I2C LSM9DS1 RaspberryPI C++ Library

Generated by Doxygen 1.8.17

Chapter 1

I2C LSM9DS1 RaspberryPI C++ Library

This is a C++11 library for the LSM9DS1 on a Raspberry PI using a callback handler for the data.

`github repository`

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 AccelSettings Struct Reference

Accelerometer settings with default values.

`#include <LSM9DS1.h>`

Public Types

- enum Scale { **A_SCALE_2G** = 2, **A_SCALE_16G** = 16, **A_SCALE_4G** = 4, **A_SCALE_8G** = 8 }
  defines all possible FSR's of the accelerometer
- enum Abw {
  A_ABW_408 = 0, A_ABW_211 = 1, A_ABW_105 = 2, A_ABW_50 = 3,
  A_ABW_OFF = -1 }
  Defines all possible anti-aliasing filter rates of the accelerometer.

Public Attributes

- Scale scale = A_SCALE_16G
  accel scale (in g) can be 2, 4, 8, or 16
- uint8_t enableX = true
  Enables accelerometer's X axis.
- uint8_t enableY = true
  Enables accelerometer's Y axis.
- uint8_t enableZ = true
  Enables accelerometer's Z axis.
- Abw bandwidth = A_ABW_OFF
  Accel cutoff freqeuncy.
- uint8_t **highResEnable** = false
- uint8_t **highResBandwidth** = 0

3.1.1 Detailed Description

Accelerometer settings with default values.

3.1.2 Member Enumeration Documentation

3.1.2.1 Abw

`enum AccelSettings::Abw`

Defines all possible anti-aliasing filter rates of the accelerometer.

Enumerator

A_ABW_408	408 Hz (0x0)
A_ABW_211	211 Hz (0x1)
A_ABW_105	105 Hz (0x2)
A_ABW_50	50 Hz (0x3)
A_ABW_OFF	no cutoff

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

- LSM9DS1.h

3.2 DeviceSettings Struct Reference

Hardware related settings.

`#include <LSM9DS1.h>`

Public Attributes

- uint8_t agAddress = LSM9DS1_AG_ADDR
  I2C acceleromter address.
- uint8_t mAddress = LSM9DS1_M_ADDR
  I2C magnetometer address.
- unsigned i2c_bus = LSM9DS1_DEFAULT_I2C_BUS
  Default I2C bus number (most likely 1)
- unsigned drdy_gpio = LSM9DS1_DRDY_GPIO
  Data ready pin (INT2) of the accelerometer.
- bool initPIGPIO = true
  If set to true the pigpio library is initialised with signals disabled.

3.2.1 Detailed Description

Hardware related settings.

3.2.2 Member Data Documentation

3.2.2.1 initPIGPIO

```
bool DeviceSettings::initPIGPIO = true
```

If set to true the pigpio library is initialised with signals disabled.

You can do your own init when setting to false before calling begin().

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

- LSM9DS1.h

3.3 GyroSettings Struct Reference

Gyroscope settings with default values.

```
#include <LSM9DS1.h>
```

Public Types

- enum Scale { G_SCALE_245DPS = 245, G_SCALE_500DPS = 500, G_SCALE_2000DPS = 2000 }
  Gyro_scale defines the possible full-scale ranges of the gyroscope.
- enum SampleRate {
  G_ODR_14_9 = 1, G_ODR_59_5 = 2, G_ODR_119 = 3, G_ODR_238 = 4,
  G_ODR_476 = 5, G_ODR_952 = 6 }
  Gyro & Acc sampling rates.

Public Attributes

- Scale scale = G_SCALE_245DPS
  gyro scale can be 245, 500, or 2000
- SampleRate sampleRate = G_ODR_14_9
  Gyro & Accelerometer sample rate.
- uint8_t enableX = true
  X axis enabled.
- uint8_t enableY = true
  Y axis enabled.
- uint8_t enableZ = true
  Z axis enabled.
- uint8_t **bandwidth** = 0
- uint8_t **lowPowerEnable** = false
- uint8_t **HPFEnable** = false
- uint8_t **HPFCutoff** = 0
- uint8_t **flipX** = false
- uint8_t **flipY** = false
- uint8_t **flipZ** = false
- uint8_t **orientation** = 0
- uint8_t **latchInterrupt** = true

3.3.1 Detailed Description

Gyroscope settings with default values.

