Bosch_BME280_Arduino

Generated by Doxygen 1.9.1

1 README 1

1.1 Bosch BME280 Arduino 1

1.1.1 About 1

1.1.2 Functionality 2

1.1.3 Namespace 2

1.1.4 Methods 2

1.1.4.1 Public 2

1.1.4.2 Example 3

1.1.5 Compatibility 3

1.1.6 Copyright 3

2 Class Index 5

2.1 Class List 5

3 File Index 7

3.1 File List 7

4 Class Documentation 9

4.1 BME::Bosch_BME280 Class Reference 9

4.1.1 Constructor & Destructor Documentation 9

4.1.1.1 Bosch_BME280() 9

4.1.2 Member Function Documentation 10

4.1.2.1 begin() 10

4.1.2.2 getHumidity() 10

4.1.2.3 getPressure() 10

4.1.2.4 getSealevelForAltitude() 11

4.1.2.5 getSensorStatus() 11

4.1.2.6 getTemperature() 11

4.1.2.7 measure() 11

4.1.2.8 setSensorStatus() 12

5 File Documentation 13

5.1 src/Bosch_BME280_Arduino.h File Reference 13

5.1.1 Detailed Description 13

Index 15

Chapter 1

README

1.1 Bosch BME280 Arduino

based on Bosch BME280_driver v3.5.1

List of content

- About
- Functionality
- Namespace
- Methods
- Example
- Compatibility
- Copyright

1.1.1 About

The Bosch BME280 is an environmental sensor which is able to measure temperature, humidity and air pressure.

This library is based on the Bosch Sensortec BME280 driver API v3.5.1, and is intended to measure these environmental signals via I^{2}C connection on an Arduino based or ESP based microcontroller.

The github repository of Bosch Sensortec is: Github BOSCH Sensor Driver

The website of the BME280 on Bosch Sensortec is: Bosch Sensortec BME280

1.1.2 Functionality

The original Bosch driver is included in this package and it has not been modified in any way. The Bosch BME280 sensor do have 3 operation modes.

1. **Sleep mode** - the sensor is in sleep mode after power on reset. No measurements are performed and power consumption is on minimum.
2. **Forced mode** - one single measurement is performed and returns then to sleep mode. The measurements can be obtained from the data registers.
3. **Normal mode** - cyclic measurements are performed. The measurements can be obtained from the data registers.

1.1.3 Namespace

This Bosch BME280 wrapper uses a namespace as `BME` so if you construct the object you have to call:

```
BME::Bosch_BME280 bme{BME280_I2C_ADDR_PRIM, 249.67F, false};
```

1.1.4 Methods

1.1.4.1 Public

The are the following public methods:

1.1.4.1.1 Constructor

You call the constructor with various parameters:

- address of the BME280 (0x76 or 0x77)
- altitude for the calculation of the sea level pressure
- a Bool - `true` if use forced mode or `false` if use normal mode
  ```
  Bosch_BME280(addr, altitude, forced_mode)
  ```

1.1.4.1.2 Init I^{2}C and Sensor Init `begin()`

1.1.4.1.3 Measurement `measure()`

1.1.4.1.4 Data Query

These four methods returns the temperature, humidity and pressure in float.

```
getTemperature()
getHumidity()
getPressure()
getSealevelForAltitude()
```

1.1.4.1.5 Sensor Status

Also it is possible to get and set the sensor status.

```
int8_t status = getSensorStatus();
setSensorStatus(status);
```

1.1.4.2 Example

See also in:

- Arduino_example.ino
- ESP32_example.ino
- ESP8266_example.ino

```
#include <Arduino.h>
#include <Bosch_BME280_Arduino.h>
BME::Bosch_BME280 bme{BME280_I2C_ADDR_PRIM, 249.67F, true};
void setup() {
    Serial.begin(115200);
    while (!Serial) {
      yield();
    }
   // SDA, SCL needed for ESPs
#if defined (ESP8266)
  Wire.begin(SDA, SCL);
#elif defined (ESP32)
  Wire.setPins(SDA, SCL);
  Wire.begin();
#else
  Wire.begin();
#endif
    // init Bosch BME 280 Sensor
    if (bme.begin() != 0) {
      Serial.println("\n\t»> ERROR: Init of Bosch BME280 Sensor failed! «<");
    }
}
void loop() {
    static unsigned long tic {millis()};
    unsigned long ms = millis();
    if (ms - tic >= 2000) {
      tic = ms;
      bme.measure();
      Serial.print("\n\tTemperature:\t");
      Serial.println(bme.getTemperature());
      Serial.print("\tHumidity:\t");
      Serial.println(bme.getHumidity());
      Serial.print("\tPressure at NN:\t");
      Serial.println(bme.getSealevelForAltitude());
    }
}
```

1.1.5 Compatibility

Tested with:

- Arduino Nano
- ESP8266
- ESP32
- Arduino Nano 33 IOT

1.1.6 Copyright

The Files of the original Bosch BME280 driver API:

- bme280.c
- bme280.h
- bme280_defs.h

are Copyright (c) 2013 - 2017 by Bosch Sensortec GmbH

back to top

Chapter 2

Class Index

2.1 Class List

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

BME::Bosch_BME280 9

Chapter 3

File Index

3.1 File List

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

src/Bosch_BME280_Arduino.h
Bosch BME280 Arduino Wrapper Class based on BME280 Bosch driver v3.5.1 13

