Erriez BMP280/BME280 library for Arduino

1.0.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

BMP280/BME280 sensor library for Arduino

This is a BMP280/BME280 temperature/pressure/humidity sensor library for Arduino.

Arduino library features

- Measurements:
  - BMP280: Temperature / pressure / approximate altitude
  - BME280: Temperature / pressure / approximate altitude / humidity
- Three operation modes:
  - Normal mode: Continues conversion
  - Forced mode: One-time conversion
  - Standby mode: Low-power, no conversion
- Sampling configuration
- Chip detect / read chip ID
- I2C interface only
- Small flash/RAM footprint

BMP280/BME280 sensor specifications

- Operating voltage: 1.71V .. 3.6V max
- Low current:
  - 1.8 uA @ 1 Hz humidity and temperature
  - 2.8 uA @ 1 Hz pressure and temperature
  - 3.6 uA @ 1 Hz humidity, pressure and temperature
  - 0.1 uA in sleep mode
- Operating range: -40...+85 °C, 0...100 % rel. humidity, 300...1100 hPa
- I2C bus interface: max 3.4 MHz
- No additional electronic components needed

Hardware

Connection Arduino board - BMX280 sensor

Pins board - BMX280	VCC	GND	SDA	SCL
Arduino UNO (ATMega328 boards)	5V	GND	A4	A5
Arduino Mega2560	5V	GND	D20	D21
Arduino Leonardo	5V	GND	D2	D3
Arduino DUE (ATSAM3X8E)	3V3	GND	20	21
ESP8266	3V3	GND	GPIO4 (D2)	GPIO5 (D1)
ESP32	3V3	GND	GPIO21	GPIO22

Other unlisted MCU's may work, but are not tested.

Examples

Examples | Erriez BMP280/BME280 sensor:

- ErriezBMX280

Documentation

- Doxygen online HTML
- Doxygen PDF
- BMP280 chip datasheet
- BME280 chip datasheet

Example

```
{c++}
#include <Wire.h>
#include <ErriezBMX280.h>

// Adjust sea level for altitude calculation
#define SEA_LEVEL_PRESSURE_HPA      1026.25

// Create BMX280 object I2C address 0x76 or 0x77
ErriezBMX280 bmx280 = ErriezBMX280(0x76);


void setup()
{
    // Initialize serial
    delay(500);
    Serial.begin(115200);
    while (!Serial) {
        ;
    }
    Serial.println(F("\nErriez BMP280/BMX280 example"));

    // Initialize I2C bus
    Wire.begin();
    Wire.setClock(400000);

    // Initialize sensor
    while (!bmx280.begin()) {
        Serial.println(F("Error: Could not detect sensor"));
        delay(3000);
    }

    // Print sensor type
    Serial.print(F("\nSensor type: "));
    switch (bmx280.getChipID()) {
        case CHIP_ID_BMP280:
```

```
            Serial.println(F("BMP280\n"));
            break;
        case CHIP_ID_BME280:
            Serial.println(F("BME280\n"));
            break;
        default:
            Serial.println(F("Unknown\n"));
            break;
    }
}

void loop()
{
    Serial.print(F("Temperature: "));
    Serial.print(bmx280.readTemperature());
    Serial.println(" C");

    if (bmx280.getChipID() == CHIP_ID_BME280) {
        Serial.print(F("Humidity:    "));
        Serial.print(bmx280.readHumidity());
        Serial.println(" %");
    }

    Serial.print(F("Pressure:    "));
    Serial.print(bmx280.readPressure() / 100.0F);
    Serial.println(" hPa");

    Serial.print(F("Altitude:    "));
    Serial.print(bmx280.readAltitude(SEA_LEVEL_PRESSURE_HPA));
    Serial.println(" m");

    Serial.println();

    delay(1000);
}
```

Output

```
{c++}
Erriez BMP280/BMX280 example

Sensor type: BME280

Temperature: 28.50 C
Humidity:    45.13 %
Pressure:    1024.88 hPa
Altitude:    11.27 m

Temperature: 28.55 C
Humidity:    45.21 %
Pressure:    1024.89 hPa
Altitude:    11.21 m

...
```

