



DIGITAL COMMUNICATION AND SYSTEMS
GESTURE CONTROL USING ARDUINO AND PYTHON

MALIHA FATIMA (CS-17085)

SYED MUHAMMAD IMRAN (CS-17102)

Department of Computer and Information Systems NED
University of Engineering and Technology

Submitted to Sir Umer Iftikhar

December 24, 2019

Abstract

The principle behind the Arduino based Hand Gesture Control of Computer is actually very simple. We have used two Ultrasonic Sensors with Arduino, place your hand in front of the Ultrasonic Sensor and calculate the distance between the hand and the sensor. Using this information, relevant actions in the computer can be performed.

The position of the Ultrasonic Sensors is very important. Two Ultrasonic Sensors are on the top of a laptop screen. The distance information from Arduino is collected by a Python Program and a special library called PyAutoGUI will convert the data into keyboard click actions.

INTRODUCTION:

You might have seen Hand Gesture Controlled Robots, where the motion of a robot is controlled by the gestures of the hand. Another interesting project based on a similar principle is an Arduino based Hand Gesture Control of your computer or laptop. In this project, we have implemented a simple Arduino based hand gesture control where you can control few functions of your web browser like switching between tabs, scrolling up and down in web pages, shift between tasks (applications), and increase or decrease the volume (in VLC Player) with the help of hand gestures.

We normally use LED Indicators, Switches, Touch Screens and LCD Displays as a part of devices. Another way to communicate with machines like Robots or Computers is with the help of Hand Gestures.

COMPONENTS:

- Arduino UNO
- Ultrasonic Sensors
- USB Cable (for Arduino)
- Few Connecting Wires
- A Laptop with internet connection

FUNCTIONS:

- Switch to Next Tab in a Web Browser
- Switch to Next Tab in a Web Browser
- Scroll Down in a Web Page
- Scroll Up in a Web Page
- Switch between two Tasks (Chrome and VLC Player)
- Increase Volume
- Decrease Volume

- **GESTURES:**

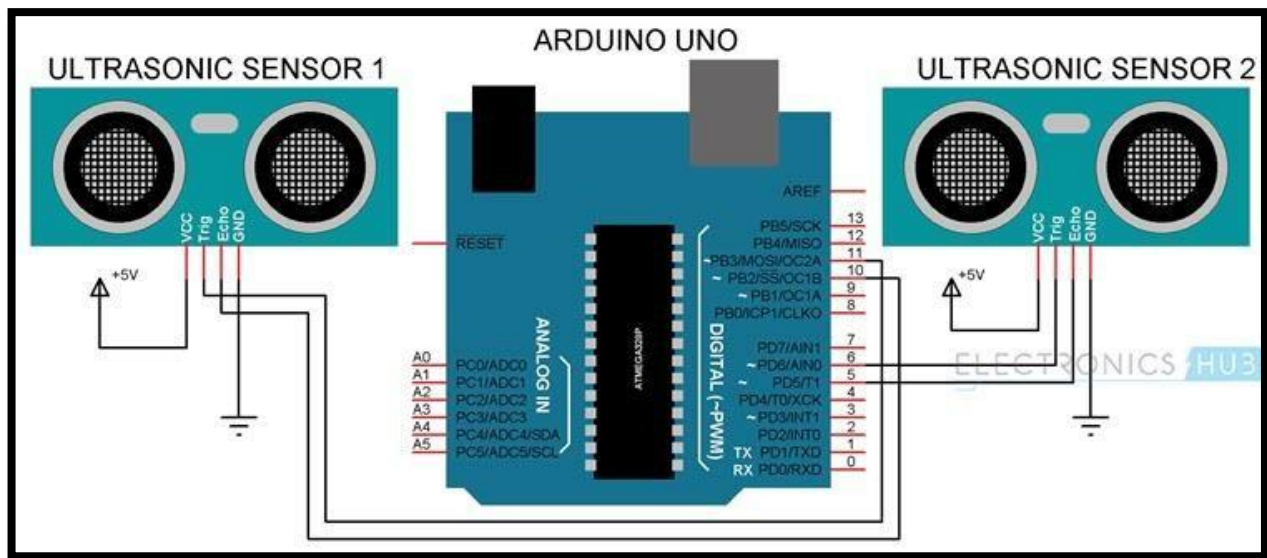
The following are the 5 different hand gestures or actions that I've programmed for demonstration purpose.

- **Gesture 1:** Place your hand in front of the Right Ultrasonic Sensor at a distance (between 15CM to 30CM) for a small duration and move your hand away from the sensor. This gesture will Scroll Down the Web Page or Decrease the Volume.
- **Gesture 2:** Place your hand in front of the Right Ultrasonic Sensor at a distance (between 15CM to 30CM) for a small duration and move your hand towards the sensor. This gesture will Scroll up the Web Page or Increase the Volume.
- **Gesture 3:** Swipe your hand in front of the Right Ultrasonic Sensor. This gesture will move to the Next Tab.
- **Gesture 4:** Swipe your hand in front of the Left Ultrasonic Sensor. This gesture will move to the Previous.
- **Gesture 5:** Swipe your hand across both the sensors (Left Sensor first). This action will Switch between Tasks.

