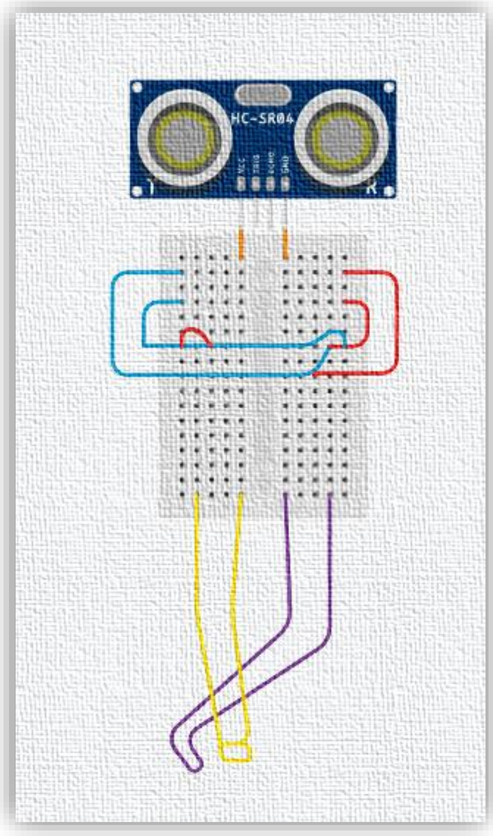


MOTION CONTROLLED GESTURE USING ARDUINO

and Python

COURSE NAME

ENR 305:
Sensors, Instruments and Experimentation

SECTION DETAILS

Section Number: 1
Group Number: 18

MEMBER DETAILS

Nihar Patel	>>	AU1940119
Purvam Sheth	>>	AU1940151
Mohit Prajapati	>>	AU1940171

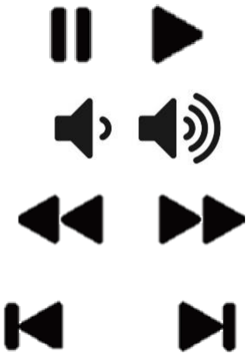
Objective

MOTION CONTROLLED GESTURE USING ARDUINO:

- Our system is a hand-based gesture driven.

We can provide basic commands like:

- Play/Pause
- Volume-Up/Volume-Down
- Fast-Forward/Rewind
- Next/Previous



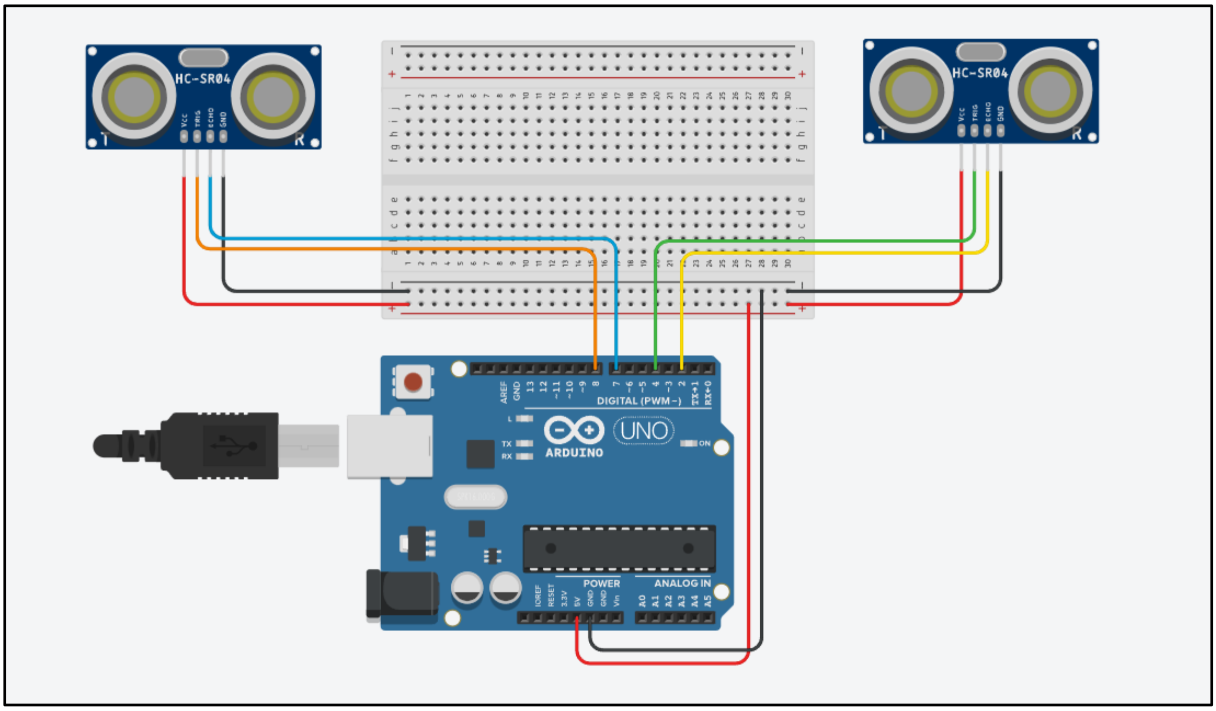
by providing different types of Hand based gesture.

