# GESTURE BASED SECURE PASSWORD AUTHENTICATION SYSTEM

APPLICATION DEVELOPMENT

Submitted by

Pranay V - 2127230701104

In partial fulfillment for the course requirement of Robotics and Artificial Intelligence Club (RAIC)

in

ELECTRONICS AND COMMUNICATION ENGINEERING
SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University)
ANNA UNIVERSITY – CHENNAI – 600 025

MAY 2025

# **ABSTRACT**

This project presents a smart and secure gesture-controlled locking system using the MPU6050 motion sensor and Arduino Uno, enhanced with a lightweight machine learning approach. Instead of relying on traditional input methods like keypads or RFID, this system lets users unlock it by performing a specific series of hand gestures, such as "UP, RIGHT, LEFT." These movements are detected and interpreted using the accelerometer and gyroscope data from the MPU6050.

To improve accuracy and make the system more adaptable, we incorporated a simple form of machine learning. By collecting gesture samples and analyzing patterns, we created a basic classifier that can distinguish between different hand movements. This classifier is implemented directly on the Arduino in the form of optimized threshold checks, mimicking how an ML model would behave—while still being efficient for the hardware constraints.

When the user performs the correct gesture sequence, the system grants access by lighting a green LED and briefly sounding a buzzer. If the input is wrong, a red LED lights up with a warning sound, signaling access denial. The LCD display provides real-time feedback to guide the user through the process.

This project blends intuitive human interaction with embedded intelligence, creating a contactless and hygienic alternative to traditional locks. It's ideal for smart home applications, personal lockers, or places where touchless access is preferred. Future improvements could include real-time training of gestures, Wi-Fi integration, or expanding it to multiuser systems.

Detects specific hand gestures to unlock using motion patterns.



# **IMPLEMENTATION**

#### **INTRODUCTION**

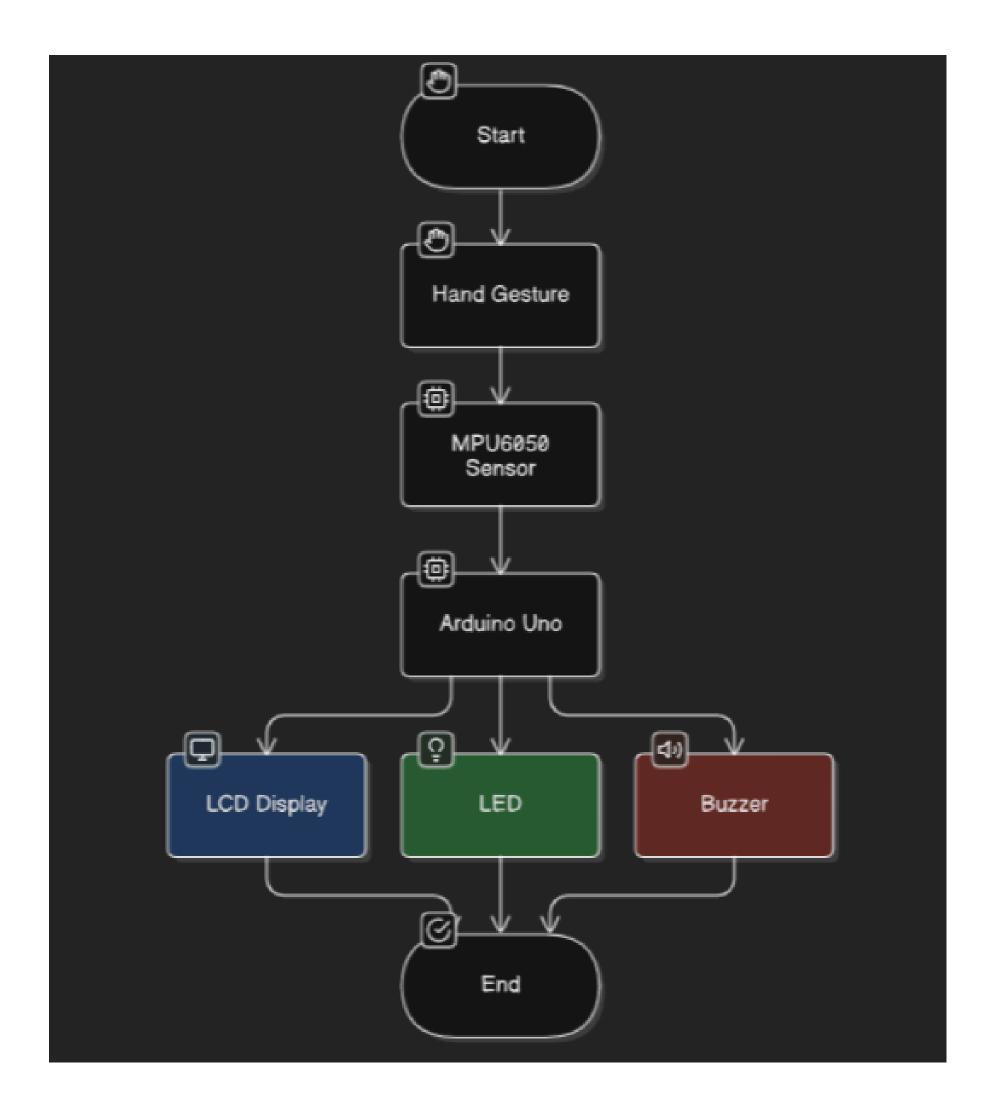
This project presents a gesture-controlled smart access system using the MPU6050 accelerometer and gyroscope sensor with Arduino Uno. It captures hand movements such as up, down, left, and right to form a gesture-based password. If the input gesture matches the predefined sequence, the system grants access, indicated by LEDs, an LCD display, and a buzzer. The project demonstrates a creative approach to secure authentication, integrating sensor data analysis, real-time feedback, and basic machine learning for improved gesture recognition and accuracy.

