). The first column is repeated for each next 'width' pixels.
width	Width of a view.
height	Height of a view.

4.8.3 Member Function Documentation

4.8.3.1 DrawAt()

```
void MonochromeView::DynamicView::DrawAt (
        const int32_t x,
        const int32_t y,
        const ViewIf & rAnotherView,
        const uint8_t drawOption = DRAW_OPT_NONE )
```

Draw at {x, y} another view.

Parameters

x	Horizontal coordinate.
y	Vertical coordinate.
rAnotherView	View to draw.
drawOption	Drawing option DRAW_OPT_TRANSPOSE or DRAW_OPT_X_MIRROR or DRAW_OPT_Y_MIRROR

4.8.3.2 DrawLine()

```
void MonochromeView::DynamicView::DrawLine (
              const int32_t x1,
              const int32_t y1,
              const int32_t x2,
              const int32_t y2,
              const bool color )
```

Draw line between {x1, y1} and {x2, y2} points.

Parameters

x1	First point position in pixles, starting from left edge.
y1	First point position in pixles, starting from top edge.
x2	Second point position in pixles, starting from left edge.
y2	Second point position in pixles, starting from top edge.
color	[0 - black, 1 - default display color]

4.8.3.3 Fill()

```
void MonochromeView::DynamicView::Fill (
              const bool color )
```

Fill the whole view with a single color.

Parameters

color	Color to fill view. [0 - black, 1 - default color].

4.8.3.4 GetPixelColor()

```
bool MonochromeView::DynamicView::GetPixelColor (
              const size_t x,
              const size_t y ) const  [override], [virtual]
```

Get color of the pixel.

Parameters

x	Horizontal coordinate.
y	Vertical coordinate.

Returns

Color of the pixel. [0 - black, 1 - default color].

Implements MonochromeView::ViewIf.

4.8.3.5 Height() `size_t MonochromeView::DynamicView::Height ( ) const [override], [virtual]`

Height of an view in pixels.

Implements MonochromeView::ViewIf.

4.8.3.6 IfViewChanged() `bool MonochromeView::DynamicView::IfViewChanged ( ) const [virtual]`

Implements MonochromeView::ViewIf.

4.8.3.7 SetPixelColor() `void MonochromeView::DynamicView::SetPixelColor (`

```
const size_t x,
const size_t y,
const bool color )
```

Set color of a pixel at {x, y}.

Parameters

x	Horizontal coordinate.
y	Vertical coordinate.
color	Color of the pixel. [0 - black, 1 - default color].

4.8.3.8 Width() `size_t MonochromeView::DynamicView::Width ( ) const [override], [virtual]`

Width of an view in pixels.

Implements MonochromeView::ViewIf.

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

- DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicView.hpp

4.9 DisplayComm::Factory Class Reference

Factory for display communication layer.

```
#include <Factory.hpp>
```

Static Public Member Functions

- static DisplayResetIf ∗ CreateDisplayReset (GPIO_TypeDef ∗const pResetPort, const uint32_t resetPin)
- static DisplayDataCmdIf ∗ CreateDataCmd (GPIO_TypeDef ∗const pDcPort, const uint32_t dcPin)
- static DisplayCommIf ∗ CreateDisplayCommSpi (SPI_TypeDef ∗const pSpi, GPIO_TypeDef ∗const pCsPort, const uint32_t csPin, const DisplayDataCmdIf ∗const pDataCmdIf)

4.9.1 Detailed Description

Factory for display communication layer.

4.9.2 Member Function Documentation

4.9.2.1 CreateDataCmd()

```
static DisplayDataCmdIf ∗ DisplayComm::Factory::CreateDataCmd (
            GPIO_TypeDef ∗const pDcPort,
            const uint32_t dcPin )  [static]
```

Create a data command driver object.

Parameters

pDcPort	Port of D/C pin.
dcPin	D/C pin number.

Note

User takes responsibility for managing lifetime of returned object!

Returns

DisplayResetIf∗ Pointer to a newly allocated D/C driver object.

4.9.2.2 CreateDisplayCommSpi()

```
static DisplayCommIf ∗ DisplayComm::Factory::CreateDisplay←
CommSpi (
            SPI_TypeDef ∗const pSpi,
            GPIO_TypeDef *const pCsPort,
            const uint32_t csPin,
            const DisplayDataCmdIf *const pDataCmdIf )  [static]
```

Create a display comm object.

