## MOTION AND POSITION TRACKING SYSTEM

## USING MPU6050

TEAM MEMBERS

1. VISHWA G
2. STEVE CHRISTIAN SAMUEL CS
3. SUNDARESH S
4. SANGEETH KUAMR LM

### Aim

To design and implement a motion and position tracking system using the MPU6050 accelerometer and gyroscope sensor interfaced with an ESP32 microcontroller, enabling real-time monitoring and alert for sudden movements or positions

### Apparatus Required

* ESP32 Development Board
* MPU6050 Accelerometer and Gyroscope Sensor Module
* OLED Display Module (optional, for data visualization)
* Buzzer (for motion alert)

### PIN TABLE

| **MPU6050 Pin** | **ESP32 Pin** |
| --- | --- |
| **VCC** | 3.3V |
| **GND** | GND |
| **SCL** | GPIO22 (default SCL) |
| **SDA** | GPIO21 (default SDA) |
| **INT** | GPIO19 (Optional) |
| **AD0** | GND (default) |

### ESP32

The **ESP32** is a low-cost, energy-efficient microcontroller from Espressif that integrates both Wi-Fi and Bluetooth features. It has a dual-core 32-bit processor, robust peripheral interfaces, and is widely used for IoT and smart devices. Thanks to its performance and versatility, ESP32 is popular in home automation, wireless communication, and wearable projects.

### MPU6050

The MPU6050 is a compact 6-axis motion tracking device that combines a 3-axis gyroscope and a 3-axis accelerometer. It features an onboard Digital Motion Processor (DMP) for advanced sensor fusion algorithms and outputs data via an I2C interface. This module is widely used for measuring rotational velocity, acceleration, orientation, and displacement in various applications such as robotics and motion sensing

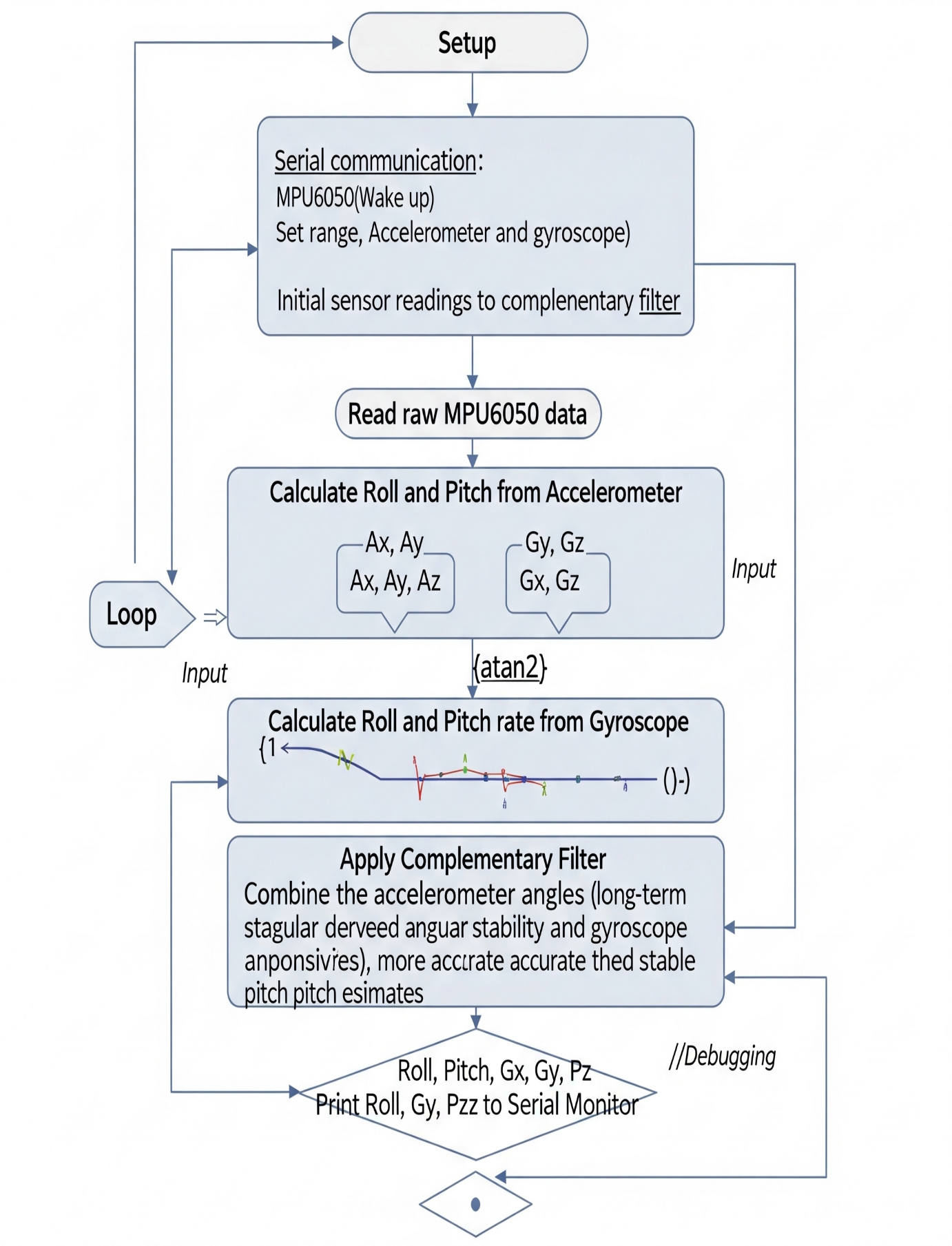
### OLED DISPLAY

An OLED display module is a self-illuminating screen that produces bright, high-contrast images by lighting each pixel individually. It is thin, lightweight, and usually communicates with microcontrollers via I2C or SPI interfaces. OLED displays are popular for their deep blacks, wide viewing angles, fast response, and low power consumption.

