DRONE ORIENTATION TRACKING USING MPU6050

Index:

1. Project Overview

This project aims to track the orientation of a drone using real-time accelerometer data from the MPU6050 sensor. The data is displayed both on an OLED screen and sent to the Serial Monitor for monitoring. This setup is ideal for drone development, robotics, and embedded system learning where real-time motion sensing and display are critical.

2. Components Used

Component	Quantity	Description
ESP32	1	Microcontroller board with built-in WiFi & Bluetooth
MPU6050	1	Accelerometer and Gyroscope sensor
OLED SSD1306	1	128x64 I2C OLED Display
Jumper Wires	-	For connections

3. Principle of Operation

The MPU6050 provides acceleration data on X, Y, and Z axes. The ESP32 reads this data via the I2C interface. The data is displayed on the OLED in real-time. The same data is also sent to the Serial Monitor via USB for debugging or logging.

4. Circuit Connections

Both MPU6050 and OLED use the I2C protocol, and therefore share the same SDA and SCL lines.

Connection Table:

Component Pin	Connect to ESP32 Pin
MPU6050 VCC	3.3V
MPU6050 GND	GND
MPU6050 SDA	GPIO 21
MPU6050 SCL	GPIO 22
OLED VCC	3.3V
OLED GND	GND
OLED SDA	GPIO 21
OLED SCL	GPIO 22

Note: SDA (Data) and SCL (Clock) lines are shared on the same I2C bus.

5. Code Explanation (Arduino IDE)

Libraries Used:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "MPU6050.h"
```

OLED Setup:

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET    -1
Adafruit_SSD1306 display (SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);
```

MPU6050 Object Creation:

```
MPU6050 mpu;
```

---

6. setup() Function

Serial Communication Setup:

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

I2C Start:

```
Wire.begin();
```

MPU6050 Initialization:

```
mpu.initialize();
if (!mpu.testConnection()) {
  Serial.println("MPU6050 connection failed!");
  while (1);
}
```

OLED Initialization and Welcome Display:

```
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
  Serial.println("OLED initialization failed!");
  while (1);
}
display.println("Drone Orientation");
display.display();
delay(1000);
```

7. loop() Function

Reading Accelerometer Data:

```
mpu.getAcceleration(&accX, &accY, &accZ);
```

Display on OLED:

```
display.setCursor(0, 16);
display.print("AccX: "); display.println(accX);
```

```
display.setCursor(0, 32);
display.print("AccY: "); display.println(accY);
display.setCursor(0, 48);
display.print("AccZ: "); display.println(accZ);
```

Serial Output:

```
Serial.print("AccX: "); Serial.print(accX);
Serial.print(" | AccY: "); Serial.print(accY);
Serial.print(" | AccZ: "); Serial.println(accZ);
```

---

8. Sample Output

OLED Output:

```
Drone Orientation
AccX: 0
AccY: 0
AccZ: 16384
```

Serial Monitor Output:

```
MPU6050 connected successfully.
AccX: 0 | AccY: 0 | AccZ: 16384
```

---

9. Troubleshooting Tips

Issue	**Solution**
OLED not displaying	Check I2C address (default is 0x3C).
MPU6050 not detected	Verify wiring; ensure it's powered with 3.3V.
Data is erratic	Add filtering or stabilize power supply.
No serial output	Make sure correct COM port is selected in Arduino IDE.

10. Applications and Extensions

- Used in drone flight stabilization.

- Helpful in gesture-based controls.

- Can be extended to measure orientation angles (pitch, roll, yaw) using gyro + accelerometer fusion.

- Add SD card to log data.

- Integrate with Bluetooth or Wi-Fi for wireless telemetry.

11. Useful Notes

- This project is ideal for beginners in motion sensing or embedded electronics.

- You can replace ESP32 with Arduino Uno, but you'll need level shifting or 5V tolerant modules.

-Always keep wiring short and stable when using I2C devices.

12. Conclusion

This project successfully demonstrates the integration of the MPU6050 accelerometer with an ESP32 and OLED display for real-time orientation tracking. The data visualization on both the OLED screen and Serial Monitor provides a robust setup for monitoring drone movements. This foundational system can further be enhanced for complete IMU-based navigation and control applications in drones and robotics.