PU 6050 Tutorial | How to Program MPU 6050 With Arduino

Hello all, welcome to another Arduino Sensor Tutorial, in this Blog, we will learn how to wire and code MPU 6050, lets start!

accelerometer gyroscope sensor

157,242 VIEWS 10 COMMENTS 23 RESPECTS

COMPONENTS AND SUPPLIES

	Arduino Nano R3	× 1

SparkFun Triple Axis Accelerometer and Gyro Breakout - MPU-6050

× 1

UTSOURCE Electronic Parts

× 1

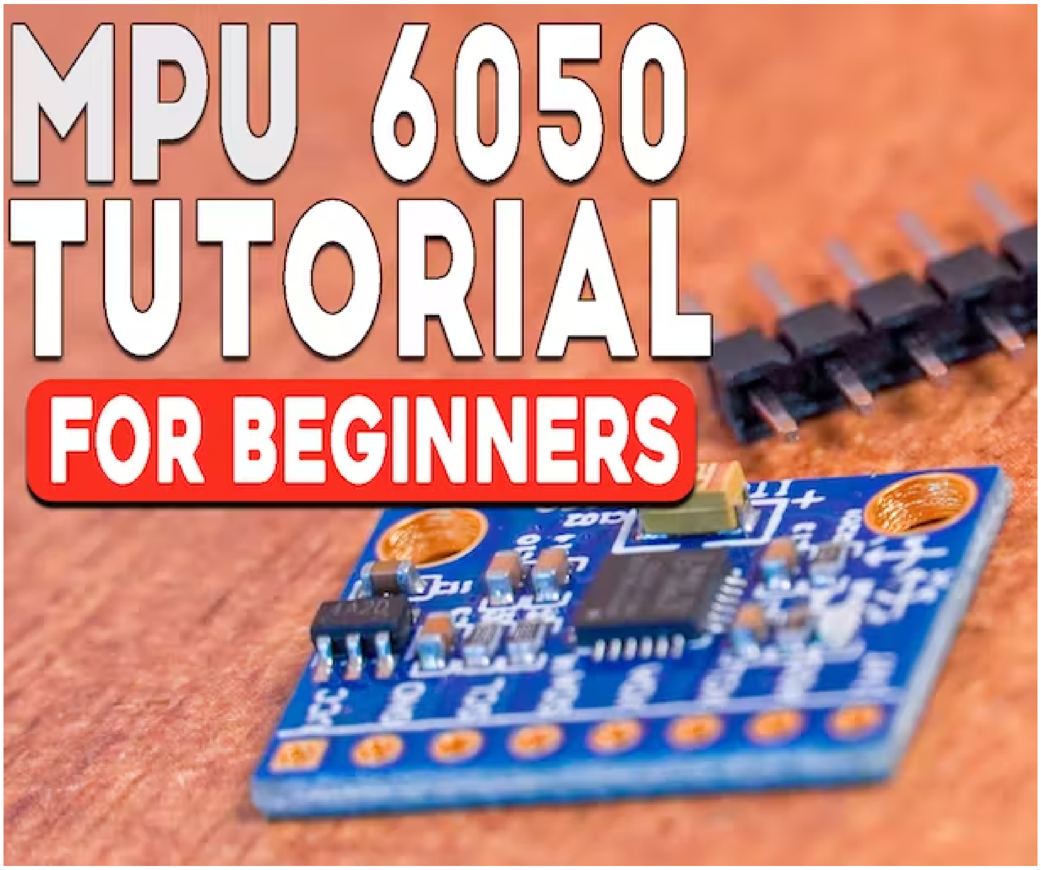
NECESSARY TOOLS AND MACHINES

Premium Female/Male Extension Jumper Wires, 40 x 6" (150mm)

APPS AND ONLINE SERVICES

Arduino IDE

ABOUT THIS PROJECT



Hello all, welcome to another Arduino Sensor Tutorial, in this Blog, we will learn how to wire and code MPU 6050 which is a 6 axis Accelerometer, with our Arduino Board, in detail, so follow till end!

