Infineon OnGoing Calibration Library

1.0.1

Generated by Doxygen 1.8.14

Contents

1 Infineon OnGoing Calibration Library **1**

1.1 Introduction . . . 1

1.2 Usage . . . 2

1.3 Example . . . 3

1.4 Other considerations . . . 3

2 Data Structure Index **5**

2.1 Data Structures . . . 5

3 File Index **7**

3.1 File List . . . 7

4 Data Structure Documentation **9**

4.1 ONGO_ANALOG_SENSOR_DATA Struct Reference . . . 9

4.1.1 Detailed Description . . . 9

4.1.2 Field Documentation . . . 9

4.1.2.1 cosN_LSB . . . 9

4.1.2.2 cosP_LSB . . . 10

4.1.2.3 diff_X . . . 10

4.1.2.4 diff_Y . . . 10

4.1.2.5 sinN_LSB . . . 10

4.1.2.6 sinP_LSB . . . 10

4.1.2.7 uncalibratedAngle . . . 11

4.2 ONGO_CALIB_DATA Struct Reference . . . 11

4.2.1 Detailed Description . . . 11

4.2.2 Field Documentation . . . 11
4.2.2.1 cos_ortho . . . 12
4.2.2.2 rotation_data . . . 12
4.2.2.3 sin_ortho . . . 12
4.3 ONGO_ROTATION_DATA_STORE Struct Reference . . . 12
4.3.1 Detailed Description . . . 13
4.3.2 Field Documentation . . . 13
4.3.2.1 amplitudeX . . . 13
4.3.2.2 amplitudeY . . . 13
4.3.2.3 angle135found . . . 13
4.3.2.4 angle45found . . . 13
4.3.2.5 magnitude135 . . . 14
4.3.2.6 magnitude45 . . . 14
4.3.2.7 maxCos . . . 14
4.3.2.8 maxSin . . . 14
4.3.2.9 minCos . . . 14
4.3.2.10 minSin . . . 15
4.3.2.11 nr_valid_rotations . . . 15
4.3.2.12 offsetX . . . 15
4.3.2.13 offsetY . . . 15
4.3.2.14 ortho_one_dir . . . 15
4.3.2.15 X135 . . . 16
4.3.2.16 X135_corr . . . 16
4.3.2.17 X45 . . . 16
4.3.2.18 X45_corr . . . 16
4.3.2.19 Y135 . . . 16
4.3.2.20 Y135_corr . . . 17
4.3.2.21 Y45 . . . 17
4.3.2.22 Y45_corr . . . 17

5 File Documentation **19**

5.1 ongocalib.c File Reference . . . 19

5.1.1 Function Documentation . . . 20

5.1.1.1 ONGO_CalibrationFindParam() . . . 20

5.1.1.2 ONGO_GetCalibAngle() . . . 20

5.1.1.3 ONGO_GetUncalibAngle() . . . 21

5.1.1.4 ONGO_InitCalibData() . . . 21

5.1.1.5 ONGO_InitCalibData2() . . . 22

5.1.1.6 ONGO_InitSensorData() . . . 23

5.2 ongocalib.h File Reference . . . 23

5.2.1 Macro Definition Documentation . . . 25

5.2.1.1 ANGLE135 . . . 25

5.2.1.2 ANGLE135_H . . . 26

5.2.1.3 ANGLE135_L . . . 26

5.2.1.4 ANGLE180_RAD . . . 26

5.2.1.5 ANGLE225 . . . 26

5.2.1.6 ANGLE225_H . . . 26

5.2.1.7 ANGLE225_L . . . 27

5.2.1.8 ANGLE315 . . . 27

5.2.1.9 ANGLE315_H . . . 27

5.2.1.10 ANGLE315_L . . . 27

5.2.1.11 ANGLE45 . . . 27

5.2.1.12 ANGLE45_H . . . 28

5.2.1.13 ANGLE45_L . . . 28

5.2.1.14 ANGLE_TOLERANCE . . . 28

5.2.1.15 ANGLE_TOLERANCE_RAD . . . 28

5.2.1.16 DEG2RADFACTOR . . . 28

5.2.1.17 RAD2DEGFACTOR . . . 29

5.2.1.18 ROTATION_VALID . . . 29

5.2.2 Typedef Documentation . . . 29

5.2.2.1 float64_t . . . 29

5.2.2.2 ONGO_ANALOG_SENSOR_DATA_t . . . 29

5.2.2.3 ONGO_CALIB_DATA_t . . . 29

5.2.2.4 ONGO_ROTATION_DATA_STORE_t . . . 29

5.2.3 Function Documentation . . . 29

5.2.3.1 ONGO_CalibrationFindParam() . . . 29

5.2.3.2 ONGO_GetCalibAngle() . . . 30

5.2.3.3 ONGO_GetUncalibAngle() . . . 30

5.2.3.4 ONGO_InitCalibData() . . . 31

5.2.3.5 ONGO_InitCalibData2() . . . 31

5.2.3.6 ONGO_InitSensorData() . . . 32

Index **35**

Chapter 1

Infineon OnGoing Calibration Library

Version

1.0.1

1.1 Introduction

This library is intended to help out hardware and software engineers, system integrators and developers with the process of calibrating analog angle sensors. An analog angle sensor does not provide any angle value, only analog sine and cosine (positive and negative) output signals. The angle has to be calculated externally.

Analog angle sensors have to be calibrated before they are used in order to achieve the specified angle accuracy. This calibration process includes a compensation of offset, amplitude and non-orthogonality of the output channels in use with the help of an external microcontroller.

Calibration is done in a continuous matter (dynamic calibration). Calibration parameters should be first initialized. New parameters will be found after 1-2 turns. The library is intended to be used in a one direction spin.

Copyright

Copyright (C) 2017 Infineon Technologies AG. All rights reserved.

Attention

Infineon Technologies AG (INFINEON) is supplying this file for use exclusively with Infineon's products. This file can be freely distributed within development tools and software supporting such microcontroller products. THIS SOFTWARE IS PROVIDED "AS IS". NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR S↩ TATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. INFINEON SHALL NOT, IN A↩ NY CIRCUMSTANCES, BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, ASPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

Warning

It is in the responsibility of the system integrator to insure that:

- the magnet is large enough to ensure that the non-homogeneity of the magnetic field in the sensing area is negligible.
- the mechanical tolerances of the sensor-magnet assembly are minimal.
- the output amplitude together with the resolution of the ADC in use matches the requirement in angle accuracy.
- the calibration parameters are initialized with correct values.
- when new calibration parameters have been found, the library will do mathematical calculations, which will take more time for one moment.