Parameters

pSpi	Pointer to the SPI interface.
pCsPort	Port of the chip select pin.
csPin	Chip select pin.
pData↩ CmdIf	Pointer to data command interface.

Note

User takes responsibility for managing lifetime of returned object!

Returns

DisplayCommIf∗ Pointer to a newly allocated communication driver object.

4.9.2.3 CreateDisplayReset()

```
static DisplayResetIf * DisplayComm::Factory::CreateDisplayReset (
        GPIO_TypeDef *const pResetPort,
        const uint32_t resetPin )  [static]
```

Create a display reset object.

Parameters

pResetPort	Port of the reset pin.
resetPin	Reset pin number.

Note

User takes responsibility for managing lifetime of returned object!

Returns

DisplayResetIf∗ Pointer to a newly allocated reset object.

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

- DisplayApp/App/DisplayComm/Inc/DisplayComm/Factory.hpp

4.10 Sh1106::Factory Class Reference

Factory of SH1106 dirvers.

```
#include <Factory.hpp>
```

Static Public Member Functions

- static MonochromeGraphicDisplay::DisplayDriverIf ∗ Create128x32Driver (DisplayComm::DisplayCommIf ∗const pDisplayCommIf, DisplayComm::DisplayResetIf ∗const pDisplayResetIf, const bool mirror↩ Vertically=false, const bool mirrorHorizontally=false, const size_t columnOffset=0U)
- static MonochromeGraphicDisplay::DisplayDriverIf ∗ Create128x64Driver (DisplayComm::DisplayCommIf ∗const pDisplayCommIf, DisplayComm::DisplayResetIf ∗const pDisplayResetIf, const bool mirror↩ Vertically=false, const bool mirrorHorizontally=false, const size_t columnOffset=0U)
- static MonochromeGraphicDisplay::DisplayDriverIf ∗ Create128x128Driver (DisplayComm::DisplayCommIf ∗const pDisplayCommIf, DisplayComm::DisplayResetIf ∗const pDisplayResetIf, const bool mirror↩ Vertically=false, const bool mirrorHorizontally=false, const size_t columnOffset=0U)

4.10.1 Detailed Description

Factory of SH1106 dirvers.

4.10.2 Member Function Documentation

4.10.2.1 Create128x128Driver()

```
static MonochromeGraphicDisplay::DisplayDriverIf * Sh1106::↩
Factory::Create128x128Driver (
             DisplayComm::DisplayCommIf *const pDisplayCommIf,
             DisplayComm::DisplayResetIf *const pDisplayResetIf,
             const bool mirrorVertically = false,
             const bool mirrorHorizontally = false,
             const size_t columnOffset = 0U )  [static]
```

Create a SH1106 driver instance for resolution 128x128 px (width x height).

Parameters

pDisplayCommIf	Pointer to the display communication interface.
pDisplayResetIf	Pointer to the display reset interface.
mirrorVertically	Set to true to mirror vertically.
mirrorHorizontally	Set to false to mirror horizontally.
columnOffset	Offset of the first column.

Note

User takes responsibility for managing lifetime of returned object!

Returns

Pointer to a newly allocated SH1106 driver object.

4.10.2.2 Create128x32Driver()

```
static MonochromeGraphicDisplay::DisplayDriverIf * Sh1106::↩
Factory::Create128x32Driver (
            DisplayComm::DisplayCommIf *const pDisplayCommIf,
            DisplayComm::DisplayResetIf *const pDisplayResetIf,
            const bool mirrorVertically = false,
            const bool mirrorHorizontally = false,
            const size_t columnOffset = 0U )  [static]
```

Create a SH1106 driver instance for resolution 128x32 px (width x height).

Parameters

pDisplayCommIf	Pointer to the display communication interface.
pDisplayResetIf	Pointer to the display reset interface.
mirrorVertically	Set to true to mirror vertically.
mirrorHorizontally	Set to false to mirror horizontally.
columnOffset	Offset of the first column.

Note

User takes responsibility for managing lifetime of returned object!

Returns

Pointer to a newly allocated SH1106 driver object.