3.3.2 Member Enumeration Documentation

3.3.2.1 SampleRate

`enum GyroSettings::SampleRate`

Gyro & Acc sampling rates.

Enumerator

G_ODR_14↩_9	14.9 Hz (1)
G_ODR_59↩_5	59.5 Hz (2)
G_ODR_119	119 Hz (3)
G_ODR_238	238 Hz (4)
G_ODR_476	476 Hz (5)
G_ODR_952	952 Hz (6)

3.3.2.2 Scale

`enum GyroSettings::Scale`

Gyro_scale defines the possible full-scale ranges of the gyroscope.

Enumerator

G_SCALE_245DPS	245 degrees per second
G_SCALE_500DPS	500 dps
G_SCALE_2000DPS	2000 dps

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

- LSM9DS1.h

3.4 LSM9DS1 Class Reference

Main class for the LSM9DS1 acceleromter which manages the data acquisition via pigpio and calls the main program via a callback handler.

`#include <LSM9DS1.h>`

Public Member Functions

- LSM9DS1 (DeviceSettings deviceSettings=DeviceSettings())

  LSM9DS1 class constructor.
- void begin (GyroSettings gyroSettings=GyroSettings(), AccelSettings accelSettings=AccelSettings(), MagSettings magSettings=MagSettings(), TemperatureSettings temperatureSettings=TemperatureSettings())

  Initializes the gyro, accelerometer, magnetometer and starts the acquistion.
- void end ()

  Ends the data acquisition and closes all IO.
- void setCallback (LSM9DS1callback *cb)

  Sets the callback which receives the samples at the sampling rate.
- bool accelAvailable ()

  Polls the accelerometer status register to check if new data is available.
- bool gyroAvailable ()

  Polls the gyroscope status register to check if new data is available.
- bool tempAvailable ()

  Polls the temperature status register to check if new data is available.
- bool magAvailable (lsm9ds1_axis axis=ALL_AXIS)

  Polls the magnetometer status register to check if new data is available.
- int16_t readGyro (lsm9ds1_axis axis)

  Read a specific axis of the gyroscope.
- int16_t readAccel (lsm9ds1_axis axis)

  Read a specific axis of the accelerometer.
- int16_t readMag (lsm9ds1_axis axis)

  Read a specific axis of the magnetometer.
- void magOffset (uint8_t axis, int16_t offset)

  Sets the magnetometer offset.
- float calcGyro (int16_t gyro)

  Convert from RAW signed 16-bit value to degrees per second This function reads in a signed 16-bit value and returns the scaled DPS.
- float calcAccel (int16_t accel)

  Convert from RAW signed 16-bit value to gravity (g's).
- float calcMag (int16_t mag)

  Convert from RAW signed 16-bit value to Gauss (Gs) This function reads in a signed 16-bit value and returns the scaled Gs.
- void setGyroScale (GyroSettings::Scale gScl)

  Set the full-scale range of the gyroscope.
- void setAccelScale (AccelSettings::Scale aScl)

  Set the full-scale range of the accelerometer.
- void setMagScale (MagSettings::Scale mScl)

  Set the full-scale range of the magnetometer.
- uint8_t getGyroIntSrc ()

  Get contents of Gyroscope interrupt source register.
- uint8_t getAccelIntSrc ()

  Get contents of accelerometer interrupt source register.
- uint8_t getMagIntSrc ()

  Get contents of magnetometer interrupt source register.
- uint8_t getInactivity ()

  Get status of inactivity interrupt.
- uint8_t getFIFOSamples ()

  Get number of FIFO samples.

3.4.1 Detailed Description

Main class for the LSM9DS1 acceleromter which manages the data acquisition via pigpio and calls the main program via a callback handler.

The constructor and the begin() function have default settings so that in the simplest case just a callback needs to be registered and then begin be called. To stop the data acquistion call end().

