

## 5. ACTUAL WORKING PRINCIPLE

The working of the project has been classified in different steps. These different steps are described as follows:

### Programming Arduino to Detect Gestures

The important part of this project is to write a program for Arduino such that it converts the distances measured by both the sensors into the appropriate commands for controlling certain actions.

We have already seen a project called PORTABLE ULTRASONIC RANGE METER, where you can measure the distance of an object placed in front of an Ultrasonic Sensor with the help of Arduino.

A similar concept is used here to measure the distance of your hand in front of both the Ultrasonic Sensors in this project. The fun part starts after calculating the distance.

The hand gestures in front of the Ultrasonic sensors can be calibrated so that they can perform five different tasks on your computer. Before taking a look at the gestures, let us first see the tasks that we can accomplish.

- 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).
- Play/Pause Video in VLC Player.
- Increase Volume.
- Decrease Volume.

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 35CM) 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 35CM) 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 Tab or Play/Pause the Video.

- **Gesture 5:** Swipe your hand across both the sensors (Left Sensor first). This action will Switch between Tasks.

Based on the above mentioned gesture, the following Arduino Program has been written.

## Arduino Code

```
const int trigPin1 = 11;
const int echoPin1 = 10;
const int trigPin2 = 6;
const int echoPin2 = 5;

long duration;
int distance1, distance2;
float r;
unsigned long temp=0;
int temp1=0;
int l=0;

void find_distance (void);

void find_distance (void)
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);

  duration = pulseIn(echoPin1, HIGH, 5000);

  r = 3.4 * duration / 2;
  distance1 = r / 100.00;
  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);
```

```

}

void setup()
{
  Serial.begin(9600);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  delay (1000);

}

void loop()
{
  find_distance();

  if(distance2<=35 && distance2>=15)
  {
    temp=millis();
    while(millis()<=(temp+300))
    find_distance();
    if(distance2<=35 && distance2>=15)
    {
      temp=distance2;
      while(distance2<=50 || distance2==0)
      {
        find_distance();
        if((temp+6)<distance2)
        {
          Serial.println("down");
        }
        else if((temp-6)>distance2)
        {
          Serial.println("up");
        }
      }
    }
    {
      Serial.println("next");
    }
  }

  else if(distance1<=35 && distance1>=15)
  {

    temp=millis();

    while(millis()<=(temp+300))

```

```

{
  find_distance();
  if(distance2<=35 && distance2>=15)
  {
    Serial.println("change");
    l=1;
    break;
  }
}

if(l==0)
{
  Serial.println("previous");
  while(distance1<=35 && distance1>=15)
  find_distance();
}
l=0;
}
}

```

The gesture mentioned above have been converted into 5 Commands that are sent to the Serial Port. Using these 5 commands, Python Program has been written to control certain Keyboard Functions in order to achieve the required task.

### Python Programming for the Project

Writing Python Program for Arduino based Hand Gesture Control is very simple. It is needed to read the Serial data from Arduino and invoke certain keyboard key presses. In order to achieve this, a special Python Module named PyAutoGUI have been installed.

### Python Code

If everything goes well till now, you can proceed to write the Python Code. If you observe the Arduino Code given above, the Arduino sends out five different texts or commands through Serial Port upon detecting appropriate hand gestures. These commands are:

- Next
- Previous
- Up
- Down
- Change

Using these commands along with few functions in PyAutoGUI (like hotkey, scroll, keyDown, press and keyUp), a simple Python Code can be written that will execute the following tasks of keyboard and mouse.

- Data = “next” --> Action = Ctrl+PgDn
- Data = “previous” --> Action = Ctrl+PgUp
- Data = “down” --> Action = Down Arrow
- Data = “up” --> Action = Up Arrow
- Data = “change” --> Action = Alt+Tab

The Python Code for Arduino based Hand Gesture Control of Computer is given below.

```
import serial
import pyautogui

Arduino_Serial = serial.Serial('com12',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.press('down')
    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 = "";
```