

I2C LSM9DS1 RaspberryPI C++ Library

Generated by Doxygen 1.8.17

1 I2C LSM9DS1 RaspberryPI C++ Library	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 AccelSettings Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Data Documentation	5
3.1.2.1 bandwidth	5
3.2 DeviceSettings Struct Reference	6
3.2.1 Detailed Description	6
3.2.2 Member Data Documentation	6
3.2.2.1 initPIGPIO	6
3.3 GyroSettings Struct Reference	6
3.3.1 Detailed Description	7
3.4 LSM9DS1 Class Reference	7
3.4.1 Detailed Description	8
3.4.2 Constructor & Destructor Documentation	8
3.4.2.1 LSM9DS1()	8
3.4.3 Member Function Documentation	9
3.4.3.1 accelAvailable()	9
3.4.3.2 begin()	9
3.4.3.3 calcAccel()	9
3.4.3.4 calcGyro()	10
3.4.3.5 calcMag()	10
3.4.3.6 gyroAvailable()	11
3.4.3.7 magAvailable()	11
3.4.3.8 magOffset()	11
3.4.3.9 readAccel()	12
3.4.3.10 readGyro()	12
3.4.3.11 readMag()	12
3.4.3.12 setCallback()	13
3.4.3.13 tempAvailable()	13
3.5 LSM9DS1callback Class Reference	13
3.5.1 Detailed Description	14
3.6 LSM9DS1Sample Struct Reference	14
3.6.1 Detailed Description	14
3.7 MagSettings Struct Reference	15
3.7.1 Detailed Description	15
3.8 TemperatureSettings Struct Reference	15
3.8.1 Detailed Description	15

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:

AccelSettings	
Accelerometer settings with default values	5
DeviceSettings	
Hardware related settings	6
GyroSettings	
Gyroscope settings with default values	6
LSM9DS1	
Main class for the LSM9DS1 accelerometer which manages the data acquisition via pigpio and calls the main program via a callback handler	7
LSM9DS1callback	
Callback interface where the callback needs to be implemented by the host application	13
LSM9DS1Sample	
Sample from the LSM9DS1	14
MagSettings	
Magnetometer settings with default values	15
TemperatureSettings	
Temperature sensor settings	15

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 frequency 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 frequency 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 accelerometer 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](#) accelerometer 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 acquisition.
- 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.
- 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](#) accelerometer 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 acquisition call [end\(\)](#).

3.4.2 Constructor & Destructor Documentation

3.4.2.1 LSM9DS1()

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

[LSM9DS1](#) class constructor.

Parameters

<i>deviceSettings</i>	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 acquisition.

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

Parameters

<i>accelSettings</i>	Accelerometer settings with default settings.
<i>gyroSettings</i>	Gyroscope settings with default settings.
<i>magSettings</i>	Magnetometer settings with default settings.
<i>temperatureSettings</i>	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

<i>accel</i>	A signed 16-bit raw reading from the accelerometer.
--------------	---

Returns

Acceleration in m/s².

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

<i>gyro</i>	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

<i>mag</i>	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 (
    lsm9dsl_axis axis = ALL_AXIS )
```

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

Parameters

<i>axis</i>	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

<i>axis</i>	can be any of X_AXIS, Y_AXIS, or Z_AXIS.
<i>offset</i>	in raw units

3.4.3.9 readAccel()

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

Read a specific axis of the accelerometer.

Parameters

<i>axis</i>	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

<i>axis</i>	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

<i>axis</i>	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 setCallback()

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

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

Parameters

<i>cb</i>	Callback interface.
-----------	---------------------

3.4.3.13 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/s².
- float [ay](#) = 0
Y Acceleration in m/s².
- float [az](#) = 0
Z Acceleration in m/s².
- 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 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

- accelAvailable
 - LSM9DS1, [9](#)
- AccelSettings, [5](#)
 - bandwidth, [5](#)
- bandwidth
 - AccelSettings, [5](#)
- begin
 - LSM9DS1, [9](#)
- calcAccel
 - LSM9DS1, [9](#)
- calcGyro
 - LSM9DS1, [10](#)
- calcMag
 - LSM9DS1, [10](#)
- DeviceSettings, [6](#)
 - initPIGPIO, [6](#)
- gyroAvailable
 - LSM9DS1, [11](#)
- GyroSettings, [6](#)
- initPIGPIO
 - DeviceSettings, [6](#)
- LSM9DS1, [7](#)
 - accelAvailable, [9](#)
 - begin, [9](#)
 - calcAccel, [9](#)
 - calcGyro, [10](#)
 - calcMag, [10](#)
 - gyroAvailable, [11](#)
 - LSM9DS1, [8](#)
 - magAvailable, [11](#)
 - magOffset, [11](#)
 - readAccel, [12](#)
 - readGyro, [12](#)
 - readMag, [12](#)
 - setCallback, [13](#)
 - tempAvailable, [13](#)
- LSM9DS1callback, [13](#)
- LSM9DS1Sample, [14](#)
- magAvailable
 - LSM9DS1, [11](#)
- magOffset
 - LSM9DS1, [11](#)
- MagSettings, [15](#)
- readAccel
 - LSM9DS1, [12](#)
- readGyro
 - LSM9DS1, [12](#)
- readMag
 - LSM9DS1, [12](#)
- setCallback
 - LSM9DS1, [13](#)
- tempAvailable
 - LSM9DS1, [13](#)
- TemperatureSettings, [15](#)