IoT BASED FALL DETECTION SYSTEM

WOKWI FILE

WOKWI link: https://wokwi.com/projects/368604930638147585

SOURCE CODE:

```
#include <Wire.h>
#include <Adafruit_MPU6050.h>
#include <WiFi.h>
#include <PubSubClient.h>
const int MPU_addr = 0x68;  // I2C address of the MPU-6050
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
float ax = 0, ay = 0, az = 0, gx = 0, gy = 0, gz = 0;
boolean fall = false;
boolean trigger1 = false;
boolean trigger2 = false;
boolean trigger3 = false;
byte trigger1count = 0;
byte trigger2count = 0;
byte trigger3count = 0;
int angleChange = 0;
// IBM Watson IoT Platform credentials
#define ORG "q6ie9a"//IBM ORGANITION ID
#define DEVICE_TYPE "Wokwi"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
//-------- Customise the above values --------
```

```
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send

char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


WiFiClient wifiClient;

PubSubClient client(server, 1883, wifiClient);


void setup() {

Serial.begin(115200);

  Wire.begin();

  Wire.beginTransmission(MPU_addr);

  Wire.write(0x6B);  // PWR_MGMT_1 register

  Wire.write(0);     // set to zero (wakes up the MPU-6050)

  Wire.endTransmission(true);

  Serial.println("Wrote to IMU");


  wificonnect();

  mqttconnect();

}

void loop() {

mpu_read();
```

```
ax = (AcX - 2050) / 16384.00;
//Serial.println(ax);
ay = (AcY - 77) / 16384.00;
//Serial.println(ay);
az = (AcZ - 1947) / 16384.00;
gx = (GyX + 270) / 131.07;
gy = (GyY - 351) / 131.07;
gz = (GyZ + 136) / 131.07;

float Raw_Amp = pow(pow(ax, 2) + pow(ay, 2) + pow(az, 2), 0.5);
int Amp = Raw_Amp * 10;
Serial.println(Amp);



if (Amp <= 2 && trigger2 == false) {
  trigger1 = true;
  Serial.println("TRIGGER 1 ACTIVATED");
}
if (trigger1 == true) {
  trigger1count++;
  if (Amp >= 12) {
    trigger2 = true;
    Serial.println("TRIGGER 2 ACTIVATED");
    trigger1 = false;
    trigger1count = 0;
  }
}
```

```
  if (trigger2 == true) {
    trigger2count++;
    angleChange = pow(pow(gx, 2) + pow(gy, 2) + pow(gz, 2), 0.5);
    if (angleChange >= 3 || Amp >= 15) {
      trigger3 = true;
      Serial.println("TRIGGER 3 ACTIVATED");
      trigger2 = false;
      trigger2count = 0;
    }
  }
  if (trigger3 == true) {
    trigger3count++;
    if (Amp <= 2 && trigger3count >= 4) {
      fall = true;
      Serial.println("FALL DETECTED");
      trigger3 = false;
      trigger3count = 0;
    } else if (Amp <= 2 && trigger3count < 4) {
      trigger3count = 0;
    }
  }
  if (Amp<=2) {
    PublishData("Fall detected!!");
    fall = false;
  }
  delay(100);
}
```

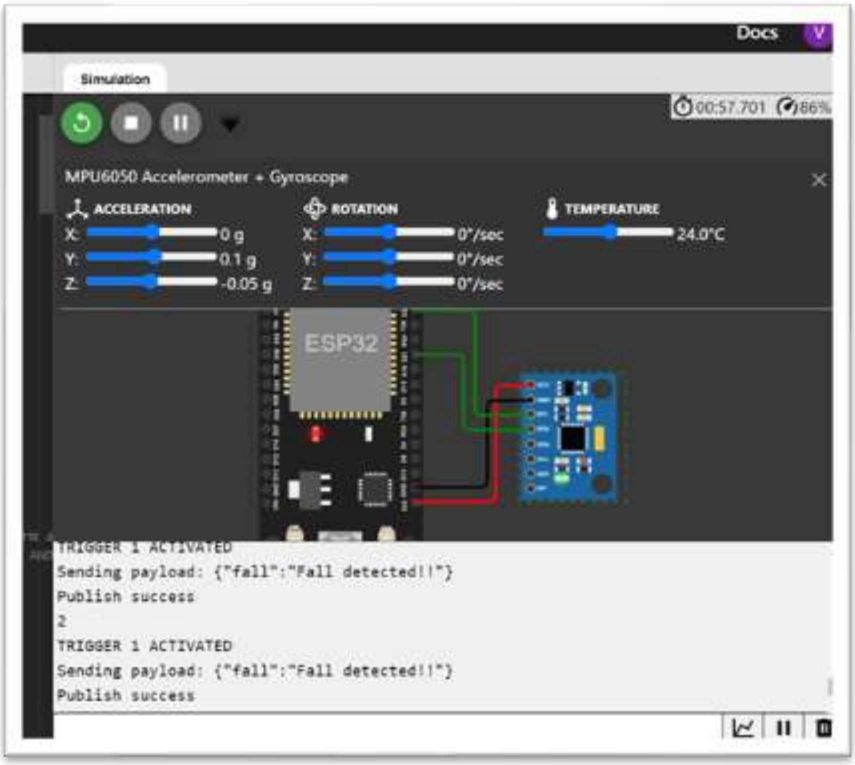
```
void mpu_read() {
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr, 14, true);  // request a total of 14 registers

  AcX = Wire.read() << 8 | Wire.read();  // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY = Wire.read() << 8 | Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ = Wire.read() << 8 | Wire.read();  // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp = Wire.read() << 8 | Wire.read();  // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX = Wire.read() << 8 | Wire.read();  // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY = Wire.read() << 8 | Wire.read();  // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ = Wire.read() << 8 | Wire.read();  // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
```

```
void PublishData(String fall) {
  mqttconnect();
  String payload = "{\"fall\":\"";
  payload += fall;
  payload += "\"}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish success");
  } else {
    Serial.println("Publish failed");
  }
  delay(2000);
}
```

RESULTS:



MPU6050 Accelerometer + Gyroscope

ACCELERATION
X: 0 g
Y: 0.1 g
Z: -0.05 g

ROTATION
X: 0°/sec
Y: 0°/sec
Z: 0°/sec

TEMPERATURE
24.0°C

```
TRIGGER 1 ACTIVATED
Sending payload: {"fall":"Fall detected!!"}
Publish success
2
TRIGGER 1 ACTIVATED
Sending payload: {"fall":"Fall detected!!"}
Publish success
```