Arduino Driver - Barometer

Generated by Doxygen 1.8.9.1

Sat Aug 15 2015 18:46:56

Contents

1 Hierarchical Index

1.1 Class Hierarchy

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

2 Class Index

2.1 Class List

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

3 File Index

3.1 File List

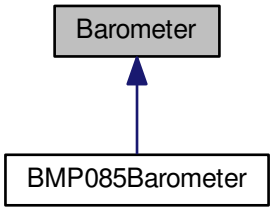
Here is a list of all files with brief descriptions:

4 Class Documentation

4.1 Barometer Class Reference

`#include <Barometer.h>`

Inheritance diagram for Barometer:



Public Member Functions

- virtual unsigned short getTemperature ()=0

- virtual unsigned short getPressure ()=0
- virtual unsigned short getAltitude ()=0

4.1.1 Detailed Description

Arduino - Barometer Driver.

Barometer.h

Barometer class.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 14 of file Barometer.h.

4.1.2 Member Function Documentation

4.1.2.1 virtual unsigned short Barometer::getAltitude ( ) [pure virtual]

Returns the current altitude.

Implemented in BMP085Barometer.

4.1.2.2 virtual unsigned short Barometer::getPressure ( ) [pure virtual]

Returns the current pressure.

Implemented in BMP085Barometer.

4.1.2.3 virtual unsigned short Barometer::getTemperature ( ) [pure virtual]

Returns the current temperature.

Implemented in BMP085Barometer.

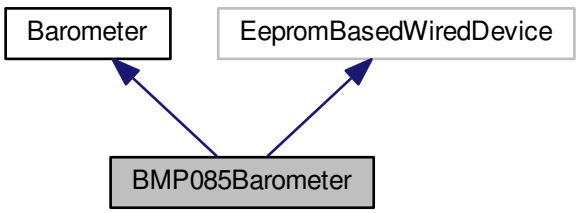
The documentation for this class was generated from the following file:

- Barometer.h

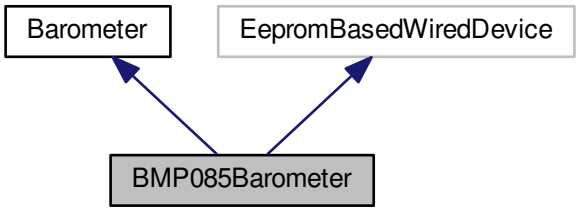
4.2 BMP085Barometer Class Reference

```
#include <BMP085Barometer.h>
```

Inheritance diagram for BMP085Barometer:



Collaboration diagram for BMP085Barometer:



Public Member Functions

- BMP085Barometer ()
- virtual unsigned short getTemperature ()
- virtual unsigned short getPressure ()
- virtual unsigned short getAltitude ()

Private Member Functions

- void readCallibration ()
- unsigned int readTemperature ()
- unsigned long readPressure ()

Private Attributes

- const unsigned char OSS
- const float PRESSURE_AT_SEA_LEVEL
- int ac1
- int ac2
- int ac3
- unsigned int ac4
- unsigned int ac5
- unsigned int ac6
- int b1
- int b2
- int mb
- int mc
- int md
- long b5

4.2.1 Detailed Description

Definition at line 19 of file BMP085Barometer.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 BMP085Barometer::BMP085Barometer ( )

Definition at line 4 of file BMP085Barometer.cpp.

4.2.3 Member Function Documentation

4.2.3.1 virtual unsigned short BMP085Barometer::getAltitude ( ) [virtual]

Returns the current altitude.

Implements Barometer.

4.2.3.2 unsigned short BMP085Barometer::getPressure ( ) [virtual]

Returns the current pressure.

Implements Barometer.

Definition at line 13 of file BMP085Barometer.cpp.

4.2.3.3 unsigned short BMP085Barometer::getTemperature ( ) [virtual]

Returns the current temperature.

Implements Barometer.

Definition at line 9 of file BMP085Barometer.cpp.

4.2.3.4 void BMP085Barometer::readCallibration ( ) [private]

Definition at line 17 of file BMP085Barometer.cpp.

4.2.3.5 unsigned long BMP085Barometer::readPressure ( ) [private]

Definition at line 47 of file BMP085Barometer.cpp.

4.2.3.6 unsigned int BMP085Barometer::readTemperature ( ) [private]

Reads the uncompressed temperatur.

Definition at line 31 of file BMP085Barometer.cpp.

4.2.4 Member Data Documentation

4.2.4.1 int BMP085Barometer::ac1 [private]

Definition at line 35 of file BMP085Barometer.h.

4.2.4.2 int BMP085Barometer::ac2 [private]

Definition at line 36 of file BMP085Barometer.h.

4.2.4.3 int BMP085Barometer::ac3 [private]

Definition at line 37 of file BMP085Barometer.h.

4.2.4.4 unsigned int BMP085Barometer::ac4 [private]

Definition at line 38 of file BMP085Barometer.h.

4.2.4.5 unsigned int BMP085Barometer::ac5 [private]

Definition at line 39 of file BMP085Barometer.h.

