

## Codes

### Arduino Code

```
#include <Arduino.h>

#include <Servo.h>

Servo servothumb;
Servo servoidex;
Servo servomiddle;
Servo servoring;
Servo servopinky;

char number[50];
char c;
int state = 0;
String myStringRec;
int stringCounter = 0;
bool stringCounterStart = false;
String myRevivedString;
int stringLength = 6;

int servoPinky=0,servoMiddle=0,servoIndex=0,servoThumb=0,servoRing=0;
int myVals[] = {0,0,0,0,0} ;

int myButton= 1;
bool portable;

void alloFF(){

servopinky.write(0);
servoidex.write(0);
servomiddle.write(0);
servothumb.write(0);
servoring.write(0);
}
void allON(){

servopinky.write(180);
servoidex.write(180);
servomiddle.write(180);
servothumb.write(180);
servoring.write(180);
}

void receiveData() {
  int i = 0;
  while (Serial.available()) {
```

```

    if (c == '$') {
        stringCounterStart = true;
    }
    if (stringCounterStart == true )
    {
        if (stringCounter < stringLength)
        {
            myRevivedString = String(myRevivedString + c);
            stringCounter++;
        }
        if (stringCounter >= stringLength) {
            stringCounter = 0; stringCounterStart = false;
            servoPinky = myRevivedString.substring(1, 2).toInt();
            servoRing = myRevivedString.substring(2, 3).toInt();
            servoMiddle = myRevivedString.substring(3, 4).toInt();
            servoIndex = myRevivedString.substring(4, 5).toInt();
            servoThumb = myRevivedString.substring(5, 6).toInt();
            myRevivedString = "";
        }
    }
}

void setup() {

    Serial.begin(9600);
    servothumb.attach(9);
    servoindex.attach(10);
    servopinky.attach(11);
    servoring.attach(12);
    servomiddle.attach(13);
    myButton = analogRead(A0);
    delay(500);

}

void loop() {
    receiveData();
    if (servoPinky ==1){ servopinky.write(180);}else{servopinky.write(0);}
    if (servoIndex ==1){ servoindex.write(180);}else{servoindex.write(0);}
    if (servoMiddle ==1){ servomiddle.write(180);}else{servomiddle.write(0);}
    if (servoThumb ==1){ servothumb.write(180);}else{servothumb.write(0);}
    if (servoRing ==1){ servoring.write(180);}else{servoring.write(0);}
}

```

## Python Code

```
from cvzone.HandTrackingModule import HandDetector
import cv2
from cvzone.SerialModule import SerialObject

# Initialize the webcam to capture video
# The '2' indicates the third camera connected to your computer; '0' would
usually refer to the built-in camera
cap = cv2.VideoCapture(0)

# Initialize the HandDetector class with the given parameters
detector = HandDetector(staticMode=False, maxHands=1, modelComplexity=1,
detectionCon=0.7, minTrackCon=0.5)

arduino = SerialObject(portNo="COM8", baudRate=9600, digits=1, max_retries=5)
# CHANGE PORT NO

# Continuously get frames from the webcam
while True:
    # Capture each frame from the webcam
    # 'success' will be True if the frame is successfully captured, 'img' will
contain the frame
    success, img = cap.read()

    # Find hands in the current frame
    # The 'draw' parameter draws landmarks and hand outlines on the image if
set to True
    # The 'flipType' parameter flips the image, making it easier for some
detections
    hands, img = detector.findHands(img, draw=True, flipType=True)

    # Check if any hands are detected
    if hands:

        # Information for the first hand detected
        hand1 = hands[0] # Get the first hand detected
        lmList1 = hand1["lmList"] # List of 21 landmarks for the first hand
        bbox1 = hand1["bbox"] # Bounding box around the first hand (x,y,w,h
coordinates)
        center1 = hand1['center'] # Center coordinates of the first hand
        handType1 = hand1["type"] # Type of the first hand ("Left" or
"Right")

        # Count the number of fingers up for the first hand
        fingers1 = detector.fingersUp(hand1)
        print(fingers1)
```

```
    arduino.sendData(fingers1)

    print(" ") # New line for better readability of the printed output

# Display the image in a window
cv2.imshow("Image", img)

# Keep the window open and update it for each frame; wait for 1
millisecond between frames
cv2.waitKey(1)
```