



POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI
KIERUNEK TELEINFORMATYKA

Programowanie mikrokontrolerów ARM w języku C/C++

Biblioteka do obsługi wyświetlacza OLED ze sterownikiem SH1106

Autor: Dawid Jaraczewski i Dominik Kała

Kierujący pracą: dr inż. Bernard Wyrwoł

Gliwice, styczeń 2022

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	2
2.1 Class List	2
3 File Index	2
3.1 File List	2
4 Class Documentation	3
4.1 MonochromeView::ConstStorageView< WIDTH, HEIGHT > Class Template Reference	3
4.1.1 Detailed Description	5
4.1.2 Constructor & Destructor Documentation	6
4.2 MonochromeView::ConstView Class Reference	6
4.2.1 Detailed Description	8
4.2.2 Constructor & Destructor Documentation	8
4.2.3 Member Function Documentation	10
4.3 DisplayComm::DisplayCommIf Class Reference	11
4.3.1 Detailed Description	11
4.3.2 Member Function Documentation	11
4.4 DisplayComm::DisplayDataCmdIf Class Reference	12
4.4.1 Detailed Description	13
4.5 MonochromeGraphicDisplay::DisplayDriverIf Class Reference	13
4.5.1 Detailed Description	14
4.5.2 Member Function Documentation	14
4.6 DisplayComm::DisplayResetIf Class Reference	16
4.6.1 Detailed Description	16
4.7 MonochromeView::DynamicStorageView< WIDTH, HEIGHT > Class Template Reference	17
4.7.1 Detailed Description	18
4.7.2 Constructor & Destructor Documentation	19
4.8 MonochromeView::DynamicView Class Reference	19
4.8.1 Detailed Description	21
4.8.2 Constructor & Destructor Documentation	22
4.8.3 Member Function Documentation	22
4.9 DisplayComm::Factory Class Reference	24
4.9.1 Detailed Description	25
4.9.2 Member Function Documentation	25
4.10 Sh1106::Factory Class Reference	26
4.10.1 Detailed Description	27
4.10.2 Member Function Documentation	27
4.11 MonochromeText::MonochromeFont Class Reference	29
4.11.1 Detailed Description	29
4.11.2 Constructor & Destructor Documentation	29

4.11.3 Member Function Documentation	30
4.12 MonochromeText::MonochromeText Class Reference	31
4.12.1 Detailed Description	31
4.12.2 Member Function Documentation	31
4.13 MonochromeView::ViewIf Class Reference	32
4.13.1 Detailed Description	34
4.13.2 Member Function Documentation	34
5 File Documentation	35
5.1 DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayCommIf.hpp File Reference	35
5.1.1 Detailed Description	36
5.2 DisplayCommIf.hpp	36
5.3 DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayDataCmdIf.hpp File Reference	36
5.3.1 Detailed Description	36
5.4 DisplayDataCmdIf.hpp	36
5.5 DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayResetIf.hpp File Reference	37
5.5.1 Detailed Description	37
5.6 DisplayResetIf.hpp	37
5.7 DisplayApp/App/MonochromeGraphicDisplay/Inc/MonochromeGraphicDisplay/DisplayDriverIf.hpp File Reference	37
5.7.1 Detailed Description	38
5.8 DisplayDriverIf.hpp	38
5.9 DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont10x7.hpp File Reference	39
5.9.1 Detailed Description	39
5.10 MonochromeFont10x7.hpp	40
5.11 DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont18x11.hpp File Reference	40
5.11.1 Detailed Description	41
5.12 MonochromeFont18x11.hpp	41
5.13 DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont26x16.hpp File Reference	41
5.13.1 Detailed Description	42
5.14 MonochromeFont26x16.hpp	42
5.15 DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont8x6.hpp File Reference	43
5.15.1 Detailed Description	43
5.16 MonochromeFont8x6.hpp	43
5.17 DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeFont.hpp File Reference	44
5.17.1 Detailed Description	45
5.18 MonochromeFont.hpp	45
5.19 DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeText.hpp File Reference	45
5.19.1 Detailed Description	46
5.20 MonochromeText.hpp	46

5.21 DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstStorageView.hpp File Reference . .	47
5.21.1 Detailed Description	47
5.22 ConstStorageView.hpp	48
5.23 DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstView.hpp File Reference	48
5.23.1 Detailed Description	49
5.24 ConstView.hpp	49
5.25 DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicStorageView.hpp File Reference	50
5.25.1 Detailed Description	50
5.26 DynamicStorageView.hpp	50
5.27 DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicView.hpp File Reference	51
5.27.1 Detailed Description	52
5.28 DynamicView.hpp	52
5.29 DisplayApp/App/MonochromeView/Inc/MonochromeView/ViewIf.hpp File Reference	53
5.29.1 Detailed Description	54
5.30 ViewIf.hpp	54
5.31 DisplayApp/App/DisplayComm/Inc/DisplayComm/Factory.hpp File Reference	55
5.31.1 Detailed Description	55
5.32 Factory.hpp	56
5.33 Factory.hpp	56
Index	59

Hierarchical Index

1.1 Class Hierarchy

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

DisplayComm::DisplayCommIf	11
DisplayComm::DisplayDataCmdIf	12
MonochromeGraphicDisplay::DisplayDriverIf	13
DisplayComm::DisplayResetIf	16
DisplayComm::Factory	24
Sh1106::Factory	26
MonochromeText::MonochromeFont	29
MonochromeText::MonochromeText	31
MonochromeView::ViewIf	32
MonochromeView::ConstView	6
MonochromeView::ConstStorageView< WIDTH, HEIGHT >	3

MonochromeView::DynamicView	19
MonochromeView::DynamicStorageView< WIDTH, HEIGHT >	17

2 Class Index

2.1 Class List

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

MonochromeView::ConstStorageView< WIDTH, HEIGHT >	3
MonochromeView::ConstView Constant view	6
DisplayComm::DisplayCommIf Interface for communication with display	11
DisplayComm::DisplayDataCmdIf Interface for changing mode of the display	12
MonochromeGraphicDisplay::DisplayDriverIf Diver interface of graphical screens	13
DisplayComm::DisplayResetIf Interface for the resetting display	16
MonochromeView::DynamicStorageView< WIDTH, HEIGHT >	17
MonochromeView::DynamicView Dynamic view	19
DisplayComm::Factory Factory for display communication layer	24
Sh1106::Factory Factory of SH1106 dirvers	26
MonochromeText::MonochromeFont Monochrome font	29
MonochromeText::MonochromeText Monochrome text	31
MonochromeView::ViewIf View interface	32

3 File Index

3.1 File List

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

DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayCommIf.hpp	35
---	-----------

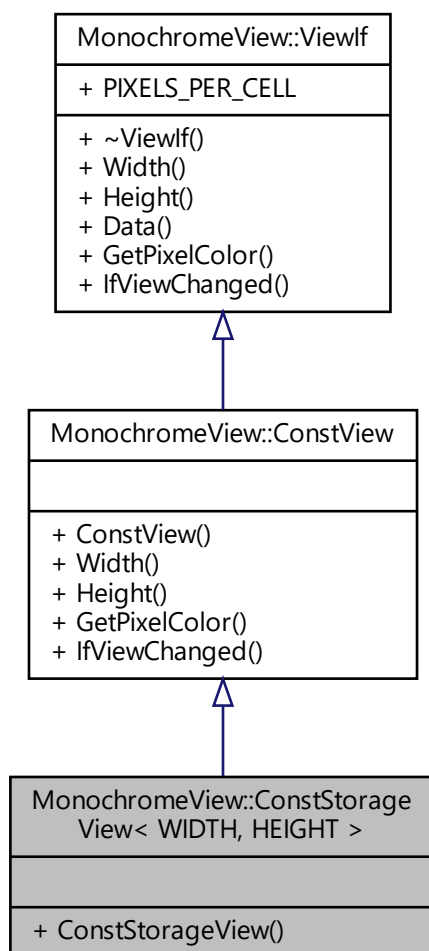
DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayDataCmdIf.hpp	36
DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayResetIf.hpp	37
DisplayApp/App/DisplayComm/Inc/DisplayComm/Factory.hpp	55
DisplayApp/App/MonochromeGraphicDisplay/Inc/MonochromeGraphicDisplay/DisplayDriverIf.hpp	37
DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeFont.hpp	44
DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeText.hpp	45
DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont10x7.hpp	39
DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont18x11.hpp	40
DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont26x16.hpp	41
DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont8x6.hpp	43
DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstStorageView.hpp	47
DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstView.hpp	48
DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicStorageView.hpp	50
DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicView.hpp	51
DisplayApp/App/MonochromeView/Inc/MonochromeView/ViewIf.hpp	53
DisplayApp/App/Sh1106/Inc/Sh1106/Factory.hpp	56

