“ARDUINO BASED HAND GESTURE CONTROL APPLICATION”

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

Computer Science and Engineering

by

Himanshu Kaushik (18BCE0579), Abhinav (18BCE0641),

Deepak Jha (18BCE0654)

Under The Guidance of Prof. Nithya S

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Subject: Microprocessors And Interfacing
Course Code: CSE2006

DECLARATION

We hereby declare that the thesis entitled “Arduino based hand gesture control application" submitted by us, for the J-component of course “Microprocessor and interfacing” is a record of bonafide work carried out by us under the supervision of Prof. NITHYA.S .

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore
Date: 5/06/2020

Himanshu Kaushik
Deepak Jha
Abhinav
Signature of the Candidates

Acknowledgements

I am grateful to VIT University for encouraging and giving us the opportunity to perform this project. I would like to thank my Project Guide, Professor NITHYA.S for her guidance and support through every step of this project. Last but not the least, I am thankful to all family and friends for their help. The completion of this project is attributed to everyone’s combined effort.

Himanshu Kaushik
Deepak Jha
Abhinav
Signature of the Candidates

Table of contents

I. ABSTRACT

In this project, we are going to build a gesture-controlled laptop. It is based on using the combination of Arduino and Python.

Instead of using a keyboard, mouse or joystick, we can use our hand gestures to control certain functions of a computer like to play/pause a video, increase/decrease the volume, move left/right in a photo slideshow, and scroll up/down in a web page and many more. This is why we decided to control VLC Media Player as a "ARDUINO BASED HAND GESTURE CONTROL APPLICATION" project.

We will use two Ultrasonic Sensors (HC-SR04) with Arduino. We will place the two sensors on the top of a laptop screen and calculate the distance between the hand and the sensor. Counting on the information from Arduino that is sent to Python through the serial port, this information will then be read by Python which is running on the computer in order to perform certain actions.

II. INTRODUCTION

Recently Gesture controlled Laptops or computers are getting very famous. This technique is called Leap motion which enables us to control certain functions on our computer/Laptop by simply waving our hand in front of it. Gestures allow the user to handle multiple points of input and even define several parameters at once. They are, therefore, a more natural form of communication. Compared to many existing interfaces, hand gestures have the advantages of being easy to use, natural, and intuitive.

Principle Behind

We will use two Ultrasonic sensors to determine the position of our hand and control a media player (VLC) based on the position.

