Kaffeknekt

Generated by Doxygen 1.13.2

Chapter 1

Namespace Index

1.1 Namespace List

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

Chapter 5

Namespace Documentation

5.1 Ui Namespace Reference

Qt namespace for UI classes generated from .ui files.

5.1.1 Detailed Description

Qt namespace for UI classes generated from .ui files.

Qt namespace containing UI classes generated from .ui files.

Chapter 6

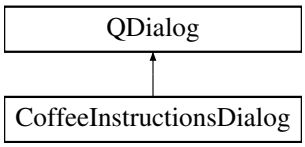
Class Documentation

6.1 CoffeeInstructionsDialog Class Reference

A dialog window that displays coffee brewing instructions.

`#include <coffeeinstructionsdialog.h>`

Inheritance diagram for CoffeeInstructionsDialog:



Public Member Functions

- CoffeeInstructionsDialog (QWidget *parent=nullptr)
  Constructor.
- ~CoffeeInstructionsDialog ()
  Destructor.

Private Attributes

- Ui::CoffeeInstructionsDialog * ui
  Pointer to the UI elements of the dialog.

6.1.1 Detailed Description

A dialog window that displays coffee brewing instructions.

This class represents a modal dialog window built using Qt, specifically for displaying instructions related to coffee preparation.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 CoffeeInstructionsDialog()

```
CoffeeInstructionsDialog::CoffeeInstructionsDialog (
            QWidget * parent = nullptr)  [explicit]
```

Constructor.

Parameters

parent	Pointer to the parent widget (default is nullptr).

Creates and initializes the CoffeeInstructionsDialog.

6.1.2.2 ~CoffeeInstructionsDialog()

```
CoffeeInstructionsDialog::~CoffeeInstructionsDialog ()
```

Destructor.

Cleans up resources used by the dialog.

6.1.3 Member Data Documentation

6.1.3.1 ui

```
Ui::CoffeeInstructionsDialog* CoffeeInstructionsDialog::ui  [private]
```

Pointer to the UI elements of the dialog.

This is generated automatically by Qt Designer (via .ui file).

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

- coffeeinstructionsdialog.h
- coffeeinstructionsdialog.cpp

6.2 DataFetcher Class Reference

Utility class for fetching sensor data from InfluxDB.

```
#include <datafetcher.h>
```

Public Member Functions

- **DataFetcher** ()

  Default constructor for DataFetcher.

Static Public Member Functions

- static QVector< DataPoint > fetchPressureWindow (qreal windowSeconds, const QString &influxUrl, const QString &token, const QString &bucket)

  Fetch pressure data points within a given time window.
- static QVector< DataPoint > fetchTempWindow (qreal windowSeconds, const QString &influxUrl, const QString &token, const QString &bucket)

  Fetch temperature data points within a given time window.

6.2.1 Detailed Description

Utility class for fetching sensor data from InfluxDB.

This class provides static methods for retrieving temperature and pressure data over a given time window from an InfluxDB source.

6.2.2 Member Function Documentation

6.2.2.1 fetchPressureWindow()

```
QVector< DataPoint > DataFetcher::fetchPressureWindow (
            qreal windowSeconds,
            const QString & influxUrl,
            const QString & token,
            const QString & bucket)  [static]
```

Fetch pressure data points within a given time window.

Parameters

windowSeconds	The size of the time window in seconds.
influxUrl	URL of the InfluxDB server.
token	Authentication token for accessing the InfluxDB.
bucket	Name of the bucket to query data from.

Returns

QVector<DataPoint> A vector of pressure data points.

6.2.2.2 fetchTempWindow()

```
QVector< DataPoint > DataFetcher::fetchTempWindow (
            qreal windowSeconds,
            const QString & influxUrl,
            const QString & token,
            const QString & bucket)  [static]
```

Fetch temperature data points within a given time window.

Parameters

windowSeconds	The size of the time window in seconds.
influxUrl	URL of the InfluxDB server.
token	Authentication token for accessing the InfluxDB.
bucket	Name of the bucket to query data from.

Returns

QVector<DataPoint> A vector of temperature data points.

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

- datafetcher.h
- datafetcher.cpp

6.3 DataPoint Struct Reference

Represents a single data sample with a timestamp and a value.

`#include <graphview.h>`

Public Attributes

- qreal **timestamp**

  Time of the data point (seconds since epoch or relative).
- qreal **value**

  Value of the data point (e.g., temperature or pressure).

6.3.1 Detailed Description

Represents a single data sample with a timestamp and a value.

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

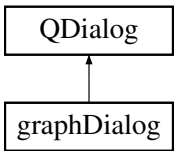
- graphview.h

6.4 graphDialog Class Reference

Dialog window for displaying real-time pressure and temperature graphs.

`#include <graphdialog.h>`

Inheritance diagram for graphDialog:



Public Slots

- void appendData (double pressure, double temperature)

  Appends pressure and temperature data to the internal buffer.
- void onDataReceived (double pressure, double temperature, const QString &flag)

  Handles data received from the WebSocketClient.

Signals

- void **dialogShown** ()

  Emitted when the dialog is shown.
- void **dialogHidden** ()

  Emitted when the dialog is hidden.
- void **flagsent** ()

  Emitted when a flag is sent.

Public Member Functions

- graphDialog (QWidget *parent=nullptr)
  Constructs a graphDialog window.
- ~**graphDialog** ()
  Destroys the graphDialog instance.
- void setWindowSeconds (qreal seconds)
  Sets the time window (in seconds) to display on the graph.

Protected Member Functions

- void showEvent (QShowEvent *ev) override
  Handles the event when the dialog is shown.
- void hideEvent (QHideEvent *ev) override
  Handles the event when the dialog is hidden.

Private Attributes

- Ui::graphDialog * **ui**
  Pointer to the UI components.
- GraphWidget * **m_graph**
  Widget used to plot the data.
- WebSocketClient * **m_wsClient**
  WebSocket client for real-time data.
- QVector< DataPoint > **m_pressure**
  Buffer for pressure data points.
- QVector< DataPoint > **m_temp**
  Buffer for temperature data points.
- qreal **m_windowSeconds** = 600.0
  Display window size in seconds.

6.4.1 Detailed Description

Dialog window for displaying real-time pressure and temperature graphs.

This class handles data visualization using a GraphWidget, and receives real-time updates via WebSocketClient. It also emits signals on show/hide events and when a specific flag is sent.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 graphDialog()

```
graphDialog::graphDialog (
            QWidget * parent = nullptr)  [explicit]
```

Constructs a graphDialog window.

Parameters

parent	The parent widget.

6.4.3 Member Function Documentation

6.4.3.1 appendData

```
void graphDialog::appendData (
            double pressure,
            double temperature)  [slot]
```

Appends pressure and temperature data to the internal buffer.

Parameters

pressure	The new pressure reading.
temperature	The new temperature reading.

6.4.3.2 hideEvent()

```
void graphDialog::hideEvent (
            QHideEvent * ev)  [override], [protected]
```

Handles the event when the dialog is hidden.

Parameters

ev	Pointer to the QHideEvent.

6.4.3.3 onDataReceived

```
void graphDialog::onDataReceived (
            double pressure,
            double temperature,
            const QString & flag)  [slot]
```

Handles data received from the WebSocketClient.

Parameters

pressure	The received pressure value.
temperature	The received temperature value.
flag	A string flag received with the data.

6.4.3.4 setWindowSeconds()

```
void graphDialog::setWindowSeconds (
            qreal seconds)  [inline]
```

Sets the time window (in seconds) to display on the graph.

Parameters

seconds	Length of the time window.

6.4.3.5 showEvent()

```
void graphDialog::showEvent (
            QShowEvent * ev)  [override], [protected]
```

Handles the event when the dialog is shown.

Parameters

ev	Pointer to the QShowEvent.

