Krobosoft

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 aclFiles Class Reference

Public Member Functions

- aclFiles ()

  aclFiles Constructor for the class.
- ~aclFiles ()

  ~aclFiles Destructor for the class.
- void setFilename (bool type=0)

  setFilename setFilename open a pop-up windows to select a file. The name is stored at filename.
- QString getFilename ()

  getFilename Getter for the private attribute filename.
- QString getFilename (QString filepath)

  getFilename Getter for the base name of a file pointed by filepath
- QStringList getFilenames ()

  getFilenames Getter for the private attribute filenames
- QString getFilePath ()

  getFilePath Getter for the private attribute filePath
- QStringList getFilePaths ()

  getFilePaths Getter for the private attribute filePaths
- bool saveToFile (QString content)

  Save content of QString into the file pointed by filename.
- QString openFile (QString filepath)

  openFile Gets the content of a file pointed by filepath

3.1.1 Member Function Documentation

3.1.1.1 getFilename() [1/2]

```
QString aclFiles::getFilename ( )
```

getFilename Getter for the private attribute filename.

Returns

QString The content of the attribute filename.

3.1.1.2 getFilename() [2/2]

```
QString aclFiles::getFilename (
            QString filepath )
```

getFilename Getter for the base name of a file pointed by filepath

Parameters

filepath	QString The path for a file

Returns

QString the base name of the file pointed by filepath

3.1.1.3 getFilenames()

```
QStringList aclFiles::getFilenames ( )
```

getFilenames Getter for the private attribute filenames

Returns

QStringList the content of the attribute filenames

3.1.1.4 getFilePath()

```
QString aclFiles::getFilePath ( )
```

getFilePath Getter for the private attribute filePath

Returns

QString the content of the attribute filePath

3.1.1.5 getFilePaths()

```
QStringList aclFiles::getFilePaths ( )
```

getFilePaths Getter for the private attribute filePaths

Returns

QStringList the content of the attribute filePaths

3.1.1.6 openFile()

```
QString aclFiles::openFile (
            QString filepath )
```

openFile Gets the content of a file pointed by filepath

Parameters

filepath	The path to a given file

Returns

QString the content of a file

3.1.1.7 saveToFile()

```
bool aclFiles::saveToFile (
            QString content )
```

Save content of QString into the file pointed by filename.

Parameters

content	QString The content of the text-edit object to be saved.

Returns

bool Confirms if the saving has occurrely succesfully.

3.1.1.8 setFilename()

```
void aclFiles::setFilename (
            bool type = 0 )
```

setFilename setFilename open a pop-up windows to select a file. The name is stored at filename.

Parameters

type	bool Decide if the dialog will be for openning (allow multiple files) ou saving (one file only)

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

- aclfiles.h
- aclfiles.cpp

3.2 MainWindow Class Reference

Inheritance diagram for MainWindow:

3.3 SerialCom Class Reference

Inheritance diagram for SerialCom:

Collaboration diagram for SerialCom:

Public Slots

- int send (const QString &msg)

  Sends a QString to the open COM port.
- int sendByte (const char &byte)

  sendByte
- bool connect (const QString &COMName)

  Used to open communication with a COM port.

Signals

- void readyRead ()

  Used as Signal to know when Serial data are available to read.

Public Member Functions

- SerialCom (qint32 baudRate=QSerialPort::Baud9600, QSerialPort::DataBits dataBytes=QSerialPort::Data8, QSerialPort::Parity parity=QSerialPort::NoParity, QSerialPort::FlowControl flowControl=QSerialPort::No↩ FlowControl, QSerialPort::StopBits stopBit=QSerialPort::OneStop)

  Constructor function, it reeds all COM ports, stores it's names, then configure the instance of QSerialPort to future use. Whit exception of baudRate, all other parameters are enums defined in QSerialPort class.
- QList< QString > getCOMList ()

  Reads the open COM ports, iterates through them and extracts it's names.
- void close ()

  Closes the COM port.
- QString read ()

  Reads all available data on the current COM port.
- bool portListChanged ()

  Function used to know if changes in the list of COM ports occurred.

3.3.1 Constructor & Destructor Documentation

3.3.1.1 SerialCom()

```
SerialCom::SerialCom (
            qint32 baudRate = QSerialPort::Baud9600,
            QSerialPort::DataBits dataBytes = QSerialPort::Data8,
            QSerialPort::Parity parity = QSerialPort::NoParity,
            QSerialPort::FlowControl flowControl = QSerialPort::NoFlowControl,
            QSerialPort::StopBits stopBit = QSerialPort::OneStop )
```

Constructor function, it reeds all COM ports, stores it's names, then configure the instance of QSerialPort to future use. Whit exception of baudRate, all other parameters are enums defined in QSerialPort class.

