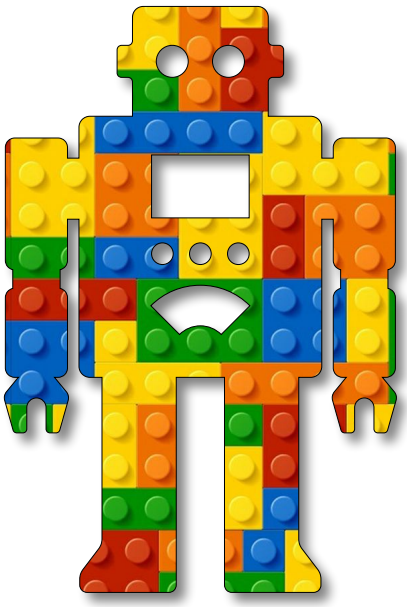
Intelligent Machines Module A Unit #11 Gyroscope



Content

Module A Unit #11 Gyroscope

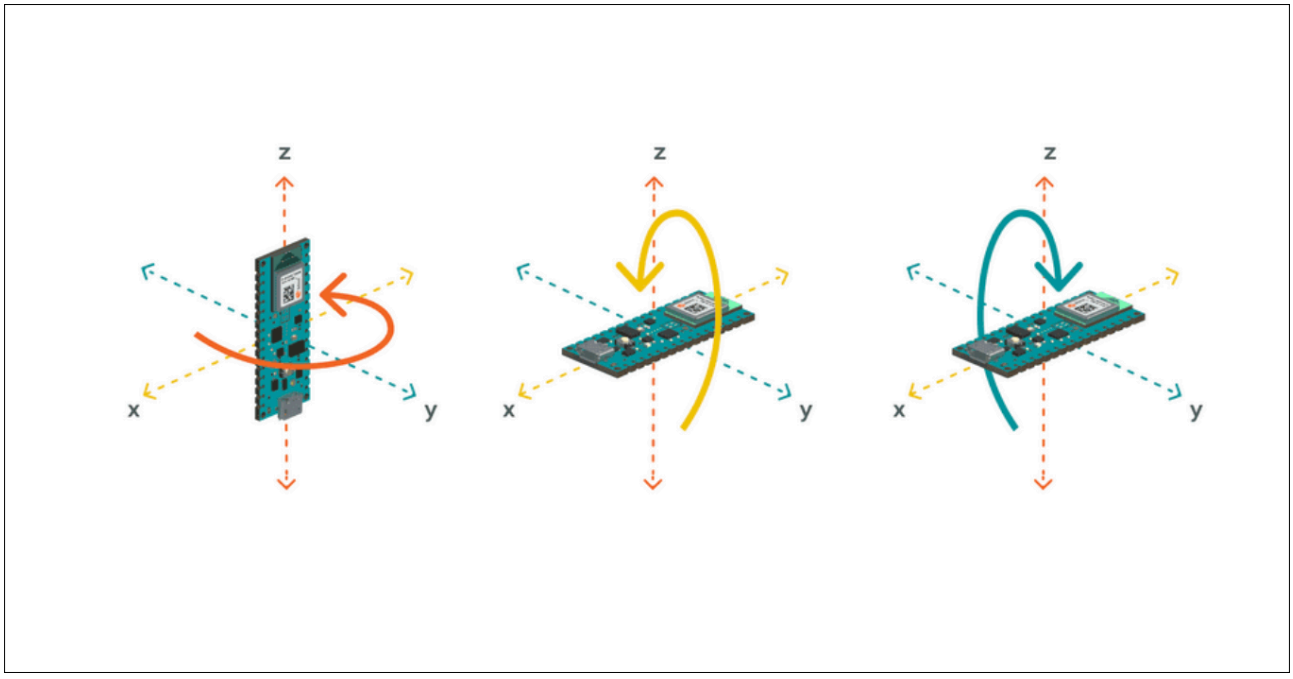
Introduction to the gyroscope

This unit will focus on the 3-axis gyroscope sensor of the IMU system on the Arduino Nano 33 BLE Rev2, to measure the direction of force on the board to emulate an object's crash. This will be achieved by utilising the values of the gyroscope's axes and later printing the return values through the Arduino IDE serial monitor.

Gyroscope sensors are also called Angular Rate Sensors or Angular Velocity Sensors. Measured in degrees per second, angular velocity is the change in the rotational angle of the object per unit of time.

In this example, we will use the gyroscope as an indicator for the direction of the force that is applied to the board. This will be achieved by swiftly rotating the board for an instant in four directions: forward, backward, to the left, and to the right. The results will be visible through the Serial Monitor.

Figure 1: the gyroscope



Sketch A11.1 gyroscope values

This prints the raw data for the gyroscope. Then we can see the data in the serial monitor. After that, we can see it in the serial plotter (see below).

Arduino sketch

```
#include <Arduino_BMI270_BMM150.h>

float x;
float y;
float z;

void setup()
{
  Serial.begin(9600);
  while (!Serial);
  if (!IMU.begin())
  {
    Serial.println("Failed to initialize IMU!");
    while (1);
  }
  Serial.print("Gyroscope sample rate = ");
  Serial.print(IMU.gyroscopeSampleRate());
  Serial.println(" Hz");
  Serial.println();
  Serial.println("Gyroscope in degrees/second");
  Serial.println("X\tY\tZ");
}

void loop()
{
  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

Very similar to the `accelerometer` sketch. It uses the same module. The serial plotter can be accessed by clicking on the icon next to the serial monitor.

🌻 Challenge

Have the RGB LED change colour depending on the angle of rotation.

Figure 2: The serial plotter on the left and the serial monitor on the right

Figure A11.1a: the serial monitor

Figure A11.1b: the serial plotter

Sketch A11.2 gyroscope collision

The collision is the direction of the force as a statement when it exceeds a certain value (threshold). Just rotate it about the x-axis in either direction or the y-axis in either direction.

Arduino sketch

```
#include <Arduino_BMI270_BMM150.h>

float x;
float y;
float z;
int plusThreshold = 30;
int minusThreshold = -30;

void setup()
{
  Serial.begin(9600);
  while (!Serial);
  Serial.println("Started");
  if (!IMU.begin())
  {
    Serial.println("Failed to initialize IMU!");
    while (1);
  }
  Serial.print("Gyroscope sample rate = ");
  Serial.print(IMU.gyroscopeSampleRate());
  Serial.println(" Hz");
  Serial.println();
  Serial.println("Gyroscope in degrees/second");
}

void loop()
```

```
{
  if (IMU.gyroscopeAvailable())
  {
    IMU.readGyroscope(x, y, z);
  }

  if(y > plusThreshold)
  {
    Serial.println("Collision front");
    delay(500);
  }

  if(y < minusThreshold)
  {
    Serial.println("Collision back");
    delay(500);
  }

  if(x < minusThreshold)
  {
    Serial.println("Collision right");
    delay(500);
  }

  if(x > plusThreshold)
  {
    Serial.println("Collision left");
    delay(500);
  }
}
```

🗒️ Notes

Rather than data, we get a message.

🌻 Challenge

Use the RGB LED as well.

Figure A11.2: Rotating in the x and y axis

```
Collision front
Collision front
Collision front
Collision back
Collision front
Collision back
Collision front
Collision back
Collision front
Collision left
Collision right
Collision left
Collision right
Collision left
Collision right
Collision left
Collision left
```