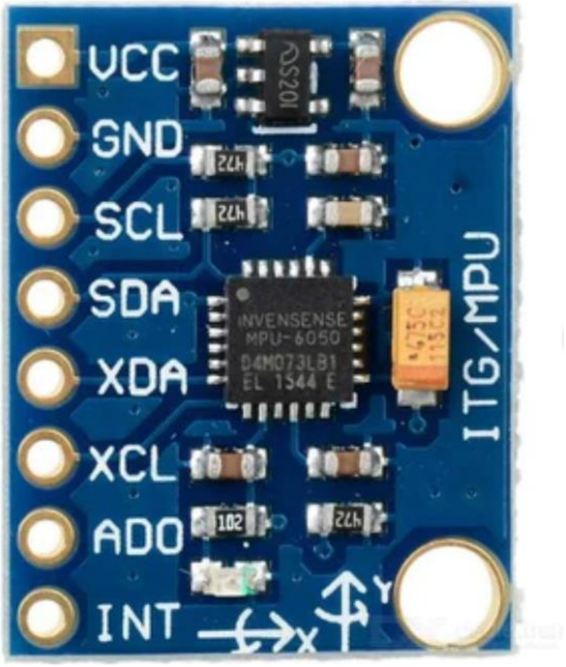
NAME : SAURABH SALVI

ROLL NO : ME22B1033

MARS TASK #3

INTRODUCTION

In task #3, we had been asked to interface Arduino with IMU Sensor(MPU6050). The sensor is actually gives us the data from the gyro and the accelerometer. The sensor with appropriate connections to Arduino was to be done. Also through the code, we must make all the values to be printed in the Serial monitor by establishing a Serial communication.



For my project, I used a Arduino Nano to interface with the IMU sensor.

Code :

- I first downloaded a library from the internet in order to directly use functions from the already existing library for MPU6050.
  Link to the library:
  https://github.com/jarzebski/Arduino-MPU6050
  To include this, I wrote #include<MPU6050.h>

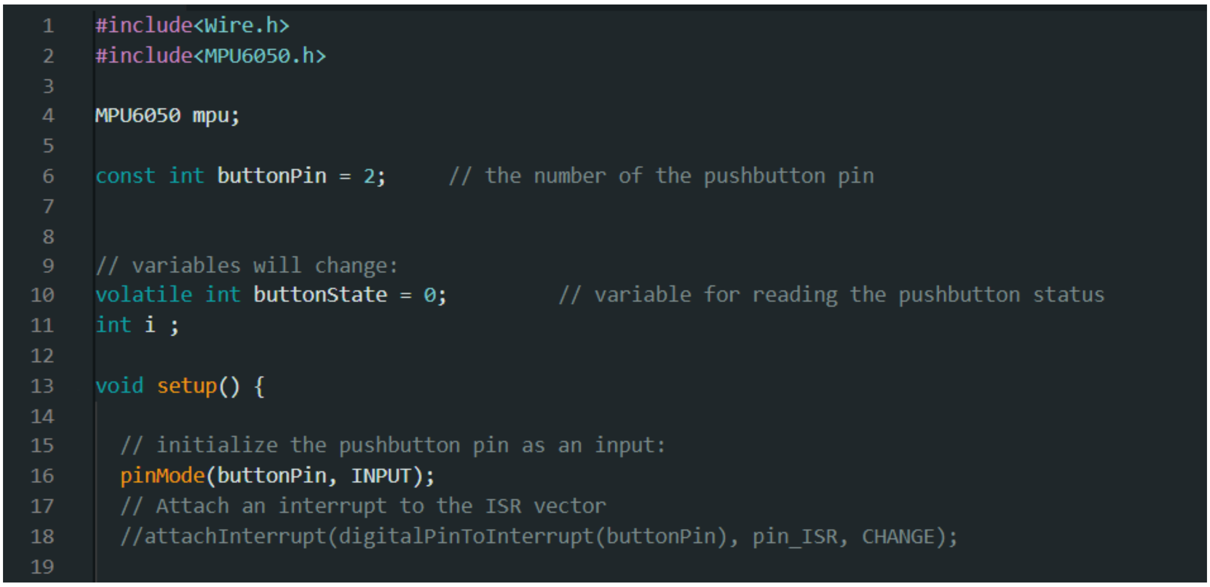
  Also, I included the preexisting library i.e. Wire.h for the sensor purpose.

