

Trivisio Colibri
2.1.2

Generated by Doxygen 1.6.3

Fri Jul 29 10:30:54 2011

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	ColibriConfig Struct Reference	5
3.1.1	Detailed Description	6
3.1.2	Member Enumeration Documentation	6
3.1.2.1	Sensor	6
3.1.3	Field Documentation	6
3.1.3.1	ascii	6
3.1.3.2	autoStart	6
3.1.3.3	freq	7
3.1.3.4	magDiv	7
3.1.3.5	raw	7
3.1.3.6	sensor	7
3.2	TrivisioIMUCalibration Struct Reference	8
3.2.1	Detailed Description	8
3.2.2	Field Documentation	8
3.2.2.1	ba	8
3.2.2.2	bg	8
3.2.2.3	bm	8
3.2.2.4	ka	9
3.2.2.5	kg	9

3.2.2.6	Kga	9
3.2.2.7	km	9
3.2.2.8	Ra	9
3.2.2.9	Rg	9
3.2.2.10	Rm	9
3.3	TrivisioIMUData Struct Reference	10
3.3.1	Detailed Description	10
3.3.2	Field Documentation	10
3.3.2.1	acc_x	10
3.3.2.2	acc_y	10
3.3.2.3	acc_z	11
3.3.2.4	gyr_x	11
3.3.2.5	gyr_y	11
3.3.2.6	gyr_z	11
3.3.2.7	mag_x	11
3.3.2.8	mag_y	11
3.3.2.9	mag_z	11
3.3.2.10	q_w	11
3.3.2.11	q_x	12
3.3.2.12	q_y	12
3.3.2.13	q_z	12
3.3.2.14	t	12
3.3.2.15	temp	12
3.4	TrivisioSensor Struct Reference	13
3.4.1	Detailed Description	13
3.4.2	Member Enumeration Documentation	13
3.4.2.1	SensorType	13
3.4.3	Field Documentation	13
3.4.3.1	dev	13
3.4.3.2	FWsubver	13
3.4.3.3	FWver	14
3.4.3.4	ID	14
3.4.3.5	type	14

4	File Documentation	15
4.1	TrivisioColibri.h File Reference	15
4.1.1	Function Documentation	16
4.1.1.1	colibriAPIVersion	16
4.1.1.2	colibriBoresight	16
4.1.1.3	colibriClose	17
4.1.1.4	colibriCreate	17
4.1.1.5	colibriDestroy	17
4.1.1.6	colibriEulerOri	18
4.1.1.7	colibriGetBoresight	18
4.1.1.8	colibriGetCalib	18
4.1.1.9	colibriGetConfig	18
4.1.1.10	colibriGetData	19
4.1.1.11	colibriGetDeviceList	19
4.1.1.12	colibriGetID	19
4.1.1.13	colibriGetJitterParam	20
4.1.1.14	colibriGetJitterStatus	20
4.1.1.15	colibriGetKa	20
4.1.1.16	colibriGetKaStatus	21
4.1.1.17	colibriGetKg	21
4.1.1.18	colibriGetKgStatus	21
4.1.1.19	colibriGetKm	22
4.1.1.20	colibriGetKmStatus	22
4.1.1.21	colibriGetSensorInfo	22
4.1.1.22	colibriGetTriggerDivisor	22
4.1.1.23	colibriOpen	23
4.1.1.24	colibriSaveSettings	23
4.1.1.25	colibriSetBoresight	23
4.1.1.26	colibriSetCalib	24
4.1.1.27	colibriSetConfig	24
4.1.1.28	colibriSetJitterParam	24
4.1.1.29	colibriSetJitterStatus	25
4.1.1.30	colibriSetKa	25
4.1.1.31	colibriSetKaStatus	25

4.1.1.32	colibriSetKg	25
4.1.1.33	colibriSetKgStatus	26
4.1.1.34	colibriSetKm	26
4.1.1.35	colibriSetKmStatus	26
4.1.1.36	colibriSetTriggerDivisor	26
4.1.1.37	colibriStart	27
4.1.1.38	colibriStop	27
4.2	TrivisioColibri.h	28
4.3	TrivisioConfig.h File Reference	31
4.3.1	Define Documentation	31
4.3.1.1	Trivisio_DECLSPEC	31
4.4	TrivisioConfig.h	32
4.5	TrivisioTypes.h File Reference	33
4.5.1	Enumeration Type Documentation	33
4.5.1.1	ColibriBoresightType	33
4.6	TrivisioTypes.h	34

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

ColibriConfig (Colibri configuration structure)	5
TrivisioIMUCalibration (TrivisioIMUCalibration)	8
TrivisioIMUData (TrivisioIMUData)	10
TrivisioSensor (Trivisio sensor structure)	13

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

TrivisioColibri.h	15
TrivisioConfig.h	31
TrivisioTypes.h	33

Chapter 3

Data Structure Documentation

3.1 ColibriConfig Struct Reference

Colibri configuration structure.

```
#include <TrivisioTypes.h>
```

Public Types

- enum `Sensor` {
 [MAG_X](#) = 0x0001, [MAG_Y](#) = 0x0002, [MAG_Z](#) = 0x0004, [MAG](#) = 0x0007,
 [ACC_X](#) = 0x0008, [ACC_Y](#) = 0x0010, [ACC_Z](#) = 0x0020, [ACC](#) = 0x0038,
 [GYR_X](#) = 0x0040, [GYR_Y](#) = 0x0080, [GYR_Z](#) = 0x0100, [GYR](#) = 0x01C0,
 [TEMP](#) = 0x0200, [GARBAGE](#) = 0x0400, [ORIENT](#) = 0x0800, [ALL](#) = 0x0BFF
}

Sensor constants.

Data Fields

- `uint8_t` [ascii](#):1
- `uint8_t` [autoStart](#):1
- `uint16_t` [freq](#)
- `uint8_t` [magDiv](#)
- `uint8_t` [raw](#):1
- enum `ColibriConfig::Sensor` [sensor](#)

Sensor constants.

3.1.1 Detailed Description

Colibri configuration structure. The fields of the [ColibriConfig](#) describes the setting of the Colibri.

Definition at line 48 of file [TrivisioTypes.h](#).

3.1.2 Member Enumeration Documentation

3.1.2.1 enum ColibriConfig::Sensor

Sensor constants.

