Multimeter GUI

Generated by Doxygen 1.8.6

Sat May 7 2016 17:02:58

Contents

Chapter 1

Main Page

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.

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:

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

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

Chapter 6

Class Index

6.1 Class List

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

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

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

f1	
f2	

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:

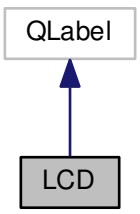
- rs22812.h

9.2 LCD Class Reference

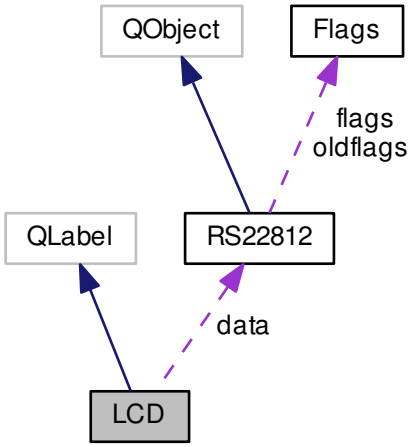
The LCD class displays the numerical value read.

`#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

data	
parent	

9.2.3 Member Function Documentation

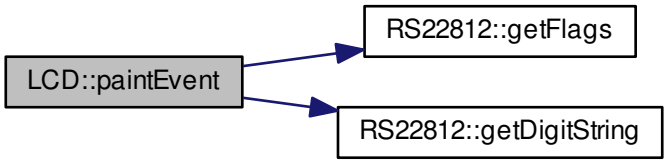
9.2.3.1 void LCD::paintEvent (QPaintEvent * *event*) [protected]

LCD::paintEvent. Paint event handler.

Parameters

event	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]

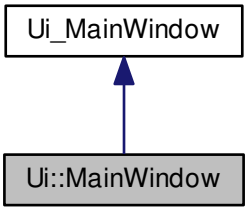
The documentation for this class was generated from the following files:

- lcd.h
- lcd.cpp

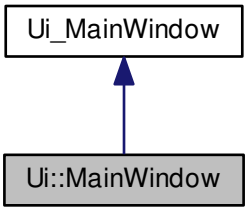
9.3 Ui::MainWindow Class Reference

```
#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:

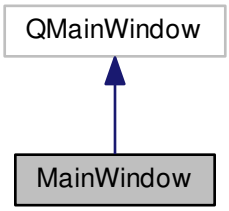
- build-multimeterGUI-Desktop-Debug/ui_mainwindow.h

9.4 MainWindow Class Reference

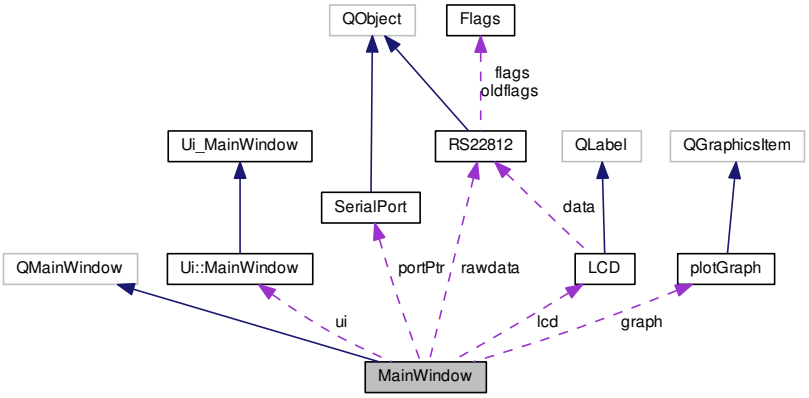
The MainWindow class.

`#include <mainwindow.h>`

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Public Member Functions

- MainWindow (QWidget *parent=0)

  MainWindow::MainWindow. Constructor.
- ~MainWindow ()

  MainWindow::~MainWindow.

Private Slots

- void on_connectButton_clicked ()

  MainWindow::on_connectButton_clicked. Open the selected port.
- void on_disconnectButton_clicked ()

  MainWindow::on_disconnectButton_clicked. Disconnect from the current port.
- void addData ()

  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.
- void resetData ()

  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.

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]

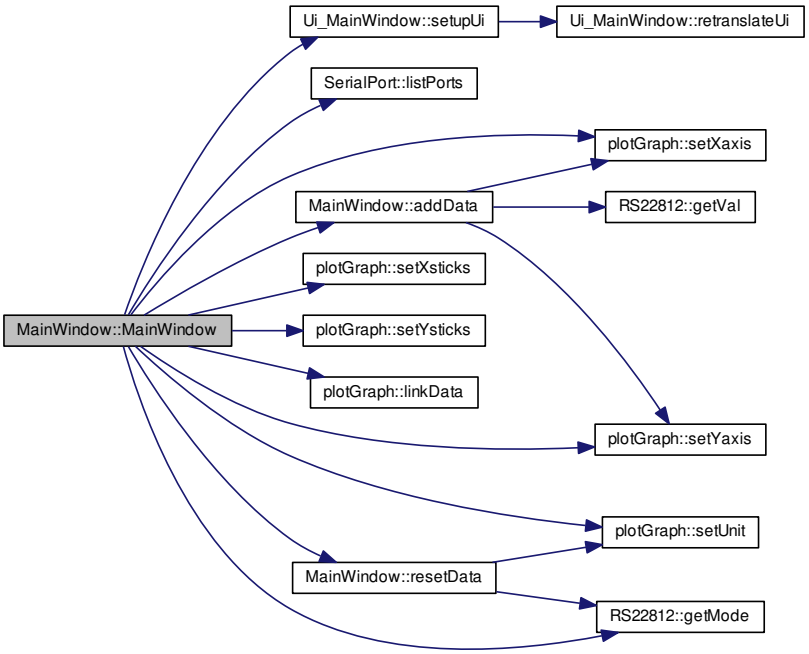
MainWindow::MainWindow. Constructor.

Parameters

parent	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 ( )

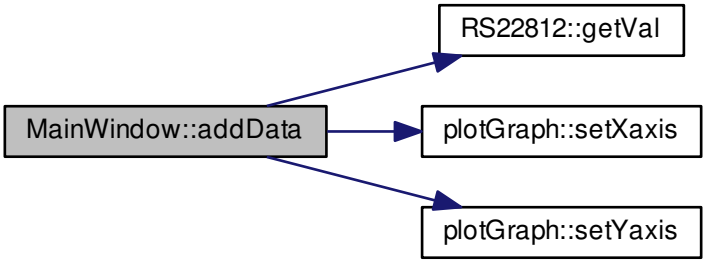
MainWindow::~MainWindow.

9.4.3 Member Function Documentation

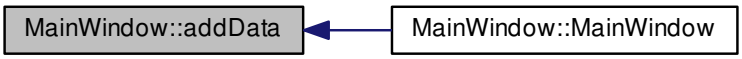
9.4.3.1 void MainWindow::addData ( ) [private], [slot]

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:



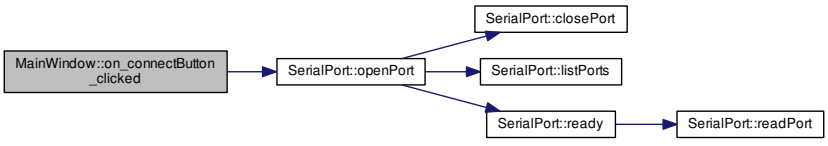
Here is the caller graph for this function:



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

MainWindow::on_connectButton_clicked. Open the selected port.

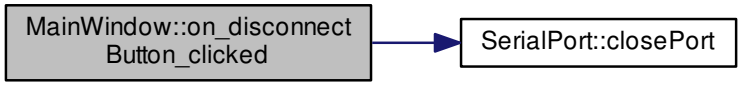
Here is the call graph for this function:



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

MainWindow::on_disconnectButton_clicked. Disconnect from the current port.

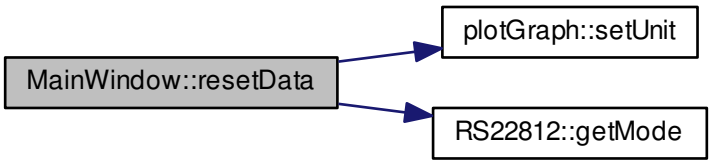
Here is the call graph for this function:



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 QMetaObject::Connection MainWindow::newData [private]

9.4.4.8 SerialPort* MainWindow::portPtr [private]

9.4.4.9 RS22812* MainWindow::rawdata [private]

9.4.4.10 QMetaObject::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:

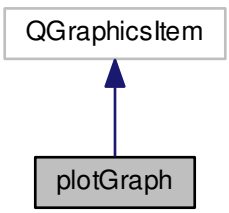
- mainwindow.h
- mainwindow.cpp

9.5 plotGraph Class Reference

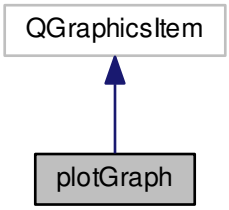
The plotGraph class.

`#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.

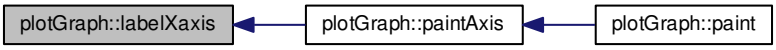
9.5.3.2 void plotGraph::labelXaxis (QPainter * *painter*, QPoint & *p1*, QPoint & *p2*) [private]

plotGraph::labelXaxis Add labels to the X axis.

Parameters

painter	
p1	
p2	

Here is the caller graph for this function:



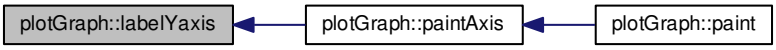
9.5.3.3 void plotGraph::labelYaxis (QPainter * *painter*, QPoint & *p1*, QPoint & *p2*) [private]

plotGraph::labelYaxis Adds labels to the Y axis.

Parameters

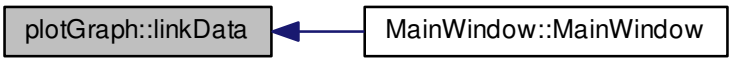