#### **TECHNOLOGY STACK**

In this project, we used an Arduino Uno as the main controller and an MPU6050 sensor to detect hand gestures through motion. A 16x2 I2C LCD display is used to show the input and access status, while LEDs and a buzzer provide visual and audio feedback. The project is programmed using Arduino C/C++ and tested using the Wokwi online simulator.

Component	Technology Used	Purpose
Microcontroller	Arduino Uno	To interface with MPU6050, process gestures, and control outputs
Sensor Module	MPU6050 (Accelerometer + Gyroscope)	To detect and interpret hand movements as gestures
Display Module	16x2 LCD (I2C)	To display access status and user feedback
Output Devices	Green & Red LEDs, Buzzer	Visual and audio feedback for gesture outcomes
Programming	Arduino IDE (C/C++)	Writing and uploading code to Arduino
Simulation Tool	Wokwi	To simulate hardware connections and test the code
ML Integration	Basic gesture classification / ML models	For learning and recognizing complex gesture patterns

#### **BLOCK DIAGRAM**



#### 1. Start:

The system begins when powered on or reset. It enters an active state, waiting for gesture input.

#### 2. Hand Gesture

- The user performs a specific hand movement in space.
- This movement is the core input used to control the system.

#### 3. MPU6050 Sensor

- The MPU6050 detects the hand's movement using its accelerometer and gyroscope.
- It measures motion data like orientation, tilt, and velocity along X, Y, and Z axes.
- These raw values are sent to the microcontroller.

#### 4. Arduino Uno:

- Arduino acts as the brain of the system.
- It receives sensor data and processes it using gesture recognition logic (could include threshold-based or ML model).
- Based on the recognized gesture, it determines if the gesture matches the predefined pattern/password.