  I name my MPU6050 sensor as mpu.
- I declare variables int buttonState, buttonPin, and i to values 0, 2 respectively.
- In void setup, as my buttonPin will be basically pushbutton, I kept it as an INPUT using pinMode function
- I establish a serial communication at 1200 baud rate.
- Then I write a code in order to check if the wiring is done with the sensor correctly or not. i.e. to check whether all the connections are proper. After that I wrote to calibrate gyro mpu.calibrateGyro()
- In void loop, I read the status of pushbutton using the digitalRead() and store it in buttonState and print the buttonState
- After that, I call the pin_ISR() function.

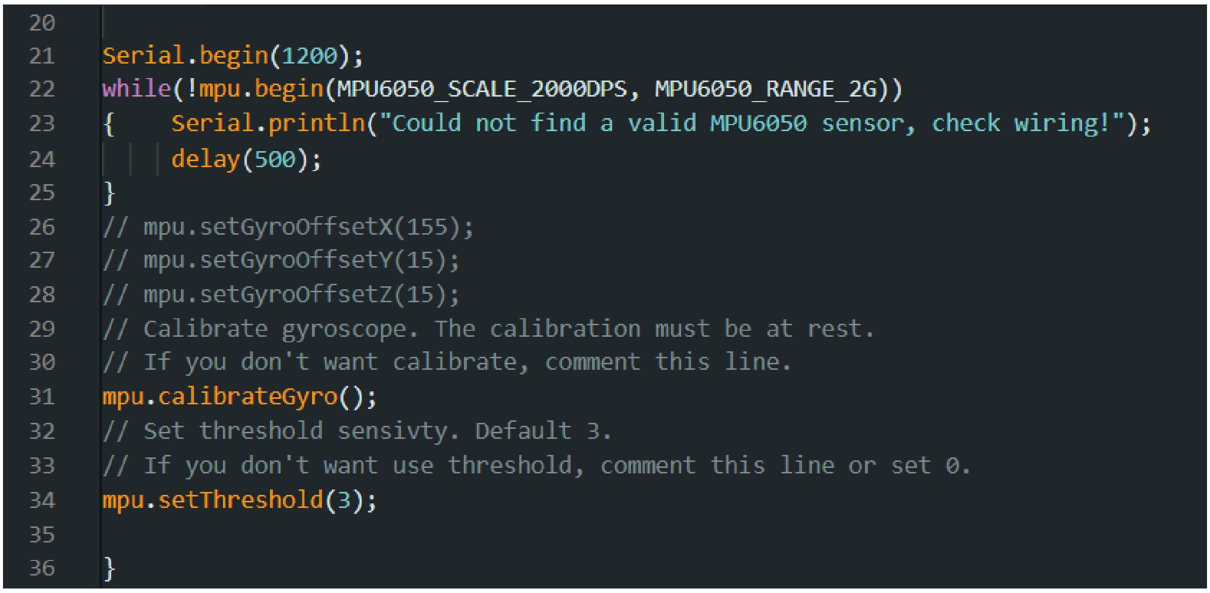
- I define a function pin_ISR with void since I don’t need any return value and in that I write my code regarding the raw data collection which will be executed if the buttonState is 1.

- I make a Vector rawGyro (its name) and in that I keep the values I get from mpu sensor using mpu.readRawGyro() and similarly I do with normGyro using mpu.readNormalizeGyro().