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

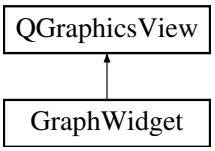
- graphdialog.h
- graphdialog.cpp

6.5 GraphWidget Class Reference

A custom graphics view for plotting pressure and temperature data in real-time.

```
#include <graphview.h>
```

Inheritance diagram for GraphWidget:



Public Member Functions

- GraphWidget (QWidget *parent=nullptr)

  Constructs a GraphWidget.
- ~**GraphWidget** ()

  Destructor.
- void **start** ()

  Starts the periodic data refresh timer.
- void setWindowSeconds (qreal seconds)

  Sets the size of the time window (in seconds) for graph display.
- void appendPressurePoint (const DataPoint &p)

  Appends a new pressure data point.
- void appendTempPoint (const DataPoint &t)

Appends a new temperature data point.

- void **refresh** ()

  Redraws the entire graph: clears scene, draws axes and data series.
- void **clearData** ()

  Clears all stored data points and the graph display.
- void **drawAxes** ()

  Draws the axes on the graph.
- void drawSeries (const QVector< DataPoint > &series, const QColor &penColor, qreal yOffset)

  Draws a time series on the graph with a given color and Y offset.

Protected Member Functions

- void resizeEvent (QResizeEvent *event) override

  Handles resizing of the widget to adjust the graph accordingly.

Private Slots

- void **fetchAndRedraw** ()

  Called periodically to fetch new data and update the graph.

Private Attributes

- QGraphicsScene * **m_scene**

  Scene for rendering the graph.
- QTimer * **m_timer**

  Timer for triggering periodic updates.
- QVector< DataPoint > **m_pressure**

  Stored pressure data points.
- QVector< DataPoint > **m_temp**

  Stored temperature data points.
- qreal **m_windowSeconds** = 600.0

  Time window for visible data (default 10 minutes).

6.5.1 Detailed Description

A custom graphics view for plotting pressure and temperature data in real-time.

This widget displays two time-series graphs (pressure and temperature) and periodically updates to reflect new incoming data.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 GraphWidget()

```
GraphWidget::GraphWidget (
            QWidget * parent = nullptr)  [explicit]
```

Constructs a GraphWidget.

Parameters

parent	The parent widget.

6.5.3 Member Function Documentation

6.5.3.1 appendPressurePoint()

```
void GraphWidget::appendPressurePoint (
            const DataPoint & p)
```

Appends a new pressure data point.

Parameters

p	The pressure data point to append.

6.5.3.2 appendTempPoint()

```
void GraphWidget::appendTempPoint (
            const DataPoint & t)
```

Appends a new temperature data point.

Parameters

t	The temperature data point to append.

6.5.3.3 drawSeries()

```
void GraphWidget::drawSeries (
            const QVector< DataPoint > & series,
            const QColor & penColor,
            qreal yOffset)
```

Draws a time series on the graph with a given color and Y offset.

Parameters

series	The data series to draw.
penColor	The color of the graph line.
yOffset	Vertical offset for the series.

6.5.3.4 resizeEvent()

```
void GraphWidget::resizeEvent (
            QResizeEvent * event)  [override], [protected]
```

Handles resizing of the widget to adjust the graph accordingly.

Parameters

event	Resize event.

6.5.3.5 setWindowSeconds()

```
void GraphWidget::setWindowSeconds (
            qreal seconds)  [inline]
```

Sets the size of the time window (in seconds) for graph display.

Parameters

seconds	Length of the display window in seconds.

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

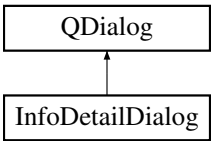
- graphview.h
- graphview.cpp

6.6 InfoDetailDialog Class Reference

A dialog that displays detailed information with navigation options.

```
#include <infodetaildialog.h>
```

Inheritance diagram for InfoDetailDialog:



Public Member Functions

- InfoDetailDialog (QWidget *parent=nullptr)
  Constructs the InfoDetailDialog.
- ~**InfoDetailDialog** ()
  Destructor.

Private Slots

- void **on_btnBack_clicked** ()
  Slot triggered when the "Back" button is clicked.
- void **on_btnHome_clicked** ()
  Slot triggered when the "Home" button is clicked.

Private Attributes

- Ui::InfoDetailDialog * **ui**

  Pointer to the UI components.

6.6.1 Detailed Description

A dialog that displays detailed information with navigation options.

This dialog provides a UI for displaying extended information and includes navigation buttons such as "Back" and "Home".

6.6.2 Constructor & Destructor Documentation

6.6.2.1 InfoDetailDialog()

```
InfoDetailDialog::InfoDetailDialog (
            QWidget * parent = nullptr)  [explicit]
```

Constructs the InfoDetailDialog.

Parameters

parent	The parent widget (optional).

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

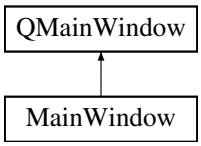
- infodetaildialog.h
- infodetaildialog.cpp

6.7 MainWindow Class Reference

The main application window for the Kaffeknekt dashboard.

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

Inheritance diagram for MainWindow:



Public Member Functions

- MainWindow (QWidget *parent=nullptr)

  Constructs the main window.
- ~**MainWindow** ()

  Destructor.
- void **hideInfoFrame** ()

  Hides the informational side frame (if visible).
- void **on_flagsent** ()

  Slot to handle when a flag is sent (e.g., for a warning or state change).

Protected Member Functions

- void showEvent (QShowEvent *event) override

  Handles the event when the main window is shown.

Private Slots

- void **on_btnHamburger_clicked** ()

  Slot triggered when the hamburger menu button is clicked.
- void **on_btnHome_clicked** ()

  Slot triggered when the home button is clicked.
- void **on_btnSettings_clicked** ()

  Slot triggered when the settings button is clicked.
- void **on_btnInfo_clicked** ()

  Slot triggered when the info button is clicked.
- void **on_btnInstructions_clicked** ()

  Slot triggered when the instructions button is clicked.
- void **on_btnSensorAnalytics_clicked** ()

  Slot triggered when the sensor analytics button is clicked.
- void **on_btnStatistics_clicked** ()

  Slot triggered when the statistics button is clicked.
- void **on_btnHere_clicked** ()

  Slot triggered when the "Here" button inside info frame is clicked.
- void **on_btnTestWarning_clicked** ()

  Slot triggered when the test warning button is clicked. Used for testing the warning popup.

Private Attributes

- Ui::Kaffeknekt * **ui**

  Pointer to UI components.
- GraphWidget * **m_graph**

  Graph widget for data display.
- Statistics * **m_statsDialog**

  Dialog for showing statistics.
- SensorAnalyticsDialog * **m_saDialog**

  Dialog for real-time analytics.
- WebSocketClient * **m_ws**

  WebSocket client for real-time data.

- bool **sideMenuVisible**

  Tracks visibility of side menu.
- int **m_cupCount** = 0

  Tracks the number of cups (usage metric).
- bool **m_warningShown** = false

  Whether a warning dialog has been shown.
- graphDialog * **m_graphDialog**

  Dialog for graph display.

6.7.1 Detailed Description

The main application window for the Kaffeknekt dashboard.

This class controls the main GUI, manages views such as graphs, settings, and instructions, and connects to real-time data via WebSocket. It also handles various UI interactions.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 MainWindow()

```
MainWindow::MainWindow (
            QWidget * parent = nullptr)
```

Constructs the main window.

Parameters

parent	Optional parent widget.

6.7.3 Member Function Documentation

6.7.3.1 showEvent()

```
void MainWindow::showEvent (
            QShowEvent * event)  [override], [protected]
```

Handles the event when the main window is shown.

Parameters

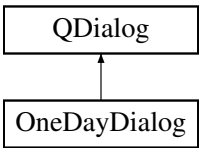
event	Pointer to the QShowEvent.