See also

`https://www.infineon.com/dgdl/Infineon-TLE5xxx(D)_Calibration_360_A↩N-v02_00-AN-v02_00-EN.pdf?fileId=5546d46264a8de7e0164f09d8bfa228d`

1.2 Usage

Note

This library assumes that you have connected the analog angle sensor to an ADC and you can easily get the SINP, COSP, SINN, COSN digitized values.

The one time calibration process requires you to:

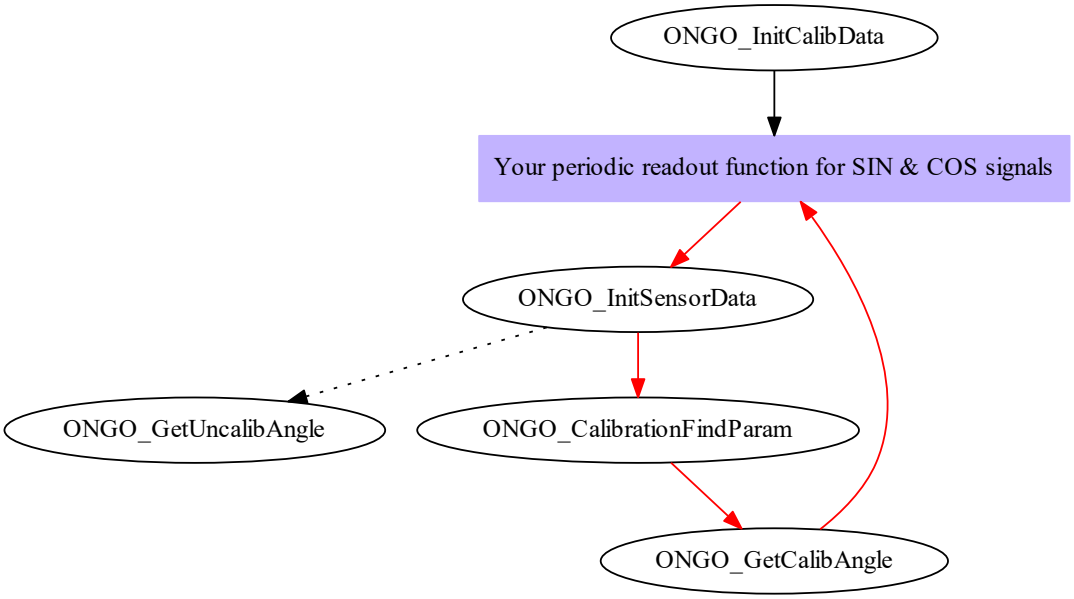
- turn the magnetic field in one direction while acquiring SIN & COS data
- afterward the firmware library will compute:
  - new amplitude & offset correction values

If your application is subjected to temperature variations you should perform dynamic/continuous calibration as the one time calibration can help reduce only the temperature independent offset.

Note

Because the calibration process is quite intensive computationally the system integrator should use either a microcontroller with an FPU or a dedicated math coprocessor/hardware CORDIC.

1.3 Example



Code Example 1:

```
ONGO_CALIB_DATA_t calibration_store_params;

float64_t angle_read = 0;
uint32_t sinP_in_LSB = 0;
uint32_t cosP_in_LSB = 0;
uint32_t sinN_in_LSB = 0;
uint32_t cosN_in_LSB = 0;

int main(void)
{
    ONGO_InitCalibData(&calibration_store_params, max_diff_sin_LSB, max_diff_cos_LSB,
     min_diff_sin_LSB, min_diff_cos_LSB, sin_45, cos_45, sin_135, cos_135);

    while(1U)
    {
            // read sensor data
        // AnalogRead(ch) is a user defined function to read the analog voltage on P and N channels
        sinP_in_LSB = AnalogRead(ch0);
        cosP_in_LSB = AnalogRead(ch1);
        sinN_in_LSB = AnalogRead(ch2);
        cosN_in_LSB = AnalogRead(ch3);
        ONGO_InitSensorData(sinP_in_LSB, cosP_in_LSB, sinN_in_LSB, cosN_in_LSB);
        ONGO_CalibrationFindParam(&calibration_store_params);
        angle_read = ONGO_GetCalibAngle(&calibration_store_params);
    }
}
```

1.4 Other considerations

See also

https://www.infineon.com/dgdl/Infineon-TLE5xxx(D)_Calibration_360_A↩
N-v02_00-AN-v02_00-EN.pdf?fileId=5546d46264a8de7e0164f09d8bfa228d

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

ONGO_ANALOG_SENSOR_DATA 9
ONGO_CALIB_DATA 11
ONGO_ROTATION_DATA_STORE 12

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

ongocalib.c 19
ongocalib.h 23

Chapter 4

Data Structure Documentation

4.1 ONGO_ANALOG_SENSOR_DATA Struct Reference

`#include <ongocalib.h>`

Data Fields

- uint32_t **sinP_LSB**
- uint32_t **cosP_LSB**
- uint32_t **sinN_LSB**
- uint32_t **cosN_LSB**
- int32_t **diff_X**
- int32_t **diff_Y**
- **float64_t uncalibratedAngle**

4.1.1 Detailed Description

Definition at line 52 of file ongocalib.h.

4.1.2 Field Documentation

4.1.2.1 cosN_LSB

`uint32_t ONGO_ANALOG_SENSOR_DATA::cosN_LSB`

negative cosine signal.

Definition at line 56 of file ongocalib.h.

4.1.2.2 cosP_LSB

```
uint32_t ONGO_ANALOG_SENSOR_DATA::cosP_LSB
```

positive cosine signal.

Definition at line 54 of file ongocalib.h.

4.1.2.3 diff_X

```
int32_t ONGO_ANALOG_SENSOR_DATA::diff_X
```

differential value for x-channel (cosine).

Definition at line 58 of file ongocalib.h.

4.1.2.4 diff_Y

```
int32_t ONGO_ANALOG_SENSOR_DATA::diff_Y
```

differential value for y-channel (sine).

Definition at line 59 of file ongocalib.h.

4.1.2.5 sinN_LSB

```
uint32_t ONGO_ANALOG_SENSOR_DATA::sinN_LSB
```

negative sine signal.

Definition at line 55 of file ongocalib.h.

4.1.2.6 sinP_LSB

```
uint32_t ONGO_ANALOG_SENSOR_DATA::sinP_LSB
```

positive sine signal.

Definition at line 53 of file ongocalib.h.

