

Multimeter GUI

Generated by Doxygen 1.8.6

Sat May 7 2016 17:02:58

Contents

1	Main Page	1
2	multimeterGUI	3
3	Todo List	5
4	Namespace Index	7
4.1	Namespace List	7
5	Hierarchical Index	9
5.1	Class Hierarchy	9
6	Class Index	11
6.1	Class List	11
7	File Index	13
7.1	File List	13
8	Namespace Documentation	15
8.1	Ui Namespace Reference	15
9	Class Documentation	17
9.1	Flags Struct Reference	17
9.1.1	Detailed Description	18
9.1.2	Member Function Documentation	18
9.1.2.1	operator!=	18
9.1.2.2	operator==	18
9.1.3	Member Data Documentation	18
9.1.3.1	A	18
9.1.3.2	Auto	18
9.1.3.3	Bat	18
9.1.3.4	Beep	18
9.1.3.5	checksum	18
9.1.3.6	continuity	18

9.1.3.7	dash	18
9.1.3.8	dBm	18
9.1.3.9	Diode	18
9.1.3.10	DP	18
9.1.3.11	F	18
9.1.3.12	hFE	18
9.1.3.13	Hold	18
9.1.3.14	Hz	18
9.1.3.15	k	19
9.1.3.16	M	19
9.1.3.17	m	19
9.1.3.18	MAX	19
9.1.3.19	MIN	19
9.1.3.20	n	19
9.1.3.21	Ohms	19
9.1.3.22	percent	19
9.1.3.23	REL	19
9.1.3.24	RS232	19
9.1.3.25	s	19
9.1.3.26	tilde	19
9.1.3.27	u	19
9.1.3.28	V	19
9.2	LCD Class Reference	19
9.2.1	Detailed Description	20
9.2.2	Constructor & Destructor Documentation	20
9.2.2.1	LCD	20
9.2.3	Member Function Documentation	21
9.2.3.1	paintEvent	21
9.2.4	Member Data Documentation	21
9.2.4.1	data	21
9.2.4.2	lbl	21
9.3	Ui::MainWindow Class Reference	21
9.4	MainWindow Class Reference	22
9.4.1	Detailed Description	24
9.4.2	Constructor & Destructor Documentation	24
9.4.2.1	MainWindow	24
9.4.2.2	~MainWindow	25
9.4.3	Member Function Documentation	25
9.4.3.1	addData	25
9.4.3.2	on_connectButton_clicked	26

9.4.3.3	on_disconnectButton_clicked	26
9.4.3.4	resetData	27
9.4.4	Member Data Documentation	27
9.4.4.1	counter	27
9.4.4.2	graph	27
9.4.4.3	label	27
9.4.4.4	lcd	27
9.4.4.5	maxData	28
9.4.4.6	minData	28
9.4.4.7	newData	28
9.4.4.8	portPtr	28
9.4.4.9	rawdata	28
9.4.4.10	rData	28
9.4.4.11	scene	28
9.4.4.12	storeData	28
9.4.4.13	timeMark	28
9.4.4.14	timeRunning	28
9.4.4.15	tmp	28
9.4.4.16	ui	28
9.5	plotGraph Class Reference	28
9.5.1	Detailed Description	30
9.5.2	Constructor & Destructor Documentation	30
9.5.2.1	plotGraph	30
9.5.2.2	plotGraph	30
9.5.3	Member Function Documentation	30
9.5.3.1	boundingRect	30
9.5.3.2	labelXaxis	30
9.5.3.3	labelYaxis	31
9.5.3.4	linkData	31
9.5.3.5	paint	31
9.5.3.6	paintAxis	32
9.5.3.7	plotData	33
9.5.3.8	real2Coord	33
9.5.3.9	setScene	34
9.5.3.10	setUnit	34
9.5.3.11	setXaxis	34
9.5.3.12	setXsticks	34
9.5.3.13	setYaxis	35
9.5.3.14	setYsticks	35
9.5.4	Member Data Documentation	35

9.5.4.1	axisMargin	35
9.5.4.2	bRect	35
9.5.4.3	data	35
9.5.4.4	nx	35
9.5.4.5	ny	35
9.5.4.6	origin	35
9.5.4.7	rightX	35
9.5.4.8	scene	35
9.5.4.9	unit	35
9.5.4.10	upperY	36
9.5.4.11	xmax	36
9.5.4.12	xmin	36
9.5.4.13	ymax	36
9.5.4.14	ymin	36
9.6	qt_meta_stringdata_Foo_t Struct Reference	36
9.6.1	Member Data Documentation	36
9.6.1.1	data	36
9.6.1.2	stringdata0	36
9.7	qt_meta_stringdata_LCD_t Struct Reference	36
9.7.1	Member Data Documentation	36
9.7.1.1	data	36
9.7.1.2	stringdata0	36
9.8	qt_meta_stringdata_LCDscreen_t Struct Reference	37
9.8.1	Member Data Documentation	37
9.8.1.1	data	37
9.8.1.2	stringdata0	37
9.9	qt_meta_stringdata_MainWindow_t Struct Reference	37
9.9.1	Member Data Documentation	37
9.9.1.1	data	37
9.9.1.2	stringdata0	37
9.10	qt_meta_stringdata_plotGraph_t Struct Reference	37
9.10.1	Member Data Documentation	37
9.10.1.1	data	37
9.10.1.2	stringdata0	37
9.11	qt_meta_stringdata_RS22812_t Struct Reference	38
9.11.1	Member Data Documentation	38
9.11.1.1	data	38
9.11.1.2	stringdata0	38
9.12	qt_meta_stringdata_SerialPort_t Struct Reference	38
9.12.1	Member Data Documentation	38

9.12.1.1	data	38
9.12.1.2	stringdata0	38
9.13	RS22812 Class Reference	38
9.13.1	Detailed Description	40
9.13.2	Constructor & Destructor Documentation	40
9.13.2.1	RS22812	40
9.13.3	Member Function Documentation	41
9.13.3.1	byte2Digit	41
9.13.3.2	getDigitString	41
9.13.3.3	getFlags	42
9.13.3.4	getMode	42
9.13.3.5	getVal	42
9.13.3.6	modeChanged	43
9.13.3.7	newData	43
9.13.3.8	newMode	43
9.13.3.9	newValue	44
9.13.3.10	resetFlags	45
9.13.4	Member Data Documentation	45
9.13.4.1	digits	45
9.13.4.2	flags	45
9.13.4.3	mode	45
9.13.4.4	oldflags	45
9.14	SerialPort Class Reference	46
9.14.1	Detailed Description	47
9.14.2	Constructor & Destructor Documentation	47
9.14.2.1	SerialPort	47
9.14.2.2	~SerialPort	47
9.14.3	Member Function Documentation	48
9.14.3.1	closePort	48
9.14.3.2	listPorts	48
9.14.3.3	openPort	49
9.14.3.4	readPort	50
9.14.3.5	ready	50
9.14.3.6	readyRead	51
9.14.4	Member Data Documentation	51
9.14.4.1	activePort	51
9.14.4.2	BAUDRATE	51
9.14.4.3	buffer	51
9.14.4.4	DATABITS	51
9.14.4.5	isOpen	51

9.14.4.6	MODE	51
9.14.4.7	PARITY	51
9.14.4.8	ports	51
9.14.4.9	readConnect	51
9.14.4.10	STOPBITS	52
9.15	Ui_MainWindow Class Reference	52
9.15.1	Member Function Documentation	53
9.15.1.1	retranslateUi	53
9.15.1.2	setupUi	53
9.15.2	Member Data Documentation	53
9.15.2.1	centralWidget	53
9.15.2.2	comboBoxPort	53
9.15.2.3	connectButton	53
9.15.2.4	disconnectButton	53
9.15.2.5	graphPlot	53
9.15.2.6	gridLayout	53
9.15.2.7	horizontalLayout	53
9.15.2.8	labelPort	54
9.15.2.9	mainToolBar	54
9.15.2.10	menuBar	54
9.15.2.11	statusBar	54
9.15.2.12	toolBar	54
9.15.2.13	verticalLayout	54
9.15.2.14	verticalLayout_2	54
10	File Documentation	55
10.1	build-multimeterGUI-Desktop-Debug/moc_datars22812.cpp File Reference	55
10.2	build-multimeterGUI-Desktop-Debug/moc_foo.cpp File Reference	55
10.2.1	Macro Definition Documentation	56
10.2.1.1	QT_MOC_LITERAL	56
10.2.2	Variable Documentation	56
10.2.2.1	qt_meta_data_Foo	56
10.2.2.2	qt_meta_stringdata_Foo	56
10.3	build-multimeterGUI-Desktop-Debug/moc_lcd.cpp File Reference	56
10.3.1	Macro Definition Documentation	57
10.3.1.1	QT_MOC_LITERAL	57
10.3.2	Variable Documentation	57
10.3.2.1	qt_meta_data_LCD	57
10.3.2.2	qt_meta_stringdata_LCD	58
10.4	build-multimeterGUI-Desktop-Debug/moc_lcdscreen.cpp File Reference	58

10.4.1	Macro Definition Documentation	59
10.4.1.1	QT_MOC_LITERAL	59
10.4.2	Variable Documentation	59
10.4.2.1	qt_meta_data_LCDscreen	59
10.4.2.2	qt_meta_stringdata_LCDscreen	59
10.5	build-multimeterGUI-Desktop-Debug/moc_mainwindow.cpp File Reference	59
10.5.1	Macro Definition Documentation	60
10.5.1.1	QT_MOC_LITERAL	60
10.5.2	Variable Documentation	60
10.5.2.1	qt_meta_data_MainWindow	60
10.5.2.2	qt_meta_stringdata_MainWindow	60
10.6	build-multimeterGUI-Desktop-Debug/moc_plotgraph.cpp File Reference	61
10.6.1	Macro Definition Documentation	61
10.6.1.1	QT_MOC_LITERAL	61
10.6.2	Variable Documentation	62
10.6.2.1	qt_meta_data_plotGraph	62
10.6.2.2	qt_meta_stringdata_plotGraph	62
10.7	build-multimeterGUI-Desktop-Debug/moc_rs22812.cpp File Reference	62
10.7.1	Macro Definition Documentation	63
10.7.1.1	QT_MOC_LITERAL	63
10.7.2	Variable Documentation	63
10.7.2.1	qt_meta_data_RS22812	63
10.7.2.2	qt_meta_stringdata_RS22812	64
10.8	build-multimeterGUI-Desktop-Debug/moc_serialport.cpp File Reference	64
10.8.1	Macro Definition Documentation	64
10.8.1.1	QT_MOC_LITERAL	64
10.8.2	Variable Documentation	65
10.8.2.1	qt_meta_data_SerialPort	65
10.8.2.2	qt_meta_stringdata_SerialPort	65
10.9	build-multimeterGUI-Desktop-Debug/ui_mainwindow.h File Reference	66
10.10	lcd.cpp File Reference	66
10.11	lcd.h File Reference	67
10.12	main.cpp File Reference	68
10.12.1	Function Documentation	68
10.12.1.1	main	68
10.13	mainwindow.cpp File Reference	68
10.14	mainwindow.h File Reference	69
10.15	plotgraph.cpp File Reference	69
10.16	plotgraph.h File Reference	70
10.17	README.md File Reference	71

10.18rs22812.cpp File Reference	71
10.19rs22812.h File Reference	71
10.20serialport.cpp File Reference	72
10.21serialport.h File Reference	72
Index	74

Chapter 1

Main Page

Multimeter GUI GUI for the RS-232 mode of the Radio Shack 22-812. Copyright (C) 2016 FJ Salguero

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Chapter 2

multimeterGUI

GUI for the Radio Shack 22-812 multimeter when used in RS232 mode.

This is a QT GUI to interface the RS 22-812. It has been developed with QT5 in OpenSuse Leap.

This code is "as-is". No warranty or responsibility is assumed by the author. You can use your code for any non-commercial purposes, and modify it as you wish with the only condition of licensing it with the same conditions. If you find it useful, make any change or have any suggestion, I will appreciate your input.

Chapter 3

Todo List

Member `MainWindow::MainWindow` (`QWidget *parent=0`)

: Temporary. It has to be set automatically.