Chapter 4

Class Documentation

4.1 BME::Bosch_BME280 Class Reference

Public Member Functions

- Bosch_BME280 (uint8_t addr=BME280_I2C_ADDR_PRIM, float altitude=249.67F, bool forced_mode=true)

  Construct a new bme::Bosch_BME280 Object.
- int8_t begin ()

  setup the I2C Wiring and init the Sensor
- int8_t measure ()

  measure function
- float getTemperature () const

  Get the temperature from the internal BME data object.
- float getHumidity () const

  Get the Humidity from the internal BME data object.
- float getPressure () const

  Get the air pressure from the internal BME data object.
- float getSealevelForAltitude () const

  Get the Sealevel For Altitude from the internal BME data object.
- int8_t getSensorStatus () const

  Get the sensor status.
- void setSensorStatus (int8_t sensor_status)

  set sensor status

4.1.1 Constructor & Destructor Documentation

4.1.1.1 Bosch_BME280()

```
BME::Bosch_BME280::Bosch_BME280 (
             uint8_t addr = BME280_I2C_ADDR_PRIM,
             float altitude = 249.67F,
             bool forced_mode = true )  [explicit]
```

Construct a new bme::Bosch_BME280 Object.

Parameters

addr	I^2C-Address for sensor (0x76 default)
altitude	Altitude for the calculation of the Air Pressure at NN
forced_mode	if true the sensor makes one measurement and goes to sleep (no continuous measurement)

4.1.2 Member Function Documentation

4.1.2.1 begin()

```
int8_t BME::Bosch_BME280::begin ( )
```

setup the I2C Wiring and init the Sensor

Returns

sensor status

Return values

0	Success
$>$*0*	Warning
$<$*0*	Fail

4.1.2.2 getHumidity()

```
float BME::Bosch_BME280::getHumidity ( ) const  [inline]
```

Get the Humidity from the internal BME data object.

Returns

humidity in %

4.1.2.3 getPressure()

```
float BME::Bosch_BME280::getPressure ( ) const  [inline]
```

Get the air pressure from the internal BME data object.

Returns

air pressure in hecto pascal (hPa)

4.1.2.4 getSealevelForAltitude()

```
float BME::Bosch_BME280::getSealevelForAltitude ( ) const  [inline]
```

Get the Sealevel For Altitude from the internal BME data object.

Returns

sea level for altitude in meter

4.1.2.5 getSensorStatus()

```
int8_t BME::Bosch_BME280::getSensorStatus ( ) const  [inline]
```

Get the sensor status.

Returns

sensor status

Return values

0	Success
>0	Warning
<0	Fail

4.1.2.6 getTemperature()

```
float BME::Bosch_BME280::getTemperature ( ) const  [inline]
```

Get the temperature from the internal BME data object.

Returns

temperature in degree celsius

4.1.2.7 measure()

```
int8_t BME::Bosch_BME280::measure ( )
```

measure function

Returns

sensor status

Return values

0	Success
>0	Warning
<0	Fail

4.1.2.8 setSensorStatus()

```
void BME::Bosch_BME280::setSensorStatus (
            int8_t sensor_status )
```

set sensor status

Parameters

sensor_status	

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

- src/Bosch_BME280_Arduino.h
- src/Bosch_BME280_Arduino.cpp

Chapter 5

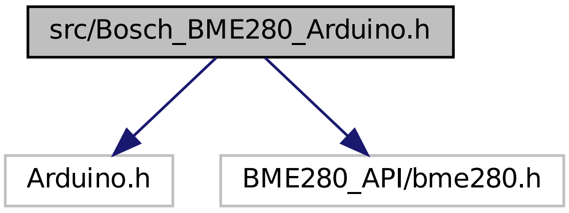
File Documentation

5.1 src/Bosch_BME280_Arduino.h File Reference

Bosch BME280 Arduino Wrapper Class based on BME280 Bosch driver v3.5.1.

```
#include <Arduino.h>
#include "BME280_API/bme280.h"
```

Include dependency graph for Bosch_BME280_Arduino.h:



Classes

- class BME::Bosch_BME280

5.1.1 Detailed Description

Bosch BME280 Arduino Wrapper Class based on BME280 Bosch driver v3.5.1.

Author

Frank Häfele

Date

21.02.2022

Version

1.2.0

Index

begin
    BME::Bosch_BME280, 10
BME::Bosch_BME280, 9
    begin, 10
    Bosch_BME280, 9
    getHumidity, 10
    getPressure, 10
    getSealevelForAltitude, 10
    getSensorStatus, 11
    getTemperature, 11
    measure, 11
    setSensorStatus, 12
Bosch_BME280
    BME::Bosch_BME280, 9

getHumidity
    BME::Bosch_BME280, 10
getPressure
    BME::Bosch_BME280, 10
getSealevelForAltitude
    BME::Bosch_BME280, 10
getSensorStatus
    BME::Bosch_BME280, 11
getTemperature
    BME::Bosch_BME280, 11

measure
    BME::Bosch_BME280, 11

setSensorStatus
    BME::Bosch_BME280, 12
src/Bosch_BME280_Arduino.h, 13