CS209/CS210 COMPUTER ARCHITECTURE PROJECT

TOPIC:

ARDUINO BASED HAND GESTURE CONTROL SYSTEM

Submitted By:

Name: Bhavika Choudhary

Roll No.: 2101AI10

Submission Date: 02/12/2022

About Project:

The project is about an Arduino-based hand gesture control system which identifies various hand gestures and perform functions related to them. Gesture control is the ability to recognize and interpret movements of the human body in order to interact with and control a computer system without direct physical contact. With the world developing at a speed unimaginable to mankind, we need smart systems which require less effort to operate.

This project, thus allows the users to remotely control and perform various functions on the computer system.

Introduction:

In this project, I have used two ultrasonic sensors to detect any pulse generated by the object (in this case, our hand) which in turn returns the time taken for the pulse to travel back and forth by getting reflected by the object. This time taken is used to calculate the distance between the sensor and the object.

Now with this, at different distances, different gestures are taken into account and functions relating to them are performed.

Description:

According to the Arduino code, it first calculates the distance between the object and the ultrasonic sensor and according to carious distances, different outputs are given in the Serial port.

This project uses Python to set the commands to be performed for various gestures.

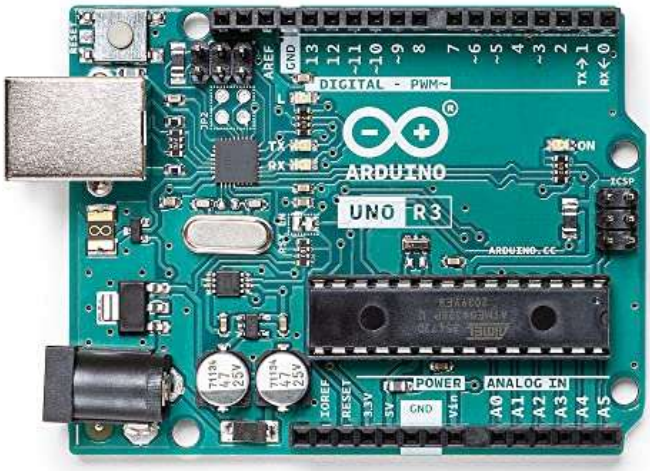
The “Serial” output is read by Python’s pyserial module and different commands like pressing spacebar to play/pause video, pressing Crtl+right key to forward the video, and so on.

Below are the mentioned functions set according to various hand gestures:

- Both hands in-front of sensors : Play/Pause Video
- Right Hand in-front of right sensor: Right Locked
- Right Hand pushed in: Rewind
- Right Hand pulled away: Forward
- Left Hand in-front of right sensor: Left Locked
- Left Hand pushed in: Volume Down
- Left Hand pulled away: Volume Up
- Any hand put really close to sensor: Play Next in queue

Components used:

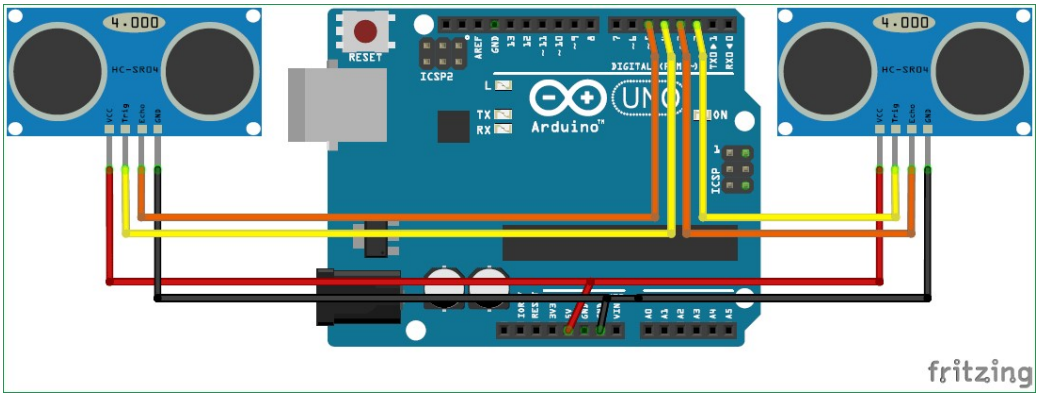
1. Arduino Uno R3: Arduino Uno is an open-source micro controller board based on the Microchip ATMega328P micro controller and developed by Arduino.cc. It is equipped with various digital and analog input/output pins, capable of interfacing with other circuits and is programmable with Arduino ID.