4 Class Documentation

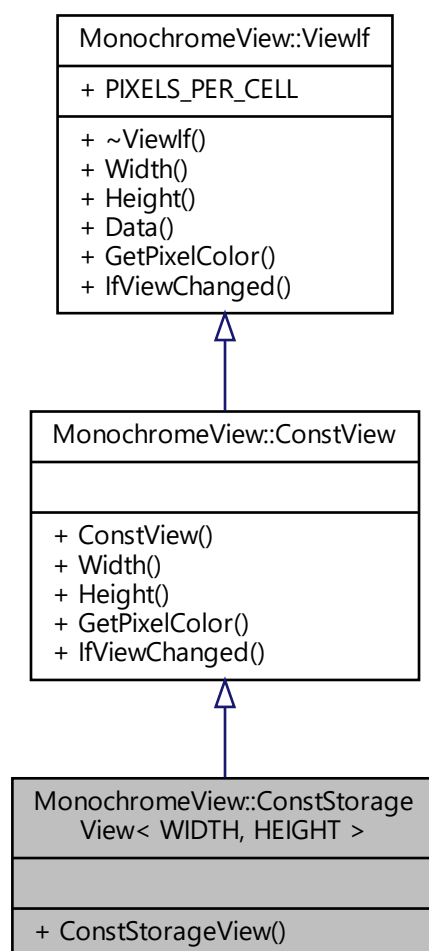
4.1 MonochromeView::ConstStorageView< WIDTH, HEIGHT > Class Template Reference

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

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



Collaboration diagram for 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

<i>WIDTH</i>	Width of a view.
<i>HEIGHT</i>	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

<i>ViewBytesTypes</i>	Types of view bytes.
-----------------------	----------------------

Parameters

<i>viewBytes</i>	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:

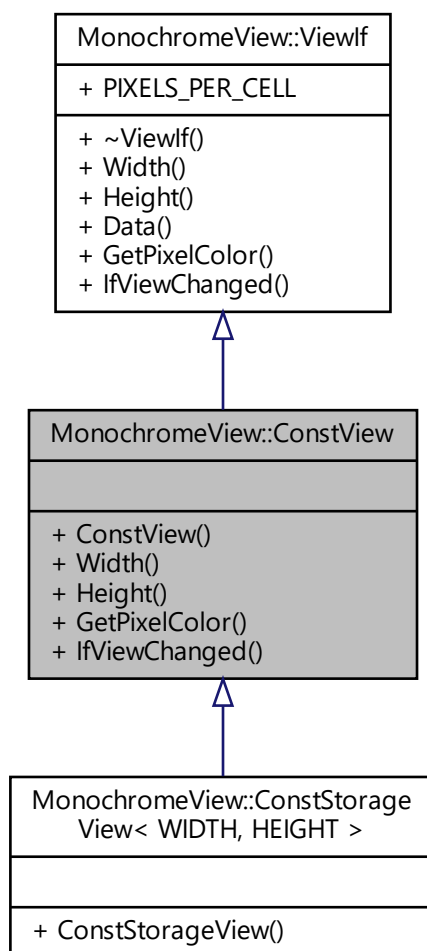
- DisplayApp/App/MonochromeView/Inc/MonochromeView/[ConstStorageView.hpp](#)

4.2 MonochromeView::ConstView Class Reference

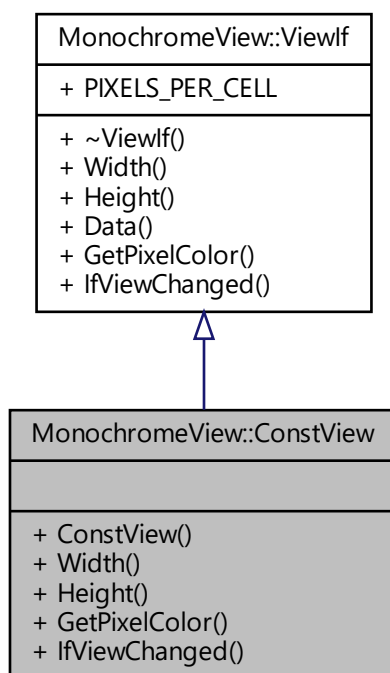
Constant view.

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

Inheritance diagram for MonochromeView::ConstView:



Collaboration diagram for MonochromeView::ConstView:



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

<i>pBuffer</i>	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.
<i>width</i>	Width of a view.
<i>height</i>	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

<i>x</i>	Horizontal coordinate.
<i>y</i>	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:

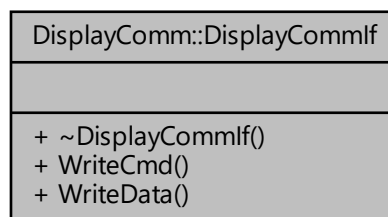
- DisplayApp/App/MonochromeView/Inc/MonochromeView/[ConstView.hpp](#)

4.3 DisplayComm::DisplayCommIf Class Reference

Interface for communication with display.

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

Collaboration diagram for DisplayComm::DisplayCommIf:



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

<i>cmd</i>	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

<i>pData</i>	Pointer to the data buffer.
<i>dataSize</i>	Data size in bytes.

Returns

Write result.

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

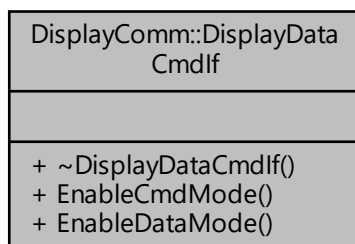
- DisplayApp/App/DisplayComm/Inc/DisplayComm/[DisplayCommIf.hpp](#)

4.4 DisplayComm::DisplayDataCmdIf Class Reference

Interface for changing mode of the display.

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

Collaboration diagram for DisplayComm::DisplayDataCmdIf:



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:

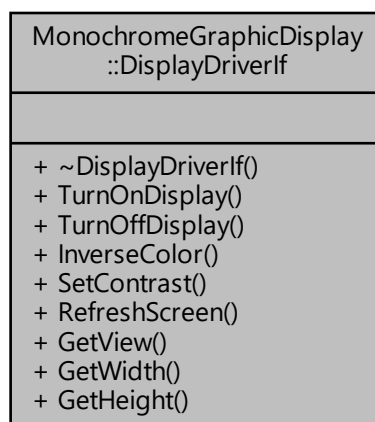
- DisplayApp/App/DisplayComm/Inc/DisplayComm/[DisplayDataCmdIf.hpp](#)

4.5 MonochromeGraphicDisplay::DisplayDriverIf Class Reference

Diver interface of graphical screens.

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

Collaboration diagram for MonochromeGraphicDisplay::DisplayDriverIf:

**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

<i>color</i>	[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

<i>value</i>	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:

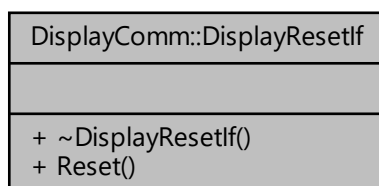
- DisplayApp/App/MonochromeGraphicDisplay/Inc/MonochromeGraphicDisplay/[DisplayDriverIf.hpp](#)

4.6 DisplayComm::DisplayResetIf Class Reference

Interface for the resetting display.

```
#include <DisplayResetIf.hpp>
```

Collaboration diagram for DisplayComm::DisplayResetIf:



Public Member Functions

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

4.6.1 Detailed Description

Interface for the resetting display.

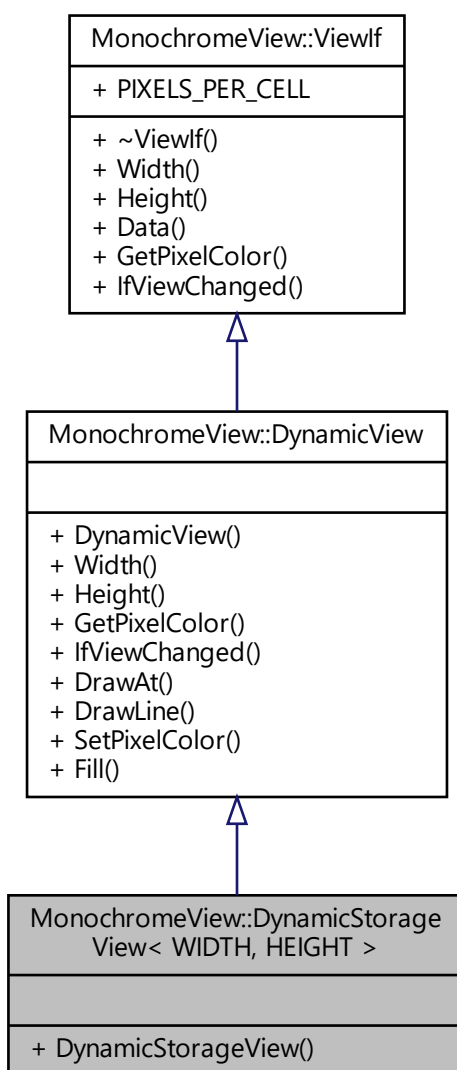
The documentation for this class was generated from the following file:

- DisplayApp/App/DisplayComm/Inc/DisplayComm/[DisplayResetIf.hpp](#)