Supplies:

Hardware

- Arduino Nano
- MPU 6050
- Jumper Wire

Software

- Arduino IDE

Step 1: Watch the Video Tutorial

Step 2: Introduction

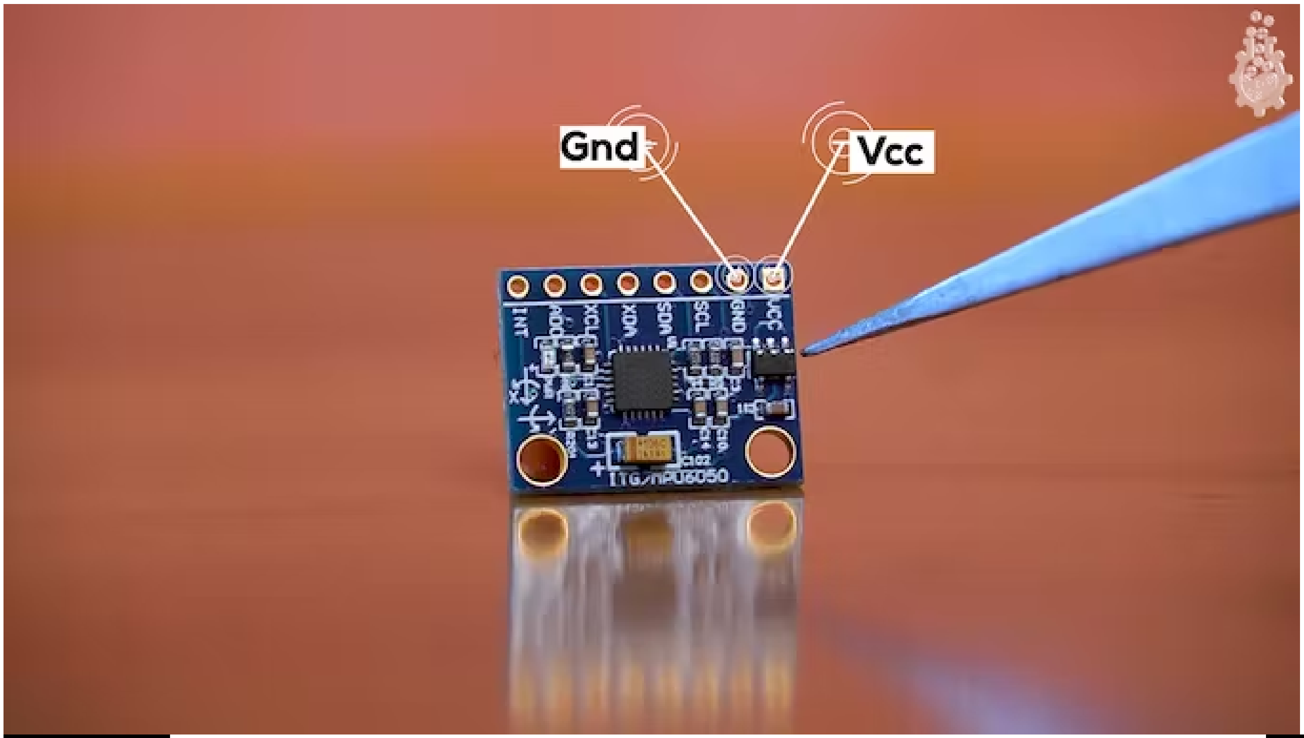
PREVIOUS

NEXT

MPU6050 is the world’s first integrated 6-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package which is theIntegrated Circuit in Middle, it is based on I2C communication protocol, rather than discussing the specifics, refer the Datasheet of MPU 6050.

MPU-6000-DATASHEET1.PDF

Step 3: Hardware



PREVIOUS

NEXT

MPU 6050 comes in a Module form, with 8 pins, but don’t worry, we will use only 4 important pins and it will be sufficient to integrate with our Arduino Board.

So we have VCC, ground, which takes any input from 2v to 5v, since this board has a voltage regulator on board and thus supports 3.3v logic high and 5v logic high.

Next we have few complimentary resistors and capacitors in SMD package and the most important PART the MPU6050 IC, which is a MEMS or say micro electro mechanical system, which changes voltage depending on change in axis position.