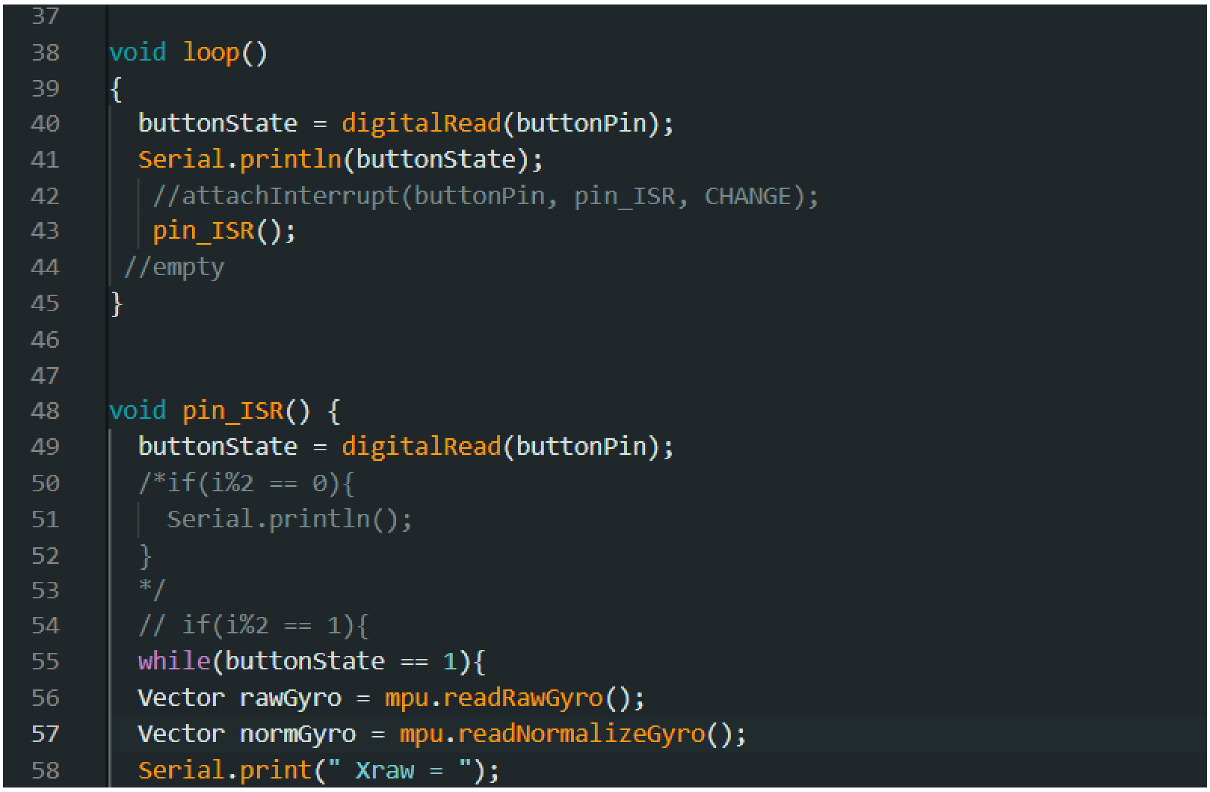
- After that, I print the X (by rawGyro.xAxis), Y, Z Raw values from gyro and X (by normGyro.xAxis), Y, Z Norm values from gyro and print them in Serial monitor and then I increment ‘i’



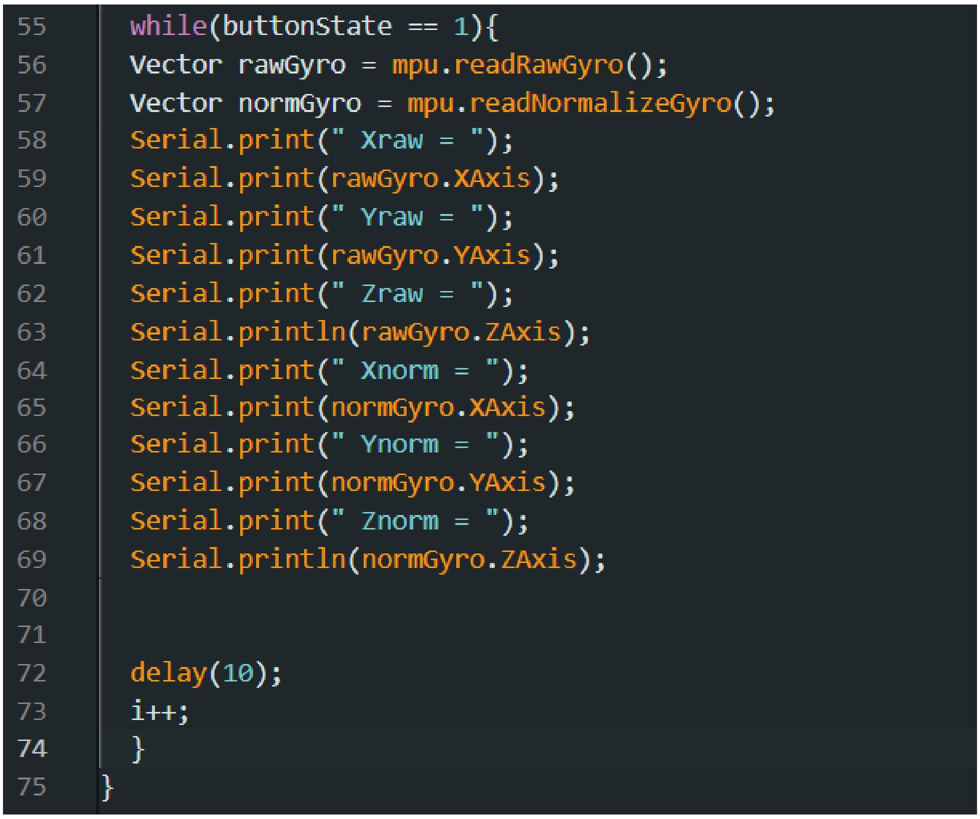
```
#include<Wire.h>
#include<MPU6050.h>

MPU6050 mpu;

const int buttonPin = 2;     // the number of the pushbutton pin


// variables will change:
volatile int buttonState = 0;         // variable for reading the pushbutton status
int i ;

void setup() {

  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  // Attach an interrupt to the ISR vector
  //attachInterrupt(digitalPinToInterrupt(buttonPin), pin_ISR, CHANGE);

