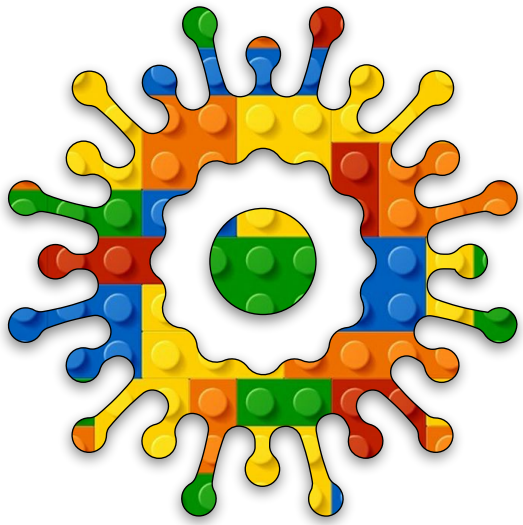
Internet of Things Module A Unit #10 IMU



Content

Module A Unit #10 IMU

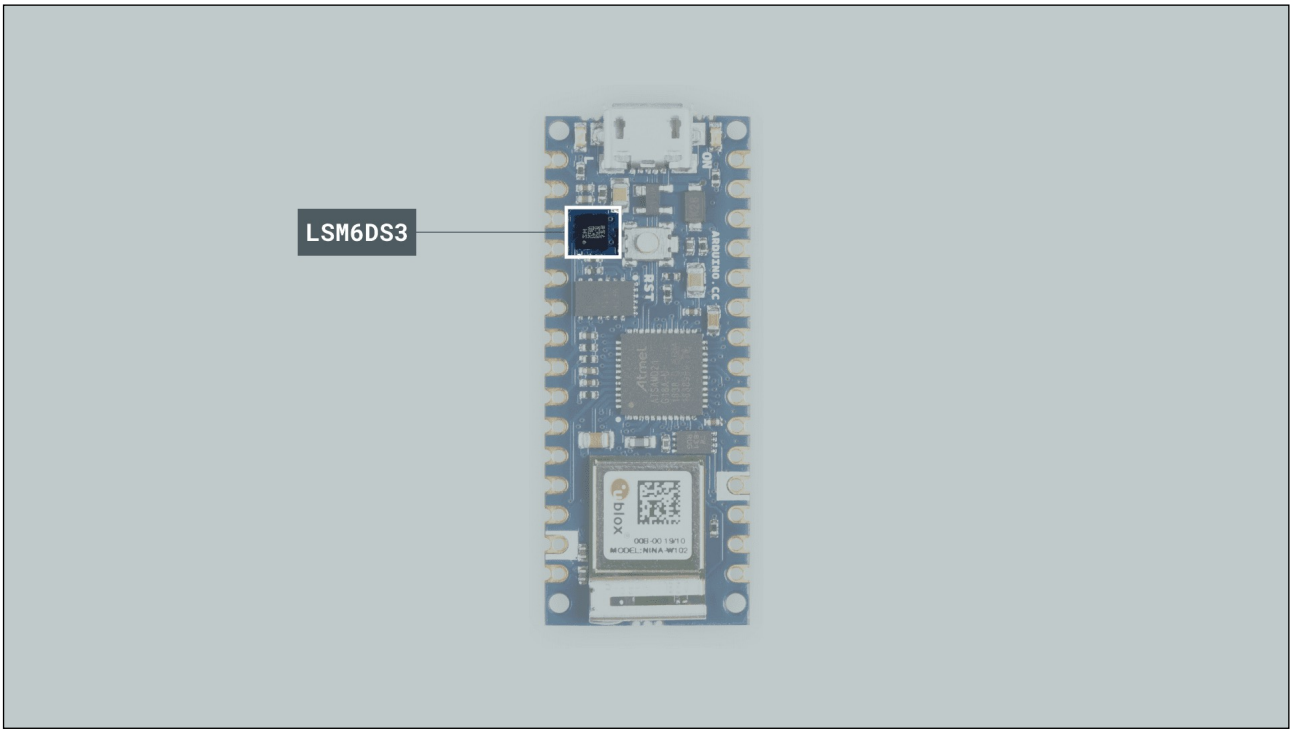
Introduction to the IMU sensor LSM6DS3

The Arduino Nano 33 IoT has a built-in sensor called an IMU.

