SensorReaderApp

Generated by Doxygen 1.9.8

1 Hierarchical Index 1

1.1 Class Hierarchy . . . 1

2 Class Index 3

2.1 Class List . . . 3

3 File Index 5

3.1 File List . . . 5

4 Class Documentation 7

4.1 MainWindow Class Reference . . . 7

4.1.1 Detailed Description . . . 9

4.1.2 Constructor & Destructor Documentation . . . 9

4.1.2.1 MainWindow() . . . 9

4.2 qt_meta_stringdata_MainWindow_t Struct Reference . . . 9

4.3 SensorData Struct Reference . . . 10

4.3.1 Detailed Description . . . 10

4.4 SensorReader Class Reference . . . 10

4.4.1 Detailed Description . . . 11

4.4.2 Constructor & Destructor Documentation . . . 11

4.4.2.1 SensorReader() . . . 11

4.4.3 Member Function Documentation . . . 12

4.4.3.1 getData() . . . 12

4.4.3.2 openPort() . . . 12

4.4.3.3 parseSensorData() . . . 12

4.4.3.4 readData() . . . 12

5 File Documentation 15

5.1 main.cpp File Reference . . . 15

5.1.1 Detailed Description . . . 15

5.1.2 Function Documentation . . . 16

5.1.2.1 main() . . . 16

5.2 MainWindow.h . . . 16

5.3 moc_predefs.h . . . 16

5.4 SensorReader.h . . . 22

Index 23

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

QMainWindow
    MainWindow 7
qt_meta_stringdata_MainWindow_t 9
SensorData 10
SensorReader 10

Chapter 2

Class Index

2.1 Class List

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

MainWindow
: Klasa reprezentująca główne okno aplikacji 7

qt_meta_stringdata_MainWindow_t 9

SensorData
: Struktura przechowująca dane odczytane z czujników 10

SensorReader
: Klasa do obsługi czujników przez port szeregowy 10

Chapter 3

File Index

3.1 File List

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

main.cpp
    Główna funkcja programu 15
MainWindow.h 16
moc_predefs.h 16
SensorReader.h 22

Chapter 4

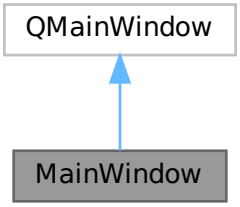
Class Documentation

4.1 MainWindow Class Reference

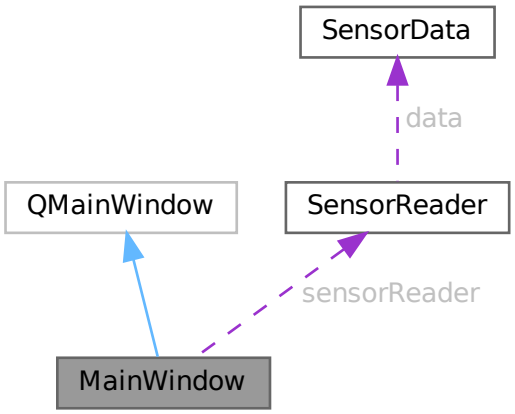
Klasa reprezentująca główne okno aplikacji.

`#include <MainWindow.h>`

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Public Member Functions

- MainWindow (SensorReader ∗reader, QWidget ∗parent=nullptr)

  Konstruktor klasy MainWindow.
- ∼**MainWindow** ()

  Destruktor klasy MainWindow.

Private Slots

- void **updateSensorData** ()

  Aktualizuje dane z czujników i wyświetla je w interfejsie.

Private Attributes

- QLabel ∗ **co2Label**

  Etykieta wyświetlająca poziom CO2.
- QLabel ∗ **co2TempLabel**

  Etykieta wyświetlająca temperaturę CO2.
- QLabel ∗ **co2HumLabel**

  Etykieta wyświetlająca wilgotność CO2.
- QLabel ∗ **pm1Label**

  Etykieta wyświetlająca poziom PM1.0.
- QLabel ∗ **pm25Label**

  Etykieta wyświetlająca poziom PM2.5.
- QLabel ∗ **pm10Label**

  Etykieta wyświetlająca poziom PM10.
- QLabel ∗ **radiationLabel**

  Etykieta wyświetlająca poziom promieniowania.
- QLabel ∗ **radiationDoseLabel**

Etykieta wyświetlająca dawkę promieniowania na godzinę.

- QLabel * **co2StatusLabel**

  Etykieta wyświetlająca status CO2.

- QLabel * **pmStatusLabel**

  Etykieta wyświetlająca status PM.

- QLabel * **radiationStatusLabel**

  Etykieta wyświetlająca status promieniowania.

- QTimer * **timer**

  Timer do cyklicznej aktualizacji danych.

- SensorReader * **sensorReader**

  Wskaźnik do obiektu SensorReader.

4.1.1 Detailed Description

Klasa reprezentująca główne okno aplikacji.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 MainWindow()

```
MainWindow::MainWindow (
            SensorReader * reader,
            QWidget * parent = nullptr )  [explicit]
```

Konstruktor klasy MainWindow.

Parameters

reader	Wskaźnik do obiektu SensorReader.
parent	Wskaźnik do obiektu nadrzędnego (domyślnie nullptr).