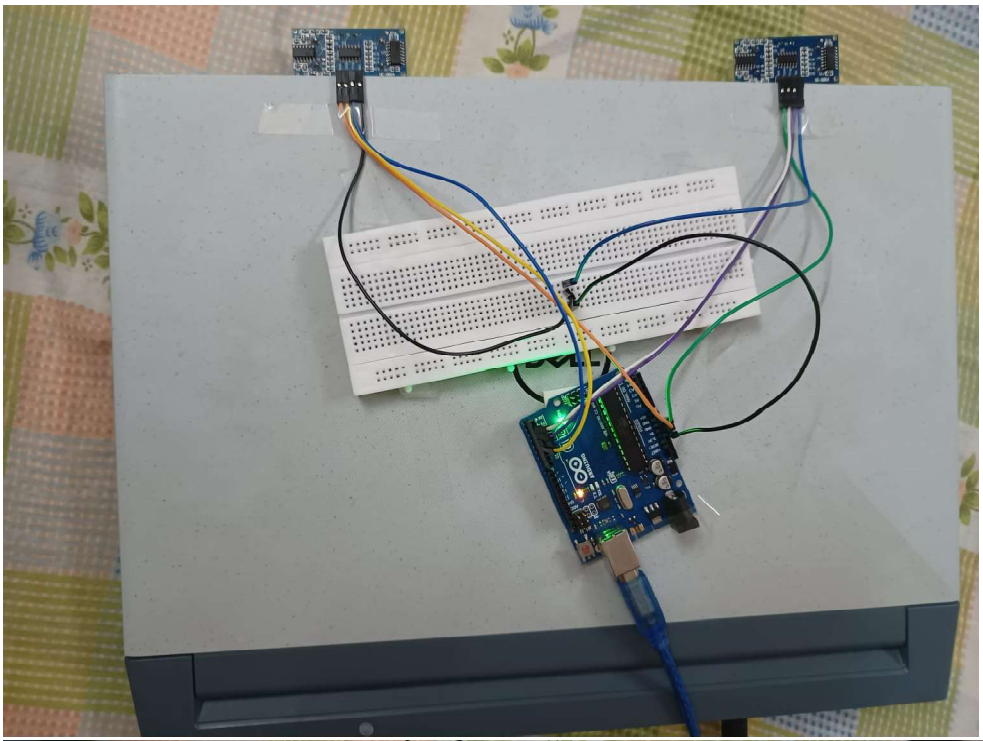
2. HC-SR04 Ultrasonic Distance Sensor: It uses SONAR to determine the distance of an object in front of it. It can measure the distance between 2cm to 400 cm with high accuracy.

