


Accessing Accelerometer Data on Nano 33 BLE Sense

Learn how to measure the relative position of the Nano 33 BLE Sense through the LSM9DS1 IMU sensor.

 AUTHOR: **Nefeli Alushi**

This tutorial will focus on the 3-axis accelerometer sensor of the **LSM9DS1** module on the Arduino Nano 33 BLE Sense, in order to measure the relative position of the board. This will be achieved by utilizing the values of the accelerometer's axes and later print the return values through the Arduino IDE Serial Monitor.

Goals

The goals of this project are:

- Understand what an LSM9DS1 module is.
- Use the LSM9DS1 library.
- Read the raw data of the accelerometer sensor.
- Convert the raw data into board positions.
- Print out live data through the Serial Monitor.

Hardware & Software Needed


- This project uses no external sensors or components.
- In this tutorial we will use the Arduino Create Web Editor to program the board.

The LSM9DS1 Inertial Module

IMU stands for: inertial measurement unit. It is an electronic device that measures and reports a body's specific force, angular rate and the orientation of the body, using a combination of accelerometers, gyroscopes, and oftentimes magnetometers. In this tutorial we will learn about the LSM9DS1 IMU module, which is included in the Arduino Nano 33 BLE Sense Board.

Tutorial Toolbox

Hardware

 Nano 33 BLE Sense

Software

Libraries

Missing something?

Check out our store and get what you need to follow this tutorial.

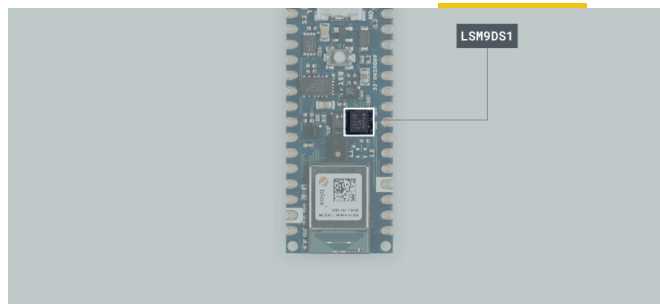
VISIT OUR STORE

Suggest Changes

The content on docs.arduino.cc is facilitated through a public [GitHub repository](#). You can read more on how to contribute in the [contribution policy](#).

EDIT THIS PAGE

The Arduino documentation is licensed under the [Creative Commons Attribution-Share Alike 4.0](#) license.



The LSM9DS1 sensor.

The LSM9DS1 is a system-in-package featuring a 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor.

The LSM9DS1 Library

The Arduino LSM9DS1 library allows us to use the Arduino Nano 33 BLE IMU module without having to go into complicated programming. The library takes care of the sensor initialization and sets its values as follows:

Accelerometer range is set at $[-4, +4]g$ ± 0.122 mg.

Gyroscope range is set at $[-2000, +2000]$ dps ± 70 mdps.

Magnetometer range is set at $[-400, +400]$ uT ± 0.014 uT.

Accelerometer Output data rate is fixed at 104 Hz.

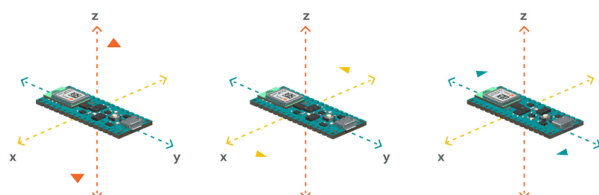
Gyroscope Output data rate is fixed at 104 Hz.

Magnetometer Output data rate is fixed at 20 Hz.

If you want to read more about the LSM9DS1 sensor module see [here](#).

Accelerometer

An accelerometer is an electromechanical device used to measure acceleration forces. Such forces may be static, like the continuous force of gravity or, as is the case with many mobile devices, dynamic to sense movement or vibrations.



How the accelerometer works.

In this example, we will use the accelerometer as a "level" that will provide information about the position of the board. With this application we will be able to read what the relative position of the board is as well as the degrees, by tilting the board up, down, left or right.