### BUZZER

A buzzer is an electronic device that produces sound as an audible alert or notification. It can be piezoelectric or electromagnetic, converting electrical signals into sound waves. Buzzers are widely used in alarms, timers, and user interface feedback

### FLOW CHART



### PROGRAM

#include <Wire.h>

#include <Adafruit\_MPU6050.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

Adafruit\_MPU6050 mpu;

// Buzzer pin

const int buzzerPin = 13; // Connect buzzer positive to GPIO 13

// OLED configuration

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define OLED\_RESET -1 // No reset pin

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

// Acceleration threshold for buzzer

const float accelerationThreshold = 10.0;

void setup() {

Serial.begin(115200);

pinMode(buzzerPin, OUTPUT);

digitalWrite(buzzerPin, LOW);

// OLED init

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Default I2C addr 0x3C

Serial.println("SSD1306 allocation failed");

while(1);

}

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

// MPU6050 init

if (!mpu.begin()) {

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

while (1);

}

Serial.println("MPU6050 Initialized");

mpu.setAccelerometerRange(MPU6050\_RANGE\_8\_G);

mpu.setGyroRange(MPU6050\_RANGE\_500\_DEG);

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

delay(100);

}

void loop() {

sensors\_event\_t a, g, temp;

mpu.getEvent(&a, &g, &temp);

// Calculate magnitude of acceleration vector

float accelMag = sqrt(a.acceleration.x \* a.acceleration.x +

a.acceleration.y \* a.acceleration.y +

a.acceleration.z \* a.acceleration.z);

// Show values on OLED

display.clearDisplay();

display.setCursor(0,0);

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

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

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

display.print("Magnitude: "); display.println(accelMag, 2);

display.display();

// Serial debug

Serial.print("Accel Mag: ");

Serial.println(accelMag);

// Buzzer alert

if (accelMag > accelerationThreshold) {

Serial.println("High motion detected! Buzzer ON");

digitalWrite(buzzerPin, HIGH);

} else {

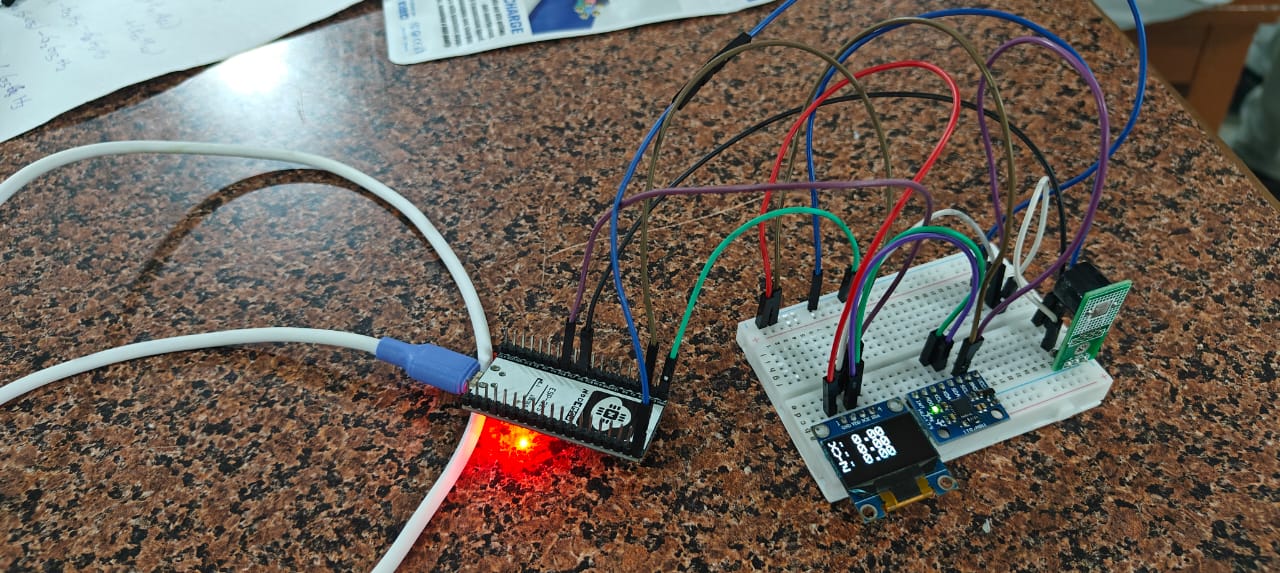
digitalWrite(buzzerPin, LOW);

}

delay(500);

}

### EXECUTION



### 