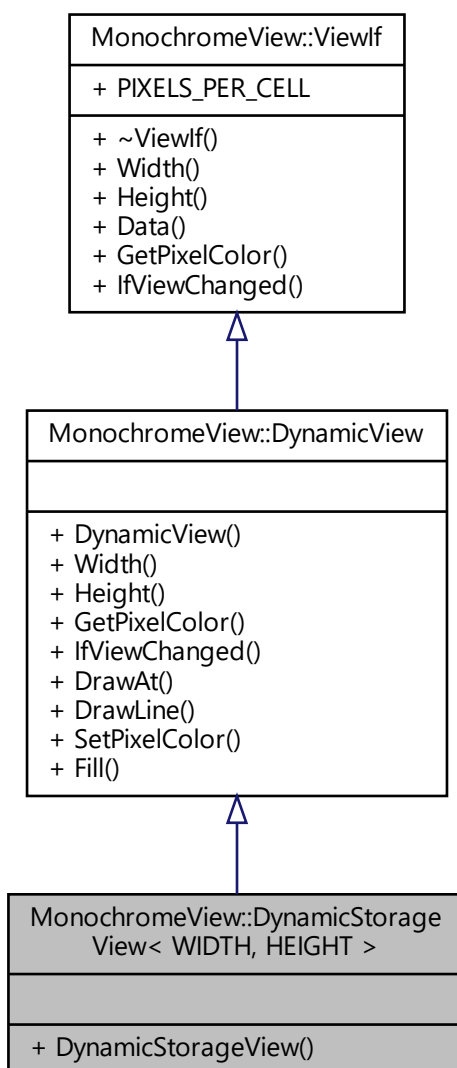
4.7 MonochromeView::DynamicStorageView< WIDTH, HEIGHT > Class Template Reference

```
#include <DynamicStorageView.hpp>
```

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



Collaboration diagram for 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

<i>WIDTH</i>	Width of a view.
<i>HEIGHT</i>	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

<i>ViewBytesTypes</i>	Types of view bytes.
-----------------------	----------------------

Parameters

<i>viewBytes</i>	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:

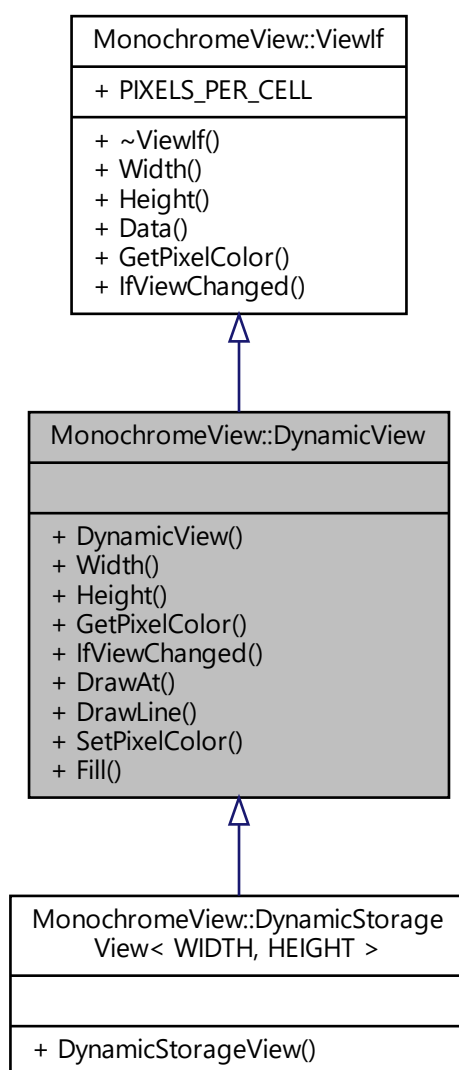
- DisplayApp/App/MonochromeView/Inc/MonochromeView/[DynamicStorageView.hpp](#)

4.8 MonochromeView::DynamicView Class Reference

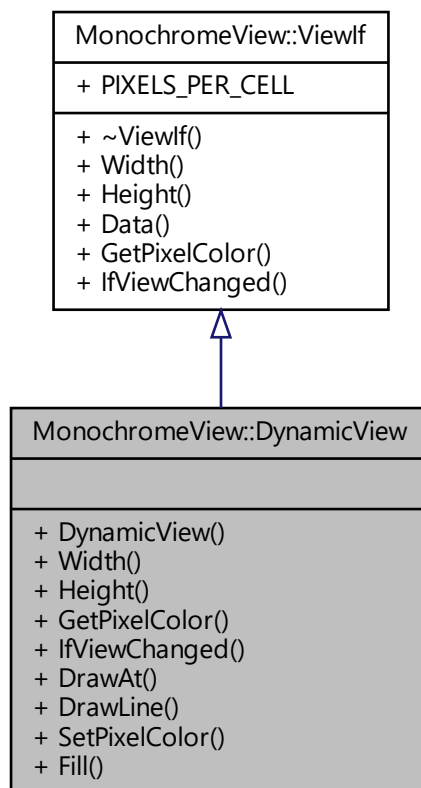
Dynamic view.

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

Inheritance diagram for MonochromeView::DynamicView:



Collaboration diagram for MonochromeView::DynamicView:



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

<i>pBuffer</i>	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.
<i>width</i>	Width of a view.
<i>height</i>	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

<i>x</i>	Horizontal coordinate.
<i>y</i>	Vertical coordinate.
<i>rAnotherView</i>	View to draw.
<i>drawOption</i>	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

<i>x1</i>	First point position in pixles, starting from left edge.
<i>y1</i>	First point position in pixles, starting from top edge.
<i>x2</i>	Second point position in pixles, starting from left edge.
<i>y2</i>	Second point position in pixles, starting from top edge.
<i>color</i>	[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

<i>color</i>	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

<i>x</i>	Horizontal coordinate.
<i>y</i>	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

<i>x</i>	Horizontal coordinate.
<i>y</i>	Vertical coordinate.
<i>color</i>	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:

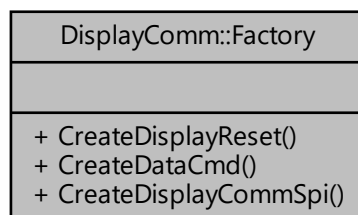
- [DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicView.hpp](#)

4.9 DisplayComm::Factory Class Reference

[Factory](#) for display communication layer.

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

Collaboration diagram for DisplayComm::Factory:



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

<i>pDcPort</i>	Port of D/C pin.
<i>dcPin</i>	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::CreateDisplayCommSpi (SPI_TypeDef *const *pSpi*, GPIO_TypeDef *const *pCsPort*, const uint32_t *csPin*, const [DisplayDataCmdIf](#) *const *pDataCmdIf*) [static]

Create a display comm object.

Parameters

<i>pSpi</i>	Pointer to the SPI interface.
<i>pCsPort</i>	Port of the chip select pin.
<i>csPin</i>	Chip select pin.
<i>pDataCmdIf</i>	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

<i>pResetPort</i>	Port of the reset pin.
<i>resetPin</i>	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:

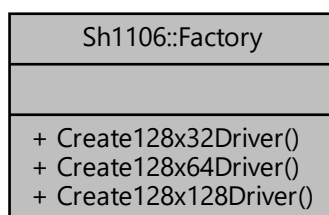
- DisplayApp/App/DisplayComm/Inc/DisplayComm/[Factory.hpp](#)

4.10 Sh1106::Factory Class Reference

[Factory](#) of SH1106 drivers.

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

Collaboration diagram for Sh1106::Factory:



Static Public Member Functions

- 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)

4.10.1 Detailed Description

[Factory](#) of SH1106 drivers.

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

<i>pDisplayCommIf</i>	Pointer to the display communication interface.
<i>pDisplayResetIf</i>	Pointer to the display reset interface.
<i>mirrorVertically</i>	Set to true to mirror vertically.
<i>mirrorHorizontally</i>	Set to false to mirror horizontally.
<i>columnOffset</i>	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

<i>pDisplayCommIf</i>	Pointer to the display communication interface.
<i>pDisplayResetIf</i>	Pointer to the display reset interface.
<i>mirrorVertically</i>	Set to true to mirror vertically.
<i>mirrorHorizontally</i>	Set to false to mirror horizontally.
<i>columnOffset</i>	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

<i>pDisplayCommIf</i>	Pointer to the display communication interface.
<i>pDisplayResetIf</i>	Pointer to the display reset interface.
<i>mirrorVertically</i>	Set to true to mirror vertically.
<i>mirrorHorizontally</i>	Set to false to mirror horizontally.
<i>columnOffset</i>	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:

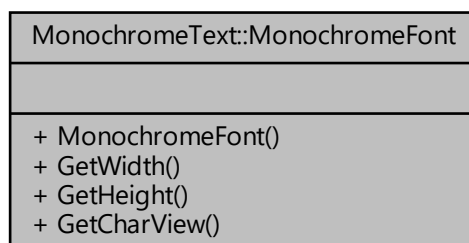
- DisplayApp/App/Sh1106/Inc/Sh1106/Factory.hpp

4.11 MonochromeText::MonochromeFont Class Reference

Monochrome font.