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

- mainwindow.h
- mainwindow.cpp
- warningdialog.cpp

6.8 OneDayDialog Class Reference

Inheritance diagram for OneDayDialog:



Public Member Functions

- **OneDayDialog** (QWidget ∗parent=nullptr)

Private Attributes

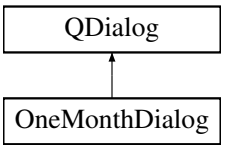
- Ui::OneDayDialog ∗ **ui**

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

- onedaydialog.h
- onedaydialog.cpp

6.9 OneMonthDialog Class Reference

Inheritance diagram for OneMonthDialog:



Public Member Functions

- **OneMonthDialog** (QWidget ∗parent=nullptr)

Private Attributes

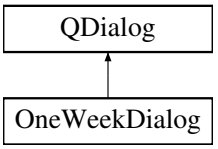
- Ui::OneMonthDialog ∗ **ui**

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

- onemonthdialog.h
- onemonthdialog.cpp

6.10 OneWeekDialog Class Reference

Inheritance diagram for OneWeekDialog:



Public Member Functions

- **OneWeekDialog** (QWidget ∗parent=nullptr)

Private Attributes

- Ui::OneWeekDialog ∗ **ui**

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

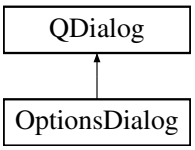
- oneweekdialog.h
- oneweekdialog.cpp

6.11 OptionsDialog Class Reference

Dialog window for displaying and modifying application options.

`#include <optionsdialog.h>`

Inheritance diagram for OptionsDialog:



Public Member Functions

- OptionsDialog (QWidget ∗parent=nullptr)

  Constructs the OptionsDialog.
- ∼**OptionsDialog** ()

  Destructor.

Private Attributes

- Ui::OptionsDialog ∗ **ui**

  Pointer to the UI components.

6.11.1 Detailed Description

Dialog window for displaying and modifying application options.

This class provides a simple Qt dialog interface for setting or displaying configuration options within the application.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 OptionsDialog()

```
OptionsDialog::OptionsDialog (
            QWidget * parent = nullptr)  [explicit]
```

Constructs the OptionsDialog.

Parameters

parent	The parent widget (optional).

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

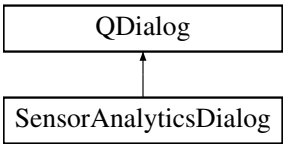
- optionsdialog.h
- optionsdialog.cpp

6.12 SensorAnalyticsDialog Class Reference

Dialog for displaying and selecting sensor analytics metrics.

```
#include <sensoranalyticsdialog.h>
```

Inheritance diagram for SensorAnalyticsDialog:



Signals

- void intervalSelected (qreal seconds)

  Emitted when the user selects a time interval.

Public Member Functions

- SensorAnalyticsDialog (QWidget *parent=nullptr)

  Constructs the SensorAnalyticsDialog.
- ~**SensorAnalyticsDialog** ()

  Destructor.

Private Slots

- void **on_btnWaterTemp_clicked** ()

  Slot triggered when the water temperature button is clicked.
- void **on_btnWaterPressure_clicked** ()

  Slot triggered when the water pressure button is clicked.
- void **on_btnPowerConsumption_clicked** ()

  Slot triggered when the power consumption button is clicked.
- void **on_day_clicked** ()

  Slot triggered when the "day" interval is selected.
- void **on_week_clicked** ()

  Slot triggered when the "week" interval is selected.
- void **on_month_clicked** ()

  Slot triggered when the "month" interval is selected.

Private Attributes

- Ui::SensorAnalyticsDialog * **ui**

  Pointer to the UI components.

6.12.1 Detailed Description

Dialog for displaying and selecting sensor analytics metrics.

This dialog allows the user to view different sensor-related data (e.g., water temperature, pressure, power consumption) and to select the time interval for data analysis.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 SensorAnalyticsDialog()

```
SensorAnalyticsDialog::SensorAnalyticsDialog (
            QWidget * parent = nullptr)  [explicit]
```

Constructs the SensorAnalyticsDialog.

Parameters

parent	The parent widget (optional).

6.12.3 Member Function Documentation

6.12.3.1 intervalSelected

```
void SensorAnalyticsDialog::intervalSelected (
            qreal seconds)  [signal]
```

Emitted when the user selects a time interval.

Parameters

seconds	The selected interval in seconds.

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

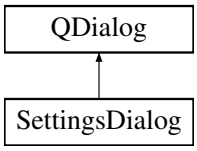
- sensoranalyticsdialog.h
- sensoranalyticsdialog.cpp

6.13 SettingsDialog Class Reference

Dialog window for configuring application settings.

`#include <settingsdialog.h>`

Inheritance diagram for SettingsDialog:



Public Member Functions

- SettingsDialog (QWidget *parent=nullptr)
  Constructs the SettingsDialog.
- ~**SettingsDialog** ()
  Destructor.

Private Attributes

- Ui::SettingsDialog * **ui**
  Pointer to the UI components.

6.13.1 Detailed Description

Dialog window for configuring application settings.

This class provides a Qt-based dialog interface where users can view and modify application-specific settings.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 SettingsDialog()

```
SettingsDialog::SettingsDialog (
            QWidget * parent = nullptr)  [explicit]
```

Constructs the SettingsDialog.

Parameters

parent	The parent widget (optional).

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

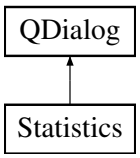
- settingsdialog.h
- settingsdialog.cpp

6.14 Statistics Class Reference

Dialog for displaying usage statistics.

`#include <statistics.h>`

Inheritance diagram for Statistics:



Public Member Functions

- Statistics (QWidget ∗parent=nullptr)
  Constructs the Statistics dialog.
- ∼**Statistics** ()
  Destructor.
- void setCupCount (int num)
  Sets the number of cups to display in the statistics.

Private Attributes

- Ui::Statistics ∗ **ui**
  Pointer to the UI components.
- int **m_cupCount** = 0
  Number of cups tracked for statistics.

6.14.1 Detailed Description

Dialog for displaying usage statistics.

This class provides a Qt dialog that displays statistical information, such as the number of cups made by the espresso machine.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 Statistics()

```
Statistics::Statistics (
            QWidget * parent = nullptr)  [explicit]
```

Constructs the Statistics dialog.

Parameters

parent	The parent widget (optional).

6.14.3 Member Function Documentation

6.14.3.1 setCupCount()

```
void Statistics::setCupCount (
            int num)
```

Sets the number of cups to display in the statistics.

Parameters

num	The number of cups.

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

- statistics.h
- statistics.cpp

6.15 WarningDialog Class Reference

Dialog for displaying a temperature-related warning to the user.

```
#include <warningdialog.h>
```

6.15.1 Detailed Description

Dialog for displaying a temperature-related warning to the user.

This dialog notifies the user when a critical temperature threshold is reached. It provides options to ignore or acknowledge the warning.

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

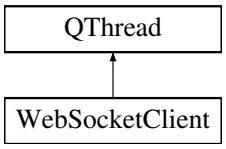
- warningdialog.h

6.16 WebSocketClient Class Reference

Threaded client for receiving real-time data via WebSocket.

`#include <websocketclient.h>`

Inheritance diagram for WebSocketClient:



Signals

- void dataReceived (double pressure, double temperature, const QString &flag)

  Emitted when new sensor data is received.

Public Member Functions

- WebSocketClient (QObject ∗parent=nullptr)

  Constructs the WebSocketClient.
- void **run** () override

  Runs the thread logic. Override this method to implement the WebSocket communication.

6.16.1 Detailed Description

Threaded client for receiving real-time data via WebSocket.

This class runs in a separate thread and is responsible for connecting to a WebSocket server to receive live pressure, temperature, and flag data.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 WebSocketClient()

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

Constructs the WebSocketClient.

Parameters

parent	Optional parent QObject.

6.16.3 Member Function Documentation

6.16.3.1 dataReceived

```
void WebSocketClient::dataReceived (
            double pressure,
            double temperature,
            const QString & flag)  [signal]
```

Emitted when new sensor data is received.

Parameters

pressure	The pressure value.
temperature	The temperature value.
flag	A string-based flag or status indicator.