4.1.2.7 uncalibratedAngle

```
float64_t ONGO_ANALOG_SENSOR_DATA::uncalibratedAngle
```

differential angle.

Definition at line 61 of file ongocalib.h.

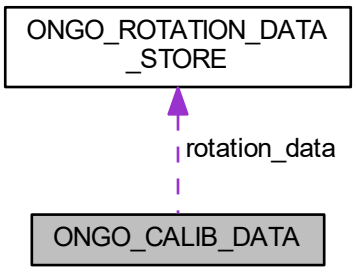
The documentation for this struct was generated from the following file:

- **ongocalib.h**

4.2 ONGO_CALIB_DATA Struct Reference

```
#include <ongocalib.h>
```

Collaboration diagram for ONGO_CALIB_DATA:



Data Fields

- **ONGO_ROTATION_DATA_STORE_t rotation_data**
- **float64_t sin_ortho**
- **float64_t cos_ortho**

4.2.1 Detailed Description

Definition at line 102 of file ongocalib.h.

4.2.2 Field Documentation

4.2.2.1 cos_ortho

float64_t ONGO_CALIB_DATA::cos_ortho

mean channel correction of orthogonality error cos (fi).

Definition at line 106 of file ongocalib.h.

4.2.2.2 rotation_data

ONGO_ROTATION_DATA_STORE_t ONGO_CALIB_DATA::rotation_data

Definition at line 103 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.2.2.3 sin_ortho

float64_t ONGO_CALIB_DATA::sin_ortho

mean channel correction of orthogonality error sin (fi).

Definition at line 105 of file ongocalib.h.

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

- **ongocalib.h**

4.3 ONGO_ROTATION_DATA_STORE Struct Reference

#include <ongocalib.h>

Data Fields

- bool **angle45found**
- bool **angle135found**
- uint32_t **nr_valid_rotations**
- int32_t **maxCos**
- int32_t **minCos**
- int32_t **maxSin**
- int32_t **minSin**
- **float64_t X45**
- **float64_t X135**
- **float64_t Y45**
- **float64_t Y135**
- **float64_t X45_corr**
- **float64_t X135_corr**
- **float64_t Y45_corr**
- **float64_t Y135_corr**
- **float64_t amplitudeX**
- **float64_t amplitudeY**
- **float64_t offsetX**
- **float64_t offsetY**
- **float64_t magnitude45**
- **float64_t magnitude135**
- **float64_t ortho_one_dir**

4.3.1 Detailed Description

Definition at line 65 of file ongocalib.h.

4.3.2 Field Documentation

4.3.2.1 amplitudeX

```
float64_t ONGO_ROTATION_DATA_STORE::amplitudeX
```

amplitude of COS signal.

Definition at line 89 of file ongocalib.h.

4.3.2.2 amplitudeY

```
float64_t ONGO_ROTATION_DATA_STORE::amplitudeY
```

amplitude of SIN signal.

Definition at line 90 of file ongocalib.h.

4.3.2.3 angle135found

```
bool ONGO_ROTATION_DATA_STORE::angle135found
```

Definition at line 67 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.4 angle45found

```
bool ONGO_ROTATION_DATA_STORE::angle45found
```

Definition at line 66 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.5 magnitude135

```
float64_t ONGO_ROTATION_DATA_STORE::magnitude135
```

M135 value of signal.

Definition at line 96 of file ongocalib.h.

4.3.2.6 magnitude45

```
float64_t ONGO_ROTATION_DATA_STORE::magnitude45
```

M45 value of signal.

Definition at line 95 of file ongocalib.h.

4.3.2.7 maxCos

```
int32_t ONGO_ROTATION_DATA_STORE::maxCos
```

max of COS differential signal.

Definition at line 72 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.8 maxSin

```
int32_t ONGO_ROTATION_DATA_STORE::maxSin
```

max of SIN differential signal.

Definition at line 74 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.9 minCos

```
int32_t ONGO_ROTATION_DATA_STORE::minCos
```

min of COS differential signal.

Definition at line 73 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.10 minSin

```
int32_t ONGO_ROTATION_DATA_STORE::minSin
```

min of SIN differential signal.

Definition at line 75 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.11 nr_valid_rotations

```
uint32_t ONGO_ROTATION_DATA_STORE::nr_valid_rotations
```

Definition at line 68 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.12 offsetX

```
float64_t ONGO_ROTATION_DATA_STORE::offsetX
```

offset of COSP signal.

Definition at line 92 of file ongocalib.h.

4.3.2.13 offsetY

```
float64_t ONGO_ROTATION_DATA_STORE::offsetY
```

offset of COSN signal.

Definition at line 93 of file ongocalib.h.

4.3.2.14 ortho_one_dir

```
float64_t ONGO_ROTATION_DATA_STORE::ortho_one_dir
```

orthogonality error of signal.

Definition at line 98 of file ongocalib.h.

4.3.2.15 X135

float64_t ONGO_ROTATION_DATA_STORE::X135

COS value at 135 deg.

Definition at line 78 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.16 X135_corr

float64_t ONGO_ROTATION_DATA_STORE::X135_corr

COS amplitude and offset corrected value at 135 deg.

Definition at line 84 of file ongocalib.h.

4.3.2.17 X45

float64_t ONGO_ROTATION_DATA_STORE::X45

COS value at 45 deg.

Definition at line 77 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.18 X45_corr

float64_t ONGO_ROTATION_DATA_STORE::X45_corr

COS amplitude and offset corrected value at 45 deg.

Definition at line 83 of file ongocalib.h.

4.3.2.19 Y135

float64_t ONGO_ROTATION_DATA_STORE::Y135

SIN value at 135 deg.

Definition at line 81 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.20 Y135_corr

```
float64_t ONGO_ROTATION_DATA_STORE::Y135_corr
```

SIN amplitude and offset corrected value at 135 deg.

Definition at line 87 of file ongocalib.h.

4.3.2.21 Y45

```
float64_t ONGO_ROTATION_DATA_STORE::Y45
```

SIN value at 45 deg.

Definition at line 80 of file ongocalib.h.

Referenced by ONGO_InitCalibData(), and ONGO_InitCalibData2().

4.3.2.22 Y45_corr

```
float64_t ONGO_ROTATION_DATA_STORE::Y45_corr
```

SIN amplitude and offset corrected value at 45 deg.

Definition at line 86 of file ongocalib.h.