Enumerator:

MAG_X Magnetometer x-axis
MAG_Y Magnetometer y-axis
MAG_Z Magnetometer z-axis
MAG All magnetometers
ACC_X Accelerometer x-axis
ACC_Y Accelerometer y-axis
ACC_Z Accelerometer z-axis
ACC All accelerometers
GYR_X Gyroscope x-axis
GYR_Y Gyroscope y-axis
GYR_Z Gyroscope z-axis
GYR All gyroscopes
TEMP Chip temperature
GARBAGE Reserved, should be 0
ORIENT Orientation
ALL All available sensors

Definition at line 51 of file [TrivisioTypes.h](#).

3.1.3 Field Documentation

3.1.3.1 uint8_t ColibriConfig::ascii

ASCII output: on/off

Definition at line 71 of file [TrivisioTypes.h](#).

3.1.3.2 uint8_t ColibriConfig::autoStart

Sensor auto-start on turn on

Definition at line 72 of file [TrivisioTypes.h](#).

3.1.3.3 uint16_t ColibriConfig::freq

Sampling frequency, 0 implies external triggering

Definition at line 70 of file [TrivisioTypes.h](#).

3.1.3.4 uint8_t ColibriConfig::magDiv

Magnetometer divisor $2^{(5+\text{magDiv})}$

Definition at line 69 of file [TrivisioTypes.h](#).

3.1.3.5 uint8_t ColibriConfig::raw

Raw data switch (raw/calibrated)

Definition at line 73 of file [TrivisioTypes.h](#).

3.1.3.6 enum ColibriConfig::Sensor ColibriConfig::sensor

Sensor constants.

3.2 TrivisioIMUCalibration Struct Reference

[TrivisioIMUCalibration](#).

```
#include <TrivisioTypes.h>
```

Data Fields

- float [ba](#) [3]
- float [bg](#) [3]
- float [bm](#) [3]
- float [ka](#) [3]
- float [kg](#) [3]
- float [Kga](#) [9]
- float [km](#) [3]
- float [Ra](#) [9]
- float [Rg](#) [9]
- float [Rm](#) [9]

3.2.1 Detailed Description

[TrivisioIMUCalibration](#). This structure contains the parameters used when calibrating the a Trivisio Colibri.

Definition at line [106](#) of file [TrivisioTypes.h](#).

3.2.2 Field Documentation

3.2.2.1 float [TrivisioIMUCalibration::ba](#)[3]

Accelerometer alignment compensation

Definition at line [109](#) of file [TrivisioTypes.h](#).

3.2.2.2 float [TrivisioIMUCalibration::bg](#)[3]

Gyroscope alignment compensation

Definition at line [113](#) of file [TrivisioTypes.h](#).

3.2.2.3 float [TrivisioIMUCalibration::bm](#)[3]

Magnetometer alignment compensation

Definition at line [118](#) of file [TrivisioTypes.h](#).

3.2.2.4 float TrivisioIMUCalibration::ka[3]

Accelerometer bias

Definition at line 110 of file [TrivisioTypes.h](#).

3.2.2.5 float TrivisioIMUCalibration::kg[3]

Gyroscope bias

Definition at line 114 of file [TrivisioTypes.h](#).

3.2.2.6 float TrivisioIMUCalibration::Kga[9]

Gyroscope scale factor

Definition at line 115 of file [TrivisioTypes.h](#).

3.2.2.7 float TrivisioIMUCalibration::km[3]

Magnetometer bias

Definition at line 119 of file [TrivisioTypes.h](#).

3.2.2.8 float TrivisioIMUCalibration::Ra[9]

Definition at line 108 of file [TrivisioTypes.h](#).

3.2.2.9 float TrivisioIMUCalibration::Rg[9]

Accelerometer scale factor

Definition at line 112 of file [TrivisioTypes.h](#).

3.2.2.10 float TrivisioIMUCalibration::Rm[9]

Acceleration influence on gyroscope

Definition at line 117 of file [TrivisioTypes.h](#).

3.3 TrivisioIMUData Struct Reference

[TrivisioIMUData](#).

```
#include <TrivisioTypes.h>
```

Data Fields

- float [acc_x](#)
- float [acc_y](#)
- float [acc_z](#)
- float [gyr_x](#)
- float [gyr_y](#)
- float [gyr_z](#)
- float [mag_x](#)
- float [mag_y](#)
- float [mag_z](#)
- float [q_w](#)
- float [q_x](#)
- float [q_y](#)
- float [q_z](#)
- uint32_t [t](#)
- float [temp](#)

3.3.1 Detailed Description

[TrivisioIMUData](#). This structure is used to return data from the IMU.

Definition at line 81 of file [TrivisioTypes.h](#).

3.3.2 Field Documentation

3.3.2.1 float TrivisioIMUData::acc_x

Accelerometer reading, x-axis

Definition at line 87 of file [TrivisioTypes.h](#).

3.3.2.2 float TrivisioIMUData::acc_y

Accelerometer reading, y-axis

Definition at line 88 of file [TrivisioTypes.h](#).

3.3.2.3 float TrivisioIMUData::acc_z

Accelerometer reading, z-axis

Definition at line 89 of file [TrivisioTypes.h](#).

3.3.2.4 float TrivisioIMUData::gyr_x

Gyroscope reading, x-axis

Definition at line 90 of file [TrivisioTypes.h](#).

3.3.2.5 float TrivisioIMUData::gyr_y

Gyroscope reading, y-axis

Definition at line 91 of file [TrivisioTypes.h](#).

3.3.2.6 float TrivisioIMUData::gyr_z

Gyroscope reading, z-axis

Definition at line 92 of file [TrivisioTypes.h](#).

3.3.2.7 float TrivisioIMUData::mag_x

Magnetometer reading, x-axis

Definition at line 84 of file [TrivisioTypes.h](#).

3.3.2.8 float TrivisioIMUData::mag_y

Magnetometer reading, y-axis

Definition at line 85 of file [TrivisioTypes.h](#).

3.3.2.9 float TrivisioIMUData::mag_z

Magnetometer reading, z-axis

Definition at line 86 of file [TrivisioTypes.h](#).

3.3.2.10 float TrivisioIMUData::q_w

Orientation quaternion, w component

Definition at line 94 of file [TrivisioTypes.h](#).

3.3.2.11 float TrivisioIMUData::q_x

Orientation quaternion, x component

Definition at line 95 of file [TrivisioTypes.h](#).

3.3.2.12 float TrivisioIMUData::q_y

Orientation quaternion, y component

Definition at line 96 of file [TrivisioTypes.h](#).