painter	
p1	
p2	

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:



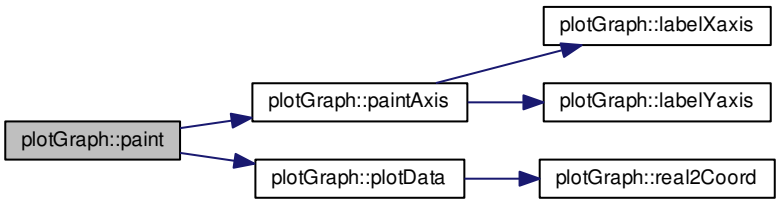
9.5.3.5 void plotGraph::paint (QPainter * *painter*, const QStyleOptionGraphicsItem * *option*, QWidget * *widget*)

plotGraph::paint. Override of the paint method of QGraphicsScene.

Parameters

painter	
option	
widget	

Here is the call graph for this function:



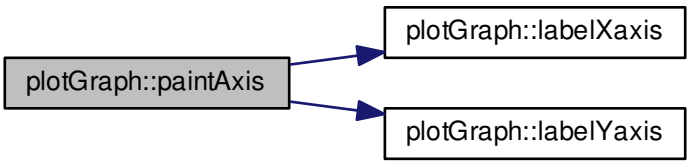
9.5.3.6 void plotGraph::paintAxis (QPainter * *painter*, const QStyleOptionGraphicsItem * *option*, QWidget * *widget*) [private]

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

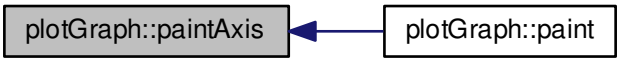
Parameters

painter	
option	
widget	

Here is the call graph for this function:



Here is the caller graph for this function:



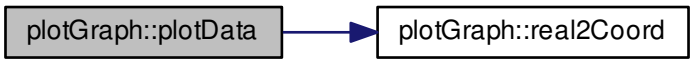
9.5.3.7 void plotGraph::plotData (QPainter * *painter,* const QStyleOptionGraphicsItem * *option,* QWidget * *widget*) [private]

plotGraph::plotData Draws the data on the widget.

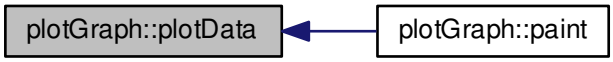
Parameters

painter	
option	
widget	

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]

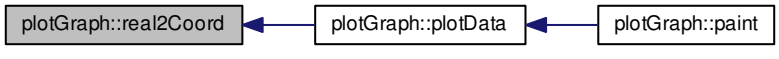
plotGraph::real2Coord Transform reading coordinates to widget coordinates.

Parameters

dpoint	

Returns

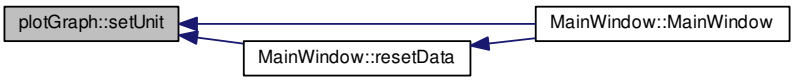
Here is the caller graph for this function:



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:



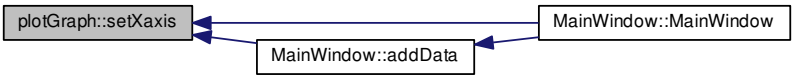
9.5.3.11 void plotGraph::setXaxis (qint64 *minVal*, qint64 *maxVal*)

plotGraph::setXaxis. Sets the maximum and minimum values of the x axis.

Parameters

minVal	
maxVal	

Here is the caller graph for this function:



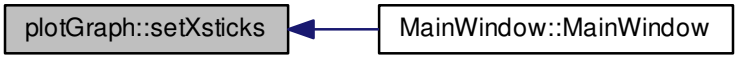
9.5.3.12 void plotGraph::setXsticks (int *nSticks*)

plotGraph::setXsticks

Parameters

nSticks	

Here is the caller graph for this function:



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

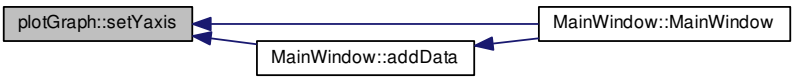
plotGraph::setYaxis

Sets the maximum and minimum values of the y axis.

Parameters

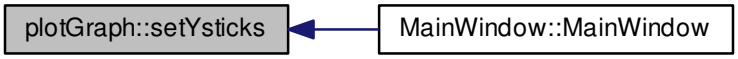
minVal	
maxVal	

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:

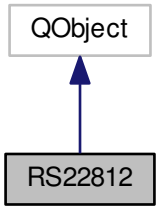
- build-multimeterGUI-Desktop-Debug/moc_serialport.cpp

9.13 RS22812 Class Reference

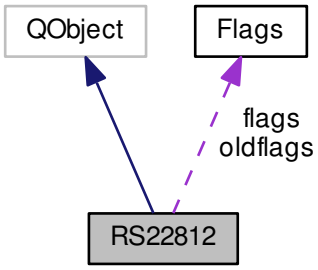
Decoding of the data sent by the Radio Shack 22-812.

`#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

byte	RS 22-812 byte value

Returns

String with the equivalent digit.

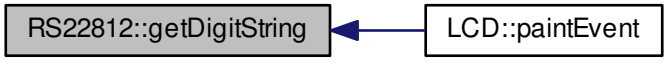
Here is the caller graph for this function:

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:

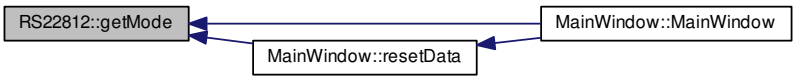
9.13.3.4 uint RS22812::getMode ( ) const

RS22812::getMode.

Returns

It returns the mode on which the multimeter is working.

Here is the caller graph for this function:

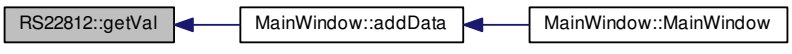