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui	15
----------	----

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

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

Flags	17
QGraphicsItem	
plotGraph	28
QLabel	
LCD	19
QMainWindow	
MainWindow	22
QObject	
RS22812	38
SerialPort	46
qt_meta_stringdata_Foo_t	36
qt_meta_stringdata_LCD_t	36
qt_meta_stringdata_LCDscreen_t	37
qt_meta_stringdata_MainWindow_t	37
qt_meta_stringdata_plotGraph_t	37
qt_meta_stringdata_RS22812_t	38
qt_meta_stringdata_SerialPort_t	38
Ui_MainWindow	52
Ui::MainWindow	21

Chapter 6

Class Index

6.1 Class List

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

Flags		
	Definition of custom data type	17
LCD		
	Displays the numerical value read	19
Ui::MainWindow		21
MainWindow		
	The MainWindow class	22
plotGraph		
	The plotGraph class	28
qt_meta_stringdata_Foo_t		36
qt_meta_stringdata_LCD_t		36
qt_meta_stringdata_LCDscreen_t		37
qt_meta_stringdata_MainWindow_t		37
qt_meta_stringdata_plotGraph_t		37
qt_meta_stringdata_RS22812_t		38
qt_meta_stringdata_SerialPort_t		38
RS22812		
	Decoding of the data sent by the Radio Shack 22-812	38
SerialPort		
	Class to manage the communication with a serial port	46
Ui_MainWindow		52

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

lcd.cpp	66
lcd.h	67
main.cpp	68
mainwindow.cpp	68
mainwindow.h	69
plotgraph.cpp	69
plotgraph.h	70
rs22812.cpp	71
rs22812.h	71
serialport.cpp	72
serialport.h	72
build-multimeterGUI-Desktop-Debug/ moc_datars22812.cpp	55
build-multimeterGUI-Desktop-Debug/ moc_foo.cpp	55
build-multimeterGUI-Desktop-Debug/ moc_lcd.cpp	56
build-multimeterGUI-Desktop-Debug/ moc_lcdscreen.cpp	58
build-multimeterGUI-Desktop-Debug/ moc_mainwindow.cpp	59
build-multimeterGUI-Desktop-Debug/ moc_plotgraph.cpp	61
build-multimeterGUI-Desktop-Debug/ moc_rs22812.cpp	62
build-multimeterGUI-Desktop-Debug/ moc_serialport.cpp	64
build-multimeterGUI-Desktop-Debug/ ui_mainwindow.h	66

Chapter 8

Namespace Documentation

8.1 Ui Namespace Reference

Classes

- class [MainWindow](#)

Chapter 9

Class Documentation

9.1 Flags Struct Reference

Definition of custom data type.

```
#include <rs22812.h>
```

Public Member Functions

- bool `operator==` (const `Flags` &f2) const
operator == Equality operator for `Flags` struct.
- bool `operator!=` (const `Flags` &f2)

Public Attributes

- bool `Hz`
- bool `Ohms`
- bool `k`
- bool `M`
- bool `F`
- bool `A`
- bool `V`
- bool `m`
- bool `u`
- bool `n`
- bool `dBm`
- bool `s`
- bool `percent`
- bool `hFE`
- bool `REL`
- bool `MIN`
- bool `Beep`
- bool `Diode`
- bool `continuity`
- bool `Bat`
- bool `Hold`
- bool `dash`
- bool `tilde`
- bool `RS232`

- bool [Auto](#)
- bool [MAX](#)
- int [DP](#)
- bool [checksum](#)

9.1.1 Detailed Description

Definition of custom data type.

9.1.2 Member Function Documentation

9.1.2.1 bool [Flags::operator!=](#) (const [Flags](#) & *f2*) [\[inline\]](#)

9.1.2.2 bool [Flags::operator==](#) (const [Flags](#) & *f2*) const [\[inline\]](#)

[operator ==](#) Equality operator for [Flags](#) struct.

Parameters

<i>f1</i>	
<i>f2</i>	

Returns

9.1.3 Member Data Documentation

9.1.3.1 bool [Flags::A](#)

9.1.3.2 bool [Flags::Auto](#)

9.1.3.3 bool [Flags::Bat](#)

9.1.3.4 bool [Flags::Beep](#)

9.1.3.5 bool [Flags::checksum](#)

9.1.3.6 bool [Flags::continuity](#)

9.1.3.7 bool [Flags::dash](#)

9.1.3.8 bool [Flags::dBm](#)

9.1.3.9 bool [Flags::Diode](#)

9.1.3.10 int [Flags::DP](#)

9.1.3.11 bool [Flags::F](#)

9.1.3.12 bool [Flags::hFE](#)

9.1.3.13 bool [Flags::Hold](#)

9.1.3.14 bool [Flags::Hz](#)

9.1.3.15 bool Flags::k

9.1.3.16 bool Flags::M

9.1.3.17 bool Flags::m

9.1.3.18 bool Flags::MAX

9.1.3.19 bool Flags::MIN

9.1.3.20 bool Flags::n

9.1.3.21 bool Flags::Ohms

9.1.3.22 bool Flags::percent

9.1.3.23 bool Flags::REL

9.1.3.24 bool Flags::RS232

9.1.3.25 bool Flags::s

9.1.3.26 bool Flags::tilde

9.1.3.27 bool Flags::u

9.1.3.28 bool Flags::V

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

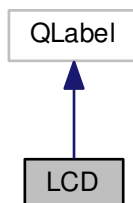
- [rs22812.h](#)

9.2 LCD Class Reference

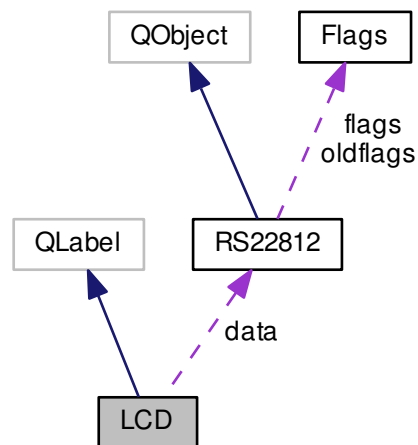
The [LCD](#) class displays the numerical value read.

```
#include <lcd.h>
```

Inheritance diagram for LCD:



Collaboration diagram for LCD:



Public Member Functions

- [LCD](#) (const [RS22812](#) *[data](#), QWidget *parent=0)
[LCD::LCD](#). Constructor.

Protected Member Functions

- void [paintEvent](#) (QPaintEvent *event)
[LCD::paintEvent](#). Paint event handler.

Private Attributes

- const QVector< QString > [lbl](#) ={"Auto","RS232","Hold","Rel","MAX","MIN","hFE","dBm","Cont","Diode","%","S"}
- const [RS22812](#) * [data](#)

9.2.1 Detailed Description

The [LCD](#) class displays the numerical value read.

This class will show a representation of the multimeter's display showing the same values that are shown in the multimeter.

9.2.2 Constructor & Destructor Documentation

9.2.2.1 [LCD::LCD](#) (const [RS22812](#) * [data](#), QWidget * [parent](#) = 0) [explicit]

[LCD::LCD](#). Constructor.

Multimeter GUI GUI for the RS-232 mode of the Radio Shack 22-812. Copyright (C) 2016 FJ Salguero

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Parameters

<i>data</i>	
<i>parent</i>	

9.2.3 Member Function Documentation

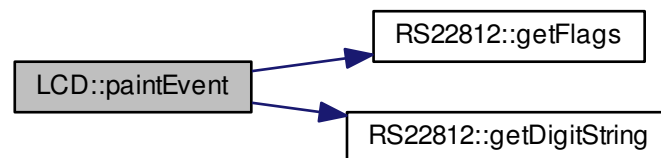
9.2.3.1 void LCD::paintEvent (QPaintEvent * *event*) [protected]

[LCD::paintEvent](#). Paint event handler.

Parameters

<i>event</i>	It redraws the LCD widget every time there is an update.
--------------	--

Here is the call graph for this function:



9.2.4 Member Data Documentation

9.2.4.1 const RS22812* LCD::data [private]

9.2.4.2 const QVector<QString> LCD::lbl ={"Auto", "RS232", "Hold", "Rel", "MAX", "MIN", "hFE", "dBm", "Cont", "Diode", "%", "S"} [private]

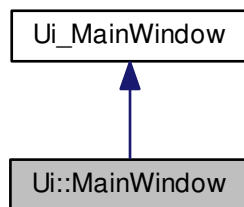
The documentation for this class was generated from the following files:

- [lcd.h](#)
- [lcd.cpp](#)

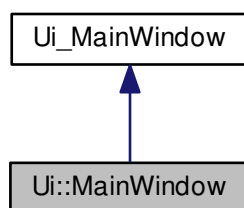
9.3 Ui::MainWindow Class Reference

```
#include <ui_mainwindow.h>
```

Inheritance diagram for Ui::MainWindow:



Collaboration diagram for Ui::MainWindow:



Additional Inherited Members

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

- [build-multimeterGUI-Desktop-Debug/ui_mainwindow.h](#)

9.4 MainWindow Class Reference

The [MainWindow](#) class.

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


Private Attributes

- [Ui::MainWindow](#) * [ui](#)
- [SerialPort](#) * [portPtr](#)
- [RS22812](#) * [rawdata](#)
- [LCD](#) * [lcd](#)
- [QLabel](#) * [label](#)
- [plotGraph](#) * [graph](#)
- [QGraphicsScene](#) * [scene](#)
- [QVector< QPair< qint64, qreal > >](#) [storeData](#)
- [qint32](#) [counter](#) =0
- [qreal](#) [minData](#) =99999999
- [qreal](#) [maxData](#) =-99999999
- [QMetaObject::Connection](#) [newData](#)
- [QMetaObject::Connection](#) [rData](#)
- [QElapsedTimer](#) * [timeMark](#)
- [bool](#) [timeRunning](#)
- [QVector< QPair< qreal, qreal > >](#) [tmp](#)

9.4.1 Detailed Description

The [MainWindow](#) class.

9.4.2 Constructor & Destructor Documentation

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

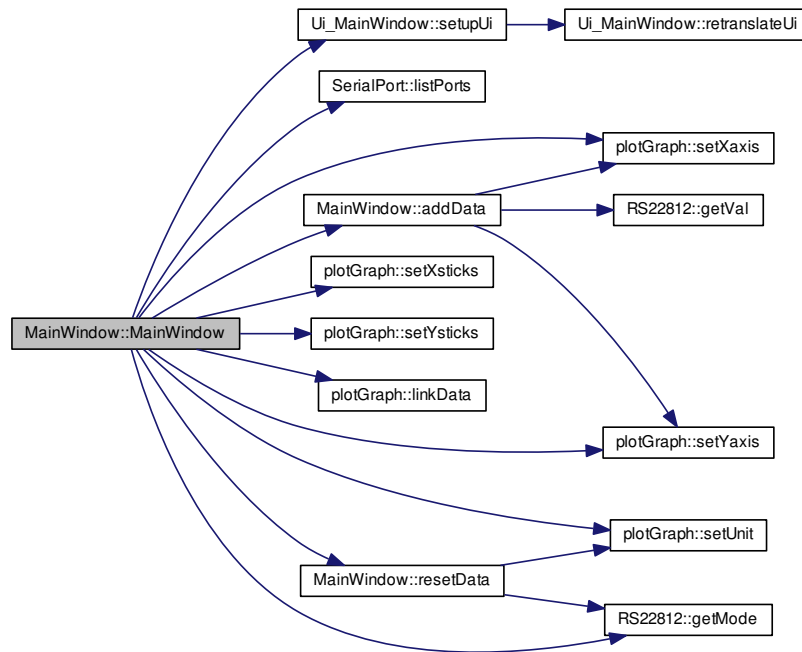
[MainWindow::MainWindow](#). Constructor.

Parameters

<i>parent</i>	The main window constructor will, in addition to create the corresponding subwidgets, populate the list of available ports and connect signals with slots.
---------------	--

Todo : Temporary. It has to be set automatically.