3.3.2.13 float TrivisioIMUData::q_z

Orientation quaternion, z component

Definition at line 97 of file [TrivisioTypes.h](#).

3.3.2.14 uint32_t TrivisioIMUData::t

Time [0.1ms]

Definition at line 83 of file [TrivisioTypes.h](#).

3.3.2.15 float TrivisioIMUData::temp

Temperature reading

Definition at line 93 of file [TrivisioTypes.h](#).

3.4 TrivisioSensor Struct Reference

Trivisio sensor structure.

```
#include <TrivisioTypes.h>
```

Public Types

- enum [SensorType](#) { [COLIBRI](#) }

Data Fields

- char [dev](#) [256]
- uint16_t [FWsubver](#)
- uint16_t [FWver](#)
- char [ID](#) [8]
- enum [TrivisioSensor::SensorType](#) type

3.4.1 Detailed Description

Trivisio sensor structure. The fields of the [TrivisioSensor](#) describes the sensor and how to access it.

Definition at line [20](#) of file [TrivisioTypes.h](#).

3.4.2 Member Enumeration Documentation

3.4.2.1 enum TrivisioSensor::SensorType

Enumerator:

COLIBRI Colibri IMU

Definition at line [25](#) of file [TrivisioTypes.h](#).

3.4.3 Field Documentation

3.4.3.1 char TrivisioSensor::dev[256]

Device name where found

Definition at line [22](#) of file [TrivisioTypes.h](#).

3.4.3.2 uint16_t TrivisioSensor::FWsubver

Firmware subversion

Definition at line [29](#) of file [TrivisioTypes.h](#).

3.4.3.3 uint16_t TrivisioSensor::FWver

Firmware version

Definition at line 28 of file [TrivisioTypes.h](#).

3.4.3.4 char TrivisioSensor::ID[8]

Unique unit ID

Definition at line 23 of file [TrivisioTypes.h](#).

3.4.3.5 enum TrivisioSensor::SensorType TrivisioSensor::type

Chapter 4

File Documentation

4.1 TrivisioColibri.h File Reference

```
#include "TrivisioConfig.h"  
#include "TrivisioConfig.h"  
#include <stdint.h>
```

Functions

- Trivisio_DECLSPEC const char * [colibriAPIVersion](#) ()
- Trivisio_DECLSPEC void [colibriBoresight](#) (void *imu, enum [ColibriBoresight-Type](#) type)
- Trivisio_DECLSPEC void [colibriClose](#) (void *imu)
- Trivisio_DECLSPEC void * [colibriCreate](#) (unsigned short bufLen)
- Trivisio_DECLSPEC void [colibriDestroy](#) (void *imu)
- Trivisio_DECLSPEC void [colibriEulerOri](#) (const struct [TrivisioIMUData](#) *data, float euler[3])
- Trivisio_DECLSPEC int [colibriGetBoresight](#) (void *imu)
- Trivisio_DECLSPEC void [colibriGetCalib](#) (void *imu, struct [TrivisioIMUCalibration](#) *calib)
- Trivisio_DECLSPEC void [colibriGetConfig](#) (void *imu, struct [ColibriConfig](#) *config)
- Trivisio_DECLSPEC void [colibriGetData](#) (void *imu, struct [TrivisioIMUData](#) *data)
- Trivisio_DECLSPEC int [colibriGetDeviceList](#) (struct [TrivisioSensor](#) *list, unsigned listLen)
- Trivisio_DECLSPEC void [colibriGetID](#) (void *imu, char *ID)
- Trivisio_DECLSPEC void [colibriGetJitterParam](#) (void *imu, float *lambda, float *th, float *tol)
- Trivisio_DECLSPEC int [colibriGetJitterStatus](#) (void *imu)
- Trivisio_DECLSPEC void [colibriGetKa](#) (void *imu, float Ka[9])

- Trivisio_DECLSPEC int [colibriGetKaStatus](#) (void *imu)
- Trivisio_DECLSPEC void [colibriGetKg](#) (void *imu, float Kg[9])
- Trivisio_DECLSPEC int [colibriGetKgStatus](#) (void *imu)
- Trivisio_DECLSPEC void [colibriGetKm](#) (void *imu, float Km[9])
- Trivisio_DECLSPEC int [colibriGetKmStatus](#) (void *imu)
- Trivisio_DECLSPEC void [colibriGetSensorInfo](#) (void *imu, struct [TrivisioSensor](#) *info)
- Trivisio_DECLSPEC uint8_t [colibriGetTriggerDivisor](#) (void *imu)
- Trivisio_DECLSPEC int [colibriOpen](#) (void *imu, const struct [ColibriConfig](#) *conf, const char *dev)
- Trivisio_DECLSPEC void [colibriSaveSettings](#) (void *imu)
- Trivisio_DECLSPEC void [colibriSetBoresight](#) (void *imu, int active)
- Trivisio_DECLSPEC void [colibriSetCalib](#) (void *imu, const struct [TrivisioIMU-Calibration](#) *calib)
- Trivisio_DECLSPEC void [colibriSetConfig](#) (void *imu, const struct [ColibriConfig](#) *config)
- Trivisio_DECLSPEC void [colibriSetJitterParam](#) (void *imu, float lambda, float th, float tol)
- Trivisio_DECLSPEC void [colibriSetJitterStatus](#) (void *imu, int status)
- Trivisio_DECLSPEC void [colibriSetKa](#) (void *imu, const float Ka[9])
- Trivisio_DECLSPEC void [colibriSetKaStatus](#) (void *imu, int active)
- Trivisio_DECLSPEC void [colibriSetKg](#) (void *imu, const float Kg[9])
- Trivisio_DECLSPEC void [colibriSetKgStatus](#) (void *imu, int active)
- Trivisio_DECLSPEC void [colibriSetKm](#) (void *imu, const float Km[9])
- Trivisio_DECLSPEC void [colibriSetKmStatus](#) (void *imu, int active)
- Trivisio_DECLSPEC void [colibriSetTriggerDivisor](#) (void *imu, uint8_t div)
- Trivisio_DECLSPEC int [colibriStart](#) (void *imu)
- Trivisio_DECLSPEC void [colibriStop](#) (void *imu)

4.1.1 Function Documentation

4.1.1.1 Trivisio_DECLSPEC const char* colibriAPIVersion ()

Get the version of the API

Returns

Returns a string that identifies the exact version of the API used.

