

# Controlling the VLC media player by hands in the air

*Madhukumar.s*

*Electrical and Electronics Student, Vellore Institute of Technology,*

*Vellore, Tamilnadu, 632014*

*madhukumar.s2017@vitstudent.ac.in*

**Abstract:** This paper describes my research on controlling computer as in playing games, increasing/decreasing volume, play/pause a video, drawing, autotyping etc..., by using our hands in the air without touching the mouse or the keyboard. Nowadays hand gesture controlled laptops are available but those are too costly. This project aims at using basic and cheap sensors and some programming for the same.

**Keywords:** Air, Control, Hand, Python, PyAutoGui, Sensors.

## I. INTRODUCTION

This document is the final report of my ISOI summer project 2018. This project has been monitored by my seniors at Instrumentation society of India, A active student chapter in VIT, Vellore. This project mainly aims at controlling keyboard and mouse functionalities by a python script and the Arduino sketch. Extending the above concept and applying to control the VLC media player's volume, forwarding/rewinding a video, play/pause a video without physically using the keyboard and mouse. The above mentioned is possible by a python program using pyautogui library, some sensors, and a microcontroller.

## II. COMPONENTS USED

Hardware components:

- 1.Arduino UNO
- 2.Ultrasonic sensor(two in number)
- 3.Breadboard
- 4.Jumper wires

Softwares used:

- 1.Python IDLE 2.7.9 + pyautogui library.
- 2.Arduino IDE

**Arduino UNO :**



The Arduino UNO is a widely used open-source microcontroller board based on the ATmega328P microcontroller and developed by Arduino.cc. It is used to interface with ultrasonic sensors.

**Ultrasonic sensors**



An ultrasonic sensor is used to measure the distance of the object present in front of the ultrasonic sensor using ultrasonic sound waves.

### Arduino IDE



The Arduino integrated development environment (IDE) is a cross-platform application that supports C and C++ programming language.

### Python IDLE 2.7.9



IDLE is an integrated development environment for Python where we write our python code.

### PyAutoGUI library

- PyAutoGUI is a Python module for programmatically controlling the mouse and keyboard.
- This module works on python version 2 & 3.
- It can also be used for image recognition.
- First, we need to install pyautogui module and we have to import in the python shell.

Basic functions for Controlling keyboard programmatically:

#### 1. Automatically typing a text:

```
>>> pyautogui.typewrite('Hi everybody')  
Hi everybody
```

The above commands type out 'Hi everybody' instantly.

```
>>> pyautogui.typewrite('Hi everybody',0.2)  
Hi everybody
```

The above command types out 'Hi everybody' with a 0.2-second delay after each character.

From these two commands, we can observe that we are programmatically pressing the alphabet keys on the keyboard.

Pressing 'enter', 'escape' and 'function keys':

```
>>> pyautogui.press('enter')  
  
>>> pyautogui.press('f1')  
  
>>> pyautogui.press('esc')
```

Keep on pressing :

```
>>> pyautogui.keyDown('shift')
```

The above command keeps on pressing shift button for a very long time.

```
>>> pyautogui.keyUp('shift')
```

The above command releases the shift key which was being pressed by keyDown('shift') command.

Pressing two key like the ctrl+up arrow, ctrl+down arrow :

```
>>> pyautogui.hotkey('ctrl','up')
```

The above command keeps on pressing Up arrow key for a very long time.

```
>>> pyautogui.hotkey('ctrl','down')
```

The above command keeps on pressing down arrow key for a very long time.

This property of controlling keyboard programmatically has been used in my project.

### III. Controlling VLC media player

Control 1: play/pause a video by pressing spacebar programmatically whenever both the right and left hands are at a distance >40 and <60 cm.

Control 2: Increasing volume by pressing ctrl+up arrow key programmatically whenever left hand alone is at a distance between 13cm and 17 cm for some time and then moved to a distance of less than 10cm.

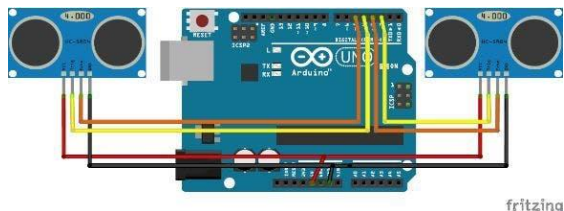
Control 3: Decreasing volume by pressing ctrl+down arrow key programmatically whenever left hand alone is at a distance between 13cm and 17 cm for some time and then moved to a distance greater than 20cm.

Control 3: Forwarding a video by pressing ctrl+right arrow key programmatically whenever right hand alone is at a distance between 13cm and 17 cm for some time and then moved to a distance greater than 20cm.

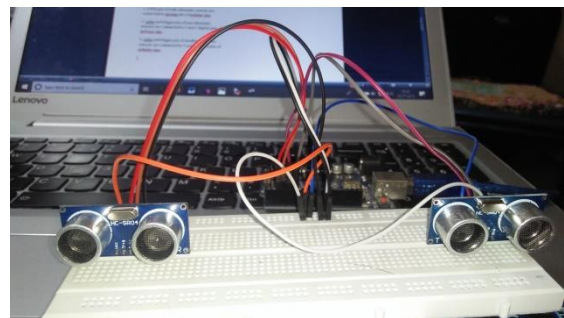
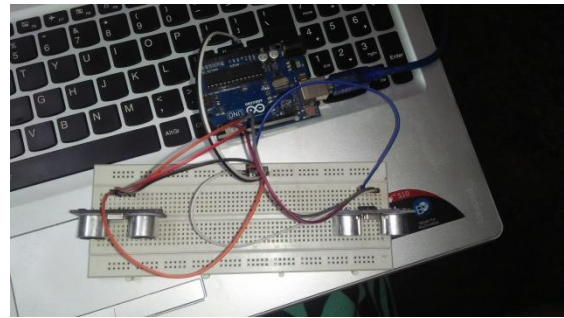
Control 3: Rewinding a video by pressing ctrl+left arrow key programmatically whenever left hand alone is at a distance between 13cm and 17 cm for some time and then moved to a distance of less than 10cm.