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

- MainWindow.h
- MainWindow.cpp

4.2 qt_meta_stringdata_MainWindow_t Struct Reference

Public Attributes

- QByteArrayData **data** [3]
- char **stringdata0** [29]

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

- moc_MainWindow.cpp

4.3 SensorData Struct Reference

Struktura przechowująca dane odczytane z czujników.

`#include <SensorReader.h>`

Public Attributes

- int **co2** = -1
  Poziom CO2 w ppm.
- int **co2_temp** = -1
  Temperatura CO2 w stopniach Celsjusza.
- int **co2_hum** = -1
  Wilgotność CO2 w procentach.
- int **pm1** = -1
  Poziom pyłu PM1.0 w µg/m^3.
- int **pm25** = -1
  Poziom pyłu PM2.5 w µg/m^3.
- int **pm10** = -1
  Poziom pyłu PM10 w µg/m^3.
- int **radiation** = -1
  Liczba zliczeń promieniowania (CPM).
- float **radiation_dose_per_hour** = -1.0
  Dawka promieniowania na godzinę w µSv/h.

4.3.1 Detailed Description

Struktura przechowująca dane odczytane z czujników.

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

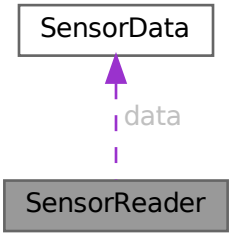
- SensorReader.h

4.4 SensorReader Class Reference

Klasa do obsługi czujników przez port szeregowy.

`#include <SensorReader.h>`

Collaboration diagram for SensorReader:



Public Member Functions

- SensorReader (const std::string &portname, int baudrate=B9600)

  Konstruktor klasy SensorReader.
- ~**SensorReader** ()

  Destruktor klasy SensorReader.
- bool openPort ()

  Otwiera port szeregowy.
- void **closePort** ()

  Zamyka port szeregowy.
- bool readData ()

  Odczytuje dane z portu szeregowego.
- SensorData getData () const

  Zwraca ostatnio odczytane dane z czujników.

Private Member Functions

- SensorData parseSensorData (const std::string &raw)

  Parsuje surowe dane z czujników.

Private Attributes

- std::string **portname**

  Nazwa portu szeregowego.
- int **baudrate**

  Prędkość transmisji.
- int **serial_port**

  Uchwyt do portu szeregowego.
- char **buffer** [256]

  Bufor do przechowywania danych z portu szeregowego.
- std::string **serial_data**

  Dane odczytane z portu szeregowego.
- SensorData **data**

  Struktura przechowująca dane z czujników.

4.4.1 Detailed Description

Klasa do obsługi czujników przez port szeregowy.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 SensorReader()

```
SensorReader::SensorReader (
            const std::string & portname,
            int baudrate = B9600 )
```

Konstruktor klasy SensorReader.

Parameters

portname	Nazwa portu szeregowego.
baudrate	Prędkość transmisji (domyślnie B9600).

4.4.3 Member Function Documentation

4.4.3.1 getData()

```
SensorData SensorReader::getData ( ) const
```

Zwraca ostatnio odczytane dane z czujników.

Returns

Struktura SensorData zawierająca dane z czujników.

4.4.3.2 openPort()

```
bool SensorReader::openPort ( )
```

Otwiera port szeregowy.

Returns

true, jeśli port został otwarty pomyślnie, false w przeciwnym razie.

4.4.3.3 parseSensorData()

```
SensorData SensorReader::parseSensorData (
            const std::string & raw )  [private]
```

Parsuje surowe dane z czujników.

Parameters

raw	Surowe dane w formie ciągu znaków.

Returns

Struktura SensorData zawierająca sparsowane dane.

4.4.3.4 readData()

```
bool SensorReader::readData ( )
```

Odczytuje dane z portu szeregowego.

Returns

true, jeśli dane zostały odczytane pomyślnie, false w przeciwnym razie.

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

- SensorReader.h
- SensorReader.cpp

Chapter 5

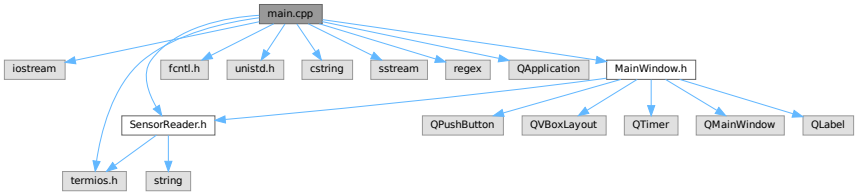
File Documentation

5.1 main.cpp File Reference

Główna funkcja programu.

```
#include <iostream>
#include "SensorReader.h"
#include <fcntl.h>
#include <unistd.h>
#include <cstring>
#include <sstream>
#include <regex>
#include <termios.h>
#include <QApplication>
#include "MainWindow.h"
```

Include dependency graph for main.cpp:



Functions

- int main (int argc, char ∗argv[ ])

  Główna funkcja programu.

5.1.1 Detailed Description

Główna funkcja programu.

5.1.2 Function Documentation

5.1.2.1 main()

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

Główna funkcja programu.

Inicjalizuje obiekt SensorReader, otwiera port szeregowy i uruchamia aplikację Qt.

Parameters

argc	Liczba argumentów wiersza poleceń.
argv	Tablica argumentów wiersza poleceń.

Returns

int Kod wyjścia programu.