4.1.1.2 Trivisio_DECLSPEC void colibriBoresight (void * imu, enum ColibriBoresightType type)

Boresights the sensor

Performs a boresight calibration of the sensor, and activates boresighting.

Parameters

imu Colibri handle.

type Tells what type boresighting should be used in the calibration.

Remarks

As of firmware version 1.407 the boresight status is saved in the sensor, before that the sensor does not remember it's boresight status. Since this command should be used when the sensor is running, the information is put on the sensor as it is stopped.

4.1.1.3 Trivisio_DECLSPEC void colibriClose (void * *imu*)

Close connection between Colibri handle and hardware

Close the connection established between given handle and IMU hardware.

Parameters

imu Colibri handle.

4.1.1.4 Trivisio_DECLSPEC void* colibriCreate (unsigned short *bufLen*)

Create new Colibri handle

The function creates and initializes a new Colibri handle which is used to interact with the physical IMU.

Parameters

bufLen size of the internal output buffer to be used. A short buffer (0) ensures minimal delay until the sensor measurement is available at the risk of lost measurements. A long buffer guarantees that no data is dropped, at the same time if data is not read fast enough there is a potential risk of a $bufLen * frequency$ before the measurement becomes available.

Returns

New Colibri handle.

4.1.1.5 Trivisio_DECLSPEC void colibriDestroy (void * *imu*)

Destroy a Colibri handle

The function destroys a Colibri handle and returns the system resources allocated by it.

Parameters

imu Colibri handle to destroy.

4.1.1.6 Trivisio_DECLSPEC void colibriEulerOri (const struct TrivisioIMUData * data, float euler[3])

Convert orientation to Euler angles

Converts the orientation given in the measured data to Euler angles.

Parameters

data The data containing the orientation information (quaternion)

euler Contains the Euler angle after the call

4.1.1.7 Trivisio_DECLSPEC int colibriGetBoresight (void * imu)

Get sensor boresighting status

Gets the current boresighting setting of sensor.

Parameters

imu Colibri handle.

Returns

Returns non-zero if boresighting is activated.

4.1.1.8 Trivisio_DECLSPEC void colibriGetCalib (void * imu, struct TrivisioIMUCalibration * calib)

Get sensor calibration

Retrieves the specific Colibri calibration of the selected unit.

Parameters

imu Colibri handle.

calib Contains the unit specific calibration data on return.

4.1.1.9 Trivisio_DECLSPEC void colibriGetConfig (void * imu, struct ColibriConfig * config)

Get Colibri configuration

Retrieves the current sensor configuration from a Colibri. Only use this function when the sensor is not running.

Parameters

imu Colibri handle.

config Current sensor configuration.

4.1.1.10 Trivisio_DECLSPEC void colibriGetData (void * *imu*, struct TrivisioIMUData * *data*)

Get sensor data

Retrieves a sensor reading from the Colibri. The function always returns a measurement! Either the last sensor reading available from the hardware, or a default empty value. To determine if the returned data is new, check the time stamp that should be monotonically increasing. The reading from a stopped sensor has the time set to 0.

Parameters

imu Colibri handle.

data Read data, see discussion above for details.

4.1.1.11 Trivisio_DECLSPEC int colibriGetDeviceList (struct TrivisioSensor * *list*, unsigned *listLen*)

Detect available sensors

Returns a list of device names which host an available Colibri.

Parameters

list List of [TrivisioSensor](#) to which to write the result.

listLen How many elements there are in list.

Returns

Returns how many devices were found. If the number of devices exceeds size of dev as provided in listLen, the listLen first devices are written to dev and -1 is returned.

4.1.1.12 Trivisio_DECLSPEC void colibriGetID (void * *imu*, char * *ID*)

Get sensor ID

Retrieves the unique Colibri ID.

Parameters

imu Colibri handle.

ID Pointer to a char* to where the ID should be written. Supplied string must be at least 7 chars long.

4.1.1.13 Trivisio_DECLSPEC void colibriGetJitterParam (void * *imu*, float * *lambda*, float * *th*, float * *tol*)

Get stationary jitter reduction parameters for sensor

Gets the parameters used for the jitter reduction.

Parameters

imu Colibri handle.

lambda Contains the smoothing factor on return.

th Determines how aggressive the jitter reduction is on

tol Determines the weighting of slow vs fast motion return.

Warning

This function is still experimental state. Do not relay on it to behave the same way or even stay in the API between different releases.

4.1.1.14 Trivisio_DECLSPEC int colibriGetJitterStatus (void * *imu*)

Get stationary jitter reduction status for sensor

Gets the status of the jitter reduction for stationary sensor.

Parameters

imu Colibri handle.

Returns

0 if jitter reduction is deactivated.

4.1.1.15 Trivisio_DECLSPEC void colibriGetKa (void * *imu*, float *Ka*[9])

Get filter gain for accelerometers

Gets the filter gain for accelerometer measurements.

Parameters

imu Colibri handle

Ka On return, contains filter gain (3x3) row major matrix.

Get filter gain for accelerometers

Gets the filter gain for accelerometer measurements.

Parameters

imu Colibri handle.

Ka Contains the filter gain (3x3) row major matrix, after the call.

4.1.1.16 Trivisio_DECLSPEC int colibriGetKaStatus (void * *imu*)

Get status of Kalman filtering of accelerometer data

Parameters

imu Colibri handle.

Returns

Non-zero if the filter is active, otherwise zero.

4.1.1.17 Trivisio_DECLSPEC void colibriGetKg (void * *imu*, float *Kg*[9])

Get filter gain for gyroscopes

Sets the filter gain for gyroscope measurements.

Parameters

imu Colibri handle.

Kg On return, contains filter gain (3x3) row major matrix.

Get filter gain for magnetometers

Sets the filter gain for magnetometer measurements.

Parameters

imu Colibri handle

Kg On return, contains filter gain (3x3) row major matrix

Get filter gain for gyroscopes

Gets the filter gain for gyroscope measurements.

Parameters

imu Colibri handle.

Kg Contains the filter gain (3x3) row major matrix, after the call.

4.1.1.18 Trivisio_DECLSPEC int colibriGetKgStatus (void * *imu*)

Get status of F/W Kalman filtering of gyroscope data

Parameters

imu Colibri handle.

Returns

Non-zero if the filter is active, otherwise zero.