Outcomes

We have achieved this goal by combining the power of Arduino and Python together.

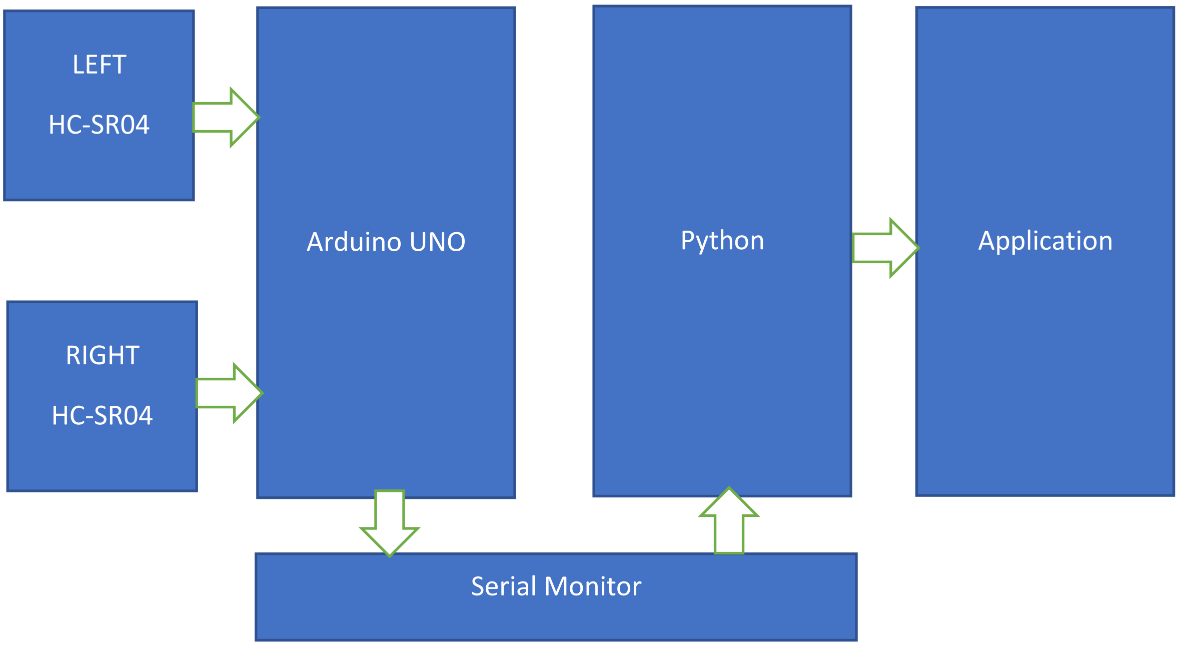
- For Inputs we have 2 Ultrasonic sensors HC-SR04.
- The Input will we processed and based on that Arduino will print the command on the Serial Monitor.
- Using a python script, we will read the commands printed on the Serial Monitor, and based on that it will give System command to press the related key to execute that command.

CIRCUIT DIAGRAM



(In TinkerCAD)