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

- **ongocalib.h**

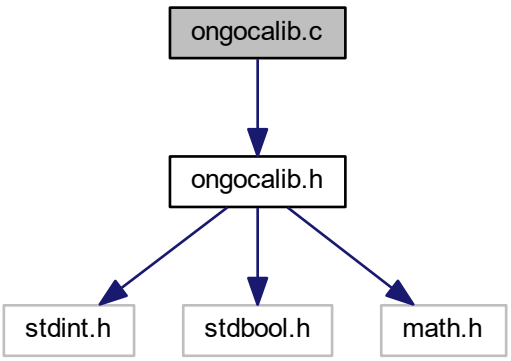
Chapter 5

File Documentation

5.1 ongocalib.c File Reference

```
#include "ongocalib.h"
```

Include dependency graph for ongocalib.c:



Functions

- void **ONGO_InitSensorData** (uint32_t sinP_in_LSB, uint32_t cosP_in_LSB, uint32_t sinN_in_LSB, uint32_t cosN_in_LSB)

  Function used for ONGO_ANALOG_SENSOR_DATA_t data assignment.
- void **ONGO_InitCalibData** (**ONGO_CALIB_DATA_t** *calib_param, int32_t max_diff_sin_LSB, int32_t max_diff_cos_LSB, int32_t min_diff_sin_LSB, int32_t min_diff_cos_LSB, **float64_t** sin_45, **float64_t** cos_45, **float64_t** sin_135, **float64_t** cos_135)

  Function used for ONGO_CALIB_DATA_t data initialization. The pointer of type ONGO_CALIB_DATA_t will be used to store calibration parameters.
- void **ONGO_InitCalibData2** (**ONGO_CALIB_DATA_t** *calib_param, int32_t max_SinP_LSB, int32_t max_CosP_LSB, int32_t max_SinN_LSB, int32_t max_CosN_LSB, int32_t min_SinP_LSB, int32_t min_CosP_LSB, int32_t min_SinN_LSB, int32_t min_CosN_LSB, **float64_t** sin_45, **float64_t** cos_45, **float64_t** sin_135, **float64_t** cos_135)

Function used for ONGO_CALIB_DATA_t data initialization. The pointer of type ONGO_CALIB_DATA_t will be used to store calibration parameters.

- **float64_t ONGO_GetUncalibAngle** (void)

  Calculates the uncalibrated angle.
- **float64_t ONGO_GetCalibAngle** (const **ONGO_CALIB_DATA_t** ∗calib_param)

  Calculates the calibrated angle.
- void **ONGO_CalibrationFindParam** (**ONGO_CALIB_DATA_t** ∗calib_param)

  This function is used to find new calibration parameters.

5.1.1 Function Documentation

5.1.1.1 ONGO_CalibrationFindParam()

```
void ONGO_CalibrationFindParam (
              ONGO_CALIB_DATA_t * calib_param )
```

This function is used to find new calibration parameters.

When new calibration parameters have been found, this function will do mathematical calculations, which will take more time for one moment. This function will extract the maximum, minimum voltage levels, amplitude, offset, and orthogonality. Based on the found values, it will calculate the necessary calibration parameters.

Parameters

in	*calib_param*	is a pointer of type ONGO_CALIB_DATA_t, this structure holds the current sensor calibration parameters.

Returns

Nothing

Definition at line 353 of file ongocalib.c.

5.1.1.2 ONGO_GetCalibAngle()

```
float64_t ONGO_GetCalibAngle (
            const  ONGO_CALIB_DATA_t * calib_param )
```

Calculates the calibrated angle.

Parameters

in	*calib_param*	is a pointer of type ONGO_CALIB_DATA_t, this structure holds the current sensor calibration parameters.

Returns

Returns the calibrated angle in degrees.

Definition at line 337 of file ongocalib.c.

5.1.1.3 ONGO_GetUncalibAngle()

```
float64_t ONGO_GetUncalibAngle (
            void  )
```

Calculates the uncalibrated angle.

This function uses differential sensor data to extract the angle.

Returns

Returns the uncalibrated angle in degrees.

Definition at line 325 of file ongocalib.c.

5.1.1.4 ONGO_InitCalibData()

```
void ONGO_InitCalibData (
              ONGO_CALIB_DATA_t * calib_param,
             int32_t max_diff_sin_LSB,
             int32_t max_diff_cos_LSB,
             int32_t min_diff_sin_LSB,
             int32_t min_diff_cos_LSB,
              float64_t sin_45,
              float64_t cos_45,
              float64_t sin_135,
              float64_t cos_135 )
```

Function used for ONGO_CALIB_DATA_t data initialization. The pointer of type ONGO_CALIB_DATA_t will be used to store calibration parameters.

This function must be called once before ONGO_CalibrationFindParam(..). Use this or ONGO_InitCalibData2(..).

Parameters

in	*calib_param*	is a pointer of type ONGO_CALIB_DATA_t.
in	*max_diff_sin_LSB*	is the maximum differential sin signal in LSB.
in	*max_diff_cos_LSB*	is the maximum differential cos signal in LSB.
in	*min_diff_sin_LSB*	is the minimum differential sin signal in LSB.
in	*min_diff_cos_LSB*	is the minimum differential cos signal in LSB.
in	*sin_45*	is the differential sin component in LSB at a 45 deg position.
in	*cos_45*	is the differential cos component in LSB at a 45 deg position.
in	*sin_135*	is the differential sin component in LSB at a 135 deg position.
in	*cos_135*	is the differential cos component in LSB at a 135 deg position.

Returns

Nothing

Definition at line 200 of file ongocalib.c.

References ONGO_ROTATION_DATA_STORE::angle135found, ONGO_ROTATION_DATA_STORE::angle45found, ONGO_ROTATION_DATA_STORE::maxCos, ONGO_ROTATION_DATA_STORE::maxSin, ONGO_ROTATIO↩N_DATA_STORE::minCos, ONGO_ROTATION_DATA_STORE::minSin, ONGO_ROTATION_DATA_STORE↩::nr_valid_rotations, ONGO_CALIB_DATA::rotation_data, ONGO_ROTATION_DATA_STORE::X135, ONGO_R↩OTATION_DATA_STORE::X45, ONGO_ROTATION_DATA_STORE::Y135, and ONGO_ROTATION_DATA_S↩TORE::Y45.

5.1.1.5 ONGO_InitCalibData2()

```
void ONGO_InitCalibData2 (
              ONGO_CALIB_DATA_t * calib_param,
             int32_t max_SinP_LSB,
             int32_t max_CosP_LSB,
             int32_t max_SinN_LSB,
             int32_t max_CosN_LSB,
             int32_t min_SinP_LSB,
             int32_t min_CosP_LSB,
             int32_t min_SinN_LSB,
             int32_t min_CosN_LSB,
              float64_t sin_45,
              float64_t cos_45,
              float64_t sin_135,
              float64_t cos_135 )
```

Function used for ONGO_CALIB_DATA_t data initialization. The pointer of type ONGO_CALIB_DATA_t will be used to store calibration parameters.