4.10.2.3 Create128x64Driver()

```
static MonochromeGraphicDisplay::DisplayDriverIf * Sh1106::↩
Factory::Create128x64Driver (
            DisplayComm::DisplayCommIf *const pDisplayCommIf,
            DisplayComm::DisplayResetIf *const pDisplayResetIf,
            const bool mirrorVertically = false,
            const bool mirrorHorizontally = false,
            const size_t columnOffset = 0U )  [static]
```

Create a SH1106 driver instance for resolution 128x64 px (width x height).

Parameters

pDisplayCommIf	Pointer to the display communication interface.
pDisplayResetIf	Pointer to the display reset interface.
mirrorVertically	Set to true to mirror vertically.
mirrorHorizontally	Set to false to mirror horizontally.
columnOffset	Offset of the first column.

Note

User takes responsibility for managing lifetime of returned object!

Returns

Pointer to a newly allocated SH1106 driver object.

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

- DisplayApp/App/Sh1106/Inc/Sh1106/Factory.hpp

4.11 MonochromeText::MonochromeFont Class Reference

Monochrome font.

```
#include <MonochromeFont.hpp>
```

Public Member Functions

- MonochromeFont (const uint8_t ∗const pCharsBytes, const size_t width, const size_t height, const size_t firstCharAsciiOffset, const size_t charsNum)
- uint8_t GetWidth () const
- uint8_t GetHeight () const
- MonochromeView::ConstView GetCharView (const char character) const

4.11.1 Detailed Description

Monochrome font.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 MonochromeFont()

```
MonochromeText::MonochromeFont::MonochromeFont (
        const uint8_t *const pCharsBytes,
        const size_t width,
        const size_t height,
        const size_t firstCharAsciiOffset,
        const size_t charsNum )
```

Construct a new Monochrome Font object

Parameters

pCharsViews	Pointer to font characters bytes array. First element should has ASCII code.
width	Font width in pixels.
height	Font height in pixels.
firstCharAsciiOffset	Offset in ASCII table of the first character in font.
charsNum	Number of characters in font.

4.11.3 Member Function Documentation

4.11.3.1 GetCharView() `MonochromeView::ConstView MonochromeText::MonochromeFont::GetCharView ( const char character ) const`

Get character view.

Parameters

character	Character to get view of.

Returns

Character view.

4.11.3.2 GetHeight() `uint8_t MonochromeText::MonochromeFont::GetHeight ( ) const`

Get the Height object

Returns

uint8_t Font height in pixels.

4.11.3.3 GetWidth() `uint8_t MonochromeText::MonochromeFont::GetWidth ( ) const`

Get the Width object

Returns

uint8_t Font width in pixels.

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

- DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeFont.hpp

4.12 MonochromeText::MonochromeText Class Reference

Monochrome text.

```
#include <MonochromeText.hpp>
```

Static Public Member Functions

- static void WriteChar (MonochromeView::DynamicView &rView, const int32_t x, const int32_t y, const MonochromeFont &rFont, const char character, const uint8_t drawOption=MonochromeView::DRAW_OPT↩ _NONE)
- static void WriteString (MonochromeView::DynamicView &rView, const int32_t x, const int32_t y, const MonochromeFont &rFont, const char ∗const pString, const uint8_t drawOption=MonochromeView::DRAW↩ _OPT_NONE)

4.12.1 Detailed Description

Monochrome text.

4.12.2 Member Function Documentation

4.12.2.1 WriteChar()

```
static void MonochromeText::MonochromeText::WriteChar (
    MonochromeView::DynamicView & rView,
    const int32_t x,
    const int32_t y,
    const MonochromeFont & rFont,
    const char character,
    const uint8_t drawOption = MonochromeView::DRAW_OPT_NONE )  [static]
```

Write 'character' at {x, y} point in the view 'rView' using font 'rFont' and draw options 'drawOption'.

Parameters

rView	View to write character in.
x	Horizontal coordinate.
y	Vertical coordinate.
rFont	Font.
character	Character to write.
drawOption	Draw options,

See also

MonochromeView::DynamicView::DrawAt

4.12.2.2 WriteString()

```
static void MonochromeText::MonochromeText::WriteString (
    MonochromeView::DynamicView & rView,
    const int32_t x,
    const int32_t y,
    const MonochromeFont & rFont,
    const char *const pString,
    const uint8_t drawOption = MonochromeView::DRAW_OPT_NONE )  [static]
```

Write string 'pString' at {x, y} point in the view 'rView' using font 'rFont' and draw options 'drawOption'.

