

SerialReceiver

Wygenerowano przez Doxygen 1.8.11

Spis treści

1	Indeks hierarchiczny	1
1.1	Hierarchia klas	1
2	Indeks klas	3
2.1	Lista klas	3
3	Dokumentacja klas	5
3.1	Dokumentacja klasy Console	5
3.1.1	Opis szczegółowy	6
3.1.2	Dokumentacja konstruktora i destruktora	6
3.1.2.1	Console(QWidget *parent=0)	6
3.1.3	Dokumentacja funkcji składowych	6
3.1.3.1	getData	6
3.1.3.2	putData(const QByteArray &data)	6
3.2	Dokumentacja klasy MainWindow	6
3.2.1	Opis szczegółowy	7
3.2.2	Dokumentacja konstruktora i destruktora	7
3.2.2.1	MainWindow(QWidget *parent=0)	7
3.3	Dokumentacja struktury SettingsDialog::Settings	7
3.3.1	Opis szczegółowy	8
3.4	Dokumentacja klasy SettingsDialog	8
3.4.1	Opis szczegółowy	9
3.4.2	Dokumentacja konstruktora i destruktora	9
3.4.2.1	SettingsDialog(QWidget *parent=0)	9
3.4.3	Dokumentacja funkcji składowych	9
3.4.3.1	settings() const	9
	Indeks	11

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

QDialog	
SettingsDialog	8
QMainWindow	
MainWindow	6
QPlainTextEdit	
Console	5
SettingsDialog::Settings	7

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Console		
	Klasa Console	5
MainWindow		
	Klasa MainWindow	6
SettingsDialog::Settings		
	Struktura Settings	7
SettingsDialog		
	Klasa SettingsDialog	8

Rozdział 3

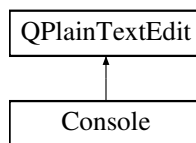
Dokumentacja klas

3.1 Dokumentacja klasy Console

Klasa `Console`.

```
#include <console.h>
```

Diagram dziedziczenia dla Console



Sygnaly

- void `getData` (const QByteArray &data)
Sygnal `getData`.

Metody publiczne

- `Console` (QWidget *parent=0)
Konstruktor.
- void `putData` (const QByteArray &data)
`putData`

Metody chronione

- virtual void `keyPressEvent` (QKeyEvent *e)
Wirtualne funkcje obecnie nieużywane.
- virtual void `mousePressEvent` (QMouseEvent *e)

3.1.1 Opis szczegółowy

Klasa [Console](#).

Klasa odpowiada za wyświetlanie przychodzących danych w oknie

3.1.2 Dokumentacja konstruktora i destruktora

3.1.2.1 `Console::Console (QWidget * parent = 0) [explicit]`

Konstruktor.

W konstruktorze ustawiane są parametry początkowe klasy

Parametry

<i>parent</i>	określa klasę nadrzędną
---------------	-------------------------

3.1.3 Dokumentacja funkcji składowych

3.1.3.1 `void Console::getData (const QByteArray & data) [signal]`

Sygnał `getData`.

Parametry

<i>data</i>	to dane wpisane z klawiatury
-------------	------------------------------

3.1.3.2 `void Console::putData (const QByteArray & data)`

`putData`

Umożliwia wypisanie danych w oknie

Parametry

<i>data</i>	to dane do wypisania
-------------	----------------------

Dokumentacja dla tej klasy została wygenerowana z pliku:

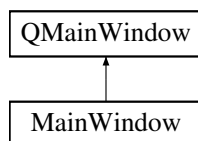
- `console.h`

3.2 Dokumentacja klasy `MainWindow`

Klasa [MainWindow](#).

```
#include <mainwindow.h>
```

Diagram dziedziczenia dla MainWindow



Metody publiczne

- [MainWindow](#) (QWidget *parent=0)
Konstruktor klasy.
- [~MainWindow](#) ()
Destruktor.

3.2.1 Opis szczegółowy

Klasa [MainWindow](#).

Klasa głównego okna aplikacji

3.2.2 Dokumentacja konstruktora i destruktora

3.2.2.1 MainWindow::MainWindow (QWidget * *parent* = 0) [explicit]

Konstruktor klasy.

Połączenie sygnałów i slotów, konfiguracja rozmieszczenia elementów

Dokumentacja dla tej klasy została wygenerowana z pliku:

- mainwindow.h

3.3 Dokumentacja struktury SettingsDialog::Settings

Struktura [Settings](#).

```
#include <settingsdialog.h>
```

Atrybuty publiczne

- QString [name](#)
Nazwa portu.
- qint32 [baudRate](#)
Szybkość transmisji.
- QString [stringBaudRate](#)
Szybkość transmisji string.
- QSerialPort::DataBits [dataBits](#)
Obiekt liczby bitów danych.
- QString [stringDataBits](#)
Liczba bitów danych.
- QSerialPort::Parity [parity](#)
Obiekt parzystości.
- QString [stringParity](#)
Nazwa parzystości.
- QSerialPort::StopBits [stopBits](#)
Obiekt bitów stopu.
- QString [stringStopBits](#)
Nazwa bitów stopu.
- QSerialPort::FlowControl [flowControl](#)
Obiekt kontroli przepływu.
- QString [stringFlowControl](#)
Nazwa kontroli przepływu.

3.3.1 Opis szczegółowy

Struktura [Settings](#).

Zawiera obiekty klasy QSerialPort i ich nazwy

Dokumentacja dla tej struktury została wygenerowana z pliku:

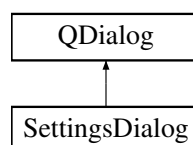
- settingsdialog.h

3.4 Dokumentacja klasy SettingsDialog

Klasa [SettingsDialog](#).

```
#include <settingsdialog.h>
```

Diagram dziedziczenia dla SettingsDialog



Komponenty

- struct [Settings](#)
Struktura [Settings](#).

Metody publiczne

- [SettingsDialog](#) (QWidget *parent=0)
Konstruktor klasy.
- [~SettingsDialog](#) ()
Destruktor.
- [Settings settings](#) () const
settings

3.4.1 Opis szczegółowy

Klasa [SettingsDialog](#).

Klasa okna dialogowego ustawień portu szeregowego.

3.4.2 Dokumentacja konstruktora i destruktora

3.4.2.1 SettingsDialog::SettingsDialog (QWidget * parent = 0) [explicit]

Konstruktor klasy.

Inicjalizacja sygnałów i slotów

3.4.3 Dokumentacja funkcji składowych

3.4.3.1 Settings SettingsDialog::settings () const

settings

Zwracane wartości

<i>Settings</i>	aktualne ustawienia
-----------------	---------------------

Dokumentacja dla tej klasy została wygenerowana z pliku:

- settingsdialog.h

Skorowidz

Console, [5](#)

 Console, [6](#)

 getData, [6](#)

 putData, [6](#)

getData

 Console, [6](#)

MainWindow, [6](#)

 MainWindow, [7](#)

putData

 Console, [6](#)

settings

 SettingsDialog, [9](#)

SettingsDialog, [8](#)

 settings, [9](#)

 SettingsDialog, [9](#)

SettingsDialog::Settings, [7](#)