Set sampling

The sensor sampling and mode can be configured with function `setSampling()`. Recommended modes of operation according to the datasheet chapter "Recommended modes of operation":

```
{c++}
// Set sampling
//
// Weather
//  - forced mode, 1 sample / minute
//  - pressure ×1, temperature ×1, humidity ×1
//  - filter off
//
// Humidity sensing
//  - forced mode, 1 sample / second
//  - pressure ×0, temperature ×1, humidity ×1
//  - filter off
//
// Indoor navigation
//  - normal mode, t standby = 0.5 ms
//  - pressure ×16, temperature ×2, humidity ×1
//  - filter coefficient 16
```

```
//
// Gaming
//  - forced mode, t standby = 0.5 ms
//  - pressure ×1, temperature ×1, humidity ×1
//  - filter off
bmx280.setSampling(BMX280_MODE_NORMAL,     // SLEEP, FORCED, NORMAL
                   BMX280_SAMPLING_X16,    // Temp:  NONE, X1, X2, X4, X8, X16
                   BMX280_SAMPLING_X16,    // Press: NONE, X1, X2, X4, X8, X16
                   BMX280_SAMPLING_X16,    // Hum:   NONE, X1, X2, X4, X8, X16 (BME280)
                   BMX280_FILTER_X16,      // OFF, X2, X4, X8, X16
                   BMX280_STANDBY_MS_500);// 0_5, 10, 20, 62_5, 125, 250, 500, 1000
```

Library dependencies

- Built-in `Wire.h`

Library installation

Please refer to the Wiki page.

Other Arduino Libraries and Sketches from Erriez

- Erriez Libraries and Sketches

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 ErriezBMX280 Class Reference

BMX280 class.

`#include <ErriezBMX280.h>`

Public Member Functions

- ErriezBMX280 (uint8_t i2cAddr)
  Constructor.
- bool begin ()
  Sensor initialization.
- uint8_t getChipID ()
  Get chip ID.
- float readTemperature ()
  Read temperature.
- float readPressure ()
  Read pressure.
- float readAltitude (float seaLevel)
  Read approximate altitude.
- float readHumidity ()
  Read humidity (BME280 only)
- void setSampling (BMX280_Mode_e mode=BMX280_MODE_NORMAL, BMX280_Sampling_e tempSampling=BMX280_SAMPLING_X16, BMX280_Sampling_e pressSampling=BMX280_SAMPLING_X16, BMX280_Sampling_e humSampling=BMX280_SAMPLING_X16, BMX280_Filter_e filter=BMX280_FILTER_OFF, BMX280_Standby_e standbyDuration=BMX280_STANDBY_MS_0_5)
  Set sampling registers.
- uint8_t read8 (uint8_t reg)
  Read from 8-bit register.
- uint16_t read16 (uint8_t reg)
  Read from 16-bit register.
- uint16_t read16_LE (uint8_t reg)
  Read from 16-bit unsigned register little endian.
- int16_t readS16_LE (uint8_t reg)
  Read from 16-bit signed register little endian.
- uint32_t read24 (uint8_t reg)
  Read from 24-bit register.
- void write8 (uint8_t reg, uint8_t value)
  Write to 8-bit register.

4.1.1 Detailed Description

BMX280 class.

Definition at line 134 of file ErriezBMX280.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 ErriezBMX280()

```
ErriezBMX280::ErriezBMX280 (
            uint8_t i2cAddr )
```

Constructor.

Parameters

i2cAddr	I2C address

Definition at line 43 of file ErriezBMX280.cpp.

4.1.3 Member Function Documentation

4.1.3.1 begin()

```
bool ErriezBMX280::begin ( )
```

Sensor initialization.

Return values

true	BMP280 or BME280 sensor detected
false	Error: No (supported) sensor detected

Definition at line 55 of file ErriezBMX280.cpp.

4.1.3.2 getChipID()

```
uint8_t ErriezBMX280::getChipID ( )
```

Get chip ID.