#### 5. LCD Display:

Shows real-time feedback like:

- "Gesture Detected"
- "Access Granted" or "Access Denied"
- Captured input sequence

#### 6. LED:

- Two LEDs (typically green and red) give visual output:
- Green LED: Access granted (correct gesture sequence)
- Red LED: Access denied (wrong sequence)

#### 7. Buzzer:

- Provides auditory feedback:
- A short beep on correct gesture.
- A long/low-pitched beep on wrong gesture or denied access.

#### 8. End:

The system resets to the initial state, ready for the next input or interaction. It waits again for a new gesture input.

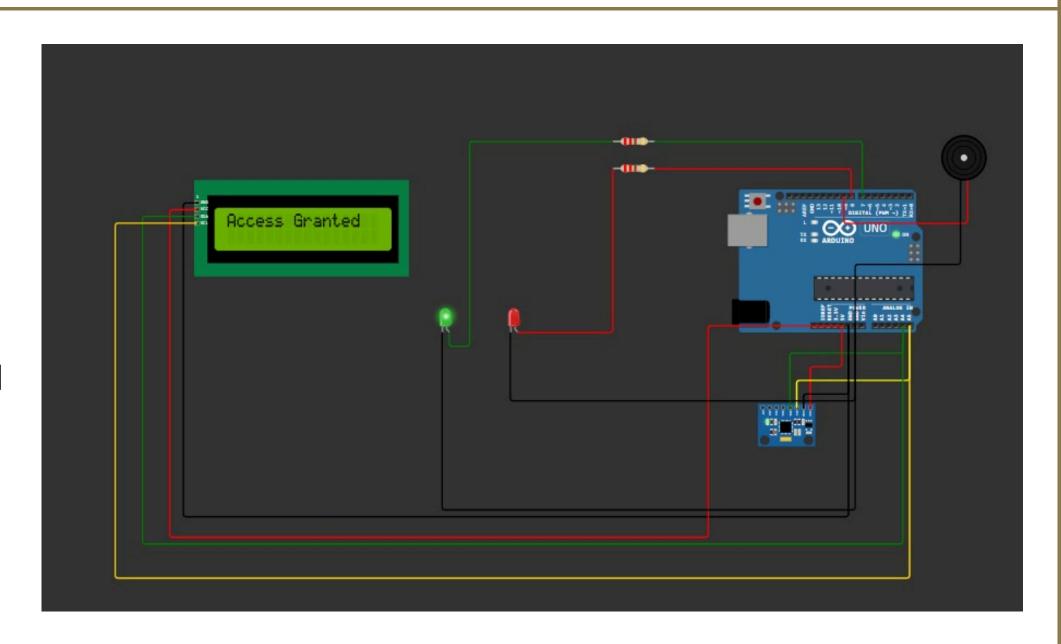
# **WORKING OF MODEL:**

# Scenario 1: Correct Gesture Sequence

- ✓ If the input gesture matches the predefined password (e.g., "UPRIGHTLEFT"):
  - Green LED (pin 7) → Turns ON
  - Red LED (pin 8) → OFF
  - Buzzer → Emits a short high-pitched tone (1000 Hz for 0.5 sec)
  - LCD → Displays "Access Granted"