4.1.1.19 Trivisio_DECLSPEC void colibriGetKm (void * *imu*, float *Km*[9])

Get filter gain for magnetometers

Gets the filter gain for magnetometer measurements.

Parameters

imu Colibri handle.

Km Contains the filter gain (3x3) row major matrix, after the call.

4.1.1.20 Trivisio_DECLSPEC int colibriGetKmStatus (void * *imu*)

Get status of Kalman filtering of magnetometer data

Parameters

imu Handle to Colibri.

Returns

Non-zero if the filter is active, otherwise zero.

4.1.1.21 Trivisio_DECLSPEC void colibriGetSensorInfo (void * *imu*, struct TrivisioSensor * *info*)

Acquire sensor information

Query the sensor for details about it.

Parameters

imu Colibri handle.

info Structure to which the information will be written on return.

4.1.1.22 Trivisio_DECLSPEC uint8_t colibriGetTriggerDivisor (void * *imu*)

Get trigger status and divisor

Gets the trigger status and divisor of a Colibri.

Parameters

imu Colibri handle

Returns

The trigger divisor. If 0 the no trigger signal is emitted.

4.1.1.23 Trivisio_DECLSPEC int colibriOpen (void * *imu*, const struct ColibriConfig * *conf*, const char * *dev*)

Open connection from Colibri handle to hardware

The tries to connect to and open up a connection to a Colibri IMU.

Parameters

imu Colibri handle to associate with sensor.

conf Configuration to open the sensor with. The default (obtained with *conf*=0) is to only turn the sensor silent.

dev Device to associate the handle with. If 0, the functions looks for sensors in the registry (Windows) and connected to /dev/ttyACM<N> (Linux).

Returns

0 on success, otherwise non-zero.

4.1.1.24 Trivisio_DECLSPEC void colibriSaveSettings (void * *imu*)

Save sensor settings permanently in sensor

Saves the current Colibri settings permanently.

Parameters

imu Colibri handle

4.1.1.25 Trivisio_DECLSPEC void colibriSetBoresight (void * *imu*, int *active*)

Set sensor boresighting status

Sets the current boresighting setting of sensor.

Parameters

imu Colibri handle.

active Activates boresighting if non-zero, otherwise turns it off.

Remarks

As of firmware version 1.407 the boresight status is saved in the sensor, before that the sensor does not remember it's boresight status. Since this command should be used when the sensor is running, the information is put on the sensor as it is stopped.

4.1.1.26 Trivisio_DECLSPEC void colibriSetCalib (void * *imu*, const struct TrivisioIMUCalibration * *calib*)

Set sensor calibration

Sets the specific Colibri calibration of the selected unit.

Parameters

imu Colibri handle.

calib Contains the unit specific calibration data.

Warning

Overwriting the factory set calibration data, which is finely tuned to every individual sensor, may have serious impact on sensor performance. Therefore, make sure to have a copy of the factory settings before overwriting and saving new calibration information on the sensor.

4.1.1.27 Trivisio_DECLSPEC void colibriSetConfig (void * *imu*, const struct ColibriConfig * *config*)

Set Colibri configuration

Set the current sensor configuration. Only use this function when the sensor is not running.

Parameters

imu Colibri handle.

config Configuration to load into the sensor, if config=0 apply the default configuration: activating all sensors, magDiv=3, freq=100, ascii=false, autoStart=1, and raw=0.

4.1.1.28 Trivisio_DECLSPEC void colibriSetJitterParam (void * *imu*, float *lambda*, float *th*, float *tol*)

Set stationary jitter reduction parameters for sensor

Sets the parameters used for the jitter reduction.

Parameters

imu Colibri handle.

lambda Contains the smoothing factor [0, 1]. A larger value makes the jitter reduction "smoother". (Defaults to 0.9.)

th Determines how aggressive the jitter reduction is [0, inf], a high value indicates aggressive jitter reduction. (Defaults to 0.01.)

tol Weighing between slow and fast motion [0, 1]. A smaller value will move quicker between the two. (Defaults to 0.1)

4.1.1.29 Trivisio_DECLSPEC void colibriSetJitterStatus (void * *imu*, int *status*)

Set stationary jitter reduction status for sensor

Sets the jitter reduction of stationary sensor.

Parameters

imu Colibri handle.

status If 0 turns off jitter reduction, otherwise activates it.

4.1.1.30 Trivisio_DECLSPEC void colibriSetKa (void * *imu*, const float *Ka*[9])

Set filter gain for accelerometers

Sets the filter gain for accelerometer measurements. The default action is no filtering.

Parameters

imu Colibri handle

Ka Filter gain (3x3) row major matrix.

4.1.1.31 Trivisio_DECLSPEC void colibriSetKaStatus (void * *imu*, int *active*)

Set status of Kalman filtering of accelerometer data

Parameters

imu Colibri handle.

active Set to non-zero if the filter should be activated, otherwise zero.

4.1.1.32 Trivisio_DECLSPEC void colibriSetKg (void * *imu*, const float *Kg*[9])

Set filter gain for gyroscopes

Sets the filter gain for gyroscope measurements. The default action is no filtering.

Parameters

imu Colibri handle.

Kg Filter gain (3x3) row major matrix.

4.1.1.33 Trivisio_DECLSPEC void colibriSetKgStatus (void * *imu*, int *active*)

Set status of Kalman filtering of gyroscope data

Parameters

imu Colibri handle.

active Set to non-zero if the filter should be activated, otherwise zero.

4.1.1.34 Trivisio_DECLSPEC void colibriSetKm (void * *imu*, const float *Km*[9])

Set filter gain for magnetometers

Sets the filter gain for magnetometer measurements. The default action is no filtering.

Parameters

imu Colibri handle.

Km Filter gain (3x3) row major matrix.

4.1.1.35 Trivisio_DECLSPEC void colibriSetKmStatus (void * *imu*, int *active*)

Set status of Kalman filtering of magnetometer data

Parameters

imu Colibri handle.

active Set to non-zero if the filter should be activated, otherwise zero.

4.1.1.36 Trivisio_DECLSPEC void colibriSetTriggerDivisor (void * *imu*, uint8_t *div*)

Set trigger status and divisor

Sets the trigger status and divisor of a Colibri.

Parameters

imu Colibri handle

div Trigger divisor, if set to 0 no trigger signal is emitted, otherwise the divisor indicates how many samples there are between emitted trigger signals.

4.1.1.37 Trivisio_DECLSPEC int colibriStart (void * *imu*)

Start measurements

Starts measurements from Colibri using the current configuration.