Here is the call graph for this function:



9.4.2.2 MainWindow::~~MainWindow ()

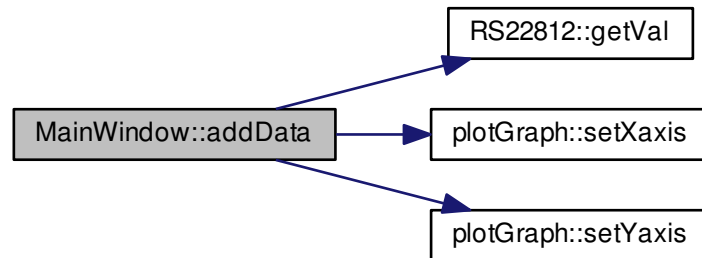
[MainWindow::~~MainWindow](#).

9.4.3 Member Function Documentation

9.4.3.1 void MainWindow::addData () [private],[slot]

[MainWindow::addData](#). Add new data to the data set. This method will be called when new data has been read from the serial port, it will add the new value to the stored set of pairs (time,value) and update the graph.

Here is the call graph for this function:



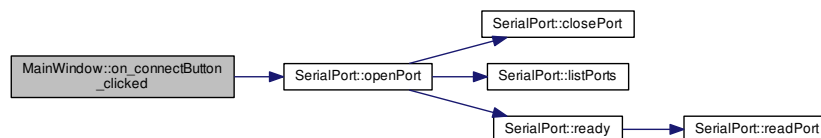
Here is the caller graph for this function:



9.4.3.2 `void MainWindow::on_connectButton_clicked () [private],[slot]`

[MainWindow::on_connectButton_clicked](#). Open the selected port.

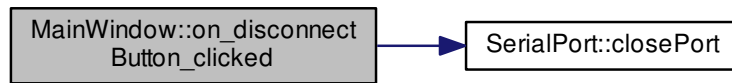
Here is the call graph for this function:



9.4.3.3 `void MainWindow::on_disconnectButton_clicked () [private],[slot]`

[MainWindow::on_disconnectButton_clicked](#). Disconnect from the current port.

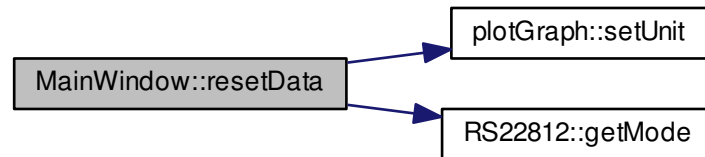
Here is the call graph for this function:



9.4.3.4 void MainWindow::resetData () [private],[slot]

[MainWindow::resetData](#). Resets the data stored in memory. This method is called when the multimeter's mode changes, clearing all the data stored in memory.

Here is the call graph for this function:



Here is the caller graph for this function:



9.4.4 Member Data Documentation

9.4.4.1 qint32 MainWindow::counter =0 [private]

9.4.4.2 plotGraph* MainWindow::graph [private]

9.4.4.3 QLabel* MainWindow::label [private]

9.4.4.4 LCD* MainWindow::lcd [private]

- 9.4.4.5 `qreal MainWindow::maxData = -99999999` [private]
- 9.4.4.6 `qreal MainWindow::minData = 99999999` [private]
- 9.4.4.7 `QObject::Connection MainWindow::newData` [private]
- 9.4.4.8 `SerialPort* MainWindow::portPtr` [private]
- 9.4.4.9 `RS22812* MainWindow::rawdata` [private]
- 9.4.4.10 `QObject::Connection MainWindow::rData` [private]
- 9.4.4.11 `QGraphicsScene* MainWindow::scene` [private]
- 9.4.4.12 `QVector<QPair<qint64,qreal>> MainWindow::storeData` [private]
- 9.4.4.13 `QElapsedTimer* MainWindow::timeMark` [private]
- 9.4.4.14 `bool MainWindow::timeRunning` [private]
- 9.4.4.15 `QVector<QPair<qreal,qreal>> MainWindow::tmp` [private]
- 9.4.4.16 `Ui::MainWindow* MainWindow::ui` [private]

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

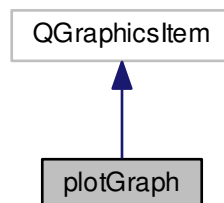
- [mainwindow.h](#)
- [mainwindow.cpp](#)

9.5 plotGraph Class Reference

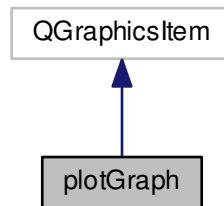
The [plotGraph](#) class.

```
#include <plotgraph.h>
```

Inheritance diagram for plotGraph:



Collaboration diagram for plotGraph:



Public Member Functions

- `plotGraph ()`
- `plotGraph (QGraphicsScene *scene)`
- void `paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`
plotGraph::paint. Override of the paint method of QGraphicsScene.
- `QRectF boundingRect () const`
plotGraph::boundingRect. Returns the bounding rectangle of the graph.
- void `setXaxis (qint64 minVal, qint64 maxVal)`
plotGraph::setXaxis. Sets the maximum and minimum values of the x axis.
- void `setYaxis (qreal minVal, qreal maxVal)`
plotGraph::setYaxis
- void `setXsticks (int nSticks)`
plotGraph::setXsticks
- void `setYsticks (int nSticks)`
- void `linkData (const QVector< QPair< qint64, qreal > > *dat)`
- void `setUnit (int U)`
- void `setScene (QGraphicsScene *scene)`

Private Member Functions

- void `paintAxis (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`
plotGraph::paintAxis Calculates the scale and limits of the axis and draws it.
- void `plotData (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`
plotGraph::plotData Draws the data on the widget.
- void `labelXaxis (QPainter *painter, QPoint &p1, QPoint &p2)`
plotGraph::labelXaxis Add labels to the X axis.
- void `labelYaxis (QPainter *painter, QPoint &p1, QPoint &p2)`
plotGraph::labelYaxis Adds labels to the Y axis.
- `QPoint real2Coord (const QPair< qreal, qreal > dpoint)`
plotGraph::real2Coord Transform reading coordinates to widget coordinates.

Private Attributes

- QRect **bRect** =QRect(0,0,0,0)
- QPoint **origin**
- QPoint **rightX**
- QPoint **upperY**
- const int **axisMargin** =40
- qreal **xmin** =0
- qreal **xmax** =1
- qreal **ymin** =0
- qreal **ymax** =1
- int **nx** =2
- int **ny** =2
- const QVector< QPair< qint64, qreal > > * **data** =NULL
- QString **unit**
- QGraphicsScene * **scene**

9.5.1 Detailed Description

The **plotGraph** class.

This class graphs the values read from the multimeter versus the time.

9.5.2 Constructor & Destructor Documentation

9.5.2.1 **plotGraph::plotGraph ()**

9.5.2.2 **plotGraph::plotGraph (QGraphicsScene * *scene*)**

9.5.3 Member Function Documentation

9.5.3.1 **QRectF plotGraph::boundingRect () const**

plotGraph::boundingRect. Returns the bounding rectangle of the graph.

Returns

So far, it returns a fake value. Need to implement.

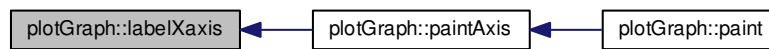
9.5.3.2 **void plotGraph::labelXaxis (QPainter * *painter*, QPoint & *p1*, QPoint & *p2*)** `[private]`

plotGraph::labelXaxis Add labels to the X axis.

Parameters

<i>painter</i>	
<i>p1</i>	
<i>p2</i>	

Here is the caller graph for this function:



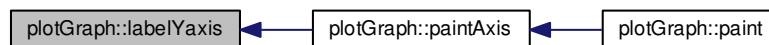
9.5.3.3 void `plotGraph::labelYaxis` (`QPainter * painter`, `QPoint & p1`, `QPoint & p2`) [private]

[plotGraph::labelYaxis](#) Adds labels to the Y axis.

Parameters

<i>painter</i>	
<i>p1</i>	
<i>p2</i>	

Here is the caller graph for this function:



9.5.3.4 void `plotGraph::linkData` (const `QVector< QPair< qint64, qreal > > * dat`)

Here is the caller graph for this function:



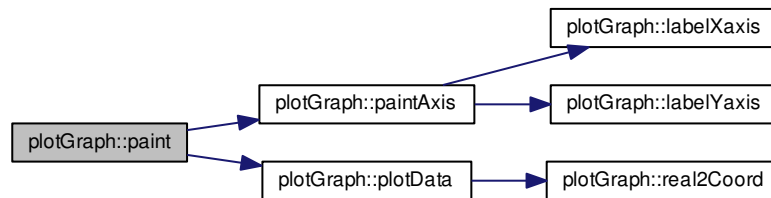
9.5.3.5 void `plotGraph::paint` (`QPainter * painter`, const `QStyleOptionGraphicsItem * option`, `QWidget * widget`)

[plotGraph::paint](#). Override of the paint method of `QGraphicsScene`.

Parameters

<i>painter</i>	
<i>option</i>	
<i>widget</i>	

Here is the call graph for this function:



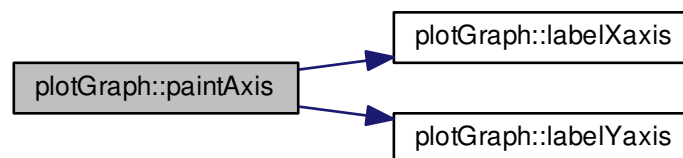
9.5.3.6 `void plotGraph::paintAxis (QPainter * painter, const QStyleOptionGraphicsItem * option, QWidget * widget)`
`[private]`

`plotGraph::paintAxis` Calculates the scale and limits of the axis and draws it.

Parameters

<i>painter</i>	
<i>option</i>	
<i>widget</i>	

Here is the call graph for this function:



Here is the caller graph for this function:



9.5.3.7 void plotGraph::plotData (QPainter * *painter*, const QStyleOptionGraphicsItem * *option*, QWidget * *widget*)
[private]

[plotGraph::plotData](#) Draws the data on the widget.

Parameters

<i>painter</i>	
<i>option</i>	
<i>widget</i>	

Here is the call graph for this function:



Here is the caller graph for this function:



9.5.3.8 QPoint plotGraph::real2Coord (const QPair< qreal, qreal > *dpoint*) [private]

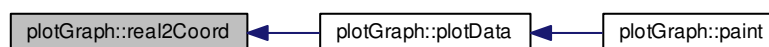
[plotGraph::real2Coord](#) Transform reading coordinates to widget coordinates.

Parameters

<i>dpoint</i>	
---------------	--

Returns

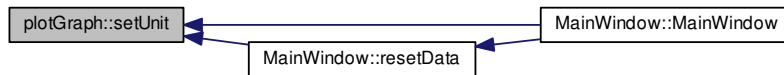
Here is the caller graph for this function:



9.5.3.9 void plotGraph::setScene (QGraphicsScene * *scene*)

9.5.3.10 void plotGraph::setUnit (int *U*)

Here is the caller graph for this function:



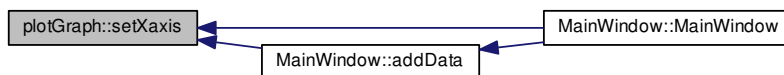
9.5.3.11 void plotGraph::setXaxis (qint64 *minVal*, qint64 *maxVal*)

[plotGraph::setXaxis](#). Sets the maximum and minimum values of the x axis.

Parameters

<i>minVal</i>	
<i>maxVal</i>	

Here is the caller graph for this function:



9.5.3.12 void plotGraph::setXsticks (int *nSticks*)

[plotGraph::setXsticks](#)

Parameters

<i>nSticks</i>	
----------------	--

Here is the caller graph for this function:



9.5.3.13 void plotGraph::setYaxis (qreal *minVal*, qreal *maxVal*)

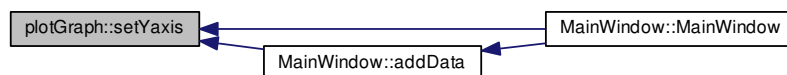
[plotGraph::setYaxis](#)

Sets the maximum and minimum values of the y axis.