Parameters

rView	View to write string in.
x	Horizontal coordinate.
y	Vertical coordinate.
rFont	Font.
pString	Null-terminated C like string to write on the display.
drawOption	Draw options,

See also

MonochromeView::DynamicView::DrawAt

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

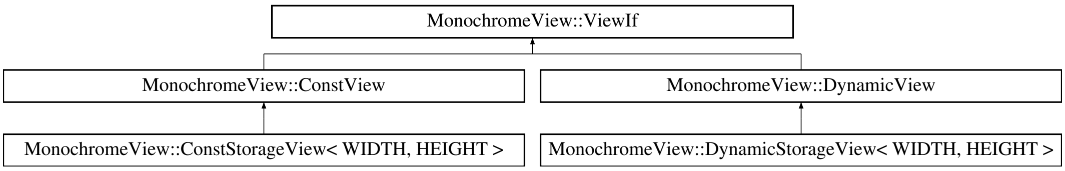
- DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeText.hpp

4.13 MonochromeView::ViewIf Class Reference

View interface.

```
#include <ViewIf.hpp>
```

Inheritance diagram for MonochromeView::ViewIf:


MonochromeView::ViewIf
MonochromeView::ConstView
MonochromeView::DynamicView
MonochromeView::ConstStorageView< WIDTH, HEIGHT >
MonochromeView::DynamicStorageView< WIDTH, HEIGHT >

Public Member Functions

- virtual size_t Width () const =0

  Width of an view in pixels.
- virtual size_t Height () const =0

  Height of an view in pixels.
- virtual const uint8_t ∗ **Data** () const =0

  Get constant data of an view.
- virtual bool GetPixelColor (const size_t x, const size_t y) const =0
- virtual bool IfViewChanged () const =0

  Check if the view has been changed since the last check.

Static Public Attributes

- static constexpr const size_t **PIXELS_PER_CELL** = 8U

4.13.1 Detailed Description

View interface.

4.13.2 Member Function Documentation

4.13.2.1 GetPixelColor()

```
virtual bool MonochromeView::ViewIf::GetPixelColor (
        const size_t x,
        const size_t y ) const  [pure virtual]
```

Get color of the pixel.

Parameters

x	Horizontal coordinate.
y	Vertical coordinate.

Returns

Color of the pixel. [0 - black, 1 - default color].

Implemented in MonochromeView::ConstView, and MonochromeView::DynamicView.

4.13.2.2 Height()

```
virtual size_t MonochromeView::ViewIf::Height ( ) const  [pure virtual]
```

Height of an view in pixels.

Implemented in MonochromeView::ConstView, and MonochromeView::DynamicView.

4.13.2.3 IfViewChanged()

```
virtual bool MonochromeView::ViewIf::IfViewChanged ( ) const  [pure virtual]
```

Check if the view has been changed since the last check.

Implemented in MonochromeView::ConstView, and MonochromeView::DynamicView.

4.13.2.4 Width()

```
virtual size_t MonochromeView::ViewIf::Width ( ) const  [pure virtual]
```

Width of an view in pixels.

Implemented in MonochromeView::ConstView, and MonochromeView::DynamicView.

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

- DisplayApp/App/MonochromeView/Inc/MonochromeView/ViewIf.hpp

5 File Documentation

5.1 DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayCommIf.hpp File Reference

```
#include <cstdint>
#include <cstddef>
```

Classes

- class DisplayComm::DisplayCommIf

  Interface for communication with display.

5.1.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.2 DisplayCommIf.hpp

Go to the documentation of this file.

```


#ifndef DISPLAY_COMM_DISPLAY_COMM_IF_HPP
#define DISPLAY_COMM_DISPLAY_COMM_IF_HPP

#include <cstdint>
#include <cstddef>

namespace DisplayComm
{

class DisplayCommIf
{
public:
    virtual ~DisplayCommIf()
    {
    }

    virtual bool WriteCmd(const uint8_t cmd) const = 0;

    virtual bool WriteData(const uint8_t* const pData, const size_t dataSize) const = 0;
};

}

#endif
```

5.3 DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayDataCmdIf.hpp File Reference

Classes

- class DisplayComm::DisplayDataCmdIf

  Interface for changing mode of the display.