Returns

Chip ID as read with begin()

Definition at line 93 of file ErriezBMX280.cpp.

4.1.3.3 read16()

```
uint16_t ErriezBMX280::read16 (
            uint8_t reg )
```

Read from 16-bit register.

Parameters

reg	Register address

Returns

16-bit register value

Definition at line 357 of file ErriezBMX280.cpp.

4.1.3.4 read16_LE()

```
uint16_t ErriezBMX280::read16_LE (
            uint8_t reg )
```

Read from 16-bit unsigned register little endian.

Parameters

reg	Register address

Returns

16-bit unsigned register value in little endian

Definition at line 329 of file ErriezBMX280.cpp.

4.1.3.5 read24()

```
uint32_t ErriezBMX280::read24 (
            uint8_t reg )
```

Read from 24-bit register.

Parameters

reg	Register address

Returns

24-bit register value

Definition at line 375 of file ErriezBMX280.cpp.

4.1.3.6 read8()

```
uint8_t ErriezBMX280::read8 (
            uint8_t reg )
```

Read from 8-bit register.

Parameters

reg	Register address

Returns

8-bit register value

Definition at line 296 of file ErriezBMX280.cpp.

4.1.3.7 readAltitude()

```
float ErriezBMX280::readAltitude (
            float seaLevel )
```

Read approximate altitude.

Parameters

seaLevel	Sea level in hPa

Returns

Altitude (float)

Definition at line 174 of file ErriezBMX280.cpp.

4.1.3.8 readHumidity()

```
float ErriezBMX280::readHumidity ( )
```

Read humidity (BME280 only)

Returns

Humidity (float)

Definition at line 187 of file ErriezBMX280.cpp.

4.1.3.9 readPressure()

```
float ErriezBMX280::readPressure ( )
```

Read pressure.

Returns

Pressure (float)

Definition at line 132 of file ErriezBMX280.cpp.

4.1.3.10 readS16_LE()

```
int16_t ErriezBMX280::readS16_LE (
            uint8_t reg )
```

Read from 16-bit signed register little endian.

Parameters

reg	Register address

Returns

16-bit signed register value in little endian

Definition at line 345 of file ErriezBMX280.cpp.

4.1.3.11 readTemperature()

```
float ErriezBMX280::readTemperature ( )
```

Read temperature.

Returns

Temperature (float)

Definition at line 104 of file ErriezBMX280.cpp.

4.1.3.12 setSampling()

```
void ErriezBMX280::setSampling (
            BMX280_Mode_e mode = BMX280_MODE_NORMAL,
            BMX280_Sampling_e tempSampling = BMX280_SAMPLING_X16,
            BMX280_Sampling_e pressSampling = BMX280_SAMPLING_X16,
            BMX280_Sampling_e humSampling = BMX280_SAMPLING_X16,
            BMX280_Filter_e filter = BMX280_FILTER_OFF,
            BMX280_Standby_e standbyDuration = BMX280_STANDBY_MS_0_5 )
```

Set sampling registers.

Parameters

mode	See BMX280_Mode_e
tempSampling	See BMX280_Sampling_e
pressSampling	See BMX280_Sampling_e
humSampling	See BMX280_Sampling_e
filter	See BMX280_Filter_e
standbyDuration	See BMX280_Standby_e

Definition at line 269 of file ErriezBMX280.cpp.

4.1.3.13 write8()

```
void ErriezBMX280::write8 (
            uint8_t reg,
            uint8_t value )
```

Write to 8-bit register.

Parameters

reg	Register address
value	8-bit register value

Definition at line 314 of file ErriezBMX280.cpp.