This function must be called once before ONGO_CalibrationFindParam(..). Use this or ONGO_InitCalibData(..).

Parameters

in	*calib_param*	is a pointer of type ONGO_CALIB_DATA_t.
in	*max_SinP_LSB*	is the maximum P channel sin signal in LSB.
in	*max_CosP_LSB*	is the maximum P channel cos signal in LSB.
in	*max_SinN_LSB*	is the maximum N channel sin signal in LSB.
in	*max_CosN_LSB*	is the maximum N channel cos signal in LSB.
in	*min_SinP_LSB*	is the minimum P channel sin signal in LSB.
in	*min_CosP_LSB*	is the minimum P channel cos signal in LSB.
in	*min_SinN_LSB*	is the minimum N channel sin signal in LSB.
in	*min_CosN_LSB*	is the minimum N channel cos signal in LSB.
in	*sin_45*	is the differential sin component in LSB at a 45 deg position.
in	*cos_45*	is the differential cos component in LSB at a 45 deg position.
in	*sin_135*	is the differential sin component in LSB at a 135 deg position.
in	*cos_135*	is the differential cos component in LSB at a 135 deg position.

Returns

Nothing

Definition at line 258 of file ongocalib.c.

References ONGO_ROTATION_DATA_STORE::angle135found, ONGO_ROTATION_DATA_STORE::angle45found, ONGO_ROTATION_DATA_STORE::maxCos, ONGO_ROTATION_DATA_STORE::maxSin, ONGO_ROTATION_DATA_STORE::minCos, ONGO_ROTATION_DATA_STORE::minSin, ONGO_ROTATION_DATA_STORE::nr_valid_rotations, ONGO_CALIB_DATA::rotation_data, ONGO_ROTATION_DATA_STORE::X135, ONGO_ROTATION_DATA_STORE::X45, ONGO_ROTATION_DATA_STORE::Y135, and ONGO_ROTATION_DATA_STORE::Y45.

5.1.1.6 ONGO_InitSensorData()

```
void ONGO_InitSensorData (
            uint32_t sinP_in_LSB,
            uint32_t cosP_in_LSB,
            uint32_t sinN_in_LSB,
            uint32_t cosN_in_LSB )
```

Function used for ONGO_ANALOG_SENSOR_DATA_t data assignment.

This function must be called after the data readout.

The initialized pointer will be used in the angle calculation and calibration process.

Parameters

in	*cosP_in_LSB*	is the sensor value in LSB corresponding to the positive cosinus.
in	*cosN_in_LSB*	is the sensor value in LSB corresponding to the negative cosinus.
in	*sinP_in_LSB*	is the sensor value in LSB corresponding to the positive sinus.
in	*sinN_in_LSB*	is the sensor value in LSB corresponding to the negative sinus.

Returns

Nothing

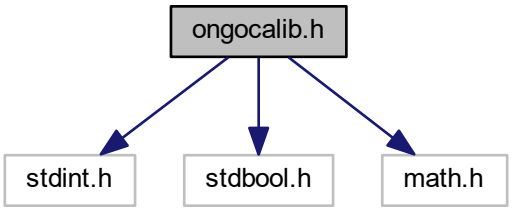
Definition at line 164 of file ongocalib.c.

5.2 ongocalib.h File Reference

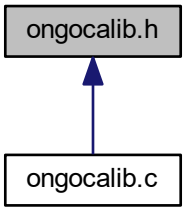
```
#include <stdint.h>
#include <stdbool.h>
```

```
#include <math.h>
```

Include dependency graph for ongocalib.h:



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



Data Structures

- struct **ONGO_ANALOG_SENSOR_DATA**
- struct **ONGO_ROTATION_DATA_STORE**
- struct **ONGO_CALIB_DATA**

Macros

- #define **RAD2DEGFACTOR** (180.00 / M_PI)
- #define **DEG2RADFACTOR** (M_PI / 180.00)
- #define **ANGLE_TOLERANCE** 0.25
- #define **ANGLE_TOLERANCE_RAD** (**ANGLE_TOLERANCE** ∗ **DEG2RADFACTOR**)
- #define **ANGLE180_RAD** (180.00 ∗ **DEG2RADFACTOR**)
- #define **ANGLE45** 45.00
- #define **ANGLE135** 135.00
- #define **ANGLE225** -135.00
- #define **ANGLE315** -45.00
- #define **ANGLE45_H** ((**ANGLE45** + **ANGLE_TOLERANCE**) ∗ **DEG2RADFACTOR**)
- #define **ANGLE45_L** ((**ANGLE45** - **ANGLE_TOLERANCE**) ∗ **DEG2RADFACTOR**)

- #define **ANGLE135_H** ((**ANGLE135** + **ANGLE_TOLERANCE**) ∗ **DEG2RADFACTOR**)
- #define **ANGLE135_L** ((**ANGLE135** - **ANGLE_TOLERANCE**) ∗ **DEG2RADFACTOR**)
- #define **ANGLE225_H** ((**ANGLE225** + **ANGLE_TOLERANCE**) ∗ **DEG2RADFACTOR**)
- #define **ANGLE225_L** ((**ANGLE225** - **ANGLE_TOLERANCE**) ∗ **DEG2RADFACTOR**)
- #define **ANGLE315_H** ((**ANGLE315** + **ANGLE_TOLERANCE**) ∗ **DEG2RADFACTOR**)
- #define **ANGLE315_L** ((**ANGLE315** - **ANGLE_TOLERANCE**) ∗ **DEG2RADFACTOR**)
- #define **ROTATION_VALID** 1u

Typedefs

- typedef double **float64_t**
- typedef struct **ONGO_ANALOG_SENSOR_DATA** **ONGO_ANALOG_SENSOR_DATA_t**
- typedef struct **ONGO_ROTATION_DATA_STORE** **ONGO_ROTATION_DATA_STORE_t**
- typedef struct **ONGO_CALIB_DATA** **ONGO_CALIB_DATA_t**

Functions

- void **ONGO_InitSensorData** (uint32_t sinP_in_LSB, uint32_t cosP_in_LSB, uint32_t sinN_in_LSB, uint32↩ _t cosN_in_LSB)