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

Collaboration diagram for MonochromeText::MonochromeFont:

**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

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

<i>character</i>	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:

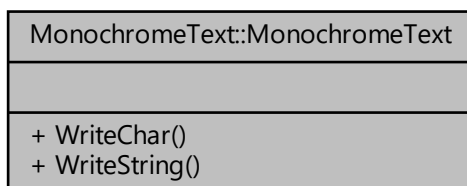
- DisplayApp/App/MonochromeText/Inc/MonochromeText/[MonochromeFont.hpp](#)

4.12 MonochromeText::MonochromeText Class Reference

Monochrome text.

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

Collaboration diagram for MonochromeText::MonochromeText:



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

<i>rView</i>	View to write character in.
<i>x</i>	Horizontal coordinate.
<i>y</i>	Vertical coordinate.
<i>rFont</i>	Font.
<i>character</i>	Character to write.
<i>drawOption</i>	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

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

See also

[MonochromeView::DynamicView::DrawAt](#)

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

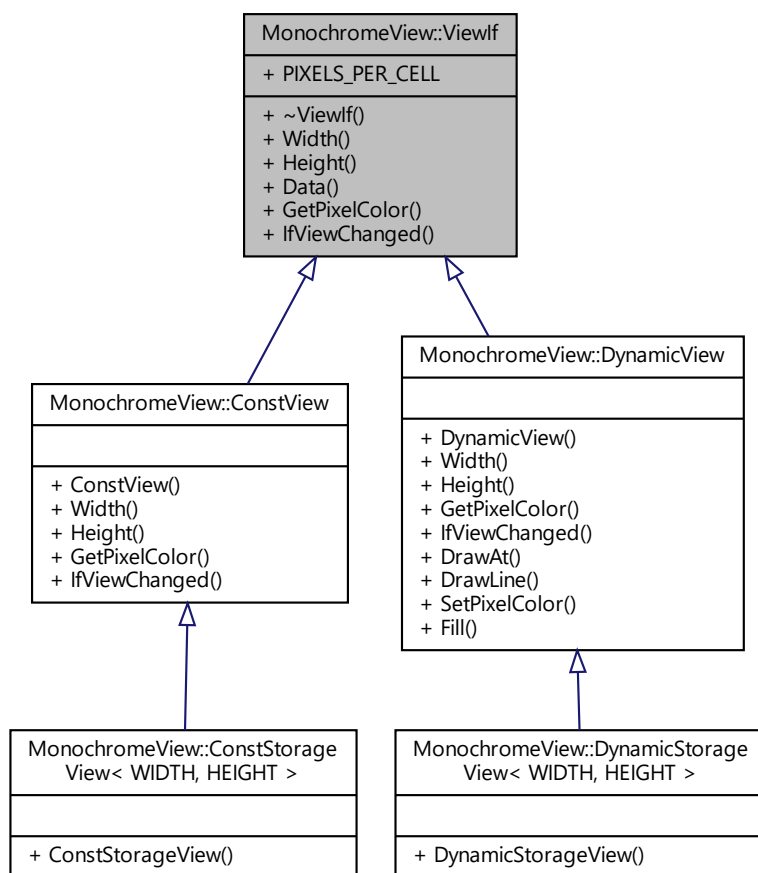
- DisplayApp/App/MonochromeText/Inc/MonochromeText/[MonochromeText.hpp](#)

4.13 MonochromeView::ViewIf Class Reference

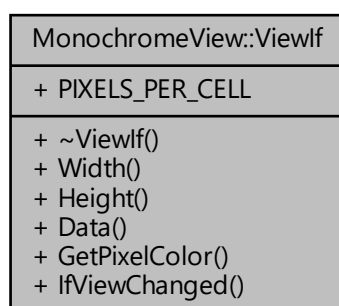
View interface.

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

Inheritance diagram for MonochromeView::ViewIf:



Collaboration diagram for MonochromeView::ViewIf:



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

<i>x</i>	Horizontal coordinate.
<i>y</i>	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:

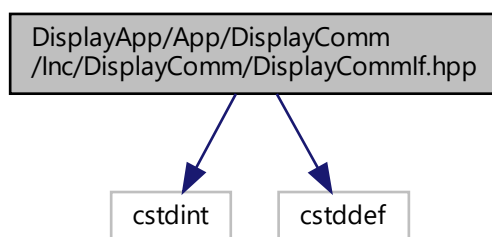
- [DisplayApp/App/MonochromeView/Inc/MonochromeView/ViewIf.hpp](#)

5 File Documentation

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

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

Include dependency graph for DisplayCommIf.hpp:



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.](#)

```
1
4
5 #ifndef DISPLAY_COMM_DISPLAY_COMM_IF_HPP
6 #define DISPLAY_COMM_DISPLAY_COMM_IF_HPP
7
8 #include <cstdint>
9 #include <cstddef>
10
11 namespace DisplayComm
12 {
13
14     class DisplayCommIf
15     {
16     public:
17         virtual ~DisplayCommIf()
18         {
19         }
20
21         virtual bool WriteCmd(const uint8_t cmd) const = 0;
22
23         virtual bool WriteData(const uint8_t* const pData, const size_t dataSize) const = 0;
24     };
25 }
26
27 #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.](#)

```
1
4
5 #ifndef DISPLAY_COMM_DISPLAY_DATA_CMD_IF_HPP
6 #define DISPLAY_COMM_DISPLAY_DATA_CMD_IF_HPP
7
8 namespace DisplayComm
9 {
10
11     class DisplayDataCmdIf
12     {
13     public:
```

```

15     virtual ~DisplayDataCmdIf()
16     {
17     }
18
19     virtual void EnableCmdMode() const = 0;
20
21     virtual void EnableDataMode() const = 0;
22 };
23
24 }
25
26 }
27
28 #endif

```

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

Classes

- class [DisplayComm::DisplayResetIf](#)
Interface for the resetting display.

5.5.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.6 DisplayResetIf.hpp

[Go to the documentation of this file.](#)

```

1
4
5 #ifndef DISPLAY_COMM_DISPLAY_RESET_IF_HPP
6 #define DISPLAY_COMM_DISPLAY_RESET_IF_HPP
7
8 namespace DisplayComm
9 {
10
11     class DisplayResetIf
12     {
13     public:
14         virtual ~DisplayResetIf()
15         {
16         }
17
18         virtual void Reset() const = 0;
19     };
20
21 }
22
23 }
24
25 #endif

```

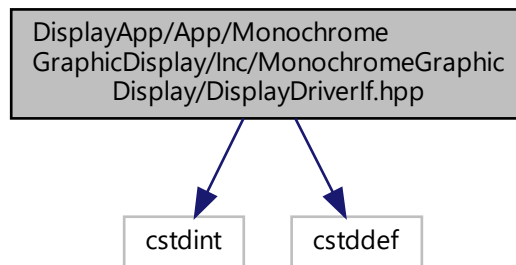
5.7 DisplayApp/App/MonochromeGraphicDisplay/Inc/MonochromeGraphicDisplay/DisplayDriverIf.hpp File Reference

```

#include <cstdint>
#include <cstddef>

```


Include dependency graph for DisplayDriverIf.hpp:



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.](#)

```

1
4
5 #ifndef MONOCHROMEGRAPHICDISPLAY_DISPLAYDRIVERIF_HPP
6 #define MONOCHROMEGRAPHICDISPLAY_DISPLAYDRIVERIF_HPP
7
8 #include <cstdint>
9 #include <cstddef>
10
11 namespace MonochromeView
12 {
13
14 class DynamicView;
15
16 }
17
18 namespace MonochromeGraphicDisplay
19 {
20
22 class DisplayDriverIf
23 {
24 public:
25     virtual ~DisplayDriverIf()
26     {
27     }
28
32     virtual bool TurnOnDisplay() = 0;
33
37     virtual bool TurnOffDisplay() = 0;
38
44     virtual bool InverseColor(bool inverse) = 0;
45
51     virtual bool SetContrast(uint8_t value) = 0;

```