5.3.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.4 DisplayDataCmdIf.hpp

Go to the documentation of this file.

```


#ifndef DISPLAY_COMM_DISPLAY_DATA_CMD_IF_HPP
#define DISPLAY_COMM_DISPLAY_DATA_CMD_IF_HPP

namespace DisplayComm
{

class DisplayDataCmdIf
{
public:
    virtual ~DisplayDataCmdIf()
    {
    }

    virtual void EnableCmdMode() const = 0;

    virtual void EnableDataMode() const = 0;
};

}

#endif
```

5.5 DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayResetIf.hpp File Reference

Classes

- class DisplayComm::DisplayResetIf

  Interface for the reseting display.

5.5.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.6 DisplayResetIf.hpp

Go to the documentation of this file.

```


#ifndef DISPLAY_COMM_DISPLAY_RESET_IF_HPP
#define DISPLAY_COMM_DISPLAY_RESET_IF_HPP

namespace DisplayComm
{

class DisplayResetIf
{
public:
    virtual ~DisplayResetIf()
    {
    }

    virtual void Reset() const = 0;
};

}

#endif
```

5.7 DisplayApp/App/MonochromeGraphicDisplay/Inc/MonochromeGraphicDisplay/↩ DisplayDriverIf.hpp File Reference

```
#include <cstdint>
#include <cstddef>
```

Classes

- class MonochromeGraphicDisplay::DisplayDriverIf

  Diver interface of graphical screens.

5.7.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.8 DisplayDriverIf.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMEGRAPHICDISPLAY_DISPLAYDRIVERIF_HPP
#define MONOCHROMEGRAPHICDISPLAY_DISPLAYDRIVERIF_HPP

#include <cstdint>
#include <cstddef>

namespace MonochromeView
{

class DynamicView;

}

namespace MonochromeGraphicDisplay
{

class DisplayDriverIf
{
public:
    virtual ~DisplayDriverIf()
    {
    }

    virtual bool TurnOnDisplay() = 0;

    virtual bool TurnOffDisplay() = 0;

    virtual bool InverseColor(bool inverse) = 0;

    virtual bool SetContrast(uint8_t value) = 0;

    virtual bool RefreshScreen() = 0;

    virtual MonochromeView::DynamicView& GetView() = 0;

    virtual size_t GetWidth() = 0;

    virtual size_t GetHeight() = 0;
};

}

#endif
```

5.9 DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont10x7.hpp File Reference

```
#include "MonochromeText/MonochromeFont.hpp"
```

Variables

- const MonochromeFont **MonochromeText::font10x7**

5.9.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.10 MonochromeFont10x7.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMETEXT_MONOCHROMEFONT10X7_HPP
#define MONOCHROMETEXT_MONOCHROMEFONT10X7_HPP

#include "MonochromeText/MonochromeFont.hpp"

namespace MonochromeText
{

extern const MonochromeFont font10x7;

}

#endif
```

5.11 DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont18x11.hpp File Reference

```
#include "MonochromeText/MonochromeFont.hpp"
```

Variables

- const MonochromeFont **MonochromeText::font18x11**

5.11.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.12 MonochromeFont18x11.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMETEXT_MONOCHROMEFONT18X11_HPP
#define MONOCHROMETEXT_MONOCHROMEFONT18X11_HPP

#include "MonochromeText/MonochromeFont.hpp"

namespace MonochromeText
{

extern const MonochromeFont font18x11;

}

#endif
```

5.13 DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont26x16.hpp File Reference

```
#include "MonochromeText/MonochromeFont.hpp"
```

Variables

- const MonochromeFont **MonochromeText::font26x16**

5.13.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.14 MonochromeFont26x16.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMETEXT_MONOCHROMEFONT26X16_HPP
#define MONOCHROMETEXT_MONOCHROMEFONT26X16_HPP

#include "MonochromeText/MonochromeFont.hpp"

namespace MonochromeText
{

extern const MonochromeFont font26x16;

}

#endif
```

5.15 DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont8x6.hpp File Reference

```
#include "MonochromeText/MonochromeFont.hpp"
```