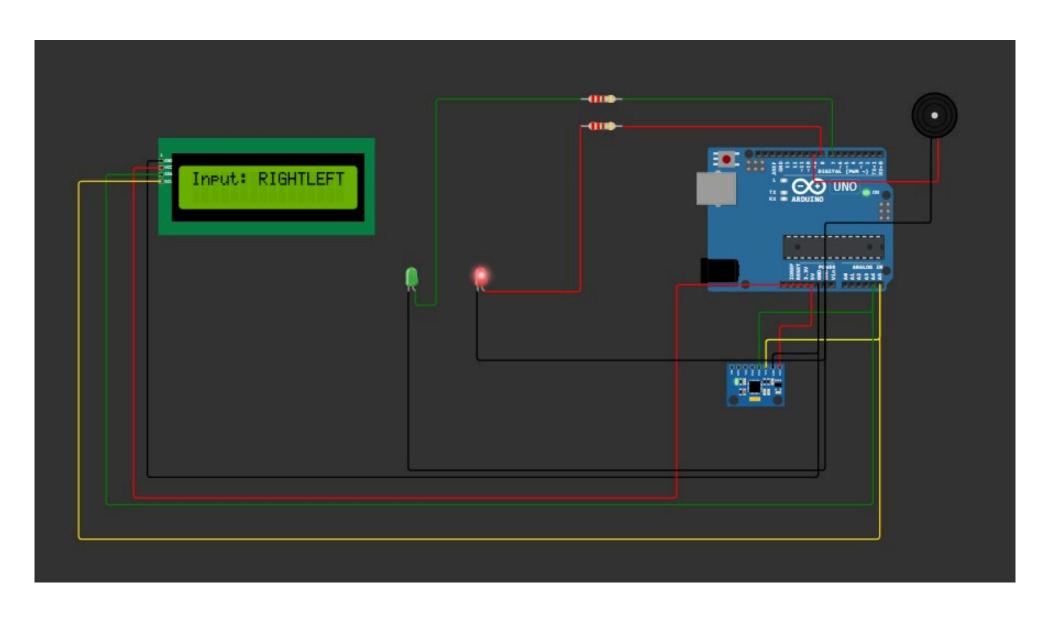
# Scenario 1: Correct Gesture Sequence

✓ If the input gesture matches the predefined password



# **Scenario 2: Wrong Gesture Sequence**

- X If the input gesture does NOT match the password:
  - Green LED → OFF
  - Red LED → Turns ON
  - Buzzer → Emits a long low-pitched tone (200 Hz for 1 sec)
  - LCD → Displays "Access Denied"



# Scenario 2: Wrong Gesture Sequence

X If the input gesture does NOT match the password:

# So Generally,

Condition	Green LED	Red LED	Buzzer	LCD Display
Correct Gesture Match	ON	OFF	Short beep	"Access Granted"
Incorrect Gesture	OFF	ON	Long beep	"Access Denied"

### **PROGRAM**

#### **HARDWARE SIMULATION**

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <MPU6050.h>
MPU6050 mpu;
LiquidCrystal_I2C lcd(0x27, 16, 2);
const int greenLED = 7;
const int redLED = 8;
const int buzzer = 9;
String password = "UPRIGHTLEFT";
String inputGesture = "";
void setup() {
 Serial.begin(9600);
 Wire.begin();
 lcd.init();
 lcd.backlight();
 pinMode(greenLED, OUTPUT);
 pinMode(redLED, OUTPUT);
 pinMode(buzzer, OUTPUT);
 mpu.initialize();
 lcd.setCursor(0, 0);
 lcd.print("Gesture Lock");
 delay(2000);
 lcd.clear();
void loop() {
 int16_t ax, ay, az;
 mpu.getAcceleration(&ax, &ay, &az);
```

```
if (ax > 15000) {
     addGesture("RIGHT"); }
else if (ax < -15000) {
     addGesture("LEFT"); }
else if (ay > 15000) {
     addGesture("UP"); }
else if (ay < -15000) {
     addGesture("DOWN"); }
if (inputGesture.length() >=
password.length()){
lcd.clear();
if (inputGesture == password) {
lcd.print("Access Granted");
digitalWrite(greenLED, HIGH);
digitalWrite(redLED, LOW);
tone(buzzer, 1000, 500); }
else {
lcd.print("Access Denied");
digitalWrite(greenLED, LOW);
digitalWrite(redLED, HIGH);
tone(buzzer, 200, 1000); }
delay(3000);
inputGesture = "";
lcd.clear();
delay(300);
void addGesture(String direction) {
if (!inputGesture.endsWith(direction)) {
inputGesture += direction;
lcd.setCursor(0, 0);
lcd.print("Input: ");
lcd.setCursor(7, 0);
lcd.print(inputGesture);
```

# MACHINE LEARNING (ML) INTEGRATION

DATASET:	gesture	password	l.csv
----------	---------	----------	-------

	Α	В	С	D
1	ax	ay	az	label
2	-17000	2500	16000	LEFT
3	18000	-1000	15800	RIGHT
4	1000	17000	14000	UP
5	-500	-18000	14500	DOWN
6	18000	3000	15500	RIGHT
7	-16500	1500	16200	LEFT
8	1200	16000	13900	UP
9	-1000	-17500	14200	DOWN

#### JUPYTER NOTEBOOK IMPLEMENTATION

