

# MPU 6050 PROJECT

## LED Gesture Control Project Using MPU6050 & Arduino

مشروع التحكم بالإضاءة بحركات اليد

How to Control a Light with Hand Gestures Using MPU6050 , Arduino | (Turn On, Off), Adjust Brightness!

بإشراف الدكتور : عيد عبود.

تقدمة : محمد غليون & شريف غزاله عينية.

بإشراف الأستاذ : محمود حازم.

# INTRODUCTION :

- ◆ This project demonstrates an innovative way to control an LED using hand gestures.
- ◆ The \*MPU6050\* motion sensor detects hand movements, and the \*Arduino Uno\* processes the data to turn the LED (on/off) or (adjust its brightness) based on the gestures.

هذا المشروع يقدم طريقة مبتكرة للتحكم في إضاءة اللد باستخدام حركات اليد.

حيث يقوم حساس الحركة باكتشاف حركات اليد وتقوم لوحة الاردوينو بمعالجة البيانات لتشغيل/إطفاء اللد أو تعديل سطوعه بناءً على الحركات.

# PROJECT COMPONENTS & THEIR ROLES :

## ◆ Arduino Uno :

- The “main microcontroller” that reads sensor data and controls the LED.
- It processes signals from the MPU6050 and sends PWM signals to adjust LED brightness.

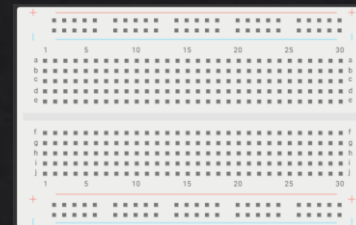
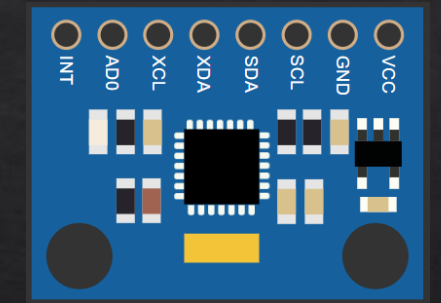
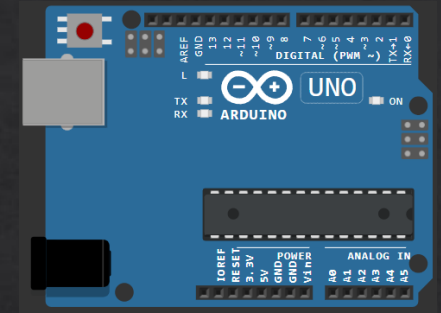
## ◆ MPU 6050 (Gyroscope + Accelerometer) :

- 6-axis motion sensor that detects :
  - ✓ Tilt (X, Y, Z angles) → Determines hand orientation.
  - ✓ Acceleration → Detects quick movements (e.g., shaking).
- Sends real-time data to Arduino for gesture recognition.

## ◆ LED : The output device that turns on/off or changes brightness based on gestures.

## ◆ Resistor : Protects the LED from excessive current (prevents burning out).

## ◆ Breadboard : Used for "easy prototyping" and connecting components without soldering.





# HOW IT WORKS ?

## Step 1: Motion Detection

- The MPU6050 continuously measures "orientation (pitch, roll, yaw)" and "acceleration".
- Example gestures:
  - ❖ Tilt [Left/Right] → Turn LED [ON/OFF].
  - ❖ Tilt [Up/Down] → [Increase/Decrease] brightness (PWM control).

## Step 2: Data Processing in Arduino

- Arduino reads "sensor values" and applies thresholds to detect gestures.
- Example logic:
  - ✓ `if (pitch > 30) { digitalWrite(LED, HIGH); } // Turn ON`
  - ✓ `else if (pitch < -30) { digitalWrite(LED, LOW); } // Turn OFF`
  - ✓ `if (roll > 30) { analogWrite(LED, brightness++); } // Brighter`
  - ✓ `if (roll < -30) { analogWrite(LED, brightness--); } // Dimmer`

## Step 3: LED Response

- Digital Control (ON/OFF)\*\* → Using `digitalWrite()`.
- Analog Control (Brightness)\*\* → Using `analogWrite()` (PWM).

## POTENTIAL APPLICATIONS :

- ❖ Smart Home Lighting – Control lights with hand movements.
  - ❖ Touchless Interfaces – Useful in hospitals or labs.
- ❖ Educational Tool – Learn about sensors, Arduino, and PWM.
- ❖ Gaming Controllers – Gesture-based input for DIY projects.

## FUTURE IMPROVEMENTS :

- ✓ Add More LEDs/RGB Control → Different colors for different gestures.
- ✓ Wireless Control (Bluetooth/Wi-Fi) → Use ESP8266/ESP32 for remote control.
  - ✓ Machine Learning – Train a model to recognize complex gestures.