Variables

- const MonochromeFont **MonochromeText::font8x6**

5.15.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.16 MonochromeFont8x6.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMETEXT_MONOCHROMEFONT8X6_HPP
#define MONOCHROMETEXT_MONOCHROMEFONT8X6_HPP

#include "MonochromeText/MonochromeFont.hpp"

namespace MonochromeText
{

extern const MonochromeFont font8x6;

}

#endif
```

5.17 DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeFont.hpp File Reference

```
#include <cstddef>
#include <cstdint>
#include "MonochromeView/ConstView.hpp"
```

Classes

- class MonochromeText::MonochromeFont

  Monochrome font.

5.17.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.18 MonochromeFont.hpp

Go to the documentation of this file.

```
#ifndef MONOCHROMETEXT_MONOCHROMEFONT_HPP
#define MONOCHROMETEXT_MONOCHROMEFONT_HPP

#include <cstddef>
#include <cstdint>

#include "MonochromeView/ConstView.hpp"

namespace MonochromeText
{

class MonochromeFont
{
public:
    MonochromeFont(const uint8_t* const pCharsBytes,
                   const size_t width,
                   const size_t height,
                   const size_t firstCharAsciiOffset,
                   const size_t charsNum);

    uint8_t GetWidth() const;

    uint8_t GetHeight() const;

    MonochromeView::ConstView GetCharView(const char character) const;

private:
    MonochromeFont(const MonochromeFont&) = delete;
    void operator=(const MonochromeFont&) = delete;

    const size_t m_Width;
    const size_t m_Height;
    const size_t m_FirstCharAsciiOffset;
    const size_t m_CharsNum;
    const size_t m_SingleCharBytesNum;
    const uint8_t* const m_pCharsBytes;

};

}

#endif
```

5.19 DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeText.hpp File Reference

```
#include <cstdint>
#include "MonochromeText/MonochromeFont.hpp"
#include "MonochromeView/DynamicView.hpp"
```

Classes

- class MonochromeText::MonochromeText

  Monochrome text.

5.19.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.20 MonochromeText.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMETEXT_MONOCHROMETEXT_HPP
#define MONOCHROMETEXT_MONOCHROMETEXT_HPP

#include <cstdint>

#include "MonochromeText/MonochromeFont.hpp"
#include "MonochromeView/DynamicView.hpp"

namespace MonochromeText
{

class MonochromeText
{
public:
    static void WriteChar(MonochromeView::DynamicView& rView,
                          const int32_t x, const int32_t y,
                          const MonochromeFont& rFont,
                          const char character,
                          const uint8_t drawOption = MonochromeView::DRAW_OPT_NONE);

    static void WriteString(MonochromeView::DynamicView& rView,
                            const int32_t x, const int32_t y,
                            const MonochromeFont& rFont,
                            const char* const pString,
                            const uint8_t drawOption = MonochromeView::DRAW_OPT_NONE);

private:
    MonochromeText(const MonochromeText&) = delete;
    void operator=(const MonochromeText&) = delete;
};

}

#endif
```

5.21 DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstStorageView.hpp File Reference

```
#include <cstdint>
#include <cstddef>
#include "MonochromeView/ConstView.hpp"
```

Classes

- class MonochromeView::ConstStorageView< WIDTH, HEIGHT >

5.21.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.22 ConstStorageView.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMEVIEW_CONSTSTORAGEVIEW_HPP
#define MONOCHROMEVIEW_CONSTSTORAGEVIEW_HPP

#include <cstdint>
#include <cstddef>

#include "MonochromeView/ConstView.hpp"

namespace MonochromeView
{

template<size_t WIDTH, size_t HEIGHT>
class ConstStorageView :
    public ConstView
{
public:
    template<typename... ViewBytesTypes>
    ConstStorageView(ViewBytesTypes... viewBytes) :
        ConstView(m_Buffer, WIDTH, HEIGHT),
        m_Buffer{static_cast<uint8_t>(viewBytes)...}
    {
    }

private:
    const uint8_t m_Buffer[WIDTH * ((HEIGHT + ViewIf::PIXELS_PER_CELL - 1) / ViewIf::PIXELS_PER_CELL)];
};

}

#endif
```

5.23 DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstView.hpp File Reference