```

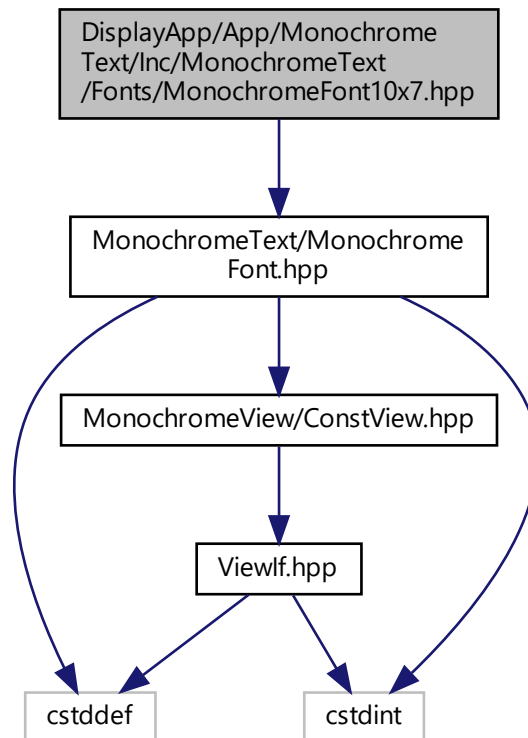
52
53     virtual bool RefreshScreen() = 0;
54
55     virtual MonochromeView::DynamicView& GetView() = 0;
56
57     virtual size_t GetWidth() = 0;
58
59     virtual size_t GetHeight() = 0;
60 };
61
62 #endif

```

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

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

Include dependency graph for MonochromeFont10x7.hpp:



Variables

- const MonochromeFont **MonochromeText::font10x7**

5.9.1 Detailed Description

Note

Copyright (c) 2021 ArmC++ - Kala, Jaraczewski

5.10 MonochromeFont10x7.hpp

[Go to the documentation of this file.](#)

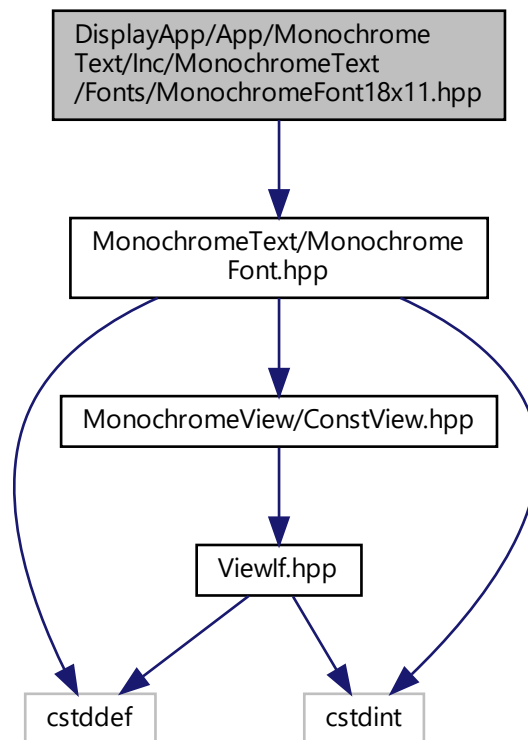
```

1
4
5 #ifndef MONOCHROMETEXT_MONOCHROMEFont10x7_HPP
6 #define MONOCHROMETEXT_MONOCHROMEFont10x7_HPP
7
8 #include "MonochromeText/MonochromeFont.hpp"
9
10 namespace MonochromeText
11 {
12
13 extern const MonochromeFont font10x7;
14
15 }
16
17 #endif

```

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

#include "MonochromeText/MonochromeFont.hpp"
 Include dependency graph for MonochromeFont18x11.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.](#)

```

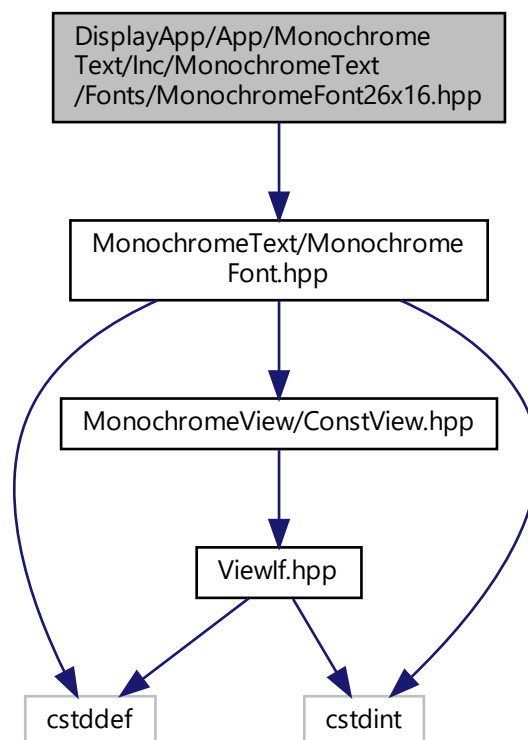
1
4
5 #ifndef MONOCHROMETEXT_MONOCHROMEFONT18X11_HPP
6 #define MONOCHROMETEXT_MONOCHROMEFONT18X11_HPP
7
8 #include "MonochromeText/MonochromeFont.hpp"
9
10 namespace MonochromeText
11 {
12
13 extern const MonochromeFont font18x11;
14
15 }
16
17 #endif

```

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

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

Include dependency graph for MonochromeFont26x16.hpp:



Variables

- const MonochromeFont **MonochromeText::font26x16**

5.13.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.14 MonochromeFont26x16.hpp

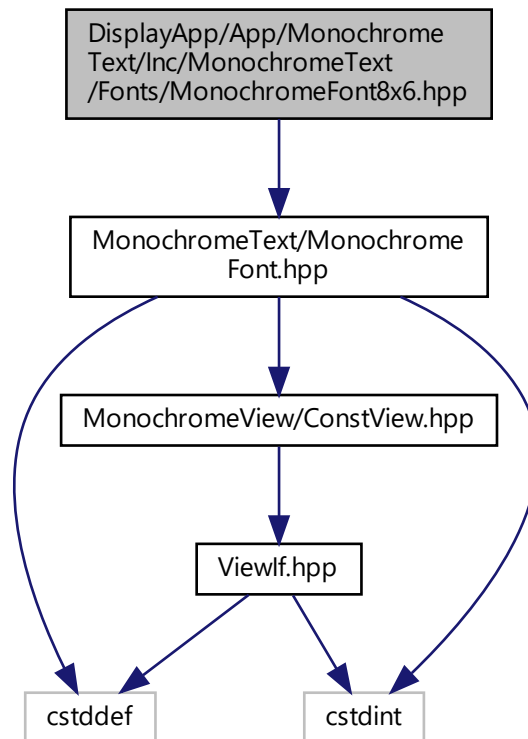
[Go to the documentation of this file.](#)

```
1
2
3
4
5 #ifndef MONOCHROMETEXT_MONOCHROMEFONT26X16_HPP
6 #define MONOCHROMETEXT_MONOCHROMEFONT26X16_HPP
7
8 #include "MonochromeText/MonochromeFont.hpp"
9
10 namespace MonochromeText
11 {
12
13 extern const MonochromeFont font26x16;
14
15 }
16
17 #endif
```

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

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

Include dependency graph for MonochromeFont8x6.hpp:



Variables

- const MonochromeFont **MonochromeText::font8x6**

5.15.1 Detailed Description

Note

Copyright (c) 2021 ArmCcpp - Kala, Jaraczewski

5.16 MonochromeFont8x6.hpp

[Go to the documentation of this file.](#)

```
1
4
5 #ifndef MONOCHROMETEXT_MONOCHROMEFONT8X6_HPP
6 #define MONOCHROMETEXT_MONOCHROMEFONT8X6_HPP
```

```

7
8 #include "MonochromeText/MonochromeFont.hpp"
9
10 namespace MonochromeText
11 {
12
13 extern const MonochromeFont font8x6;
14
15 }
16
17 #endif

```

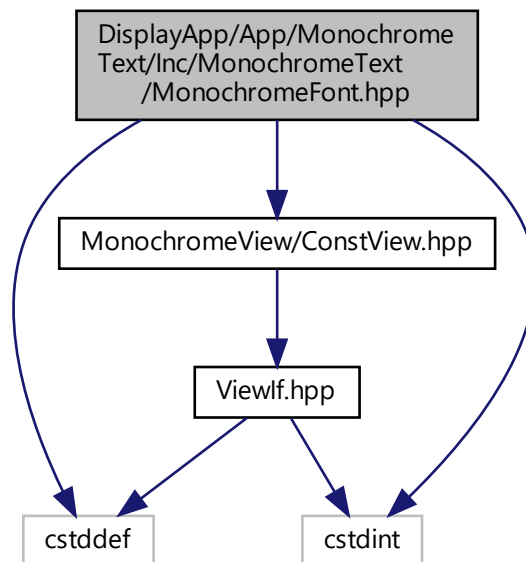
5.17 DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeFont.hpp File Reference

```

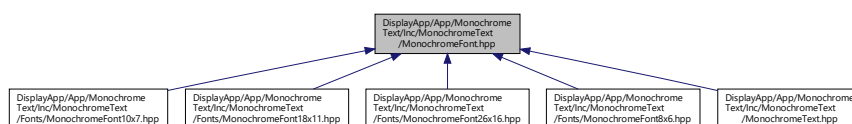
#include <cstdint>
#include <cstddef>
#include "MonochromeView/ConstView.hpp"

```

Include dependency graph for MonochromeFont.hpp:



This graph shows which files directly or indirectly include this file:



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.](#)

