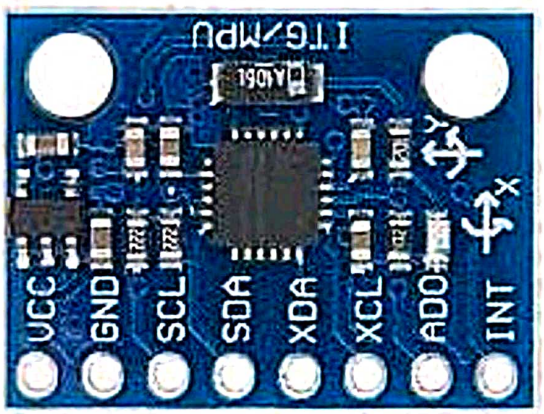
Practical – 12

AIM: What is MPU6050 sensor? Explain it's Structure, How it works, Applications. Programming & Interfacing of MPU6050 Sensor with ESP8266 using OTA.

- **What is MPU6050?**

- MPU6050 is a MEMS-based 6-axis motion tracking device. It has an on-chip gyroscope and accelerometer sensors along with temperature sensor.
- This module is of very small in size, has low power consumption requirements, highly accurate, has high repeatability, high shock tolerance, it has application-specific performance programmability and low consumer price points. MPU6050 is an Accelerometer & Gyro sensor combined in a single chip

- **MPU6050 Circuit:**



- **Working of MPU6050:**

- The MPU6050 is a Micro-Electro-Mechanical Systems (MEMS) that consists of a 3-axis Accelerometer and 3-axis Gyroscope inside it. This helps us to measure acceleration, velocity, orientation, displacement and many other motion-related parameters of a system or object. This module also has a (DMP) Digital Motion Processor inside it which is powerful enough to perform complex calculations and thus free up the work for Microcontroller.
- The module also has two auxiliary pins which can be used to interface external IIC modules like a magnetometer, however, it is optional. This module also has well documented and revised libraries available hence it's very easy to use with famous platforms like Arduino.

- **Application of MPU6050:**

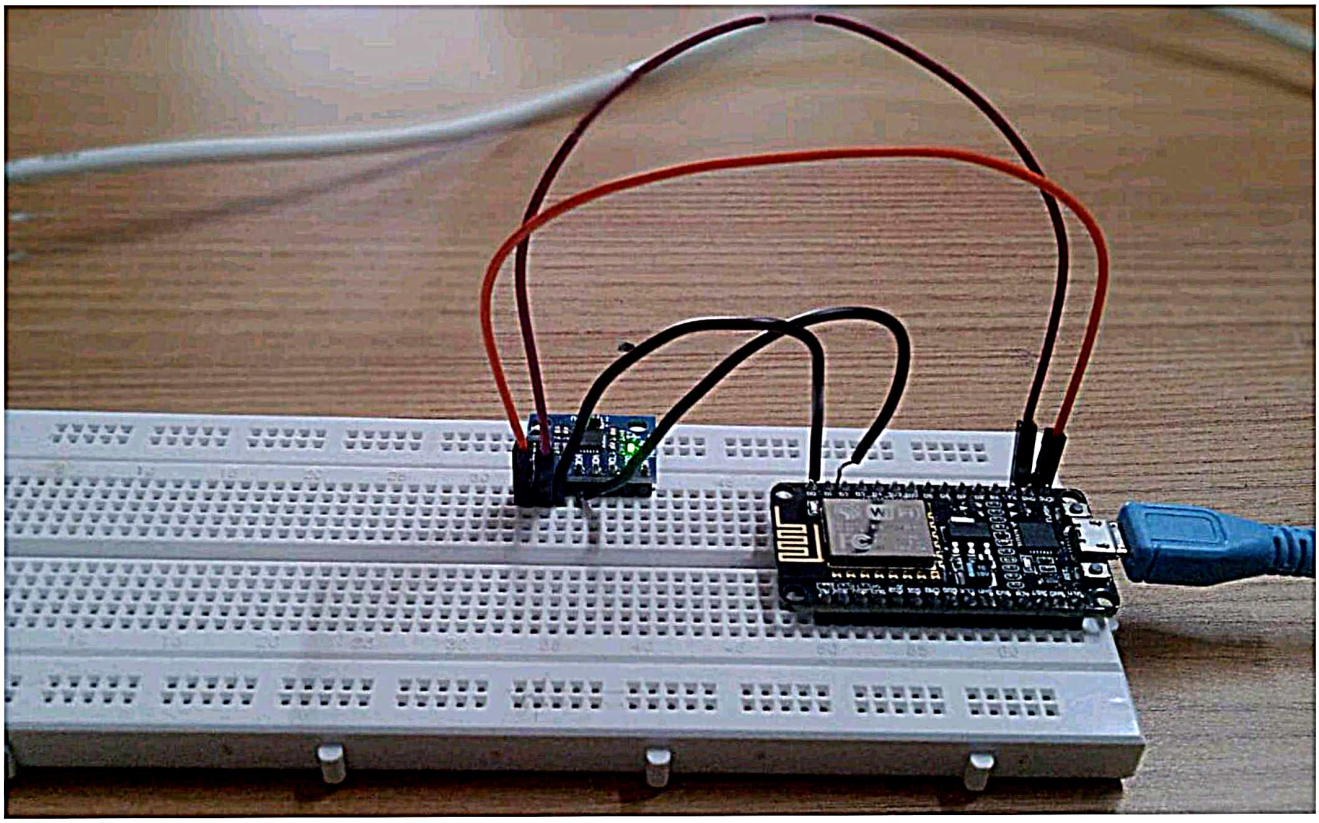
1. Drones / Quadcopters as direction controller
2. Self-balancing robots
3. IMU measurement
4. Tilt sensor
5. Robotic arm controls.
6. Handset and portable gaming
7. Motion-based game controllers