5.2 MainWindow.h

```
00001 #ifndef MAINWINDOW_H
00002 #define MAINWINDOW_H
00003
00004 #include <QMainWindow>
00005 #include <QLabel>
00006 #include <QPushButton>
00007 #include <QVBoxLayout>
00008 #include <QTimer>
00009 #include "SensorReader.h"
00010
00015 class MainWindow : public QMainWindow
00016 {
00017     Q_OBJECT
00018
00019 public:
00025     explicit MainWindow(SensorReader* reader, QWidget *parent = nullptr);
00026
00030     ~MainWindow();
00031
00032 private slots:
00036     void updateSensorData();
00037
00038 private:
00039     QLabel* co2Label;
00040     QLabel* co2TempLabel;
00041     QLabel* co2HumLabel;
00042     QLabel* pm1Label;
00043     QLabel* pm25Label;
00044     QLabel* pm10Label;
00045     QLabel* radiationLabel;
00046     QLabel* radiationDoseLabel;
00047     QLabel* co2StatusLabel;
00048     QLabel* pmStatusLabel;
00049     QLabel* radiationStatusLabel;
00050
00051     QTimer* timer;
00052     SensorReader* sensorReader;
00053 };
00054
00055 #endif
```

5.3 moc_predefs.h

```
00001 #define __DBL_MIN_EXP__ (-1021)
```

```
00002 #define __cpp_nontype_template_parameter_auto 201606L
00003 #define __UINT_LEAST16_MAX__ 0xffff
00004 #define __FLT16_HAS_QUIET_NAN__ 1
00005 #define __ATOMIC_ACQUIRE 2
00006 #define __FLT128_MAX_10_EXP__ 4932
00007 #define __FLT_MIN__ 1.17549435082228750796873653722224568e-38F
00008 #define __GCC_IEC_559_COMPLEX 2
00009 #define __cpp_aggregate_nsdmi 201304L
00010 #define __UINT_LEAST8_TYPE__ unsigned char
00011 #define __SIZEOF_FLOAT80__ 16
00012 #define __BFLT16_DENORM_MIN__ 9.18354961579912115600575419704879436e-41BF16
00013 #define __INTMAX_C(c) c ## L
00014 #define __CHAR_BIT__ 8
00015 #define __UINT8_MAX__ 0xff
00016 #define __SCHAR_WIDTH__ 8
00017 #define __WINT_MAX__ 0xffffffffU
00018 #define __FLT32_MIN_EXP__ (-125)
00019 #define __cpp_static_assert 201411L
00020 #define __BFLT16_MIN_10_EXP__ (-37)
00021 #define __ORDER_LITTLE_ENDIAN__ 1234
00022 #define __WCHAR_MAX__ 0x7fffffff
00023 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 1
00024 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 1
00025 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 1
00026 #define __GCC_ATOMIC_CHAR_LOCK_FREE 2
00027 #define __GCC_IEC_559 2
00028 #define __FLT32X_DECIMAL_DIG__ 17
00029 #define __FLT_EVAL_METHOD__ 0
00030 #define __cpp_binary_literals 201304L
00031 #define __FLT64_DECIMAL_DIG__ 17
00032 #define __cpp_noexcept_function_type 201510L
00033 #define __GCC_ATOMIC_CHAR32_T_LOCK_FREE 2
00034 #define __cpp_variadic_templates 200704L
00035 #define __UINT_FAST64_MAX__ 0xffffffffffffffffUL
00036 #define __SIG_ATOMIC_TYPE__ int
00037 #define __DBL_MIN_10_EXP__ (-307)
00038 #define __FINITE_MATH_ONLY__ 0
00039 #define __cpp_variable_templates 201304L
00040 #define __FLT32X_MAX_EXP__ 1024
00041 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 1
00042 #define __FLT32_HAS_DENORM__ 1
00043 #define __UINT_FAST8_MAX__ 0xff
00044 #define __cpp_rvalue_reference 200610L
00045 #define __cpp_nested_namespace_definitions 201411L
00046 #define __DEC64_MAX_EXP__ 385
00047 #define __INT8_C(c) c
00048 #define __LDBL_HAS_INFINITY__ 1
00049 #define __INT_LEAST8_WIDTH__ 8
00050 #define __cpp_variadic_using 201611L
00051 #define __UINT_LEAST64_MAX__ 0xffffffffffffffffUL
00052 #define __INT_LEAST8_MAX__ 0x7f
00053 #define __cpp_attributes 200809L
00054 #define __cpp_capture_star_this 201603L
00055 #define __SHRT_MAX__ 0x7fff
00056 #define __LDBL_MAX__ 1.189731495357231765021263853030970 21e+4932L
00057 #define __FLT64X_MAX_10_EXP__ 4932
00058 #define __cpp_if_constexpr 201606L
00059 #define __BFLT16_MAX_10_EXP__ 38
00060 #define __BFLT16_MAX_EXP__ 128
00061 #define __LDBL_IS_IEC_60559__ 1
00062 #define __FLT64X_HAS_QUIET_NAN__ 1
00063 #define __UINT_LEAST8_MAX__ 0xff
00064 #define __GCC_ATOMIC_BOOL_LOCK_FREE 2
00065 #define __FLT128_DENORM_MIN__ 6.47517511943802511092443895822764655e-4966F128
00066 #define __UINTMAX_TYPE__ long unsigned int
00067 #define __cpp_nsdmi 200809L
00068 #define __BFLT16_DECIMAL_DIG__ 4
00069 #define __linux 1
00070 #define __DEC32_EPSILON__ 1E-6DF
00071 #define __FLT_EVAL_METHOD_TS_18661_3__ 0
00072 #define __OPTIMIZE__ 1
00073 #define __UINT32_MAX__ 0xffffffffU
00074 #define __GXX_EXPERIMENTAL_CXX0X__ 1