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

- websocketclient.h
- websocketclient.cpp

Chapter 7

File Documentation

7.1 coffeeinstructionsdialog.h

```
00001 #ifndef COFFEEINSTRUCTIONSDIALOG_H
00002 #define COFFEEINSTRUCTIONSDIALOG_H
00003
00004 #include <QDialog>
00005
00010 namespace Ui {
00011 class CoffeeInstructionsDialog;
00012 }
00013
00021 class CoffeeInstructionsDialog : public QDialog
00022 {
00023     Q_OBJECT
00024
00025 public:
00032     explicit CoffeeInstructionsDialog(QWidget *parent = nullptr);
00033
00039     ~CoffeeInstructionsDialog();
00040
00041 private:
00047     Ui::CoffeeInstructionsDialog *ui;
00048 };
00049
00050 #endif // COFFEEINSTRUCTIONSDIALOG_H
```

7.2 datafetcher.h

```
00001 #ifndef DATAFETCHER_H
00002 #define DATAFETCHER_H
00003 #include <QDebug>
00004 #include <QObject>
00005 #include <QVector>
00006 #include "graphview.h"    // for DataPoint
00007
00015 class DataFetcher
00016 {
00017 public:
00021     DataFetcher();
00022
00032     static QVector<DataPoint> fetchPressureWindow(qreal windowSeconds,
00033                                                   const QString& influxUrl,
00034                                                   const QString& token,
00035                                                   const QString& bucket);
00036
00046     static QVector<DataPoint> fetchTempWindow(qreal windowSeconds,
00047                                               const QString& influxUrl,
00048                                               const QString& token,
00049                                               const QString& bucket);
00050 };
00051
00052 #endif // DATAFETCHER_H
```

7.3 graphdialog.h

```
00001 //graphdialog.h
00002 #ifndef GRAPHDIALOG_H
00003 #define GRAPHDIALOG_H
00004
00005 #include <QDialog>
00006 #include <QSizePolicy>
00007
00008 #include "graphview.h"
00009 #include "websocketclient.h"
00010
00015 namespace Ui { class graphDialog; }
00016
00025 class graphDialog : public QDialog {
00026     Q_OBJECT
00027
00028 public:
00033     explicit graphDialog(QWidget *parent = nullptr);
00034
00038     ~graphDialog();
00039
00044     void setWindowSeconds(qreal seconds) { m_windowSeconds = seconds; }
00045
00046 signals:
00050     void dialogShown();
00051
00055     void dialogHidden();
00056
00060     void flagsent();
00061
00062 public slots:
00068     void appendData(double pressure, double temperature);
00069
00076     void onDataReceived(double pressure,
00077                         double temperature,
00078                         const QString& flag);
00079
00080 private:
00081     Ui::graphDialog *ui;
00082     GraphWidget     *m_graph;
00083     WebSocketClient *m_wsClient;
00084
00085     QVector<DataPoint> m_pressure;
00086     QVector<DataPoint> m_temp;
00087     qreal              m_windowSeconds = 600.0;
00088
00089 protected:
00094     void showEvent(QShowEvent* ev) override;
00095
00100     void hideEvent(QHideEvent* ev) override;
00101 };
00102
00103 #endif // GRAPHDIALOG_H
```

7.4 graphview.h

```
00001 #ifndef GRAPHWIDGET_H
00002 #define GRAPHWIDGET_H
00003
00004 #include <QGraphicsView>
00005 #include <QGraphicsScene>
00006 #include <QTimer>
00007
00008 #include <QResizeEvent>
00009 #include <QGraphicsSimpleTextItem>
00010
00015 struct DataPoint {
00016     qreal timestamp;
00017     qreal value;
00018 };
00019
00027 class GraphWidget : public QGraphicsView {
00028     Q_OBJECT
00029 public:
00034     explicit GraphWidget(QWidget* parent = nullptr);
00035
00039     ~GraphWidget();
00040
00044     void start();
00045
00050     void setWindowSeconds(qreal seconds) { m_windowSeconds = seconds; }
```

```
00051
00056     void appendPressurePoint(const DataPoint& p);
00057
00062     void appendTempPoint(const DataPoint& t);
00063
00067     void refresh();
00068
00072     void clearData();
00073
00077     void drawAxes();
00078
00085     void drawSeries(const QVector<DataPoint>& series,
00086                     const QColor& penColor,
00087                     qreal yOffset);
00088
00089 private slots:
00093     void fetchAndRedraw();
00094
00095 protected:
00100     void resizeEvent(QResizeEvent* event) override;
00101
00102 private:
00103     QGraphicsScene* m_scene;
00104     QTimer*         m_timer;
00105
00106     QVector<DataPoint> m_pressure;
00107     QVector<DataPoint> m_temp;
00108
00109     qreal m_windowSeconds = 600.0;
00110 };
00111
00112 #endif // GRAPHWIDGET_H
```

7.5 infodetaildialog.h

```
00001 #ifndef INFODETAILDIALOG_H
00002 #define INFODETAILDIALOG_H
00003
00004 #include <QDialog>
00005
00010 namespace Ui {
00011 class InfoDetailDialog;
00012 }
00013
00021 class InfoDetailDialog : public QDialog
00022 {
00023     Q_OBJECT
00024
00025 public:
00030     explicit InfoDetailDialog(QWidget *parent = nullptr);
00031
00035     ~InfoDetailDialog();
00036
00037 private slots:
00041     void on_btnBack_clicked();
00042
00046     void on_btnHome_clicked();
00047
00048 private:
00049     Ui::InfoDetailDialog *ui;
00050 };
00051
00052 #endif // INFODETAILDIALOG_H
```

7.6 mainwindow.h

```
00001 #ifndef MAINWINDOW_H
00002 #define MAINWINDOW_H
00003
00004 #include "sensoranalyticsdialog.h"
00005 #include "coffeeinstructionsdialog.h"
00006 #include "graphview.h"
00007 #include "warningdialog.h"
00008 #include "websocketclient.h"
00009 #include "graphdialog.h"
00010 #include <QMainWindow>
00011 #include <QTimer>
00012 #include <QGraphicsScene>
```

```
00013 #include <QGraphicsView>
00014 #include "statistics.h"
00015 #include "ui_mainwindow.h"
00016 #include "infodetaildialog.h"
00017 #include "datafetcher.h"
00018 #include <QFrame>
00019 #include <QPushButton>
00020
00021 QT_BEGIN_NAMESPACE
00026 namespace Ui {
00027 class Kaffeknekt;
00028 }
00029 QT_END_NAMESPACE
00030
00038 class MainWindow : public QMainWindow
00039 {
00040     Q_OBJECT
00041
00042 public:
00047     MainWindow(QWidget *parent = nullptr);
00048
00052     ~MainWindow();
00053
00057     void hideInfoFrame();
00058
00062     void on_flagsent();
00063
00064 private slots:
00065     // Frame 1 (main menu)
00069     void on_btnHamburger_clicked();
00070
00074     void on_btnHome_clicked();
00075
00079     void on_btnSettings_clicked();
00080
00084     void on_btnInfo_clicked();
00085
00086     // Frame 2 (submenu)
00090     void on_btnInstructions_clicked();
00091
00095     void on_btnSensorAnalytics_clicked();
00096
00100     void on_btnStatistics_clicked();
00101
00105     void on_btnHere_clicked();
00106
00111     void on_btnTestWarning_clicked();
00112
00113 protected:
00118     void showEvent(QShowEvent *event) override;
00119
00120 private:
00121     Ui::Kaffeknekt *ui;
00122     GraphWidget    *m_graph;
00123     Statistics          *m_statsDialog;
00124     SensorAnalyticsDialog *m_saDialog;
00125     WebSocketClient       *m_ws;
00126     bool sideMenuVisible;
00127     int    m_cupCount        = 0;
00128     bool   m_warningShown     = false;
00129     graphDialog     *m_graphDialog;
00130 };
00131
00132 #endif // MAINWINDOW_H
```