Creating the Program

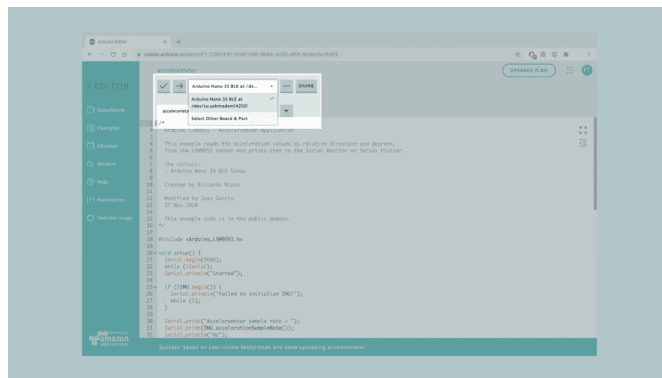
SimpleAccelerometer sketch and once it opens, rename it as **Accelerometer**.



Finding the library in the Web Editor.

2. Connecting the board

Now, connect the Arduino Nano 33 BLE Sense to the computer and make sure that the Web Editor recognizes it, if so, the board and port should appear as shown in the image below. If they don't appear, follow the [instructions](#) to install the plugin that will allow the Editor to recognize your board.



Selecting the board.

3. Printing the relative position

Now we will need to modify the code on the example, in order to print the relative position of the board as we move it in different angles.

Let's start by initializing the the x, y, z axes as `float` data types, and the `int` `degreesX = 0;` and `int` `degreesY = 0;` variables before the `setup()` .

In the `setup()` we should **remove** the following lines of code:

COPY

```
1 Serial.println();
2 Serial.println("Acceleration in G's");
3 Serial.println("X\tY\tZ");
```

Since the raw values of the three axes will not be required, we can remove the lines which will print these. Similarly, we should **remove** the following lines from the `loop()` :

```

2 Serial.print("\t");
3 Serial.print(y);
4 Serial.print('\t');
5 Serial.println(z);

```

Instead, in the `loop()` we tell the sensor to begin reading the values for the three axes. In this example we will not be using the readings from the Z axis as it is not required for this application to function, therefore you could remove it.

After the `IMU.readAcceleration` initialization, we add four `if` statements for the board's different positions. The statements will calculate the direction in which the board will be tilting towards, as well as provide the axe's degree values.

COPY

```

1 if(x > 0.1){
2   x = 100*x;
3   degreesX = map(x, 0, 97, 0, 90);
4   Serial.print("Tilting up ");
5   Serial.print(degreesX);
6   Serial.println(" degrees");
7 }
8 if(x < -0.1){
9   x = 100*x;
10  degreesX = map(x, 0, -100, 0, 90);
11  Serial.print("Tilting down ");
12  Serial.print(degreesX);
13  Serial.println(" degrees");
14 }
15 if(y > 0.1){
16   y = 100*y;
17   degreesY = map(y, 0, 97, 0, 90);
18   Serial.print("Tilting left ");
19   Serial.print(degreesY);
20   Serial.println(" degrees");
21 }
22 if(y < -0.1){
23   y = 100*y;
24   degreesY = map(y, 0, -100, 0, 90);
25   Serial.print("Tilting right ");
26   Serial.print(degreesY);
27   Serial.println(" degrees");
28 }

```

Lastly, we `Serial.print` the results value and add a `delay(1000);` .

Note: For the following code to properly work, the board's facing direction and inclination during the initialization of the code, need to be specific. More information will be shared on the "testing it out" section.

