Object Detection

5.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

Main Page

a Cross platform app for circle object detection written in c++ 11/14 with Opencv and Qt

Author

Mohamed Khaled (icraus)

Version

5.0
Project page: https://github.com/Icraus/Object-Detector

Features:

1- **Cross platform**
.linux
.windows
.android(todo Add Tests, add android_ios view)
.raspian (not Tested)

2- **Can Be extended using Plugins**
Plugins are dynamically linked librarys loaded at run time The Object detector uses this feature for better detection See ObjectDetector::AddFilters

3- **Performance:** the Imageprocessing library provides high performance as it uses c++ and qt so there is no run time overhead even when using plugins

Utility Frameworks provides

:
1- From cv::Mat to QImage Library
2- a Qt Designer widget Plugin For using Opencv Video Capture
3- SerialPort Tool For Debugging and testing Serial ports
4- SerialPort Model/View widget for loading serial ports
5- Image Filters Plugins used to extend Object Detector Functionality

Note

by default the library is linked as shared library but u can compile it for static compilation
by default the project is compiled using c++14 but you can compile it using c++11

Todo :

1- add Cuda Support to provide parallel computing capilities
2- improved UI using QML
3- add Device interface to provide a unified api for device interactions like serial and bluetooth
4- migration to opencv Classifier for object tracking and detection

The Core Part of this project is the ImageProcessors library

See also

ImageProcessors

Chapter 2

Todo List

Class ImageProcessor::AbstractImageProcessor

add cv::Mat cache(may be using flyweight pattern) To avoid heavy copy of cv::Mat obejcts.

Member ImageProcessor::DetectColor::detectColor ()

parallize thresholding operationg.

Class ImageProcessor::Dilate

add Other Morphological Operations like erode.

page Main Page

:
1- add Cuda Support to provide parallel computing capilities
2- improved UI using QML
3- add Device interface to provide a unified api for device interactions like serial and bluetooth
4- migration to opencv Classifier for object tracking and detection

Chapter 3

Namespace Index

3.1 Namespace List

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

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Class Index

5.1 Class List

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

Chapter 6

Namespace Documentation

6.1 CircleDetectorPlugins Namespace Reference

Common namespace For all Plugins interfaces.

Classes

- class ImageProcessorPluginIFace

  The ImageProcessorPluginIFace is and interface used to apply filters to Images.

6.1.1 Detailed Description

Common namespace For all Plugins interfaces.

6.2 ImageProcessor Namespace Reference

Common Namespace for all Image Processor Algorithms.

Classes

- class AbstractImageProcessor

  The ImageProcessor::AbstractImageProcessor is an Abstract Base Class For All Image Processor Classes.
- class DetectCircle

  this class is used To Detect circles in an image
- class DetectColor

  this class is used to Detect Color given it's range(min, max) of hsv colors.
- class Dilate

  this Class is used to perform morphological dilate operation on image see `Morphological Operation`.
- class ObjectDetection

  this class is used to detect a a colored circle object(s)
- class ObjectDetectorBuilder

Variables

- class IMG_PROC_LIB **AbstractImageProcessor**
- class IMG_PROC_LIB **DetectCircle**
- class IMG_PROC_LIB **DetectColor**
- class IMG_PROC_LIB **Dilate**
- class IMG_PROC_LIB **ObjectDetection**
- class IMG_PROC_LIB **ObjectDetectorBuilder**

6.2.1 Detailed Description

Common Namespace for all Image Processor Algorithms.

Author

Mohamed Khaled

See also

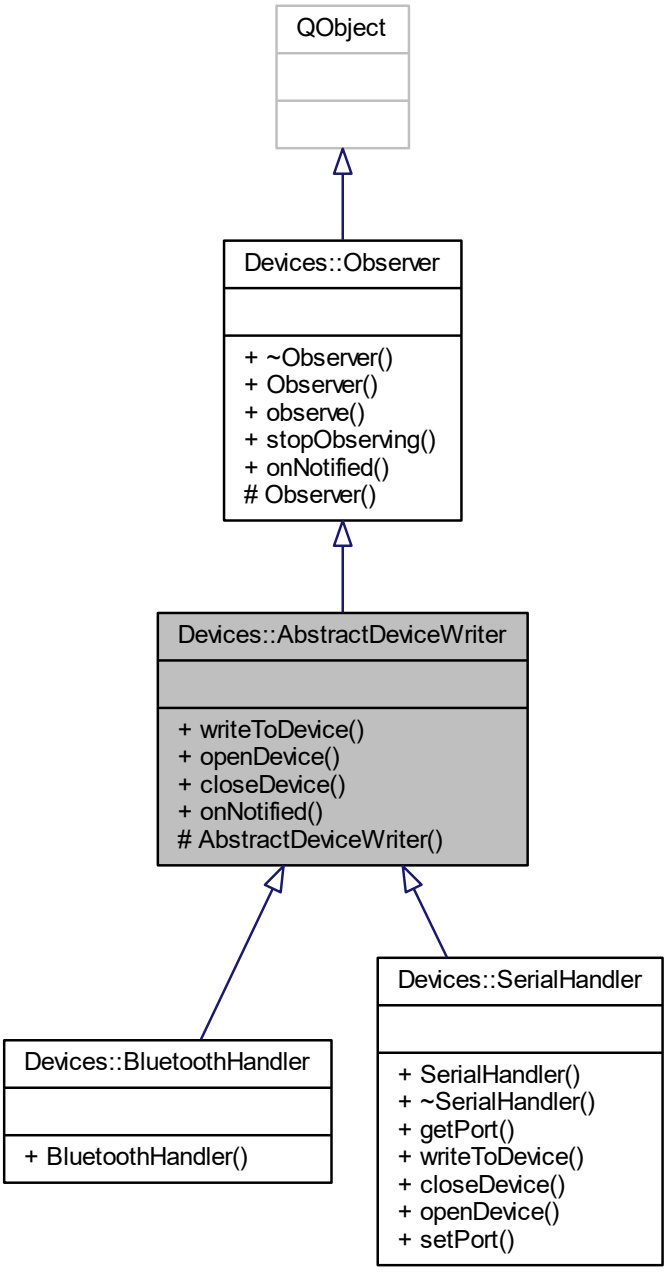
ImageProcessor::AbstractImageProcessor

Chapter 7

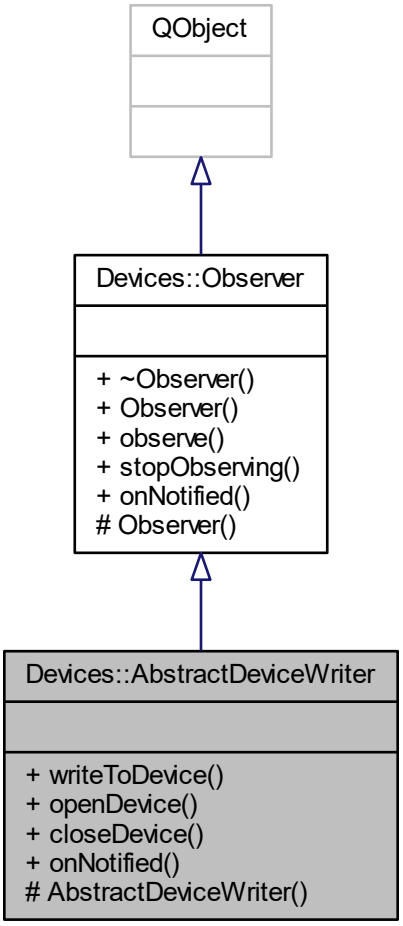
Class Documentation

7.1 Devices::AbstractDeviceWriter Class Reference

Inheritance diagram for Devices::AbstractDeviceWriter:



Collaboration diagram for Devices::AbstractDeviceWriter:



Public Slots

- virtual void **onNotified** (const ObservableData &dt) override

Signals

- void **errorIODevice** (const QString &)

Public Member Functions

- virtual void **writeToDevice** (const QString &)=0
- virtual void **openDevice** ()=0
- virtual void **closeDevice** ()=0

Protected Member Functions

- **AbstractDeviceWriter** (QObject ∗parent=nullptr)

7.1.1 Detailed Description

Definition at line 14 of file abstractdevicewriter.h.

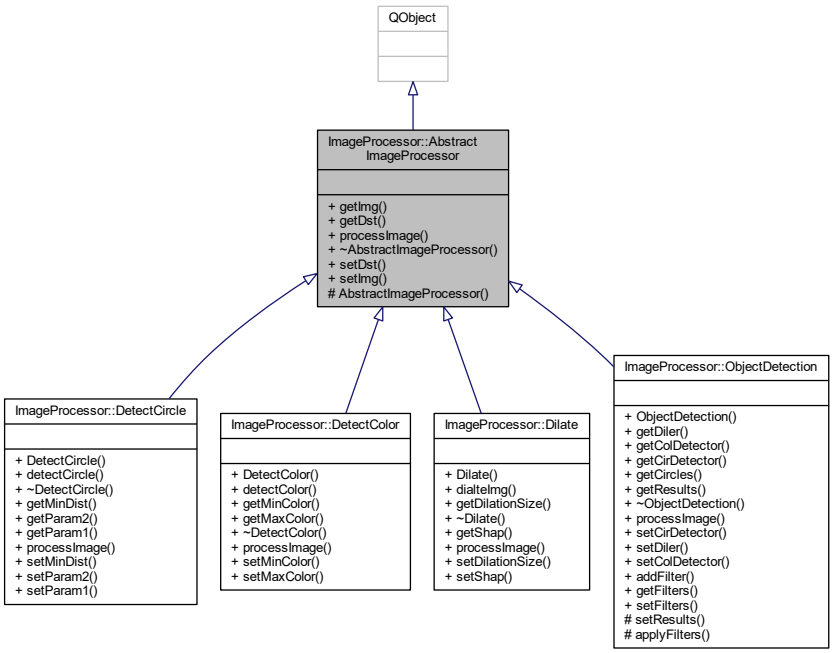
The documentation for this class was generated from the following files:

- object-detector/src/DevicesInterfaces/DeviceHandler/abstractdevicewriter.h
- object-detector/src/DevicesInterfaces/DeviceHandler/abstractdevicewriter.cpp

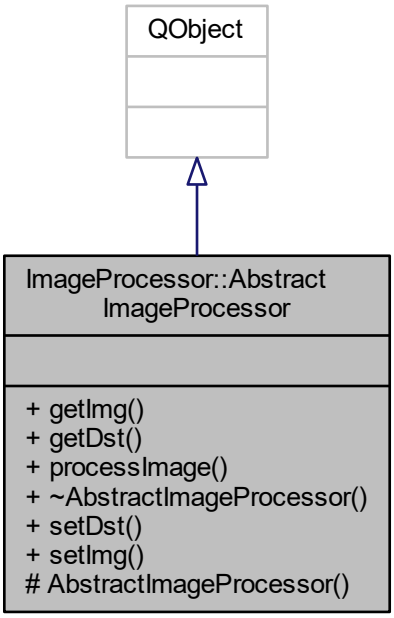
7.2 ImageProcessor::AbstractImageProcessor Class Reference

The ImageProcessor::AbstractImageProcessor is an Abstract Base Class For All Image Processor Classes.

Inheritance diagram for ImageProcessor::AbstractImageProcessor:



Collaboration diagram for ImageProcessor::AbstractImageProcessor:



Classes

- class **_AbstractImageProcessorImpl**

Public Slots

- virtual void setDst (const cv::Mat &dst)

  sets The output of the operation
- virtual void **setImg** (const cv::Mat &img)

Signals

- void imageChanged (const cv::Mat &img)

  this Signal Is Emited When the source cv::Mat Object Changed.
- void dstChanged (const cv::Mat &img)

  this Signal Is Emited When the destnation cv::Mat Object Changed. ***example:***

Public Member Functions

- cv::Mat getImg () const

  AbstractImageProcessor::getImg.
- cv::Mat getDst () const

  returns A cv::Mat Object which represents the output of the image processing operation
- virtual QVariant processImage ()=0

  Pure Virtual Function representes the operation to be done on the Image to be processed.

Protected Member Functions

- AbstractImageProcessor (QObject *parent=nullptr)

  accpets A pointer To the Parent Class For The Qt Meta-object Model See Qt Meta-Object

7.2.1 Detailed Description

The ImageProcessor::AbstractImageProcessor is an Abstract Base Class For All Image Processor Classes.

this Class Provides A Common Data Structure For All Image Processor Classes That Inherits From It The class Provides cv::Mat Img Variable Which is used as The Soruce Image and cv::Mat dst Varaible Which is the Destnation cv::Mat Object.

Note

using getDst and getImg is using deep copy of image to avoid invalid access of member objects.

```
using namespace ImageProcessor;
std::unique_ptr<ImageProcessor::AbstractImageProcessor> imgproc =
      std::make_unique<ImageProcessor::DetectColor>(this);
imgproc->setImg(cv::imread(IMG_PATH));
connect(imgproc.get(), &AbstractImageProcessor::dstChanged,[=](const auto
      & img){qDebug() << "Image Changed";});
imgproc->processImage();
auto img = imgproc->getDst(); //returns cv::Mat and prints "Image Changed"
```

See also

ImageProcessor

Todo add cv::Mat cache(may be using flyweight pattern) To avoid heavy copy of cv::Mat obejcts.

Definition at line 11 of file abstractimageprocessor.h.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 AbstractImageProcessor()

```
AbstractImageProcessor::AbstractImageProcessor (
            QObject * parent = nullptr )  [protected]
```

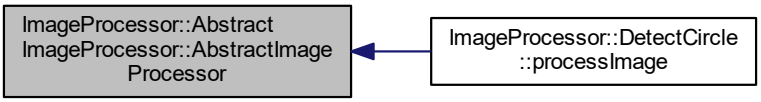
accpets A pointer To the Parent Class For The Qt Meta-object Model See Qt Meta-Object

Parameters

parent	A QObject Object as a parent

Definition at line 9 of file abstractimageprocessor.cpp.

Here is the caller graph for this function:



7.2.3 Member Function Documentation

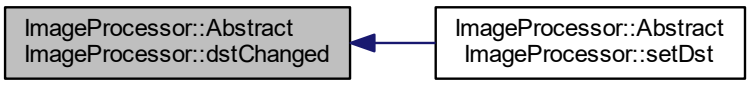
7.2.3.1 dstChanged

```
ImageProcessor::AbstractImageProcessor::dstChanged (
            const cv::Mat & img )  [signal]
```

this Signal Is Emited When the destnation cv::Mat Object Changed. **example:**

```
using namespace ImageProcessor;
std::unique_ptr<ImageProcessor::AbstractImageProcessor> imgproc =
      std::make_unique<ImageProcessor::DetectColor>(this);
imgproc->setImg(cv::imread(IMG_PATH));
connect(imgproc.get(), &AbstractImageProcessor::dstChanged,[=](const auto
      & img){qDebug() << "Image Changed";});
imgproc->processImage();
auto img = imgproc->getDst(); //returns cv::Mat and prints "Image Changed"
```

Here is the caller graph for this function:



7.2.3.2 getDst()

```
cv::Mat AbstractImageProcessor::getDst ( ) const
```

returns A cv::Mat Object which represents the output of the image processing operation

Returns

cv::Mat

Definition at line 20 of file abstractimageprocessor.cpp.