```
[9]: import pandas as pd
      from sklearn.model_selection import train_test_split
      from sklearn.neighbors import KNeighborsClassifier
      from sklearn.metrics import classification_report, accuracy_score
      import joblib
[10]: data = pd.read_csv("gesture_data.csv")
[11]: X = data[['ax', 'ay', 'az']]
      y = data['label']
[12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
[13]: model = KNeighborsClassifier(n_neighbors=3)
      model.fit(X_train, y_train)
[13]:
             KNeighborsClassifier
      KNeighborsClassifier(n_neighbors=3)
[14]: y_pred = model.predict(X_test)
       print("Accuracy:", accuracy_score(y_test, y_pred))
      print("Report:\n", classification_report(y_test, y_pred))
       Accuracy: 0.0
       Report:
                     precision
                                  recall f1-score
                                                    support
              LEFT
                         0.00
                                   0.00
                                             0.00
                                                        1.0
              RIGHT
                         0.00
                                   0.00
                                             0.00
                                                        1.0
                         0.00
                                   0.00
                                             0.00
                                                        0.0
          accuracy
                                             0.00
                                                        2.0
          macro avg
                         0.00
                                   0.00
                                             0.00
                                                        2.0
       weighted avg
                         0.00
                                   0.00
                                             0.00
                                                        2.0
[15]: joblib.dump(model, "gesture_model.pkl")
[15]: ['gesture_model.pkl']
```

# Advantages

- Contactless Authentication Enables secure access using simple hand gestures without physical input devices.
- **Cost-Effective** Utilizes affordable components like MPU6050, Arduino, and LCD, making it budget-friendly.
- User-Friendly Easy to operate and understand, suitable for various age groups.
- **Real-Time Processing** Provides instant feedback via LEDs, LCD, and buzzer based on gesture input.
- Extendable Design Can be upgraded with more complex gestures or integrated with other ML models for smart automation.

# **Challenges**

- **Sensor Sensitivity & Noise:** The MPU6050 sensor is prone to capturing unwanted movements or vibrations, which can reduce the accuracy of gesture detection.
- Limited Dataset for Training: A small or imbalanced dataset can make it hard for the ML model to generalize, leading to incorrect gesture classification.

# **Conclusion**

This gesture-based security system leverages the MPU6050 sensor and Arduino to recognize hand gestures for authentication. By combining hardware components with a simple machine learning model, it effectively identifies motion patterns and verifies them against a predefined password. Visual and audio feedback through LEDs, an LCD display, and a buzzer ensures a clear user experience. The project not only enhances the way we interact with devices but also opens doors to contactless, efficient, and affordable access control systems. It demonstrates the potential of integrating embedded systems with machine learning for real-world smart applications in automation and security.

# **References**

- Hand Gesture Controlled Robot using Arduino and MPU6050 <u>Phttps://www.researchgate.net/publication/363765370\_Hand\_Gesture\_Controlled Robot\_using\_Arduino\_and\_MPU6050</u>