```
#include "ViewIf.hpp"
```

Classes

- class MonochromeView::ConstView

  Constant view.

5.23.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.24 ConstView.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMEVIEW_CONSTVIEW_HPP
#define MONOCHROMEVIEW_CONSTVIEW_HPP

#include "ViewIf.hpp"

namespace MonochromeView
{
```

```

class ConstView :
    public ViewIf
{
public:
    ConstView(const uint8_t* const pBuffer, const size_t width, const size_t height);

    size_t Width() const override;

    size_t Height() const override;

    bool GetPixelColor(const size_t x, const size_t y) const override;

    bool IfViewChanged() const;

private:
    const uint8_t* Data() const override;

    const uint8_t* const m_pBuffer;
    const size_t m_Width;
    const size_t m_Height;
};

}

#endif
```

5.25 DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicStorageView.hpp File Reference

```
#include <cstdint>
#include <cstddef>
#include "MonochromeView/DynamicView.hpp"
```

Classes

- class MonochromeView::DynamicStorageView< WIDTH, HEIGHT >

5.25.1 Detailed Description

Note

5.26 DynamicStorageView.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMEVIEW_DYNAMICSTORAGEVIEW_HPP
#define MONOCHROMEVIEW_DYNAMICSTORAGEVIEW_HPP

#include <cstdint>
#include <cstddef>

#include "MonochromeView/DynamicView.hpp"

namespace MonochromeView
{

template<size_t WIDTH, size_t HEIGHT>
class DynamicStorageView :
    public DynamicView
{
public:
    template<typename... ViewBytesTypes>
    DynamicStorageView(ViewBytesTypes... viewBytes) :
        DynamicView(m_Buffer, WIDTH, HEIGHT),
        m_Buffer{static_cast<uint8_t>(viewBytes)...}
    {
    }

private:
    DynamicStorageView(const DynamicStorageView&) = delete;
    void operator=(const DynamicStorageView&) = delete;

    uint8_t m_Buffer[WIDTH * ((HEIGHT + ViewIf::PIXELS_PER_CELL - 1) / ViewIf::PIXELS_PER_CELL)];
};

}

#endif
```

5.27 DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicView.hpp File Reference

```
#include "ViewIf.hpp"
```

Classes

- class MonochromeView::DynamicView

  Dynamic view.

Variables

- constexpr const uint8_t **MonochromeView::DRAW_OPT_NONE** = 0x00U

  No additional draw options.
- constexpr const uint8_t **MonochromeView::DRAW_OPT_TRANSPOSE** = 0x01U

  Draw transposition of a view.
- constexpr const uint8_t **MonochromeView::DRAW_OPT_X_MIRROR** = 0x02U

  Mirror a view horizontally.
- constexpr const uint8_t **MonochromeView::DRAW_OPT_Y_MIRROR** = 0x04U

  Mirror a view vertically.
- constexpr const uint8_t **MonochromeView::DRAW_OPT_NEGATIVE_COLORS** = 0x08U

  Negative colors of a view.

5.27.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.28 DynamicView.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMEVIEW_DYNAMICVIEW_HPP
#define MONOCHROMEVIEW_DYNAMICVIEW_HPP

#include "ViewIf.hpp"

namespace MonochromeView
{

constexpr const uint8_t DRAW_OPT_NONE = 0x00U;

constexpr const uint8_t DRAW_OPT_TRANSPOSE = 0x01U;

constexpr const uint8_t DRAW_OPT_X_MIRROR = 0x02U;

constexpr const uint8_t DRAW_OPT_Y_MIRROR = 0x04U;

constexpr const uint8_t DRAW_OPT_NEGATIVE_COLORS = 0x08U;

class DynamicView :
    public ViewIf
{
public:
    DynamicView(uint8_t* const pBuffer, const size_t width, const size_t height);

    size_t Width() const override;

    size_t Height() const override;

    bool GetPixelColor(const size_t x, const size_t y) const override;

    bool IfViewChanged() const;

    void DrawAt(const int32_t x,
                const int32_t y,
                const ViewIf& rAnotherView,
                const uint8_t drawOption = DRAW_OPT_NONE);

    void DrawLine(const int32_t x1, const int32_t y1,
                  const int32_t x2, const int32_t y2,
                  const bool color);

    void SetPixelColor(const size_t x, const size_t y, const bool color);

    void Fill(const bool color);

private:
    DynamicView(const DynamicView&)  = delete;
    void operator=(const DynamicView&) = delete;

    const uint8_t* Data() const override;

    uint8_t* const m_pBuffer;
    const size_t m_Width;
    const size_t m_Height;
    mutable bool m_IfViewChanged;
};

}

#endif
```

5.29 DisplayApp/App/MonochromeView/Inc/MonochromeView/ViewIf.hpp File Reference

```
#include <cstddef>
#include <cstdint>
```

Classes

- class MonochromeView::ViewIf

  View interface.