7.2.3.3 getImg()

```
cv::Mat AbstractImageProcessor::getImg ( ) const
```

AbstractImageProcessor::getImg.

Returns

cv::Matreturns A copy of The Source Image uses cv::Mat::clone(). **example:**

```
cv::Mat img = cv::imread(IMG_PATH);
imgproc.setImg(img);
cv::Mat img2 = imgproc.getImg(); //uses Deep Copy like img2 = img.clone();
```

Definition at line 46 of file abstractimageprocessor.cpp.

7.2.3.4 processImage()

```
ImageProcessor::AbstractImageProcessor::processImage ( )  [pure virtual]
```

Pure Virtual Function representes the operation to be done on the Image to be processed.

Exceptions

cv::Exception.not	garunteed to throw this exception

Warning

not exception nor thread safe.

Returns

QVariant Object which represents the output of the processing operation and it doesn't have to be cv::Mat.

Implemented in ImageProcessor::ObjectDetection, ImageProcessor::DetectCircle, ImageProcessor::Dilate, and ImageProcessor::DetectColor.

7.2.3.5 setDst

```
void AbstractImageProcessor::setDst (
            const cv::Mat & dst )  [virtual], [slot]
```

sets The output of the operation

Parameters

dst	

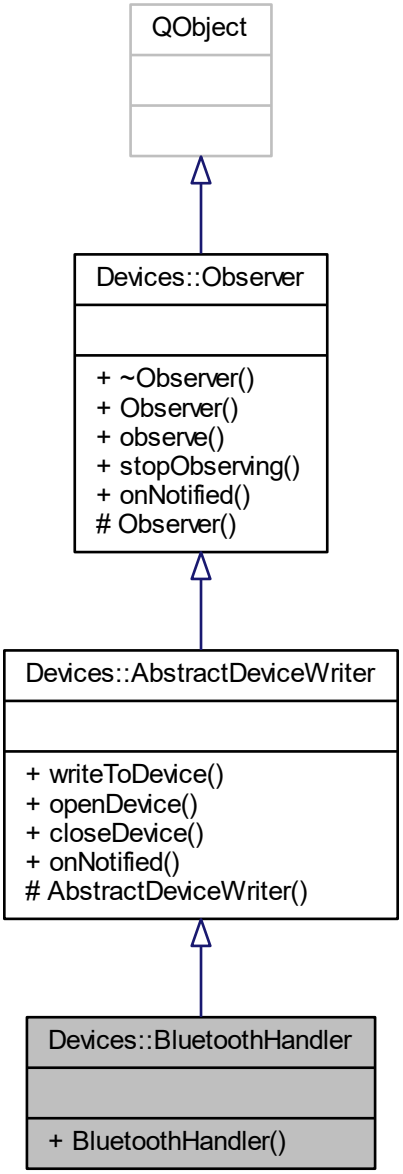
Definition at line 30 of file abstractimageprocessor.cpp.

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

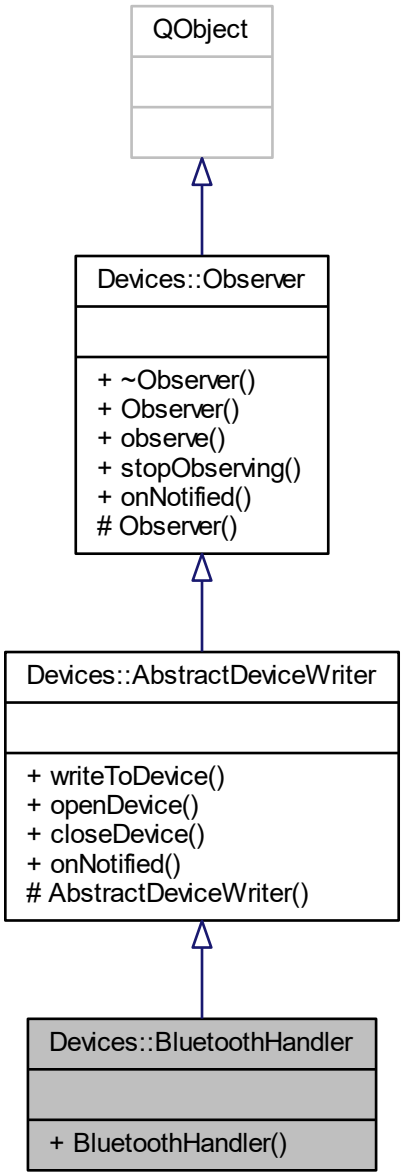
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/abstractimageprocessor.h
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/abstractimageprocessor.cpp

7.3 Devices::BluetoothHandler Class Reference

Inheritance diagram for Devices::BluetoothHandler:



Collaboration diagram for Devices::BluetoothHandler:



Public Member Functions

- **BluetoothHandler** (QObject *parent=nullptr)

Additional Inherited Members

7.3.1 Detailed Description

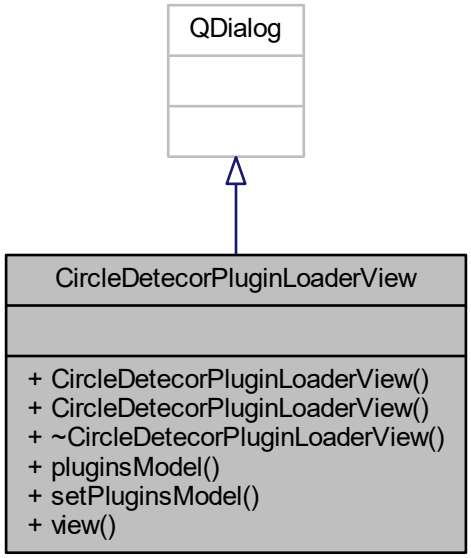
Definition at line 10 of file bluetoothhandler.h.

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

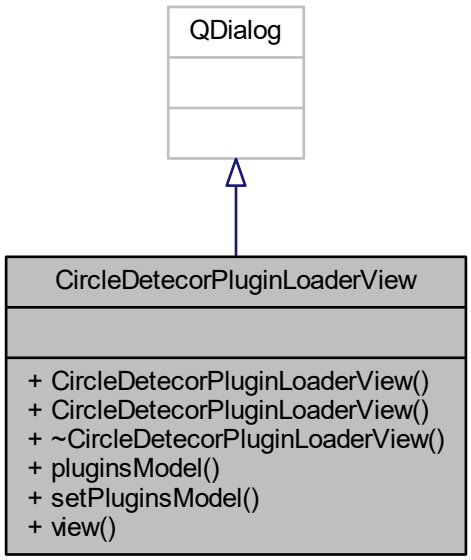
- object-detector/src/DevicesInterfaces/DeviceHandler/Bluetooth/bluetoothhandler.h
- object-detector/src/DevicesInterfaces/DeviceHandler/Bluetooth/bluetoothhandler.cpp

7.4 CircleDetecorPluginLoaderView Class Reference

Inheritance diagram for CircleDetecorPluginLoaderView:



Collaboration diagram for CircleDetecorPluginLoaderView:



Classes

- class **_CircleDetecorPluginLoaderViewImpl**

Signals

- void **filterChanged** (PluginSharedPointer selected)

Public Member Functions

- **CircleDetecorPluginLoaderView** (QWidget *parent=0)
- **CircleDetecorPluginLoaderView** (QString path, QWidget *parent=0)
- CircleDetectorPluginModel * **pluginsModel** () const
- void **setPluginsModel** (CircleDetectorPluginModel *pluginsModel)
- QTableView * **view** () const

7.4.1 Detailed Description

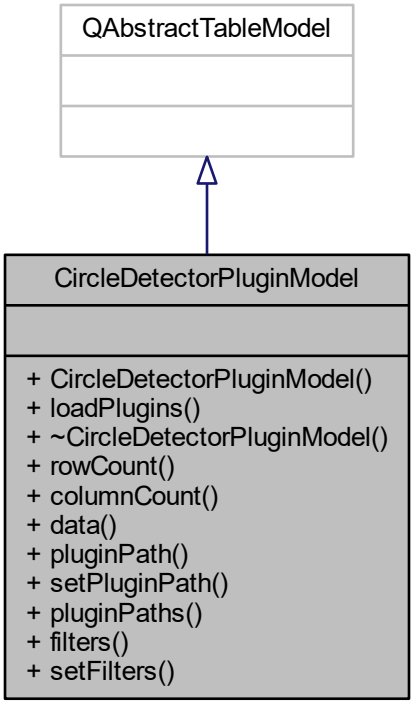
Definition at line 12 of file circledetecorpluginloaderview.h.

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

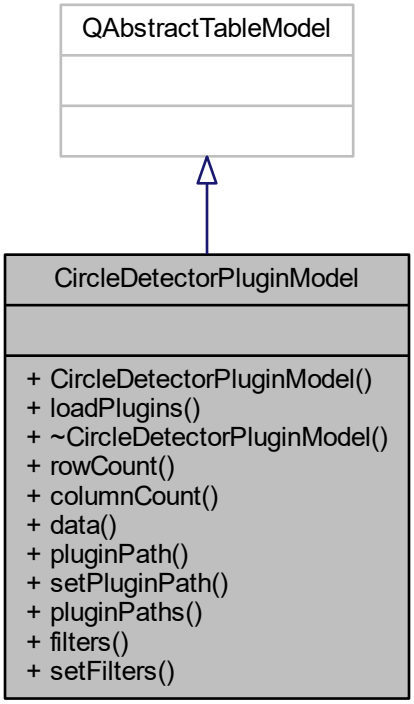
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/PluginLoader/CircleDetectorPluginLoader↩ View/circledetecorpluginloaderview.h
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/PluginLoader/CircleDetectorPluginLoader↩ View/circledetecorpluginloaderview.cpp

7.5 CircleDetectorPluginModel Class Reference

Inheritance diagram for CircleDetectorPluginModel:



Collaboration diagram for CircleDetectorPluginModel:



Classes

- class **_CircleDetectorPluginModelImpl**

Public Types

- enum **Type** {
  FILTER = Qt::UserRole + 1, **FILTER_NAME**, **FILTER_DESCRIPTION**, **FILTER_AUTHOR**,
  FILTER_PATH }

Public Member Functions

- **CircleDetectorPluginModel** (QString path=QDir::currentPath(), QObject *parent=nullptr)
- void **loadPlugins** ()
- virtual int **rowCount** (const QModelIndex &parent) const override
- virtual int **columnCount** (const QModelIndex &parent) const override
- virtual QVariant **data** (const QModelIndex &index, int role) const override
- QString **pluginPath** () const
- void **setPluginPath** (const QString &pluginPath)
- QList< QString > **pluginPaths** () const
- PluginSharedPointerList **filters** () const
- void **setFilters** (const PluginSharedPointerList &filters)

7.5.1 Detailed Description

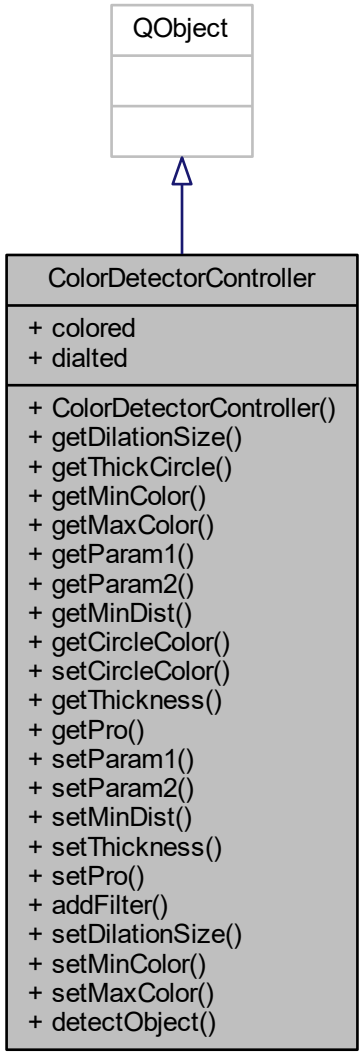
Definition at line 14 of file circledetectorpluginmodel.h.

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

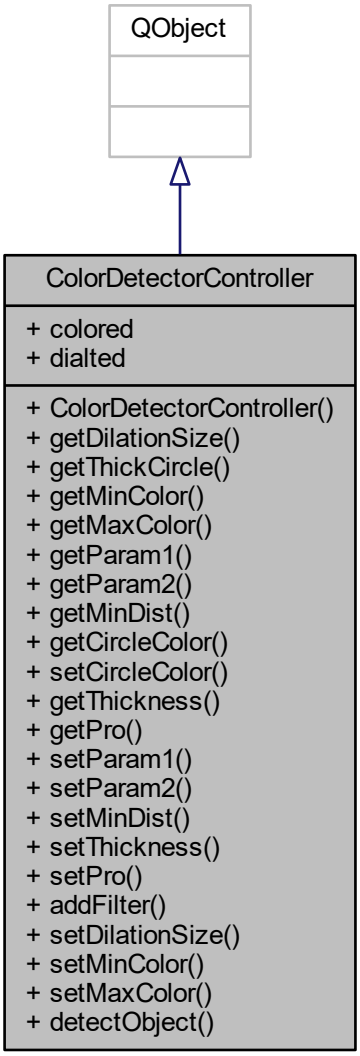
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/PluginLoader/CircleDetectorPluginModel/circledetectorpluginmodel.h
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/PluginLoader/CircleDetectorPluginModel/circledetectorpluginmodel.cpp

7.6 ColorDetectorController Class Reference

Inheritance diagram for ColorDetectorController:



Collaboration diagram for ColorDetectorController:



Public Slots

- void **setParam1** (int value)
- void **setParam2** (int value)
- void **setMinDist** (int value)
- void **setThickness** (int value)
- void **setPro** (ObjectDetection ∗value)
- void **addFilter** (PluginSharedPointer filter)
- void **setDilationSize** (int value)
- void **setMinColor** (const cv::Scalar &value)
- void **setMaxColor** (const cv::Scalar &value)
- Q_INVOKABLE QImage **detectObject** (const cv::Mat &t)

Signals

- void **dilationSizeChanged** (int)
- void **minColorChanged** (const cv::Scalar &)
- void **maxColorChanged** (const cv::Scalar &)
- void **param1Changed** (int)
- void **param2Changed** (int)
- void **minDistChanged** (int)
- void **xyrChanged** (int, int, int)

Public Member Functions

- **ColorDetectorController** (QObject *parent=nullptr)
- int **getDilationSize** () const
- int **getThickCircle** () const
- cv::Scalar **getMinColor** () const
- cv::Scalar **getMaxColor** () const
- int **getParam1** () const
- int **getParam2** () const
- int **getMinDist** () const
- cv::Scalar **getCircleColor** () const
- void **setCircleColor** (const cv::Scalar &value)
- int **getThickness** () const
- ObjectDetection * **getPro** () const

Public Attributes

- cv::Mat **colored**
- cv::Mat **dialted**

7.6.1 Detailed Description

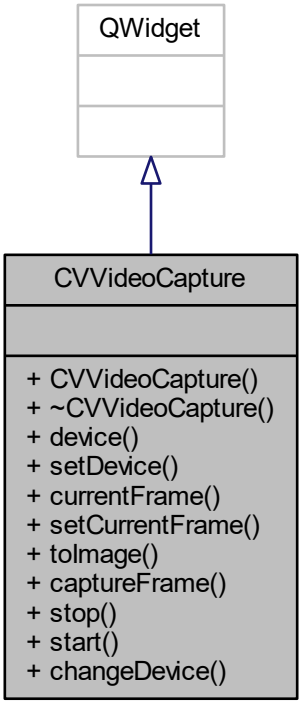
Definition at line 9 of file colordetectorcontroller.h.