7.7 moc_predefs.h

```
00001 #define __DBL_MIN_EXP__ (-1021)
00002 #define __LDBL_MANT_DIG__ 113
00003 #define __cpp_attributes 200809L
00004 #define __cpp_nontype_template_parameter_auto 201606L
00005 #define __UINT_LEAST16_MAX__ 0xffff
00006 #define __ARM_SIZEOF_WCHAR_T 4
00007 #define __ATOMIC_ACQUIRE 2
00008 #define __FLT128_MAX_10_EXP__ 4932
00009 #define __FLT_MIN__ 1.17549435082228750796873653722224568e-38F
00010 #define __GCC_IEC_559_COMPLEX 2
00011 #define __cpp_aggregate_nsdmi 201304L
00012 #define __UINT_LEAST8_TYPE__ unsigned char
00013 #define __INTMAX_C(c) c ## L
00014 #define __CHAR_BIT__ 8
00015 #define __UINT8_MAX__ 0xff
00016 #define __USER_LABEL_PREFIX__
```

```
00017 #define __WINT_MAX__ 0xffffffffU
00018 #define __cpp_static_assert 201411L
00019 #define __WCHAR_MAX__ 0xffffffffU
00020 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 1
00021 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 1
00022 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 1
00023 #define __GCC_ATOMIC_CHAR_LOCK_FREE 2
00024 #define __GCC_IEC_559 2
00025 #define __FLT32X_DECIMAL_DIG__ 17
00026 #define __FLT_EVAL_METHOD__ 0
00027 #define __cpp_binary_literals 201304L
00028 #define __FLT64_DECIMAL_DIG__ 17
00029 #define __cpp_noexcept_function_type 201510L
00030 #define __GCC_ATOMIC_CHAR32_T_LOCK_FREE 2
00031 #define __cpp_variadic_templates 200704L
00032 #define __UINT_FAST32_TYPE__ long unsigned int
00033 #define __UINT_FAST64_MAX__ 0xffffffffffffffffUL
00034 #define __SIG_ATOMIC_TYPE__ int
00035 #define __DBL_MIN_10_EXP__ (-307)
00036 #define __FINITE_MATH_ONLY__ 0
00037 #define __cpp_variable_templates 201304L
00038 #define __FLT32X_MAX_EXP__ 1024
00039 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 1
00040 #define __GNUC_PATCHLEVEL__ 0
00041 #define __FLT32_HAS_DENORM__ 1
00042 #define __UINT_FAST8_MAX__ 0xff
00043 #define __cpp_rvalue_reference 200610L
00044 #define __cpp_nested_namespace_definitions 201411L
00045 #define __INT8_C(c) c
00046 #define __INT_LEAST8_WIDTH__ 8
00047 #define __cpp_variadic_using 201611L
00048 #define __UINT_LEAST64_MAX__ 0xffffffffffffffffUL
00049 #define __INT_LEAST8_MAX__ 0x7f
00050 #define __cpp_capture_star_this 201603L
00051 #define __SHRT_MAX__ 0x7fff
00052 #define __STDC_ISO_10646__ 201706L
00053 #define __LDBL_MAX__ 1.189731495357231765085759326628007 02e+4932L
00054 #define __ARM_FEATURE_IDIV 1
00055 #define __FLT64X_MAX_10_EXP__ 4932
00056 #define __cpp_if_constexpr 201606L
00057 #define __FLT64_NORM_MAX__ 1.797693134862315708145274237317043 57e+308F64
00058 #define __LDBL_IS_IEC_60559__ 2
00059 #define __ARM_FP 14
00060 #define __FLT64X_IS_IEC_60559__ 2
00061 #define __FLT64X_HAS_QUIET_NAN__ 1
00062 #define __WINT_TYPE__ unsigned int
00063 #define __UINT_LEAST8_MAX__ 0xff
00064 #define __FLT128_DENORM_MIN__ 6.475175119438025110924438958227646 55e-4966F128
00065 #define __UINTMAX_TYPE__ long unsigned int
00066 #define __cpp_nsdmi 200809L
00067 #define __linux 1
00068 #define __CHAR_UNSIGNED__ 1
00069 #define __UINT32_MAX__ 0xffffffffU
00070 #define __GXX_EXPERIMENTAL_CXX0X__ 1
00071 #define __DBL_DENORM_MIN__ double(4.940656458412465441765687928682213 72e-324L)
00072 #define __AARCH64_CMODEL_SMALL__ 1
00073 #define __LDBL_MAX_EXP__ 16384
00074 #define __INT_FAST32_WIDTH__ 64
00075 #define __FLT128_MIN_EXP__ (-16381)
00076 #define __FLT128_MIN_10_EXP__ (-4931)
00077 #define __FLT32X_IS_IEC_60559__ 2
00078 #define __INT_LEAST16_WIDTH__ 16
00079 #define __SCHAR_MAX__ 0x7f
00080 #define __FLT128_MANT_DIG__ 113
00081 #define __DBL_MAX__ double(1.797693134862315708145274237317043 57e+308L)
00082 #define __FLT32X_DIG__ 15
00083 #define __WCHAR_MIN__ 0U
00084 #define __INT64_C(c) c ## L
00085 #define __GCC_ATOMIC_POINTER_LOCK_FREE 2
00086 #define __FLT_MAX__ 3.402823466385288598117041834845169 25e+38F
00087 #define __SIZEOF_INT__ 4
00088 #define __FLT32X_MANT_DIG__ 53
00089 #define __GCC_ATOMIC_CHAR16_T_LOCK_FREE 2
00090 #define __cpp_aligned_new 201606L
00091 #define __FLT32_MAX_10_EXP__ 38
00092 #define __FLT64X_EPSILON__ 1.925929944387235853055977942584927 32e-34F64x
00093 #define __STDC_HOSTED__ 1
00094 #define __cpp_decltype_auto 201304L
00095 #define __DBL_DIG__ 15
00096 #define __FLT32_DIG__ 6
00097 #define __FLT_EPSILON__ 1.192092895507812500000000000000000 00e-7F
00098 #define __GXX_WEAK__ 1
00099 #define __SHRT_WIDTH__ 16
00100 #define __FLT32_IS_IEC_60559__ 2
00101 #define __LDBL_MIN__ 3.362103143112093506262677817321752 60e-4932L
00102 #define __DBL_IS_IEC_60559__ 2
00103 #define __FLT16_HAS_QUIET_NAN__ 1
```

```
00104 #define __cpp_threadsafe_static_init 200806L
00105 #define __ARM_SIZEOF_MINIMAL_ENUM 4
00106 #define __cpp_enumerator_attributes 201411L
00107 #define __FLT64X_DENORM_MIN__ 6.47517511943802511092443895822764655e-4966F64x
00108 #define __FP_FAST_FMA 1
00109 #define __FLT32X_HAS_INFINITY__ 1
00110 #define __INT32_MAX__ 0x7fffffff
00111 #define __FLT16_DIG__ 3
00112 #define __INT_WIDTH__ 32
00113 #define __SIZEOF_LONG__ 8
00114 #define __STDC_IEC_559__ 1
00115 #define __UINT16_C(c) c
00116 #define __DECIMAL_DIG__ 36
00117 #define __FLT64_EPSILON__ 2.22044604925031308084726333618164062e-16F64
00118 #define __gnu_linux__ 1
00119 #define __INT16_MAX__ 0x7fff
00120 #define __FLT64_MIN_EXP__ (-1021)
00121 #define __FLT64X_MIN_10_EXP__ (-4931)
00122 #define __LDBL_HAS_QUIET_NAN__ 1
00123 #define __FLT16_MIN_EXP__ (-13)
00124 #define __FLT64_MANT_DIG__ 53
00125 #define __FLT64X_MANT_DIG__ 113
00126 #define __GNUC__ 12
00127 #define __pie__ 2
00128 #define __GXX_RTTI 1
00129 #define __FLT16_DECIMAL_DIG__ 5
00130 #define __FLT_HAS_DENORM__ 1
00131 #define __SIZEOF_LONG_DOUBLE__ 16
00132 #define __cpp_rtti 199711L
00133 #define __STDC_UTF_16__ 1
00134 #define __FLT64_MAX_10_EXP__ 308
00135 #define __FLT16_MAX_10_EXP__ 4
00136 #define __FLT32_HAS_INFINITY__ 1
00137 #define __cpp_raw_strings 200710L
00138 #define __INT_FAST32_MAX__ 0x7fffffffffffffffL
00139 #define __DBL_HAS_INFINITY__ 1
00140 #define __INT64_MAX__ 0x7fffffffffffffffL
00141 #define __HAVE_SPECULATION_SAFE_VALUE 1
00142 #define __cpp_fold_expressions 201603L
00143 #define __INTPTR_WIDTH__ 64
00144 #define __FLT64X_HAS_INFINITY__ 1
00145 #define __cpp_delegating_constructors 200604L
00146 #define __FLT32X_HAS_DENORM__ 1
00147 #define __INT_FAST16_TYPE__ long int
00148 #define __cpp_template_auto 201606L
00149 #define __LDBL_HAS_DENORM__ 1
00150 #define __cplusplus 201703L
00151 #define __cpp_ref_qualifiers 200710L
00152 #define __DEPRECATED 1
00153 #define __cpp_rvalue_references 200610L
00154 #define __DBL_MAX_EXP__ 1024
00155 #define __WCHAR_WIDTH__ 32
00156 #define __FLT64_MAX__ 1.79769313486231570814527423731704357e+308F64
00157 #define __FLT32_MAX__ 3.40282346638528859811704183484516925e+38F32
00158 #define __GCC_ATOMIC_LONG_LOCK_FREE 2
00159 #define __FLT16_MANT_DIG__ 11
00160 #define __FLT32_HAS_QUIET_NAN__ 1
00161 #define __LONG_LONG_MAX__ 0x7fffffffffffffffLL
00162 #define __SIZEOF_SIZE_T__ 8
00163 #define __ARM_ALIGN_MAX_PWR 28
00164 #define __SIZEOF_WINT_T__ 4
00165 #define __LONG_LONG_WIDTH__ 64
00166 #define __cpp_initializer_lists 200806L
00167 #define __WCHAR_UNSIGNED__ 1
00168 #define __FLT32_MAX_EXP__ 128
00169 #define __cpp_hex_float 201603L
00170 #define __ARM_FP16_FORMAT_IEEE 1
00171 #define __FP_FAST_FMAF32x 1
00172 #define __FLT128_HAS_INFINITY__ 1
00173 #define __FLT_MIN_EXP__ (-125)
00174 #define __PIE__ 2
00175 #define __GCC_HAVE_DWARF2_CFI_ASM 1
00176 #define __cpp_lambdas 200907L
00177 #define __FLT32X_MIN_EXP__ (-1021)
00178 #define __INT_FAST64_TYPE__ long int
00179 #define __ARM_FP16_ARGS 1
00180 #define __DBL_DECIMAL_DIG__ 17
00181 #define __FP_FAST_FMAF 1
00182 #define __FLT128_NORM_MAX__ 1.18973149535723176508575932662800702e+4932F128
00183 #define __FLT64_DENORM_MIN__ 4.94065645841246544176568792868221372e-324F64
00184 #define __DBL_MIN__ double(2.22507385850720138309023271733240406e-308L)
00185 #define __ARM_FEATURE_CLZ 1
00186 #define __FLT16_DENORM_MIN__ 5.96046447753906250000000000000000000e-8F16
00187 #define __unix__ 1
00188 #define __FLT64X_NORM_MAX__ 1.18973149535723176508575932662800702e+4932F64x
00189 #define __SIZEOF_POINTER__ 8
00190 #define __LP64__ 1
```

```
00191 #define __DBL_HAS_QUIET_NAN__ 1
00192 #define __FLT_EVAL_METHOD_C99__ 0
00193 #define __FLT32X_EPSILON__ 2.220446049250313080847263336181640 62e-16F32x
00194 #define __LDBL_DECIMAL_DIG__ 36
00195 #define __aarch64__ 1
00196 #define __FLT64_MIN_10_EXP__ (-307)
00197 #define __INT_FAST64_WIDTH__ 64
00198 #define __FLT64X_DECIMAL_DIG__ 36
00199 #define __REGISTER_PREFIX__
00200 #define __UINT16_MAX__ 0xffff
00201 #define __INTMAX_WIDTH__ 64
00202 #define __GXX_ABI_VERSION 1017
00203 #define __AARCH64EL__ 1
00204 #define __LDBL_HAS_INFINITY__ 1
00205 #define __UINT8_TYPE__ unsigned char
00206 #define __FLT_DIG__ 6
00207 #define __NO_INLINE__ 1
00208 #define __DEC_EVAL_METHOD__ 2
00209 #define __FLT_MANT_DIG__ 24
00210 #define __FLT16_MIN_10_EXP__ (-4)
00211 #define __VERSION__ "12.2.0"
00212 #define __UINT64_C(c) c ## UL
00213 #define __cpp_unicode_characters 201411L
00214 #define _STDC_PREDEF_H 1
00215 #define __INT_LEAST32_MAX__ 0x7fffffff
00216 #define __GCC_ATOMIC_INT_LOCK_FREE 2
00217 #define __FLT128_MAX_EXP__ 16384
00218 #define __FLT32_MANT_DIG__ 24
00219 #define __FLOAT_WORD_ORDER__ __ORDER_LITTLE_ENDIAN__
00220 #define __FLT16_MAX_EXP__ 16
00221 #define __BIGGEST_ALIGNMENT__ 16
00222 #define __STDC_IEC_60559_COMPLEX__ 201404L