9.13.3.5 float RS22812::getVal ( ) const

RS22812::getVal Returns the numeric value of the reading.

Returns

Here is the caller graph for this function:

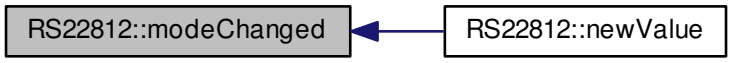


9.13.3.6 bool RS22812::modeChanged ( ) [private]

RS22812::modeChanged Checks the flags structure to see whether the read mode changed.

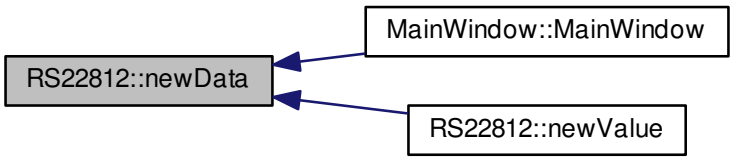
Returns

Here is the caller graph for this function:



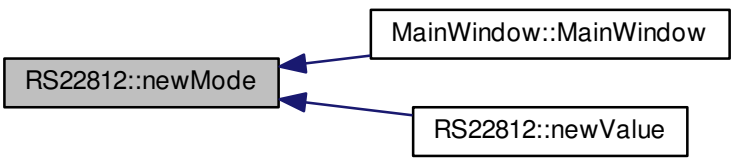
9.13.3.7 void RS22812::newData ( ) [signal]

Here is the caller graph for this function:



9.13.3.8 void RS22812::newMode ( ) [signal]

Here is the caller graph for this function:



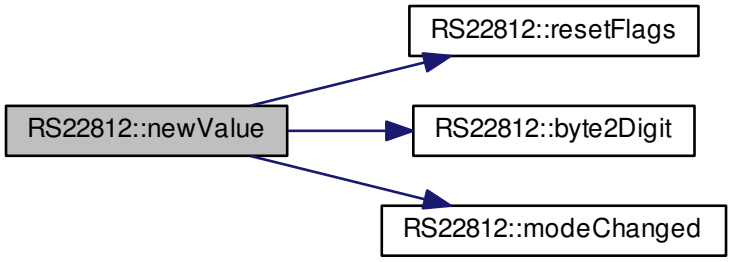
9.13.3.9 void RS22812::newValue (const QByteArray & *data*) [slot]

RS22812::newValue Reads a new packet and reformats the data.

Parameters

data	

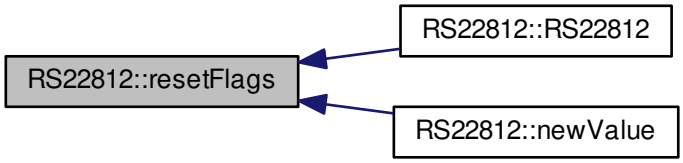
Here is the call graph for this function:



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:

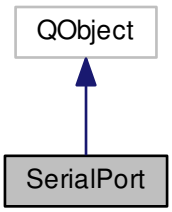
- rs22812.h
- build-multimeterGUI-Desktop-Debug/moc_rs22812.cpp
- rs22812.cpp

9.14 SerialPort Class Reference

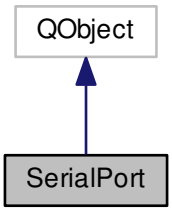
Class to manage the communication with a serial port.

`#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

parent	

9.14.2.2 SerialPort::~SerialPort ( )

SerialPort::~SerialPort.

Here is the call graph for this function:

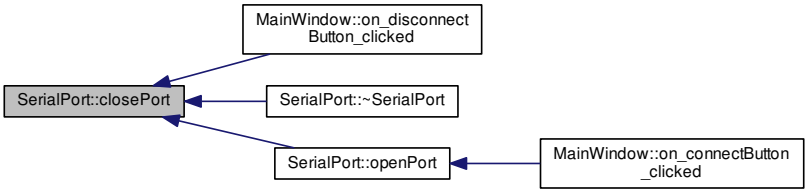
9.14.3 Member Function Documentation

9.14.3.1 bool SerialPort::closePort ( )

SerialPort::closePort.

Returns

Here is the caller graph for this function:



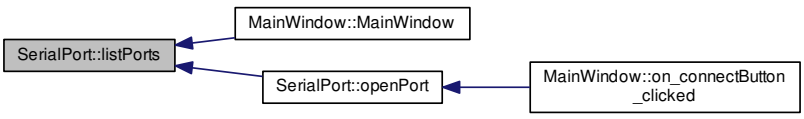
9.14.3.2 QList< QSerialPortInfo > SerialPort::listPorts ( )

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.

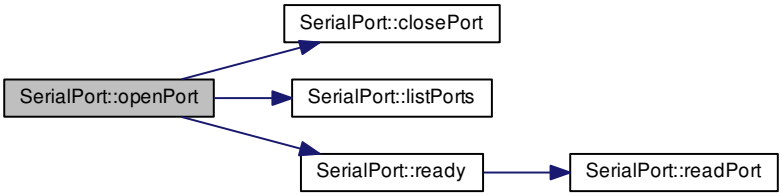
Parameters

portName	Name of the port as returned from QSerialPortInfo.

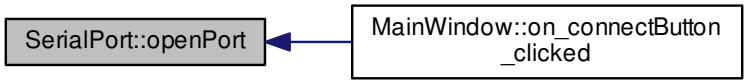
Returns

If sucessful, returns 1. Else it returns 0;

Here is the call graph for this function:



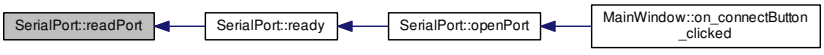
Here is the caller graph for this function:



9.14.3.4 void SerialPort::readPort ( ) [private]

SerialPort::readPort Reads the available data.

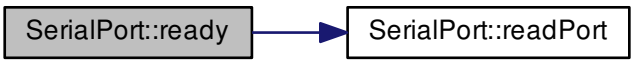
Here is the caller graph for this function:



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]

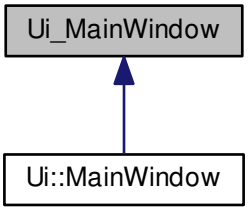
The documentation for this class was generated from the following files:

- 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)

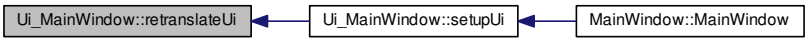
Public Attributes

- 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

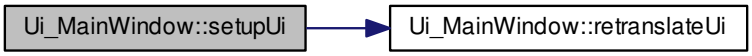
9.15.1.1 void Ui_MainWindow::retranslateUi (QMainWindow * *MainWindow*) [inline]

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

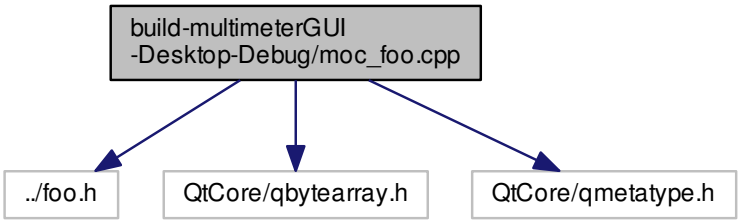
File Documentation

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
}
```