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

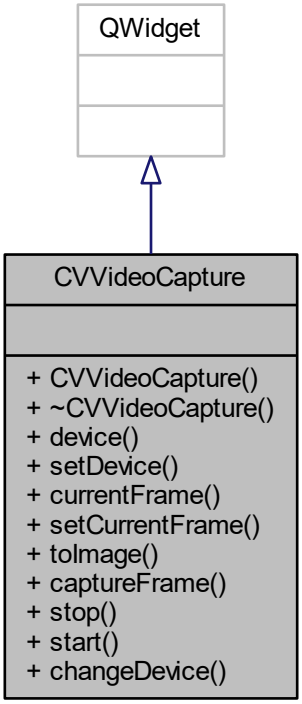
- object-detector/src/View/DesktopView/colordetectorcontroller.h
- object-detector/src/View/DesktopView/colordetectorcontroller.cpp

7.7 CVVideoCapture Class Reference

Inheritance diagram for CVVideoCapture:



Collaboration diagram for CVVideoCapture:



Public Slots

- virtual void **captureFrame** ()
- virtual void **stop** ()
- virtual void **start** ()
- virtual void **changeDevice** (int device)

Signals

- void **frameCaptured** (cv::Mat frame)
- void **stopped** ()
- void **started** ()
- void **deviceChanged** (int device)

Public Member Functions

- **CVVideoCapture** (QWidget ∗parent=0)
- int **device** () const
- void **setDevice** (int device)
- cv::Mat **currentFrame** () const
- void **setCurrentFrame** (const cv::Mat &currentFrame)
- QImage **toImage** (const cv::Mat &m)

7.7.1 Detailed Description

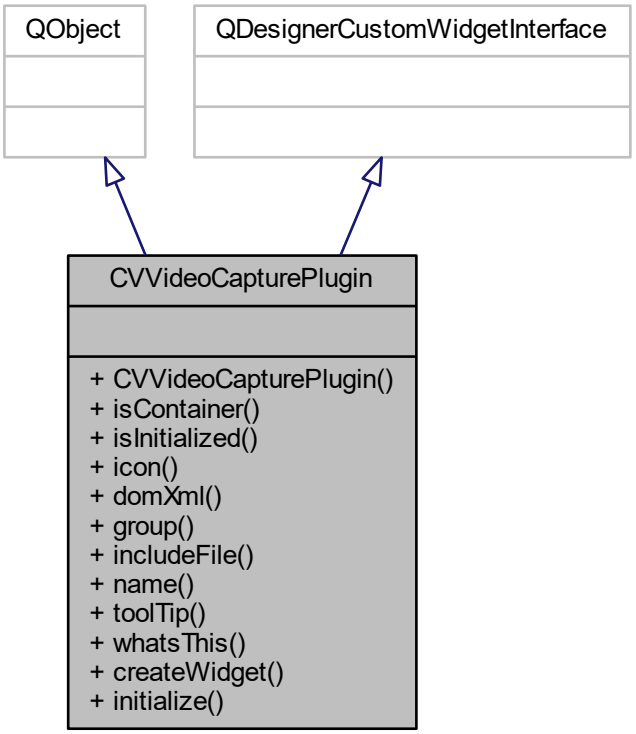
Definition at line 15 of file cvvideocapture.h.

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

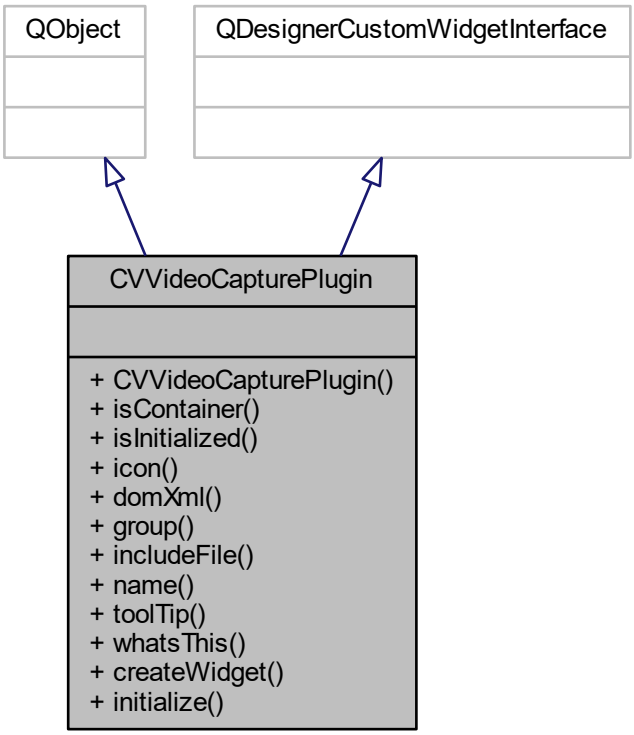
- object-detector/src/Utilities/Plugins/CVVideoCaptureLib/cvvideocapture.h
- object-detector/src/Utilities/Plugins/CVVideoCaptureLib/cvvideocapture.cpp

7.8 CVVideoCapturePlugin Class Reference

Inheritance diagram for CVVideoCapturePlugin:



Collaboration diagram for CVVideoCapturePlugin:



Public Member Functions

- **CVVideoCapturePlugin** (QObject ∗parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **domXml** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget ∗ **createWidget** (QWidget ∗parent)
- void **initialize** (QDesignerFormEditorInterface ∗core)

7.8.1 Detailed Description

Definition at line 6 of file cvvideocaptureplugin.h.

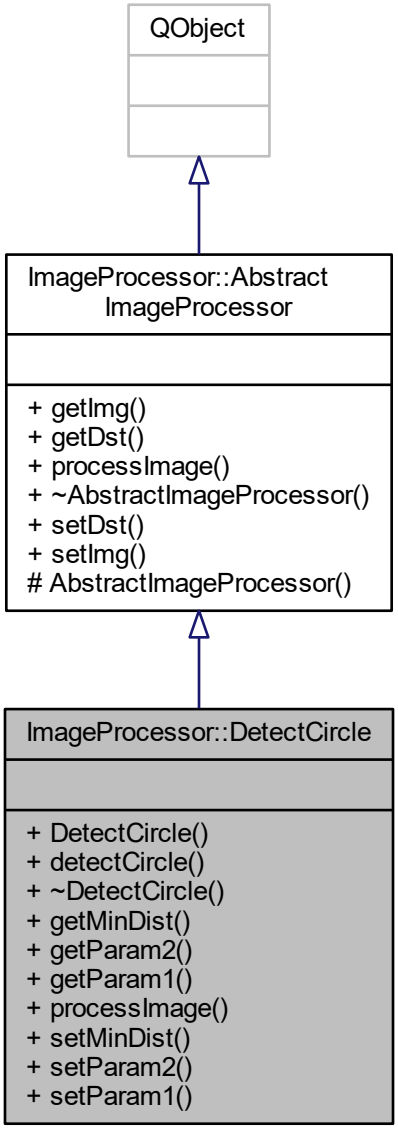
The documentation for this class was generated from the following files:

- object-detector/src/Utilities/Plugins/CVVideoCapture/cvvideocaptureplugin.h
- object-detector/src/Utilities/Plugins/CVVideoCapture/cvvideocaptureplugin.cpp

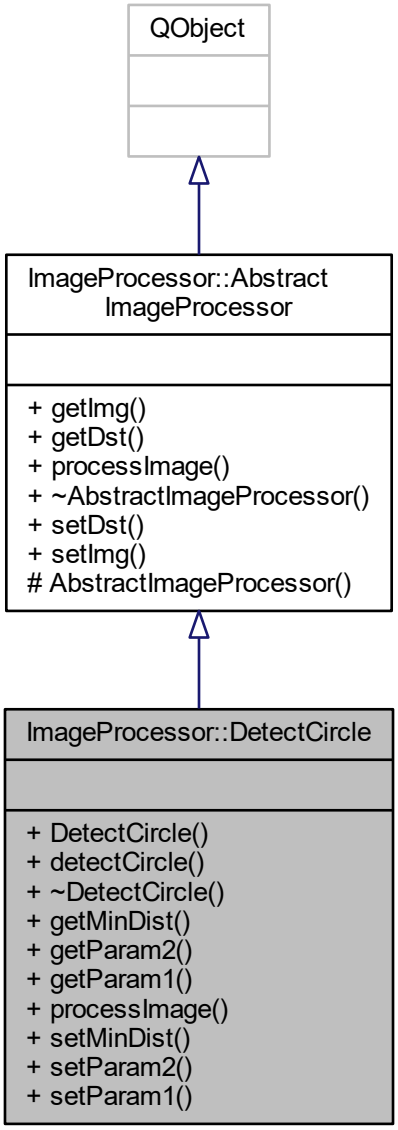
7.9 ImageProcessor::DetectCircle Class Reference

this class is used To Detect circles in an image

Inheritance diagram for ImageProcessor::DetectCircle:



Collaboration diagram for ImageProcessor::DetectCircle:



Classes

- class **_DetectCircleImpl**

Public Slots

- void setMinDist (int value)

  sets the minimum value between two circles
- void **setParam2** (int value)
- void **setParam1** (int value)

Signals

- void circlesDetected (const std::vector< cv::Vec3f > &)

  this signal is emitted after detecting All the circles in the image being processed.

Public Member Functions

- **DetectCircle** (QObject *parent=nullptr)
- std::vector< cv::Vec3f > detectCircle () const

  this helper Function is used to detect circles in an Image using HoughCircle Algorithm.
- int **getMinDist** () const
- int **getParam2** () const
- int **getParam1** () const
- QVariant processImage () override

  reimplemented Function.

Additional Inherited Members

7.9.1 Detailed Description

this class is used To Detect circles in an image

using this class you can detect x, y and radius of a circle in an image after calling DetectColor::processImage there are many possible ways to get The output for example you can connect the signal ImageProcessor::DetectCircle::circlesDetected to any QObject Slot that takes std::vector<cv::Vec3f> as a parameter another way is to use the return of processImage and convert the QVarient to std::vector<cv::Vec3f>

```
connect(crcleDetector, &DetectCircle::circlesDetected, [=](const
     std::vector<cv::Vec3f>& vec){
    for_each(vec.begin(), vec.end(), [](cv::Vec3f v){ qDebug() << v[0] << ", " << v[1] << ", " << v[2];});
});
std::vector<cv::Vec3f> vec = crcleDetector.processImage().value<std::vector<cv::Vec3f>>(); //and prints
      each x, y, r of every circle.
```

See also

ImageProcessor::DetectCircle::circlesDetected

Definition at line 12 of file detectcircle.h.

7.9.2 Member Function Documentation

7.9.2.1 circlesDetected

```
ImageProcessor::DetectCircle::circlesDetected (
            const std::vector< cv::Vec3f > & ) [signal]
```

this signal is emitted after detecting All the circles in the image being processed.

can Be connected with other objects to get the circles in an image.

```
connect(crcleDetector, &DetectCircle::circlesDetected, [=](const
      std::vector<cv::Vec3f>& vec){
     for_each(vec.begin(), vec.end(), [](cv::Vec3f v){ qDebug() << v[0] << ", " << v[1] << ", " << v[2];});
}; //prints each x, y, r of every circle.
```

7.9.2.2 detectCircle()

```
std::vector< cv::Vec3f > DetectCircle::detectCircle ( ) const
```

this helper Function is used to detect circles in an Image using HoughCircle Algorithm.

Returns

a vector of 3 points vector each represents the x, y, r of all circles in the image.

See also

ImageProcessor::DetectCirlce::processImage.

Definition at line 66 of file detectcircle.cpp.

7.9.2.3 processImage()

```
QVariant DetectCircle::processImage ( ) [override], [virtual]
```

reimplemented Function.

this function is reimplented to process a thresholded image of grayscale type to Detect All Cirlces.

See also

ImageProcessor::AbstractImageProcessor::processImage

Returns

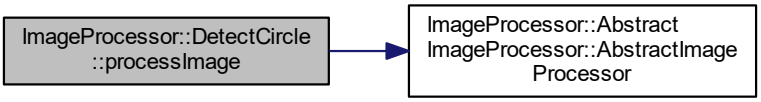
std::vector<cv::Vec3f> a vector of cv::Vec3f where each index in it represents a circle which centers are x, y and radius.

```
using circleVec = std::vector<cv::Vec3f>;
auto i = circleDetector.processImage().value<circleVec>();
for(auto a : i){
     qDebug() << "x = " << a[0] << ", y = " << a[1] << ", r = " << a[2]; //should iterate on each element
       and prints it's x, y, and radius.
}
```

Implements ImageProcessor::AbstractImageProcessor.

Definition at line 51 of file detectcircle.cpp.

Here is the call graph for this function:



7.9.2.4 setMinDist

```
void DetectCircle::setMinDist (
             int value )  [slot]
```

sets the minimum value between two circles

Parameters

value	

See also

ImageProcessor::DetectCircle::getMinDist

Definition at line 14 of file detectcircle.cpp.

