

## I2C LSM9DS1 RaspberryPI C++ Library

Generated by Doxygen 1.8.17



<b>1 I2C LSM9DS1 RaspberryPI C++ Library</b>	<b>1</b>
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 AccelSettings Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Enumeration Documentation	6
3.1.2.1 Abw	6
3.2 DeviceSettings Struct Reference	6
3.2.1 Detailed Description	6
3.2.2 Member Data Documentation	7
3.2.2.1 initPIGPIO	7
3.3 GyroSettings Struct Reference	7
3.3.1 Detailed Description	8
3.3.2 Member Enumeration Documentation	8
3.3.2.1 SampleRate	8
3.3.2.2 Scale	8
3.4 LSM9DS1 Class Reference	8
3.4.1 Detailed Description	10
3.4.2 Constructor & Destructor Documentation	10
3.4.2.1 LSM9DS1()	10
3.4.3 Member Function Documentation	10
3.4.3.1 accelAvailable()	10
3.4.3.2 begin()	10
3.4.3.3 calcAccel()	11
3.4.3.4 calcGyro()	11
3.4.3.5 calcMag()	12
3.4.3.6 gyroAvailable()	12
3.4.3.7 magAvailable()	12
3.4.3.8 magOffset()	13
3.4.3.9 readAccel()	13
3.4.3.10 readGyro()	13
3.4.3.11 readMag()	14
3.4.3.12 setAccelScale()	14
3.4.3.13 setCallback()	14
3.4.3.14 setGyroScale()	14
3.4.3.15 setMagScale()	15
3.4.3.16 tempAvailable()	15
3.5 LSM9DS1callback Class Reference	15
3.5.1 Detailed Description	16
3.6 LSM9DS1Sample Struct Reference	16

3.6.1 Detailed Description . . . . .	16
3.7 MagSettings Struct Reference . . . . .	17
3.7.1 Detailed Description . . . . .	17
3.7.2 Member Enumeration Documentation . . . . .	17
3.7.2.1 Scale . . . . .	17
3.8 TemperatureSettings Struct Reference . . . . .	18
3.8.1 Detailed Description . . . . .	18
<b>Index</b>	<b>19</b>

## 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:

<a href="#">AccelSettings</a>	
Accelerometer settings with default values . . . . .	5
<a href="#">DeviceSettings</a>	
Hardware related settings . . . . .	6
<a href="#">GyroSettings</a>	
Gyroscope settings with default values . . . . .	7
<a href="#">LSM9DS1</a>	
Main class for the <a href="#">LSM9DS1</a> accelerometer which manages the data acquisition via pigpio and calls the main program via a callback handler . . . . .	8
<a href="#">LSM9DS1callback</a>	
Callback interface where the callback needs to be implemented by the host application . . . . .	15
<a href="#">LSM9DS1Sample</a>	
Sample from the <a href="#">LSM9DS1</a> . . . . .	16
<a href="#">MagSettings</a>	
Magnetometer settings with default values . . . . .	17
<a href="#">TemperatureSettings</a>	
Temperature sensor settings . . . . .	18





## 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 frequency.*
- 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 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 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](#) 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.*
- void [setGyroScale](#) ([GyroSettings::Scale](#) gScI)  
*Set the full-scale range of the gyroscope.*
- void [setAccelScale](#) ([AccelSettings::Scale](#) aScI)  
*Set the full-scale range of the accelerometer.*
- void [setMagScale](#) ([MagSettings::Scale](#) mScI)  
*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](#) 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 <a href="#">DeviceSettings</a> 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/s<sup>2</sup>.

**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 (
    lsm9ds1_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 (
    lsm9dsl_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 (
    lsm9dsl_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 (
    lsm9dsl_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 setAccelScale()

```
void LSM9DS1::setAccelScale (
    AccelSettings::Scale aSc1 )
```

Set the full-scale range of the accelerometer.