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 LSM6DS3 IMU module, which is included in the Arduino Nano 33 IoT Board.

The LSM6DS3 is a system-in-package featuring a 3D digital linear acceleration sensor and a 3D digital angular rate sensor. In other words, an accelerometer and a gyroscope, which gives it six axes of measurement.

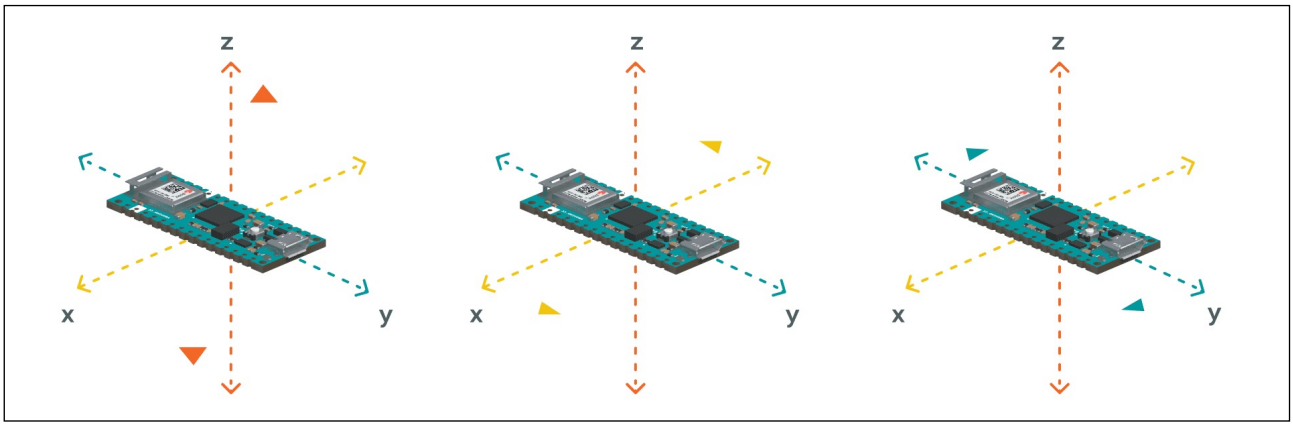
Figure 1: LSM6DS3 IMU module



Accelerometer & Gyroscope

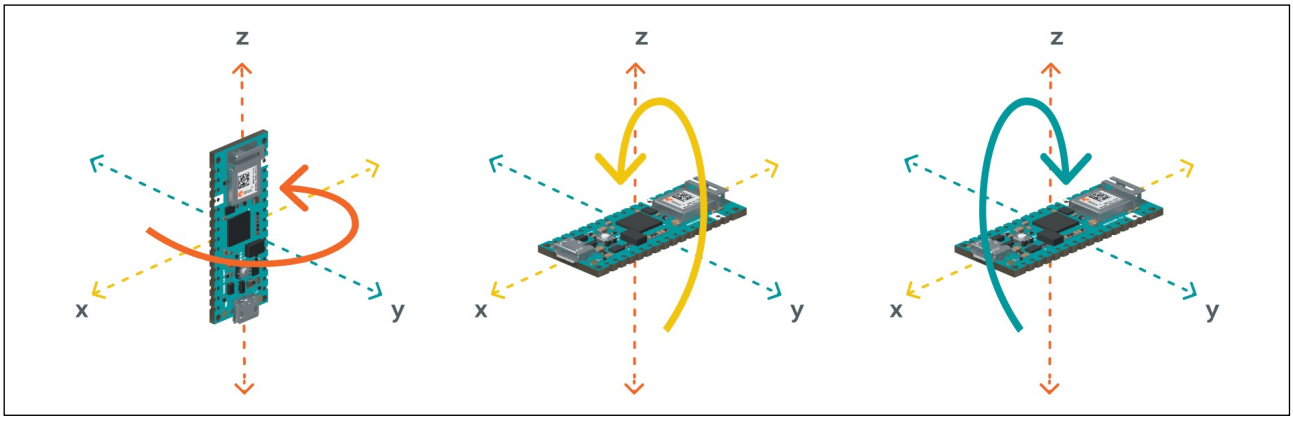
The accelerometer measures horizontal and vertical movement.