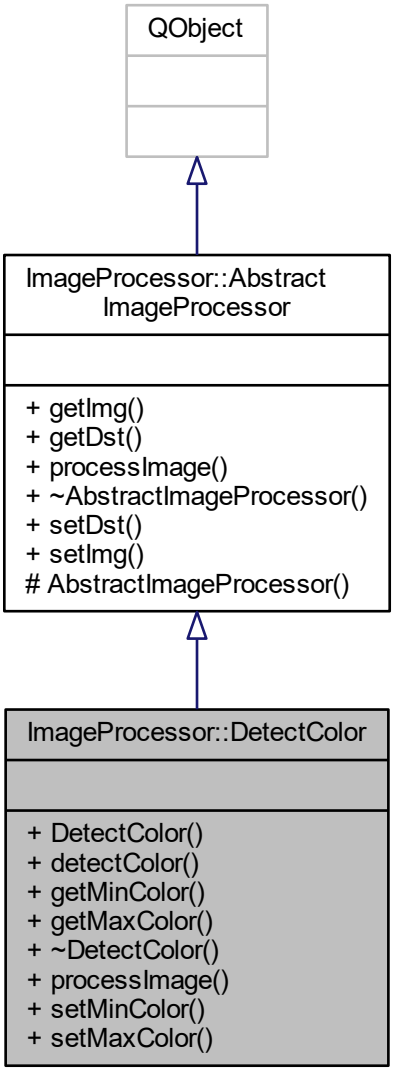
The documentation for this class was generated from the following files:

- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/detectcircle.h
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/detectcircle.cpp

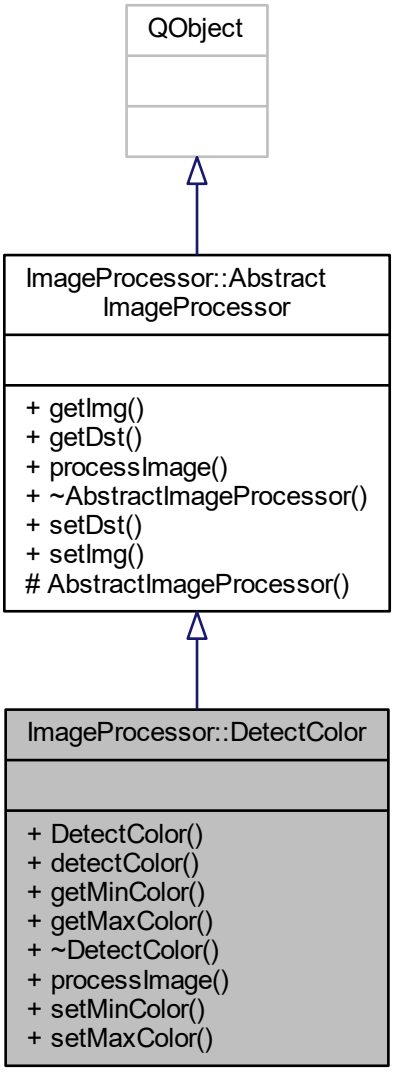
7.10 ImageProcessor::DetectColor Class Reference

this class is used to Detect Color given it's range(min, max) of hsv colors.

Inheritance diagram for ImageProcessor::DetectColor:



Collaboration diagram for ImageProcessor::DetectColor:



Classes

- class **_DetectColorImpl**

Public Slots

- void setMinColor (const cv::Scalar &value)

  sets The Minimum hsv color for the Detector
- void setMaxColor (const cv::Scalar &value)

  sets The maximum hsv color for the Detector

Public Member Functions

- **DetectColor** (QObject *parent=nullptr)
- void detectColor ()

  Helper Function Used To Detect Color.
- cv::Scalar getMinColor () const

  returns the Minimum HSV Color space Of Threshold Detector
- cv::Scalar getMaxColor () const

  returns the maximum HSV Color space Of Threshold Detector
- virtual QVariant processImage () override

  Pure Virtual Function representes the operation to be done on the Image to be processed.

Additional Inherited Members

7.10.1 Detailed Description

this class is used to Detect Color given it's range(min, max) of hsv colors.

Note

Default Color To Detect Is Yellow

```
  cv::Scalar minColor = cv::Scalar(20, 100, 100);
 cv::Scalar  maxColor = cv::Scalar(30, 255, 255); // detect Yellow Color
 ImageProcessor::DetectColor *proc = new DetectColor(nullptr);
 proc->setMaxColor(maxColor);
proc->setMinColor(minColor);
proc->processImage();
auto img = proc->getDst();
cv::nameWindow("window");
cv::imshow("window", img);
cv::waitKey(0);
```

Definition at line 13 of file detectcolor.h.

7.10.2 Member Function Documentation

7.10.2.1 detectColor()

```
void DetectColor::detectColor ( )
```

Helper Function Used To Detect Color.

Todo parallize thresholding operationg.

Version

2.0

Definition at line 59 of file detectcolor.cpp.

7.10.2.2 getMaxColor()

```
cv::Scalar DetectColor::getMaxColor ( ) const
```

returns the maximum HSV Color space Of Threshold Detector

Returns

cv::Scalar

Definition at line 29 of file detectcolor.cpp.

7.10.2.3 getMinColor()

```
cv::Scalar DetectColor::getMinColor ( ) const
```

returns the Minimum HSV Color space Of Threshold Detector

Returns

cv::Scalar

Definition at line 13 of file detectcolor.cpp.

7.10.2.4 processImage()

```
QVariant DetectColor::processImage ( )  [override], [virtual]
```

Pure Virtual Function representes the operation to be done on the Image to be processed.

Exceptions

cv::Exception.not	garunteed to throw this exception

Warning

not exception nor thread safe.

Returns

QVariant Object which represents the output of the processing operation and it doesn't have to be cv::Mat.

Implements ImageProcessor::AbstractImageProcessor.

Definition at line 48 of file detectcolor.cpp.

7.10.2.5 setMaxColor

```
void DetectColor::setMaxColor (
            const cv::Scalar & value )  [slot]
```

sets The maximum hsv color for the Detector

Parameters

cv::Scalar	of the HSV Color

Definition at line 43 of file detectcolor.cpp.

7.10.2.6 setMinColor

```
void DetectColor::setMinColor (
            const cv::Scalar & value )  [slot]
```

sets The Minimum hsv color for the Detector

Parameters

cv::Scalar	of the HSV Color

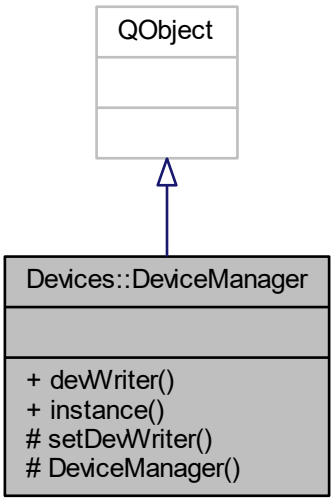
Definition at line 21 of file detectcolor.cpp.

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

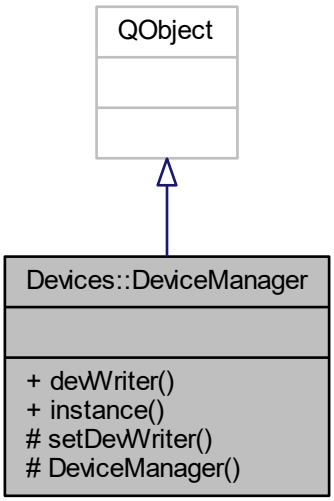
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/detectcolor.h
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/detectcolor.cpp

7.11 Devices::DeviceManager Class Reference

Inheritance diagram for Devices::DeviceManager:



Collaboration diagram for Devices::DeviceManager:



Public Member Functions

- Devices::AbstractDeviceWriter * **devWriter** () const

Static Public Member Functions

- static DeviceManager & **instance** (QObject *parent=nullptr)

Protected Member Functions

- void **setDevWriter** (Devices::AbstractDeviceWriter *devWriter)
- **DeviceManager** (QObject *parent=nullptr)

7.11.1 Detailed Description

Definition at line 12 of file devicemanager.h.

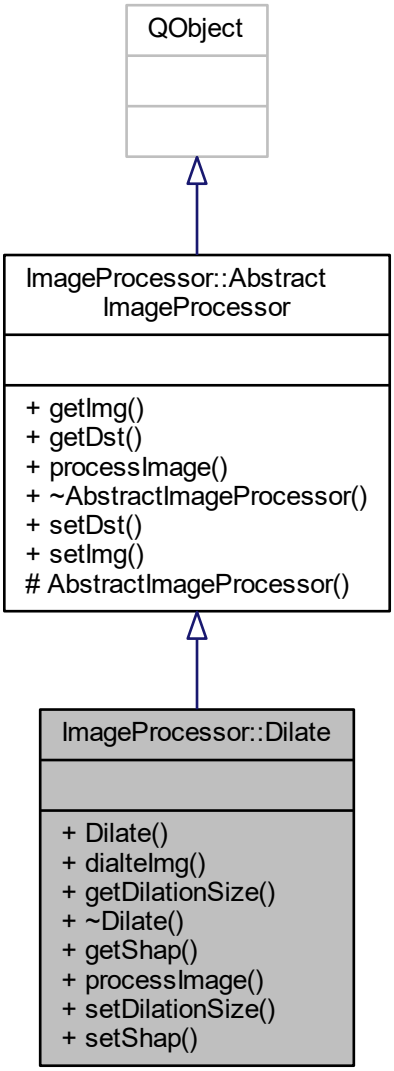
The documentation for this class was generated from the following files:

- object-detector/src/DevicesInterfaces/DeviceHandler/devicemanager.h
- object-detector/src/DevicesInterfaces/DeviceHandler/devicemanager.cpp

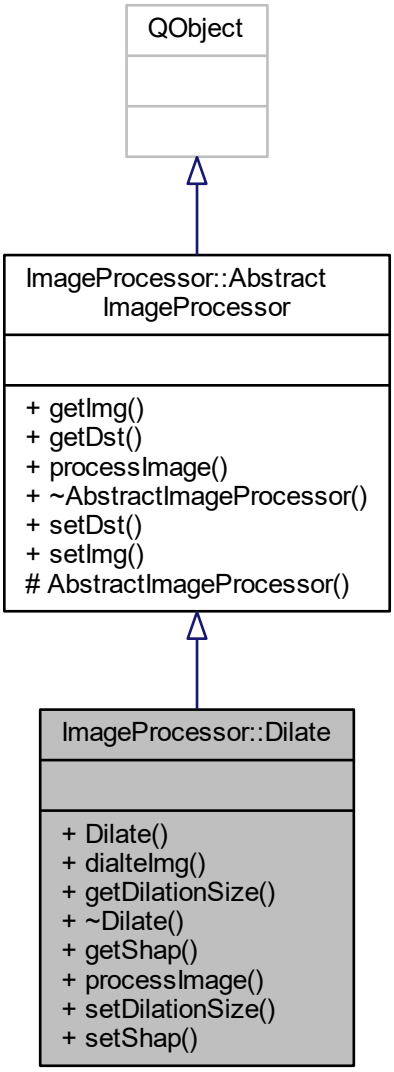
7.12 ImageProcessor::Dilate Class Reference

this Class is used to perform morphological dilate operation on image see Morphological Operation.

Inheritance diagram for ImageProcessor::Dilate:



Collaboration diagram for ImageProcessor::Dilate:



Classes

- class **_DilateImpl**

Public Slots

- void setDilationSize (int value)

  sets the Dilation Size of the Morphological Operation
- void setShap (const cv::MorphShapes &value)

  sets The Shape of the Dialtion pixels

Public Member Functions

- **Dilate** (QObject *parent=nullptr)
- void **dialteImg** ()
- int getDilationSize () const

  returns the dilation Size
- cv::MorphShapes **getShap** () const
- virtual QVariant processImage () override

  Pure Virtual Function representes the operation to be done on the Image to be processed.

Additional Inherited Members

7.12.1 Detailed Description

this Class is used to perform morphological dilate operation on image see Morphological Operation.

Note

the image must be binary (black and white) and in grayscale.

```
 ImageProcessor::Dialte dial{this};
dial.setImg(cv::imread(BINARY_IMG_PATH));
dial.processImage();
auto dst = dial.getDst();
cv::imshow("window", dst);
cv::waitKey(0);
```

Todo add Other Morphological Operations like erode.

Definition at line 13 of file dilate.h.

7.12.2 Member Function Documentation

7.12.2.1 getDilationSize()

```
int Dilate::getDilationSize ( ) const
```

returns the dilation Size

Returns

Definition at line 13 of file dilate.cpp.

7.12.2.2 processImage()

```
QVariant Dilate::processImage ( )  [override], [virtual]
```

Pure Virtual Function representes the operation to be done on the Image to be processed.

Exceptions

cv::Exception.not	garunteed to throw this exception

Warning

not exception nor thread safe.

Returns

QVariant Object which represents the output of the processing operation and it doesn't have to be cv::Mat.

Implements ImageProcessor::AbstractImageProcessor.

Definition at line 46 of file dilate.cpp.

7.12.2.3 setDilationSize

```
void Dilate::setDilationSize (
            int value )  [slot]
```

sets the Dilation Size of the Morphological Operation

Definition at line 26 of file dilate.cpp.

7.12.2.4 setShap

```
void Dilate::setShap (
            const cv::MorphShapes & value )  [slot]
```

sets The Shape of the Dialtion pixels

Parameters

enum	of cv::MorphShapes.

See also

ImageProcessor::Dilate.

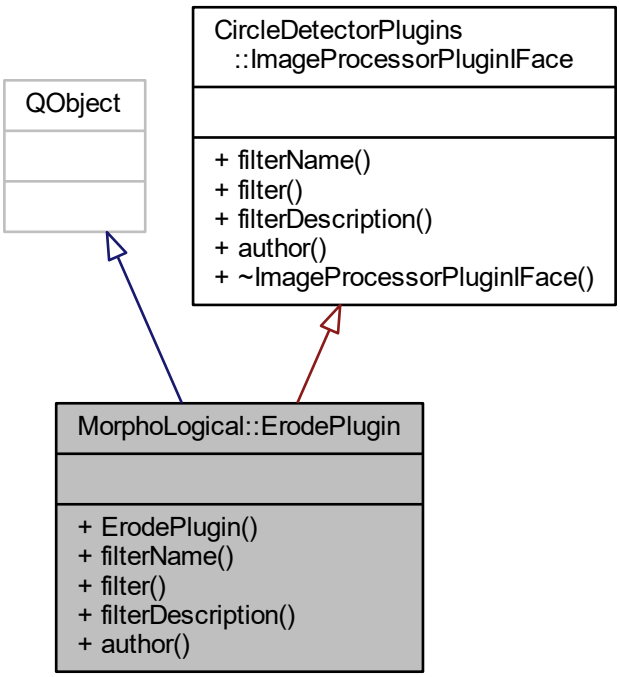
Definition at line 41 of file dilate.cpp.