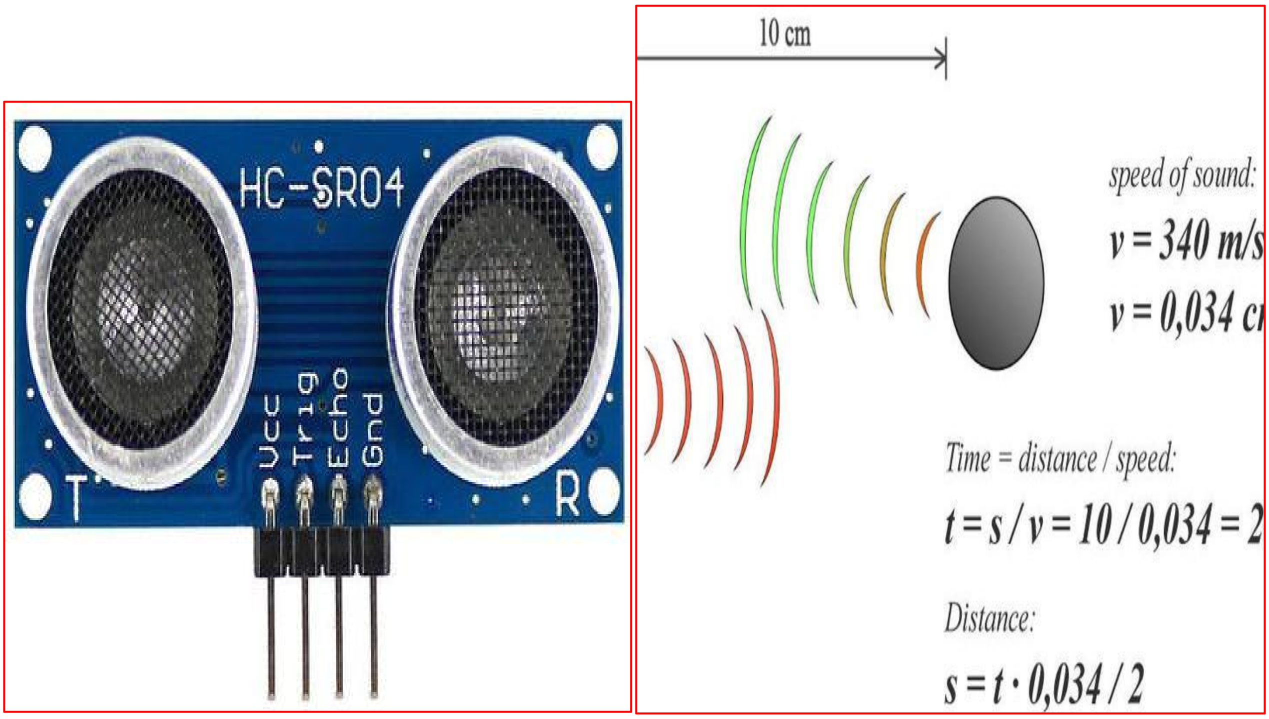

Fig 1: Principle

The idea of ultrasonic sensor came from the bats and dolphins, they estimate distance by using the echolocation process—sound waves transmitted, bounced back and received, with the time difference used to calculate the distance of objects.

First of all we need to trigger the ultrasonic sensor module to transmit signal by using Arduino and then wait for receive ECHO. Arduino reads the time between triggering and Received ECHO. We know that speed of sound is around 340m/s. so we can calculate distance by using given formula: Distance= (travel time/2) * speed of sound Where speed of sound around 340 meter per second.

III. OVERVIEW

The **concept** behind the project is very simple. We will place two Ultrasonic (US) sensors on top of our monitor and will read the distance between the monitor and our hand using Arduino, based on this value of distance we will perform certain actions. To perform actions on our computer we use Python *pyautogui* library. The commands from Arduino are sent to the computer through serial port (USB). This data will be

then read by python which is running on the computer and based on the read data an action will be performed.

The circuit diagram of Arduino part of the project is shown in the following image. It consists of an Arduino UNO board and two Ultrasonic Sensors and we can power up all these components from the laptop's USB Port.

ARCHITECTURE DIAGRAM:

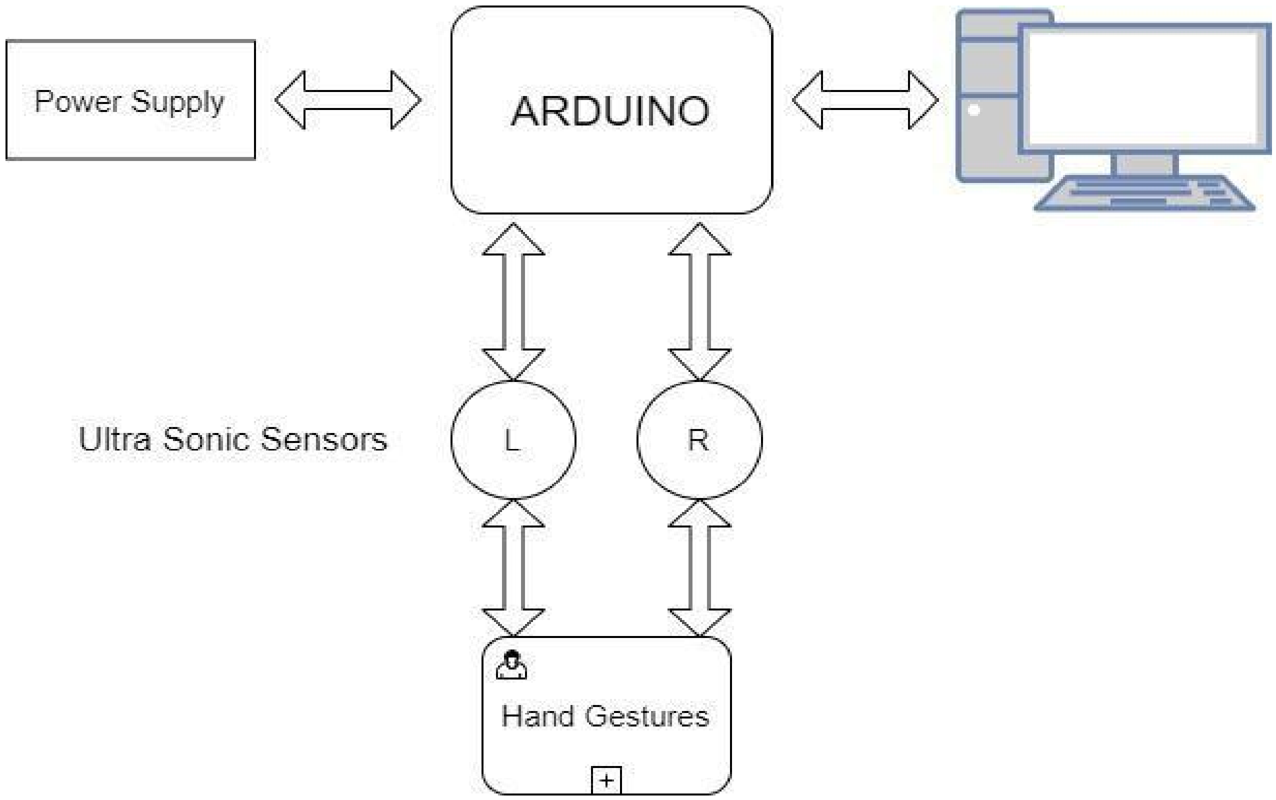

Fig 2: Architecture Diagram

Hand Gestures:

The following are the 5 commands hand gestures that we will implement with the help of Arduino UNO and python libraries.

<u>First hand gesture</u>: It allows us to 'Play/Pause' VLC by placing the two hands in front of the right/left Ultrasonic Sensor at a particular

far distance.

Second gesture: It allows us to 'Rewind' the video by placing a hand in front of the left sensor at a particular far distance.

Third gesture: It allows us to 'Forward' the video by placing a hand in front of the right sensor at a particular far distance.

Forth gesture: It allows us to 'Increase Volume' of the video by placing a hand in front of the left sensor at a particular far distance and moving away from the Sensor.

Fifth gesture: It allows us to 'Decrease Volume' of the video by placing a hand in front of the left sensor at a particular far distance and get near to the sensor.

IV. LITERATURE SURVEY

In [1] ,most of the human computer interaction interfaces that are designed today require explicit instructions from the user in the form of keyboard taps or mouse clicks A novel method to recognize hand gestures for human computer interaction, using computer vision and image processing techniques, is proposed in this paper. However, this kind of input also raises issues that are not relevant with traditional input. On the user's side, these problems are to learn, to remember and to accurately execute gestures user's side, these problems are to learn, to remember and to accurately execute gestures.

In [2], a hand gesture recognition system provide a natural, innovative and modern way of non verbal communication. It has a wide area of application in human computer interaction and sign language. The intention of this paper is to discuss a novel approach of hand gesture recognition based on detection of some shape based features. these problems are to learn, to remember and to accurately execute gestures. The developer has to provide a system that correctly recognizes these gestures.

In [3], a gesture recognition based Human Computer Interaction control system is developed via Lab VIEW in this paper. Firstly, the

gesture sample data is collected by using five bending sensor of data glove, then, in order to improve the recognized precision, the data collected is pre-processed and the optimize the SVM kernel parameter value is found by using the improved PSO algorithm. Furthermore, to solve the existing problems of lower precision and poor real-time ability in gesture recognition algorithm, an improved PSO-SVM classification algorithm of
hand-gesture recognition is proposed.

In [4] ,there are a lot of home appliances and personal computers around us.. In this study, as an interface focusing on the ease of use, we develop a system to control personal computer by Arduino based Hand Gesture Control of Computer Application applying the natural behavior of human. Research issues include the definition of the association between the PC operations and gestures, the recognition of hand gestures, the adjustment of the error of the gestures, and how to realize the system.

In [5], as there are new developments and innovation in the field of computer technology, size of electronic devices is decreasing rapidly. Thus, there is a need of new input interface for such devices. By applying vision technology and controlling the devices by natural hand gestures, we can reduce the work space required. In this paper, we propose a novel approach that uses a video device to control the Laptop using gestures. Simple interfaces already exist, such as embedded keyboard, folder keyboard and mini-keyboard. However, these interfaces need some amount of space to use and cannot be used while moving.

