DY271-China

0.1

Generated by Doxygen 1.8.11

Contents

Chapter 1

New Class for the GY271-Fakesensor

When you have a GY271-Magnetometer with the I2C-address 0x0D, you can use this libary. You can get the magneticfield in diffrent Formats, like (floats)Gaus, µT, mT and (int)RAW

Simple usage:

```
#include "MechaQMC5883.h"
#include <Wire.h>

//Objects must be declared before Setup
MechaQMC5883 qmc(0x0D);

void setup() {
  Wire.begin();                                          //Is needed for the Sensor
  Serial.begin(9600);                                    //Is needed for print_xx-methods

  qmc.init();                                            //Init with Defaultmode
  qmc.setMode(Mode_Continuous,ODR_10Hz,RNG_2G,
      OSR_512);   //Change the Sensormode
  qmc.setCalibValues(1.377,1.314,1.47);                  //Setup the CalibrationValues if needed in order
       of (x,y,z)
                                                         //To get these Values use the "Calibration.ino"
}

void loop() {

  qmc.print_uT();                        //Print the Magneticfield in µT
  qmc.print_T();                         //Print the Temperature in °C

}
```

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 files with brief descriptions:

Chapter 4

Class Documentation

4.1 FLOATDATA Struct Reference

`#include <MechaQMC5883.h>`

Collaboration diagram for FLOATDATA:

Public Attributes

- float x
- float y
- float z

4.1.1 Detailed Description

Datacontainer for xyz Floats (like Gaus, µT, mT)

Author

Philipp Heise

Date

02.06.2017

4.1.2 Member Data Documentation

4.1.2.1 float FLOATDATA::x

4.1.2.2 float FLOATDATA::y

4.1.2.3 float FLOATDATA::z

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

- C:/Users/Privat/Desktop/GY271-Magnetometer-China/Full-Demo/MechaQMC5883.h

4.2 MechaQMC5883 Class Reference

Use the GY271-QMC5883-Magnetometer.

`#include <MechaQMC5883.h>`

Collaboration diagram for MechaQMC5883:

Public Member Functions

- MechaQMC5883 ()
- MechaQMC5883 (uint8_t addr)
- void init ()
- void setMode (uint8_t mode, uint8_t odr, uint8_t rng, uint8_t osr)
- void setCalibValues (float x, float y, float z)
- RAWDATA getRaw ()
- FLOATDATA get_Gaus ()
- FLOATDATA get_mT ()
- FLOATDATA get_uT ()
- FLOATDATA calc_Gaus (RAWDATA xyzRAW)
- FLOATDATA calc_mT (RAWDATA xyzRAW)
- FLOATDATA calc_uT (RAWDATA xyzRAW)
- void print_Raw ()
- void print_Gaus ()
- void print_mT ()
- void print_uT ()
- void print_Gaus (int dez)
- void print_mT (int dez)
- void print_uT (int dez)
- int getRawT ()
- float get_T ()
- float calc_T (int RAWT)
- void print_RawT ()
- void print_T ()
- void print_T (int dez)
- void softReset ()

4.2.1 Detailed Description

Use the GY271-QMC5883-Magnetometer.

The class for the GY271-Fake

Author

Philipp Heise

Date

02.06.2017

If the class not working, checkout your wireing. Use diffrent methods to get the Values which you want.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 MechaQMC5883::MechaQMC5883 ( )

Here is the call graph for this function:

4.2.2.2 MechaQMC5883::MechaQMC5883 (uint8_t *addr*)

If needed you can use a other Sensor-Address

Here is the call graph for this function:

4.2.3 Member Function Documentation

4.2.3.1 FLOATDATA MechaQMC5883::calc_Gaus (RAWDATA *xyzRAW*)

Can use to calculate the Gaus-Value from spezified RAWDATA-Input

Parameters

in	*RAWDATA*	The RAWDATA which you want to calculate.

Returns

FLOATDATA as Gaus

Here is the caller graph for this function:

4.2.3.2 FLOATDATA MechaQMC5883::calc_mT (RAWDATA *xyzRAW*)

Can use to calculate the mT-Value from spezified RAWDATA-Input

Parameters

in	*RAWDATA*	The RAWDATA which you want to calculate.