This IC also has SCL SDA, which are I2C pins and XDA and XCL which are auxiliary Serial pins, we won't use them with Arduino for this tutorial, we have AD0 which is address select between Auxiliary and Primary ports, lastly we have INT interrupt pin,

connections for our Arduino UNO and NANO are as following:

VCC - 5v

GND - GND

SCL - A5

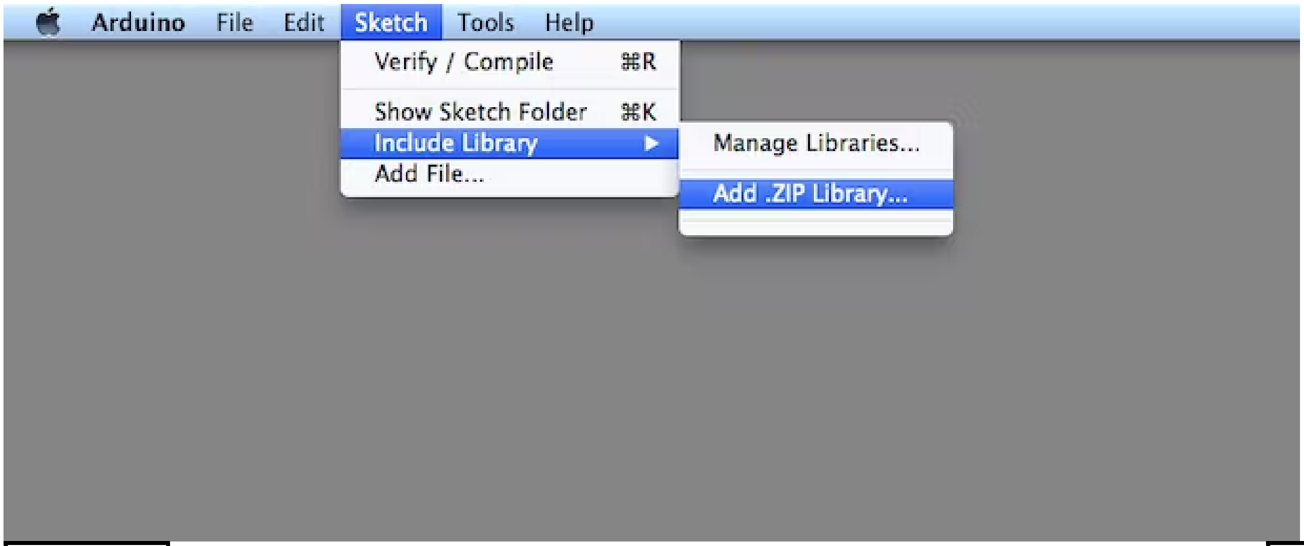
SDA - A4

(only SDA and SCL pins change for other Arduino boards.)

And that's all for connection

(find all the components at UTSOURCE)

Step 4: Install Libraries



PREVIOUS

NEXT

Before we start Coding, we will need a library called as Arduino MPU-6050 by jarzebski,

also we will need Wire Library, which is inbuilt, so we will just install MPU - 6050 Library.