In [6], touch-less hand gesture is one emerging technology for human computer interaction. In this paper, we investigate the feasibility of a hybrid system using frequency modulated continuous wave (FMCW) radar and ultrasound sensors with one transmitter and one receiver to detect hand movement for controlling a computer. This kind of input also raises issues that are not relevant with traditional input. On the user's side, these problems are to learn, to remember and to accurately execute gestures.

In [7],human Computer Interaction keeps moving toward interfaces which are more natural and intuitive to use, in comparison to traditional keyboard and mouse. Hand gestures are an important modality for human computer interaction (HCI). These interfaces need some amount of space to use and cannot be used while moving.

In [8],the article shows the design and implementation of a gesture control system, which determines the gesture from the movement of the hands. The movement detected with distance measurement sensors, and the presented control interface determines the assigned gesture according to the hand's movement. The developer not only has to ensure that gestures are quickly and correctly recognized, but also has to provide a guide that allows a rapid and easy learning of these gestures.

In [9], in the recent days gesture controlled devices are getting more attention. There are different types of gesture controlling techniques. In this work, gesture of the user controls the movement of the mobile robot. The developed system is classified into gesture unit and mobile robot unit. This proposed system is developed at low cost with better efficiency. It can be implemented in many other applications such as holding the speakers to assist dumb people, toys etc. The teaching of multi-touch and mid-air gestures is more difficult than that of single-touch gestures. In the case of the latter, the hand posture is irrelevant - users only need to follow a path correctly to perform a command.

V. DESIGN AND ANALYSIS

To control the PC with hand gestures , just connect the two Ultrasonic sensors with Arduino. We know US sensor work with 5V and hence they are powered by the on board Voltage regulator of Arduino. The Arduino can be connected to the PC/Laptop for powering the module and also for Serial communication. Once the connections are done place them on your monitor as shown below. I have used a double side tape to stick it on my monitor but you can use your own creativity. After securing it in a place we can proceed with the Programming.

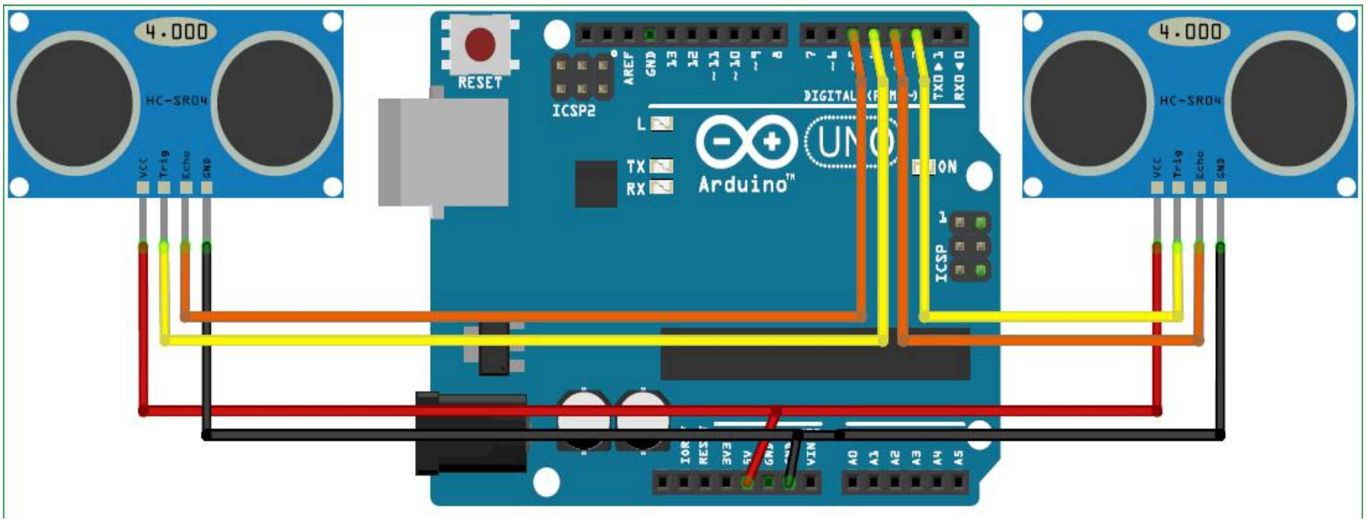

Fig 3: Circuit Diagram

All programs we start with defining the I/O pins as shown below. The two UltraSonic sensors are connected to Digital pins 2,3,4 and 5 and are powered by +5V pin. The trigger pins are output pin and Echo pins are input pins.