Parameters

<i>minVal</i>	
<i>maxVal</i>	

Here is the caller graph for this function:



9.5.3.14 void plotGraph::setYsticks (int *nSticks*)

Here is the caller graph for this function:



9.5.4 Member Data Documentation

9.5.4.1 const int plotGraph::axisMargin =40 [private]

9.5.4.2 QRect plotGraph::bRect =QRect(0,0,0,0) [private]

9.5.4.3 const QVector<QPair<qint64,qreal> >* plotGraph::data =NULL [private]

9.5.4.4 int plotGraph::nx =2 [private]

9.5.4.5 int plotGraph::ny =2 [private]

9.5.4.6 QPoint plotGraph::origin [private]

9.5.4.7 QPoint plotGraph::rightX [private]

9.5.4.8 QGraphicsScene* plotGraph::scene [private]

9.5.4.9 QString plotGraph::unit [private]

9.5.4.10 `QPoint plotGraph::upperY` [private]

9.5.4.11 `qreal plotGraph::xmax =1` [private]

9.5.4.12 `qreal plotGraph::xmin =0` [private]

9.5.4.13 `qreal plotGraph::ymax =1` [private]

9.5.4.14 `qreal plotGraph::ymin =0` [private]

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

- [plotgraph.h](#)
- [plotgraph.cpp](#)

9.6 qt_meta_stringdata_Foo_t Struct Reference

Public Attributes

- `QByteArrayData data` [1]
- `char stringdata0` [4]

9.6.1 Member Data Documentation

9.6.1.1 `QByteArrayData qt_meta_stringdata_Foo_t::data`[1]

9.6.1.2 `char qt_meta_stringdata_Foo_t::stringdata0`[4]

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

- `build-multimeterGUI-Desktop-Debug/moc_foo.cpp`

9.7 qt_meta_stringdata_LCD_t Struct Reference

Public Attributes

- `QByteArrayData data` [1]
- `char stringdata0` [4]

9.7.1 Member Data Documentation

9.7.1.1 `QByteArrayData qt_meta_stringdata_LCD_t::data`[1]

9.7.1.2 `char qt_meta_stringdata_LCD_t::stringdata0`[4]

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

- `build-multimeterGUI-Desktop-Debug/moc_lcd.cpp`

9.8 qt_meta_stringdata_LCDscreen_t Struct Reference

Public Attributes

- QByteArrayData [data](#) [1]
- char [stringdata0](#) [10]

9.8.1 Member Data Documentation

9.8.1.1 QByteArrayData qt_meta_stringdata_LCDscreen_t::data[1]

9.8.1.2 char qt_meta_stringdata_LCDscreen_t::stringdata0[10]

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

- build-multimeterGUI-Desktop-Debug/[moc_lcdscreen.cpp](#)

9.9 qt_meta_stringdata_MainWindow_t Struct Reference

Public Attributes

- QByteArrayData [data](#) [6]
- char [stringdata0](#) [83]

9.9.1 Member Data Documentation

9.9.1.1 QByteArrayData qt_meta_stringdata_MainWindow_t::data[6]

9.9.1.2 char qt_meta_stringdata_MainWindow_t::stringdata0[83]

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

- build-multimeterGUI-Desktop-Debug/[moc_mainwindow.cpp](#)

9.10 qt_meta_stringdata_plotGraph_t Struct Reference

Public Attributes

- QByteArrayData [data](#) [1]
- char [stringdata0](#) [10]

9.10.1 Member Data Documentation

9.10.1.1 QByteArrayData qt_meta_stringdata_plotGraph_t::data[1]

9.10.1.2 char qt_meta_stringdata_plotGraph_t::stringdata0[10]

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

- build-multimeterGUI-Desktop-Debug/[moc_plotgraph.cpp](#)

9.11 qt_meta_stringdata_RS22812_t Struct Reference

Public Attributes

- QByteArrayData [data](#) [6]
- char [stringdata0](#) [39]

9.11.1 Member Data Documentation

9.11.1.1 QByteArrayData qt_meta_stringdata_RS22812_t::data[6]

9.11.1.2 char qt_meta_stringdata_RS22812_t::stringdata0[39]

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

- build-multimeterGUI-Desktop-Debug/[moc_rs22812.cpp](#)

9.12 qt_meta_stringdata_SerialPort_t Struct Reference

Public Attributes

- QByteArrayData [data](#) [5]
- char [stringdata0](#) [35]

9.12.1 Member Data Documentation

9.12.1.1 QByteArrayData qt_meta_stringdata_SerialPort_t::data[5]

9.12.1.2 char qt_meta_stringdata_SerialPort_t::stringdata0[35]

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

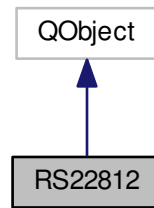
- build-multimeterGUI-Desktop-Debug/[moc_serialport.cpp](#)

9.13 RS22812 Class Reference

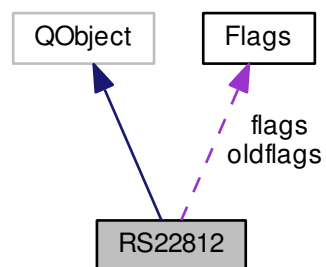
Decoding of the data sent by the Radio Shack 22-812.

```
#include <rs22812.h>
```


Inheritance diagram for RS22812:



Collaboration diagram for RS22812:



Public Slots

- void [newValue](#) (const QByteArray &data)
RS22812::newValue Reads a new packet and reformats the data.

Signals

- void [newMode](#) ()
- void [newData](#) ()

Public Member Functions

- [RS22812](#) (QObject *parent=0)
- float [getVal](#) () const
RS22812::getVal Returns the numeric value of the reading.
- [Flags](#) [getFlags](#) () const
RS22812::getFlags Gets the current flags structure.
- QString [getDigitString](#) () const

- [RS22812::getDigitString](#) It returns the multimeter reading in string format.
- `uint getMode () const`
[RS22812::getMode](#).

Private Member Functions

- `bool modeChanged ()`
[RS22812::modeChanged](#) Checks the flags structure to see whether the read mode changed.
- `QString byte2Digit (uchar byte)`
[RS22812::byte2Digit](#) Translates the RS 22-812 byte value of the [LCD](#) mapping into a digit.
- `void resetFlags (Flags &f)`
[RS22812::resetFlags](#) Sets all the flags to false.

Private Attributes

- `uint mode`
- `Flags flags`
- `Flags oldflags`
- `QString digits`

9.13.1 Detailed Description

Decoding of the data sent by the Radio Shack 22-812.

Part of the information was obtained from http://sigrok.org/wiki/RadioShack_22-812 and <https://code.google.com/archive/p/rs22812/>

RS 22-812 sends 9bytes packets. Each packect is a mapping of the [LCD](#) of the screen plus some extra information.
 Bit Byte 7 6 5 4 3 2 1 0 0 ----- Mode ----- 1 Hz Ohms K M F A V m 2 u n dBm s %
 hFE REL MIN 3 4D 4C 4G 4B DP3 4E 4F 4A 4 3D 3C 3G 3B DP2 3E 3F 3A 5 2D 2C 2G 2B DP1 2E 2F 2A 6 1D 1C
 1G 1B MAX 1E 1F 1A 7 Beep Diode Bat Hold - ~ RS232 Auto 8 ----- Checksum -----

The LED mapping is: | -A- | | | F B | | | -G- | | | E C | | | -D- |

So, the equivalence between int value and digit are: 215 : "0", 80 : "1", 181 : "2", 241 : "3", 114 : "4", 227 : "5", 231 : "6", 81 : "7", 247 : "8", 243 : "9", 39 : "F", 55 : "P", 167 : "E", 135 : "C", 134 : "L", 118 : "H", 6 : "I", 102 : "h", 36 : "r", 166 : "t", 100 : "n", 32 : "-", 0 : " "

And the possible modes are: 0=DC V 1=AC V 2=DC uA 3=DC mA 4=DC A 5=AC uA 6=AC mA 7=AC A 8=OHM 9=CAP 10=HZ 11=NET HZ 12=AMP HZ 13=DUTY 14=NET DUTY 15=AMP DUTY 16=WIDTH 17=NET WIDTH 18=AMP WIDTH 19=DIODE 20=CONT 21=HFE 22=LOGIC 23=DBM 24=EF 25=TEMP

9.13.2 Constructor & Destructor Documentation

9.13.2.1 RS22812::RS22812 (QObject * parent = 0) [explicit]

Multimeter GUI GUI for the RS-232 mode of the Radio Shack 22-812. Copyright (C) 2016 FJ Salguero

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Here is the call graph for this function:



9.13.3 Member Function Documentation

9.13.3.1 QString RS22812::byte2Digit (uchar *byte*) [private]

[RS22812::byte2Digit](#) Translates the RS 22-812 byte value of the [LCD](#) mapping into a digit.

Parameters

<i>byte</i>	RS 22-812 byte value
-------------	----------------------

Returns

String with the equivalent digit.

Here is the caller graph for this function:



9.13.3.2 QString RS22812::getDigitString () const

[RS22812::getDigitString](#) It returns the multimeter reading in string format.

Returns

Here is the caller graph for this function:



9.13.3.3 Flags RS22812::getFlags () const

[RS22812::getFlags](#) Gets the current flags structure.

Returns

Here is the caller graph for this function:



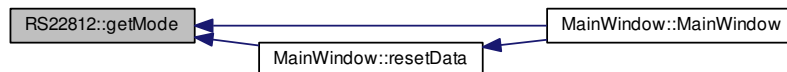
9.13.3.4 uint RS22812::getMode () const

[RS22812::getMode](#).

Returns

It returns the mode on which the multimeter is working.

Here is the caller graph for this function:

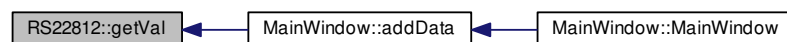


9.13.3.5 float RS22812::getVal () const

[RS22812::getVal](#) Returns the numeric value of the reading.

Returns

Here is the caller graph for this function:



9.13.3.6 `bool RS22812::modeChanged () [private]`

[RS22812::modeChanged](#) Checks the flags structure to see whether the read mode changed.

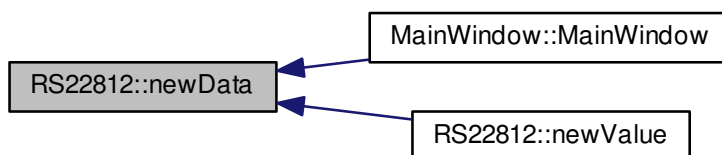
Returns

Here is the caller graph for this function:



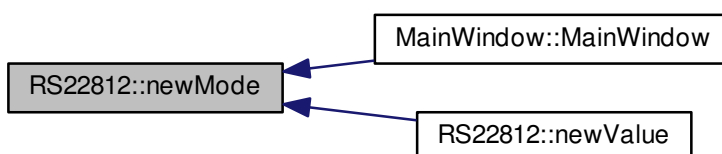
9.13.3.7 `void RS22812::newData () [signal]`

Here is the caller graph for this function:



9.13.3.8 `void RS22812::newMode () [signal]`

Here is the caller graph for this function:



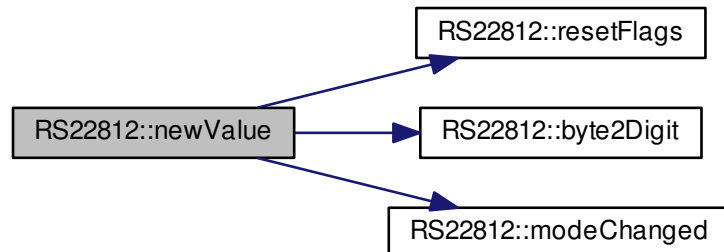
9.13.3.9 void RS22812::newValue (const QByteArray & *data*) [slot]

[RS22812::newValue](#) Reads a new packet and reformats the data.

Parameters

<i>data</i>	
-------------	--

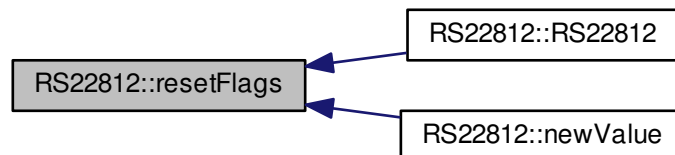
Here is the call graph for this function:



9.13.3.10 void RS22812::resetFlags (Flags & f) [private]

[RS22812::resetFlags](#) Sets all the flags to false.