10.2.2.2 const qt_meta_stringdata_Foo_t qt_meta_stringdata_Foo [static]

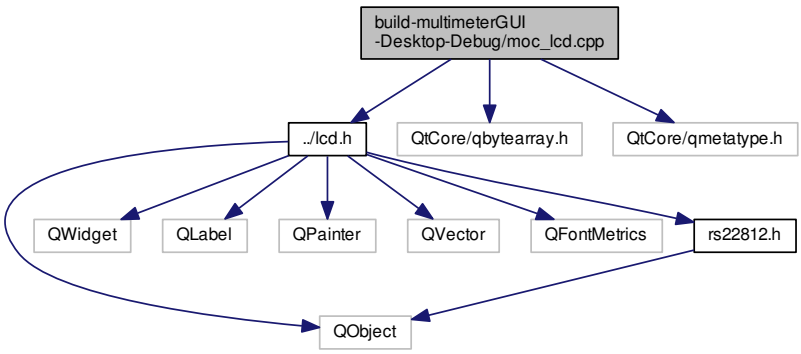
Initial value:

```
= {
    {
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
}
```

10.3.2.2 const qt_meta_stringdata_LCD_t qt_meta_stringdata_LCD [static]

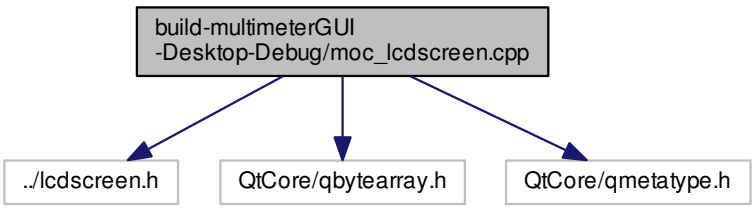
Initial value:

```
= {
    {
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:

```
= {

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

       0
}
```

10.4.2.2 const qt_meta_stringdata_LCDscreen_t qt_meta_stringdata_LCDscreen [static]

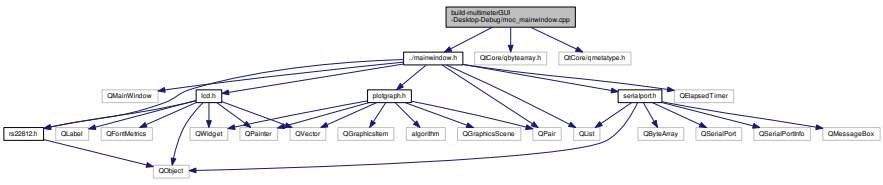
Initial value:

```
= {
    {
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]

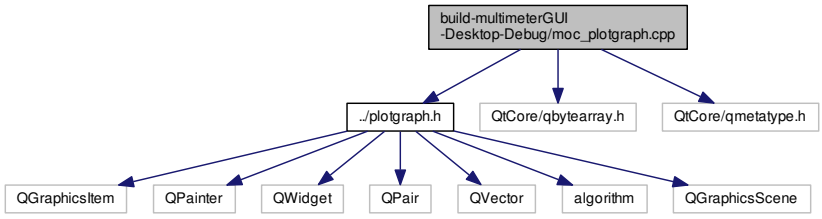
Initial value:

```
= {
    {
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]

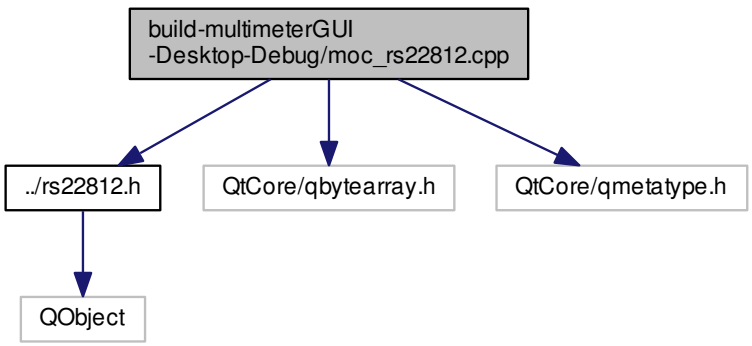
Initial value:

```
= {
    {
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,
       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]

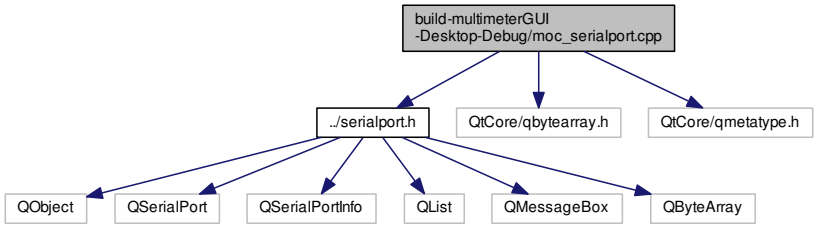
Initial value:

```
= {
    {
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:

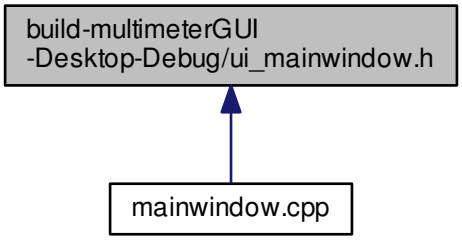
```
= {
    {
QT_MOC_LITERAL(0, 0, 10),
QT_MOC_LITERAL(1, 11, 9),
QT_MOC_LITERAL(2, 21, 0),
QT_MOC_LITERAL(3, 22, 6),
QT_MOC_LITERAL(4, 29, 5)

    },
    "SerialPort\0readyRead\0\0buffer\0ready"
}
```

10.9 build-multimeterGUI-Desktop-Debug/ui_mainwindow.h File Reference

```
#include <QtCore/QVariant>
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QGraphicsView>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QMainWindow>
#include <QtWidgets/QMenuBar>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QStatusBar>
#include <QtWidgets/QToolBar>
#include <QtWidgets/QVBoxLayout>
#include <QtWidgets/QWidget>
```

Include dependency graph for ui_mainwindow.h:

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



Classes

- class Ui_MainWindow
- class Ui::MainWindow

Namespaces

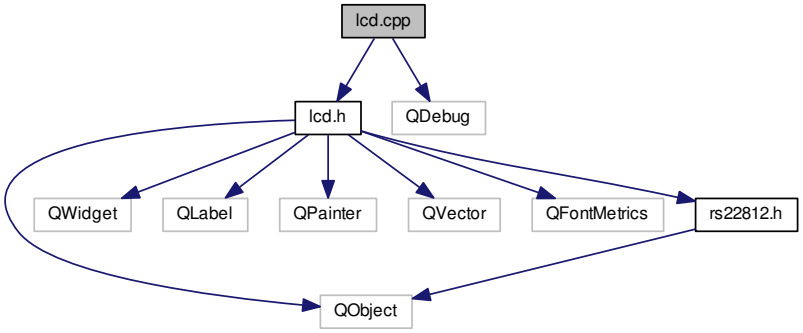
- Ui

10.10 lcd.cpp File Reference