  Function used for ONGO_ANALOG_SENSOR_DATA_t data assignment.
- void **ONGO_InitCalibData** (**ONGO_CALIB_DATA_t** ∗calib_param, int32_t max_diff_sin_LSB, int32_↩ t max_diff_cos_LSB, int32_t min_diff_sin_LSB, int32_t min_diff_cos_LSB, **float64_t** sin_45, **float64_↩ t** cos_45, **float64_t** sin_135, **float64_t** cos_135)

  Function used for ONGO_CALIB_DATA_t data initialization. The pointer of type ONGO_CALIB_DATA_t will be used to store calibration parameters.
- void **ONGO_InitCalibData2** (**ONGO_CALIB_DATA_t** ∗calib_param, int32_t max_SinP_LSB, int32_t max↩ _CosP_LSB, int32_t max_SinN_LSB, int32_t max_CosN_LSB, int32_t min_SinP_LSB, int32_t min_Cos↩ P_LSB, int32_t min_SinN_LSB, int32_t min_CosN_LSB, **float64_t** sin_45, **float64_t** cos_45, **float64_t** sin_135, **float64_t** cos_135)

  Function used for ONGO_CALIB_DATA_t data initialization. The pointer of type ONGO_CALIB_DATA_t will be used to store calibration parameters.
- **float64_t** **ONGO_GetUncalibAngle** (void)

  Calculates the uncalibrated angle.
- **float64_t** **ONGO_GetCalibAngle** (const **ONGO_CALIB_DATA_t** ∗calib_param)

  Calculates the calibrated angle.
- void **ONGO_CalibrationFindParam** (**ONGO_CALIB_DATA_t** ∗param_store)

  This function is used to find new calibration parameters.

5.2.1 Macro Definition Documentation

5.2.1.1 ANGLE135

```
#define ANGLE135 135.00
```

angle position used for the calibration process.

Definition at line 35 of file ongocalib.h.

5.2.1.2 ANGLE135_H

```
#define ANGLE135_H (( ANGLE135 + ANGLE_TOLERANCE) * DEG2RADFACTOR)
```

angle position limit in radians used for the calibration process.

Definition at line 41 of file ongocalib.h.

5.2.1.3 ANGLE135_L

```
#define ANGLE135_L (( ANGLE135 - ANGLE_TOLERANCE) * DEG2RADFACTOR)
```

angle position limit in radians used for the calibration process.

Definition at line 42 of file ongocalib.h.

5.2.1.4 ANGLE180_RAD

```
#define ANGLE180_RAD (180.00 * DEG2RADFACTOR)
```

angle position in radians used for the calibration process.

Definition at line 32 of file ongocalib.h.

5.2.1.5 ANGLE225

```
#define ANGLE225 -135.00
```

angle position used for the calibration process.

Definition at line 36 of file ongocalib.h.

5.2.1.6 ANGLE225_H

```
#define ANGLE225_H (( ANGLE225 + ANGLE_TOLERANCE) * DEG2RADFACTOR)
```

angle position limit in radians used for the calibration process.

Definition at line 43 of file ongocalib.h.

5.2.1.7 ANGLE225_L

```
#define ANGLE225_L (( ANGLE225 - ANGLE_TOLERANCE) * DEG2RADFACTOR)
```

angle position limit in radians used for the calibration process.

Definition at line 44 of file ongocalib.h.

5.2.1.8 ANGLE315

```
#define ANGLE315 -45.00
```

angle position used for the calibration process.

Definition at line 37 of file ongocalib.h.

5.2.1.9 ANGLE315_H

```
#define ANGLE315_H (( ANGLE315 + ANGLE_TOLERANCE) * DEG2RADFACTOR)
```

angle position limit in radians used for the calibration process.

Definition at line 45 of file ongocalib.h.

5.2.1.10 ANGLE315_L

```
#define ANGLE315_L (( ANGLE315 - ANGLE_TOLERANCE) * DEG2RADFACTOR)
```

angle position limit in radians used for the calibration process.

Definition at line 46 of file ongocalib.h.

5.2.1.11 ANGLE45

```
#define ANGLE45 45.00
```

angle position used for the calibration process.

Definition at line 34 of file ongocalib.h.

5.2.1.12 ANGLE45_H

```
#define ANGLE45_H (( ANGLE45 + ANGLE_TOLERANCE) * DEG2RADFACTOR)
```

angle position limit in radians used for the calibration process.

Definition at line 39 of file ongocalib.h.

5.2.1.13 ANGLE45_L

```
#define ANGLE45_L (( ANGLE45 - ANGLE_TOLERANCE) * DEG2RADFACTOR)
```

angle position limit in radians used for the calibration process.

Definition at line 40 of file ongocalib.h.

5.2.1.14 ANGLE_TOLERANCE

```
#define ANGLE_TOLERANCE 0.25
```

tolerance angle used for calibration process.

Definition at line 30 of file ongocalib.h.

5.2.1.15 ANGLE_TOLERANCE_RAD

```
#define ANGLE_TOLERANCE_RAD ( ANGLE_TOLERANCE * DEG2RADFACTOR)
```

tolerance angle in radians used for the calibration process.

Definition at line 31 of file ongocalib.h.

5.2.1.16 DEG2RADFACTOR

```
#define DEG2RADFACTOR (M_PI / 180.00)
```

Definition at line 29 of file ongocalib.h.

5.2.1.17 RAD2DEGFACTOR

```
#define RAD2DEGFACTOR (180.00 / M_PI)
```

Definition at line 28 of file ongocalib.h.

5.2.1.18 ROTATION_VALID

```
#define ROTATION_VALID 1u
```

Definition at line 48 of file ongocalib.h.

5.2.2 Typedef Documentation

5.2.2.1 float64_t

```
typedef double  float64_t
```

Definition at line 50 of file ongocalib.h.

5.2.2.2 ONGO_ANALOG_SENSOR_DATA_t

```
typedef struct  ONGO_ANALOG_SENSOR_DATA  ONGO_ANALOG_SENSOR_DATA_t
```

5.2.2.3 ONGO_CALIB_DATA_t

```
typedef struct  ONGO_CALIB_DATA  ONGO_CALIB_DATA_t
```

5.2.2.4 ONGO_ROTATION_DATA_STORE_t

```
typedef struct  ONGO_ROTATION_DATA_STORE  ONGO_ROTATION_DATA_STORE_t
```

5.2.3 Function Documentation

5.2.3.1 ONGO_CalibrationFindParam()

```
void ONGO_CalibrationFindParam (
            ONGO_CALIB_DATA_t * calib_param )
```

This function is used to find new calibration parameters.

