Screen control using hand gestures

Section: 3

Group No: 14

Submitted to: Prof. Ashok Ranade

Group members:

AU1940292 Karan Shah

AU1940199 Digvijaysinh Gohil

AU1940280 Pankil Sheth

- ## Objective:

  Human Machine Interface is a system comprising of hardware and software that helps in communicating and exchanging information between the user and the machine.
  We normally take use of LEDs, Switches, Touch Screens and LCD Displays as a part of HMI devices. Another way to communicate with machines like Robots or Computers is with the help of Hand Movements.
  Instead of using keyboard or mouse we can use our hand gestures to control certain functions of a computer like forward/rewind a video, increase/decrease the volume of a video, scroll up/down through the web pages and many more.
  The objective of the project is a simple Arduino based hand gesture controlling where you can control functions of your web browser like switching between tabs, scrolling up and down in web pages, shift between tasks (applications), changing the songs in Spotify, play or pause a video and increase or decrease the volume or mute the video (in VLC Player and YouTube) with the help of hand gestures.

- ## Outcomes:

  The outcome of the project is the different functions happening on screen by different hand movements at specific distance from the sensors. The following gestures will be resulted at the mentioned distances from the sensors.

  Gesture 1: Place your hand in front of the Right Ultrasonic Sensor at a distance (between 10CM to 20CM) for a small period of time 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 10CM to 20CM) 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 switch application.

  Gesture 4: Place your hand in front of the Left Ultrasonic Sensor at a distance (between 10CM to 20CM) for a small duration and move your hand away from the sensor. This gesture will rewind the video and change the Spotify songs.

Gesture 5: Place your hand in front of the Left Ultrasonic Sensor at a distance (between 10CM to 20CM) for a small duration and move your hand away from the sensor. This gesture will forward the video and change the Spotify songs.

Gesture 6: Swipe your hand in front of the Left Ultrasonic Sensor. This gesture will mute the video.

• Circuit Diagram:
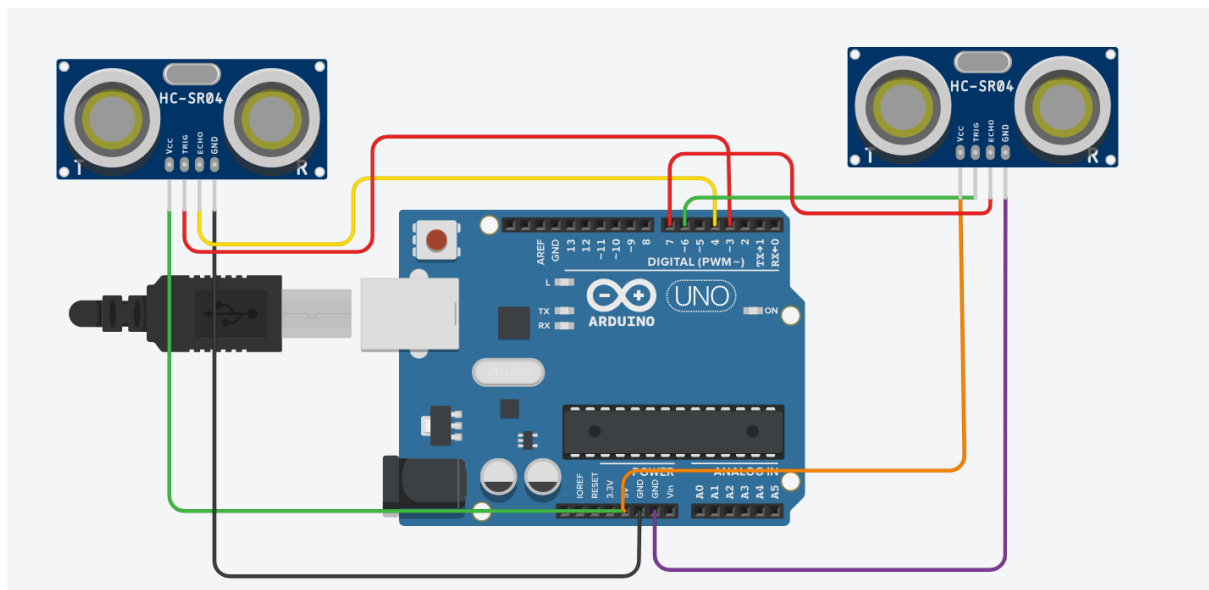
Hardware components used in the circuit are:

    (a) Arduino UNO
    (b) Jumper wires
    (c) HC-SR04 Sensor
    (d) USB wire

• Code of the Program:

ARDUINO CODE

```
const int trigger1 = 3;
const int echo1 = 4;
```

```c
const int trigger2 = 6;

const int echo2 = 7;

long time_taken;

int dist,distL,distR;

long duration;

float r;

unsigned long temp=0;

int temp1=0;

int l=0;

void find_distance (void);

void find_distance (void)

{
  digitalWrite(trigger1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger1, LOW);
  duration = pulseIn(echo1, HIGH, 5000);
  r = 3.4 * duration / 2;
  distL = r / 100.00;
   digitalWrite(trigger2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger2, LOW);

  duration = pulseIn(echo2, HIGH, 5000);
  r = 3.4 * duration / 2;
  distR = r / 100.00;
```

```arduino
  delay(100);

}

void setup() {

Serial.begin(9600);


pinMode(trigger1, OUTPUT);

pinMode(echo1, INPUT);

pinMode(trigger2, OUTPUT);

pinMode(echo2, INPUT);

}

void calculate_distance(int trigger, int echo)

{

digitalWrite(trigger, LOW);

delayMicroseconds(2);

digitalWrite(trigger, HIGH);

delayMicroseconds(10);

digitalWrite(trigger, LOW);


time_taken = pulseIn(echo, HIGH);

dist= time_taken*0.034/2;

if (dist>50)

dist = 50;

}

void loop() {

calculate_distance(trigger1,echo1);

distL =dist;

calculate_distance(trigger2,echo2);

distR =dist;

if (distL>=10 && distL<=20)
```

```
{
  delay(50); //Hand Hold Time
  calculate_distance(trigger1,echo1);
  distL =dist;
  if (distL>=10 && distL<=20)
  {
    Serial.println("Left Locked");
    while(distL<=40)
    {
      calculate_distance(trigger1,echo1);
      distL =dist;
      if (distL<15) //Hand pushed in
      {Serial.println ("Rewind"); delay (300);}
      if (distL>20) //Hand pulled out
      {Serial.println ("Forward"); delay (300);}
    }
  }
}

if (distR>=10 && distR<=20)
{
  delay(50);
  calculate_distance(trigger2,echo2);
  distR =dist;
  if (distR>=10 && distR<=20)
  {
    Serial.println("Right Locked");
    while(distR<=40)
    {
```

```
      calculate_distance(trigger2,echo2);

      distR =dist;

      if (distR<15 )

      {Serial.println ("Vup"); delay (300);}

      if (distR>20)

      {Serial.println ("Vdown"); delay (300);}

    }

  }

}




   if(distR<=8 && distR>=0)

 {

   temp=millis();

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

   find_distance();


   {

    Serial.println("next");

   }

   }

     if(distL<=8 && distL>=0)

 {

   temp=millis();

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

   find_distance();


   {

    Serial.println("Mute");
```

```
      }
     }
delay(200);

}
```

PYTHON CODE

```python
import serial

import time

import pyautogui


ArduinoSerial = serial.Serial('com5',9600)

time.sleep(2)

while 1:

    incoming = str (ArduinoSerial.readline())

    print incoming

    if 'Mute' in incoming:

        pyautogui.hotkey('m')


    if 'Rewind' in incoming:

        pyautogui.hotkey('ctrl', 'left')


    if 'Forward' in incoming:

        pyautogui.hotkey('ctrl', 'right')


    if 'Vup' in incoming:

        pyautogui.hotkey('down')


    if 'Vdown' in incoming:
```
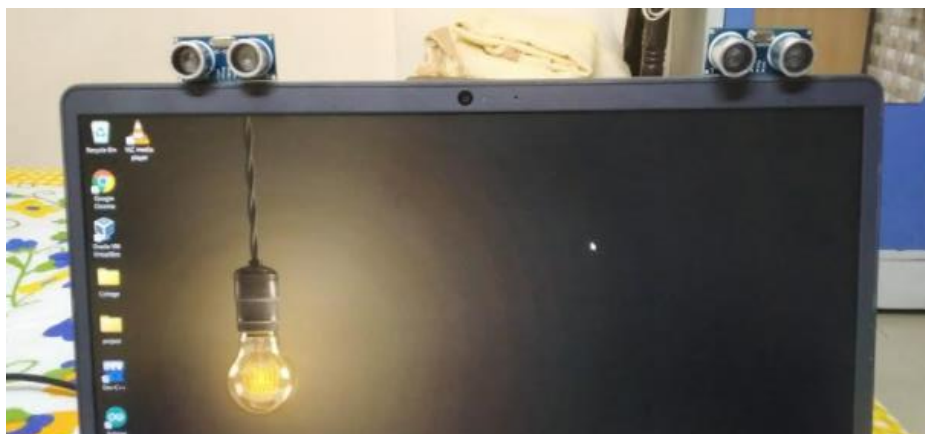
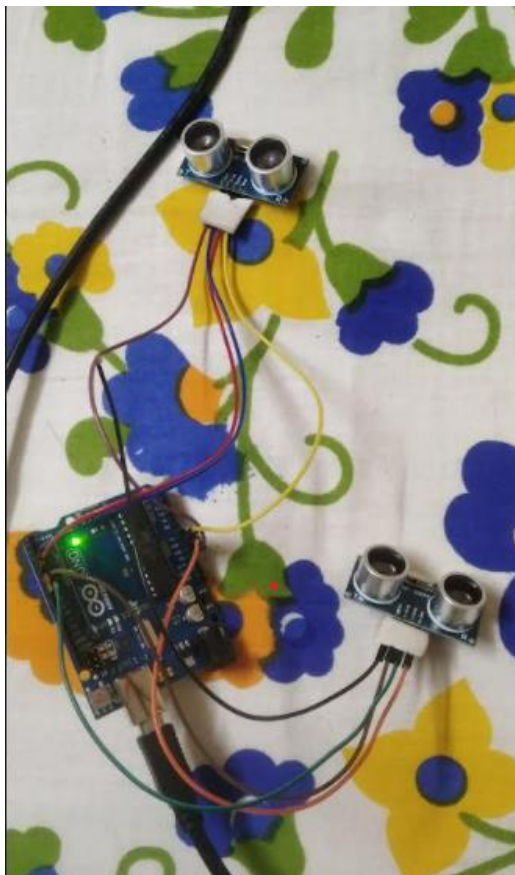```
    pyautogui.hotkey('up')

if 'next' in incoming:

    pyautogui.hotkey('alt', 'tab')

incoming = "";
```

- ## **Photos**

- ## **Demonstration Video**

  Below mentioned link can be used to view project demonstration.
  [Screen Control Using Hand Gestures](#)