3.4.2 Constructor & Destructor Documentation

3.4.2.1 LSM9DS1()

```
LSM9DS1::LSM9DS1 (
            DeviceSettings deviceSettings = DeviceSettings() )
```

LSM9DS1 class constructor.

Parameters

deviceSettings	is defined in DeviceSettings The deviceSettings has default values for standard wiring.

3.4.3 Member Function Documentation

3.4.3.1 accelAvailable()

```
bool LSM9DS1::accelAvailable ( )
```

Polls the accelerometer status register to check if new data is available.

Returns

true if data is available.

3.4.3.2 begin()

```
void LSM9DS1::begin (
            GyroSettings gyroSettings = GyroSettings(),
            AccelSettings accelSettings = AccelSettings(),
            MagSettings magSettings = MagSettings(),
            TemperatureSettings temperatureSettings = TemperatureSettings() )
```

Initializes the gyro, accelerometer, magnetometer and starts the acquistion.

This will set up the scale and output rate of each sensor.

Parameters

accelSettings	Accelerometer settings with default settings.
gyroSettings	Gyroscope settings with default settings.
magSettings	Magnetometer settings with default settings.
temperatureSettings	Temperature sensor settings with default settings.

3.4.3.3 calcAccel()

```
float LSM9DS1::calcAccel (
            int16_t accel )
```

Convert from RAW signed 16-bit value to gravity (g's).

This function reads in a signed 16-bit value and returns the scaled g's. This function relies on aScale and aRes being correct.

Parameters

accel	A signed 16-bit raw reading from the accelerometer.

Returns

Acceleration in m/s^2.

3.4.3.4 calcGyro()

```
float LSM9DS1::calcGyro (
            int16_t gyro )
```

Convert from RAW signed 16-bit value to degrees per second This function reads in a signed 16-bit value and returns the scaled DPS.

This function relies on gScale and gRes being correct.

Parameters

gyro	A signed 16-bit raw reading from the gyroscope.

Returns

Rotation in deg/s.

3.4.3.5 calcMag()

```
float LSM9DS1::calcMag (
            int16_t mag )
```

Convert from RAW signed 16-bit value to Gauss (Gs) This function reads in a signed 16-bit value and returns the scaled Gs.

This function relies on mScale and mRes being correct.

Parameters

mag	A signed 16-bit raw reading from the magnetometer.

Returns

Magnetic field strength in Gauss.

3.4.3.6 gyroAvailable()

```
bool LSM9DS1::gyroAvailable ( )
```

Polls the gyroscope status register to check if new data is available.

Returns

true if data is available.

3.4.3.7 magAvailable()

```
bool LSM9DS1::magAvailable (
            lsm9ds1_axis axis = ALL_AXIS )
```

Polls the magnetometer status register to check if new data is available.

Parameters

axis	can be either X_AXIS, Y_AXIS, Z_AXIS, to check for new data on one specific axis. Or ALL_AXIS (default) to check for new data on all axes.

Returns

true if data is available.

3.4.3.8 magOffset()

```
void LSM9DS1::magOffset (
            uint8_t axis,
            int16_t offset )
```

Sets the magnetometer offset.

Parameters

axis	can be any of X_AXIS, Y_AXIS, or Z_AXIS.
offset	in raw units

3.4.3.9 readAccel()

```
int16_t LSM9DS1::readAccel (
            lsm9ds1_axis axis )
```

Read a specific axis of the accelerometer.

Parameters

axis	can be any of X_AXIS, Y_AXIS, or Z_AXIS.

Returns

A 16-bit signed integer with sensor data on requested axis.

3.4.3.10 readGyro()

```
int16_t LSM9DS1::readGyro (
            lsm9ds1_axis axis )
```

Read a specific axis of the gyroscope.

Parameters

axis	can be any of X_AXIS, Y_AXIS, or Z_AXIS.

Returns

A 16-bit signed integer with sensor data on requested axis.

3.4.3.11 readMag()

```
int16_t LSM9DS1::readMag (
            lsm9ds1_axis axis )
```

Read a specific axis of the magnetometer.

Parameters

axis	can be any of X_AXIS, Y_AXIS, or Z_AXIS.

Returns

A 16-bit signed integer with sensor data on requested axis.