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

- src/ErriezBMX280.h
- src/ErriezBMX280.cpp

Chapter 5

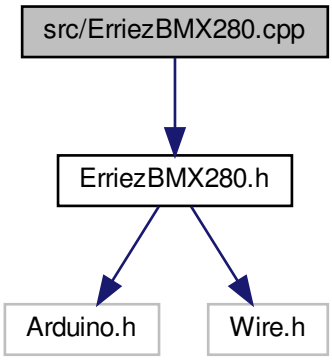
File Documentation

5.1 src/ErriezBMX280.cpp File Reference

BMP280/BME280 sensor library for Arduino.

```
#include "ErriezBMX280.h"
```

Include dependency graph for ErriezBMX280.cpp:



5.1.1 Detailed Description

BMP280/BME280 sensor library for Arduino.

BMP280 supports temperature and pressure BME280 supports temperature, pressure and humidity

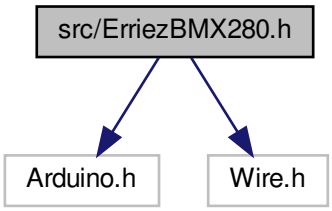
```
Source:          https://github.com/Erriez/ErriezBMX280
Documentation:   https://erriez.github.io/ErriezBMX280
```

5.2 src/ErriezBMX280.h File Reference

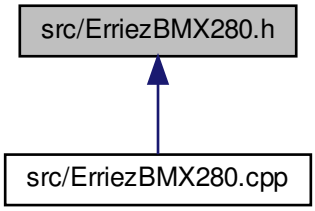
BMP280/BME280 sensor library for Arduino.

```
#include <Arduino.h>
#include <Wire.h>
```

Include dependency graph for ErriezBMX280.h:



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



Classes

- class ErriezBMX280

  BMX280 class.

Macros

- #define BMX280_I2C_ADDR 0x76

  I2C address.
- #define BMX280_I2C_ADDR_ALT 0x77

  I2C alternative address.
- #define BMX280_REG_DIG_T1 0x88

  Temperature coefficient register.

- #define BMX280_REG_DIG_T2 0x8A
  Temperature coefficient register.
- #define BMX280_REG_DIG_T3 0x8C
  Temperature coefficient register.
- #define BMX280_REG_DIG_P1 0x8E
  Pressure coefficient register.
- #define BMX280_REG_DIG_P2 0x90
  Pressure coefficient register.
- #define BMX280_REG_DIG_P3 0x92
  Pressure coefficient register.
- #define BMX280_REG_DIG_P4 0x94
  Pressure coefficient register.
- #define BMX280_REG_DIG_P5 0x96
  Pressure coefficient register.
- #define BMX280_REG_DIG_P6 0x98
  Pressure coefficient register.
- #define BMX280_REG_DIG_P7 0x9A
  Pressure coefficient register.
- #define BMX280_REG_DIG_P8 0x9C
  Pressure coefficient register.
- #define BMX280_REG_DIG_P9 0x9E
  Pressure coefficient register.
- #define BME280_REG_DIG_H1 0xA1
  Humidity coefficient register.
- #define BME280_REG_DIG_H2 0xE1
  Humidity coefficient register.
- #define BME280_REG_DIG_H3 0xE3
  Humidity coefficient register.
- #define BME280_REG_DIG_H4 0xE4
  Humidity coefficient register.
- #define BME280_REG_DIG_H5 0xE5
  Humidity coefficient register.
- #define BME280_REG_DIG_H6 0xE7
  Humidity coefficient register.
- #define BME280_REG_CHIPID 0xD0
  Chip ID register.
- #define BME280_REG_RESET 0xE0
  Reset register.
- #define BME280_REG_CTRL_HUM 0xF2
  BME280: Control humidity register.
- #define BMX280_REG_STATUS 0XF3
  Status register.
- #define BMX280_REG_CTRL_MEAS 0xF4
  Control measure register.
- #define BMX280_REG_CONFIG 0xF5
  Config register.
- #define BMX280_REG_PRESS 0xF7
  Pressure data register.
- #define BMX280_REG_TEMP 0xFA
  Temperature data register.
- #define BME280_REG_HUM 0xFD

Humidity data register.