5.29.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.30 ViewIf.hpp

Go to the documentation of this file.

```


#ifndef MONOCHROMEVIEW_VIEWIF_HPP
#define MONOCHROMEVIEW_VIEWIF_HPP

#include <cstddef>
#include <cstdint>

namespace MonochromeView
{

class ViewIf
{
public:
    static constexpr const size_t PIXELS_PER_CELL = 8U;

    virtual ~ViewIf()
    {
    }

    virtual size_t Width() const = 0;

    virtual size_t Height() const = 0;

    virtual const uint8_t* Data() const = 0;

    virtual bool GetPixelColor(const size_t x, const size_t y) const = 0;

    virtual bool IfViewChanged() const = 0;
};

}

#endif
```

5.31 DisplayApp/App/DisplayComm/Inc/DisplayComm/Factory.hpp File Reference

```
#include <cstdint>
#include "spi.h"
#include "gpio.h"
```

Classes

- class DisplayComm::Factory

  Factory for display communication layer.

5.31.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.32 Factory.hpp

Go to the documentation of this file.

```

#ifndef DISPLAY_COMM_FACTORY_HPP
#define DISPLAY_COMM_FACTORY_HPP

#include <cstdint>

#include "spi.h"
#include "gpio.h"

namespace DisplayComm
{

class DisplayCommIf;
class DisplayResetIf;
class DisplayDataCmdIf;

class Factory
{
public:
    static DisplayResetIf* CreateDisplayReset(GPIO_TypeDef* const pResetPort,
                                              const uint32_t resetPin);

    static DisplayDataCmdIf* CreateDataCmd(GPIO_TypeDef * const pDcPort,
                                           const uint32_t dcPin);

    static DisplayCommIf* CreateDisplayCommSpi(SPI_TypeDef* const pSpi,
                                               GPIO_TypeDef* const pCsPort,
                                               const uint32_t csPin,
                                               const DisplayDataCmdIf* const pDataCmdIf);

private:
    Factory(const Factory&) = delete;
    void operator=(const Factory&) = delete;
};

}

#endif
```

5.33 Factory.hpp

```

#ifndef SH1106_FACTORY_HPP
#define SH1106_FACTORY_HPP

#include <cstddef>

namespace MonochromeGraphicDisplay
{

class DisplayDriverIf;

}

namespace DisplayComm
{

class DisplayCommIf;
class DisplayResetIf;

}

namespace Sh1106
{

class Factory
{
public:
    static MonochromeGraphicDisplay::DisplayDriverIf* Create128x32Driver(DisplayComm::DisplayCommIf*
    const pDisplayCommIf,
                                                                         DisplayComm::DisplayResetIf*
    const pDisplayResetIf,
                                                                         const bool mirrorVertically =
    false,
                                                                         const bool mirrorHorizontally =
    false,
                                                                         const size_t columnOffset = 0U);

    static MonochromeGraphicDisplay::DisplayDriverIf* Create128x64Driver(DisplayComm::DisplayCommIf* const pDisplayCommIf,
                                                                         DisplayComm::DisplayResetIf* const pDisplayResetIf,
                                                                         const bool mirrorVertically = false,
                                                                         const bool mirrorHorizontally = false,
                                                                         const size_t columnOffset = 0U);

    static MonochromeGraphicDisplay::DisplayDriverIf* Create128x128Driver(DisplayComm::DisplayCommIf* const pDisplayCommIf,
                                                                          DisplayComm::DisplayResetIf* const pDisplayResetIf,
                                                                          const bool mirrorVertically = false,
                                                                          const bool mirrorHorizontally = false,
                                                                          const size_t columnOffset = 0U);

private:
    Factory(const Factory&) = delete;
    void operator=(const Factory&) = delete;
};

}

#endif
```

Index