```
#include "lcd.h"
```

```
#include <QDebug>
```

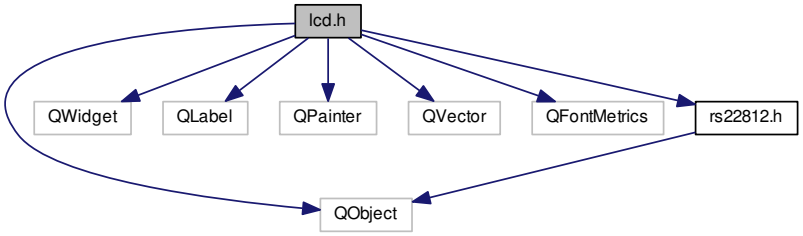
Include dependency graph for lcd.cpp:



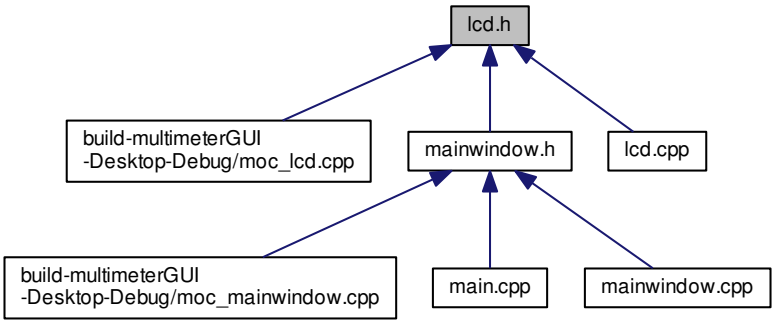
10.11 lcd.h File Reference

```
#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

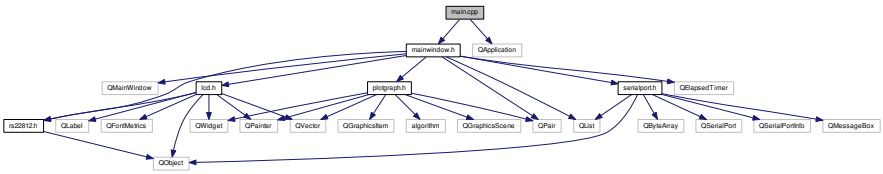
- class LCD

  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[])

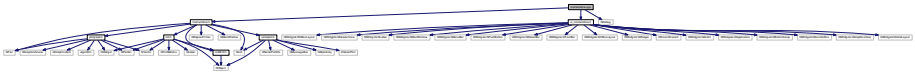
10.12.1 Function Documentation

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>
```

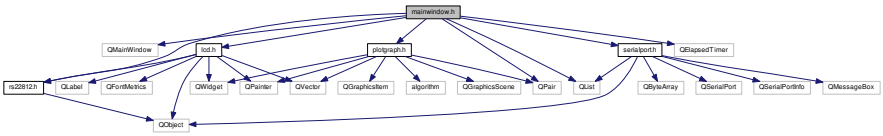
Include dependency graph for mainwindow.cpp:



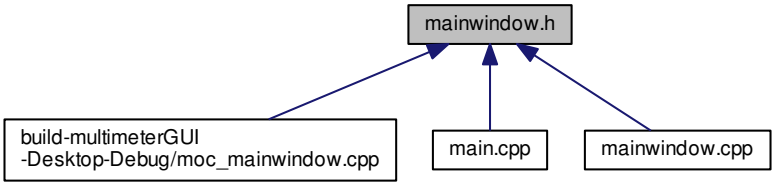
10.14 mainwindow.h File Reference

```
#include <QMainWindow>
#include <QList>
#include "serialport.h"
#include "rs22812.h"
#include "lcd.h"
#include "plotgraph.h"
#include <QPair>
#include <QElapsedTimer>
```

Include dependency graph for mainwindow.h:



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



Classes

- class MainWindow

  The MainWindow class.

Namespaces

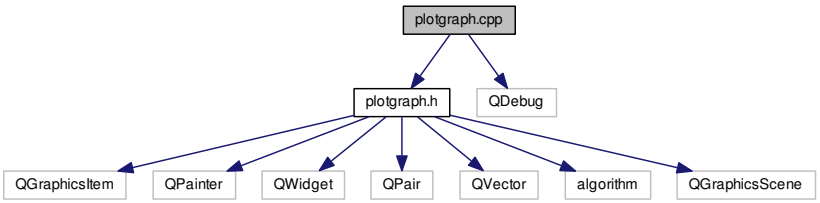
- Ui

10.15 plotgraph.cpp File Reference