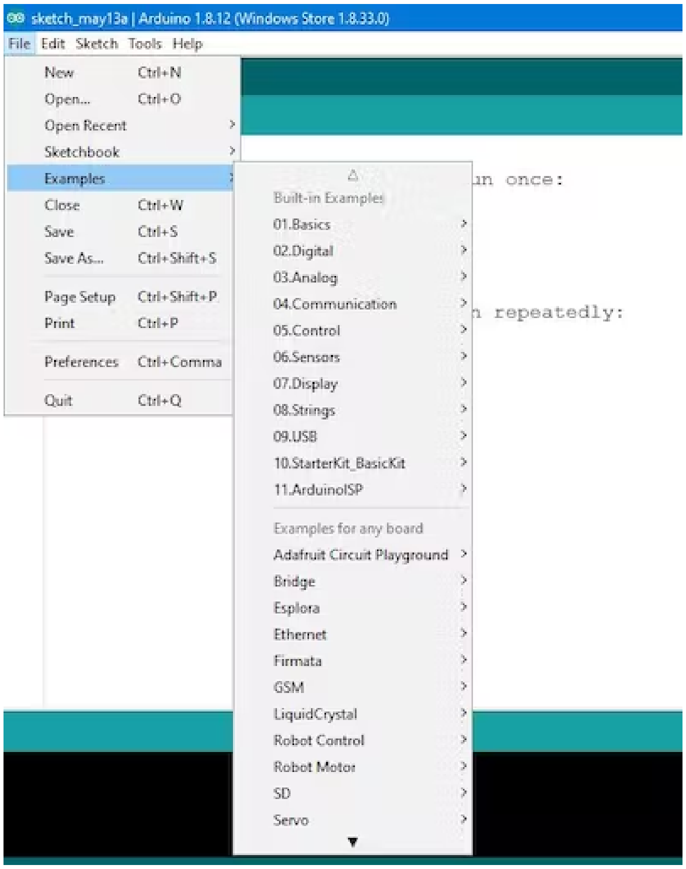
here is the link to MPU6050 Library.

To install a new library into your Arduino IDE you can use the Library Manager.

- Open the IDE and click to the "Sketch" menu and then Include Library > select the option to "Add.ZIP Library".
- Navigate to the.zip file's location and open it.

for more information on importing, refer https://www.arduino.cc/en/guide/libraries

Step 5: Open Gyroscope Example.



PREVIOUS NEXT

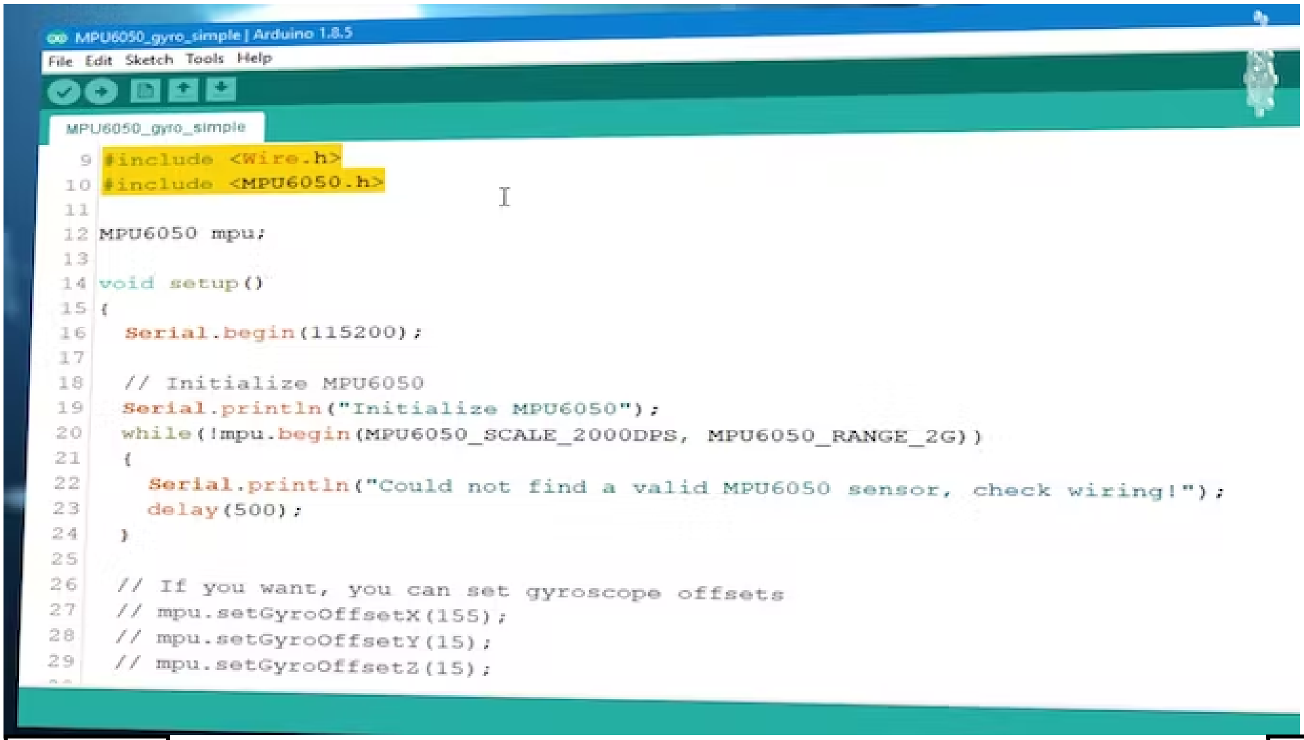
once the MPU-6050 library is added to Arduino IDE, we have quite a list of examples to choose from, like