## Project Code :

```
#include <Wire.h>
#include <MPU6050_light.h>

MPU6050 mpu(Wire); // Instantiate an MPU6050 object
int ledPin = 10; // Pin for the LED (PWM-capable pin)
float threshold = 0.3; // Gesture threshold to detect significant movement
int ledBrightness = 0; // Start with the LED off (brightness = 0)
bool ledOn;

void setup()
{
    Serial.begin(9600); // Start serial communication
    Wire.begin(); // Start I2C communication
    byte status = mpu.begin();
    Serial.print(F("MPU6050 status: "));
    Serial.println(status);
    while(status!=0){ } // stop everything if could not connect to MPU6050

    Serial.println(F("Calculating offsets, do not move MPU6050"));
    delay(1000);
    mpu.calcOffsets(true,true); // gyro and accelero
    Serial.println("Done!\n");

    pinMode(ledPin, OUTPUT); // Set LED pin as output
}
```

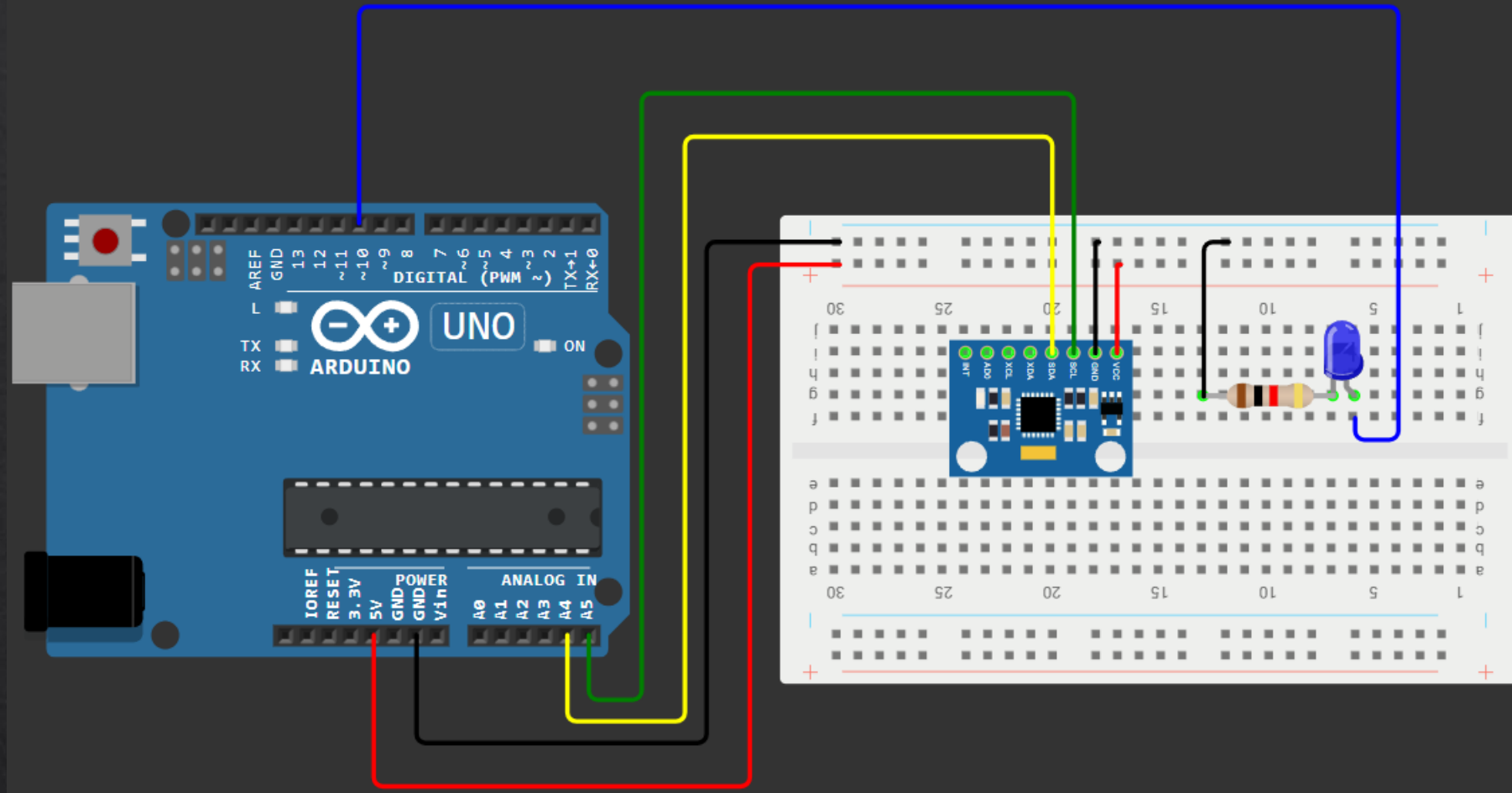
```
void loop()
{
    // Read raw accelerometer and gyroscope values
    mpu.update();
    float ax, ay, az;
    float rx, ry, rz;
    ax = mpu.getAccX(); ay = mpu.getAccY(); az = mpu.getAccZ();
    rx = mpu.getAngleX(); ry = mpu.getAngleY(); rz = mpu.getAngleZ();

    // Calculate the accelerometer values for tilt detection
    float tiltLeftRight = ax; // X-axis accelerometer value

    Serial.print("tiltLeftRight: ");
    Serial.println(tiltLeftRight);

    // Detect tilt gestures for left/right (to control on/off state)
    if (tiltLeftRight > threshold)
    {
        Serial.println("Tilt Left Detected");
        digitalWrite(ledPin, LOW); // Turn LED off
        ledOn = false;
    }
    else if (tiltLeftRight < -threshold)
    {
        Serial.println("Tilt Right Detected");
        digitalWrite(ledPin, HIGH); // Turn LED on
        ledOn = true;
    }
    if (ledOn)
    {
        ledBrightness = map(rz, -90, 90 , 0,255);
        analogWrite(ledPin, ledBrightness); // Update LED brightness
    }

    delay(100); // Debounce delay to avoid multiple triggers
}
```



## CONCLUSION :

This project is a (simple yet powerful) example of [sensor-based automation]. By combining “MPU6050 + Arduino”, we can create \*touchless control systems\* with endless possibilities for expansion.