```



```

Serial.begin(1200);
while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
{    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
     delay(500);
}
// mpu.setGyroOffsetX(155);
// mpu.setGyroOffsetY(15);
// mpu.setGyroOffsetZ(15);
// Calibrate gyroscope. The calibration must be at rest.
// If you don't want calibrate, comment this line.
mpu.calibrateGyro();
// Set threshold sensivty. Default 3.
// If you don't want use threshold, comment this line or set 0.
mpu.setThreshold(3);

}
```



```

void loop()
{
  buttonState = digitalRead(buttonPin);
  Serial.println(buttonState);
   //attachInterrupt(buttonPin, pin_ISR, CHANGE);
   pin_ISR();
 //empty
}


void pin_ISR() {
  buttonState = digitalRead(buttonPin);
  /*if(i%2 == 0){
    Serial.println();
  }
  */
  // if(i%2 == 1){
  while(buttonState == 1){
  Vector rawGyro = mpu.readRawGyro();
  Vector normGyro = mpu.readNormalizeGyro();
  Serial.print(" Xraw = ");
```



```
    while(buttonState == 1){
    Vector rawGyro = mpu.readRawGyro();
    Vector normGyro = mpu.readNormalizeGyro();
    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);


    delay(10);
    i++;
    }
}
```

Procedure :

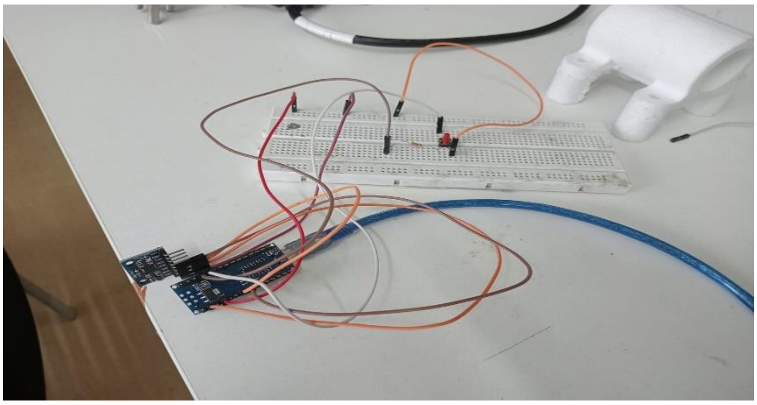
I made connections with the Arduino Nano such that ,

VCC - 5v

GND - GND

SCL - A5

SDA - A4



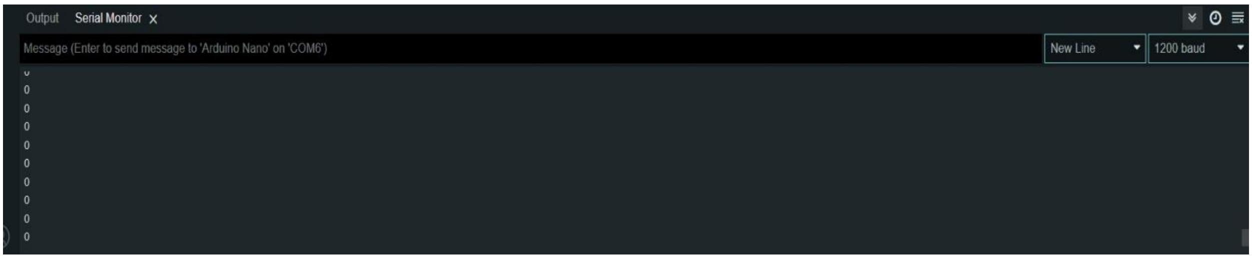
- The board was set to Arduino Nano before uploading the code and I searched on internet about the basic code for interfacing this sensor and arduino.

- After checking if the raw data is correctly shown on the Serial monitor or not, I started with building of pushbutton and for that I connected a pin of pushbutton to digital pin 2 and other 2 pins to the 5V rail on breadboard and to ground.

- Since before even proceeding to pushbutton, it was necessary to see if we are able to get the data without any more complications.