Figure 2: accelerometer



The gyroscope measures rotational movement.

Figure 3: gyroscope

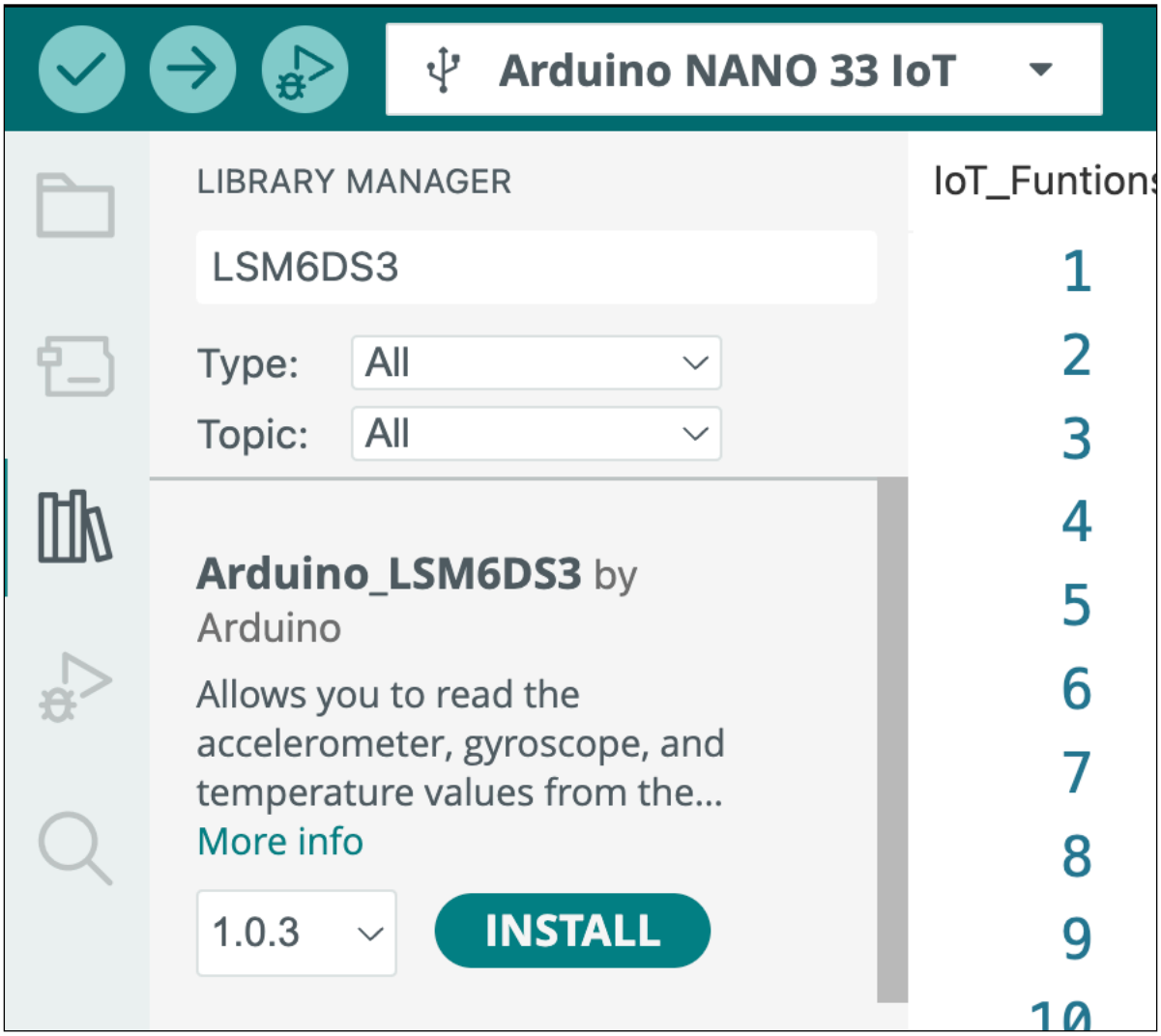


Installing the library

Before we do anything, we need to upload the library so we can use the built-in sensor LSM6DS3 module. Go to the IDE and click on the bookcase (library) icon on the left. This is the LIBRARY MANAGER. When there, type in LSM6DS3 in the search box, and it just pops up at the top. Click INSTALL.

If, for some reason, it is already installed, make sure it is the latest version. Next, we can upload a sketch to read the sensor module.

Figure 1: library manager



Sketch A10.1 accelerometer basics

This is the accelerometer code for the sensor. It prints out the values for the `x`, `y` and `z` axes. If you move the device around, tilt and rotate it, you will see that the values are between `-1` and `1` on the `Serial Monitor`. I have added a bit of a `delay()` to slow down the data sent to the console.

```
#include <Arduino_LSM6DS3.h>

void setup()
{
  Serial.begin(9600);
  while (!Serial);
  if (!IMU.begin())
  {
    Serial.println("Failed to initialize IMU!");
    while (1);
  }
}

void loop()
{
  float x, y, z;
  if (IMU.accelerationAvailable())
  {
    IMU.readAcceleration(x, y, z);
    Serial.print(x);
    Serial.print('\t');
    Serial.print(y);
    Serial.print('\t');
    Serial.println(z);
  }
  delay(100);
```

```
}
```

🗒️ Notes

Now look at the `Serial Plotter` to appreciate how the data...

🛠️ Code Explanation

`#include <Arduino_LSM6DS3.h>`	This is the library which has to have been already installed
`while (!Serial);`	Just waits for the serial port to open
`if (!IMU.begin())`	Checks to see if it hasn't started
`while (1);`	This creates a continuous loop until the IMU has kicked in
`if (IMU.accelerationAvailable())`	Checks to see if the data is available
`IMU.readAcceleration(x, y, z);`	Reads the data for x, y and z axis
`Serial.print('\t');`	Inserts a tab space