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

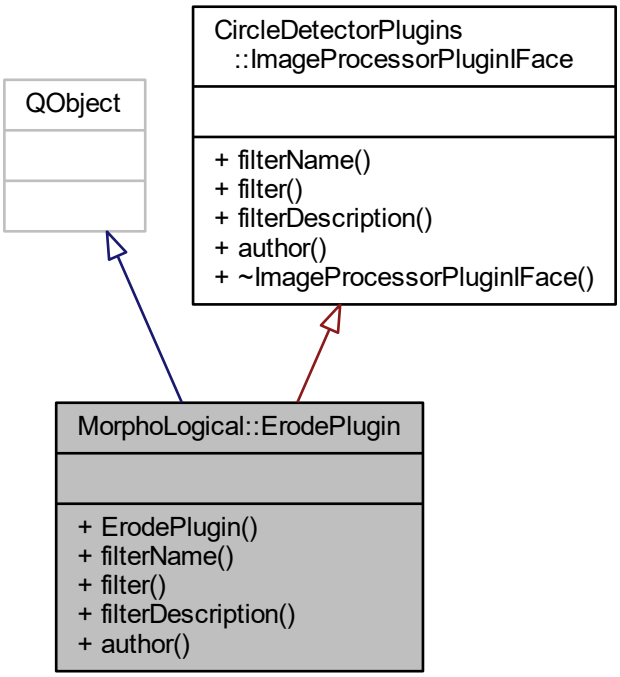
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/dilate.h
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/dilate.cpp

7.13 MorphoLogical::ErodePlugin Class Reference

Inheritance diagram for MorphoLogical::ErodePlugin:



Collaboration diagram for MorphoLogical::ErodePlugin:



Public Member Functions

- **ErodePlugin** (QObject *parent=0)
- QString filterName () const

  filterName used to describe the filter name
- cv::Mat filter (cv::Mat src) const

  filter this member function is used to apply a filter on an image
- virtual QString filterDescription () const

  filterDescription used to describe filter description
- virtual QString author () const

  author used to describe the author of the plugin

7.13.1 Detailed Description

Definition at line 7 of file erodeplugin.h.

7.13.2 Member Function Documentation

7.13.2.1 author()

```
QString MorphoLogical::ErodePlugin::author ( ) const  [virtual]
```

author used to describe the author of the plugin

Returns

Implements CircleDetectorPlugins::ImageProcessorPluginIFace.

Definition at line 28 of file erodeplugin.cpp.

7.13.2.2 filter()

```
cv::Mat MorphoLogical::ErodePlugin::filter (
            cv::Mat src ) const  [virtual]
```

filter this member function is used to apply a filter on an image

Parameters

src	a cv::Mat object of the source image

Returns

cv::Mat object of the image after applying filter

Implements CircleDetectorPlugins::ImageProcessorPluginIFace.

Definition at line 14 of file erodeplugin.cpp.

7.13.2.3 filterDescription()

```
QString MorphoLogical::ErodePlugin::filterDescription ( ) const  [virtual]
```

filterDescription used to describe filter description

Returns

Implements CircleDetectorPlugins::ImageProcessorPluginIFace.

Definition at line 23 of file erodeplugin.cpp.

7.13.2.4 filterName()

```
QString MorphoLogical::ErodePlugin::filterName ( ) const  [virtual]
```

filterName used to describe the filter name

Returns

QString object of the filter name for metadata

Implements CircleDetectorPlugins::ImageProcessorPluginIFace.

Definition at line 9 of file erodeplugin.cpp.

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

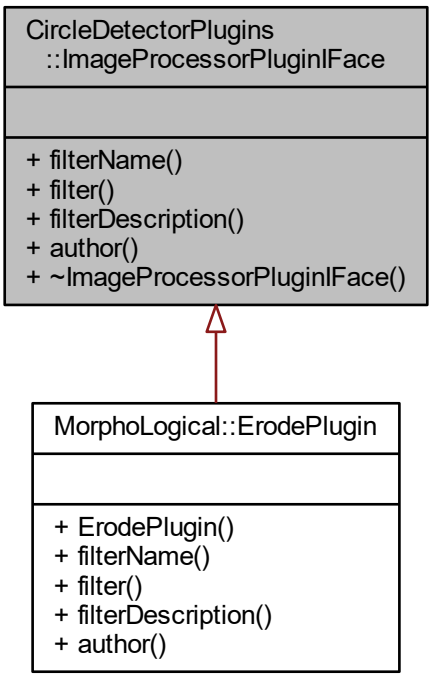
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/ErodePlugin/erodeplugin.h
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/ErodePlugin/erodeplugin.cpp

7.14 CircleDetectorPlugins::ImageProcessorPluginIFace Class Reference

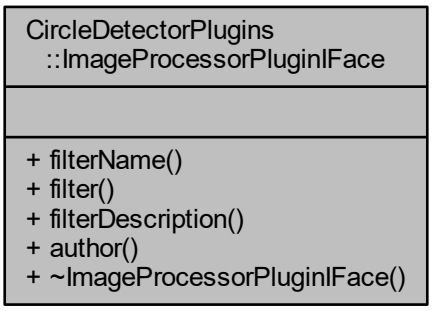
The ImageProcessorPluginIFace is and interface used to apply filters to Images.

```
#include <imageprocessorpluginiface.h>
```

Inheritance diagram for CircleDetectorPlugins::ImageProcessorPluginIFace:



Collaboration diagram for CircleDetectorPlugins::ImageProcessorPluginIFace:



Public Member Functions

- virtual QString filterName () const =0

  filterName used to describe the filter name
- virtual cv::Mat filter (cv::Mat src) const =0

  filter this member function is used to apply a filter on an image
- virtual QString filterDescription () const =0

  filterDescription used to describe filter description
- virtual QString author () const =0

  author used to describe the author of the plugin

7.14.1 Detailed Description

The ImageProcessorPluginIFace is and interface used to apply filters to Images.

this interface is used to implement plugins to apply filters on image the filter is compiled as a plugin for Qt in qmake file

```
CONFIG += release plugin
```

to implement the plugin you must implement both the QObject class and the interface for exmaple

```
 class ErodePlugin : public QObject,
      CircleDetectorPlugins::ImageProcessorPluginIFace
{
   Q_OBJECT
   Q_INTERFACES(CircleDetectorPlugins::ImageProcessorPluginIFace
      )
   Q_PLUGIN_METADATA(IID ErodePlugin_iid)
  // implement virtual members
 }
```

Author

Mohamed Khaled

Version

1.0

Definition at line 38 of file imageprocessorpluginiface.h.

7.14.2 Member Function Documentation

7.14.2.1 author()

```
virtual QString CircleDetectorPlugins::ImageProcessorPluginIFace::author ( ) const  [pure
virtual]
```

author used to describe the author of the plugin

Returns

Implemented in MorphoLogical::ErodePlugin.

7.14.2.2 filter()

```
virtual cv::Mat CircleDetectorPlugins::ImageProcessorPluginIFace::filter (
            cv::Mat src ) const  [pure virtual]
```

filter this member function is used to apply a filter on an image

Parameters

src	a cv::Mat object of the source image

Returns

cv::Mat object of the image after applying filter

Implemented in MorphoLogical::ErodePlugin.

7.14.2.3 filterDescription()

```
virtual QString CircleDetectorPlugins::ImageProcessorPluginIFace::filterDescription ( ) const
[pure virtual]
```

filterDescription used to describe filter description

Returns

Implemented in MorphoLogical::ErodePlugin.

7.14.2.4 filterName()

```
virtual QString CircleDetectorPlugins::ImageProcessorPluginIFace::filterName ( ) const  [pure
virtual]
```

filterName used to describe the filter name

Returns

QString object of the filter name for metadata

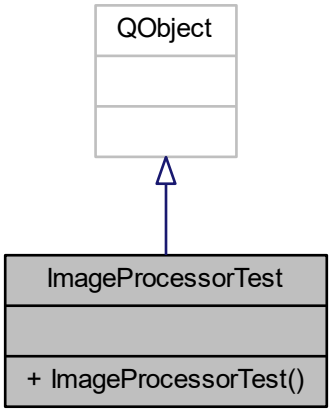
Implemented in MorphoLogical::ErodePlugin.

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

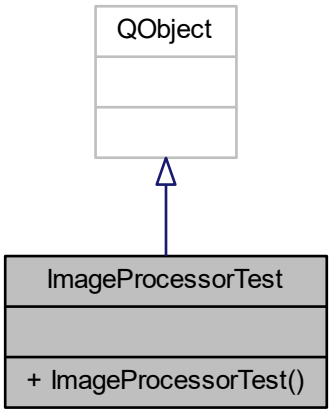
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/ErodePlugin/imageprocessorpluginiface.h

7.15 ImageProcessorTest Class Reference

Inheritance diagram for ImageProcessorTest:



Collaboration diagram for ImageProcessorTest:



7.15.1 Detailed Description

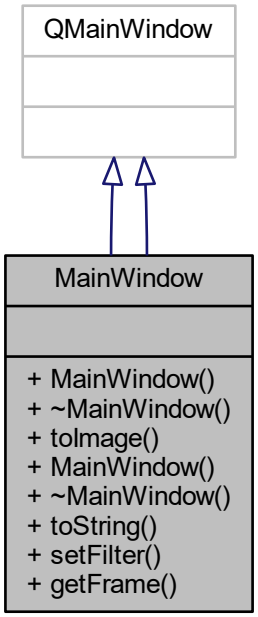
Definition at line 8 of file tst_imageprocessortest.cpp.

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

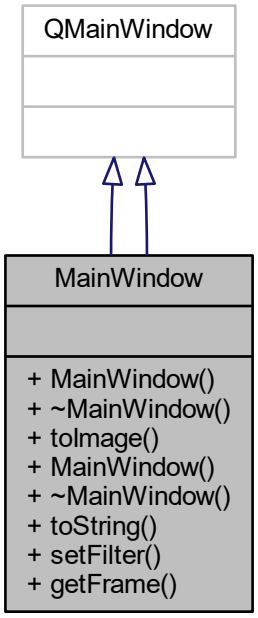
- object-detector/src/CircleDetector/Tests/ImageProcessorTest/tst_imageprocessortest.cpp

7.16 MainWindow Class Reference

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Public Slots

- void **setFilter** (PluginSharedPointer filter)
- void **getFrame** (cv::Mat frame)

Public Member Functions

- **MainWindow** (QWidget ∗parent=0)
- QImage **toImage** (const cv::Mat &m)
- **MainWindow** (QWidget ∗parent=0)
- QString **toString** (int x)

7.16.1 Detailed Description

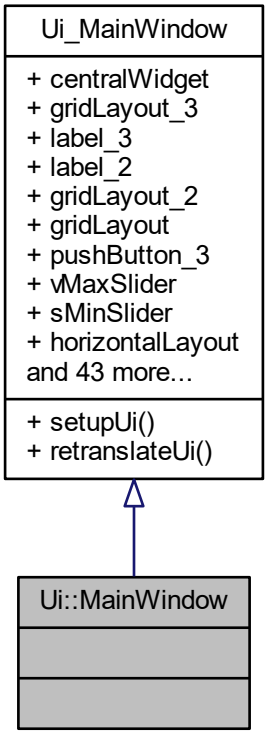
Definition at line 13 of file mainwindow.h.

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

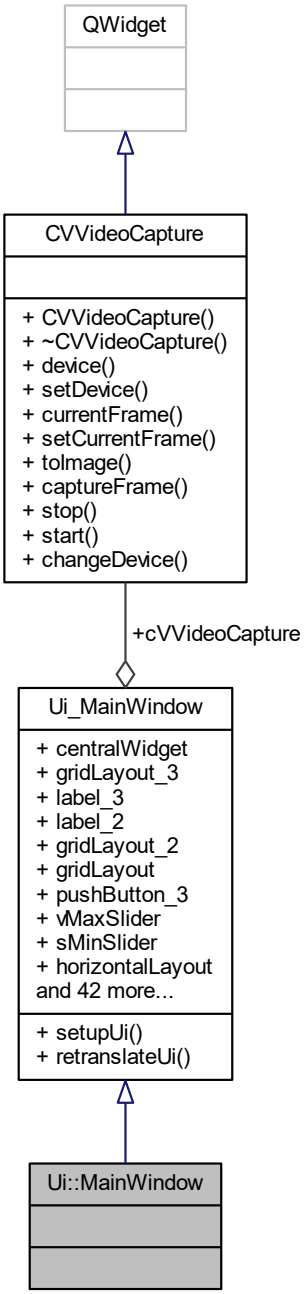
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/PluginLoader/Testing/mainwindow.h
- object-detector/src/Utilities/Plugins/CircleDetectorPlugins/PluginLoader/Testing/mainwindow.cpp

7.17 Ui::MainWindow Class Reference

Inheritance diagram for Ui::MainWindow:



Collaboration diagram for Ui::MainWindow:



Additional Inherited Members

7.17.1 Detailed Description

Definition at line 439 of file ui_mainwindow.h.