00223 #define __INT32_C(c) c
00224 #define __cpp_aggregate_bases 201603L
00225 #define __FLT128_HAS_DENORM__ 1
00226 #define __FLT128_DIG__ 33
00227 #define __SCHAR_WIDTH__ 8
00228 #define __ORDER_PDP_ENDIAN__ 3412
00229 #define __ARM_64BIT_STATE 1
00230 #define __INT_FAST32_TYPE__ long int
00231 #define __FLT128_MIN__ 3.362103143112093506262677817321752 60e-4932F128
00232 #define __UINT_LEAST16_TYPE__ short unsigned int
00233 #define __SIZE_TYPE__ long unsigned int
00234 #define __UINT64_MAX__ 0xffffffffffffffffUL
00235 #define __FLT_IS_IEC_60559__ 2
00236 #define __GNUC_WIDE_EXECUTION_CHARSET_NAME "UTF-32LE"
00237 #define __FLT64X_DIG__ 33
00238 #define __ARM_FEATURE_FMA 1
00239 #define __INT8_TYPE__ signed char
00240 #define __GNUG__ 12
00241 #define __cpp_digit_separators 201309L
00242 #define __ELF__ 1
00243 #define __GCC_ASM_FLAG_OUTPUTS__ 1
00244 #define __GCC_ATOMIC_TEST_AND_SET_TRUEVAL 1
00245 #define __FLT_RADIX__ 2
00246 #define __INT_LEAST16_TYPE__ short int
00247 #define __ARM_ARCH_PROFILE 65
00248 #define __LDBL_EPSILON__ 1.925929944387235853055977942584927 32e-34L
00249 #define __UINTMAX_C(c) c ## UL
00250 #define __GLIBCXX_BITSIZE_INT_N_0 128
00251 #define __ARM_PCS_AAPCS64 1
00252 #define __SIG_ATOMIC_MAX__ 0x7fffffff
00253 #define __INT_LEAST64_WIDTH__ 64
00254 #define __GCC_ATOMIC_WCHAR_T_LOCK_FREE 2
00255 #define __STDC_IEC_60559_BFP__ 201404L
00256 #define __SIZEOF_PTRDIFF_T__ 8
00257 #define __ATOMIC_RELAXED 0
00258 #define __FLT_EVAL_METHOD_TS_18661_3__ 0
00259 #define unix 1
00260 #define __cpp_guaranteed_copy_elision 201606L
00261 #define __LDBL_DIG__ 33
00262 #define __FLT64_IS_IEC_60559__ 2
00263 #define __FLT16_IS_IEC_60559__ 2
00264 #define __INT_FAST16_MAX__ 0x7fffffffffffffffL
00265 #define __GCC_CONSTRUCTIVE_SIZE 64
00266 #define __FLT64_DIG__ 15
00267 #define __UINT_FAST32_MAX__ 0xffffffffffffffffUL
00268 #define __UINT_LEAST64_TYPE__ long unsigned int
00269 #define __FLT16_EPSILON__ 9.765625000000000000000000000000000 0e-4F16
00270 #define __FLT_HAS_QUIET_NAN__ 1
00271 #define __FLT_MAX_10_EXP__ 38
00272 #define __LONG_MAX__ 0x7fffffffffffffffL
00273 #define __FLT64X_HAS_DENORM__ 1
00274 #define __FLT_HAS_INFINITY__ 1
00275 #define __GNUC_EXECUTION_CHARSET_NAME "UTF-8"
00276 #define __unix 1
00277 #define __cpp_unicode_literals 200710L
```

```
00278 #define __DBL_HAS_DENORM__ 1
00279 #define __UINT_FAST16_TYPE__ long unsigned int
00280 #define __FLT32X_HAS_QUIET_NAN__ 1
00281 #define __CHAR16_TYPE__ short unsigned int
00282 #define __PRAGMA_REDEFINE_EXTNAME 1
00283 #define __SIZE_WIDTH__ 64
00284 #define __INT_LEAST16_MAX__ 0x7fff
00285 #define __FLT16_NORM_MAX__ 6.55040000000000000000000000000000000e+4F16
00286 #define __FLT32_DENORM_MIN__ 1.40129846432481707092372958328991613e-45F32
00287 #define __INT_LEAST64_TYPE__ long int
00288 #define __INT16_TYPE__ short int
00289 #define __INT_LEAST8_TYPE__ signed char
00290 #define __FLT16_MAX__ 6.55040000000000000000000000000000000e+4F16
00291 #define __cpp_structured_bindings 201606L
00292 #define __INT_FAST8_MAX__ 0x7f
00293 #define __ARM_ARCH 8
00294 #define __FLT128_MAX__ 1.18973149535723176508575932662800702e+4932F128
00295 #define __INTPTR_MAX__ 0x7fffffffffffffffL
00296 #define __cpp_sized_deallocation 201309L
00297 #define linux 1
00298 #define __FLT64_HAS_QUIET_NAN__ 1
00299 #define __INTMAX_TYPE__ long int
00300 #define __FLT64X_MIN_EXP__ (-16381)
00301 #define __FLT32_MIN_10_EXP__ (-37)
00302 #define __PTRDIFF_WIDTH__ 64
00303 #define __FLT64_HAS_INFINITY__ 1
00304 #define __FLT64X_MAX__ 1.18973149535723176508575932662800702e+4932F64x
00305 #define __FLT16_HAS_INFINITY__ 1
00306 #define __STDCPP_DEFAULT_NEW_ALIGNMENT__ 16
00307 #define __SIG_ATOMIC_MIN__ (-__SIG_ATOMIC_MAX__ - 1)
00308 #define __cpp_nontype_template_args 201411L
00309 #define __PTRDIFF_MAX__ 0x7fffffffffffffffL
00310 #define __cpp_return_type_deduction 201304L
00311 #define __INTPTR_TYPE__ long int
00312 #define __UINT16_TYPE__ short unsigned int
00313 #define __WCHAR_TYPE__ unsigned int
00314 #define __cpp_range_based_for 201603L
00315 #define __pic__ 2
00316 #define __UINTPTR_MAX__ 0xffffffffffffffffUL
00317 #define __ARM_ARCH_8A 1
00318 #define __ARM_FEATURE_UNALIGNED 1
00319 #define __cpp_decltype 200707L
00320 #define __FLT32_DECIMAL_DIG__ 9
00321 #define __INT_FAST64_MAX__ 0x7fffffffffffffffL
00322 #define __FLT_NORM_MAX__ 3.40282346638528859811704183484516925e+38F
00323 #define __FLT64X_MAX_EXP__ 16384
00324 #define __UINT_FAST64_TYPE__ long unsigned int
00325 #define __cpp_inline_variables 201606L
00326 #define __INT_MAX__ 0x7fffffff
00327 #define __STDCPP_THREADS__ 1
00328 #define __INT64_TYPE__ long int
00329 #define __FLT_MAX_EXP__ 128
00330 #define __ORDER_BIG_ENDIAN__ 4321
00331 #define __DBL_MANT_DIG__ 53
00332 #define __cpp_inheriting_constructors 201511L
00333 #define __INT_LEAST64_MAX__ 0x7fffffffffffffffL
00334 #define __FP_FAST_FMAF32 1
00335 #define __UINT_LEAST32_TYPE__ unsigned int
00336 #define __SIZEOF_SHORT__ 2
00337 #define __FLT32_NORM_MAX__ 3.40282346638528859811704183484516925e+38F32
00338 #define __LDBL_MIN_EXP__ (-16381)
00339 #define __GCC_ATOMIC_BOOL_LOCK_FREE 2
00340 #define __SIG_ATOMIC_WIDTH__ 32
00341 #define __WINT_WIDTH__ 32
00342 #define __FP_FAST_FMAF64 1
00343 #define __FLT32X_MAX_10_EXP__ 308
00344 #define __SIZEOF_INT128__ 16
00345 #define __FLT16_MIN__ 6.10351562500000000000000000000000000e-5F16
00346 #define __LDBL_MAX_10_EXP__ 4932
00347 #define __DBL_EPSILON__ double(2.22044604925031308084726333618164062e-16L)
00348 #define __FLT32_MIN_EXP__ (-125)
00349 #define _LP64 1
00350 #define __UINT8_C(c) c
00351 #define __FLT64_MAX_EXP__ 1024
00352 #define __INT_LEAST32_TYPE__ int
00353 #define __SIZEOF_WCHAR_T__ 4
00354 #define __ARM_NEON 1
00355 #define __FLT128_HAS_QUIET_NAN__ 1
00356 #define __INTMAX_MAX__ 0x7fffffffffffffffL
00357 #define __UINT_FAST8_TYPE__ unsigned char
00358 #define __INT_FAST8_TYPE__ signed char
00359 #define __cpp_namespace_attributes 201411L
00360 #define __FLT64X_MIN__ 3.36210314311209350626267781732175260e-4932F64x
00361 #define __GNUC_STDC_INLINE__ 1
00362 #define __FLT64_HAS_DENORM__ 1
00363 #define __FLT32_EPSILON__ 1.19209289550781250000000000000000000e-7F32
00364 #define __FLT16_HAS_DENORM__ 1
```

```
00365 #define __STDC_UTF_32__ 1
00366 #define __INT_FAST8_WIDTH__ 8
00367 #define __FLT32X_MAX__ 1.79769313486231570814527423731704357e+308F32x
00368 #define __cpp_alias_templates 200704L
00369 #define __DBL_NORM_MAX__ double(1.79769313486231570814527423731704357e+308L)
00370 #define __BYTE_ORDER__ __ORDER_LITTLE_ENDIAN__
00371 #define __ARM_ALIGN_MAX_STACK_PWR 16
00372 #define __LDBL_DENORM_MIN__ 6.475175119438025110924438958227646554e-4966L
00373 #define __GCC_DESTRUCTIVE_SIZE 256
00374 #define __cpp_runtime_arrays 198712L
00375 #define __UINT64_TYPE__ long unsigned int
00376 #define __UINT32_C(c) c ## U
00377 #define __FLT32X_MIN__ 2.22507385850720138309023271733240406e-308F32x
00378 #define __WINT_MIN__ 0U
00379 #define __FLT128_IS_IEC_60559__ 2
00380 #define __INT8_MAX__ 0x7f
00381 #define __LONG_WIDTH__ 64
00382 #define __PIC__ 2
00383 #define __FLT32X_NORM_MAX__ 1.79769313486231570814527423731704357e+308F32x
00384 #define __CHAR32_TYPE__ unsigned int
00385 #define __cpp_constexpr 201603L
00386 #define __cpp_deduction_guides 201703L
00387 #define __ARM_FEATURE_NUMERIC_MAXMIN 1
00388 #define __INT32_TYPE__ int
00389 #define __SIZEOF_DOUBLE__ 8
00390 #define __cpp_exceptions 199711L
00391 #define __FLT64_MIN__ 2.22507385850720138309023271733240406e-308F64
00392 #define __FLT_DENORM_MIN__ 1.40129846432481707092372958328991613e-45F
00393 #define __INT_LEAST32_WIDTH__ 32
00394 #define __SIZEOF_FLOAT__ 4
00395 #define __ATOMIC_CONSUME 1
00396 #define __GNUC_MINOR__ 2
00397 #define __GLIBCXX_TYPE_INT_N_0 __int128
00398 #define __INT_FAST16_WIDTH__ 64
00399 #define __UINTMAX_MAX__ 0xffffffffffffffffUL
00400 #define __FLT32X_DENORM_MIN__ 4.94065645841246544176568792868221372e-324F32x
00401 #define __cpp_template_template_args 201611L
00402 #define __DBL_MAX_10_EXP__ 308
00403 #define __INT16_C(c) c
00404 #define __ARM_ARCH_ISA_A64 1
00405 #define __STDC__ 1
00406 #define __PTRDIFF_TYPE__ long int
00407 #define __FLT32_MIN__ 1.17549435082228750796873653722224568e-38F32
00408 #define __ATOMIC_SEQ_CST 5
00409 #define __EXCEPTIONS 1
00410 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 1
00411 #define __UINT32_TYPE__ unsigned int
00412 #define __FLT32X_MIN_10_EXP__ (-307)
00413 #define __UINTPTR_TYPE__ long unsigned int
00414 #define __linux__ 1
00415 #define __LDBL_MIN_10_EXP__ (-4931)
00416 #define __cpp_generic_lambdas 201304L
00417 #define __FLT128_EPSILON__ 1.92592994438723585305597794258492732e-34F128
00418 #define __SIZEOF_LONG_LONG__ 8
00419 #define __cpp_user_defined_literals 200809L
00420 #define __FLT128_DECIMAL_DIG__ 36
00421 #define __GCC_ATOMIC_LLONG_LOCK_FREE 2
00422 #define __FLT_DECIMAL_DIG__ 9
00423 #define __UINT_FAST16_MAX__ 0xffffffffffffffffUL
00424 #define __STDC_IEC_559_COMPLEX__ 1
00425 #define __LDBL_NORM_MAX__ 1.18973149535723176508575932662800702e+4932L
00426 #define __FLT_MIN_10_EXP__ (-37)
00427 #define __GCC_ATOMIC_SHORT_LOCK_FREE 2
00428 #define __ORDER_LITTLE_ENDIAN__ 1234
00429 #define __SIZE_MAX__ 0xffffffffffffffffUL
00430 #define _GNU_SOURCE 1
00431 #define __UINT_LEAST32_MAX__ 0xffffffffU
00432 #define __cpp_init_captures 201304L
00433 #define __ATOMIC_ACQ_REL 4
00434 #define __ATOMIC_RELEASE 3
```