The Serial communication between Arduino and python takes places at a baud rate of 9600.

VI. CONCLUSION

In this project, I 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.

Such Gesture based Control of Computers is already present and a company called Leap Motion has been implementing such technology in computers.

The idea to control the behavior of any device is very futuristic. This type of hand gesture control of computers can be used for VR (Virtual Reality), AR (Augmented Reality), 3D Design, Reading Sign Language, etc. This can provide solution to many modern problems.

VII. REFERENCES

[1] Agrawal, R., & Gupta, N. (2016, February). Real time hand gesture recognition for human computer interaction. In 2016 IEEE 6th International Conference on Advanced Computing (IACC) (pp. 470-475). IEEE.

[2] Panwar, M., & Mehra, P. S. (2011, November). Hand gesture recognition for human computer interaction. In 2011 International Conference on Image Information Processing (pp. 1-7). IEEE.

[3] Zhi-heng, W., Jiang-tao, C., Jin-guo, L., & Zi-qi, Z. (2017, May). Design of human-computer interaction control system based on hand-gesture recognition. In 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC) (pp. 143-147). IEEE.

[4] Yamagishi, K., Jing, L., & Cheng, Z. (2014, December). A system for controlling personal computers by hand gestures using a wireless sensor device. In 2014 IEEE International Symposium on Independent Computing (ISIC) (pp. 1-7). IEEE.

[5] Singh, D. K. (2015, April). Recognizing hand gestures for human computer interaction. In 2015 International Conference on Communications and Signal Processing (ICCSP) (pp. 0379-0382). IEEE.

[6] Nguyen, M. Q., & Li, C. (2018, April). Radar and ultrasound hybrid system for human computer interaction. In 2018 IEEE Radar Conference (RadarConf18) (pp. 1476-1480). IEEE.

[7] Parkale, Y. V. (2012, January). Gesture based operating system control. In 2012 Second International Conference on Advanced Computing & Communication Technologies (pp. 318-323). IEEE.

[8] Sziladi, G., Ujbanyi, T., Katona, J., & Kovari, A. (2017, September). The analysis of hand gesture based cursor position control during solve an IT related task. In 2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom) (pp. 000413-000418). IEEE.

[9] Rajkanna, U., Mathankumar, M., & Gunasekaran, K. (2014, March). Hand gesture based mobile robot control using PIC microcontroller. In 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014] (pp. 1687-1691). IEEE.