Parameters

imu Colibri handle.

Returns

1 on success.

4.1.1.38 Trivisio_DECLSPEC void colibriStop (void * *imu*)

Stop measurements

Stop measurements with Colibri.

Parameters

imu Colibri handle.

4.2 TrivisioColibri.h

```
00001 #pragma once
00002
00003 #include "TrivisioConfig.h"
00004 #include "TrivisioTypes.h"
00005
00006
00007 #ifdef __cplusplus
00008 extern "C" {
00009 #endif
00010
00026 Trivisio_DECLSPEC
00027 void* colibriCreate(unsigned short bufLen);
00028
00036 Trivisio_DECLSPEC
00037 void colibriDestroy(void* imu);
00038
00039
00053 Trivisio_DECLSPEC
00054 int colibriOpen(void* imu, const struct ColibriConfig* conf, const char* dev);
00055
00063 Trivisio_DECLSPEC
00064 void colibriClose(void* imu);
00065
00066
00074 Trivisio_DECLSPEC
00075 void colibriGetSensorInfo(void* imu, struct TrivisioSensor* info);
00076
00077
00086 Trivisio_DECLSPEC
00087 int colibriStart(void* imu);
00088
00095 Trivisio_DECLSPEC
00096 void colibriStop(void* imu);
00097
00098
00107 Trivisio_DECLSPEC
00108 void colibriGetID(void* imu, char* ID);
00109
00122 Trivisio_DECLSPEC
00123 void colibriGetData(void* imu, struct TrivisioIMUData* data);
00124
00125
00134 Trivisio_DECLSPEC
00135 void colibriGetConfig(void* imu, struct ColibriConfig* config);
00136
00148 Trivisio_DECLSPEC
00149 void colibriSetConfig(void* imu, const struct ColibriConfig* config);
00150
00151
00158 Trivisio_DECLSPEC
00159 int colibriGetKaStatus(void* imu);
00160
00167 Trivisio_DECLSPEC
00168 void colibriSetKaStatus(void* imu, int active);
00169
00178 Trivisio_DECLSPEC
00179 void colibriSetKa(void* imu, const float Ka[9]);
00180
00188 Trivisio_DECLSPEC
00189 void colibriGetKa(void* imu, float Ka[9]);
```

```
00190
00191
00198 Trivisio_DECLSPEC
00199 int colibriGetKgStatus(void* imu);
00200
00207 Trivisio_DECLSPEC
00208 void colibriSetKgStatus(void* imu, int active);
00209
00218 Trivisio_DECLSPEC
00219 void colibriSetKg(void* imu, const float Kg[9]);
00220
00228 Trivisio_DECLSPEC
00229 void colibriGetKg(void* imu, float Kg[9]);
00230
00231
00238 Trivisio_DECLSPEC
00239 int colibriGetKmStatus(void* imu);
00240
00247 Trivisio_DECLSPEC
00248 void colibriSetKmStatus(void* imu, int active);
00249
00258 Trivisio_DECLSPEC
00259 void colibriSetKm(void* imu, const float Km[9]);
00260
00268 Trivisio_DECLSPEC
00269 void colibriGetKg(void* imu, float Kg[9]);
00270
00271
00280 Trivisio_DECLSPEC
00281 void colibriGetKa(void* imu, float Ka[9]);
00282
00291 Trivisio_DECLSPEC
00292 void colibriGetKg(void* imu, float Kg[9]);
00293
00302 Trivisio_DECLSPEC
00303 void colibriGetKm(void* imu, float Km[9]);
00304
00311 Trivisio_DECLSPEC
00312 void colibriGetCalib(void* imu, struct TrivisioIMUCalibration* calib);
00313
00326 Trivisio_DECLSPEC
00327 void colibriSetCalib(void* imu, const struct TrivisioIMUCalibration* calib);
00328
00334 Trivisio_DECLSPEC
00335 void colibriSaveSettings(void* imu);
00336
00337
00345 Trivisio_DECLSPEC
00346 int colibriGetBoresight(void* imu);
00347
00361 Trivisio_DECLSPEC
00362 void colibriSetBoresight(void* imu, int active);
00363
00379 Trivisio_DECLSPEC
00380 void colibriBoresight(void* imu, enum ColibriBoresightType type);
00381
00382
00390 Trivisio_DECLSPEC
00391 int colibriGetJitterStatus(void* imu);
00392
00393
00407 Trivisio_DECLSPEC
```

```
00408 void colibriGetJitterParam(void* imu, float* lambda, float* th, float* tol);
00409
00410
00418 Trivisio_DECLSPEC
00419 void colibriSetJitterStatus(void* imu, int status);
00420
00433 Trivisio_DECLSPEC
00434 void colibriSetJitterParam(void* imu, float lambda, float th, float tol);
00435
00442 Trivisio_DECLSPEC
00443 uint8_t colibriGetTriggerDivisor(void* imu);
00444
00453 Trivisio_DECLSPEC
00454 void colibriSetTriggerDivisor(void* imu, uint8_t div);
00455
00456
00464 Trivisio_DECLSPEC
00465 void colibriEulerOri(const struct TrivisioIMUData* data, float euler[3]);
00466
00467
00479 Trivisio_DECLSPEC
00480 int colibriGetDeviceList(struct TrivisioSensor* list, unsigned listLen);
00481
00482
00488 Trivisio_DECLSPEC
00489 const char* colibriAPIVersion();
00490
00491 #ifdef __cplusplus
00492 }
00493 #endif
```

4.3 TrivisioConfig.h File Reference

Defines

- #define [Trivisio_DECLSPEC](#)

4.3.1 Define Documentation

4.3.1.1 #define Trivisio_DECLSPEC

Definition at line [10](#) of file [TrivisioConfig.h](#).

4.4 TrivisioConfig.h

```
00001 #pragma once
00002
00003 #if defined(WIN32) && BUILD_SHARED_LIBS
00004 #   ifdef Trivisio_EXPORTS
00005 #       define Trivisio_DECLSPEC __declspec(dllexport)
00006 #   else
00007 #       define Trivisio_DECLSPEC __declspec(dllimport)
00008 #   endif
00009 #else
00010 #   define Trivisio_DECLSPEC
00011 #endif
```

4.5 TrivisioTypes.h File Reference

```
#include "TrivisioConfig.h"  
#include <stdint.h>
```

Data Structures

- struct [ColibriConfig](#)
Colibri configuration structure.
- struct [TrivisioIMUCalibration](#)
TrivisioIMUCalibration.
- struct [TrivisioIMUData](#)
TrivisioIMUData.
- struct [TrivisioSensor](#)
Trivisio sensor structure.