7.8 onedaydialog.h

```
00001 #ifndef ONEDAYDIALOG_H
00002 #define ONEDAYDIALOG_H
00003
00004 #include <QDialog>
00005
00006 namespace Ui {
00007 class OneDayDialog;
00008 }
00009
00010 class OneDayDialog : public QDialog
```

```
00011 {
00012     Q_OBJECT
00013
00014 public:
00015     explicit OneDayDialog(QWidget *parent = nullptr);
00016     ~OneDayDialog();
00017
00018 private:
00019     Ui::OneDayDialog *ui;
00020 };
00021
00022 #endif // ONEDAYDIALOG_H
```

7.9 onemonthdialog.h

```
00001 #ifndef ONEMONTHDIALOG_H
00002 #define ONEMONTHDIALOG_H
00003
00004 #include <QDialog>
00005
00006 namespace Ui {
00007 class OneMonthDialog;
00008 }
00009
00010 class OneMonthDialog : public QDialog
00011 {
00012     Q_OBJECT
00013
00014 public:
00015     explicit OneMonthDialog(QWidget *parent = nullptr);
00016     ~OneMonthDialog();
00017
00018 private:
00019     Ui::OneMonthDialog *ui;
00020 };
00021
00022 #endif // ONEMONTHDIALOG_H
```