Key functions:

- "next" Action = Ctrl+PgDn
- "previous" Action = Ctrl+PgUp
- "down" Action = Down Arrow
- "up" Action = Up Arrow
- "change" Action = Alt+Tab
- Keypress = Up Arrow = Increase Volume
- Keypress = Down Arrow = Decrease Volume

CIRCUIT DIAGRAM:



Conclusion:

In this project, we have implemented Arduino based Hand Gesture Control of Your Computer, where few hand gestures made in front of the computer will perform certain tasks in the computer without using mouse or keyboard. This type of hand gesture control of computers can be used for VR (Virtual Reality), AR (Augmented Reality), 3D Design, Reading Sign Language, etc.

ARDUINO CODE:

```
//Setting pins

const int vcc = 13;

const int trigPin1 = 11; // trigger o/p sensr 1

const int echoPin1 = 10; // Echo i/p sensr 1

const int trigPin2 = 6; // trigger o/p sensr 2

const int echoPin2 = 5; // echo i/p sensr 2

// Considering Variables

long duration;

int distance1, distance2;

float r;

unsigned long temp=0;

int temp1=0;

int l=0; // for special purpose of identification

//Defining function for distance

void find_distance (void);

// main function of distance

void find_distance (void)

{
```

```
// left sensor

digitalWrite(trigPin1, LOW); // low for 0 //
high for 1 // delay in ms

delayMicroseconds(2);

digitalWrite(trigPin1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin1, LOW);

// time calculation of receiving signal

duration = pulseIn(echoPin1, HIGH, 5000);

r = 3.4 * duration / 2; // calculation
in cm because cm mein hoti hai 340/100

distance1 = r / 100.00;

// Working of right sensor

digitalWrite(trigPin2, LOW);

delayMicroseconds(2);

digitalWrite(trigPin2, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin2, LOW);

duration = pulseIn(echoPin2, HIGH, 5000);

r = 3.4 * duration / 2;
```

```

distance2 = r / 100.00;

delay(100);
}

// those things which should be run at once

void setup()

{
    Serial.begin(9600); //serial for instance
    of specific port timings

    pinMode(trigPin1, OUTPUT); // defining
    behaviour of every pin

    pinMode(echoPin1, INPUT);

    pinMode(trigPin2, OUTPUT);

    pinMode(echoPin2, INPUT);

    pinMode(vcc,OUTPUT);

    delay (1000);}

// for continuous loop

void loop()

{

    digitalWrite(vcc, HIGH); // We assumed this
    pin for getting an extra vcc pin always high

    find_distance(); // for getting distance
    continuously

    if(distance2<=30 && distance2>=15) //
    checking range from 15 to 30 cm

```

```

{

    temp=millis();          // store current
    time for comaparing in ms

    while(millis()<=(temp+300))    // Futher
    checking for 300 ms to identify gesture

    find_distance();

    if(distance2<=30 && distance2>=15) // if
    statement for identifiaion

    {

        temp=distance2;          // store
        Position of hand for comparing

        while(distance2<=50 || distance2==0)    //
        loop to check whether hand is still there

        {

            find_distance();          // call to get
            latest distance

            if((temp+6)<distance2)          //
            calibrating distance to cnfrm away
            movementof hand

            Serial.println("down");          // send
            "down" serially.

        }

        else if((temp-6)>distance2)          //
        checking for movement towards sensr

        Serial.println("up");          // send "up"
        serially.

```

<pre> } } } else // swipping identified { Serial.println("next"); // send "next" serially. } } else if(distance1<=30 && distance1>=15) // calculation for left sensor { temp=millis(); while(millis())<=(temp+300)) { find_distance(); if(distance2<=30 && distance2>=15) // checking movement of hand in from os both sensors left + right { </pre>	<pre> Serial.println("change"); // send "change" serially. l=1; // store 1 in variable l for flagging usage break; // breaking loop } } if(l==0) // if only movement in left sensor { serial.println("previous"); // send "previous" serially. while(distance1<=30 && distance1>=15) // loop to check stability of hand in left sensor find_distance(); } l=0; // make l=0 for the next round. } } </pre>
---	--

PYTHON CODE:

```
import serial
import pyautogui

Arduino_Serial = serial.Serial('COM8',9600)

while 1:
    incoming_data = str (Arduino_Serial.readline())
    print incoming_data

    if 'next' in incoming_data:
        pyautogui.hotkey('ctrl', 'pgdn')

    if 'previous' in incoming_data:
        pyautogui.hotkey('ctrl', 'pgup')

    if 'down' in incoming_data:

        pyautogui.scroll(-100)

    if 'up' in incoming_data:
        #pyautogui.press('up')
        pyautogui.scroll(100)

    if 'change' in incoming_data:
        pyautogui.keyDown('alt')
        pyautogui.press('tab')
        pyautogui.keyUp('alt')

    incoming_data = "";
```