00075 #define __DBL_DENORM_MIN__ double(4.94065645841246544176568792868221372e-324L)
00076 #define __FLT128_MIN_EXP__ (-16381)
00077 #define __WINT_MIN__ 0U
00078 #define __FLT128_MIN_10_EXP__ (-4931)
00079 #define __FLT32X_IS_IEC_60559__ 1
00080 #define __INT_LEAST16_WIDTH__ 16
00081 #define __SCHAR_MAX__ 0x7f
00082 #define __FLT128_MANT_DIG__ 113
00083 #define __WCHAR_MIN__ (-__WCHAR_MAX__ - 1)
00084 #define __INT64_C(c) c ## L
00085 #define __GCC_ATOMIC_POINTER_LOCK_FREE 2
00086 #define __ATOMIC_SEQ_CST 5
00087 #define __unix 1
00088 #define __INT_LEAST64_MAX__ 0x7fffffffffffffffL
```

```
00089 #define __FLT32X_MANT_DIG__ 53
00090 #define __GCC_ATOMIC_CHAR16_T_LOCK_FREE 2
00091 #define __cpp_aligned_new 201606L
00092 #define __FLT32_MAX_10_EXP__ 38
00093 #define __FLT64X_EPSILON__ 1.08420217248550443400745280086994171e-19F64x
00094 #define __STDC_HOSTED__ 1
00095 #define __DEC64_MIN_EXP__ (-382)
00096 #define __cpp_decltype_auto 201304L
00097 #define __DBL_DIG__ 15
00098 #define __FLT_EPSILON__ 1.19209289550781250000000000000000000e-7F
00099 #define __GXX_WEAK__ 1
00100 #define __SHRT_WIDTH__ 16
00101 #define __FLT32_IS_IEC_60559__ 1
00102 #define __LDBL_MIN__ 3.36210314311209350626267781732175260e-4932L
00103 #define __DBL_IS_IEC_60559__ 1
00104 #define __DEC32_MAX__ 9.999999E96DF
00105 #define __cpp_threadsafe_static_init 200806L
00106 #define __cpp_enumerator_attributes 201411L
00107 #define __FLT64X_DENORM_MIN__ 3.64519953188247460252840593361941982e-4951F64x
00108 #define __FLT32X_HAS_INFINITY__ 1
00109 #define __unix__ 1
00110 #define __INT_WIDTH__ 32
00111 #define __STDC_IEC_559__ 1
00112 #define __STDC_ISO_10646__ 201706L
00113 #define __DECIMAL_DIG__ 21
00114 #define __STDC_IEC_559_COMPLEX__ 1
00115 #define __FLT64_EPSILON__ 2.22044604925031308084726333618164062e-16F64
00116 #define __gnu_linux__ 1
00117 #define __INT16_MAX__ 0x7fff
00118 #define __FLT64_MIN_EXP__ (-1021)
00119 #define __FLT64X_MIN_10_EXP__ (-4931)
00120 #define __LDBL_HAS_QUIET_NAN__ 1
00121 #define __cpp_return_type_deduction 201304L
00122 #define __FLT16_MIN_EXP__ (-13)
00123 #define __FLT64_MANT_DIG__ 53
00124 #define __FLT64X_MANT_DIG__ 64
00125 #define __BFLT16_DIG__ 2
00126 #define __GNUC__ 14
00127 #define __GXX_RTTI 1
00128 #define __pie__ 2
00129 #define __MMX__ 1
00130 #define __FLT_HAS_DENORM__ 1
00131 #define __SIZEOF_LONG_DOUBLE__ 16
00132 #define __BIGGEST_ALIGNMENT__ 16
00133 #define __STDC_UTF_16__ 1
00134 #define __FLT64_MAX_10_EXP__ 308
00135 #define __BFLT16_IS_IEC_60559__ 0
00136 #define __FLT16_MAX_10_EXP__ 4
00137 #define __cpp_delegating_constructors 200604L
00138 #define __DBL_MAX__ double(1.79769313486231570814527423731704357e+308L)
00139 #define __cpp_raw_strings 200710L
00140 #define __INT_FAST32_MAX__ 0x7fffffffffffffffL
00141 #define __DBL_HAS_INFINITY__ 1
00142 #define __INT64_MAX__ 0x7fffffffffffffffL
00143 #define __SIZEOF_FLOAT__ 4
00144 #define __HAVE_SPECULATION_SAFE_VALUE 1
00145 #define __cpp_fold_expressions 201603L
00146 #define __DEC32_MIN_EXP__ (-94)
00147 #define __INTPTR_WIDTH__ 64
00148 #define __UINT_LEAST32_MAX__ 0xffffffffU
00149 #define __FLT32X_HAS_DENORM__ 1
00150 #define __INT_FAST16_TYPE__ long int
00151 #define __MMX_WITH_SSE__ 1
00152 #define __LDBL_HAS_DENORM__ 1
00153 #define __SEG_GS 1
00154 #define __BFLT16_EPSILON__ 7.81250000000000000000000000000000000e-3BF16
00155 #define __cplusplus 201703L
00156 #define __cpp_ref_qualifiers 200710L
00157 #define __DEC32_MIN__ 1E-95DF
00158 #define __DEPRECATED 1
00159 #define __cpp_rvalue_references 200610L
00160 #define __DBL_MAX_EXP__ 1024
00161 #define __WCHAR_WIDTH__ 32
00162 #define __FLT32_MAX__ 3.40282346638528859811704183484516925e+38F32
00163 #define __DEC128_EPSILON__ 1E-33DL
00164 #define __FLT16_DECIMAL_DIG__ 5
00165 #define __SSE2_MATH__ 1
00166 #define __ATOMIC_HLE_RELEASE 131072
00167 #define __PTRDIFF_MAX__ 0x7fffffffffffffffL
00168 #define __amd64 1
00169 #define __ATOMIC_HLE_ACQUIRE 65536
00170 #define __GNUG__ 14
00171 #define __LONG_LONG_MAX__ 0x7fffffffffffffffLL
00172 #define __SIZEOF_SIZE_T__ 8
00173 #define __BFLT16_HAS_INFINITY__ 1
00174 #define __FLT64X_MIN_EXP__ (-16381)
00175 #define __SIZEOF_WINT_T__ 4
```

```
00176 #define __FLT32X_DIG__ 15