4.2.4.6 unsigned int BMP085Barometer::ac6 [private]

Definition at line 40 of file BMP085Barometer.h.

4.2.4.7 int BMP085Barometer::b1 [private]

Definition at line 41 of file BMP085Barometer.h.

4.2.4.8 int BMP085Barometer::b2 [private]

Definition at line 42 of file BMP085Barometer.h.

4.2.4.9 long BMP085Barometer::b5 [private]

Definition at line 46 of file BMP085Barometer.h.

4.2.4.10 int BMP085Barometer::mb [private]

Definition at line 43 of file BMP085Barometer.h.

4.2.4.11 int BMP085Barometer::mc [private]

Definition at line 44 of file BMP085Barometer.h.

4.2.4.12 int BMP085Barometer::md [private]

Definition at line 45 of file BMP085Barometer.h.

4.2.4.13 const unsigned char BMP085Barometer::OSS [private]

Definition at line 30 of file BMP085Barometer.h.

4.2.4.14 const float BMP085Barometer::PRESSURE_AT_SEA_LEVEL [private]

Definition at line 33 of file BMP085Barometer.h.

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

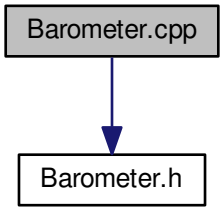
- BMP085Barometer.h
- BMP085Barometer.cpp

5 File Documentation

5.1 Barometer.cpp File Reference

```
#include "Barometer.h"
```

Include dependency graph for Barometer.cpp:

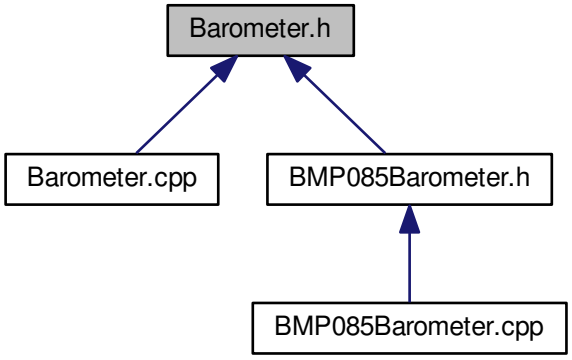


5.2 Barometer.cpp

```
00001 #include "Barometer.h"
```

5.3 Barometer.h File Reference

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



Classes

- class Barometer

5.4 Barometer.h

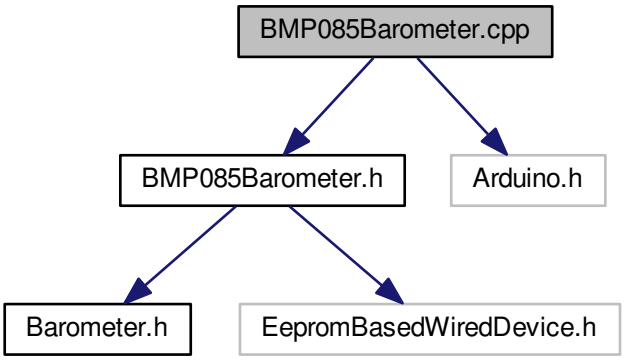
```
00001
00011 #ifndef __ARDUINO_BAROMETER_DRIVER_BAROMETER_H__
00012 #define __ARDUINO_BAROMETER_DRIVER_BAROMETER_H__ 1
00013
00014 class Barometer {
00015
00016 public:
```

```
00017
00021     virtual unsigned short getTemperature() = 0;
00022
00026     virtual unsigned short getPressure() = 0;
00027
00031     virtual unsigned short getAltitude() = 0;
00032 };
00033
00034 #endif // __ARDUINO_BAROMETER_DRIVER_BAROMETER_H__
```

5.5 BMP085Barometer.cpp File Reference

```
#include "BMP085Barometer.h"
#include <Arduino.h>
```

Include dependency graph for BMP085Barometer.cpp:



5.6 BMP085Barometer.cpp

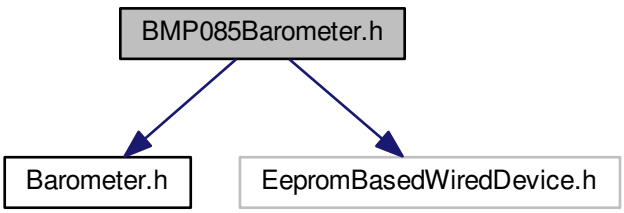
```
00001 #include "BMP085Barometer.h"
00002 #include <Arduino.h>
00003
00004 BMP085Barometer::BMP085Barometer()
00005         : EepromBasedWiredDevice(BMP085_ADDRESS, 0x01, EepromBasedWiredDevice::BIG_ENDIAN),
      OSS(0), PRESSURE_AT_SEA_LEVEL(101325) {
00006     readCallibration();
00007 }
00008
00009 unsigned short BMP085Barometer::getTemperature() {
00010
00011 }
00012
00013 unsigned short BMP085Barometer::getPressure() {
00014
00015 }
00016
00017 void BMP085Barometer::readCallibration() {
00018     readBlock(0xaa, (unsigned char*) &ac1, 2);
00019     readBlock(0xac, (unsigned char*) &ac2, 2);
00020     readBlock(0xae, (unsigned char*) &ac3, 2);
00021     readBlock(0xb0, (unsigned char*) &ac4, 2);
00022     readBlock(0xb2, (unsigned char*) &ac5, 2);
00023     readBlock(0xb4, (unsigned char*) &ac6, 2);
00024     readBlock(0xb6, (unsigned char*) &b1, 2);
00025     readBlock(0xb8, (unsigned char*) &b2, 2);
00026     readBlock(0xba, (unsigned char*) &mb, 2);
00027     readBlock(0xbc, (unsigned char*) &mc, 2);
00028     readBlock(0xbe, (unsigned char*) &md, 2);
00029 }
00030
00031 unsigned int BMP085Barometer::readTemperature() {
```

```
00032     unsigned int temperature;
00033
00034     // Write 0x2E into Register 0xF4
00035     // This requests a temperature reading
00036     unsigned char temperatureRequest[] = { 0x2e };
00037     writeBlock(0xf4, temperatureRequest, 0x01);
00038
00039     // Wait at least 4.5ms
00040     delay(5);
00041
00042     // Read two bytes from registers 0xF6 and 0xF7
00043     readBlock(0xf6, (unsigned char *) &temperature, 0x02);
00044     return temperature;
00045 }
00046
00047 unsigned long BMP085Barometer::readPressure() {
00048
00049     unsigned long pressure = 0;
00050
00051     // Write 0x34+(OSS<<6) into register 0xF4
00052     // Request a pressure reading w/ over-sampling setting
00053     unsigned char pressureRequest[] = { (unsigned char)(0x34 + (OSS << 6)) };
00054     writeBlock(0xf4, pressureRequest, 0x01);
00055
00056     // Wait for conversion, delay time dependent on OSS
00057     delay(2 + (3 << OSS));
00058
00059     // Read register 0xF6 (MSB), 0xF7 (LSB), and 0xF8 (XLSB)
00060     readBlock(0xf6, (unsigned char *) &pressure, 0x03);
00061
00062     // Adjust over-sampling
00063     pressure = (pressure & ~0xff) | (pressure & 0xff) >> (8 - OSS);
00064
00065     return pressure;
00066 }
```

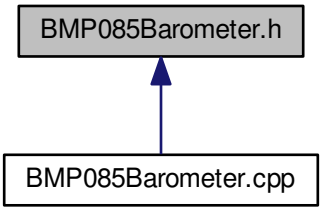
5.7 BMP085Barometer.h File Reference

```
#include <Barometer.h>
#include <EepromBasedWiredDevice.h>
```

Include dependency graph for BMP085Barometer.h:



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



Classes

- class BMP085Barometer

Macros

- #define BMP085_ADDRESS 0x77

5.7.1 Macro Definition Documentation

5.7.1.1 #define BMP085_ADDRESS 0x77

Arduino - Barometer Driver.

BMP085Barometer.h

BMP085Barometer class.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 14 of file BMP085Barometer.h.

5.8 BMP085Barometer.h

```
00001
00011 #ifndef __ARDUINO_BAROMETER_DRIVER_BMP085_BAROMETER_H__
00012 #define __ARDUINO_BAROMETER_DRIVER_BMP085_BAROMETER_H__ 1
00013
00014 #define BMP085_ADDRESS      0x77
00015
00016 #include <Barometer.h>
00017 #include <EepromBasedWiredDevice.h>
00018
00019 class BMP085Barometer: public Barometer, public EepromBasedWiredDevice {
00020
00021     /*
00022      * OSS selects which mode the BMP085 operates in, and can be set to either 0, 1, 2, or 3.
00023      * OSS determines how many samples the BMP085 will take before it sends over its uncompensated
00024      * pressure reading. With OSS set to 0, the BMP085 will consume the least current.
00025      * Setting OSS to 3 increases resolution, as it samples pressure eight times before
00026      * producing a reading, this comes at a cost of more power usage. If you want to change OSS,
00027      * just set it accordingly at the top of the program. Try changing OSS to 3,
00028      * does the data become more stable?
00029      */
00030     const unsigned char OSS;
```

```
00031
00032     // Pressure at sea level (Pa)
00033     const float PRESSURE_AT_SEA_LEVEL;
00034
00035     int ac1;
00036     int ac2;
00037     int ac3;
00038     unsigned int ac4;
00039     unsigned int ac5;
00040     unsigned int ac6;
00041     int b1;
00042     int b2;
00043     int mb;
00044     int mc;
00045     int md;
00046     long b5;
00047
00048 public:
00049
00050     BMP085Barometer();
00051
00052     virtual unsigned short getTemperature();
00053
00054     virtual unsigned short getPressure();
00055
00056     virtual unsigned short getAltitude();
00057
00058 private:
00059
00060     void readCallibration();
00061
00065     unsigned int readTemperature();
00066
00067     unsigned long readPressure();
00068 };
00069
00070 #endif // __ARDUINO_BAROMETER_DRIVER_BMP085_BAROMETER_H__
```

Index