7.10 oneweekdialog.h

```
00001 #ifndef ONEWEEKDIALOG_H
00002 #define ONEWEEKDIALOG_H
00003
00004 #include <QDialog>
00005
00006 namespace Ui {
00007 class OneWeekDialog;
00008 }
00009
00010 class OneWeekDialog : public QDialog
00011 {
00012     Q_OBJECT
00013
00014 public:
00015     explicit OneWeekDialog(QWidget *parent = nullptr);
00016     ~OneWeekDialog();
00017
00018 private:
00019     Ui::OneWeekDialog *ui;
00020 };
00021
00022 #endif // ONEWEEKDIALOG_H
```

7.11 optionsdialog.h

```
00001 #ifndef OPTIONSDIALOG_H
00002 #define OPTIONSDIALOG_H
00003
00004 #include <QDialog>
00005
00010 namespace Ui {
00011 class OptionsDialog;
00012 }
```

```
00013
00021 class OptionsDialog : public QDialog
00022 {
00023     Q_OBJECT
00024
00025 public:
00030     explicit OptionsDialog(QWidget *parent = nullptr);
00031
00035     ~OptionsDialog();
00036
00037 private:
00038     Ui::OptionsDialog *ui;
00039 };
00040
00041 #endif // OPTIONSDIALOG_H
```

7.12 sensoranalyticsdialog.h

```
00001 #ifndef SENSORANALYTICSDIALOG_H
00002 #define SENSORANALYTICSDIALOG_H
00003
00004 #include <QDialog>
00005 #include <QDebug>
00006
00011 namespace Ui {
00012 class SensorAnalyticsDialog;
00013 }
00014
00023 class SensorAnalyticsDialog : public QDialog
00024 {
00025     Q_OBJECT
00026
00027 public:
00032     explicit SensorAnalyticsDialog(QWidget *parent = nullptr);
00033
00037     ~SensorAnalyticsDialog();
00038
00039 private slots:
00043     void on_btnWaterTemp_clicked();
00044
00048     void on_btnWaterPressure_clicked();
00049
00053     void on_btnPowerConsumption_clicked();
00054
00058     void on_day_clicked();
00059
00063     void on_week_clicked();
00064
00068     void on_month_clicked();
00069
00070 signals:
00075     void intervalSelected(qreal seconds);
00076
00077 private:
00078     Ui::SensorAnalyticsDialog *ui;
00079 };
00080
00081 #endif // SENSORANALYTICSDIALOG_H
```

7.13 settingsdialog.h

```
00001 #ifndef SETTINGSDIALOG_H
00002 #define SETTINGSDIALOG_H
00003
00004 #include <QDialog>
00005
00010 namespace Ui {
00011 class SettingsDialog;
00012 }
00013
00021 class SettingsDialog : public QDialog
00022 {
00023     Q_OBJECT
00024
00025 public:
00030     explicit SettingsDialog(QWidget *parent = nullptr);
00031
00035     ~SettingsDialog();
```

```
00036
00037 private:
00038     Ui::SettingsDialog *ui;
00039 };
00040
00041 #endif // SETTINGSDIALOG_H
```

7.14 statistics.h

```
00001 #ifndef STATISTICS_H
00002 #define STATISTICS_H
00003
00004 #include <QDialog>
00005
00010 namespace Ui {
00011 class Statistics;
00012 }
00013
00021 class Statistics : public QDialog
00022 {
00023     Q_OBJECT
00024
00025 public:
00030     explicit Statistics(QWidget *parent = nullptr);
00031
00035     ~Statistics();
00036
00041     void setCupCount(int num);
00042
00043 private:
00044     Ui::Statistics *ui;
00045     int             m_cupCount = 0;
00046 };
00047
00048 #endif // STATISTICS_H
```

7.15 warningdialog.h

```
00001 #ifndef WARNINGDIALOG_H
00002 #define WARNINGDIALOG_H
00003
00004 #include <QDialog>
00005
00010 namespace Ui { class WarningDialog; }
00011
00019 class WarningDialog : public QDialog {
00020     Q_OBJECT
00021
00022     public:
```

7.16 websocketclient.h

```
00015 class WebSocketClient : public QThread {
00016     Q_OBJECT
00017
00018 public:
00023     explicit WebSocketClient(QObject* parent = nullptr);
00024
00029     void run() override;
00030
00031 signals:
00038     void dataReceived(double pressure, double temperature, const QString& flag);
00039 };
```

Index