\*All distances are from the ultrasonic sensor and the hand.

### Circuit diagram/connection:



- 1.. VCC pin of both ultrasonic sensors is connected to the 5v pin of Arduino Uno.
2. GND pin of both ultrasonic sensors is connected to ground pin of Arduino UNO.
3. echo and trigger pin of one ultrasonic sensor is connected to 3 and 2 digital pins of Arduino Uno.
4. echo and trigger pin of another ultrasonic sensor are connected to 5 and 4 digital pins of Arduino UNO.



### Arduino code.

```
const int trigger1 = 2; //Trigger pin of 1st Sensor
const int echo1 = 3; //Echo pin of 1st Sensor
const int trigger2 = 4; //Trigger pin of 2nd Sensor
const int echo2 = 5; //Echo pin of 2nd Sensor
long time_taken;
int dist, distL, distR;
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>60)
    dist = 60;
}

void loop() { //infinite loop
  calculate_distance(trigger1,echo1);
  distL =dist; //get distance of left sensor
  calculate_distance(trigger2,echo2);
```

```

distR =dist; //get distance of right sensor
//Pause Modes -Hold
if ((distL >40 && distR<40) && (distL <60 && distR<60)) //Detect both hand
{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==13 && distL<=17)
{
  delay(100); //Hand Hold Time
  calculate_distance(trigger1,echo1);
  distL =dist;
  if (distL==13 && distL<=17)
  {
    Serial.println("Left Locked");
    while(distL<=40)
    {
      calculate_distance(trigger1,echo1);
      distL =dist;
      if (distL<10) //Hand pushed in
      {Serial.println ("Volume Increased"); delay (300);}
      if (distL>20) //Hand pulled out
      {Serial.println ("Volume Decreased"); delay (300);}
    }
  }
}
//Lock Right - Control Mode
if (distR==13 && distR<=17)
{
  delay(100); //Hand Hold Time
  calculate_distance(trigger2,echo2);
  distR =dist;
  if (distR==13 && distR<=17)
  {
    Serial.println("Right Locked");
    while(distR<=40)
    {
      calculate_distance(trigger2,echo2);
      distR =dist;
      if (distR<10) //Right hand pushed in
      {Serial.println ("Rewind"); delay (300);}
      if (distR>20) //Right hand pulled out
      {Serial.println ("Forward"); delay (300);}
    }
  }
}
delay(200);
}

```

\*[Arduino code](#) to download.

- whenever both the right and left hands are at a distance >40 and <60 cm Arduino code it prints 'play/pause' in the serial monitor.
- whenever left hand alone is at a distance between 13cm and 17 cm for some time and then moved to a distance less than 10cm Arduino code prints 'Volume Increased' in the serial monitor.
- whenever left hand alone is at a distance between 13cm and 17 cm for some time and then moved to a distance greater than 20cm Arduino code prints 'Volume Decreased' in the serial monitor.
- whenever right hand alone is at a distance between 13cm and 17 cm for

some time and then moved to a distance greater than 20cm. arduino code prints 'Forward' in the serial monitor.

- whenever right hand alone is at a distance between 13cm and 17 cm for some time and then moved to a distance of less than 10cm. Arduino code prints 'Rewind' in the serial monitor.

### Python code

```

import serial #Serial imported for Serial communication
import time #Required to use delay functions
import pyautogui #Required to to perform actions
ArduinoSerial = serial.Serial('/com5',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 'Rewind' in incoming:
        pyautogui.hotkey('ctrl', 'left')
    if 'Forward' in incoming:
        pyautogui.hotkey('ctrl', 'right')
    if 'Volume Increased' in incoming:
        pyautogui.hotkey('ctrl', 'up')

    if 'Volume Decreased' in incoming:
        pyautogui.hotkey('ctrl', 'down')
    incoming = "";

```

\*Download [python code](#)

This python code gets the serial data from the Arduino.

- If it gets 'Play/Pause' string then the code presses down the space bar key for 0.2 seconds.
- If it gets 'Rewind' string then the code presses down the Ctrl+left arrow key.
- If it gets 'Forward' string then the code presses down the Ctrl+right arrow key
- If it gets 'Volume Increased' string then the code presses down the Ctrl+up arrow key.

- If it gets 'Volume Decreased' string then the code presses down the Ctrl+down arrow key.



[Click here to watch controlling VLC media player](#)

#### **IV. Conclusion**

So finally we can control the keyboard and functionalities programmatically without physically touching the keyboard/mouse.

Controlling VLC media player is just a simple application of what we can do using pyautogui python module and ultrasonic sensor.

#### **V. Future Goals**

- We can extend this project to play a car game using the ultrasonic sensor or IR sensors.



[GTA 3 click here to watch video](#)

[Arduino code](#)

[Python code](#)

- We can also use webcam and image recognition to control the computer functionalities programmatically.

#### **VI. Acknowledgement**

I would like to extend my sincere thanks to my Seniors at ISOI technical team, VIT, Vellore for accepting this project as my ISOI's summer project and their constant help throughout this project.