Enumerations

- enum [ColibriBoresightType](#) { [COLIBRI_HEADING_RESET](#) = 1, [COLIBRI_OBJECT_RESET](#) = 2, [COLIBRI_ALIGNMENT_RESET](#) = 3 }

4.5.1 Enumeration Type Documentation

4.5.1.1 enum ColibriBoresightType

Enumerator:

COLIBRI_HEADING_RESET
COLIBRI_OBJECT_RESET Heading reset
COLIBRI_ALIGNMENT_RESET Object reset

Definition at line 37 of file [TrivisioTypes.h](#).

4.6 TrivisioTypes.h

```
00001 #pragma once
00002
00003 #include "TrivisioConfig.h"
00004
00005 #if WIN32
00006 # include "pstdint.h"
00007 #else
00008 # include <stdint.h>
00009 #endif
00010
00011 #ifdef __cplusplus
00012 extern "C" {
00013 #endif
00014
00020 struct TrivisioSensor
00021 {
00022     char dev[256];
00023     char ID[8];
00024     /* \brief Sensor type */
00025     enum SensorType {
00026         COLIBRI
00027     } type;
00028     uint16_t FWver;
00029     uint16_t FWsubver;
00030 };
00031
00032
00033 /* \brief Colibri boresighting types
00034 *
00035 * The boresighting modes that the Colibri supports.
00036 */
00037 enum ColibriBoresightType {
00038     COLIBRI_HEADING_RESET = 1,
00039     COLIBRI_OBJECT_RESET = 2,
00040     COLIBRI_ALIGNMENT_RESET = 3
00041 };
00042
00043
00048 struct ColibriConfig
00049 {
00051     enum Sensor {
00052         MAG_X = 0x0001,
00053         MAG_Y = 0x0002,
00054         MAG_Z = 0x0004,
00055         MAG     = 0x0007,
00056         ACC_X = 0x0008,
00057         ACC_Y = 0x0010,
00058         ACC_Z = 0x0020,
00059         ACC     = 0x0038,
00060         GYR_X = 0x0040,
00061         GYR_Y = 0x0080,
00062         GYR_Z = 0x0100,
00063         GYR     = 0x01C0,
00064         TEMP  = 0x0200,
00065         GARBAGE = 0x0400,
00066         ORIENT= 0x0800,
00067         ALL    = 0x0BFF
00068     } sensor;
00069     uint8_t magDiv;
00070     uint16_t freq;
```

```
00071  uint8_t  ascii:1;
00072  uint8_t  autoStart:1;
00073  uint8_t  raw:1;
00074  };
00075
00076
00081  struct TrivisioIMUData
00082  {
00083      uint32_t  t;
00084      float  mag_x;
00085      float  mag_y;
00086      float  mag_z;
00087      float  acc_x;
00088      float  acc_y;
00089      float  acc_z;
00090      float  gyr_x;
00091      float  gyr_y;
00092      float  gyr_z;
00093      float  temp;
00094      float  q_w;
00095      float  q_x;
00096      float  q_y;
00097      float  q_z;
00098  };
00099
00100
00106  struct TrivisioIMUCalibration
00107  {
00108      float  Ra[9];
00109      float  ba[3];
00110      float  ka[3];
00112      float  Rg[9];
00113      float  bg[3];
00114      float  kg[3];
00115      float  Kga[9];
00117      float  Rm[9];
00118      float  bm[3];
00119      float  km[3];
00121  };
00122
00123  #ifdef __cplusplus
00124  }
00125  #endif
```

Index

- ACC
 - ColibriConfig, 6
- ACC_X
 - ColibriConfig, 6
- ACC_Y
 - ColibriConfig, 6
- ACC_Z
 - ColibriConfig, 6
- acc_x
 - TrivisioIMUData, 10
- acc_y
 - TrivisioIMUData, 10
- acc_z
 - TrivisioIMUData, 10
- ALL
 - ColibriConfig, 6
- ascii
 - ColibriConfig, 6
- autoStart
 - ColibriConfig, 6
- ba
 - TrivisioIMUCalibration, 8
- bg
 - TrivisioIMUCalibration, 8
- bm
 - TrivisioIMUCalibration, 8
- COLIBRI
 - TrivisioSensor, 13
- COLIBRI_ALIGNMENT_RESET
 - TrivisioTypes.h, 33
- COLIBRI_HEADING_RESET
 - TrivisioTypes.h, 33
- COLIBRI_OBJECT_RESET
 - TrivisioTypes.h, 33
- colibriAPIVersion
 - TrivisioColibri.h, 16
- colibriBoresight
 - TrivisioColibri.h, 16
- ColibriBoresightType
 - TrivisioTypes.h, 33
- colibriClose
 - TrivisioColibri.h, 17
- ColibriConfig, 5
 - ACC, 6
 - ACC_X, 6
 - ACC_Y, 6
 - ACC_Z, 6
 - ALL, 6
 - ascii, 6
 - autoStart, 6
 - freq, 6
 - GARBAGE, 6
 - GYR, 6
 - GYR_X, 6
 - GYR_Y, 6
 - GYR_Z, 6
 - MAG, 6
 - MAG_X, 6
 - MAG_Y, 6
 - MAG_Z, 6
 - magDiv, 7
 - ORIENT, 6
 - raw, 7
 - Sensor, 6
 - sensor, 7
 - TEMP, 6
- colibriCreate
 - TrivisioColibri.h, 17
- colibriDestroy
 - TrivisioColibri.h, 17
- colibriEulerOri
 - TrivisioColibri.h, 17
- colibriGetBoresight
 - TrivisioColibri.h, 18
- colibriGetCalib
 - TrivisioColibri.h, 18
- colibriGetConfig
 - TrivisioColibri.h, 18
- colibriGetData
 - TrivisioColibri.h, 18