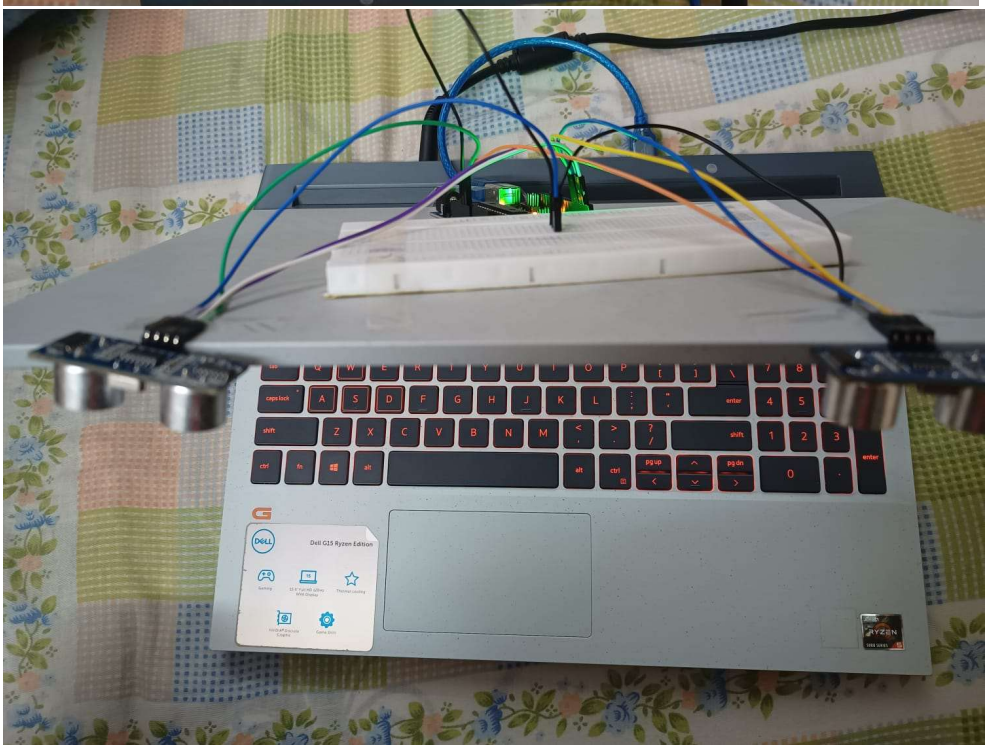
3. Jumper Wires, Breadboard

Circuit Diagram:



Full Circuit Images:





Connections:

Trig Pin (Sensor 1) - D1 (Arduino)

Echo Pin (Sensor 1) - D2 (Arduino)

Trig Pin (Sensor 2) - D3 (Arduino)

Echo Pin (Sensor 2) - D4 (Arduino)

Gnd pin of both sensors – Gnd of Arduino

VCC pin of both sensors – 5V of Arduino

Arduino Code:

```
const int trigger1 = 1; //Trigger pin of 1st Sensor
const int echo1 = 2; //Echo pin of 1st Sensor
const int trigger2 = 3; //Trigger pin of 2nd Sensor
const int echo2 = 4;//Echo pin of 2nd Sensor

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;
  delay(100);
}
```

```
void setup() {
Serial.begin(9600);

pinMode(trigger1, OUTPUT);
pinMode(echo1, INPUT);
pinMode(trigger2, OUTPUT);
pinMode(echo2, INPUT);
}

/*Function to calculate distance*/
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() { //infinite loopy
calculate_distance(trigger1,echo1);
distL =dist; //get distance of left sensor

calculate_distance(trigger2,echo2);
distR =dist; //get distance of right sensor

//Uncomment for debugging
/*Serial.print("L=");
Serial.println(distL);
Serial.print("R=");
Serial.println(distR);
*/
```

```
//Pause Modes -Hold
if ((distL >25 && distR>25) && (distL <50 && distR<50)) //Detect both hands
{Serial.println("Play/Pause"); delay (500);}

calculate_distance(trigger1,echo1);
distL =dist;

calculate_distance(trigger2,echo2);
distR =dist;



//Control Modes

//Lock Left - Control Mode
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 ("Vup"); delay (300);}
      if (distL>20) //Hand pulled out
      {Serial.println ("Vdown"); delay (300);}
    }
  }
}

//Lock Right - Control Mode
if (distR>=10 && distR<=20)
{
  delay(50); //Hand Hold Time
  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 ) //Right hand pushed in
      {Serial.println ("Rewind"); delay (300);}
      if (distR>20) //Right hand pulled out
      {Serial.println ("Forward"); delay (300);}
    }
  }
}



//Swipe Next - Control Mode
 if(distR<=8 && distR>=0)
  {
    temp=millis();
    while(millis()<=(temp+300))
    find_distance();

    {
      Serial.println("next");
    }
     }




delay(200);
}
```

Python Code:

import serial #for Serial communication

import time

import pyautogui

ArduinoSerial = serial.Serial('com6',9600)

time.sleep(2) #wait for 2 seconds for the communication to get established

```
while 1:
    incoming = str(ArduinoSerial.readline()) #read the serial data and print it as line
    print(incoming)



    if 'Play/Pause' in incoming:
        pyautogui.typewrite(['space'], 0.2)

    if 'Rewind' in incoming:
        pyautogui.hotkey('ctrl', 'left')

    if 'Forward' in incoming:
        pyautogui.hotkey('ctrl', 'right')

    if 'Vup' in incoming:
        pyautogui.hotkey('ctrl', 'down')

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

    if 'next' in incoming:
        pyautogui.typewrite('n',0.2)

    incoming = "";
```

Note: For Python Code to work, one must install pyserial and pyautogui modules.

Conclusion:

With all this, I conclude my project of Arduino based Hand gesture control system which can be used to control your computer system remotely, hence reducing human physical effort

Thank you