BLOCK DIAGRAM



CODE

Arduino

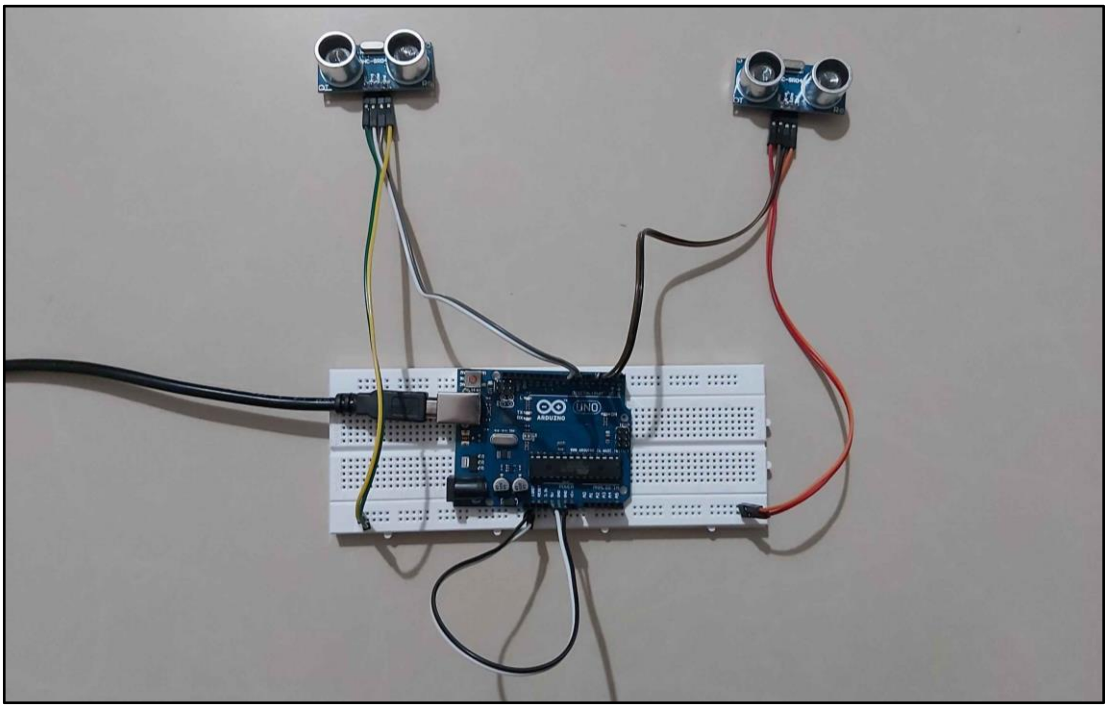
```
// C++ code
//
//Arduino Code:
/*
 * Program for gesture control
 * Controlled using Python
 */

long time_taken;
int dist,distL,distR;

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

//Function to calculate distance
long calculate_distance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the distance in centimeter.
  dist = 2.54*0.006783*pulseIn(echoPin, HIGH);
  return dist;
}

void loop() {

  distL = calculate_distance(8, 7); //get distance of left sensor
  distR = calculate_distance(4, 2); //get distance of right sensor

  //Serial.println(distL);
  //Serial.println(distR);

  // L                    R
  // 0- 5: Play/Pause     0-5
  // 5-15: Vup/Vdown      Near to Far/Far to Near
  //15-25: Brightness     Near to Far
  //25-35: Forward/Rewind Near to Far/Far to Near
  //

  //Play/Pause
  if ((distL > 0 && distL <= 5) && (distR > 0 && distR <= 5)) //Detect
both hands
  {
    Serial.println("Play/Pause");
    delay (1000);
  }



  //Volume Up/Volume Down
  if (distL > 5 && distL <= 15)
  {
    delay(500); //Hand Hold Time
    distL = calculate_distance(8, 7); //get distance of left sensor
    if (distL > 5 && distL <= 15)
    {
      while(distL > 5 && distL <= 15)
      {
        distR = calculate_distance(4, 2); //get distance of right
sensor
        if (distR < 15) //Hand pushed in
        {
          if (distR < 5)
          {
            break; //to break
          }
          Serial.println ("Volume-Down");
          delay (500);
        }
        if (distR >= 15) //Hand pulled out
        {
          if (distR >= 30)
          {
            break; //to break
          }
          Serial.println ("Volume-Up");
          delay (500);
        }
      }
    }
  }



  //Fast Foward/Rewind
  if (distL > 15 && distL <= 25)
  {
    delay(1000); //Hand Hold Time
    distL = calculate_distance(8, 7); //get distance of left sensor
    if (distL > 15 && distL <= 25)
    {
      while(distL > 15 && distL <= 25)
      {
        distR = calculate_distance(4, 2); //get distance of right
sensor
        if (distR < 15) //Hand pushed in
        {
          if (distR < 5)
          {
            break; //to exit
          }
          Serial.println ("Rewind");
          delay (1500);
        }
        if (distR >= 15) //Hand pulled
        {
          if (distR >= 30)
          {
            break; //to exit
          }
          Serial.println ("Fast-Forward");
          delay (1500);
        }
      }
    }
  }



  //Next/Previous
  if (distL > 25 && distL <= 35)
  {
    delay(1500); //Hand Hold Time
    distL = calculate_distance(8, 7); //get distance of left sensor
    if (distL > 25 && distL <= 35)
    {
      while(distL > 25 && distL <= 35)
      {
        distR = calculate_distance(4, 2); //get distance of right
sensor
        if (distR < 15) //Hand pushed in
        {
          if (distR < 5)
          {
            break; //to exit
          }
          Serial.println ("Previous");
          delay (2500);
        }
        if (distR >= 15) //Hand pulled out
        {
          if (distR >= 30)
          {
            break; //to exit
          }
          Serial.println ("Next");
          delay (2500);
        }
      }
    }
  }

delay(100);
}
```