3.4.3.12 setAccelScale()

```
void LSM9DS1::setAccelScale (
            AccelSettings::Scale aScl )
```

Set the full-scale range of the accelerometer.

Parameters

The	desired accelerometer scale.

3.4.3.13 setCallback()

```
void LSM9DS1::setCallback (
            LSM9DS1callback * cb )  [inline]
```

Sets the callback which receives the samples at the sampling rate.

Parameters

cb	Callback interface.

3.4.3.14 setGyroScale()

```
void LSM9DS1::setGyroScale (
            GyroSettings::Scale gScl )
```

Set the full-scale range of the gyroscope.

This function can be called to set the scale of the gyroscope to 245, 500, or 200 degrees per second.

Parameters

gScl	The desired gyroscope scale.

3.4.3.15 setMagScale()

```
void LSM9DS1::setMagScale (
            MagSettings::Scale mScl )
```

Set the full-scale range of the magnetometer.

The desired magnetometer scale.

3.4.3.16 tempAvailable()

```
bool LSM9DS1::tempAvailable ( )
```

Polls the temperature status register to check if new data is available.

Returns

true if data is available.

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

- LSM9DS1.h

3.5 LSM9DS1callback Class Reference

Callback interface where the callback needs to be implemented by the host application.

```
#include <LSM9DS1.h>
```

Public Member Functions

- virtual void hasSample (LSM9DS1Sample sample)=0

  Called after a sample has arrived.

3.5.1 Detailed Description

Callback interface where the callback needs to be implemented by the host application.

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

- LSM9DS1.h

3.6 LSM9DS1Sample Struct Reference

Sample from the LSM9DS1.

`#include <LSM9DS1.h>`

Public Attributes

- float ax = 0
  X Acceleration in m/s^2.
- float ay = 0
  Y Acceleration in m/s^2.
- float az = 0
  Z Acceleration in m/s^2.
- float gx = 0
  X Rotation in deg/s.
- float gy = 0
  Y Rotation in deg/s.
- float gz = 0
  Z Rotation in deg/s.
- float mx = 0
  X Magnetic field in Gauss.
- float my = 0
  Y Magnetic field in Gauss.
- float mz = 0
  Z Magnetic field in Gauss.
- float temperature = 0
  Chip temperature.

3.6.1 Detailed Description

Sample from the LSM9DS1.

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

- LSM9DS1.h

3.7 MagSettings Struct Reference

Magnetometer settings with default values.

`#include <LSM9DS1.h>`

Public Types

- enum Scale { M_SCALE_4GS = 4, M_SCALE_8GS = 8, M_SCALE_12GS = 12, M_SCALE_16GS = 16 }

  Defines all possible FSR's of the magnetometer.
- enum SampleRate {
  M_ODR_0625, **M_ODR_125**, **M_ODR_250**, **M_ODR_5**,
  M_ODR_10, **M_ODR_20**, **M_ODR_40**, **M_ODR_80** }

  Defines all possible output data rates of the magnetometer.

Public Attributes

- uint8_t **enabled** = true
- Scale **scale** = M_SCALE_16GS
- SampleRate sampleRate = M_ODR_80

  Output data rate of the magnetometer.
- uint8_t **tempCompensationEnable** = false
- uint8_t XYPerformance = 3

  magPerformance can be any value between 0-3 0 = Low power mode 2 = high performance 1 = medium performance 3 = ultra-high performance
- uint8_t **ZPerformance** = 3
- uint8_t **lowPowerEnable** = false

3.7.1 Detailed Description

Magnetometer settings with default values.

3.7.2 Member Enumeration Documentation

3.7.2.1 Scale

`enum MagSettings::Scale`

Defines all possible FSR's of the magnetometer.

Enumerator

M_SCALE_4GS	4Gs
M_SCALE_8GS	8Gs
M_SCALE_12GS	12Gs
M_SCALE_16GS	16Gs

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

- LSM9DS1.h

3.8 TemperatureSettings Struct Reference

Temperature sensor settings.

`#include <LSM9DS1.h>`

Public Attributes

- uint8_t **enabled** = true

3.8.1 Detailed Description

Temperature sensor settings.

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

- LSM9DS1.h

Index