4. Complete code

If you choose to skip the code building section, the complete code can be found below:

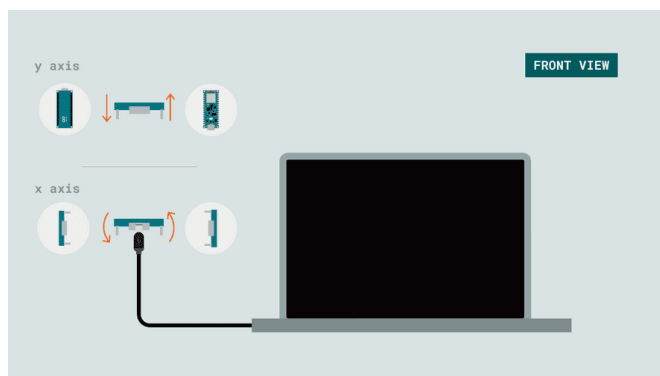
```

2  Arduino LSM9DS1 - Accelerometer Application
3
4  This example reads the acceleration values as relative to the
5  from the LSM9DS1 sensor and prints them to the Serial Monitor.
6
7  The circuit:
8  - Arduino Nano 33 BLE Sense
9
10 Created by Riccardo Rizzo
11
12 Modified by Jose García
13 27 Nov 2020
14
15 This example code is in the public domain.
16 */
17
18 #include <Arduino_LSM9DS1.h>
19
20 float x, y, z;
21 int degreesX = 0;
22 int degreesY = 0;
23
24 void setup() {
25   Serial.begin(9600);
26   while (!Serial);
27   Serial.println("Started");
28
29   if (!IMU.begin()) {
30     Serial.println("Failed to initialize IMU!");
31     while (1);
32   }
33
34   Serial.print("Accelerometer sample rate = ");

```

Testing It Out

In order to get a correct reading of the board data, before uploading the sketch to the board hold the board in your hand, from the side of the USB port. The board should be facing up and "pointing" away from you. The image below illustrates the board's position and how it works:

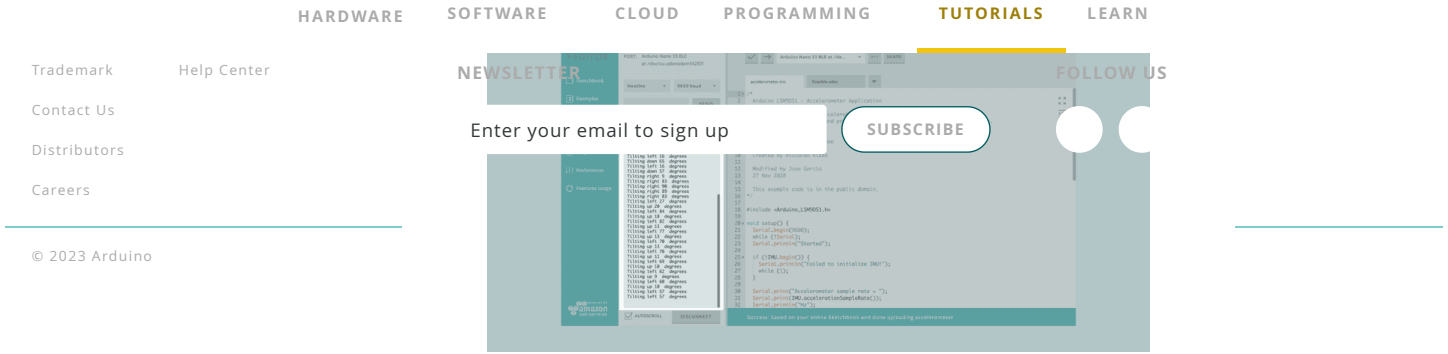


Interacting with the X and Y axes.

Now, you can verify and upload the sketch to the board and open the Monitor from the menu on the left.

If you tilt the board upwards, downwards, right or left, you will see the results printing every second according to the direction of your movement!

Here is a screenshot of the sketch returning these values:



Printing out the "tilt condition" of the board.

Troubleshoot

Sometimes errors occur, if the code is not working there are some common issues we can troubleshoot:

- Missing a bracket or a semicolon.
- Arduino board connected to the wrong port.
- Accidental interruption of cable connection.
- The initial position of the board is not as instructed. In this case you can refresh the page and try again.

Conclusion

In this simple tutorial we learned what an IMU sensor module is, how to use the **LSM9DS1** library, and how to use an Arduino Nano 33 BLE Sense to get data. Furthermore, we utilized the 3-axis accelerometer sensor, in order to measure and print out the degrees and relative position of the board.