00177 #define __LONG_LONG_WIDTH__ 64
00178 #define __cpp_initializer_lists 200806L
00179 #define __FLT32_MAX_EXP__ 128
00180 #define __cpp_hex_float 201603L
00181 #define __GXX_ABI_VERSION 1019
00182 #define __FLT_MIN_EXP__ (-125)
00183 #define __GCC_HAVE_DWARF2_CFI_ASM 1
00184 #define __x86_64 1
00185 #define __cpp_lambdas 200907L
00186 #define __INT_FAST64_TYPE__ long int
00187 #define __BFLT16_MAX__ 3.389531389251535475904708003714878 67e+38BF16
00188 #define __FLT64_DENORM_MIN__ 4.94065645841246544176568792868221372e-324F64
00189 #define __cpp_template_auto 201606L
00190 #define __FLT16_DENORM_MIN__ 5.960464477539062500000000000000000000e-8F16
00191 #define __FLT128_EPSILON__ 1.92592994438723585305597794258492732e-34F128
00192 #define __FLT64X_NORM_MAX__ 1.189731495357231765021263853030970 21e+4932F64x
00193 #define __SIZEOF_POINTER__ 8
00194 #define __SIZE_TYPE__ long unsigned int
00195 #define __LP64__ 1
00196 #define __DBL_HAS_QUIET_NAN__ 1
00197 #define __FLT32X_EPSILON__ 2.220446049250313080847263336181640 62e-16F32x
00198 #define __LDBL_MAX_EXP__ 16384
00199 #define __DECIMAL_BID_FORMAT__ 1
00200 #define __FLT64_MIN_10_EXP__ (-307)
00201 #define __FLT16_MIN_10_EXP__ (-4)
00202 #define __FLT64X_DECIMAL_DIG__ 21
00203 #define __DEC128_MIN__ 1E-6143DL
00204 #define __REGISTER_PREFIX__
00205 #define __UINT16_MAX__ 0xffff
00206 #define __FLT128_HAS_INFINITY__ 1
00207 #define __FLT32_MIN__ 1.17549435082228750796873653722224568e-38F32
00208 #define __UINT8_TYPE__ unsigned char
00209 #define __FLT_DIG__ 6
00210 #define __DEC_EVAL_METHOD__ 2
00211 #define __FLT_MANT_DIG__ 24
00212 #define __LDBL_DECIMAL_DIG__ 21
00213 #define __VERSION__ "14.2.0"
00214 #define __UINT64_C(c) c ## UL
00215 #define __cpp_unicode_characters 201411L
00216 #define _STDC_PREDEF_H 1
00217 #define __INT_LEAST32_MAX__ 0x7fffffff
00218 #define __GCC_ATOMIC_INT_LOCK_FREE 2
00219 #define __FLT128_MAX_EXP__ 16384
00220 #define __FLT32_MANT_DIG__ 24
00221 #define __FLOAT_WORD_ORDER__ __ORDER_LITTLE_ENDIAN__
00222 #define __FLT32X_MIN_EXP__ (-1021)
00223 #define __STDC_IEC_60559_COMPLEX__ 201404L
00224 #define __cpp_aggregate_bases 201603L
00225 #define __BFLT16_MIN__ 1.17549435082228750796873653722224568e-38BF16
00226 #define __FLT128_HAS_DENORM__ 1
00227 #define __FLT32_DECIMAL_DIG__ 9
00228 #define __FLT128_DIG__ 33
00229 #define __INT32_C(c) c
00230 #define __DEC64_EPSILON__ 1E-15DD
00231 #define __ORDER_PDP_ENDIAN__ 3412
00232 #define __DEC128_MIN_EXP__ (-6142)
00233 #define __DEC128_MAX__ 9.999999999999999999999999999999999E6144DL
00234 #define __INT_FAST32_TYPE__ long int
00235 #define __UINT_LEAST16_TYPE__ short unsigned int
00236 #define __DEC128_MAX_EXP__ 6145
00237 #define unix 1
00238 #define __DBL_HAS_DENORM__ 1
00239 #define __cpp_rtti 199711L
00240 #define __UINT64_MAX__ 0xffffffffffffffffUL
00241 #define __FLT_IS_IEC_60559__ 1
00242 #define __GNUC_WIDE_EXECUTION_CHARSET_NAME "UTF-32LE"
00243 #define __FLT64X_DIG__ 18
00244 #define __INT8_TYPE__ signed char
00245 #define __cpp_digit_separators 201309L
00246 #define __ELF__ 1
00247 #define __GCC_ASM_FLAG_OUTPUTS__ 1
00248 #define __UINT32_TYPE__ unsigned int
00249 #define __BFLT16_HAS_QUIET_NAN__ 1
00250 #define __FLT_RADIX__ 2
00251 #define __INT_LEAST16_TYPE__ short int
00252 #define __LDBL_EPSILON__ 1.08420217248550443400745280086994171e-19L
00253 #define __UINTMAX_C(c) c ## UL
00254 #define __FLT16_DIG__ 3
00255 #define __k8 1
00256 #define __FLT32X_MIN__ 2.22507385850720138309023271733240406e-308F32x
00257 #define __SIG_ATOMIC_MAX__ 0x7fffffff
00258 #define __cpp_constexpr 201603L
00259 #define __GCC_ATOMIC_WCHAR_T_LOCK_FREE 2
00260 #define __USER_LABEL_PREFIX__
00261 #define __STDC_IEC_60559_BFP__ 201404L
00262 #define __SIZEOF_PTRDIFF_T__ 8
```

```
00263 #define __FLT64X_HAS_INFINITY__ 1
00264 #define __SIZEOF_LONG__ 8
00265 #define __LDBL_DIG__ 18
00266 #define __FLT64_IS_IEC_60559__ 1
00267 #define __x86_64__ 1
00268 #define __FLT16_IS_IEC_60559__ 1
00269 #define __FLT16_MAX_EXP__ 16
00270 #define __DEC32_SUBNORMAL_MIN__ 0.000001E-95DF
00271 #define __INT_FAST16_MAX__ 0x7fffffffffffffffL
00272 #define __GCC_CONSTRUCTIVE_SIZE 64
00273 #define __FLT64_DIG__ 15
00274 #define __UINT_FAST32_MAX__ 0xffffffffffffffffUL