Python

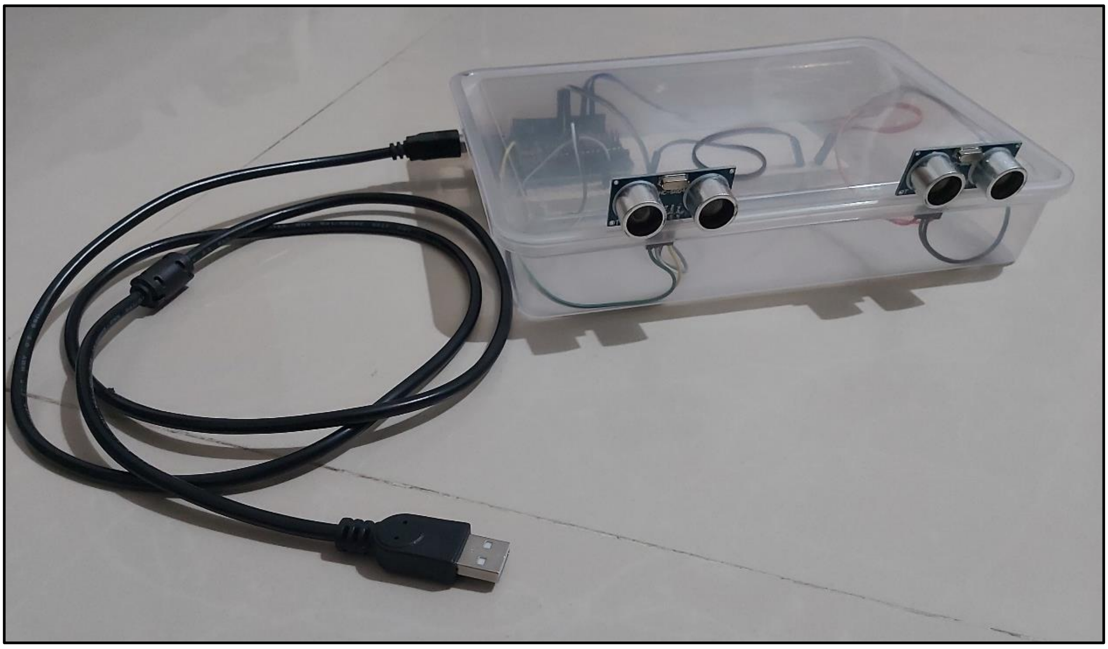
```
#Python Code:

import serial #Serial imported for Serial communication
import time #Required to use delay functions
import pyautogui

ArduinoSerial = serial.Serial('COM6',9600) #Create Serial port object
called arduinoSerialData
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 'Volume-Down' in incoming:
        pyautogui.hotkey('down')

    if 'Volume-Up' in incoming:
        pyautogui.hotkey('up')
    if 'Fast-Forward' in incoming:
        pyautogui.hotkey('ctrl', 'right')

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

    if 'Next' in incoming:
        pyautogui.hotkey('n')

    if 'Previous' in incoming:
        pyautogui.hotkey('p')

    incoming = "";
```

IMAGES

Circuit



- Project



DESCRIPTION

- For Play/Pause we need to put our both hands within 5cm of Ultrasonic Sensor.
- From the Left Ultrasonic sensor, we can change operation like switch between Volume Function, Time skip function and Neighbor Track.
- While from the Right Ultrasonic sensor, we can decide the subdivision inside the operation. We can choose between Volume-Up or Volume-Down, Fast-Forward or Rewind and Next or Previous.
- We can cancel a function by getting Left or Right hand outside the Range.

____**x**____