Figure A10.1a: serial monitor

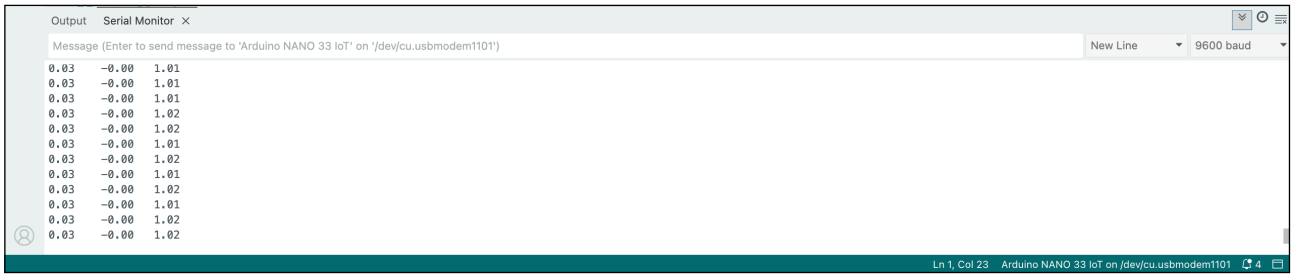
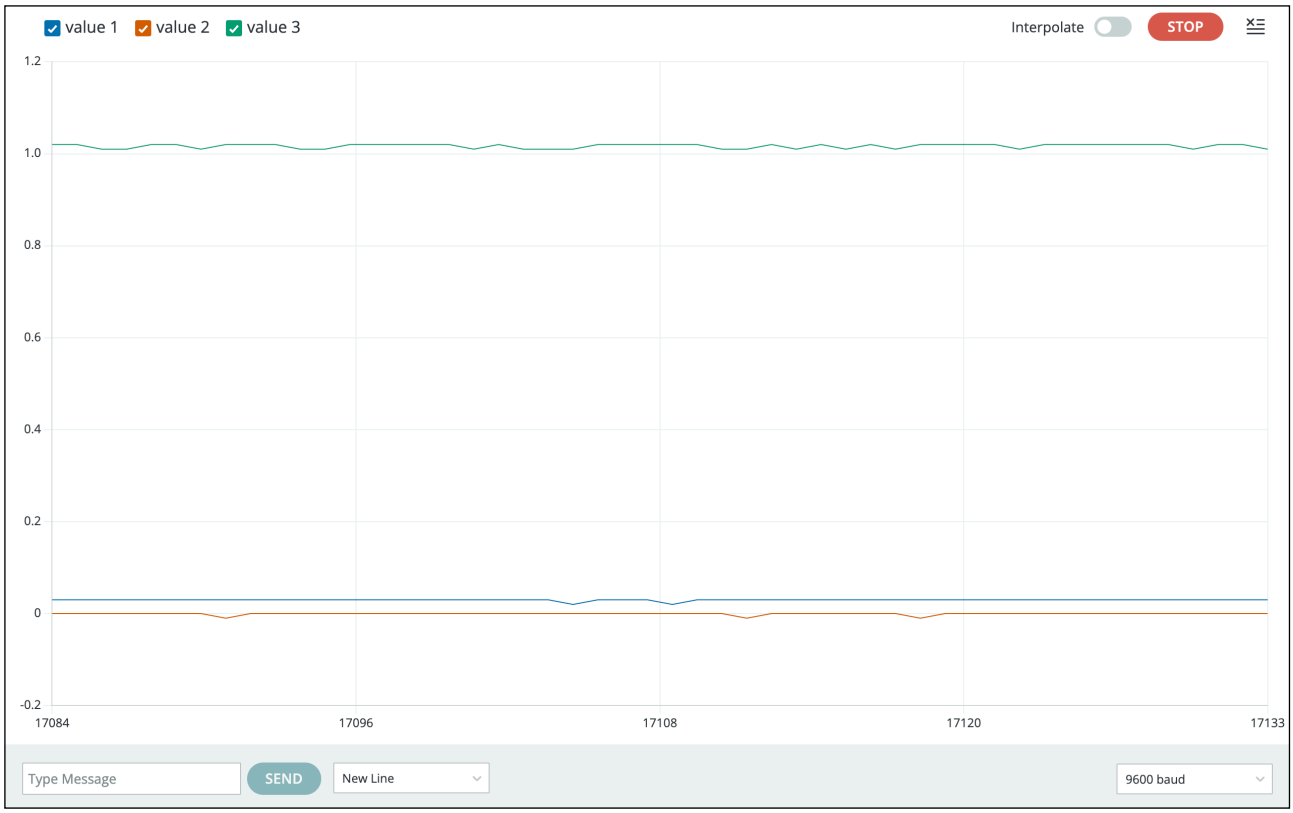
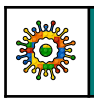


Figure A10.1b: serial plotter





Sketch A10.2 gyroscope basics

This is almost identical to the code for the accelerometer. Rotate the device in different directions.

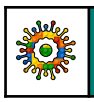
```
#include <Arduino_LSM6DS3.h>

void setup()
{
  Serial.begin(9600);
  while (!Serial);
  if (!IMU.begin())
  {
    Serial.println("Failed to initialize IMU!");
    while (1);
  }
}

void loop()
{
  float x; float y; float z;
  if (IMU.gyroscopeAvailable())
  {
    IMU.readGyroscope(x, y, z);
    Serial.print(x);
    Serial.print('\t');
    Serial.print(y);
    Serial.print('\t');
    Serial.println(z);
    delay(100);
  }
}
```

🗒️ Notes

The gyroscope measures the rotational acceleration; the values range from -500 to 500. It does not measure the angle. For that, you can use the accelerometer, as shown in the next sketch.