- Gesture-based mouse control system based on MPU6050 and Kalman filter <a href="https://www.researchgate.net/publication/370379787">https://www.researchgate.net/publication/370379787</a> Gesture-based mouse control system based on MPU6050 and Kalman filter technique
- Hand Gestures Controlled Robot using Arduino <a href="https://www.researchgate.net/publication/351765928\_Hand\_Gestures\_Controlled\_Robot\_using\_Arduino">https://www.researchgate.net/publication/351765928\_Hand\_Gestures\_Controlled\_Robot\_using\_Arduino</a>
- Machine Learning for Gesture Recognition (IITM Thesis)
   <a href="https://eescholars.iitm.ac.in/sites/default/files/eethesis/ee19m025.pdf">https://eescholars.iitm.ac.in/sites/default/files/eethesis/ee19m025.pdf</a>
- Hand Gesture Recognition Robot Using IoT Techniques
- <u>https://tijer.org/tijer/papers/TIJER2405132.pdf</u>
- Hand Gesture Recognition for Game Control
   <a href="https://www.jetir.org/papers/JETIRFX06090.pdf">https://www.jetir.org/papers/JETIRFX06090.pdf</a>
- Gesture Control Device Build Manual IIT Madras <u>Antips://respark.iitm.ac.in/wp-content/uploads/2023/05/Build-Manual-Gesture-Control-Device-V1.1\_10May23-1.pdf</u>
- Low-Power Gesture Recognition Using Accelerometers (ETH Zurich)
   https://pub.tik.ee.ethz.ch/students/2014-FS/SA-2014-21.pdf
- Gesture Controlled Mecanum Wheel Car Using ESP32 and MPU6050 <u>https://www.jetir.org/papers/JETIR2408587.pdf</u>
- A Novel Accelerometer-Based Gesture Recognition System
   <u>https://www.researchgate.net/publication/220323033\_A\_Novel\_Accelerometer-Based\_Gesture\_Recognition\_System</u>
- A Hand Gestures Recognition System for Industrial Robot Control <a href="https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12599/125992I/A-hand-gestures-recognition-system-for-industrial-robot-control/10.1117/12.2673362.full">https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12599/125992I/A-hand-gestures-recognition-system-for-industrial-robot-control/10.1117/12.2673362.full</a>

# **References**

- Gesture to Speech Using Flex Sensors and MPU6050
- https://www.ijeat.org/wp-content/uploads/papers/v8i6/F9167088619.pdf
- IMU Based Hand Gesture Recognition using Arduino and Python <a href="https://www.researchgate.net/publication/353267647\_IMU\_Based\_Hand\_Gesture\_Recognition\_using\_Arduino\_and\_Python">https://www.researchgate.net/publication/353267647\_IMU\_Based\_Hand\_Gesture\_Recognition\_using\_Arduino\_and\_Python</a>
  - Gesture Recognition with ML and OpenCV (GitHub)
- https://github.com/simongeek/hand-gesture-recognition
- Gesture Controlled System Using MPU6050 with Arduino and Python <a href="https://www.hackster.io/news/gesture-controlled-system-using-mpu6050-and-python-25cbf6">https://www.hackster.io/news/gesture-controlled-system-using-mpu6050-and-python-25cbf6</a>
  - Real-Time Hand Gesture Recognition for Smart Home Automation
- <u>https://www.jetir.org/papers/JETIR2109274.pdf</u>
- ML-Based Hand Gesture Recognition for Wearables MDPI Sensors
- https://www.mdpi.com/1424-8220/21/4/1240
- Mid-Air Gesture and Voice Alert System
- <u>Anttps://ijcrt.org/papers/IJCRT2104495.pdf</u>
  - Wireless Gesture Controlled Robot
- https://www.ijfans.org/uploads/paper/9571ab0e2146abac4b320f0962486839.pdf
- Gesture Keyboard Using Arduino
- https://ijcrt.org/papers/IJCRT2105569.pdf
- Hand Gesture Recognition for Human-Machine Interface MDPI Sensors
- https://www.mdpi.com/1424-8220/19/18/3827
- Gesture Controlled Rocker-Bogie System KSCST

  https://www.kscst.org.in/spp/46 series/46s spp/02 Exhibition Projects/311 46S BE
  2310.pdf
- Accelerometer-Based Gesture Control Robot Car RCCIIT
- https://rcciit.org.in/students\_projects/projects/aeie/2019/GR4.pdf