- #define CHIP_ID_BMP280 0x58

  BMP280 chip ID.
- #define CHIP_ID_BME280 0x60

  BME280 chip ID.
- #define RESET_KEY 0xB6

  Reset value for reset register.
- #define STATUS_IM_UPDATE 0

  im_update bit in status register

Enumerations

- enum BMX280_Mode_e { BMX280_MODE_SLEEP = 0b00, BMX280_MODE_FORCED = 0b01, BMX280_MODE_NORMAL = 0b11 }

  Sleep mode bits ctrl_meas register.
- enum BMX280_Sampling_e { BMX280_SAMPLING_NONE = 0b000, BMX280_SAMPLING_X1 = 0b001, BMX280_SAMPLING_X2 = 0b010, BMX280_SAMPLING_X4 = 0b011, BMX280_SAMPLING_X8 = 0b100, BMX280_SAMPLING_X16 = 0b101 }

  Sampling bits registers ctrl_hum, ctrl_meas.
- enum BMX280_Filter_e { BMX280_FILTER_OFF = 0b000, BMX280_FILTER_X2 = 0b001, BMX280_FILTER_X4 = 0b010, BMX280_FILTER_X8 = 0b011, BMX280_FILTER_X16 = 0b100 }

  Filter bits config register.
- enum BMX280_Standby_e { BMX280_STANDBY_MS_0_5 = 0b000, BMX280_STANDBY_MS_10 = 0b110, BMX280_STANDBY_MS_20 = 0b111, BMX280_STANDBY_MS_62_5 = 0b001, BMX280_STANDBY_MS_125 = 0b010, BMX280_STANDBY_MS_250 = 0b011, BMX280_STANDBY_MS_500 = 0b100, BMX280_STANDBY_MS_1000 = 0b101 }

  Standby duration bits config register.

5.2.1 Detailed Description

BMP280/BME280 sensor library for Arduino.

BMP280 supports temperature and pressure BME280 supports temperature, pressure and humidity

```
Source:          https://github.com/Erriez/ErriezBMX280
Documentation:   https://erriez.github.io/ErriezBMX280
```

5.2.2 Enumeration Type Documentation

5.2.2.1 BMX280_Filter_e

```
enum BMX280_Filter_e
```

Filter bits config register.

Enumerator

BMX280_FILTER_OFF	Filter off.
BMX280_FILTER_X2	x2 Filter
BMX280_FILTER_X4	x4 Filter
BMX280_FILTER_X8	x8 Filter
BMX280_FILTER_X16	x16 Filter

Definition at line 109 of file ErriezBMX280.h.

5.2.2.2 BMX280_Mode_e

```
enum BMX280_Mode_e
```

Sleep mode bits ctrl_meas register.

Enumerator

BMX280_MODE_SLEEP	Sleep mode.
BMX280_MODE_FORCED	Forced mode.
BMX280_MODE_NORMAL	Normal mode.

Definition at line 88 of file ErriezBMX280.h.

5.2.2.3 BMX280_Sampling_e

```
enum BMX280_Sampling_e
```

Sampling bits registers ctrl_hum, ctrl_meas.

Enumerator

BMX280_SAMPLING_NONE	Sampling disabled.
BMX280_SAMPLING_X1	x1 Sampling
BMX280_SAMPLING_X2	x2 Sampling
BMX280_SAMPLING_X4	x4 Sampling
BMX280_SAMPLING_X8	x8 Sampling
BMX280_SAMPLING_X16	x16 Sampling

Definition at line 97 of file ErriezBMX280.h.

5.2.2.4 BMX280_Standby_e

```
enum BMX280_Standby_e
```

Standby duration bits config register.

Enumerator

BMX280_STANDBY_MS_0_5	0.5m standby
BMX280_STANDBY_MS_10	10ms standby
BMX280_STANDBY_MS_20	20ms standby
BMX280_STANDBY_MS_62↩ _5	62.5 standby
BMX280_STANDBY_MS_125	125ms standby
BMX280_STANDBY_MS_250	250ms standby
BMX280_STANDBY_MS_500	500ms standby
BMX280_STANDBY_MS_1000	1s standby

Definition at line 120 of file ErriezBMX280.h.

Index