```

1
4
5 #ifndef MONOCHROMETEXT_MONOCHROMEFONT_HPP
6 #define MONOCHROMETEXT_MONOCHROMEFONT_HPP
7
8 #include <cstdint>
9 #include <cstdint>
10
11 #include "MonochromeView/ConstView.hpp"
12
13 namespace MonochromeText
14 {
15
16     class MonochromeFont
17     {
18     public:
19         MonochromeFont(const uint8_t* const pCharsBytes,
20                         const size_t width,
21                         const size_t height,
22                         const size_t firstCharAsciiOffset,
23                         const size_t charsNum);
24
25         uint8_t GetWidth() const;
26
27         uint8_t GetHeight() const;
28
29         MonochromeView::ConstView GetCharView(const char character) const;
30
31     private:
32         MonochromeFont(const MonochromeFont&) = delete;
33         void operator=(const MonochromeFont&) = delete;
34
35         const size_t m_Width;
36         const size_t m_Height;
37         const size_t m_FirstCharAsciiOffset;
38         const size_t m_CharsNum;
39         const size_t m_SingleCharBytesNum;
40         const uint8_t* const m_pCharsBytes;
41
42     };
43
44 }
45 #endif

```

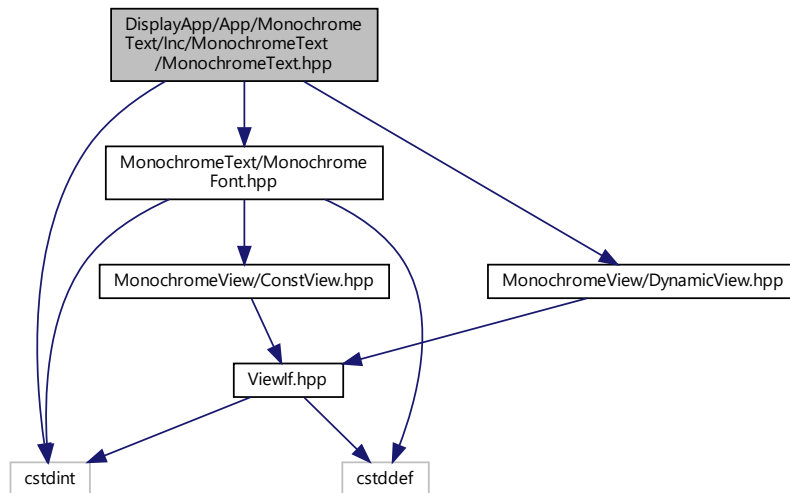
5.19 DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeText.hpp File Reference

```

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

```


Include dependency graph for MonochromeText.hpp:



Classes

- class [MonochromeText::MonochromeText](#)
Monochrome text.

5.19.1 Detailed Description

Note

Copyright (c) 2021 ArmC++ - Kala, Jaraczewski

5.20 MonochromeText.hpp

[Go to the documentation of this file.](#)

```

1
4
5 #ifndef MONOCHROMETEXT_MONOCHROMETEXT_HPP
6 #define MONOCHROMETEXT_MONOCHROMETEXT_HPP
7
8 #include <stdint>
9
10 #include "MonochromeText/MonochromeFont.hpp"
11 #include "MonochromeView/DynamicView.hpp"
12
13 namespace MonochromeText
14 {
15
17 class MonochromeText
18 {
19 public:
20     static void WriteChar(MonochromeView::DynamicView& rView,
21                          const int32_t x, const int32_t y,
22                          const MonochromeFont& rFont,
23                          const char character,
24                          const uint8_t drawOption = MonochromeView::DRAW_OPT_NONE);
25
26     static void WriteString(MonochromeView::DynamicView& rView,
27                            const int32_t x, const int32_t y,

```

```

44         const MonochromeFont& rFont,
45         const char* const pString,
46         const uint8_t drawOption = MonochromeView::DRAW_OPT_NONE);
47
48 private:
49     MonochromeText(const MonochromeText&) = delete;
50     void operator=(const MonochromeText&) = delete;
51 };
52
53 }
54
55 #endif

```

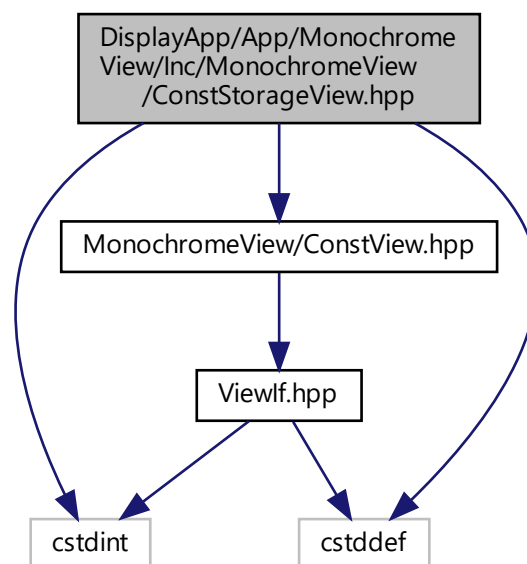
5.21 DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstStorageView.hpp File Reference

```

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

```

Include dependency graph for ConstStorageView.hpp:



Classes

- class `MonochromeView::ConstStorageView< WIDTH, HEIGHT >`

5.21.1 Detailed Description

Note

Copyright (c) 2021 ArmCp - Kala, Jaraczewski

5.22 ConstStorageView.hpp

[Go to the documentation of this file.](#)

```

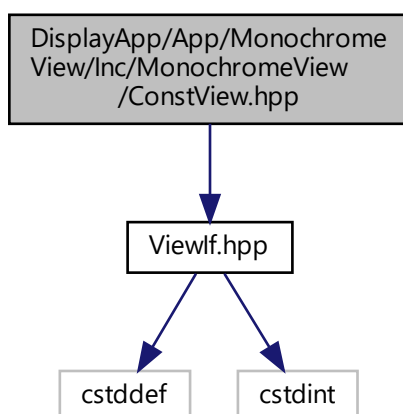
1
4
5 #ifndef MONOCHROMEVIEW_CONSTSTORAGEVIEW_HPP
6 #define MONOCHROMEVIEW_CONSTSTORAGEVIEW_HPP
7
8 #include <cstdint>
9 #include <cstddef>
10
11 #include "MonochromeView/ConstView.hpp"
12
13 namespace MonochromeView
14 {
15
20 template<size_t WIDTH, size_t HEIGHT>
21 class ConstStorageView :
22     public ConstView
23 {
24 public:
25     template<typename... ViewBytesTypes>
26     ConstStorageView(ViewBytesTypes... viewBytes) :
27         ConstView(m_Buffer, WIDTH, HEIGHT),
28         m_Buffer{static_cast<uint8_t>(viewBytes)...}
29     {
30     }
31
32 private:
33     const uint8_t m_Buffer[WIDTH * ((HEIGHT + ViewIf::PIXELS_PER_CELL - 1) / ViewIf::PIXELS_PER_CELL)];
34 };
35
36 #endif

```

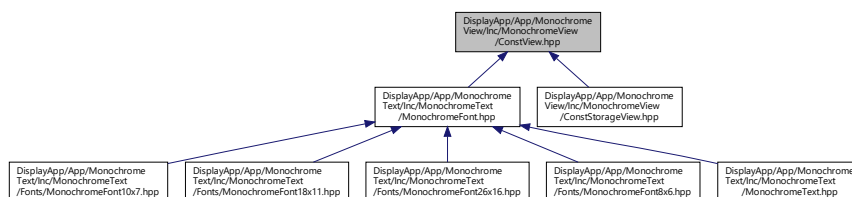
5.23 DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstView.hpp File Reference

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

Include dependency graph for ConstView.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [MonochromeView::ConstView](#)

Constant view.

5.23.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.24 ConstView.hpp

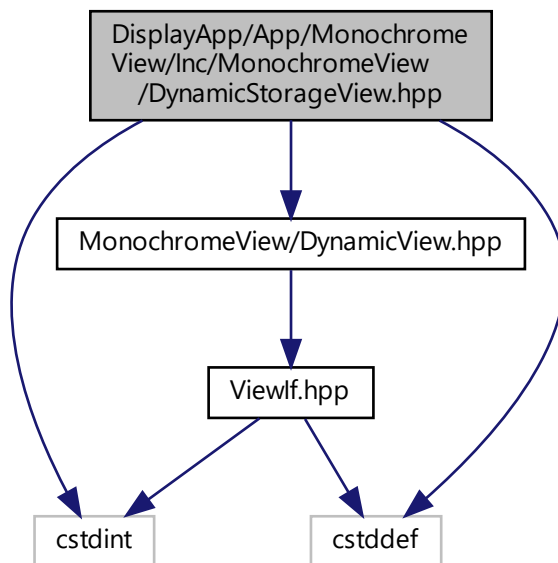
[Go to the documentation of this file.](#)

```

1
4
5 #ifndef MONOCHROMEVIEW_CONSTVIEW_HPP
6 #define MONOCHROMEVIEW_CONSTVIEW_HPP
7
8 #include "ViewIf.hpp"
9
10 namespace MonochromeView
11 {
12
13 class ConstView :
14     public ViewIf
15 {
16 public:
17     ConstView(const uint8_t* const pBuffer, const size_t width, const size_t height);
18
19     size_t Width() const override;
20
21     size_t Height() const override;
22
23     bool GetPixelColor(const size_t x, const size_t y) const override;
24
25     bool IfViewChanged() const;
26
27 private:
28     const uint8_t* Data() const override;
29
30     const uint8_t* const m_pBuffer;
31     const size_t m_Width;
32     const size_t m_Height;
33 };
34
35 #endif
  
```

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