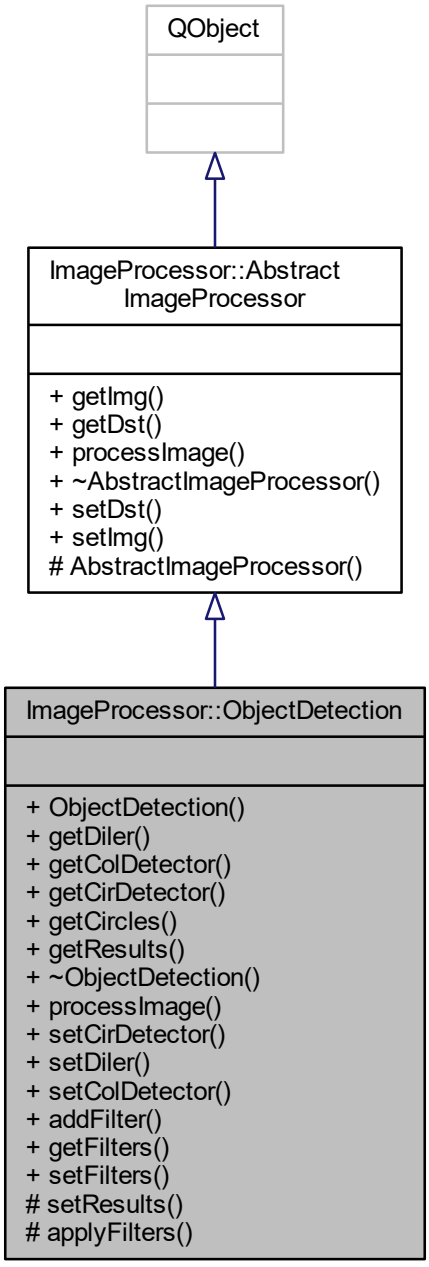
The documentation for this class was generated from the following file:

- object-detector/src/View/DesktopView/View/ui_mainwindow.h

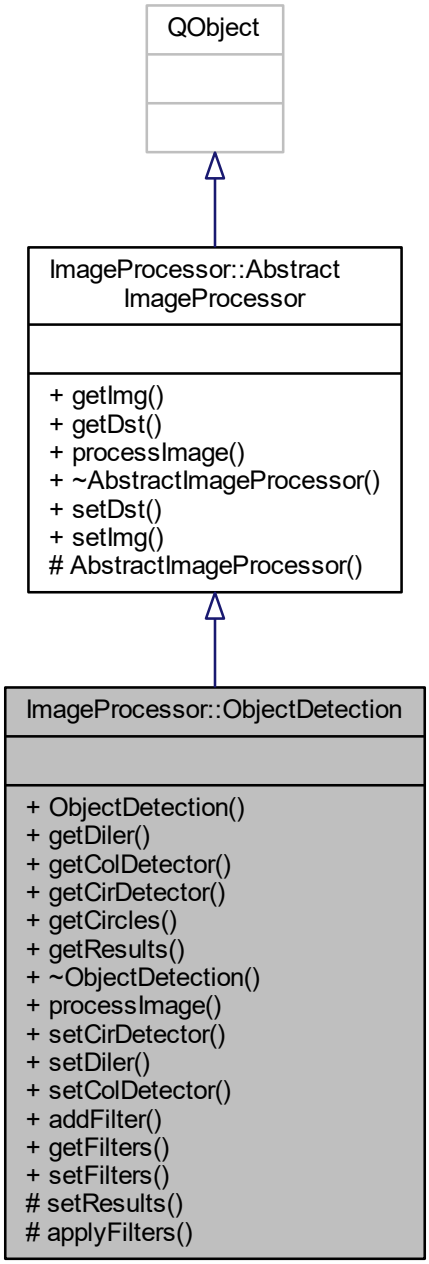
7.18 ImageProcessor::ObjectDetection Class Reference

this class is used to detect a a colored circle object(s)

Inheritance diagram for ImageProcessor::ObjectDetection:



Collaboration diagram for ImageProcessor::ObjectDetection:



Classes

- class **_ObjectDetectionImpl**

Public Slots

- void setCirDetector (DetectCircle *cirDetector)

ObjectDetection::setCirDetector sets the circleDetector.

- void setDiler (Dilate *diler)

  ObjectDetection::setDiler sets the dilate object.
- void **setColDetector** (DetectColor *colDetector)
- void addFilter (PluginSharedPointer proc)

  ObjectDetection::addFilter adds a filter from a plugin.
- std::vector< PluginSharedPointer > **getFilters** () const
- void **setFilters** (const std::vector< PluginSharedPointer > &value)

Public Member Functions

- ObjectDetection (QObject *parent=nullptr)

  ObjectDetection::ObjectDetection.
- Dilate * getDiler () const

  ObjectDetection::getDiler.
- DetectColor * getColDetector () const

  ObjectDetection::getColDetector.
- DetectCircle * getCirDetector () const

  ObjectDetection::getCirDetector.
- std::vector< cv::Vec3f > getCircles ()

  ObjectDetection::getCircles.
- QVariant **getResults** () const
- virtual QVariant processImage () override

  Pure Virtual Function representes the operation to be done on the Image to be processed.

Protected Member Functions

- void **setResults** (QVariant res)
- cv::Mat applyFilters (cv::Mat dst) const

  ObjectDetection::applyFilters applies filters to the image.

Additional Inherited Members

7.18.1 Detailed Description

this class is used to detect a a colored circle object(s)

the class first thresholds the image to get the colored object using cv::threshold algorthim so you must supply a max and min colors as cv::Scalar using ImageProcessor::DetectColor after detecting color it applies a Morphological operation (dilation) using cv::Dilate algorthim and guassian blur to decrease the noise using cv::GuassianBlur using ImageProcssor::Dilate then applies filters using Plugins CircleDetectorPlugins::ImageProcessorPluginIFace then detect Circles using cv::HoughCircles ImageProcessor::DetectCircle

```
 ImageProcessor::ObjectDetection objdet;
 objdet.setImg(cv::imread(IMG_PATH));
 auto var = objdet.processImage();
 auto vec = var.value<std::vector<cv::Vec3f>>(); //or use var.getResults();
for(auto i : vec){
     cout << "x: " << i[0] << " y:" << i[1] << " radius: " << i[2] << endl;
}
```

Author

Mohamed Khaled

Version

5.0

Definition at line 17 of file objectdetection.h.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 ObjectDetection()

```
ObjectDetection::ObjectDetection (
            QObject * parent = nullptr )  [explicit]
```

ObjectDetection::ObjectDetection.

Parameters

parent	

Definition at line 9 of file objectdetection.cpp.

7.18.3 Member Function Documentation

7.18.3.1 addFilter

```
void ObjectDetection::addFilter (
            PluginSharedPointer proc )  [slot]
```

ObjectDetection::addFilter adds a filter from a plugin.

See also

CircleDetectorPlugins::ImageProcessorPluginIFace

Parameters

proc	

Definition at line 73 of file objectdetection.cpp.

7.18.3.2 applyFilters()

```
Mat ObjectDetection::applyFilters (
            cv::Mat dst ) const  [protected]
```

ObjectDetection::applyFilters applies filters to the image.

Exceptions

cv::Excpetion	

Warning

neither exception safe nor thread safe

Parameters

dst	

Returns

Definition at line 104 of file objectdetection.cpp.

7.18.3.3 getCircles()

```
std::vector< Vec3f > ObjectDetection::getCircles ( )
```

ObjectDetection::getCircles.

Returns

returns the std::vector<cv::Vec3f> which encapsulate the data about circles like x,y and radius

Definition at line 42 of file objectdetection.cpp.

7.18.3.4 getCirDetector()

```
DetectCircle * ObjectDetection::getCirDetector ( ) const
```

ObjectDetection::getCirDetector.

Returns

a pointer to ImageProcessor::DetectCircle used to to detect circles

Definition at line 34 of file objectdetection.cpp.

7.18.3.5 getColDetector()

```
DetectColor * ObjectDetection::getColDetector ( ) const
```

ObjectDetection::getColDetector.

Returns

a pointer to the ImageProcessor::DetectColor used to detect colors

Definition at line 26 of file objectdetection.cpp.

7.18.3.6 getDiler()

```
Dilate * ObjectDetection::getDiler ( ) const
```

ObjectDetection::getDiler.

Returns

an ImageProcessor::Dilate pointer to object

Definition at line 17 of file objectdetection.cpp.

7.18.3.7 processImage()

```
QVariant ObjectDetection::processImage ( )  [override], [virtual]
```

Pure Virtual Function representes the operation to be done on the Image to be processed.

Exceptions

cv::Exception.not	garunteed to throw this exception

Warning

not exception nor thread safe.

Returns

QVariant Object which represents the output of the processing operation and it doesn't have to be cv::Mat.

Implements ImageProcessor::AbstractImageProcessor.

Definition at line 113 of file objectdetection.cpp.

7.18.3.8 setCirDetector

```
void ObjectDetection::setCirDetector (
            DetectCircle * cirDetector )  [slot]
```

ObjectDetection::setCirDetector sets the circleDetector.

Parameters

cirDetector	

Definition at line 51 of file objectdetection.cpp.

7.18.3.9 setDiler

```
void ObjectDetection::setDiler (
            Dilate * diler )  [slot]
```

ObjectDetection::setDiler sets the dilate object.

Parameters

diler	

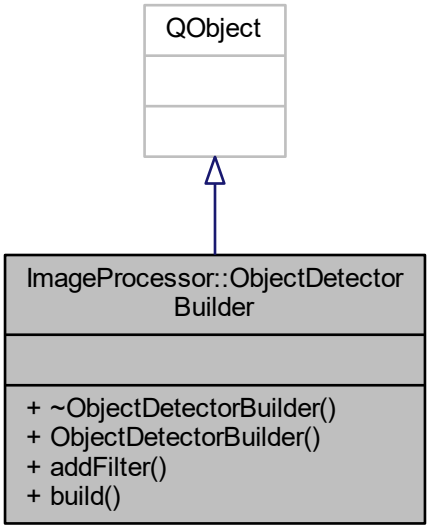
Definition at line 59 of file objectdetection.cpp.

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

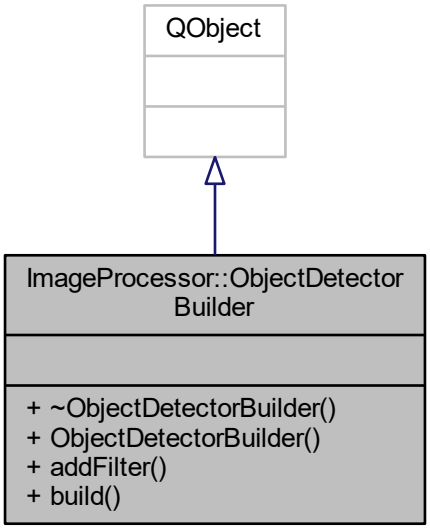
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/objectdetection.h
- object-detector/src/CircleDetector/ImageProcessors/ImageProcessor/objectdetection.cpp

7.19 ImageProcessor::ObjectDetectorBuilder Class Reference

Inheritance diagram for ImageProcessor::ObjectDetectorBuilder:



Collaboration diagram for ImageProcessor::ObjectDetectorBuilder:



Classes

- class **_ObjectDetectorBuilderImpl**

Public Member Functions

- **ObjectDetectorBuilder** (QObject ∗parent=nullptr)
- void **addFilter** (PluginSharedPointer proc)
- std::unique_ptr< ObjectDetection > **build** ()

7.19.1 Detailed Description

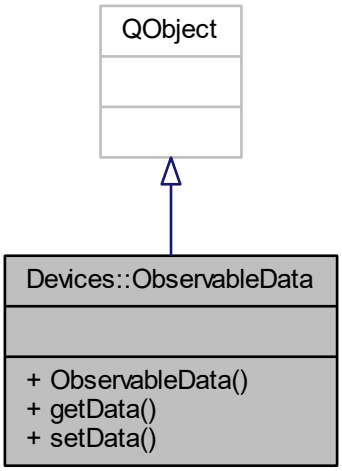
Definition at line 12 of file objectdetectorbuilder.h.

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

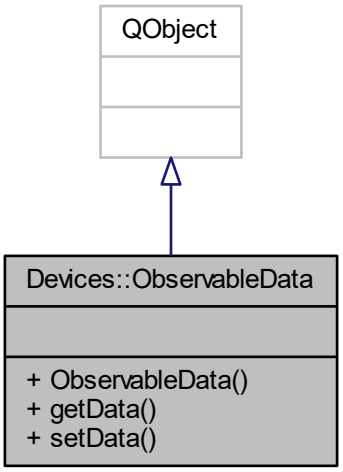
- object-detector/src/CircleDetector/ImageProcessors/objectdetectorbuilder.h
- object-detector/src/CircleDetector/ImageProcessors/objectdetectorbuilder.cpp

7.20 Devices::ObservableData Class Reference

Inheritance diagram for Devices::ObservableData:



Collaboration diagram for Devices::ObservableData:



Public Member Functions

- **ObservableData** (QObject ∗parent=nullptr)
- QString **getData** () const
- void **setData** (const QString &value)

7.20.1 Detailed Description

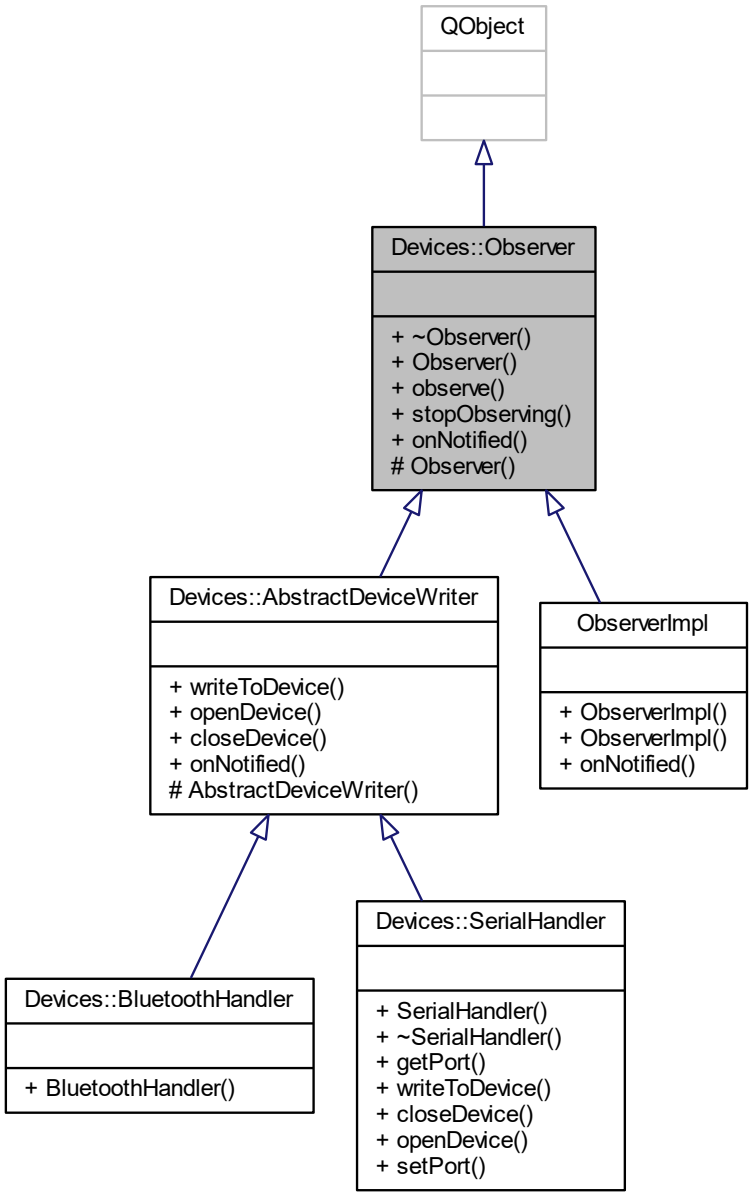
Definition at line 11 of file observabledata.h.