- MPU6050_accel_pitch_roll
- MPU6050_accel_simple
- MPU6050_free_fall
- MPU6050_gyro_pitch_roll_yaw
- MPU6050_gyro_simple
- MPU6050_motion
- MPU6050_temperature

we need to start slow to understand the Library and Basics, so lets start with *MPU6050_gyro_simple*example.

(did you know MPU6050 has a Tempreature Sensor as well, but not very accurate one so we didn't discuss it here!)

Step 6: Understand the Code


MPU6050_gyro_simple | Arduino 1.8.5
File Edit Sketch Tools Help
MPU6050_gyro_simple

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

void setup()
{
  Serial.begin(115200);

  // Initialize MPU6050
  Serial.println("Initialize MPU6050");
  while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(500);
  }

  // If you want, you can set gyroscope offsets
  // mpu.setGyroOffsetX(155);
  // mpu.setGyroOffsetY(15);
  // mpu.setGyroOffsetZ(15);
```

PREVIOUS NEXT

Basically in this example, we will see if our sensor is working so we will display the sensor data on serial monitor.

So we *begin* the *serial monitor* in setup part.

```
Serial.begin(115200);
```

In this *While Loop*, the sensor test sequence is executed.

```
while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
{    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
delay(500);
}
```

> Sometimes we are making a project, and have to set our sensor in a specific orientation, we need offsets, we don’t need it for this tutorial,

- but to change offset, simply un-comment these lines.

```
// mpu.setGyroOffsetX(155);
// mpu.setGyroOffsetY(15);
// mpu.setGyroOffsetZ(15);
( uncomment by removing "//" )
```

- there is a calibration line, which will virtually set our sensor flat.

```
// Calibrate gyroscope. The calibration must be at rest.
// If you don't want calibrate, comment this line.
mpu.calibrateGyro();
```

> Remember offset and calibration are two different things, offset will give you defined calibration, for example, you can mount this sensor at weird angle and yet it will act as reference point for zero.

- Next we have sensitivity, which at default is 3.

```
// Set threshold sensivty. Default 3.
// If you don't want use threshold, comment this line or set 0.
mpu.setThreshold(3);
```

- In check loop section, basic hardware checking is done.

```
void checkSettings()
{
Serial.println();

Serial.print(" * Sleep Mode:        ");
Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");

Serial.print(" * Clock Source:      ");
switch(mpu.getClockSource())
{
case MPU6050_CLOCK_KEEP_RESET:     Serial.println("Stops the clock and keeps the timing generator in reset"); break;
case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL with external 19.2MHz reference"); break;
case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL with external 32.768kHz reference"); break;
case MPU6050_CLOCK_PLL_ZGYRO:      Serial.println("PLL with Z axis gyroscope reference"); break;
case MPU6050_CLOCK_PLL_YGYRO:      Serial.println("PLL with Y axis gyroscope refe
```

> I highly suggest to leave this loop as it is.

- in loop section, which is most important part of this entire code, that is getting the values from our sensor. First we need to call the values, using mpu.readRawGyro or mpu.readNormalizeGyro, now the concept of raw and normalized is such that raw are basically numbers and normalized values are values which go through filters and calculations or you can say, processed data.

```
void loop()
{
Vector rawGyro = mpu.readRawGyro();
Vector normGyro = mpu.readNormalizeGyro();
```

- We have 3 axis, called as x y and z, which can be called using variable name which we set as rawGyro, followed by axis name, to make a project, we will need these 3 values of x, y and z using this variablename.axis command.