#### Parameters

<i>The</i>	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

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

### 3.4.3.14 setGyroScale()

```
void LSM9DS1::setGyroScale (
    GyroSettings::Scale gSc1 )
```

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

<i>gScI</i>	The desired gyroscope scale.
-------------	------------------------------

#### 3.4.3.15 setMagScale()

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

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<sup>2</sup>.*
- float [ay](#) = 0  
*Y Acceleration in m/s<sup>2</sup>.*
- float [az](#) = 0  
*Z Acceleration in m/s<sup>2</sup>.*
- 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

<a href="#">M_SCALE_4GS</a>	4Gs
<a href="#">M_SCALE_8GS</a>	8Gs
<a href="#">M_SCALE_12GS</a>	12Gs
<a href="#">M_SCALE_16GS</a>	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

A\_ABW\_105  
    AccelSettings, 6  
A\_ABW\_211  
    AccelSettings, 6  
A\_ABW\_408  
    AccelSettings, 6  
A\_ABW\_50  
    AccelSettings, 6  
A\_ABW\_OFF  
    AccelSettings, 6  
Abw  
    AccelSettings, 6  
accelAvailable  
    LSM9DS1, 10  
AccelSettings, 5  
    A\_ABW\_105, 6  
    A\_ABW\_211, 6  
    A\_ABW\_408, 6  
    A\_ABW\_50, 6  
    A\_ABW\_OFF, 6  
    Abw, 6  
  
begin  
    LSM9DS1, 10  
  
calcAccel  
    LSM9DS1, 11  
calcGyro  
    LSM9DS1, 11  
calcMag  
    LSM9DS1, 11  
  
DeviceSettings, 6  
    initPIGPIO, 7  
  
G\_ODR\_119  
    GyroSettings, 8  
G\_ODR\_14\_9  
    GyroSettings, 8  
G\_ODR\_238  
    GyroSettings, 8  
G\_ODR\_476  
    GyroSettings, 8  
G\_ODR\_59\_5  
    GyroSettings, 8  
G\_ODR\_952  
    GyroSettings, 8  
G\_SCALE\_2000DPS  
    GyroSettings, 8  
G\_SCALE\_245DPS

    GyroSettings, 8  
G\_SCALE\_500DPS  
    GyroSettings, 8  
gyroAvailable  
    LSM9DS1, 12  
GyroSettings, 7  
    G\_ODR\_119, 8  
    G\_ODR\_14\_9, 8  
    G\_ODR\_238, 8  
    G\_ODR\_476, 8  
    G\_ODR\_59\_5, 8  
    G\_ODR\_952, 8  
    G\_SCALE\_2000DPS, 8  
    G\_SCALE\_245DPS, 8  
    G\_SCALE\_500DPS, 8  
    SampleRate, 8  
    Scale, 8  
  
initPIGPIO  
    DeviceSettings, 7  
  
LSM9DS1, 8  
    accelAvailable, 10  
    begin, 10  
    calcAccel, 11  
    calcGyro, 11  
    calcMag, 11  
    gyroAvailable, 12  
    LSM9DS1, 10  
    magAvailable, 12  
    magOffset, 12  
    readAccel, 13  
    readGyro, 13  
    readMag, 13  
    setAccelScale, 14  
    setCallback, 14  
    setGyroScale, 14  
    setMagScale, 15  
    tempAvailable, 15  
LSM9DS1callback, 15  
LSM9DS1Sample, 16  
  
M\_SCALE\_12GS  
    MagSettings, 17  
M\_SCALE\_16GS  
    MagSettings, 17  
M\_SCALE\_4GS  
    MagSettings, 17  
M\_SCALE\_8GS  
    MagSettings, 17

- magAvailable
  - LSM9DS1, [12](#)
- magOffset
  - LSM9DS1, [12](#)
- MagSettings, [17](#)
  - M\_SCALE\_12GS, [17](#)
  - M\_SCALE\_16GS, [17](#)
  - M\_SCALE\_4GS, [17](#)
  - M\_SCALE\_8GS, [17](#)
  - Scale, [17](#)
- readAccel
  - LSM9DS1, [13](#)
- readGyro
  - LSM9DS1, [13](#)
- readMag
  - LSM9DS1, [13](#)
- SampleRate
  - GyroSettings, [8](#)
- Scale
  - GyroSettings, [8](#)
  - MagSettings, [17](#)
- setAccelScale
  - LSM9DS1, [14](#)
- setCallback
  - LSM9DS1, [14](#)
- setGyroScale
  - LSM9DS1, [14](#)
- setMagScale
  - LSM9DS1, [15](#)
- tempAvailable
  - LSM9DS1, [15](#)
- TemperatureSettings, [18](#)