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

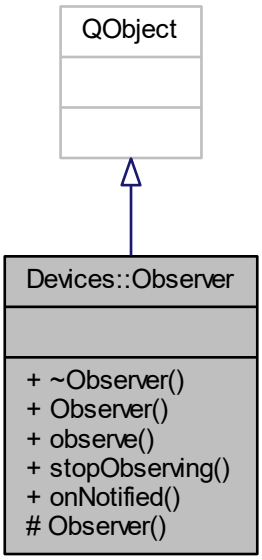
- object-detector/src/DevicesInterfaces/DeviceHandler/observabledata.h
- object-detector/src/DevicesInterfaces/DeviceHandler/observabledata.cpp

7.21 Devices::Observer Class Reference

Inheritance diagram for Devices::Observer:



Collaboration diagram for Devices::Observer:



Classes

- class **_ObserverImpl**

Public Slots

- void **observe** (Subject ∗sub)
- void **stopObserving** (Subject ∗sub)
- virtual void **onNotified** (const ObservableData &dt)=0

Public Member Functions

- **Observer** (const Observer &)=default

Protected Member Functions

- **Observer** (QObject ∗parent=nullptr)

7.21.1 Detailed Description

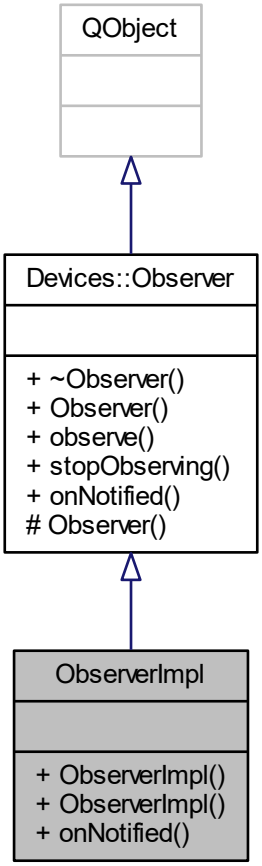
Definition at line 15 of file observer.h.

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

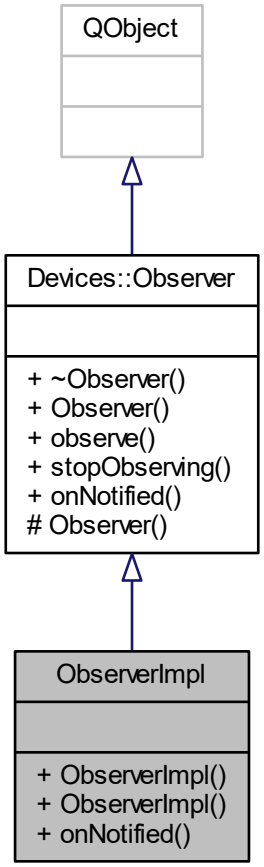
- object-detector/src/DevicesInterfaces/DeviceHandler/observer.h
- object-detector/src/DevicesInterfaces/DeviceHandler/observer.cpp

7.22 ObserverImpl Class Reference

Inheritance diagram for ObserverImpl:



Collaboration diagram for ObserverImpl:



Public Slots

- virtual void **onNotified** (const ObservableData &dt) override

Public Member Functions

- **ObserverImpl** (QObject *parent)
- **ObserverImpl** (const ObserverImpl &)=default

Additional Inherited Members

7.22.1 Detailed Description

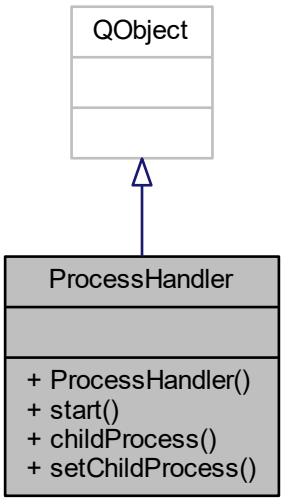
Definition at line 7 of file observerimpl.h.

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

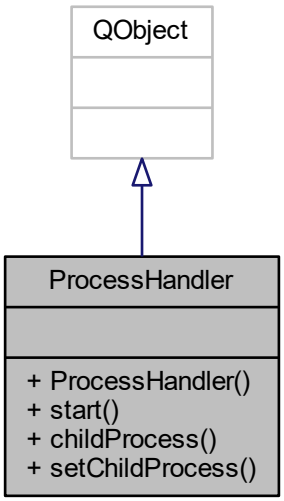
- object-detector/src/DevicesInterfaces/Tests/TestObserverSubject/observerimpl.h
- object-detector/src/DevicesInterfaces/Tests/TestObserverSubject/observerimpl.cpp

7.23 ProcessHandler Class Reference

Inheritance diagram for ProcessHandler:



Collaboration diagram for ProcessHandler:



Public Slots

- void **setChildProcess** (QProcess ∗childProcess)

Signals

- void **started** ()

Public Member Functions

- **ProcessHandler** (QObject ∗parent=nullptr)
- void **start** (QString cmd, QStringList lst)
- QProcess ∗ **childProcess** () const

7.23.1 Detailed Description

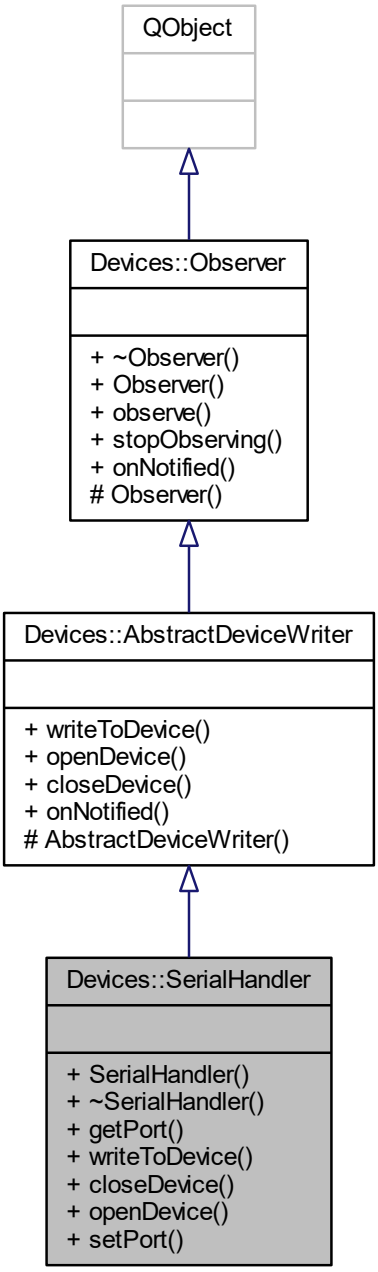
Definition at line 7 of file processhandler.h.

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

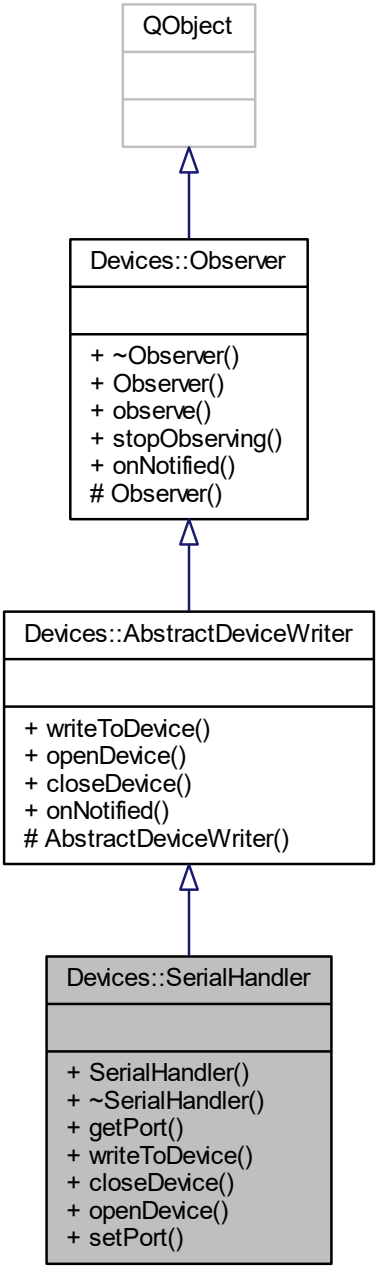
- object-detector/src/Utilities/Tools/SerialMonitorTool/SerialMonitor/processhandler.h
- object-detector/src/Utilities/Tools/SerialMonitorTool/SerialMonitor/processhandler.cpp

7.24 Devices::SerialHandler Class Reference

Inheritance diagram for Devices::SerialHandler:



Collaboration diagram for Devices::SerialHandler:



Classes

- class **_SerialHandlerImpl**

Public Slots

- void **setPort** (QSerialPort *port)

Public Member Functions

- **SerialHandler** (QObject ∗parent=nullptr)
- QSerialPort ∗ **getPort** ()
- virtual void **writeToDevice** (const QString &str) override
- virtual void **closeDevice** () override
- virtual void **openDevice** () override

Additional Inherited Members

7.24.1 Detailed Description

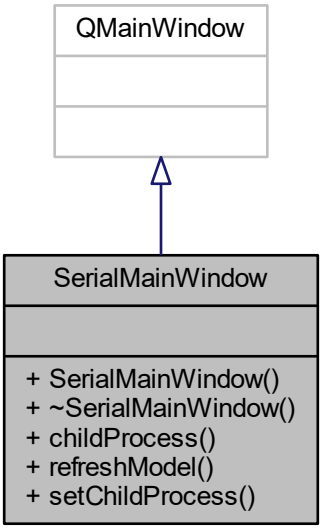
Definition at line 14 of file serialhandler.h.

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

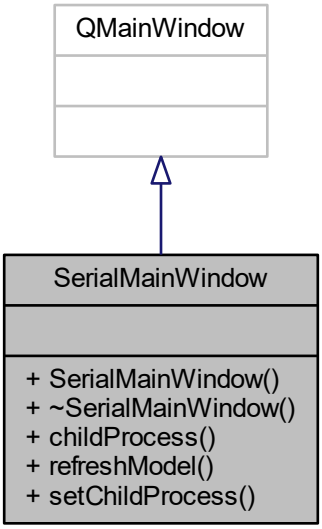
- object-detector/src/DevicesInterfaces/DeviceHandler/Serial/serialhandler.h
- object-detector/src/DevicesInterfaces/DeviceHandler/Serial/serialhandler.cpp

7.25 SerialMainWindow Class Reference

Inheritance diagram for SerialMainWindow:



Collaboration diagram for SerialMainWindow:



Public Slots

- void **refreshModel** ()
- void **setChildProcess** (QProcess ∗childProcess)

Public Member Functions

- **SerialMainWindow** (QWidget ∗parent=0)
- QProcess ∗ **childProcess** () const

7.25.1 Detailed Description

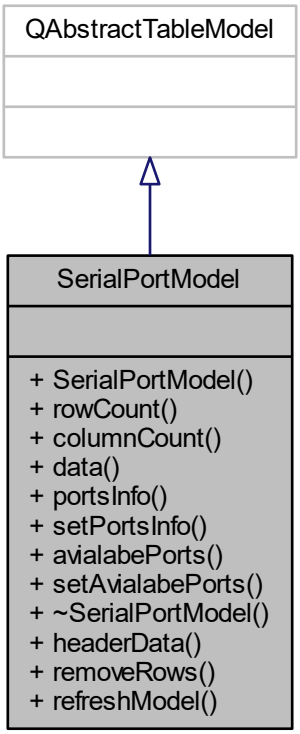
Definition at line 13 of file serialmainwindow.h.

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

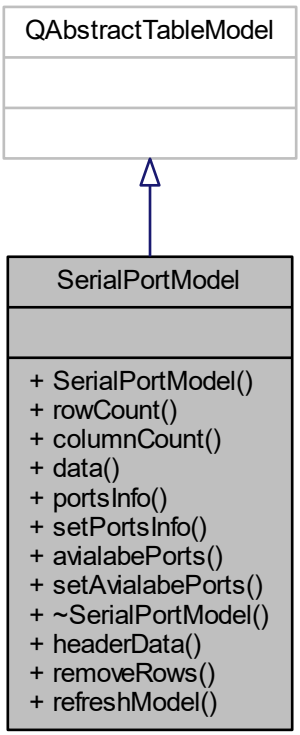
- object-detector/src/Utilities/Tools/SerialMonitorTool/SerialMonitor/serialmainwindow.h
- object-detector/src/Utilities/Tools/SerialMonitorTool/SerialMonitor/serialmainwindow.cpp

7.26 SerialPortModel Class Reference

Inheritance diagram for SerialPortModel:



Collaboration diagram for SerialPortModel:



Public Types

- enum **type** {
  PORT = Qt::UserRole + 1, **PORT_NAME**, **PORT_DESCRIPTION**, **PORT_SYSTEM_LOCATION**, **MANUFACTURER**, **SERIAL_NUM** }

Public Slots

- void **refreshModel** ()

Public Member Functions

- **SerialPortModel** (QObject ∗parent=nullptr)
- virtual int **rowCount** (const QModelIndex &parent) const override
- virtual int **columnCount** (const QModelIndex &parent) const override
- virtual QVariant **data** (const QModelIndex &index, int role) const override
- QList< QSerialPortInfo > **portsInfo** () const
- void **setPortsInfo** (const QList< QSerialPortInfo > &portsInfo)
- QList< QSerialPort ∗ > **avialabePorts** () const
- void **setAvialabePorts** (const QList< QSerialPort ∗> &avialabePorts)
- virtual QVariant **headerData** (int section, Qt::Orientation orientation, int role) const override
- virtual bool **removeRows** (int row, int count, const QModelIndex &parent) override

7.26.1 Detailed Description

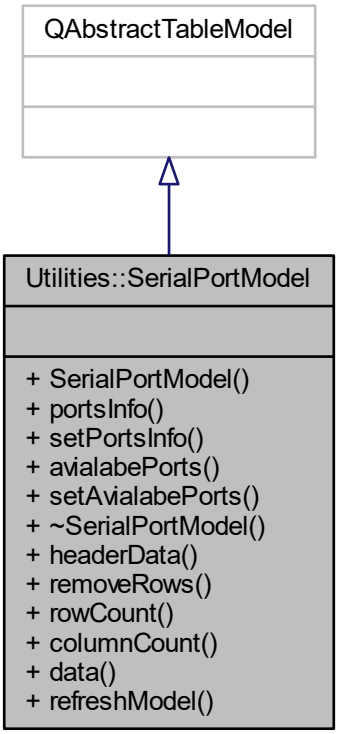
Definition at line 9 of file serialportmodel.h.