```
#include <stdint>
#include <stddef>
#include "MonochromeView/DynamicView.hpp"
Include dependency graph for DynamicStorageView.hpp:
```



Classes

- class [MonochromeView::DynamicStorageView< WIDTH, HEIGHT >](#)

5.25.1 Detailed Description

Note

Copyright (c) 2021 ArmCpp - Kala, Jaraczewski

5.26 DynamicStorageView.hpp

[Go to the documentation of this file.](#)

```
1
4
5 #ifndef MONOCHROMEVIEW_DYNAMICSTORAGEVIEW_HPP
6 #define MONOCHROMEVIEW_DYNAMICSTORAGEVIEW_HPP
7
8 #include <stdint>
9 #include <stddef>
10
11 #include "MonochromeView/DynamicView.hpp"
12
```

```

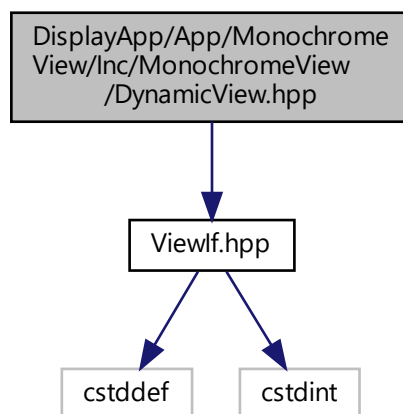
13 namespace MonochromeView
14 {
15
20 template<size_t WIDTH, size_t HEIGHT>
21 class DynamicStorageView :
22     public DynamicView
23 {
24 public:
25     template<typename... ViewBytesTypes>
26     DynamicStorageView(ViewBytesTypes... viewBytes) :
27         DynamicView(m_Buffer, WIDTH, HEIGHT),
28         m_Buffer{static_cast<uint8_t>(viewBytes)...}
29     {
30     }
31
32 private:
33     DynamicStorageView(const DynamicStorageView&) = delete;
34     void operator=(const DynamicStorageView&) = delete;
35
36     uint8_t m_Buffer[WIDTH * ((HEIGHT + ViewIf::PIXELS_PER_CELL - 1) / ViewIf::PIXELS_PER_CELL)];
37 };
38
39 #endif

```

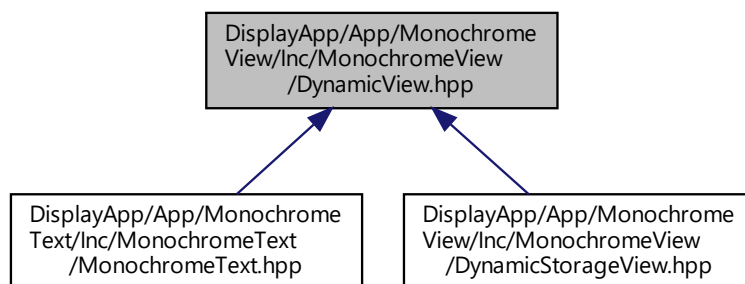
5.27 DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicView.hpp File Reference

#include "ViewIf.hpp"

Include dependency graph for DynamicView.hpp:



This graph shows which files directly or indirectly include this file:



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 ArmCcpp - Kala, Jaraczewski

5.28 DynamicView.hpp

[Go to the documentation of this file.](#)

```

1
4
5 #ifndef MONOCHROMEVIEW_DYNAMICVIEW_HPP
6 #define MONOCHROMEVIEW_DYNAMICVIEW_HPP
7
8 #include "ViewIf.hpp"
9
10 namespace MonochromeView
11 {
  
```

```

12
14 constexpr const uint8_t DRAW_OPT_NONE = 0x00U;
15
17 constexpr const uint8_t DRAW_OPT_TRANSPOSE = 0x01U;
18
20 constexpr const uint8_t DRAW_OPT_X_MIRROR = 0x02U;
21
23 constexpr const uint8_t DRAW_OPT_Y_MIRROR = 0x04U;
24
26 constexpr const uint8_t DRAW_OPT_NEGATIVE_COLORS = 0x08U;
27
29 class DynamicView :
30     public ViewIf
31 {
32 public:
33     DynamicView(uint8_t* const pBuffer, const size_t width, const size_t height);
34
35     size_t Width() const override;
36
37     size_t Height() const override;
38
39     bool GetPixelColor(const size_t x, const size_t y) const override;
40
41     bool IfViewChanged() const;
42
43     void DrawAt(const int32_t x,
44                 const int32_t y,
45                 const ViewIf& rAnotherView,
46                 const uint8_t drawOption = DRAW_OPT_NONE);
47
48     void DrawLine(const int32_t x1, const int32_t y1,
49                   const int32_t x2, const int32_t y2,
50                   const bool color);
51
52     void SetPixelColor(const size_t x, const size_t y, const bool color);
53
54     void Fill(const bool color);
55
56 private:
57     DynamicView(const DynamicView&) = delete;
58     void operator=(const DynamicView&) = delete;
59
60     const uint8_t* Data() const override;
61
62     uint8_t* const m_pBuffer;
63     const size_t m_Width;
64     const size_t m_Height;
65     mutable bool m_IfViewChanged;
66 };
67
68 }
69
70 #endif

```

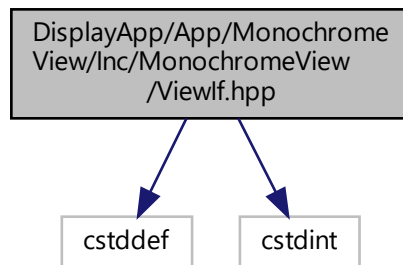
5.29 DisplayApp/App/MonochromeView/Inc/MonochromeView/ViewIf.hpp File Reference

```

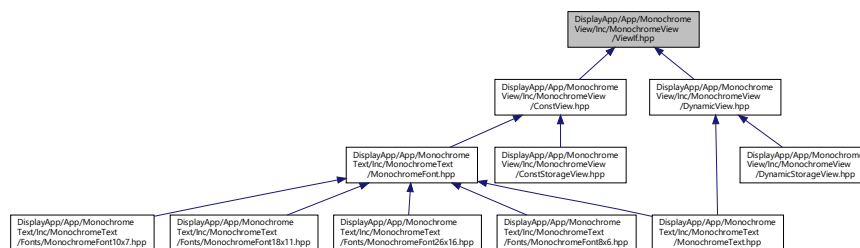
#include <cstdint>
#include <cstdint>

```


Include dependency graph for ViewIf.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class `MonochromeView::ViewIf`
View interface.

5.29.1 Detailed Description

Note

Copyright (c) 2021 ArmC++ - Kala, Jaraczewski

5.30 ViewIf.hpp

[Go to the documentation of this file.](#)

```

1
4
5 #ifndef MONOCHROMEVIEW_VIEWIF_HPP
6 #define MONOCHROMEVIEW_VIEWIF_HPP
7
8 #include <cstdint>
9 #include <cstddef>
10
11 namespace MonochromeView
12 {
  
```

```

13
15 class ViewIf
16 {
17 public:
18     static constexpr const size_t PIXELS_PER_CELL = 8U;
19
20     virtual ~ViewIf()
21     {
22     }
23
24     virtual size_t Width() const = 0;
25
26     virtual size_t Height() const = 0;
27
28     virtual const uint8_t* Data() const = 0;
29
30     virtual bool GetPixelColor(const size_t x, const size_t y) const = 0;
31
32     virtual bool IfViewChanged() const = 0;
33 };
34
35 #endif

```

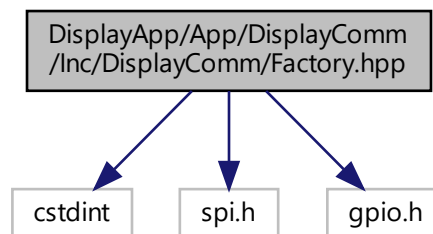
5.31 DisplayApp/App/DisplayComm/Inc/DisplayComm/Factory.hpp File Reference

```

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

```

Include dependency graph for Factory.hpp:



Classes

- class [DisplayComm::Factory](#)
Factory for display communication layer.

5.31.1 Detailed Description

Note

Copyright (c) 2021 ArmCxx - Kala, Jaraczewski

5.32 Factory.hpp

[Go to the documentation of this file.](#)