00275 #define __UINT_LEAST64_TYPE__ long unsigned int
00276 #define __FLT16_EPSILON__ 9.765625000000000000000000000000000e-4F16
00277 #define __FLT_HAS_QUIET_NAN__ 1
00278 #define __FLT_MAX_10_EXP__ 38
00279 #define __FLT64X_HAS_DENORM__ 1
00280 #define __DEC128_SUBNORMAL_MIN__ 0.000000000000000000000000000000001E-6143DL
00281 #define __FLT_HAS_INFINITY__ 1
00282 #define __GNUC_EXECUTION_CHARSET_NAME "UTF-8"
00283 #define __cpp_unicode_literals 200710L
00284 #define __UINT_FAST16_TYPE__ long unsigned int
00285 #define __DEC64_MAX__ 9.99999999999999E384DD
00286 #define __INT_FAST32_WIDTH__ 64
00287 #define __CHAR16_TYPE__ short unsigned int
00288 #define __PRAGMA_REDEFINE_EXTNAME 1
00289 #define __SIZE_WIDTH__ 64
00290 #define __SEG_FS 1
00291 #define __INT_LEAST16_MAX__ 0x7fff
00292 #define __FLT16_NORM_MAX__ 6.550400000000000000000000000000000e+4F16
00293 #define __DEC64_MANT_DIG__ 16
00294 #define __FLT32_DENORM_MIN__ 1.40129846432481707092372958328991613e-45F32
00295 #define __SIG_ATOMIC_WIDTH__ 32
00296 #define __INT_LEAST64_TYPE__ long int
00297 #define __INT16_TYPE__ short int
00298 #define __INT_LEAST8_TYPE__ signed char
00299 #define __FLT16_MAX__ 6.550400000000000000000000000000000e+4F16
00300 #define __FLT128_MIN__ 3.362103143112093506262677781732175260e-4932F128
00301 #define __cpp_structured_bindings 201606L
00302 #define __SIZEOF_INT__ 4
00303 #define __DEC32_MAX_EXP__ 97
00304 #define __INT_FAST8_MAX__ 0x7f
00305 #define __FLT128_MAX__ 1.189731495357231765085759326628007 02e+4932F128
00306 #define __INTPTR_MAX__ 0x7fffffffffffffffL
00307 #define __cpp_sized_deallocation 201309L
00308 #define __cpp_guaranteed_copy_elision 201606L
00309 #define linux 1
00310 #define __FLT64_HAS_QUIET_NAN__ 1
00311 #define __FLT32_MIN_10_EXP__ (-37)
00312 #define __EXCEPTIONS 1
00313 #define __UINT16_C(c) c
00314 #define __PTRDIFF_WIDTH__ 64
00315 #define __LDBL_MANT_DIG__ 64
00316 #define __cpp_range_based_for 201603L
00317 #define __INT_FAST16_WIDTH__ 64
00318 #define __FLT64_HAS_INFINITY__ 1
00319 #define __FLT64X_MAX__ 1.189731495357231765021263853030970 21e+4932F64x
00320 #define __FLT16_HAS_INFINITY__ 1
00321 #define __STDCPP_DEFAULT_NEW_ALIGNMENT__ 16
00322 #define __SIG_ATOMIC_MIN__ (-__SIG_ATOMIC_MAX__ - 1)
00323 #define __code_model_small__ 1
00324 #define __GCC_ATOMIC_LONG_LOCK_FREE 2
00325 #define __cpp_nontype_template_args 201411L
00326 #define __DEC32_MANT_DIG__ 7
00327 #define __k8__ 1
00328 #define __INTPTR_TYPE__ long int
00329 #define __UINT16_TYPE__ short unsigned int
00330 #define __WCHAR_TYPE__ int
00331 #define __pic__ 2
00332 #define __UINTPTR_MAX__ 0xffffffffffffffffUL
00333 #define __INT_FAST64_WIDTH__ 64
00334 #define __cpp_decltype 200707L
00335 #define __INT_FAST64_MAX__ 0x7fffffffffffffffL
00336 #define __GCC_ATOMIC_TEST_AND_SET_TRUEVAL 1
00337 #define __FLT_NORM_MAX__ 3.402823466385288598117041834845169 25e+38F
00338 #define __FLT32_HAS_INFINITY__ 1
00339 #define __FLT64X_MAX_EXP__ 16384
00340 #define __UINT_FAST64_TYPE__ long unsigned int
00341 #define __cpp_inline_variables 201606L
00342 #define __BFLT16_MIN_EXP__ (-125)
00343 #define __INT_MAX__ 0x7fffffff
00344 #define __linux__ 1
00345 #define __INT64_TYPE__ long int
00346 #define __FLT_MAX_EXP__ 128
00347 #define __ORDER_BIG_ENDIAN__ 4321
00348 #define __DBL_MANT_DIG__ 53
00349 #define __cpp_inheriting_constructors 201511L
```

```
00350 #define __SIZEOF_FLOAT128__ 16
00351 #define __BFLT16_MANT_DIG__ 8
00352 #define __DEC64_MIN__ 1E-383DD
00353 #define __WINT_TYPE__ unsigned int
00354 #define __UINT_LEAST32_TYPE__ unsigned int
00355 #define __SIZEOF_SHORT__ 2
00356 #define __FLT32_NORM_MAX__ 3.40282346638528859811704183484516925e+38F32
00357 #define __SSE__ 1
00358 #define __LDBL_MIN_EXP__ (-16381)
00359 #define __FLT64_MAX__ 1.79769313486231570814527423731704357e+308F64
00360 #define __amd64__ 1
00361 #define __WINT_WIDTH__ 32
00362 #define __INT_LEAST64_WIDTH__ 64
00363 #define __FLT32X_MAX_10_EXP__ 308
00364 #define __cpp_namespace_attributes 201411L
00365 #define __SIZEOF_INT128__ 16
00366 #define __FLT16_MIN__ 6.10351562500000000000000000000000000e-5F16
00367 #define __FLT64X_IS_IEC_60559__ 1
00368 #define __LDBL_MAX_10_EXP__ 4932
00369 #define __ATOMIC_RELAXED 0