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

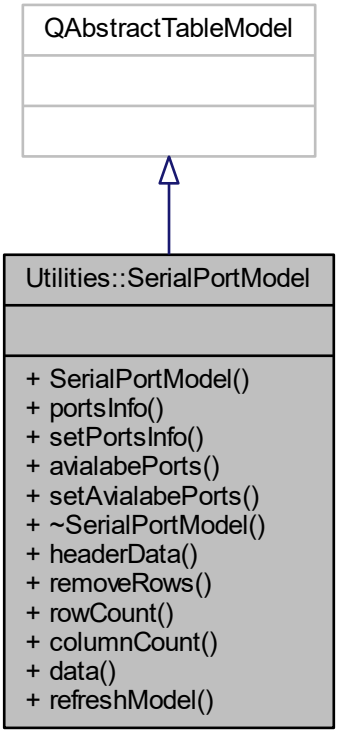
- object-detector/src/Utilities/Tools/SerialMonitorTool/SerialMonitor/serialportmodel.h
- object-detector/src/Utilities/Tools/SerialMonitorTool/SerialMonitor/serialportmodel.cpp

7.27 Utilities::SerialPortModel Class Reference

Inheritance diagram for Utilities::SerialPortModel:



Collaboration diagram for Utilities::SerialPortModel:



Public Types

- enum **type** {
  PORT = Qt::UserRole + 1, **PORT_INFO**, **PORT_NAME**, **PORT_DESCRIPTION**,
  PORT_SYSTEM_LOCATION, **MANUFACTURER**, **SERIAL_NUM** }

Public Slots

- void **refreshModel** ()

Public Member Functions

- **SerialPortModel** (QObject ∗parent=nullptr)
- QList< QSerialPortInfo > **portsInfo** () const
- void **setPortsInfo** (const QList< QSerialPortInfo > &portsInfo)
- QList< QSerialPort ∗ > **avialabePorts** () const
- void **setAvialabePorts** (const QList< QSerialPort ∗> &avialabePorts)
- virtual QVariant **headerData** (int section, Qt::Orientation orientation, int role) const override
- virtual bool **removeRows** (int row, int count, const QModelIndex &parent) override
- virtual int **rowCount** (const QModelIndex &parent) const override
- virtual int **columnCount** (const QModelIndex &parent) const override
- virtual QVariant **data** (const QModelIndex &index, int role) const override

7.27.1 Detailed Description

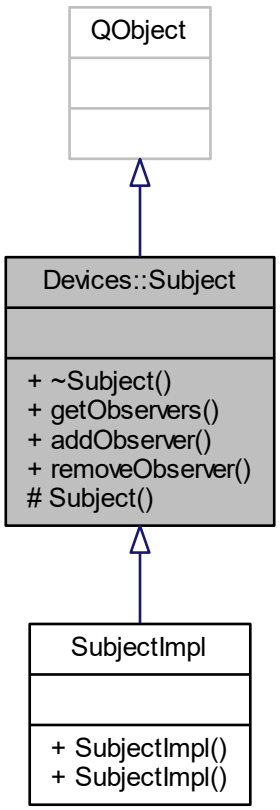
Definition at line 16 of file serialportmodel.h.

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

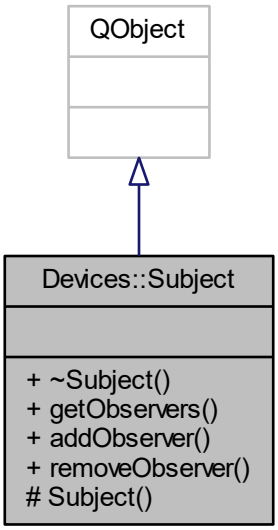
- object-detector/src/Utilities/SerialPortModel/serialportmodel.h
- object-detector/src/Utilities/SerialPortModel/serialportmodel.cpp

7.28 Devices::Subject Class Reference

Inheritance diagram for Devices::Subject:



Collaboration diagram for Devices::Subject:



Classes

- class **_SubjectImpl**

Public Slots

- void **addObserver** (Observer *obs)
- void **removeObserver** (Observer *obs)

Signals

- void **notifyObservers** (const ObservableData &)

Public Member Functions

- std::vector< Observer * > **getObservers** () const

Protected Member Functions

- **Subject** (QObject *parent=nullptr)

7.28.1 Detailed Description

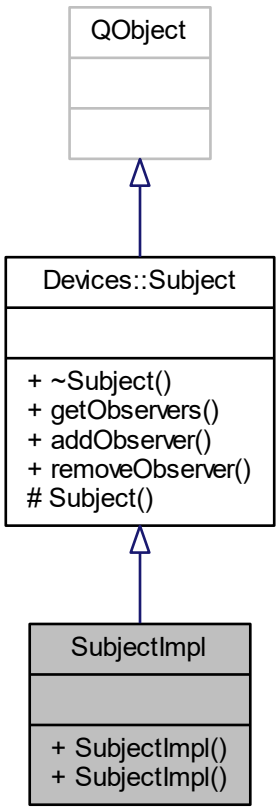
Definition at line 18 of file subject.h.

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

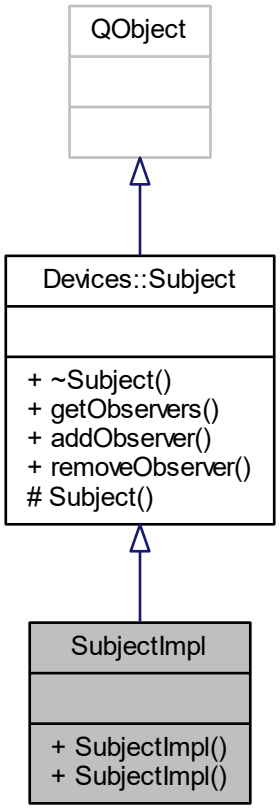
- object-detector/src/DevicesInterfaces/DeviceHandler/subject.h
- object-detector/src/DevicesInterfaces/DeviceHandler/subject.cpp

7.29 SubjectImpl Class Reference

Inheritance diagram for SubjectImpl:



Collaboration diagram for SubjectImpl:



Public Member Functions

- **SubjectImpl** (QObject ∗parent=nullptr)
- **SubjectImpl** (const SubjectImpl &)=default

Additional Inherited Members

7.29.1 Detailed Description

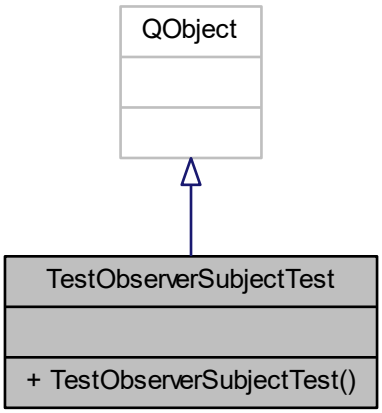
Definition at line 5 of file subjectimpl.h.

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

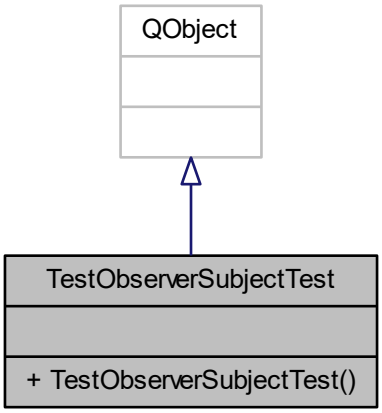
- object-detector/src/DevicesInterfaces/Tests/TestObserverSubject/subjectimpl.h
- object-detector/src/DevicesInterfaces/Tests/TestObserverSubject/subjectimpl.cpp

7.30 TestObserverSubjectTest Class Reference

Inheritance diagram for TestObserverSubjectTest:



Collaboration diagram for TestObserverSubjectTest:



7.30.1 Detailed Description

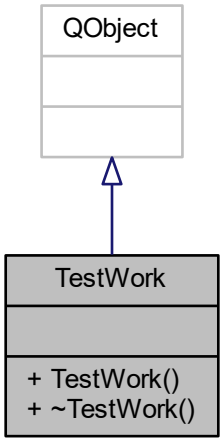
Definition at line 11 of file tst_testobserversubjecttest.cpp.

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

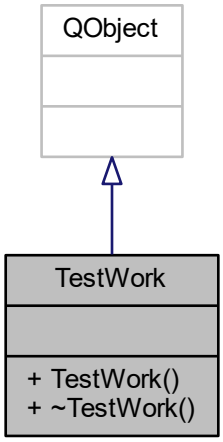
- object-detector/src/DevicesInterfaces/Tests/TestObserverSubject/tst_testobserversubjecttest.cpp

7.31 TestWork Class Reference

Inheritance diagram for TestWork:



Collaboration diagram for TestWork:



7.31.1 Detailed Description

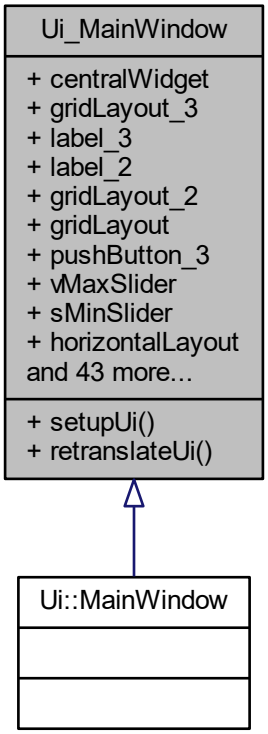
Definition at line 6 of file tst_testwork.cpp.

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

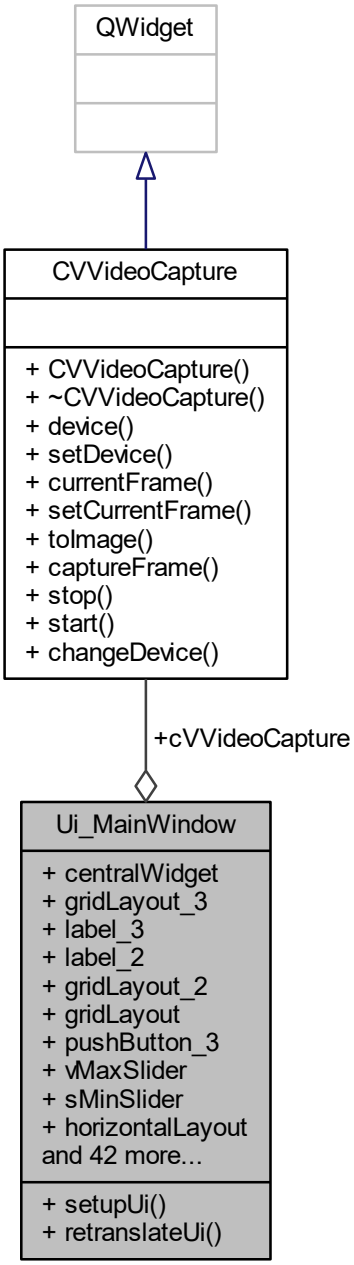
- object-detector/src/CircleDetector/Tests/DetectColorTest/tst_testwork.cpp

7.32 Ui_MainWindow Class Reference

Inheritance diagram for Ui_MainWindow:



Collaboration diagram for Ui_MainWindow:



Public Member Functions

- void **setupUi** (QMainWindow ∗MainWindow)
- void **retranslateUi** (QMainWindow ∗MainWindow)

Public Attributes

- QWidget ∗ **centralWidget**

- QGridLayout * **gridLayout_3**
- QLabel * **label_3**
- QLabel * **label_2**
- QGridLayout * **gridLayout_2**
- QGridLayout * **gridLayout**
- QPushButton * **pushButton_3**
- QSlider * **vMaxSlider**
- QSlider * **sMinSlider**
- QHBoxLayout * **horizontalLayout**
- QLabel * **radiusLabel**
- QLabel * **yLabelCenter**
- QLabel * **xCenterLabel**
- QGroupBox * **groupBox_2**
- QVBoxLayout * **verticalLayout_2**
- QSlider * **minDistSlider**
- QSlider * **param1Slider**
- QSlider * **param2Slider**
- QPushButton * **pushButton_2**
- QSlider * **circleThicknessSlider**
- QSlider * **vMinSlider**
- QSlider * **sMaxSlider**
- QSlider * **hMinSlider**
- QPushButton * **pushButton**
- QSlider * **hMaxSlider**
- QLabel * **label_4**
- QLabel * **label_6**
- QLabel * **label_7**
- QLabel * **label_5**
- QLabel * **label_8**
- QSpinBox * **spinBox_3**
- QSpinBox * **spinBox_4**
- QSpinBox * **spinBox_2**
- QSpinBox * **spinBox_5**
- QSpinBox * **spinBox_6**
- QSpinBox * **spinBox_7**
- QGroupBox * **groupBox**
- QVBoxLayout * **verticalLayout**
- QRadioButton * **radioButton**
- QRadioButton * **radioButton_2**
- QRadioButton * **radioButton_3**
- QRadioButton * **radioButton_4**
- QSlider * **dilSlider**
- QPushButton * **pushButton_6**
- QPushButton * **pushButton_5**
- QSpinBox * **spinBox**
- QVBoxLayout * **verticalLayout_3**
- QTableView * **tableView**
- QPushButton * **pushButton_4**
- CVVideoCapture * **cVVideoCapture**
- QMenuBar * **menuBar**
- QToolBar * **mainToolBar**
- QStatusBar * **statusBar**

7.32.1 Detailed Description

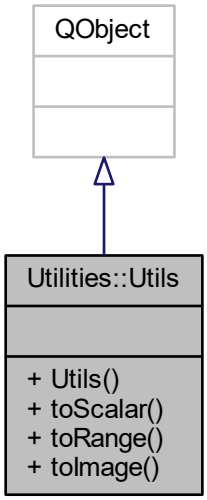
Definition at line 36 of file ui_mainwindow.h.

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

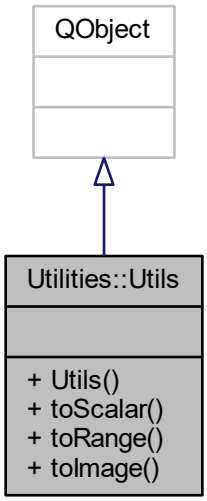
- object-detector/src/View/DesktopView/View/ui_mainwindow.h

7.33 Utilities::Utils Class Reference

Inheritance diagram for Utilities::Utils:



Collaboration diagram for Utilities::Utils:



Public Member Functions

- **Utils** (QObject ∗parent=nullptr)

Static Public Member Functions

- static cv::Scalar **toScalar** (QColor color)
- static std::tuple< cv::Scalar, cv::Scalar > **toRange** (QColor color)
- static QImage **toImage** (const cv::Mat &m)

7.33.1 Detailed Description

Definition at line 12 of file utils.h.

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

- object-detector/src/Utilities/Utils/Utilities/utils.h
- object-detector/src/Utilities/Utils/Utilities/utils.cpp

Index