```
Serial.print(" Xraw = ");
Serial.print(rawGyro.XAxis);
Serial.print(" Yraw = ");
Serial.print(rawGyro.YAxis);
Serial.print(" Zraw = ");
Serial.println(rawGyro.ZAxis);
Serial.print(" Xnorm = ");
Serial.print(normGyro.XAxis);
Serial.print(" Ynorm = ");
Serial.print(normGyro.YAxis);
Serial.print(" Znorm = ");
Serial.println(normGyro.ZAxis);
```

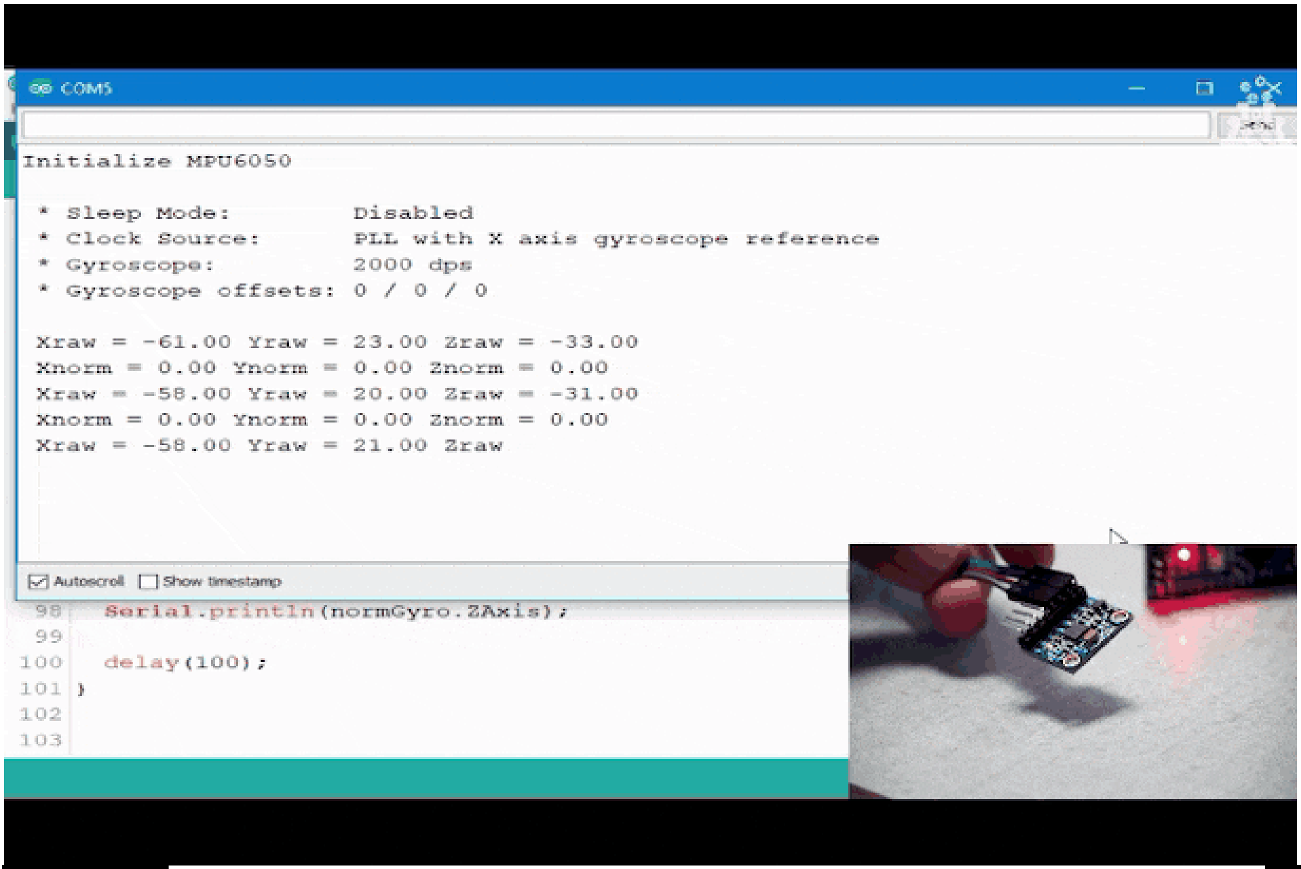
- atlast, to view the values at comfortable speed, lets increase the delay from 10 to 1000

