**IOT Project**

**Aim**

To design and implement a system that monitors the position and orientation of an object using real-time accelerometer and gyroscope data and displays the data on an OLED screen using the Wokwi Simulator.

**Problem Statement**

In many devices like robots, drones, and fitness trackers, it's important to know how an object moves or rotates. But making such a system with real hardware can be expensive and hard for beginners.

This project offers a simple and low-cost way to understand and try out this idea using a simulator, sensors, and a small display. It's helpful for students or anyone who wants to learn how these systems work.

**Scope of the Solution**

* Real-time display of orientation data (roll, pitch, yaw).
* Simulated environment using Wokwi.
* Display output on OLED screen.
* GitHub and documentation for further development.

**Required Component**

* ESP32 Development Board
* MPU6050 Sensor Module
* OLED Display
* Jumper wires
* Wokwi (for simulation)

**Code**

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_MPU6050.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

Adafruit\_MPU6050 mpu;

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

void setup() {

  Serial.begin(115200);

if (!mpu.begin()) {

    Serial.println("Failed to find MPU6050 chip");

    while (1) {

      delay(10);

    }

  }

  Serial.println("Found MPU6050 chip");

  mpu.setAccelerometerRange(MPU6050\_RANGE\_2\_G);

  mpu.setGyroRange(MPU6050\_RANGE\_250\_DEG);

  mpu.setFilterBandwidth(MPU6050\_BAND\_21\_HZ);

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

    Serial.println(F("SSD1306 allocation failed"));

    for (;;);

  }

  display.display();

  delay(2000);

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(SSD1306\_WHITE);

  display.println(F("MPU6050 and OLED"));

  display.display();

  delay(1000);

}

void loop() {

    sensors\_event\_t a, g, temp;

    mpu.getEvent(&a, &g, &temp);

    display.clearDisplay();

    display.setCursor(0, 0);

    display.print("AccX: "); display.println(a.acceleration.x);

    display.print("AccY: "); display.println(a.acceleration.y);

    display.print("AccZ: "); display.println(a.acceleration.z);

    display.print("GyroX: ");  display.println(g.gyro.x);

    display.print("GyroY: ");  display.println(g.gyro.y);

    display.print("GyroZ: ");  display.println(g.gyro.z);

    display.display();

    delay(500);

}

**Flowchart of the code**

1. Turn on the Serial Monitor to enable messages on the computer screen.
2. Try to detect the MPU6050 sensor:
   * If the sensor is found, print “Found MPU6050 chip”.
   * If not found, print an error message and stop the program.
3. Set the sensor’s sensitivity:
   * Accelerometer range to ±2g
   * Gyroscope range to ±250 degrees per second
   * Filter bandwidth to 21 Hz
4. Initialize the OLED display:
   * If successful, continue
   * If it fails, print an error and stop the program
5. Display a welcome message: “MPU6050 and OLED” on the screen for 2 seconds
6. Main Loop:

* Read new data from the sensor:

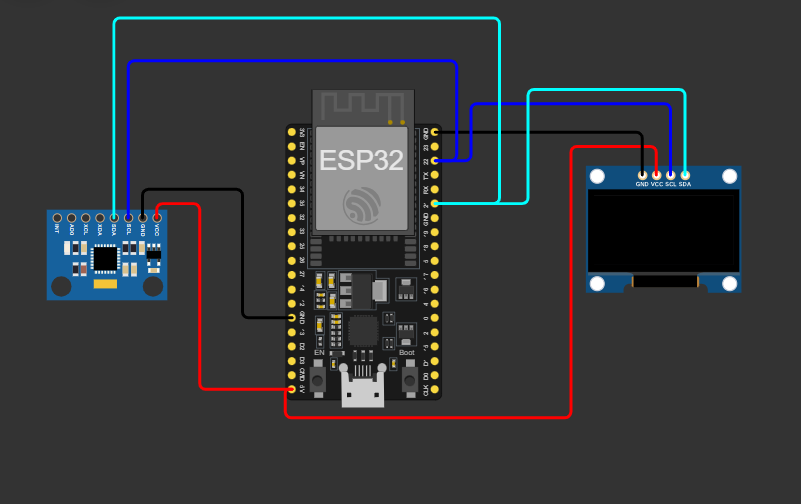
Acceleration values (X, Y, Z axis)

Gyroscope values (X, Y, Z axis)

* Clear the OLED screen to remove old data
* Print the latest sensor values to the OLED screen
* Refresh the display to show updated readings
* Wait for 500 milliseconds

1. Repeat the loop to continue showing live sensor data

**Simulated Circuit:**

****

**Video of the Demo:**

<https://drive.google.com/file/d/1bpTtUigJmdgq5zW_CZuIUvWy7FmR0EPv/view?usp=sharing>