```
#include "plotgraph.h"
```

```
#include <QDebug>
```

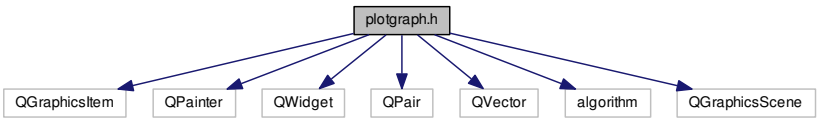
Include dependency graph for plotgraph.cpp:



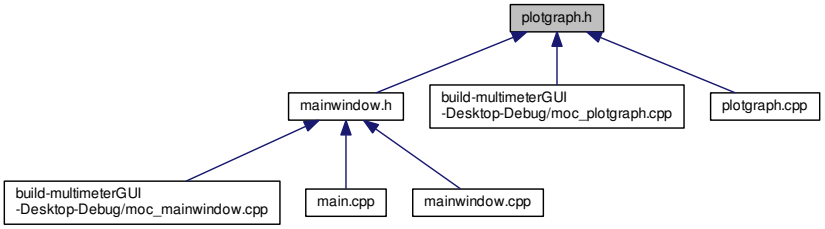
10.16 plotgraph.h File Reference

```
#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

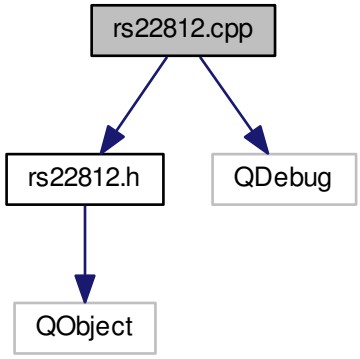
  The plotGraph class.

10.17 README.md File Reference

10.18 rs22812.cpp File Reference

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

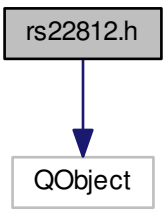
Include dependency graph for rs22812.cpp:



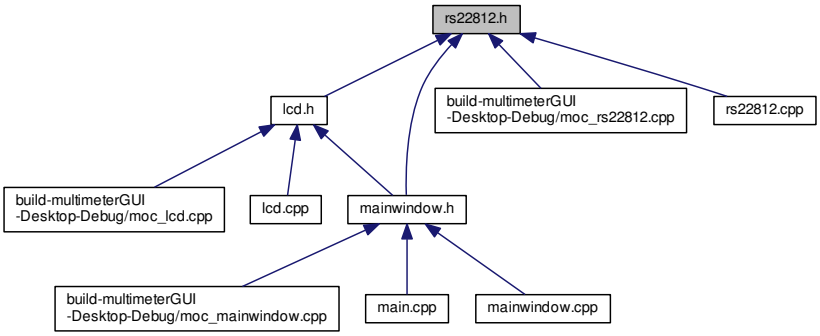
10.19 rs22812.h File Reference

```
#include <QObject>
```

Include dependency graph for rs22812.h:



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



Classes

- struct Flags

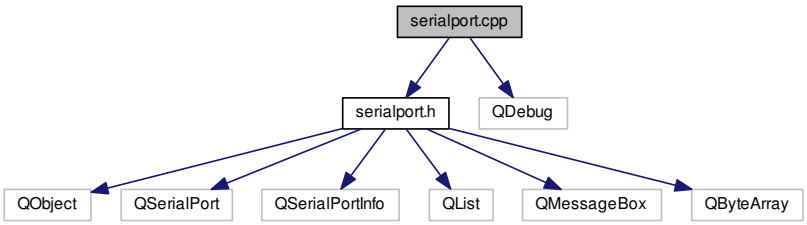
  Definition of custom data type.
- class RS22812

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

10.20 serialport.cpp File Reference

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

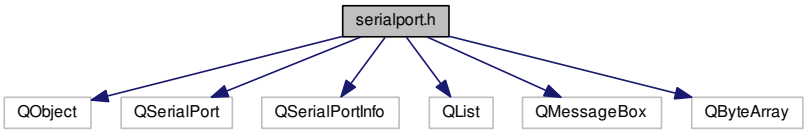
Include dependency graph for serialport.cpp:



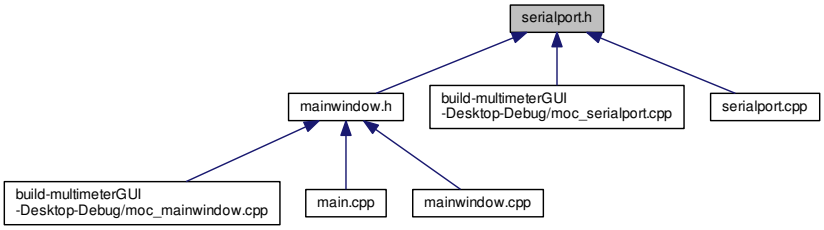
10.21 serialport.h File Reference

```
#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