When new calibration parameters have been found, this function will do mathematical calculations, which will take more time for one moment. This function will extract the maximum, minimum voltage levels, amplitude, offset, and orthogonality. Based on the found values, it will calculate the necessary calibration parameters.

Parameters

in	*calib_param*	is a pointer of type ONGO_CALIB_DATA_t, this structure holds the current sensor calibration parameters.

Returns

Nothing

Definition at line 353 of file ongocalib.c.

5.2.3.2 ONGO_GetCalibAngle()

```
float64_t ONGO_GetCalibAngle (
            const  ONGO_CALIB_DATA_t * calib_param )
```

Calculates the calibrated angle.

Parameters

in	*calib_param*	is a pointer of type ONGO_CALIB_DATA_t, this structure holds the current sensor calibration parameters.

Returns

Returns the calibrated angle in degrees.

Definition at line 337 of file ongocalib.c.

5.2.3.3 ONGO_GetUncalibAngle()

```
float64_t ONGO_GetUncalibAngle (
            void  )
```

Calculates the uncalibrated angle.

This function uses differential sensor data to extract the angle.

Returns

Returns the uncalibrated angle in degrees.

Definition at line 325 of file ongocalib.c.

5.2.3.4 ONGO_InitCalibData()

```
void ONGO_InitCalibData (
              ONGO_CALIB_DATA_t * calib_param,
             int32_t max_diff_sin_LSB,
             int32_t max_diff_cos_LSB,
             int32_t min_diff_sin_LSB,
             int32_t min_diff_cos_LSB,
              float64_t sin_45,
              float64_t cos_45,
              float64_t sin_135,
              float64_t cos_135 )
```

Function used for ONGO_CALIB_DATA_t data initialization. The pointer of type ONGO_CALIB_DATA_t will be used to store calibration parameters.

This function must be called once before ONGO_CalibrationFindParam(..). Use this or ONGO_InitCalibData2(..).

Parameters

in	*calib_param*	is a pointer of type ONGO_CALIB_DATA_t.
in	*max_diff_sin_LSB*	is the maximum differential sin signal in LSB.
in	*max_diff_cos_LSB*	is the maximum differential cos signal in LSB.
in	*min_diff_sin_LSB*	is the minimum differential sin signal in LSB.
in	*min_diff_cos_LSB*	is the minimum differential cos signal in LSB.
in	*sin_45*	is the differential sin component in LSB at a 45 deg position.
in	*cos_45*	is the differential cos component in LSB at a 45 deg position.
in	*sin_135*	is the differential sin component in LSB at a 135 deg position.
in	*cos_135*	is the differential cos component in LSB at a 135 deg position.

Returns

Nothing

Definition at line 200 of file ongocalib.c.

References ONGO_ROTATION_DATA_STORE::angle135found, ONGO_ROTATION_DATA_STORE::angle45found, ONGO_ROTATION_DATA_STORE::maxCos, ONGO_ROTATION_DATA_STORE::maxSin, ONGO_ROTATIO↩N_DATA_STORE::minCos, ONGO_ROTATION_DATA_STORE::minSin, ONGO_ROTATION_DATA_STORE↩::nr_valid_rotations, ONGO_CALIB_DATA::rotation_data, ONGO_ROTATION_DATA_STORE::X135, ONGO_R↩OTATION_DATA_STORE::X45, ONGO_ROTATION_DATA_STORE::Y135, and ONGO_ROTATION_DATA_S↩TORE::Y45.

5.2.3.5 ONGO_InitCalibData2()

```
void ONGO_InitCalibData2 (
              ONGO_CALIB_DATA_t * calib_param,
             int32_t max_SinP_LSB,
             int32_t max_CosP_LSB,
             int32_t max_SinN_LSB,
```

```
            int32_t max_CosN_LSB,
            int32_t min_SinP_LSB,
            int32_t min_CosP_LSB,
            int32_t min_SinN_LSB,
            int32_t min_CosN_LSB,
             float64_t sin_45,
             float64_t cos_45,
             float64_t sin_135,
             float64_t cos_135 )
```

Function used for ONGO_CALIB_DATA_t data initialization. The pointer of type ONGO_CALIB_DATA_t will be used to store calibration parameters.

This function must be called once before ONGO_CalibrationFindParam(..). Use this or ONGO_InitCalibData(..).

Parameters

in	*calib_param*	is a pointer of type ONGO_CALIB_DATA_t.
in	*max_SinP_LSB*	is the maximum P channel sin signal in LSB.
in	*max_CosP_LSB*	is the maximum P channel cos signal in LSB.
in	*max_SinN_LSB*	is the maximum N channel sin signal in LSB.
in	*max_CosN_LSB*	is the maximum N channel cos signal in LSB.
in	*min_SinP_LSB*	is the minimum P channel sin signal in LSB.
in	*min_CosP_LSB*	is the minimum P channel cos signal in LSB.
in	*min_SinN_LSB*	is the minimum N channel sin signal in LSB.
in	*min_CosN_LSB*	is the minimum N channel cos signal in LSB.
in	*sin_45*	is the differential sin component in LSB at a 45 deg position.
in	*cos_45*	is the differential cos component in LSB at a 45 deg position.
in	*sin_135*	is the differential sin component in LSB at a 135 deg position.
in	*cos_135*	is the differential cos component in LSB at a 135 deg position.

Returns

Nothing

Definition at line 258 of file ongocalib.c.

References ONGO_ROTATION_DATA_STORE::angle135found, ONGO_ROTATION_DATA_STORE::angle45found, ONGO_ROTATION_DATA_STORE::maxCos, ONGO_ROTATION_DATA_STORE::maxSin, ONGO_ROTATIO↩N_DATA_STORE::minCos, ONGO_ROTATION_DATA_STORE::minSin, ONGO_ROTATION_DATA_STORE↩::nr_valid_rotations, ONGO_CALIB_DATA::rotation_data, ONGO_ROTATION_DATA_STORE::X135, ONGO_R↩OTATION_DATA_STORE::X45, ONGO_ROTATION_DATA_STORE::Y135, and ONGO_ROTATION_DATA_S↩TORE::Y45.

5.2.3.6 ONGO_InitSensorData()

```
void ONGO_InitSensorData (
            uint32_t sinP_in_LSB,
            uint32_t cosP_in_LSB,
```

```
            uint32_t sinN_in_LSB,
            uint32_t cosN_in_LSB )
```

Function used for ONGO_ANALOG_SENSOR_DATA_t data assignment.

This function must be called after the data readout.

The initialized pointer will be used in the angle calculation and calibration process.

