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. The callback handler is called at the sampling rate of the accelerometer of the LSM9DS1.

`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 Attributes

- uint8_t scale = 16

  accel scale (in g) can be 2, 4, 8, or 16
- uint8_t **enableX** = true
- uint8_t **enableY** = true
- uint8_t **enableZ** = true
- int8_t bandwidth = -1

  Accel cutoff freqeuncy can be any value between -1 - 3.
- uint8_t **highResEnable** = false
- uint8_t **highResBandwidth** = 0

3.1.1 Detailed Description

Accelerometer settings with default values.

3.1.2 Member Data Documentation

3.1.2.1 bandwidth

```
int8_t AccelSettings::bandwidth = -1
```

Accel cutoff freqeuncy can be any value between -1 - 3.

-1 = bandwidth determined by sample rate 0 = 408 Hz 2 = 105 Hz 1 = 211 Hz 3 = 50 Hz

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 Attributes

- uint16_t scale = 245

  gyro scale can be 245, 500, or 2000
- uint8_t sampleRate = 1

  gyro & accelerometer sample rate (Hz): value between 1-6 1 = 14.9 4 = 238 2 = 59.5 5 = 476 3 = 119 6 = 952
- 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 **enableX** = true
- uint8_t **enableY** = true
- uint8_t **enableZ** = true
- uint8_t **latchInterrupt** = true

3.3.1 Detailed Description

Gyroscope settings with default values.

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.
- uint16_t 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.
- uint8_t accelAvailable ()

  Polls the accelerometer status register to check if new data is available.
- uint8_t gyroAvailable ()

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

- uint8_t tempAvailable ()

  Polls the temperature status register to check if new data is available.
- uint8_t 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.
- 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()

```
uint8_t LSM9DS1::accelAvailable ( )
```

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

Output: 1 - New data available 0 - No new data available

3.4.3.2 begin()

```
uint16_t 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.

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.

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.

3.4.3.6 gyroAvailable()

```
uint8_t LSM9DS1::gyroAvailable ( )
```

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

Output: 1 - New data available 0 - No new data available

3.4.3.7 magAvailable()

```
uint8_t 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. Output: 1 - New data available 0 - No new data 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. Output: 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. Output: 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. Output: A 16-bit signed integer with sensor data on requested axis.

3.4.3.12 setCallback()

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

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

Parameters

cb	Callback interface.

3.4.3.13 tempAvailable()

```
uint8_t LSM9DS1::tempAvailable ( )
```

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

Output: 1 - New data available 0 - No new data 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.

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 Attributes

- uint8_t **enabled** = true
- uint8_t **scale** = 4
- uint8_t sampleRate = 7

  mag data rate can be 0-7 0 = 0.625 Hz 4 = 10 Hz 1 = 1.25 Hz 5 = 20 Hz 2 = 2.5 Hz 6 = 40 Hz 3 = 5 Hz 7 = 80 Hz
- 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.

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