```

1
4
5 #ifndef DISPLAY_COMM_FACTORY_HPP
6 #define DISPLAY_COMM_FACTORY_HPP
7
8 #include <cstdint>
9
10 #include "spi.h"
11 #include "gpio.h"
12
13 namespace DisplayComm
14 {
15
16 class DisplayCommIf;
17 class DisplayResetIf;
18 class DisplayDataCmdIf;
19
20 class Factory
21 {
22 public:
23     static DisplayResetIf* CreateDisplayReset(GPIO_TypeDef* const pResetPort,
24                                               const uint32_t resetPin);
25
26     static DisplayDataCmdIf* CreateDataCmd(GPIO_TypeDef * const pDcPort,
27                                           const uint32_t dcPin);
28
29     static DisplayCommIf* CreateDisplayCommSpi(SPI_TypeDef* const pSpi,
30                                               GPIO_TypeDef* const pCsPort,
31                                               const uint32_t csPin,
32                                               const DisplayDataCmdIf* const pDataCmdIf);
33 private:
34     Factory(const Factory&) = delete;
35     void operator=(const Factory&) = delete;
36 };
37
38 }
39
40 #endif

```

5.33 Factory.hpp

```

1
4
5 #ifndef SH1106_FACTORY_HPP
6 #define SH1106_FACTORY_HPP
7
8 #include <cstdint>
9
10 namespace MonochromeGraphicDisplay
11 {
12
13 class DisplayDriverIf;
14
15 }
16
17 namespace DisplayComm
18 {
19
20 class DisplayCommIf;
21 class DisplayResetIf;
22
23 }
24
25 namespace Sh1106
26 {
27
28 class Factory
29 {
30 public:
31     static MonochromeGraphicDisplay::DisplayDriverIf* Create128x32Driver(DisplayComm::DisplayCommIf*
32     const pDisplayCommIf,
33                                     DisplayComm::DisplayResetIf*
34     const pDisplayResetIf,
35     const bool mirrorVertically =
36     false,
37     const bool mirrorHorizontally =
38     false,
39     const size_t columnOffset = 0U);
40
41 }
42
43 }

```

```
48
60     static MonochromeGraphicDisplay::DisplayDriverIf* Create128x64Driver(DisplayComm::DisplayCommIf*
61     const pDisplayCommIf,
62                                     DisplayComm::DisplayResetIf*
63     const pDisplayResetIf,
64                                     const bool mirrorVertically =
65     false,
66                                     const bool mirrorHorizontally =
67     false,
68                                     const size_t columnOffset = 0U);
77     static MonochromeGraphicDisplay::DisplayDriverIf* Create128x128Driver(DisplayComm::DisplayCommIf*
78     const pDisplayCommIf,
79                                     DisplayComm::DisplayResetIf*
80     const pDisplayResetIf,
81                                     const bool mirrorVertically =
82     false,
83                                     const bool mirrorHorizontally =
84     false,
85                                     const size_t columnOffset =
86     0U);
87 private:
88     Factory(const Factory&) = delete;
89     void operator=(const Factory&) = delete;
90 };
91 #endif
```


Index

ConstStorageView
 MonochromeView::ConstStorageView< WIDTH,
 HEIGHT >, 6

ConstView
 MonochromeView::ConstView, 8

Create128x128Driver
 Sh1106::Factory, 27

Create128x32Driver
 Sh1106::Factory, 27

Create128x64Driver
 Sh1106::Factory, 28

CreateDataCmd
 DisplayComm::Factory, 25

CreateDisplayCommSpi
 DisplayComm::Factory, 25

CreateDisplayReset
 DisplayComm::Factory, 26

DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayComm.hpp, 35, 36

DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayDataCmd.hpp, 36

DisplayApp/App/DisplayComm/Inc/DisplayComm/DisplayReset.hpp, 37

DisplayApp/App/DisplayComm/Inc/DisplayComm/Factory.hpp, 55, 56

DisplayApp/App/MonochromeGraphicDisplay/Inc/MonochromeGraphicDisplay/DisplayDriverIf.hpp, 37, 38

DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont10x7.hpp, 39, 40

DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont18x18.hpp, 40, 41

DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont26x16.hpp, 41, 42

DisplayApp/App/MonochromeText/Inc/MonochromeText/Fonts/MonochromeFont8x8.hpp, 43

DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeFont.hpp, 44, 45

DisplayApp/App/MonochromeText/Inc/MonochromeText/MonochromeText.hpp, 45, 46

DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstStorageView.hpp, 47, 48

DisplayApp/App/MonochromeView/Inc/MonochromeView/ConstView.hpp, 48, 49

DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicStorageView.hpp, 50

DisplayApp/App/MonochromeView/Inc/MonochromeView/DynamicView.hpp, 51, 52

DisplayApp/App/MonochromeView/Inc/MonochromeView/ViewIf.hpp, 53, 54

DisplayApp/App/Sh1106/Inc/Sh1106/Factory.hpp, 56

DisplayComm::DisplayCommIf, 11
 WriteCmd, 11
 WriteData, 12

DisplayComm::DisplayDataCmdIf, 12

DisplayComm::DisplayResetIf, 16

DisplayComm::Factory, 24
 CreateDataCmd, 25
 CreateDisplayCommSpi, 25
 CreateDisplayReset, 26

DrawAt
 MonochromeView::DynamicView, 22

DrawLine
 MonochromeView::DynamicView, 22

DynamicStorageView
 MonochromeView::DynamicStorageView< WIDTH,
 HEIGHT >, 19

DynamicView
 MonochromeView::DynamicView, 22

Fill
 MonochromeView::DynamicView, 23

GetCharView
 MonochromeText::MonochromeFont, 30

GetHeight
 MonochromeGraphicDisplay::DisplayDriverIf, 14
 MonochromeText::MonochromeFont, 30

GetPixelColor
 MonochromeView::ConstView, 10
 MonochromeView::DynamicView, 23
 MonochromeView::ViewIf, 34

GetSize
 MonochromeGraphicDisplay/DisplayDriverIf.hpp, 14
 MonochromeGraphicDisplay::DisplayDriverIf, 14

GetWidth
 MonochromeGraphicDisplay::DisplayDriverIf, 14
 MonochromeText::MonochromeFont, 30

Height
 MonochromeFont26x16.hpp, 41, 42
 MonochromeView::ConstView, 10
 MonochromeView::DynamicView, 23
 MonochromeView::ViewIf, 34

InverseColor
 MonochromeView::ConstView, 10
 MonochromeView::DynamicView, 23
 MonochromeView::ViewIf, 34

InverseColor
 MonochromeGraphicDisplay::DisplayDriverIf, 14

MonochromeFont
 DynamicStorageView.hpp, 50
 MonochromeText::MonochromeFont, 29

MonochromeGraphicDisplay::DisplayDriverIf, 13
 GetHeight, 14
 GetView, 14
 GetWidth, 14
 InverseColor, 14
 RefreshScreen, 15
 SetContrast, 15
 TurnOffDisplay, 15
 TurnOnDisplay, 15

MonochromeText::MonochromeFont, 29

- GetCharView, [30](#)
- GetHeight, [30](#)
- GetWidth, [30](#)
- MonochromeFont, [29](#)
- MonochromeText::MonochromeText, [31](#)
 - WriteChar, [31](#)
 - WriteString, [32](#)
- MonochromeView::ConstStorageView< WIDTH, HEIGHT >, [3](#)
 - ConstStorageView, [6](#)
- MonochromeView::ConstView, [6](#)
 - ConstView, [8](#)
 - GetPixelColor, [10](#)
 - Height, [10](#)
 - IfViewChanged, [10](#)
 - Width, [10](#)
- MonochromeView::DynamicStorageView< WIDTH, HEIGHT >, [17](#)
 - DynamicStorageView, [19](#)
- MonochromeView::DynamicView, [19](#)
 - DrawAt, [22](#)
 - DrawLine, [22](#)
 - DynamicView, [22](#)
 - Fill, [23](#)
 - GetPixelColor, [23](#)
 - Height, [23](#)
 - IfViewChanged, [23](#)
 - SetPixelColor, [24](#)
 - Width, [24](#)
- MonochromeView::ViewIf, [32](#)
 - GetPixelColor, [34](#)
 - Height, [34](#)
 - IfViewChanged, [34](#)
 - Width, [35](#)
- RefreshScreen
 - MonochromeGraphicDisplay::DisplayDriverIf, [15](#)
- SetContrast
 - MonochromeGraphicDisplay::DisplayDriverIf, [15](#)
- SetPixelColor
 - MonochromeView::DynamicView, [24](#)
- Sh1106::Factory, [26](#)
 - Create128x128Driver, [27](#)
 - Create128x32Driver, [27](#)
 - Create128x64Driver, [28](#)
- TurnOffDisplay
 - MonochromeGraphicDisplay::DisplayDriverIf, [15](#)
- TurnOnDisplay
 - MonochromeGraphicDisplay::DisplayDriverIf, [15](#)
- Width
 - MonochromeView::ConstView, [10](#)
 - MonochromeView::DynamicView, [24](#)
 - MonochromeView::ViewIf, [35](#)
- WriteChar
 - MonochromeText::MonochromeText, [31](#)
- WriteCmd
 - DisplayComm::DisplayCommIf, [11](#)
 - WriteData
 - DisplayComm::DisplayCommIf, [12](#)
 - WriteString
 - MonochromeText::MonochromeText, [32](#)