- colibriGetDeviceList
 - TrivisioColibri.h, 19
- colibriGetID
 - TrivisioColibri.h, 19
- colibriGetJitterParam
 - TrivisioColibri.h, 19
- colibriGetJitterStatus
 - TrivisioColibri.h, 20
- colibriGetKa
 - TrivisioColibri.h, 20
- colibriGetKaStatus
 - TrivisioColibri.h, 20
- colibriGetKg
 - TrivisioColibri.h, 21
- colibriGetKgStatus
 - TrivisioColibri.h, 21
- colibriGetKm
 - TrivisioColibri.h, 21
- colibriGetKmStatus
 - TrivisioColibri.h, 22
- colibriGetSensorInfo
 - TrivisioColibri.h, 22
- colibriGetTriggerDivisor
 - TrivisioColibri.h, 22
- colibriOpen
 - TrivisioColibri.h, 22
- colibriSaveSettings
 - TrivisioColibri.h, 23
- colibriSetBoresight
 - TrivisioColibri.h, 23
- colibriSetCalib
 - TrivisioColibri.h, 23
- colibriSetConfig
 - TrivisioColibri.h, 24
- colibriSetJitterParam
 - TrivisioColibri.h, 24
- colibriSetJitterStatus
 - TrivisioColibri.h, 24
- colibriSetKa
 - TrivisioColibri.h, 25
- colibriSetKaStatus
 - TrivisioColibri.h, 25
- colibriSetKg
 - TrivisioColibri.h, 25
- colibriSetKgStatus
 - TrivisioColibri.h, 25
- colibriSetKm
 - TrivisioColibri.h, 26
- colibriSetKmStatus
 - TrivisioColibri.h, 26
- colibriSetTriggerDivisor
 - TrivisioColibri.h, 26
- colibriStart
 - TrivisioColibri.h, 26
- colibriStop
 - TrivisioColibri.h, 27
- dev
 - TrivisioSensor, 13
- freq
 - ColibriConfig, 6
- FWsubver
 - TrivisioSensor, 13
- FWver
 - TrivisioSensor, 13
- GARBAGE
 - ColibriConfig, 6
- GYR
 - ColibriConfig, 6
- GYR_X
 - ColibriConfig, 6
- GYR_Y
 - ColibriConfig, 6
- GYR_Z
 - ColibriConfig, 6
- gyr_x
 - TrivisioIMUData, 11
- gyr_y
 - TrivisioIMUData, 11
- gyr_z
 - TrivisioIMUData, 11
- ID
 - TrivisioSensor, 14
- ka
 - TrivisioIMUCalibration, 8
- kg
 - TrivisioIMUCalibration, 9
- Kga
 - TrivisioIMUCalibration, 9
- km
 - TrivisioIMUCalibration, 9
- MAG
 - ColibriConfig, 6
- MAG_X
 - ColibriConfig, 6
- MAG_Y

- ColibriConfig, 6
- MAG_Z
 - ColibriConfig, 6
- mag_x
 - TrivisioIMUData, 11
- mag_y
 - TrivisioIMUData, 11
- mag_z
 - TrivisioIMUData, 11
- magDiv
 - ColibriConfig, 7
- ORIENT
 - ColibriConfig, 6
- q_w
 - TrivisioIMUData, 11
- q_x
 - TrivisioIMUData, 11
- q_y
 - TrivisioIMUData, 12
- q_z
 - TrivisioIMUData, 12
- Ra
 - TrivisioIMUCalibration, 9
- raw
 - ColibriConfig, 7
- Rg
 - TrivisioIMUCalibration, 9
- Rm
 - TrivisioIMUCalibration, 9
- Sensor
 - ColibriConfig, 6
- sensor
 - ColibriConfig, 7
- SensorType
 - TrivisioSensor, 13
- t
 - TrivisioIMUData, 12
- TEMP
 - ColibriConfig, 6
- temp
 - TrivisioIMUData, 12
- Trivisio_DECLSPEC
 - TrivisioConfig.h, 31
- TrivisioColibri.h, 15
 - colibriAPIVersion, 16
 - colibriBoresight, 16
 - colibriClose, 17
 - colibriCreate, 17
 - colibriDestroy, 17
 - colibriEulerOri, 17
 - colibriGetBoresight, 18
 - colibriGetCalib, 18
 - colibriGetConfig, 18
 - colibriGetData, 18
 - colibriGetDeviceList, 19
 - colibriGetID, 19
 - colibriGetJitterParam, 19
 - colibriGetJitterStatus, 20
 - colibriGetKa, 20
 - colibriGetKaStatus, 20
 - colibriGetKg, 21
 - colibriGetKgStatus, 21
 - colibriGetKm, 21
 - colibriGetKmStatus, 22
 - colibriGetSensorInfo, 22
 - colibriGetTriggerDivisor, 22
 - colibriOpen, 22
 - colibriSaveSettings, 23
 - colibriSetBoresight, 23
 - colibriSetCalib, 23
 - colibriSetConfig, 24
 - colibriSetJitterParam, 24
 - colibriSetJitterStatus, 24
 - colibriSetKa, 25
 - colibriSetKaStatus, 25
 - colibriSetKg, 25
 - colibriSetKgStatus, 25
 - colibriSetKm, 26
 - colibriSetKmStatus, 26
 - colibriSetTriggerDivisor, 26
 - colibriStart, 26
 - colibriStop, 27
- TrivisioConfig.h, 31
 - Trivisio_DECLSPEC, 31
- TrivisioIMUCalibration, 8
 - ba, 8
 - bg, 8
 - bm, 8
 - ka, 8
 - kg, 9
 - Kga, 9
 - km, 9
 - Ra, 9
 - Rg, 9
 - Rm, 9
- TrivisioIMUData, 10

- acc_x, [10](#)
- acc_y, [10](#)
- acc_z, [10](#)
- gyr_x, [11](#)
- gyr_y, [11](#)
- gyr_z, [11](#)
- mag_x, [11](#)
- mag_y, [11](#)
- mag_z, [11](#)
- q_w, [11](#)
- q_x, [11](#)
- q_y, [12](#)
- q_z, [12](#)
- t, [12](#)
- temp, [12](#)
- TrivisioSensor, [13](#)
 - COLIBRI, [13](#)
 - dev, [13](#)
 - FWsubver, [13](#)
 - FWver, [13](#)
 - ID, [14](#)
 - SensorType, [13](#)
 - type, [14](#)
- TrivisioTypes.h, [33](#)
 - COLIBRI_ALIGNMENT_RESET, [33](#)
 - COLIBRI_HEADING_RESET, [33](#)
 - COLIBRI_OBJECT_RESET, [33](#)
 - ColibriBoresightType, [33](#)
- type
 - TrivisioSensor, [14](#)