VIII. ANNEXURE

ARDUINO CODE -

```
const int trigPin1 = 11; // the number of the trigger output pin ( sensor 1 )
const int echoPin1 = 10; // the number of the echo input pin ( sensor 1 )
const int trigPin2 = 6;  // the number of the trigger output pin ( sensor 2 )
const int echoPin2 = 5;  // the number of the echo input pin ( sensor 2 )

///// variables used for distance calculation
long duration;
int distance1, distance2;
float r;
unsigned long temp=0;
int temp1=0;
int l=0;



void find_distance (void);
// this function returns the value in cm.
/*we should not trigger the both ultrasonic sensor at the same time.
it might cause error result due to the intraction of the both soundswaves.*/

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

    duration = pulseIn(echoPin1, HIGH, 5000);// here this pulsein function wont wait
                                       // more then 5000us for the ultrasonic
                                       // sound to came back. (due to this it
                                       // wont measure more than 60cm)



  r = 3.4 * duration / 2;                   // calculation to get the measurement
                                     // in cm using the time returned by the
                                     // pulsein function.
  distance1 = r / 100.00;

  /////////upper part for left sensor and lower part for 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);
}


void setup()
{
 Serial.begin(9600);
 pinMode(trigPin1, OUTPUT); // initialize the trigger and echo pins of both the
                  // sensor as input and output:
 pinMode(echoPin1, INPUT);
 pinMode(trigPin2, OUTPUT);
 pinMode(echoPin2, INPUT);
 delay (1000);
}



void loop()
{
 find_distance(); // this function will stores the current distance measured by
                     // the ultrasonic sensor in the global variable "distance1 and
                     //  distance2"
           // no matter what, the program has to call this "find_distance"
                     // function continuously to get the distance value at all time.

 if(distance2<=35 && distance2>=15) // once if we placed our hands in front of
                                         // the right sensor in the range between 15
                                         // to 35cm this condition becomes true.
 {
  temp=millis();           // store the current time in the variable
                                         // temp. (" millis " Returns the number of
                                         // milliseconds since the Arduino board began
                                         // running the current program )

  while(millis()<=(temp+300))   // this loop measures the distance for
                                         // another 300 milliseconds. ( it helps to
                                         // find the difference between the swipe and
                                         // stay of our hand in front of the right
                                         // sensor )
  find_distance();
  if(distance2<=35 && distance2>=15) // this condition will true if we place our
                                          // hand in front of the right sensor for
                                          // more then 300 milli seconds.
  {
   temp=distance2;              // store the current position of our
                                                // hand in the variable temp.

   while(distance2<=50 || distance2==0)   // this loop will run until we
                                                // remove our hand in front of the
                                                // right sensor.
   {
    find_distance();            // call this function continuously to
                                                // get the live data.
   if((temp+6)<distance2)           // this condition becomes true if we
                                         // moves our hand away from the right
                                         // sensor (**but in front of it ).
                                         //here " temp+6 " is for calibration.
   {
   Serial.println("down");          // send "down" serially.
   }
   else if((temp-6)>distance2)      // this condition becomes true if we
                                         // moves our hand closer to the right
                                         // sensor.
   {
    Serial.println("up");           // send "up" serially.
   }
  }
 }
 else                           // this condition becomes true, if we
                                         // only swipe in front of the right
                                         // sensor .
 {
  Serial.println("next");          // send "next" serially.
 }
}

else if(distance1<=35 && distance1>=15)   // once if we placed our hands in
                                        // front of the left sensor in the
                                        // range between 15 to 35cm this
                                        // condition becomes true.
{

 temp=millis();

 while(millis()<=(temp+300))
 {
   find_distance();
   if(distance2<=35 && distance2>=15)  /* if our hand detects in the right
                                       sensor before 300 milli seconds this
                                       condition becomes true. ( usually it happens if
                                       we swipe our hand from left to right sensor )*/
   {
     Serial.println("Play/Pause");       // send "Play/Pause" serially.
     l=1;                         /* store 1 in variable l. ( it avoids
                                       the program to enter into the upcoming if
                                       condition )*/
     break;
   }
 }

 if(l==0)                        /* this condition will become true,
                                       only if we swipe our hand in front of left
                                       sensor.*/
 {
 Serial.println("previous");          // send "previous" serially.
 while(distance1<=35 && distance1>=15) /* this loop will rotate untill we
                                       removes our hand infront of the left
```

```
                                                        sensor. this will avoid not to enter this if
                                                        condition again.*/
    find_distance();
    }
    l=0;                          // make l=0 for the next round.
   }

 }
```

PYTHON CODE -

```
import serial                                # add Serial library for serial communication
import pyautogui                               # add pyautogui library for
                                                              #programmatically controlling
                                                              #the mouse and keyboard.

Arduino_Serial = serial.Serial('com12',9600)      # Initialize serial and Create
                                                   #Serial port object called Arduino_Serial

while 1:
    incoming_data = str (Arduino_Serial.readline()) # read the serial data and
                                                               # print it as line

    print incoming_data                          # print the incoming Serial data


    if 'next' in incoming_data:                   # if incoming data is 'next'
        pyautogui.hotkey('ctrl', 'right')          # perform "ctrl+right" operation
                                                              # which forwards the video

    if 'previous' in incoming_data:                 # if incoming data is 'previous'
        pyautogui.hotkey('ctrl', 'left')           # perform "ctrl+left" operation
                                                              #which rewinds the video

    if 'down' in incoming_data:                    # if incoming data is 'down'
        pyautogui.hotkey(‘ctrl’,'down')            # performs "ctrl+down arrow"
                                                           # operation which reduces the
                                                              # volume
        )

    if 'up' in incoming_data:                     # if incoming data is 'up'
        pyautogui.hotkey(‘ctrl’+'up')              # performs "up arrow" operation
                                                               #which scrolls up the page


    if 'Play/Pause' in incoming_data:                  # if incoming data is
         puautogui.typewrite([‘space’],0.2              )# Play/Pause


    incoming_data = "";                          # clears the dat
```