Returns

FLOATDATA as mT

Here is the caller graph for this function:

4.2.3.3 float MechaQMC5883::calc_T (int *RAWT*)

Calculate ℃ from RAW-Input

Parameters

in	*RAWT*	The value that you want to cast

Returns

Temperature as Float

Here is the caller graph for this function:

4.2.3.4 FLOATDATA MechaQMC5883::calc_uT (RAWDATA *xyzRAW*)

Can use to calculate the uT-Value from spezified RAWDATA-Input

Parameters

in	*RAWDATA*	The RAWDATA which you want to calculate.

Returns

FLOATDATA as uT

Here is the caller graph for this function:

4.2.3.5 FLOATDATA MechaQMC5883::get_Gaus ( )

Reads the Sensordata and calculate the Gaus-Value

Returns

FLOATDATA as Gaus

Here is the call graph for this function:

Here is the caller graph for this function:

4.2.3.6 FLOATDATA MechaQMC5883::get_mT ( )

Reads the Sensordata and calculate the mT-Value

Returns

FLOATDATA as mT

Here is the call graph for this function:

Here is the caller graph for this function:

4.2.3.7 float MechaQMC5883::get_T ( )

Reads the Temperature in RAW and calculate ℃

Returns

Temperature as Float

Here is the call graph for this function:

Here is the caller graph for this function:

4.2.3.8 FLOATDATA MechaQMC5883::get_uT ( )

Reads the Sensordata and calculate the µT-Value

Returns

FLOATDATA as µT

Here is the call graph for this function:

Here is the caller graph for this function:

4.2.3.9 RAWDATA MechaQMC5883::getRaw ( )

Reads the Sensordata

Returns

RAWDATA from the Sensor as an Integer

Here is the caller graph for this function:

4.2.3.10 int MechaQMC5883::getRawT ( )

Reads the Temperature in RAW from the Sensor

Returns

temperature as Integer

Here is the caller graph for this function:

4.2.3.11 void MechaQMC5883::init ( )

Setup the Sensor in Continuous-Mode and 200Hz Samplerate by Fullscale-Range (+-8 Gaus)

Here is the call graph for this function:

4.2.3.12 void MechaQMC5883::print_Gaus ( )

Print the Gaus-Data like "|x=...|y=...|z=...|Gaus\r\n"

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.13 void MechaQMC5883::print_Gaus (int *dez*)

Print the Gaus-Data like "|x=...|y=...|z=...|Gaus\r\n" with variable precision

See also

print_Gaus()

Parameters

in	*dez*	Percision

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.14 void MechaQMC5883::print_mT ( )

Print the mT-Data like "|x=...|y=...|z=...|mT\r\n"

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.15 void MechaQMC5883::print_mT (int *dez*)

Print the mT-Data like "|x=...|y=...|z=...|mT\r\n" with variable precision

See also

print_mT()

Parameters

in	*dez*	Percision

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.16 void MechaQMC5883::print_Raw ( )

Print the RAW-Data like "|x=...|y=...|z=...|RAW\r\n"

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.17 void MechaQMC5883::print_RawT ( )

Print the Temperature like "|Temp=...|RAW\r\n"

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.18 void MechaQMC5883::print_T ( )

Print the Temperature like "|Temp=...|C\r\n"

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.19 void MechaQMC5883::print_T (int *dez*)

Print the Gaus-Data like "|Temp=...|C\r\n" with variable precision

See also

print_T()

Parameters

in	*dez*	Percision

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.20 void MechaQMC5883::print_uT ( )

Print the uT-Data like "|x=...|y=...|z=...|uT\r\n"

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.21 void MechaQMC5883::print_uT (int *dez*)

Print the uT-Data like "|x=...|y=...|z=...|uT\r\n" with variable precision

See also

print_mT()

Parameters

in	*dez*	Percision

Warning

You must use Serial.begin() in Setup-Loop

Here is the call graph for this function:

4.2.3.22 void MechaQMC5883::setCalibValues (float *x*, float *y*, float *z*)

Calibrate the Sensordata

Warning

Factors are not used with RAWDATA

Parameters

in	*x*	Is the factor which is multiplied with the Original-x-Value
in	*y*	Is the factor which is multiplied with the Original-y-Value
in	*z*	Is the factor which is multiplied with the Original-z-Value

Here is the caller graph for this function:

4.2.3.23 void MechaQMC5883::setMode (uint8_t *mode*, uint8_t *odr*, uint8_t *rng*, uint8_t *osr*)

Change the Sensormode

Parameters

in	*mode*	Valid Values are: Mode_Standby,Mode_Continuous
in	*odr*	Valid Values are: ODR_10Hz,ODR_50Hz,ODR_100Hz,ODR_200Hz
in	*rng*	Valid Values are: RNG_2G,RNG_8G
in	*osr*	Valid Values are: OSR_512,OSR_256,OSR_128,OSR_64

Here is the caller graph for this function:

4.2.3.24 void MechaQMC5883::softReset ( )

Perform a Softreset on the Sensor

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

- C:/Users/Privat/Desktop/GY271-Magnetometer-China/Full-Demo/MechaQMC5883.h
- C:/Users/Privat/Desktop/GY271-Magnetometer-China/Full-Demo/MechaQMC5883.cpp

4.3 RAWDATA Struct Reference

`#include <MechaQMC5883.h>`

Collaboration diagram for RAWDATA:

Public Attributes

- int x
- int y
- int z

4.3.1 Detailed Description

Datacontainer for xyz Integers

Author

Philipp Heise

Date

02.06.2017

4.3.2 Member Data Documentation

4.3.2.1 int RAWDATA::x

4.3.2.2 int RAWDATA::y

4.3.2.3 int RAWDATA::z

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

- C:/Users/Privat/Desktop/GY271-Magnetometer-China/Full-Demo/MechaQMC5883.h

Chapter 5

File Documentation

5.1 C:/Users/Privat/Desktop/GY271-Magnetometer-China/Full-Demo/MechaQMC5883.cpp File Reference

```
#include "MechaQMC5883.h"
#include <Wire.h>
```

Include dependency graph for MechaQMC5883.cpp:

5.2 C:/Users/Privat/Desktop/GY271-Magnetometer-China/Full-Demo/MechaQMC5883.h File Reference

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

Include dependency graph for MechaQMC5883.h: This graph shows which files directly or indirectly include this file:

Classes

- struct RAWDATA
- struct FLOATDATA
- class MechaQMC5883

  Use the GY271-QMC5883-Magnetometer.

Macros

- #define QMC5883_ADDR 0x0D
- #define Mode_Standby 0b00000000
- #define Mode_Continuous 0b00000001
- #define ODR_10Hz 0b00000000
- #define ODR_50Hz 0b00000100
- #define ODR_100Hz 0b00001000
- #define ODR_200Hz 0b00001100
- #define RNG_2G 0b00000000
- #define RNG_8G 0b00010000
- #define OSR_512 0b00000000
- #define OSR_256 0b01000000
- #define OSR_128 0b10000000
- #define OSR_64 0b11000000

5.2.1 Macro Definition Documentation

5.2.1.1 #define Mode_Continuous 0b00000001

5.2.1.2 #define Mode_Standby 0b00000000

5.2.1.3 #define ODR_100Hz 0b00001000

5.2.1.4 #define ODR_10Hz 0b00000000

5.2.1.5 #define ODR_200Hz 0b00001100

5.2.1.6 #define ODR_50Hz 0b00000100

5.2.1.7 #define OSR_128 0b10000000

5.2.1.8 #define OSR_256 0b01000000

5.2.1.9 #define OSR_512 0b00000000

5.2.1.10 #define OSR_64 0b11000000

5.2.1.11 #define QMC5883_ADDR 0x0D

5.2.1.12 #define RNG_2G 0b00000000

5.2.1.13 #define RNG_8G 0b00010000

Index