- In the process, at many instants, debugging was required to spot on the mistakes which were not allowing the desired result to come.

Output :



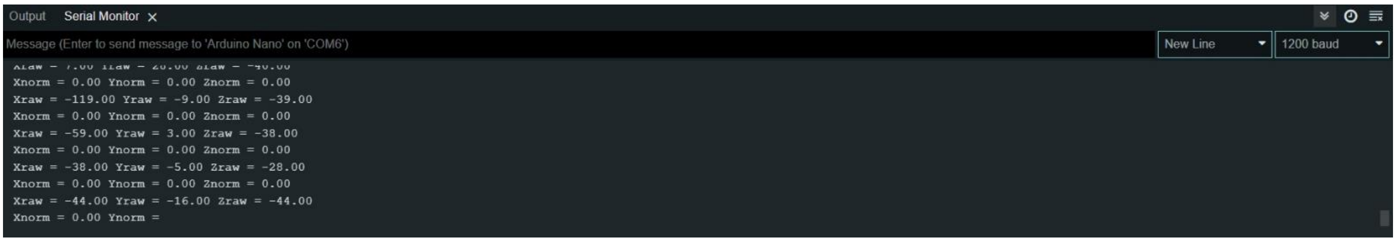
Message (Enter to send message to 'Arduino Nano' on 'COM6')

New Line

1200 baud

```
0
0
0
0
0
0
0
0
0
0
```

Fig. before pressing pushbutton



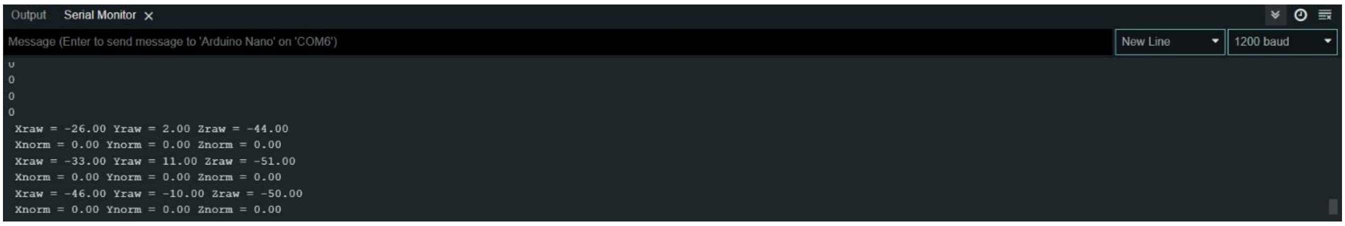
Message (Enter to send message to 'Arduino Nano' on 'COM6')

New Line

1200 baud

```
Xraw = 7.00 Yraw = 20.00 Zraw = -40.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -119.00 Yraw = -9.00 Zraw = -39.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -59.00 Yraw = 3.00 Zraw = -38.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -38.00 Yraw = -5.00 Zraw = -28.00
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
Xraw = -44.00 Yraw = -16.00 Zraw = -44.00
Xnorm = 0.00 Ynorm =
```

Fig. after pressing pushbutton



Message (Enter to send message to 'Arduino Nano' on 'COM6')

New Line

1200 baud

```
0
0
0
0
 Xraw = -26.00 Yraw = 2.00 Zraw = -44.00
 Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
 Xraw = -33.00 Yraw = 11.00 Zraw = -51.00
 Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
 Xraw = -46.00 Yraw = -10.00 Zraw = -50.00
 Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00
```

Fig. At the instant when pushbutton was pressed

Resources :

https://www.instructables.com/MPU-6050-Tutorial-How-to-Program-MPU-6050-With-Ard/

https://arduino.stackexchange.com/questions/17/avrdude-stk500-getsync-not-in-sync-resp-0x00-aka-some-dude-named-avr-won

https://www.allaboutcircuits.com/technical-articles/using-interrupts-on-arduino/

https://www.electronicwings.com/arduino/mpu6050-interfacing-with-arduino-uno

https://reference.arduino.cc/reference/tr/language/functions/external-interrupts/attachinterrupt/

Things I learnt :

From this task, I got to learn that how can we use the MPU6050 sensor in order to obtain the raw data from the sensor .

Since this sensor is basically a 3 axis gyro sensor and gyro being a very important component in deciding the navigation or the path or deciding stability of the concerned system , I got to learn about this important part as well.

Also, It was my first time of using Arduino ‘Nano’ and working on it with hardware was also quite a good experience for me.

Overall, this task was really interesting to work on, since the applications of this sensor are wide and is a very useful sensor.