Here is the caller graph for this function:



9.13.4 Member Data Documentation

9.13.4.1 QString RS22812::digits [private]

9.13.4.2 Flags RS22812::flags [private]

9.13.4.3 uint RS22812::mode [private]

9.13.4.4 Flags RS22812::oldflags [private]

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

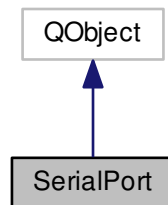
- [rs22812.h](#)
- [build-multimeterGUI-Desktop-Debug/moc_rs22812.cpp](#)
- [rs22812.cpp](#)

9.14 SerialPort Class Reference

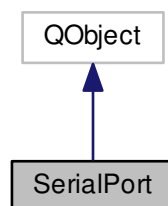
Class to manage the communication with a serial port.

```
#include <serialport.h>
```

Inheritance diagram for SerialPort:



Collaboration diagram for SerialPort:



Public Slots

- void `ready` ()
`SerialPort::ready` Re-emits the `readyRead` signal.

Signals

- void `readyRead` (QByteArray `buffer`)

Public Member Functions

- `SerialPort` (QObject *parent=0)
`SerialPort::SerialPort` Constructor.
- `~SerialPort` ()
`SerialPort::~~SerialPort`.

- bool [openPort](#) (const QString portName)
SerialPort::openPort Opens a serial port for reading. If port name is empty, it does nothing. If port name is not available. Does nothing and raises a warning.
- bool [closePort](#) ()
SerialPort::closePort.
- QList< QSerialPortInfo > [listPorts](#) ()
SerialPort::listPorts Obtains and return the list of available ports.

Private Member Functions

- void [readPort](#) ()
SerialPort::readPort Reads the available data.

Private Attributes

- QSerialPortInfo * [ports](#)
- QSerialPort * [activePort](#)
- bool [isOpen](#)
- QByteArray [buffer](#)
- QMetaObject::Connection [readConnect](#)
- const QSerialPort::OpenMode [MODE](#) =QSerialPort::ReadOnly

Static Private Attributes

- static const QSerialPort::BaudRate [BAUDRATE](#) =QSerialPort::Baud4800
- static const QSerialPort::DataBits [DATABITS](#) =QSerialPort::Data8
- static const QSerialPort::StopBits [STOPBITS](#) =QSerialPort::OneStop
- static const QSerialPort::Parity [PARITY](#) =QSerialPort::NoParity

9.14.1 Detailed Description

Class to manage the communication with a serial port.

RS 22-812 sends 9bytes long packets with the codified information. This class is meant to read those packets and send it to the [RS22812](#) class to store and interpret.

9.14.2 Constructor & Destructor Documentation

9.14.2.1 SerialPort::SerialPort (QObject * *parent* = 0) [explicit]

[SerialPort::SerialPort](#) Constructor.

Parameters

<i>parent</i>	
---------------	--

9.14.2.2 SerialPort::~SerialPort ()

[SerialPort::~SerialPort](#).

Here is the call graph for this function:



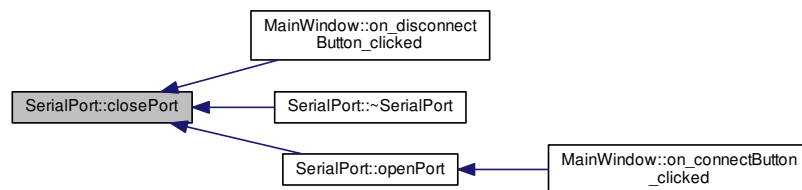
9.14.3 Member Function Documentation

9.14.3.1 `bool SerialPort::closePort ()`

[SerialPort::closePort](#).

Returns

Here is the caller graph for this function:



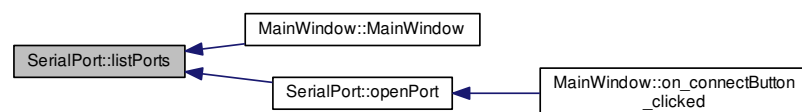
9.14.3.2 `QList< QSerialPortInfo > SerialPort::listPorts ()`

[SerialPort::listPorts](#) Obtains and return the list of available ports.

Returns

`QList<QSerialPortInfo>`

Here is the caller graph for this function:



9.14.3.3 bool SerialPort::openPort (const QString *portName*)

[SerialPort::openPort](#) Opens a serial port for reading. If port name is empty, it does nothing. If port name is not available. Does nothing and raises a warning.

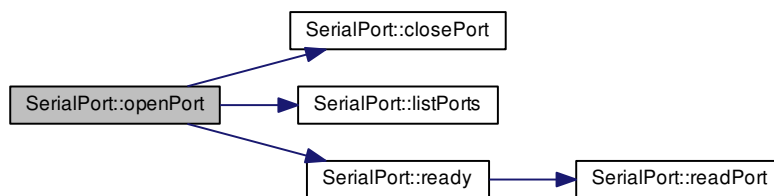
Parameters

<i>portName</i>	Name of the port as returned from QSerialPortInfo.
-----------------	--

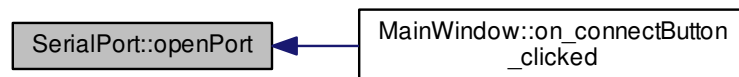
Returns

If successful, returns 1. Else it returns 0;

Here is the call graph for this function:



Here is the caller graph for this function:



9.14.3.4 void SerialPort::readPort () [private]

[SerialPort::readPort](#) Reads the available data.

Here is the caller graph for this function:



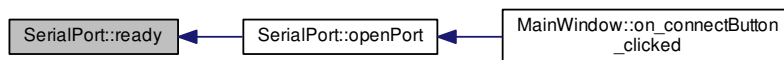
9.14.3.5 void SerialPort::ready () [slot]

[SerialPort::ready](#) Re-emits the readyRead signal.

Here is the call graph for this function:



Here is the caller graph for this function:



9.14.3.6 `void SerialPort::readyRead (QByteArray buffer)` [signal]

Here is the caller graph for this function:



9.14.4 Member Data Documentation

9.14.4.1 `QSerialPort* SerialPort::activePort` [private]

9.14.4.2 `const QSerialPort::BaudRate SerialPort::BAUDRATE =QSerialPort::Baud4800` [static],[private]

9.14.4.3 `QByteArray SerialPort::buffer` [private]

9.14.4.4 `const QSerialPort::DataBits SerialPort::DATABITS =QSerialPort::Data8` [static],[private]

9.14.4.5 `bool SerialPort::isOpen` [private]

9.14.4.6 `const QSerialPort::OpenMode SerialPort::MODE =QSerialPort::ReadOnly` [private]

9.14.4.7 `const QSerialPort::Parity SerialPort::PARITY =QSerialPort::NoParity` [static],[private]

9.14.4.8 `QSerialPortInfo* SerialPort::ports` [private]

9.14.4.9 `QMetaObject::Connection SerialPort::readConnect` [private]

9.14.4.10 `const QSerialPort::StopBits SerialPort::STOPBITS =QSerialPort::OneStop` `[static],[private]`

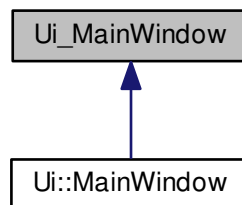
The documentation for this class was generated from the following files:

- [serialport.h](#)
- build-multimeterGUI-Desktop-Debug/[moc_serialport.cpp](#)
- [serialport.cpp](#)

9.15 Ui_MainWindow Class Reference

```
#include <ui_mainwindow.h>
```

Inheritance diagram for Ui_MainWindow:



Public Member Functions

- void [setupUi](#) (QMainWindow *[MainWindow](#))
- void [retranslateUi](#) (QMainWindow *[MainWindow](#))

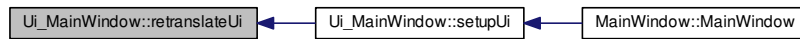
Public Attributes

- QWidget * [centralWidget](#)
- QGridLayout * [gridLayout](#)
- QHBoxLayout * [horizontalLayout](#)
- QVBoxLayout * [verticalLayout_2](#)
- QPushButton * [connectButton](#)
- QPushButton * [disconnectButton](#)
- QVBoxLayout * [verticalLayout](#)
- QLabel * [labelPort](#)
- QComboBox * [comboBoxPort](#)
- QGraphicsView * [graphPlot](#)
- QMenuBar * [menuBar](#)
- QToolBar * [mainToolBar](#)
- QStatusBar * [statusBar](#)
- QToolBar * [toolBar](#)

9.15.1 Member Function Documentation

9.15.1.1 void Ui_MainWindow::retranslateUi (QMainWindow * *MainWindow*) [inline]

Here is the caller graph for this function:



9.15.1.2 void Ui_MainWindow::setupUi (QMainWindow * *MainWindow*) [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



9.15.2 Member Data Documentation

9.15.2.1 QWidget* Ui_MainWindow::centralWidget

9.15.2.2 QComboBox* Ui_MainWindow::comboBoxPort

9.15.2.3 QPushButton* Ui_MainWindow::connectButton

9.15.2.4 QPushButton* Ui_MainWindow::disconnectButton

9.15.2.5 QGraphicsView* Ui_MainWindow::graphPlot

9.15.2.6 QGridLayout* Ui_MainWindow::gridLayout

9.15.2.7 QHBoxLayout* Ui_MainWindow::horizontalLayout

9.15.2.8 `QLabel* Ui_MainWindow::labelPort`

9.15.2.9 `QToolBar* Ui_MainWindow::mainToolBar`

9.15.2.10 `QMenuBar* Ui_MainWindow::menuBar`

9.15.2.11 `QStatusBar* Ui_MainWindow::statusBar`

9.15.2.12 `QToolBar* Ui_MainWindow::toolBar`

9.15.2.13 `QVBoxLayout* Ui_MainWindow::verticalLayout`

9.15.2.14 `QVBoxLayout* Ui_MainWindow::verticalLayout_2`

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

- `build-multimeterGUI-Desktop-Debug/ui_mainwindow.h`

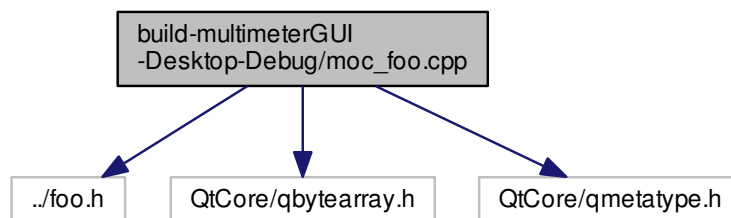
Chapter 10

File Documentation

10.1 build-multimeterGUI-Desktop-Debug/moc_datars22812.cpp File Reference

10.2 build-multimeterGUI-Desktop-Debug/moc_foo.cpp File Reference

```
#include "../foo.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
Include dependency graph for moc_foo.cpp:
```



Classes

- struct `qt_meta_stringdata_Foo_t`

Macros

- `#define QT_MOC_LITERAL(idx, ofs, len)`

Variables

- static const `qt_meta_stringdata_Foo_t qt_meta_stringdata_Foo`
- static const uint `qt_meta_data_Foo []`

10.2.1 Macro Definition Documentation

10.2.1.1 `#define QT_MOC_LITERAL(idx, ofs, len)`

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_Foo_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

10.2.2 Variable Documentation

10.2.2.1 `const uint qt_meta_data_Foo[]` `[static]`

Initial value:

```
= {
    7,
    0,
    0,    0,
    0,    0,
    0,    0,
    0,    0,
    0,    0,
    0,    0,
    0,
    0,
    0
}
```

10.2.2.2 `const qt_meta_stringdata_Foo_t qt_meta_stringdata_Foo` `[static]`

Initial value:

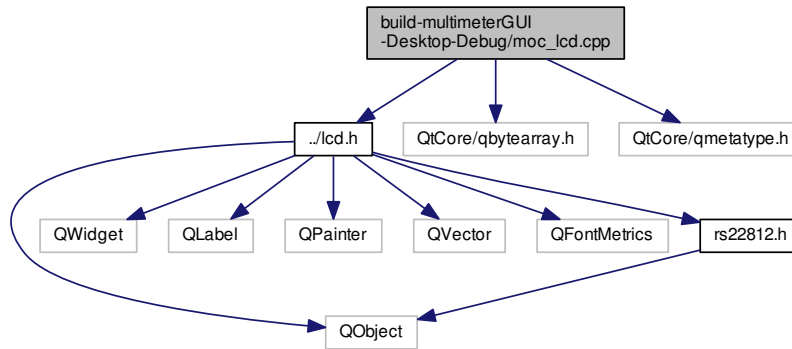
```
= {
    {
        QT_MOC_LITERAL(0, 0, 3)

    },
    "Foo"
}
```