Parameters

in	*cosP_in_LSB*	is the sensor value in LSB corresponding to the positive cosinus.
in	*cosN_in_LSB*	is the sensor value in LSB corresponding to the negative cosinus.
in	*sinP_in_LSB*	is the sensor value in LSB corresponding to the positive sinus.
in	*sinN_in_LSB*	is the sensor value in LSB corresponding to the negative sinus.

Returns

Nothing

Definition at line 164 of file ongocalib.c.

Index

ANGLE135
ongocalib.h, 25
ANGLE135_H
ongocalib.h, 25
ANGLE135_L
ongocalib.h, 26
ANGLE180_RAD
ongocalib.h, 26
ANGLE225
ongocalib.h, 26
ANGLE225_H
ongocalib.h, 26
ANGLE225_L
ongocalib.h, 26
ANGLE315
ongocalib.h, 27
ANGLE315_H
ongocalib.h, 27
ANGLE315_L
ongocalib.h, 27
ANGLE45
ongocalib.h, 27
ANGLE45_H
ongocalib.h, 27
ANGLE45_L
ongocalib.h, 28
ANGLE_TOLERANCE_RAD
ongocalib.h, 28
ANGLE_TOLERANCE
ongocalib.h, 28
amplitudeX
ONGO_ROTATION_DATA_STORE, 13
amplitudeY
ONGO_ROTATION_DATA_STORE, 13
angle135found
ONGO_ROTATION_DATA_STORE, 13
angle45found
ONGO_ROTATION_DATA_STORE, 13

cos_ortho
ONGO_CALIB_DATA, 11
cosN_LSB
ONGO_ANALOG_SENSOR_DATA, 9
cosP_LSB
ONGO_ANALOG_SENSOR_DATA, 9

DEG2RADFACTOR
ongocalib.h, 28
diff_X
ONGO_ANALOG_SENSOR_DATA, 10
diff_Y
ONGO_ANALOG_SENSOR_DATA, 10

float64_t
ongocalib.h, 29

magnitude135
ONGO_ROTATION_DATA_STORE, 13
magnitude45
ONGO_ROTATION_DATA_STORE, 14
maxCos
ONGO_ROTATION_DATA_STORE, 14
maxSin
ONGO_ROTATION_DATA_STORE, 14
minCos
ONGO_ROTATION_DATA_STORE, 14
minSin
ONGO_ROTATION_DATA_STORE, 14

nr_valid_rotations
ONGO_ROTATION_DATA_STORE, 15

ONGO_ANALOG_SENSOR_DATA_t
ongocalib.h, 29
ONGO_ANALOG_SENSOR_DATA, 9
cosN_LSB, 9
cosP_LSB, 9
diff_X, 10
diff_Y, 10
sinN_LSB, 10
sinP_LSB, 10
uncalibratedAngle, 10
ONGO_CALIB_DATA_t
ongocalib.h, 29
ONGO_CALIB_DATA, 11
cos_ortho, 11
rotation_data, 12
sin_ortho, 12
ONGO_CalibrationFindParam
ongocalib.c, 20
ongocalib.h, 29
ONGO_GetCalibAngle
ongocalib.c, 20
ongocalib.h, 30
ONGO_GetUncalibAngle
ongocalib.c, 21
ongocalib.h, 30
ONGO_InitCalibData
ongocalib.c, 21
ongocalib.h, 30

ONGO_InitCalibData2
ongocalib.c, 22
ongocalib.h, 31
ONGO_InitSensorData
ongocalib.c, 23
ongocalib.h, 32
ONGO_ROTATION_DATA_STORE_t
ongocalib.h, 29
ONGO_ROTATION_DATA_STORE, 12
amplitudeX, 13
amplitudeY, 13
angle135found, 13
angle45found, 13
magnitude135, 13
magnitude45, 14
maxCos, 14
maxSin, 14
minCos, 14
minSin, 14
nr_valid_rotations, 15
offsetX, 15
offsetY, 15
ortho_one_dir, 15
X135, 15
X135_corr, 16
X45, 16
X45_corr, 16
Y135, 16
Y135_corr, 16
Y45, 17
Y45_corr, 17
offsetX
ONGO_ROTATION_DATA_STORE, 15
offsetY
ONGO_ROTATION_DATA_STORE, 15
ongocalib.c, 19
ONGO_CalibrationFindParam, 20
ONGO_GetCalibAngle, 20
ONGO_GetUncalibAngle, 21
ONGO_InitCalibData, 21
ONGO_InitCalibData2, 22
ONGO_InitSensorData, 23
ongocalib.h, 23
ANGLE135, 25
ANGLE135_H, 25
ANGLE135_L, 26
ANGLE180_RAD, 26
ANGLE225, 26
ANGLE225_H, 26
ANGLE225_L, 26
ANGLE315, 27
ANGLE315_H, 27
ANGLE315_L, 27
ANGLE45, 27
ANGLE45_H, 27
ANGLE45_L, 28
ANGLE_TOLERANCE_RAD, 28
ANGLE_TOLERANCE, 28
DEG2RADFACTOR, 28
float64_t, 29
ONGO_ANALOG_SENSOR_DATA_t, 29
ONGO_CALIB_DATA_t, 29
ONGO_CalibrationFindParam, 29
ONGO_GetCalibAngle, 30
ONGO_GetUncalibAngle, 30
ONGO_InitCalibData, 30
ONGO_InitCalibData2, 31
ONGO_InitSensorData, 32
ONGO_ROTATION_DATA_STORE_t, 29
RAD2DEGFACTOR, 28
ROTATION_VALID, 29
ortho_one_dir
ONGO_ROTATION_DATA_STORE, 15

RAD2DEGFACTOR
ongocalib.h, 28
ROTATION_VALID
ongocalib.h, 29
rotation_data
ONGO_CALIB_DATA, 12

sin_ortho
ONGO_CALIB_DATA, 12
sinN_LSB
ONGO_ANALOG_SENSOR_DATA, 10
sinP_LSB
ONGO_ANALOG_SENSOR_DATA, 10

uncalibratedAngle
ONGO_ANALOG_SENSOR_DATA, 10

X135
ONGO_ROTATION_DATA_STORE, 15
X135_corr
ONGO_ROTATION_DATA_STORE, 16
X45
ONGO_ROTATION_DATA_STORE, 16
X45_corr
ONGO_ROTATION_DATA_STORE, 16

Y135
ONGO_ROTATION_DATA_STORE, 16
Y135_corr
ONGO_ROTATION_DATA_STORE, 16
Y45
ONGO_ROTATION_DATA_STORE, 17
Y45_corr
ONGO_ROTATION_DATA_STORE, 17