Hand Gesture Controlled LED System Manual

Project Overview :

This project allows you to control **five LEDs** connected to an **Arduino Uno** using **hand gestures** detected via **OpenCV and MediaPipe** in **Visual Studio Code**. It's a fun and interactive project that combines computer vision with hardware control.

Software Required:

- **Arduino IDE** (for uploading code to Arduino)
- **Visual Studio Code (VS Code)** (for running Python script)
- **Python 3**
- **OpenCV**
- **MediaPipe**

Hardware Required:

- Arduino Uno
- USB Cable
- 5 LEDs
- 5 Resistors (220 ohm)
- Jumper wires
- Breadboard

---

software Installation Guide :

Install Arduino IDE:

1. Go to: https://www.arduino.cc/en/software
2. Download the IDE for your system and install it.
3. Open Arduino IDE and select your board: Tools > Board > Arduino Uno.
4. Select Port: Tools > Port > COMx

Install Python and Required Libraries:

1. Download Python: https://www.python.org/downloads/
2. Open terminal (CMD) and run:

pip install opencv-python mediapipe pyserial

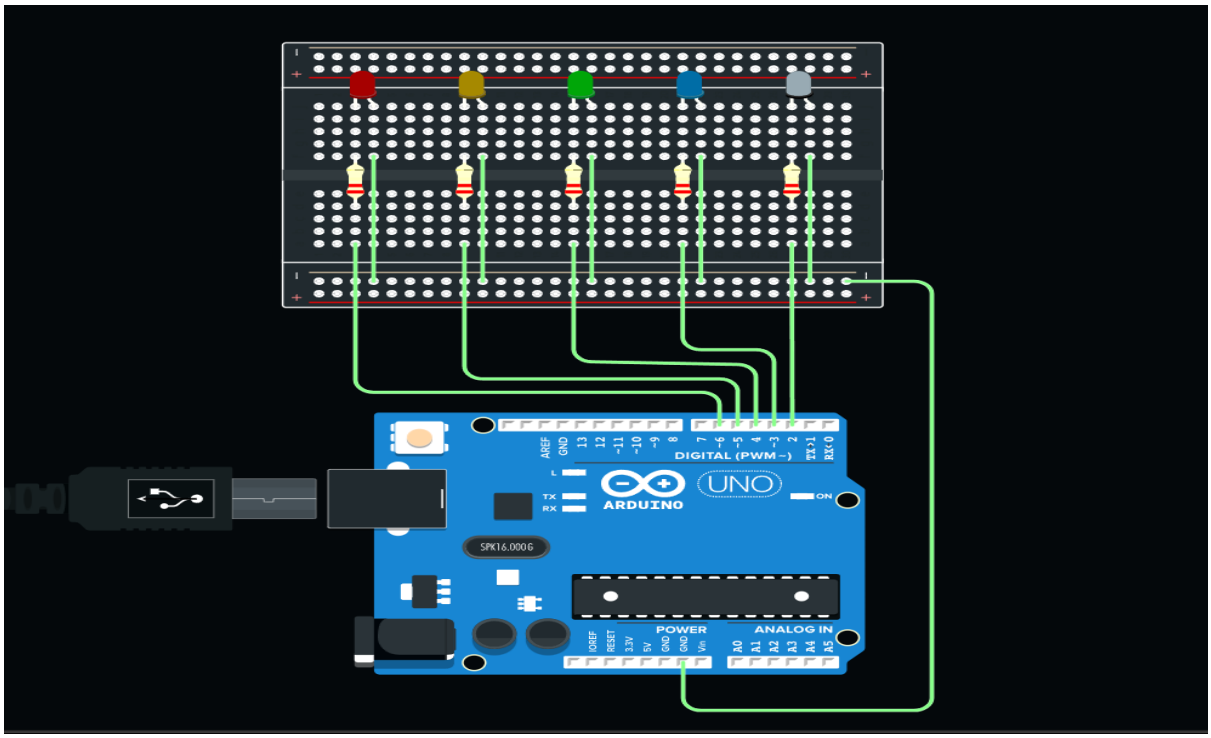
Install Visual Studio Code:

1. Download from: https://code.visualstudio.com/
2. Install Python extension from extensions tab.

Wiring the LEDs:

- Connect:
    - LED 1: Pin 2
    - LED 2: Pin 3
    - LED 3: Pin 4
    - LED 4: Pin 5
    - LED 5: Pin 6
- The short leg of each LED is connected to GND, and the long leg of each LED is connected to the respective pins through a 220-ohm resistor.

Circuit Diagram :



Arduino Code :

```
void setup() {
  Serial.begin(9600);
  for (int i = 2; i <= 6; i++) {
    pinMode(i, OUTPUT);
  }
}

void loop() {
  if (Serial.available()) {
    char ch = Serial.read();
    int count = ch - '0';

    for (int i = 0; i < 5; i++) {
      if (i < count) {
        digitalWrite(2 + i, HIGH);
      } else {
        digitalWrite(2 + i, LOW);
      }
    }
  }
}
```

Arduino Code Explanation:

- Serial.begin(9600): Starts serial communication.
- pinMode(i, OUTPUT): Sets pins 2 to 6 as output.
- Serial.read(): Reads the number of fingers sent from Python.
- count = ch - '0': Converts ASCII character to number.
- Turns ON number of LEDs equal to fingers shown.

Python Code (OpenCV + MediaPipe) :

```
import cv2

import mediapipe as mp

import serial

import time


arduino = serial.Serial('COM7', 9600)

time.sleep(2)


mp_hands = mp.solutions.hands

hands = mp_hands.Hands(max_num_hands=1, min_detection_confidence=0.7,
min_tracking_confidence=0.7)

mp_draw = mp.solutions.drawing_utils


finger_tips = [4, 8, 12, 16, 20]

cap = cv2.VideoCapture(0)


prev_count = -1


while True:

    ret, frame = cap.read()

    if not ret:

        break


    frame = cv2.flip(frame, 1)

    h, w, _ = frame.shape

    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    results = hands.process(rgb)


    finger_count = 0
```

```
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            lm_list = [(int(lm.x * w), int(lm.y * h)) for lm in hand_landmarks.landmark]

            if lm_list[finger_tips[0]][0] > lm_list[finger_tips[0] - 1][0]:
                finger_count += 1

            for id in finger_tips[1:]:
                if lm_list[id][1] < lm_list[id - 2][1]:
                    finger_count += 1

            mp_draw.draw_landmarks(frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)

    if finger_count != prev_count:
        arduino.write(str(finger_count).encode())
        prev_count = finger_count

    cv2.putText(frame, f"Fingers: {finger_count}", (10, 60),
                cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0, 255, 0), 3)

    cv2.imshow("Hand Gesture LED Control", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
arduino.close()
```

Python Code Explanation:

- Uses MediaPipe to detect hands and finger landmarks.
- Counts how many fingers are up by comparing coordinates.
- Sends the count to Arduino via serial.
- cv2.imshow: Displays video feed with finger count.
- arduino.write(): Sends finger count to Arduino.

How It Works:

1. Python detects your hand and counts fingers.
2. Sends number to Arduino.
3. Arduino lights up that many LEDs.

How to Run:

1. Upload Arduino code to Uno using Arduino IDE.
2. Run Python script in VS Code.
3. Show fingers to webcam to control LEDs.