Parameters

baudRate	qint32 Baud rate of the port.
dataBytes	QSerialPort::DataBits Number of data bits.
parity	QSerialPort::Parity Parity type.
flowControl	QSerialPort::FlowControl Type of flow control.
stopBit	QSerialPort::StopBits Set stop bit configuration.

3.3.2 Member Function Documentation

3.3.2.1 connect

```
bool SerialCom::connect (
            const QString & COMName )  [slot]
```

Used to open communication with a COM port.

Parameters

COMName	const QString & Qstring that contain the name of the desired COM port.

Returns

Returns true if it succeeds in establish the connection, false otherwise.

3.3.2.2 getCOMList()

```
QList< QString > SerialCom::getCOMList ( )
```

Reads the open COM ports, iterates through them and extracts it's names.

Returns

QList<QString> Return the list of COM ports existents.

3.3.2.3 portListChanged()

```
bool SerialCom::portListChanged ( )
```

Function used to know if changes in the list of COM ports occurred.

Returns

Returns true if COM ports were added or subtracted, false otherwise.

3.3.2.4 read()

```
QString SerialCom::read ( )
```

Reads all available data on the current COM port.

Returns

Return COM buffer data.

3.3.2.5 send

```
int SerialCom::send (
            const QString & msg )  [slot]
```

Sends a QString to the open COM port.

Parameters

msg	const QString & The QString to be sent.

Returns

int Returns true if it succeeds is sending the message, false otherwise.

3.3.2.6 sendByte

```
int SerialCom::sendByte (
            const char & byte )  [slot]
```

sendByte

Parameters

byte	const char & Byte to be sent.

Returns

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

- serialcom.h
- serialcom.cpp

3.4 SerialTerminal Class Reference

Inheritance diagram for SerialTerminal:

Collaboration diagram for SerialTerminal:

Public Slots

- void setCommand (const QString &newCommand)

  It change the current command to be sent. Used to iterate through the log entries.

Signals

- void emitDataReady (const QString &data)

  Signal emitted when user press Enter or Return.
- void emitByteReady (const char &byte)

  Sgnal emitted user press a single key.
- void emitGetLog (const int &linesFromEnd)

  Requests the log object to send a log line.
- void emitStoreCommand (const QString &command)

  emitStoreCommand Signal to store a command in the log.

Public Member Functions

- SerialTerminal ()

  Sets the color of the terminal.
- QString getCommand ()

  Used to return the string to be sent to the serial Port.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗e) override

  Override function, that listens for Enter or Return presses.

Protected Attributes

- QString command

  QString that holds the command typed.
- int lastEditPosition =0

  Holds the last edition position. When the user presses a key is treated as if the cursor is in this position. This is set eatch time the writing capability is enable, and when the user presses arrow keys.
- int logLine =0

  logLine int Holds the position of the line requested to the logger.

3.4.1 Member Function Documentation

3.4.1.1 emitByteReady

```
void SerialTerminal::emitByteReady (
            const char & byte )  [signal]
```

Sgnal emitted user press a single key.

Parameters

byte	char & byte to be sent.

3.4.1.2 emitDataReady

```
void SerialTerminal::emitDataReady (
            const QString & data )  [signal]
```

Signal emitted when user press Enter or Return.

Parameters

data	const QString & The String that is to be sent out of the terminal.

3.4.1.3 emitGetLog

```
void SerialTerminal::emitGetLog (
            const int & linesFromEnd )  [signal]
```

Requests the log object to send a log line.

Parameters

linesFromEnd	const int & Position of the desired line. Counting from the end of the file (where is the last entry).

3.4.1.4 emitStoreCommand

```
void SerialTerminal::emitStoreCommand (
            const QString & command )  [signal]
```

emitStoreCommand Signal to store a command in the log.

Parameters

command	const QString & Referenc to the command string to be stored on the log.

3.4.1.5 getCommand()

```
QString SerialTerminal::getCommand ( )  [inline]
```

Used to return the string to be sent to the serial Port.

Returns

Qstring String to be sent out of the terminal.

3.4.1.6 setCommand

```
void SerialTerminal::setCommand (
            const QString & newCommand )  [slot]
```

It change the current command to be sent. Used to iterate through the log entries.

Parameters

newCommand	const QString & The new value for the command.

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

- serialcom.h
- serialcom.cpp

Index