00370 #define __DBL_EPSILON__ double(2.22044604925031308084726333618164062e-16L)
00371 #define __INT_LEAST32_TYPE__ int
00372 #define _LP64 1
00373 #define __UINT8_C(c) c
00374 #define __FLT64_MAX_EXP__ 1024
00375 #define __SIZEOF_WCHAR_T__ 4
00376 #define __GNUC_PATCHLEVEL__ 0
00377 #define __FLT128_NORM_MAX__ 1.18973149535723176508575932662800702e+4932F128
00378 #define __FLT64_NORM_MAX__ 1.79769313486231570814527423731704357e+308F64
00379 #define __FLT128_HAS_QUIET_NAN__ 1
00380 #define __INTMAX_MAX__ 0x7fffffffffffffffL
00381 #define __INT_FAST8_TYPE__ signed char
00382 #define __FLT64X_MIN__ 3.36210314311209350626267781732175260e-4932F64x
00383 #define __STDCPP_THREADS__ 1
00384 #define __BFLT16_HAS_DENORM__ 1
00385 #define __GNUC_STDC_INLINE__ 1
00386 #define __FLT64_HAS_DENORM__ 1
00387 #define __FLT32_EPSILON__ 1.19209289550781250000000000000000000e-7F32
00388 #define __FLT16_HAS_DENORM__ 1
00389 #define __DBL_DECIMAL_DIG__ 17
00390 #define __STDC_UTF_32__ 1
00391 #define __INT_FAST8_WIDTH__ 8
00392 #define __FXSR__ 1
00393 #define __FLT32X_MAX__ 1.79769313486231570814527423731704357e+308F32x
00394 #define __DBL_NORM_MAX__ double(1.79769313486231570814527423731704357e+308L)
00395 #define __BYTE_ORDER__ __ORDER_LITTLE_ENDIAN__
00396 #define __GCC_DESTRUCTIVE_SIZE 64
00397 #define __INTMAX_WIDTH__ 64
00398 #define __cpp_runtime_arrays 198712L
00399 #define __FLT32_DIG__ 6
00400 #define __UINT64_TYPE__ long unsigned int
00401 #define __UINT32_C(c) c ## U
00402 #define __cpp_alias_templates 200704L
00403 #define __FLT_DENORM_MIN__ 1.40129846432481707092372958328991613e-45F
00404 #define __FLT128_IS_IEC_60559__ 1
00405 #define __INT8_MAX__ 0x7f
00406 #define __LONG_WIDTH__ 64
00407 #define __DBL_MIN__ double(2.22507385850720138309023271733240406e-308L)
00408 #define __PIC__ 2
00409 #define __INT32_MAX__ 0x7fffffff
00410 #define __UINT_FAST32_TYPE__ long unsigned int
00411 #define __FLT16_MANT_DIG__ 11
00412 #define __FLT32X_NORM_MAX__ 1.79769313486231570814527423731704357e+308F32x
00413 #define __CHAR32_TYPE__ unsigned int
00414 #define __FLT_MAX__ 3.40282346638528859811704183484516925e+38F
00415 #define __SSE2__ 1
00416 #define __cpp_deduction_guides 201703L
00417 #define __BFLT16_NORM_MAX__ 3.38953138925153547590470800371487867e+38BF16
00418 #define __INT32_TYPE__ int
00419 #define __SIZEOF_DOUBLE__ 8
00420 #define __cpp_exceptions 199711L
00421 #define __FLT_MIN_10_EXP__ (-37)
00422 #define __FLT64_MIN__ 2.22507385850720138309023271733240406e-308F64
00423 #define __INT_LEAST32_WIDTH__ 32
00424 #define __INTMAX_TYPE__ long int
00425 #define __GLIBCXX_BITSIZE_INT_N_0 128
00426 #define __FLT32X_HAS_QUIET_NAN__ 1
00427 #define __ATOMIC_CONSUME 1
00428 #define __GNUC_MINOR__ 2
00429 #define __GLIBCXX_TYPE_INT_N_0 __int128
00430 #define __UINTMAX_MAX__ 0xffffffffffffffffUL
00431 #define __PIE__ 2
00432 #define __FLT32X_DENORM_MIN__ 4.94065645841246544176568792868221372e-324F32x
00433 #define __cpp_template_template_args 201611L
00434 #define __DBL_MAX_10_EXP__ 308
00435 #define __LDBL_DENORM_MIN__ 3.64519953188247460252840593361941982e-4951L
00436 #define __INT16_C(c) c
```

```
00437 #define __STDC__ 1
00438 #define __PTRDIFF_TYPE__ long int
00439 #define __LONG_MAX__ 0x7fffffffffffffffL
00440 #define __FLT32X_MIN_10_EXP__ (-307)
00441 #define __UINTPTR_TYPE__ long unsigned int
00442 #define __DEC64_SUBNORMAL_MIN__ 0.000000000000001E-383DD
00443 #define __DEC128_MANT_DIG__ 34
00444 #define __LDBL_MIN_10_EXP__ (-4931)
00445 #define __cpp_generic_lambdas 201304L
00446 #define __SSE_MATH__ 1
00447 #define __SIZEOF_LONG_LONG__ 8
00448 #define __cpp_user_defined_literals 200809L
00449 #define __FLT128_DECIMAL_DIG__ 36
00450 #define __GCC_ATOMIC_LLONG_LOCK_FREE 2
00451 #define __FLT32_HAS_QUIET_NAN__ 1
00452 #define __FLT_DECIMAL_DIG__ 9
00453 #define __UINT_FAST16_MAX__ 0xffffffffffffffffUL
00454 #define __LDBL_NORM_MAX__ 1.189731495357231765021263853030970 21e+4932L
00455 #define __GCC_ATOMIC_SHORT_LOCK_FREE 2
00456 #define __SIZE_MAX__ 0xffffffffffffffffUL
00457 #define __UINT_FAST8_TYPE__ unsigned char
00458 #define _GNU_SOURCE 1
00459 #define __cpp_init_captures 201304L
00460 #define __ATOMIC_ACQ_REL 4
00461 #define __ATOMIC_RELEASE 3
```