```
delay(10);
}
```

> since our example code is ready and we understand what we did in code, its time to upload the code and view results.

Code can be Found here!

Step 7: Check Results in Serial Port



```
Initialize MPU6050

 * Sleep Mode:          Disabled
 * Clock Source:        PLL with X axis gyroscope reference
 * Gyroscope:           2000 dps
 * Gyroscope offsets:   0 / 0 / 0

 Xraw = -61.00 Yraw = 23.00 Zraw = -33.00
 Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
 Xraw = -58.00 Yraw = 20.00 Zraw = -31.00
 Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
 Xraw = -58.00 Yraw = 21.00 Zraw
```

```
  Serial.println(normGyro.ZAxis);

  delay(100);
}
```

PREVIOUS

NEXT

once the upload is done, its time to open up the serial Monitor and observe output:

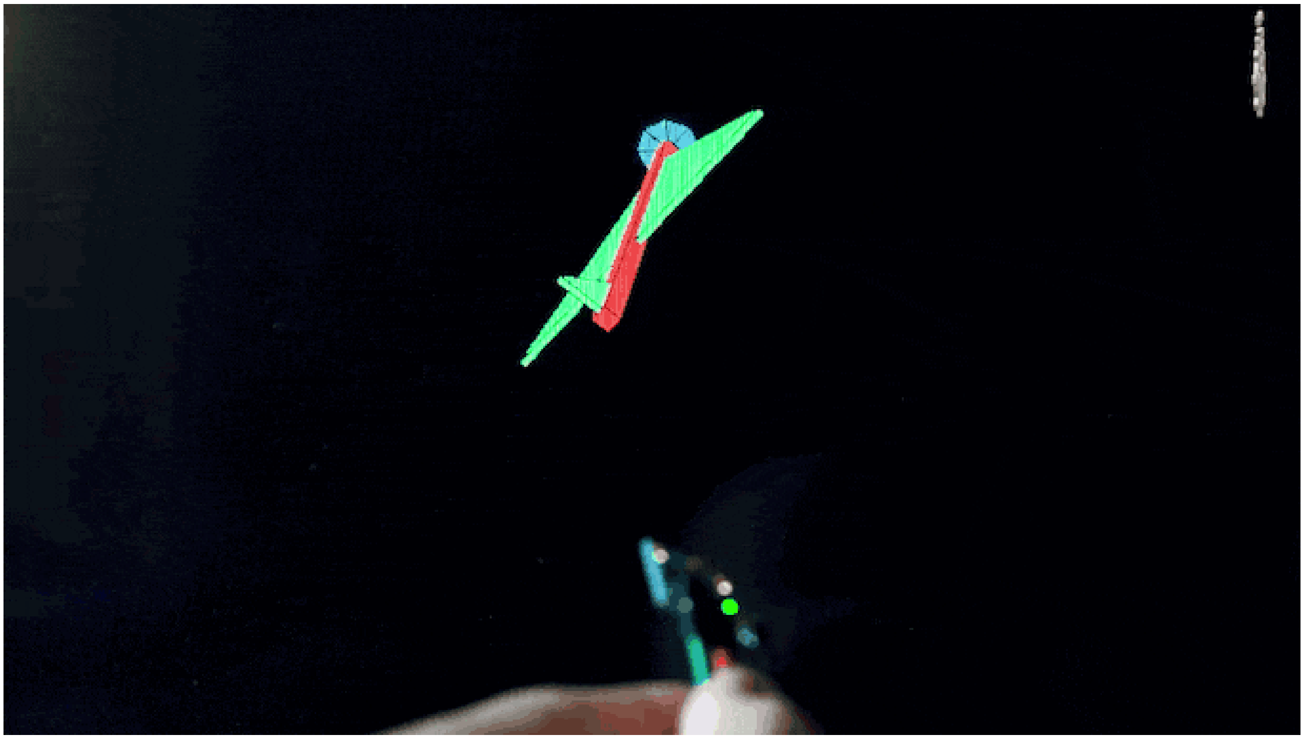
Don’t forget to match the serial port with the baud rate we defined in start of code which is *115200*

If our wiring and hardware is proper, we should get values of each axis on Serial port for both Raw and Normalized gyroscopic change in sensor.