CODE:

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
Adafruit_MPU6050 mpu;

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

 // Try to initialize!
 if (!mpu.begin()) {
  Serial.println("Failed to find MPU6050 chip");
  while (1) {
   delay(10);
  }
 }
 Serial.println("MPU6050 Found!");
 // set accelerometer range to +-8G
 mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
 // set gyro range to +- 500 deg/s
 mpu.setGyroRange(MPU6050_RANGE_500_DEG);
 // set filter bandwidth to 21 Hz
 mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
 delay(100);
}
```

```
void loop() {
  /* Get new sensor events with the readings */
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  /* Print out the values */
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");

  Serial.print("Rotation X: ");
  Serial.print(g.gyro.x);
  Serial.print(", Y: ");
  Serial.print(g.gyro.y);
  Serial.print(", Z: ");
  Serial.print(g.gyro.z);
  Serial.println(" rad/s");

  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degC");

  Serial.println("");
  delay(500);
}
```

OUTPUT:



File Edit Sketch Tools Help

Practical-12

```
  // set filter bandwidth to 21 Hz
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);

  delay(100);
}

void loop() {
  /* Get new sensor events with the readings */
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  /* Print out the values */
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");

  Serial.print("Rotation X: ");
  Serial.print(g.gyro.x);
  Serial.print(", Y: ");
  Serial.print(g.gyro.y);
  Serial.print(", Z: ");
  Serial.print(g.gyro.z);
  Serial.println(" rad/s");

  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degC");

  Serial.println("");
  delay(500);
```

COM7

Send

```
Acceleration X: 0.22, Y: 0.14, Z: 8.09 m/s^2
Rotation X: -0.02, Y: -0.04, Z: 0.03 rad/s
Temperature: 32.39 degC

Acceleration X: 0.23, Y: 0.11, Z: 8.11 m/s^2
Rotation X: -0.03, Y: -0.04, Z: 0.03 rad/s
Temperature: 32.38 degC

Acceleration X: 0.25, Y: 0.12, Z: 8.09 m/s^2
Rotation X: -0.03, Y: -0.04, Z: 0.03 rad/s
Temperature: 32.37 degC

Acceleration X: 0.26, Y: 0.16, Z: 8.06 m/s^2
Rotation X: -0.02, Y: -0.04, Z: 0.03 rad/s
Temperature: 32.37 degC

Acceleration X: 0.23, Y: 0.13, Z: 8.08 m/s^2
Rotation X: -0.03, Y: -0.04, Z: 0.03 rad/s
Temperature: 32.38 degC

Acceleration X: 0.24, Y: 0.13, Z: 8.11 m/s^2
Rotation X: -0.02, Y: -0.04, Z: 0.03 rad/s
Temperature: 32.39 degC

Acceleration X: 0.26, Y: 0.09, Z: 8.09 m/s^2
Rotation X: -0.02, Y: -0.04, Z: 0.03 rad/s
Temperature: 32.38 degC
```

Autoscroll Show timestamp Newline 115200 baud Clear output

Practical-12 | Arduino 1.8.19

File Edit Sketch Tools Help

Practical-12

```
  // set filter bandwidth to 21 Hz
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);

  delay(100);
}

void loop() {
  /* Get new sensor events with the readings */
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  /* Print out the values */
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");

  Serial.print("Rotation X: ");
  Serial.print(g.gyro.x);
  Serial.print(", Y: ");
  Serial.print(g.gyro.y);
  Serial.print(", Z: ");
  Serial.print(g.gyro.z);
  Serial.println(" rad/s");

  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degC");
```