10.3 build-multimeterGUI-Desktop-Debug/moc_lcd.cpp File Reference

```
#include "../lcd.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_lcd.cpp:



Classes

- struct [qt_meta_stringdata_LCD_t](#)

Macros

- `#define` [QT_MOC_LITERAL](#)(idx, ofs, len)

Variables

- static const [qt_meta_stringdata_LCD_t](#) [qt_meta_stringdata_LCD](#)
- static const uint [qt_meta_data_LCD](#) []

10.3.1 Macro Definition Documentation

10.3.1.1 `#define` [QT_MOC_LITERAL](#)(idx, ofs, len)

Value:

```

Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt\_meta\_stringdata\_LCD\_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )

```

10.3.2 Variable Documentation

10.3.2.1 `const` uint [qt_meta_data_LCD](#)[] [static]

Initial value:

```

= {

    7,
    0,
    0,    0,
    0,    0,

```

```

    0,    0,
    0,    0,
    0,    0,
    0,
    0,
    0,
    0
}

```

10.3.2.2 `const qt_meta_stringdata_LCD_t qt_meta_stringdata_LCD [static]`

Initial value:

```

= {
{
QT_MOC_LITERAL(0, 0, 3)

},
"LCD"
}

```

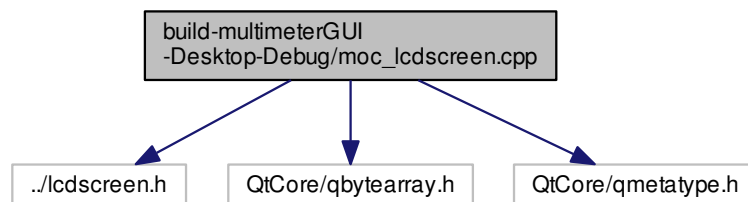
10.4 build-multimeterGUI-Desktop-Debug/moc_lcdscreen.cpp File Reference

```

#include "../lcdscreen.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>

```

Include dependency graph for moc_lcdscreen.cpp:



Classes

- struct `qt_meta_stringdata_LCDscreen_t`

Macros

- `#define QT_MOC_LITERAL(idx, ofs, len)`

Variables

- static const `qt_meta_stringdata_LCDscreen_t qt_meta_stringdata_LCDscreen`
- static const uint `qt_meta_data_LCDscreen []`

10.4.1 Macro Definition Documentation

10.4.1.1 #define QT_MOC_LITERAL(*idx*, *ofs*, *len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_LCDscreen_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

10.4.2 Variable Documentation

10.4.2.1 `const uint qt_meta_data_LCDscreen[]` `[static]`

Initial value:

$$= \{$$

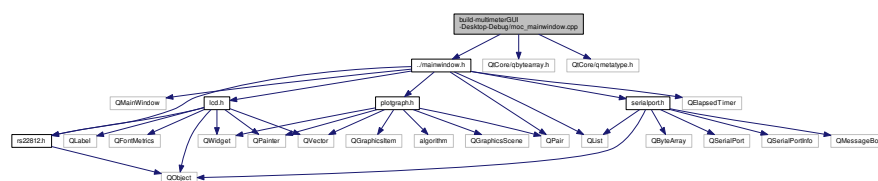
10.4.2.2 `const qt_meta_stringdata_LCDscreen t qt_meta_stringdata_LCDscreen [static]`

Initial value:

```
= {
    {
        QT_MOC_LITERAL(0, 0, 9)
    },
    "LCDscreen"
}
```

10.5 build-multimeterGUI-Desktop-Debug/moc_mainwindow.cpp File Reference

```
#include "../mainwindow.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
Include dependency graph for moc_mainwindow.cpp:
```



Classes

- struct [qt_meta_stringdata_MainWindow_t](#)

Macros

- `#define QT_MOC_LITERAL(idx, ofs, len)`

Variables

- static const [qt_meta_stringdata_MainWindow_t](#) [qt_meta_stringdata_MainWindow](#)
- static const uint [qt_meta_data_MainWindow](#) []

10.5.1 Macro Definition Documentation

10.5.1.1 `#define QT_MOC_LITERAL(idx, ofs, len)`

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_MainWindow_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

10.5.2 Variable Documentation

10.5.2.1 `const uint qt_meta_data_MainWindow[]` [static]

Initial value:

```
= {
    7,
    0,
    0, 0,
    4, 14,
    0, 0,
    0, 0,
    0, 0,
    0,
    0,
    1, 0, 34, 2, 0x08 ,
    3, 0, 35, 2, 0x08 ,
    4, 0, 36, 2, 0x08 ,
    5, 0, 37, 2, 0x08 ,

    QMetaType::Void,
    QMetaType::Void,
    QMetaType::Void,
    QMetaType::Void,
    0
}
```

10.5.2.2 `const qt_meta_stringdata_MainWindow_t qt_meta_stringdata_MainWindow` [static]

Initial value:

```

= {
{
    QT_MOC_LITERAL(0, 0, 10),
    QT_MOC_LITERAL(1, 11, 24),
    QT_MOC_LITERAL(2, 36, 0),
    QT_MOC_LITERAL(3, 37, 27),
    QT_MOC_LITERAL(4, 65, 7),
    QT_MOC_LITERAL(5, 73, 9)
},
    "MainWindow\0on_connectButton_clicked\0"
    "\0on_disconnectButton_clicked\0addData\0"
    "resetData"
}

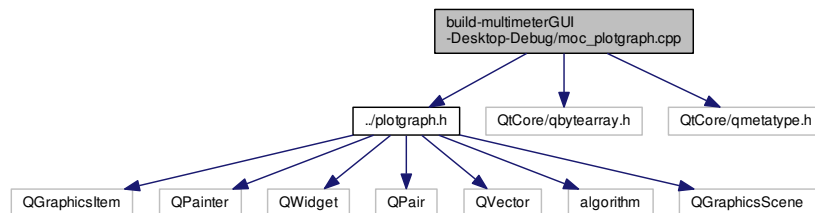
```

10.6 build-multimeterGUI-Desktop-Debug/moc_plotgraph.cpp File Reference

```

#include "../plotgraph.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
Include dependency graph for moc_plotgraph.cpp:

```



Classes

- struct [qt_meta_stringdata_plotGraph_t](#)

Macros

- #define [QT_MOC_LITERAL](#)(idx, ofs, len)

Variables

- static const [qt_meta_stringdata_plotGraph_t](#) [qt_meta_stringdata_plotGraph](#)
- static const uint [qt_meta_data_plotGraph](#) []

10.6.1 Macro Definition Documentation

10.6.1.1 #define QT_MOC_LITERAL(idx, ofs, len)

Value:

```

Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt\_meta\_stringdata\_plotGraph\_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )

```

10.6.2 Variable Documentation

10.6.2.1 `const uint qt_meta_data_plotGraph[]` `[static]`

Initial value:

```
= {
    7,
    0,
    0,    0,
    0,    0,
    0,    0,
    0,    0,
    0,    0,
    4,
    0,

    0
}
```

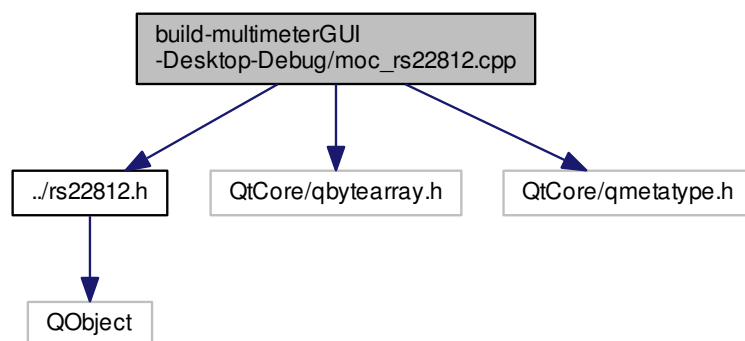
10.6.2.2 `const qt_meta_stringdata_plotGraph_t qt_meta_stringdata_plotGraph` `[static]`

Initial value:

```
= {
    {
        QT_MOC_LITERAL(0, 0, 9)
    },
    "plotGraph"
}
```

10.7 build-multimeterGUI-Desktop-Debug/moc_rs22812.cpp File Reference

```
#include "../rs22812.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
Include dependency graph for moc_rs22812.cpp:
```



Classes

- struct [qt_meta_stringdata_RS22812_t](#)

Macros

- #define [QT_MOC_LITERAL](#)(idx, ofs, len)

Variables

- static const
[qt_meta_stringdata_RS22812_t](#) qt_meta_stringdata_RS22812
- static const uint [qt_meta_data_RS22812](#) []

10.7.1 Macro Definition Documentation

10.7.1.1 #define QT_MOC_LITERAL(idx, ofs, len)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt\_meta\_stringdata\_RS22812\_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

10.7.2 Variable Documentation

10.7.2.1 const uint qt_meta_data_RS22812[] [static]

Initial value:

```
= {

    7,
    0,
    0,    0,
    3,    14,
    0,    0,
    0,    0,
    0,    0,
    0,    0,
    0,
    2,

    1,    0,    29,    2, 0x06 ,
    3,    0,    30,    2, 0x06 ,

    4,    1,    31,    2, 0x0a ,

    QMetaType::Void,
    QMetaType::Void,

    QMetaType::Void, QMetaType::QByteArray,    5,

    0
}
```

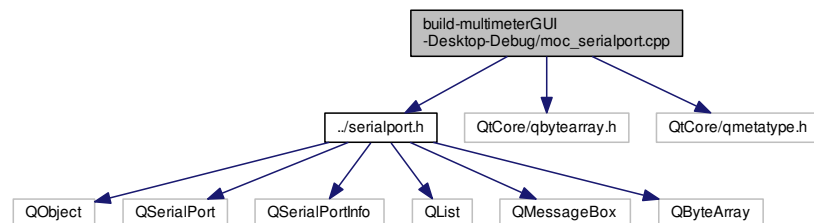
10.7.2.2 `const qt_meta_stringdata_RS22812_t qt_meta_stringdata_RS22812 [static]`

Initial value:

```
= {
{
QT_MOC_LITERAL(0, 0, 7),
QT_MOC_LITERAL(1, 8, 7),
QT_MOC_LITERAL(2, 16, 0),
QT_MOC_LITERAL(3, 17, 7),
QT_MOC_LITERAL(4, 25, 8),
QT_MOC_LITERAL(5, 34, 4)
},
"RS22812\0newMode\0\0newData\0newValue\0"
"data"
}
```

10.8 build-multimeterGUI-Desktop-Debug/moc_serialport.cpp File Reference

```
#include "../serialport.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
Include dependency graph for moc_serialport.cpp:
```



Classes

- struct [qt_meta_stringdata_SerialPort_t](#)

Macros

- `#define` [QT_MOC_LITERAL](#)(idx, ofs, len)

Variables

- static const [qt_meta_stringdata_SerialPort_t qt_meta_stringdata_SerialPort](#)
- static const uint [qt_meta_data_SerialPort \[\]](#)

10.8.1 Macro Definition Documentation

10.8.1.1 `#define QT_MOC_LITERAL(idx, ofs, len)`

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_SerialPort_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

10.8.2 Variable Documentation

10.8.2.1 `const uint qt_meta_data_SerialPort[]` [static]

Initial value:

```
= {
    7,
    0,
    0,    0,
    2,    14,
    0,    0,
    0,    0,
    0,    0,
    0,
    1,

    1,    1,    24,    2, 0x06 ,

    4,    0,    27,    2, 0x0a ,

    QMetaType::Void, QMetaType::QByteArray,    3,

    QMetaType::Void,

    0
}
```

10.8.2.2 `const qt_meta_stringdata_SerialPort_t qt_meta_stringdata_SerialPort` [static]

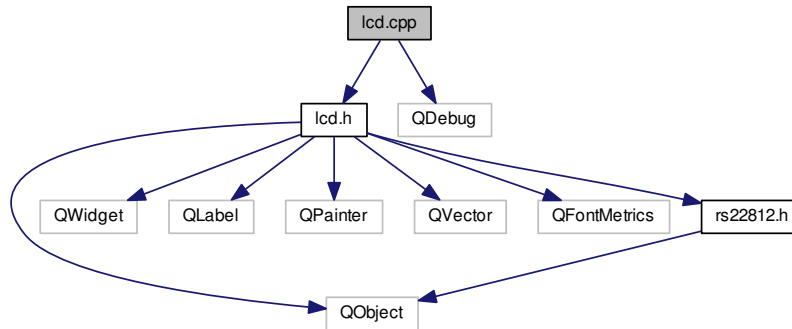
Initial value:

```
= {
    {
        QT_MOC_LITERAL(0, 0, 10),
        QT_MOC_LITERAL(1, 11, 9),
        QT_MOC_LITERAL(2, 21, 0),
        QT_MOC_LITERAL(3, 22, 6),
        QT_MOC_LITERAL(4, 29, 5)
    },
    "SerialPort\0readyRead\0\0buffer\0ready"
}
```



```
#include <QDebug>
```

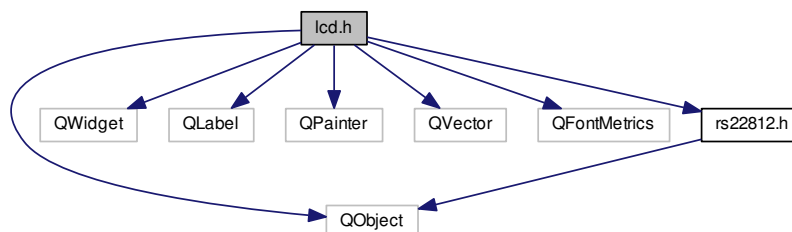
Include dependency graph for lcd.cpp:



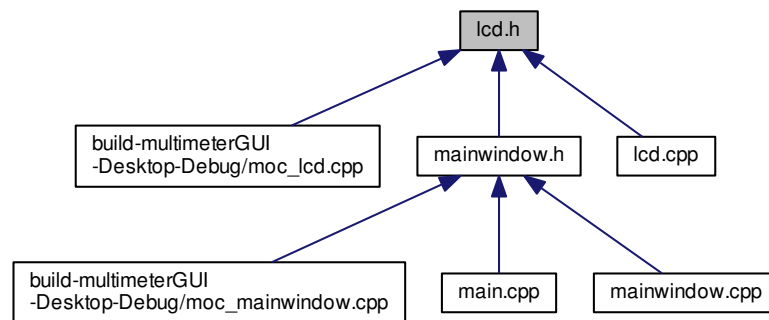
10.11 lcd.h File Reference

```
#include <QObject>
#include <QWidget>
#include <QLabel>
#include <QPainter>
#include <QVector>
#include <QFontMetrics>
#include "rs22812.h"
```

Include dependency graph for lcd.h:



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



Classes

- class [LCD](#)

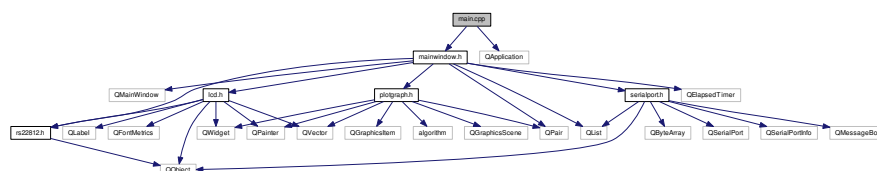
The [LCD](#) class displays the numerical value read.

10.12 main.cpp File Reference

```
#include "mainwindow.h"
```

```
#include <QApplication>
```

Include dependency graph for `main.cpp`:



Functions

- int [main](#) (int argc, char *argv[])

10.12.1 Function Documentation

10.12.1.1 int `main` (int *argc*, char * *argv*[])

10.13 mainwindow.cpp File Reference

```
#include "mainwindow.h"
```

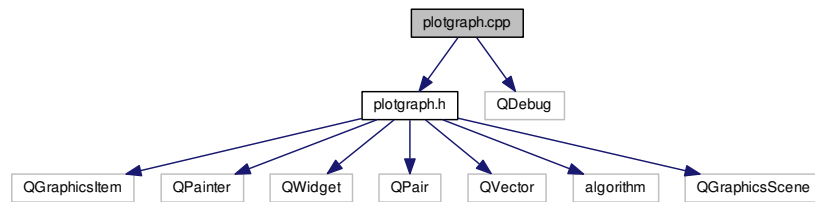
```
#include "ui_mainwindow.h"
```

```
#include <QDebug>
```



```
#include <QDebug>
```

Include dependency graph for plotgraph.cpp:



10.16 plotgraph.h File Reference

```
#include <QGraphicsItem>
```

```
#include <QPainter>
```

```
#include <QWidget>
```

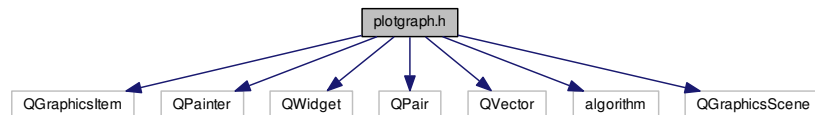
```
#include <QPair>
```

```
#include <QVector>
```

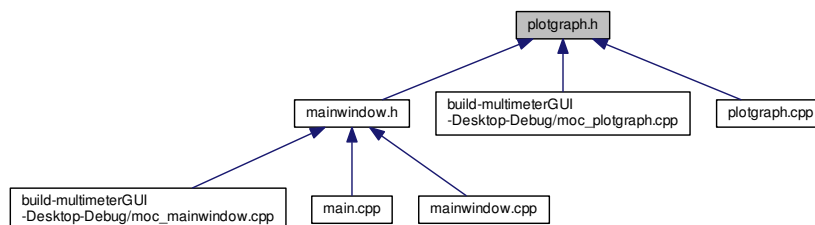
```
#include <algorithm>
```

```
#include <QGraphicsScene>
```

Include dependency graph for plotgraph.h:



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



Classes

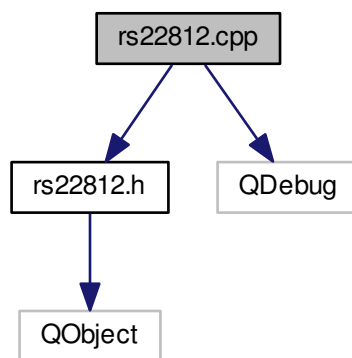
- class [plotGraph](#)

The [plotGraph](#) class.

10.17 README.md File Reference

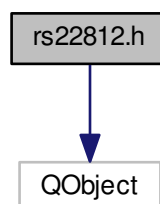
10.18 rs22812.cpp File Reference

```
#include "rs22812.h"  
#include <QDebug>  
Include dependency graph for rs22812.cpp:
```

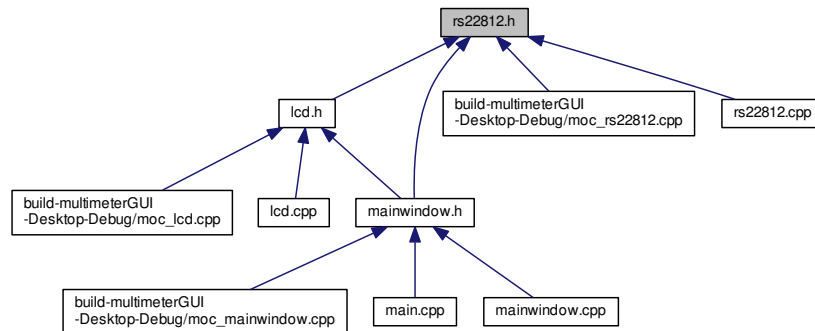


10.19 rs22812.h File Reference

```
#include <QObject>  
Include dependency graph for rs22812.h:
```



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



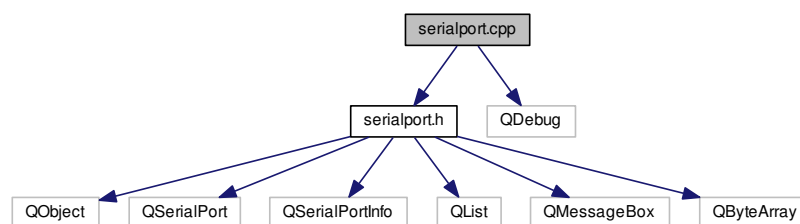
Classes

- struct [Flags](#)
Definition of custom data type.
- class [RS22812](#)
Decoding of the data sent by the Radio Shack 22-812.

10.20 serialport.cpp File Reference

```
#include "serialport.h"
#include <QDebug>
```

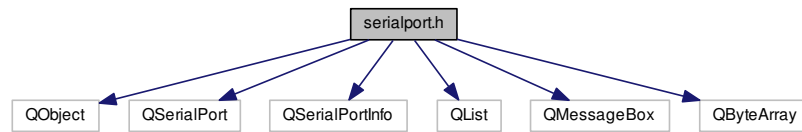
Include dependency graph for serialport.cpp:



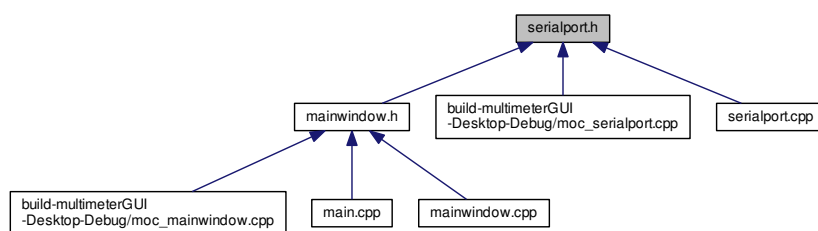
10.21 serialport.h File Reference

