

```

from cvzone.HandTrackingModule import HandDetector
import cv2
import math
import serial
import time

arduino = serial.Serial('COM3', 9600) # e.g., '/dev/ttyUSB0' on Linux/Mac
time.sleep(2)

def set_servo_angle(angle_one, angle_two, angle_three):
    if 0 <= angle_one and angle_two and angle_three <= 180:
        command_one = f"{angle_one}\n"
        command_two = f"{angle_two}\n"
        command_three = f"{angle_three}\n"
        arduino.write(command_one.encode())
        arduino.write(command_two.encode())
        arduino.write(command_three.encode())
        print(f"Sent angle: {angle_one},{angle_two},{angle_three}")
    else:
        print("Angle must be between 0 and 180")

cap = cv2.VideoCapture(0)
detector = HandDetector(detectionCon=0.8, maxHands=1)

def find_angle(p1, p2, p3):
    # Calculates the angle at p2 using the cosine rule
    a = math.dist(p2, p3)
    b = math.dist(p1, p3)
    c = math.dist(p1, p2)

    # Avoid division by zero
    if a * b == 0:
        return 0

    angle = math.acos((a**2 + c**2 - b**2) / (2 * a * c))
    return math.degrees(angle)

while True:
    success, img = cap.read()
    hands, img = detector.findHands(img)

    if hands:
        hand = hands[0]
        lmList = hand["lmList"] # List of 21 landmarks

        # Example: Index finger joint angle
        p1 = lmList[1] # MCP (base joint)

```

```

p2 = lmList[2] # PIP (middle joint)
p4 = lmList[4] # DIP (near tip)
p16 = lmList[16]
p14 = lmList[14]
p13 = lmList[13]
p8 = lmList[8]
p6 = lmList[6]
p5 = lmList[5]

felix = find_angle(p1, p2, p4)
joe = find_angle(p16, p14, p13)
marcus = find_angle(p8, p6, p5)

set_servo_angle(int(felix - abs(felix - 180)),int(joe),int(marcus))

cv2.putText(img, f'Angle: {int(felix)} deg', (10, 70),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 255), 2)

cv2.putText(img, f'Angle: {int(joe)} deg', (10, 70),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 255), 2)

cv2.putText(img, f'Angle: {int(marcus)} deg', (10, 70),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 255), 2)

print(f"Thumb Angle: {felix:.2f}")
print(f"Finger One Angle: {joe:.2f}")
print(f"Finger Two Angle: {marcus:.2f}")

cv2.imshow("Image", img)
if cv2.waitKey(1) == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()

```