Sketch A10.3 degrees of tilt

We have calculated the degree of tilt from the data values.

```
#include <Arduino_LSM6DS3.h>

int degreesX = 0;
int degreesY = 0;

void setup()
{
  Serial.begin(9600);
  while (!Serial);
  if (!IMU.begin())
  {
    Serial.println("Failed to initialize IMU!");
    while (1);
  }
}

void loop()
{
  float x; float y; float z;
  if (IMU.accelerationAvailable())
  {
    IMU.readAcceleration(x, y, z);
  }

  if (x > 0.1)
  {
    x = 100 * x;
    degreesX = map(x, 0, 97, 0, 90);
```

```
  Serial.print("Tilting up ");
  Serial.print(degreesX);
  Serial.println("  degrees");
}

if (x < -0.1)
{
  x = 100 * x;
  degreesX = map(x, 0, -100, 0, 90);
  Serial.print("Tilting down ");
  Serial.print(degreesX);
  Serial.println("  degrees");
}

if (y > 0.1)
{
  y = 100 * y;
  degreesY = map(y, 0, 97, 0, 90);
  Serial.print("Tilting left ");
  Serial.print(degreesY);
  Serial.println("  degrees");
}

if (y < -0.1)
{
  y = 100 * y;
  degreesY = map(y, 0, -100, 0, 90);
  Serial.print("Tilting right ");
  Serial.print(degreesY);
  Serial.println("  degrees");
}
```

```
  if (x < 0.1 && x > -0.1 && y < 0.1 && y > -0.1)
  {
    Serial.println("level");
  }
  delay(1000);
}
```

🗒️ Notes

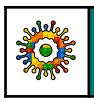
We get a description of the angle, in which direction, and by how many degrees.

🛠️ Code Explanation

`if (x < 0.1 && x > -0.1 && y < 0.1 && y > -0.1)`	Lots of conditional statements, in other words anything other than a tilt.

Figure A10.3


Arduino NANO 33 IoT
IoT_Funtions.ino
degreesY = map(y, 0, 97, 0, 90);
Serial.print("Tilting left ");
Serial.print(degreesY);
Serial.println(" degrees");
}
if (y < -0.1)
{
y = 100 * y;
degreesY = map(y, 0, -100, 0, 90);
Serial.print("Tilting right ");
Serial.print(degreesY);
Serial.println(" degrees");
}
if (x < 0.1 && x > -0.1 && y < 0.1 && y > -0.1)
{
Serial.println("level");
}
delay(1000);
}
Output
Serial Monitor
Message (Enter to send message to 'Arduino NANO 33 IoT' on '/dev/cu.usbmodem1101')
New Line
9600 baud
Tilting right 68 degrees
Tilting right 70 degrees
Tilting left 65 degrees
Tilting left 90 degrees
Tilting up 70 degrees
Tilting up 90 degrees
Tilting up 90 degrees
Tilting down 82 degrees
Tilting left 12 degrees
Tilting down 86 degrees
Tilting left 20 degrees
level
Building sketch
Ln 68, Col 1
Arduino NANO 33 IoT on /dev/cu.usbmodem1101



Sketch A10.4 LED tilt warning

Add the LED traffic lights if you have removed them. We are going to light up the red LED when the tilt is beyond 45°.

```
#include <Arduino_LSM6DS3.h>

void setup()
{
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  while (!Serial);
  if (!IMU.begin())
  {
    Serial.println("Failed to initialize IMU!");
    while (1);
  }
}

void loop()
{
  float x; float y; float z;
  if (IMU.accelerationAvailable())
  {
    IMU.readAcceleration(x, y, z);
  }

  if (x > 0.5 || x < -0.5 || y > 0.5 || y < -0.5)
  {
    digitalWrite(2, HIGH);
  }
  else
  {
```

```
    digitalWrite(2, LOW);
  }
}
```

🗒️ Notes

Whichever direction you tilt it, you should see the red LED light up when it is around 45°.

🌻 Challenges

1. Add the other two LEDs for more gradual warning.
2. Have the brightness of the LED linked to the rotational acceleration (gyroscope).