5.4 SensorReader.h

```
00001 #ifndef SENSORREADER_H
00002 #define SENSORREADER_H
00003
00004 #include <string>
00005 #include <termios.h>
00006
00011 struct SensorData {
00012     int co2 = -1;
00013     int co2_temp = -1;
00014     int co2_hum = -1;
00015     int pm1 = -1;
00016     int pm25 = -1;
00017     int pm10 = -1;
00018     int radiation = -1;
00019     float radiation_dose_per_hour = -1.0;
00020 };
00021
00026 class SensorReader {
00027 public:
00033     SensorReader(const std::string& portname, int baudrate = B9600);
00034
00038     ~SensorReader();
00039
00044     bool openPort();
00045
00049     void closePort();
00050
00055     bool readData();
00056
00061     SensorData getData() const;
00062
00063 private:
00064     std::string portname;
00065     int baudrate;
00066     int serial_port;
00067     char buffer[256];
00068     std::string serial_data;
00069     SensorData data;
00070
00076     SensorData parseSensorData(const std::string& raw);
00077 };
00078
00079 #endif
```

Index

getData
    SensorReader, 12

main
    main.cpp, 16
main.cpp, 15
    main, 16
MainWindow, 7
    MainWindow, 9

openPort
    SensorReader, 12

parseSensorData
    SensorReader, 12

qt_meta_stringdata_MainWindow_t, 9

readData
    SensorReader, 12

SensorData, 10
SensorReader, 10
    getData, 12
    openPort, 12
    parseSensorData, 12
    readData, 12
    SensorReader, 11