```
Initialize MPU6050
* Sleep Mode:        Disabled
* Clock Source:      PLL with X axis gyroscope reference
* Gyroscope:         2000 dps
* Gyroscope offsets: 0 / 0 / 0
Xraw = -65.00 Yraw = 25.00 Zraw = -29.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -62.00 Yraw = 20.00 Zraw = -32.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -64.00 Yraw = 25.00 Zraw = -30.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -66.00 Yraw = 23.00 Zraw = -29.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -67.00 Yraw = 20.00 Zraw = -33.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -64.00 Yraw = 22.00 Zraw = -31.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
```

here is the logs of what we got on Serial Port.

Step 8: What Next?



Now this was the basic part, we can also use our sensor to control a model Paper plane using processing software, but that's a tutorial for next blog.

Subscribe and hit the notification button, to not miss any future blog/ video from Mission Critical, also give this post thumps up and share it with your friends, until the next one, GoodBye!

CODE

Code snippet #3

Code snippet #6

Code snippet #8

Code snippet #10

Code snippet #3 Plain text

```
// mpu.setGyro
// mpu.setGyro
// mpu.setGyro

( uncomment by
```

Github

https://github.com/jarzebski/Arduino-MPU6050

jarzebski / **Arduino-MPU6050**

337 255

MPU6050 Triple Axis Gyroscope & Accelerometer Arduino Library — Read More

Latest commit to the **master** branch on 4-17-2020 Download as zip

COMMENTS

Please log in or sign up to comment.

Jean Perardel
3 years ago

Great tutorial !
I just discovered that AD0 Pin with a pull up change the I2C address from 0x68 to 0x69. As RTC DS1307 also use 0x68 on my project (very bad luck to have same address...) I would have been stocked without this AD0 ! :p

1 thank

Mission Critical
3 years ago

thank you !

Timbob
2 years ago

Fantastic tutorial. Where is the "subscribe" option? Where is the next lesson?

learnerforlife
2 years ago

Thank you for the tutorial.....I am able to serial plot/print values..... however, while plotting, the plotter hangs after 1 min of data log... can you please help me in this regard?

freeriderbiking
a year ago

Hi There - I am really new to Arduino and coding etc but can see how much potential there is if I can get over the initial step learning curve.
I really want to be able to use a MPU6050 for a specific project and so have set up the sensor with an Arduino Uno.
Your examples are excellent and the explanations too. But I have run into a problem when verifying the MPU6050_gyro_simple example.
I get an error message:

'class MPU6050' has no member named 'begin'
The line in the sketch highlighted is:
while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))

I am sure this is really basic stuff and easy to fix but I am so inexperienced that I don't know what this means....
Any help with this would be very much appreciated as I seem to be stuck before I can even test the sensor.
😐

1 thank

coolroger123
a year ago

The problem is because of copy paste. It adds some invisible blocks which are creating errors. I faced the same problem. There are ways to format it and making it error free. However i typed the same program, and it worked for me.
Hope it helps

codingwelp
6 months ago

I also have the same problem. I retyped it, but it still doesn't work. Any other ideas?

Edit: I got it working. There was another solution, which was to delete all the other MPU6050 libraries.

drhenkpenk
6 months ago

I fixed this problem. this problem relating to other MPU6050 . if you remove other mpu6050 libraries , you can fix it .

jaiswal_sourabh
6 months ago

What to do if I connect my SCL to A0 and SDA to A1, what will be the necessary changes to my codes?

bobfound
5 months ago

good tutorial, but like many others I had problems with library commands not working. That's because I had 4 other MPU6050 libraries installed. I liked this one (Jarzebski) more than the others because of all the great examples, so I deleted all the others. Works now.