```
#include <QObject>
#include <QSerialPort>
#include <QSerialPortInfo>
#include <QList>
#include <QMessageBox>
#include <QByteArray>
```

Include dependency graph for serialport.h:



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



Classes

- class [SerialPort](#)
Class to manage the communication with a serial port.

Index

~MainWindow
 MainWindow, [25](#)
~SerialPort
 SerialPort, [47](#)

A
 Flags, [18](#)
activePort
 SerialPort, [51](#)
addData
 MainWindow, [25](#)
Auto
 Flags, [18](#)
axisMargin
 plotGraph, [35](#)

BAUDRATE
 SerialPort, [51](#)
bRect
 plotGraph, [35](#)

Bat
 Flags, [18](#)
Beep
 Flags, [18](#)
boundingRect
 plotGraph, [30](#)

buffer
 SerialPort, [51](#)
build-multimeterGUI-Desktop-Debug/moc_datars22812.-
 cpp, [55](#)
build-multimeterGUI-Desktop-Debug/moc_foo.cpp, [55](#)
build-multimeterGUI-Desktop-Debug/moc_lcd.cpp, [56](#)
build-multimeterGUI-Desktop-Debug/moc_lcdscreen.-
 cpp, [58](#)
build-multimeterGUI-Desktop-Debug/moc_mainwindow.-
 cpp, [59](#)
build-multimeterGUI-Desktop-Debug/moc_plotgraph.-
 cpp, [61](#)
build-multimeterGUI-Desktop-Debug/moc_rs22812.cpp,
 [62](#)
build-multimeterGUI-Desktop-Debug/moc_serialport.-
 cpp, [64](#)
build-multimeterGUI-Desktop-Debug/ui_mainwindow.h,
 [66](#)
byte2Digit
 RS22812, [41](#)

centralWidget
 Ui_MainWindow, [53](#)
checksum

 Flags, [18](#)
closePort
 SerialPort, [48](#)
comboBoxPort
 Ui_MainWindow, [53](#)
connectButton
 Ui_MainWindow, [53](#)
continuity
 Flags, [18](#)
counter
 MainWindow, [27](#)

DATABITS
 SerialPort, [51](#)

dBm
 Flags, [18](#)

DP
 Flags, [18](#)

dash
 Flags, [18](#)

data
 LCD, [21](#)
 plotGraph, [35](#)
 qt_meta_stringdata_Foo_t, [36](#)
 qt_meta_stringdata_LCD_t, [36](#)
 qt_meta_stringdata_LCDscreen_t, [37](#)
 qt_meta_stringdata_MainWindow_t, [37](#)
 qt_meta_stringdata_plotGraph_t, [37](#)
 qt_meta_stringdata_RS22812_t, [38](#)
 qt_meta_stringdata_SerialPort_t, [38](#)

digits
 RS22812, [45](#)

Diode
 Flags, [18](#)

disconnectButton
 Ui_MainWindow, [53](#)

F
 Flags, [18](#)

Flags, [17](#)
 A, [18](#)
 Auto, [18](#)
 Bat, [18](#)
 Beep, [18](#)
 checksum, [18](#)
 continuity, [18](#)
 dBm, [18](#)
 DP, [18](#)
 dash, [18](#)
 Diode, [18](#)

- F, 18
- hFE, 18
- Hold, 18
- Hz, 18
- k, 18
- M, 19
- m, 19
- MAX, 19
- MIN, 19
- n, 19
- Ohms, 19
- operator==, 18
- percent, 19
- REL, 19
- RS232, 19
- s, 19
- tilde, 19
- u, 19
- V, 19
- flags
 - RS22812, 45
- getDigitString
 - RS22812, 41
- getFlags
 - RS22812, 42
- getMode
 - RS22812, 42
- getVal
 - RS22812, 42
- graph
 - MainWindow, 27
- graphPlot
 - Ui_MainWindow, 53
- gridLayout
 - Ui_MainWindow, 53
- hFE
 - Flags, 18
- Hold
 - Flags, 18
- horizontalLayout
 - Ui_MainWindow, 53
- Hz
 - Flags, 18
- isOpen
 - SerialPort, 51
- k
 - Flags, 18
- LCD, 19
 - data, 21
 - LCD, 20
 - lbl, 21
 - LCD, 20
 - paintEvent, 21
- label
 - MainWindow, 27
- labelPort
 - Ui_MainWindow, 53
- labelXaxis
 - plotGraph, 30
- labelYaxis
 - plotGraph, 31
- lbl
 - LCD, 21
- lcd
 - MainWindow, 27
- lcd.cpp, 66
- lcd.h, 67
- linkData
 - plotGraph, 31
- listPorts
 - SerialPort, 48
- M
 - Flags, 19
- m
 - Flags, 19
- MAX
 - Flags, 19
- MIN
 - Flags, 19
- MODE
 - SerialPort, 51
- main
 - main.cpp, 68
- main.cpp, 68
- main, 68
- mainToolBar
 - Ui_MainWindow, 54
- MainWindow, 22
 - ~MainWindow, 25
 - addData, 25
 - counter, 27
 - graph, 27
 - label, 27
 - lcd, 27
 - MainWindow, 24
 - MainWindow, 24
 - maxData, 27
 - minData, 28
 - newData, 28
 - on_connectButton_clicked, 26
 - on_disconnectButton_clicked, 26
 - portPtr, 28
 - rData, 28
 - rawdata, 28
 - resetData, 27
 - scene, 28
 - storeData, 28
 - timeMark, 28
 - timeRunning, 28
 - tmp, 28
 - ui, 28
- mainwindow.cpp, 68

- mainwindow.h, 69
- maxData
 - MainWindow, 27
- menuBar
 - Ui_MainWindow, 54
- minData
 - MainWindow, 28
- moc_foo.cpp
 - QT_MOC_LITERAL, 56
 - qt_meta_data_Foo, 56
 - qt_meta_stringdata_Foo, 56
- moc_lcd.cpp
 - QT_MOC_LITERAL, 57
 - qt_meta_data_LCD, 57
 - qt_meta_stringdata_LCD, 58
- moc_lcdscreen.cpp
 - QT_MOC_LITERAL, 59
 - qt_meta_data_LCDscreen, 59
 - qt_meta_stringdata_LCDscreen, 59
- moc_mainwindow.cpp
 - QT_MOC_LITERAL, 60
 - qt_meta_data_MainWindow, 60
 - qt_meta_stringdata_MainWindow, 60
- moc_plotgraph.cpp
 - QT_MOC_LITERAL, 61
 - qt_meta_data_plotGraph, 62
 - qt_meta_stringdata_plotGraph, 62
- moc_rs22812.cpp
 - QT_MOC_LITERAL, 63
 - qt_meta_data_RS22812, 63
 - qt_meta_stringdata_RS22812, 63
- moc_serialport.cpp
 - QT_MOC_LITERAL, 64
 - qt_meta_data_SerialPort, 65
 - qt_meta_stringdata_SerialPort, 65
- mode
 - RS22812, 45
- modeChanged
 - RS22812, 42
- n
 - Flags, 19
- newData
 - MainWindow, 28
 - RS22812, 43
- newMode
 - RS22812, 43
- newValue
 - RS22812, 43
- nx
 - plotGraph, 35
- ny
 - plotGraph, 35
- Ohms
 - Flags, 19
- oldflags
 - RS22812, 45
- on_connectButton_clicked
 - MainWindow, 26
- on_disconnectButton_clicked
 - MainWindow, 26
- openPort
 - SerialPort, 48
- operator==
 - Flags, 18
- origin
 - plotGraph, 35
- PARITY
 - SerialPort, 51
- paint
 - plotGraph, 31
- paintAxis
 - plotGraph, 32
- paintEvent
 - LCD, 21
- percent
 - Flags, 19
- plotData
 - plotGraph, 32
- plotGraph, 28
 - axisMargin, 35
 - bRect, 35
 - boundingRect, 30
 - data, 35
 - labelXaxis, 30
 - labelYaxis, 31
 - linkData, 31
 - nx, 35
 - ny, 35
 - origin, 35
 - paint, 31
 - paintAxis, 32
 - plotData, 32
 - plotGraph, 30
 - plotGraph, 30
 - real2Coord, 33
 - rightX, 35
 - scene, 35
 - setScene, 33
 - setUnit, 34
 - setXaxis, 34
 - setXsticks, 34
 - setYaxis, 34
 - setYsticks, 35
 - unit, 35
 - upperY, 35
 - xmax, 36
 - xmin, 36
 - ymax, 36
 - ymin, 36
- plotgraph.cpp, 69
- plotgraph.h, 70
- portPtr
 - MainWindow, 28
- ports
 - SerialPort, 51

- QT_MOC_LITERAL
 - moc_foo.cpp, [56](#)
 - moc_lcd.cpp, [57](#)
 - moc_lcdscreen.cpp, [59](#)
 - moc_mainwindow.cpp, [60](#)
 - moc_plotgraph.cpp, [61](#)
 - moc_rs22812.cpp, [63](#)
 - moc_serialport.cpp, [64](#)
- qt_meta_data_Foo
 - moc_foo.cpp, [56](#)
- qt_meta_data_LCD
 - moc_lcd.cpp, [57](#)
- qt_meta_data_LCDscreen
 - moc_lcdscreen.cpp, [59](#)
- qt_meta_data_MainWindow
 - moc_mainwindow.cpp, [60](#)
- qt_meta_data_RS22812
 - moc_rs22812.cpp, [63](#)
- qt_meta_data_SerialPort
 - moc_serialport.cpp, [65](#)
- qt_meta_data_plotGraph
 - moc_plotgraph.cpp, [62](#)
- qt_meta_stringdata_Foo
 - moc_foo.cpp, [56](#)
- qt_meta_stringdata_Foo_t, [36](#)
 - data, [36](#)
 - stringdata0, [36](#)
- qt_meta_stringdata_LCD
 - moc_lcd.cpp, [58](#)
- qt_meta_stringdata_LCD_t, [36](#)
 - data, [36](#)
 - stringdata0, [36](#)
- qt_meta_stringdata_LCDscreen
 - moc_lcdscreen.cpp, [59](#)
- qt_meta_stringdata_LCDscreen_t, [37](#)
 - data, [37](#)
 - stringdata0, [37](#)
- qt_meta_stringdata_MainWindow
 - moc_mainwindow.cpp, [60](#)
- qt_meta_stringdata_MainWindow_t, [37](#)
 - data, [37](#)
 - stringdata0, [37](#)
- qt_meta_stringdata_RS22812
 - moc_rs22812.cpp, [63](#)
- qt_meta_stringdata_RS22812_t, [38](#)
 - data, [38](#)
 - stringdata0, [38](#)
- qt_meta_stringdata_SerialPort
 - moc_serialport.cpp, [65](#)
- qt_meta_stringdata_SerialPort_t, [38](#)
 - data, [38](#)
 - stringdata0, [38](#)
- qt_meta_stringdata_plotGraph
 - moc_plotgraph.cpp, [62](#)
- qt_meta_stringdata_plotGraph_t, [37](#)
 - data, [37](#)
 - stringdata0, [37](#)
- rData
 - MainWindow, [28](#)
- README.md, [71](#)
- REL
 - Flags, [19](#)
- RS22812, [38](#)
 - byte2Digit, [41](#)
 - digits, [45](#)
 - flags, [45](#)
 - getDigitString, [41](#)
 - getFlags, [42](#)
 - getMode, [42](#)
 - getVal, [42](#)
 - mode, [45](#)
 - modeChanged, [42](#)
 - newData, [43](#)
 - newMode, [43](#)
 - newValue, [43](#)
 - oldflags, [45](#)
 - RS22812, [40](#)
 - resetFlags, [45](#)
 - RS22812, [40](#)
- RS232
 - Flags, [19](#)
- rawdata
 - MainWindow, [28](#)
- readConnect
 - SerialPort, [51](#)
- readPort
 - SerialPort, [50](#)
- ready
 - SerialPort, [50](#)
- readyRead
 - SerialPort, [51](#)
- real2Coord
 - plotGraph, [33](#)
- resetData
 - MainWindow, [27](#)
- resetFlags
 - RS22812, [45](#)
- retranslateUi
 - Ui_MainWindow, [53](#)
- rightX
 - plotGraph, [35](#)
- rs22812.cpp, [71](#)
- rs22812.h, [71](#)
- s
 - Flags, [19](#)
- STOPBITS
 - SerialPort, [51](#)
- scene
 - MainWindow, [28](#)
 - plotGraph, [35](#)
- SerialPort, [46](#)
 - ~SerialPort, [47](#)
 - activePort, [51](#)
 - BAUDRATE, [51](#)
 - buffer, [51](#)
 - closePort, [48](#)

- DATABITS, 51
- isOpen, 51
- listPorts, 48
- MODE, 51
- openPort, 48
- PARITY, 51
- ports, 51
- readConnect, 51
- readPort, 50
- ready, 50
- readyRead, 51
- STOPBITS, 51
- SerialPort, 47
- SerialPort, 47
- serialport.cpp, 72
- serialport.h, 72
- setScene
 - plotGraph, 33
- setUnit
 - plotGraph, 34
- setXaxis
 - plotGraph, 34
- setXsticks
 - plotGraph, 34
- setYaxis
 - plotGraph, 34
- setYsticks
 - plotGraph, 35
- setupUi
 - Ui_MainWindow, 53
- statusBar
 - Ui_MainWindow, 54
- storeData
 - MainWindow, 28
- stringdata0
 - qt_meta_stringdata_Foo_t, 36
 - qt_meta_stringdata_LCD_t, 36
 - qt_meta_stringdata_LCDscreen_t, 37
 - qt_meta_stringdata_MainWindow_t, 37
 - qt_meta_stringdata_plotGraph_t, 37
 - qt_meta_stringdata_RS22812_t, 38
 - qt_meta_stringdata_SerialPort_t, 38
- tilde
 - Flags, 19
- timeMark
 - MainWindow, 28
- timeRunning
 - MainWindow, 28
- tmp
 - MainWindow, 28
- toolBar
 - Ui_MainWindow, 54
- u
 - Flags, 19
- Ui, 15
- ui
 - MainWindow, 28
- Ui::MainWindow, 21
- Ui_MainWindow, 52
 - centralWidget, 53
 - comboBoxPort, 53
 - connectButton, 53
 - disconnectButton, 53
 - graphPlot, 53
 - gridLayout, 53
 - horizontalLayout, 53
 - labelPort, 53
 - mainToolBar, 54
 - menuBar, 54
 - retranslateUi, 53
 - setupUi, 53
 - statusBar, 54
 - toolBar, 54
 - verticalLayout, 54
 - verticalLayout_2, 54
- unit
 - plotGraph, 35
- upperY
 - plotGraph, 35
- V
 - Flags, 19
- verticalLayout
 - Ui_MainWindow, 54
- verticalLayout_2
 - Ui_MainWindow, 54
- xmax
 - plotGraph, 36
- xmin
 - plotGraph, 36
- ymax
 - plotGraph, 36
- ymin
 - plotGraph, 36