

Arduino based Hand Gesture Control of Computer Application

Abstract:

You might have seen Hand Gesture Controlled Robots, where the motion of a robot is controlled by the gestures of the hand. Another interesting project based on a similar principle is an Arduino based Hand Gesture Control of your computer or laptop. Human Machine Interface or HMI is a system comprising of hardware and software that helps in communication and exchange of information between the user (human operator) and the machine. We normally use LED Indicators, 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 Gestures. Instead of using a keyboard, mouse or joystick, we can use our hand gestures to control certain functions of a computer like play/pause a video, move left/right in a photo slide show, scroll up/down in a web page and many more. In this project, we have implemented a simple Arduino based hand gesture control where you to determine the position of our hand and control a media player (VLC) based on the position.

Keywords : Arduino, Gesture, Laptop, PyAutoGUI, Python, Ultrasonic

INTRODUCTION

Humans interact in the physical world by the means of the five senses. However, gestures have been an important means of communication in the physical world from ancient times, even before the invention of any language. In this era of machines taking control of every complex works, interactions with machines have become more important than ever. Since this paper deals with gesture controlled laptop, the primary focus will be on the use of hand gestures for specific applications only. There are several ways to capture a human gesture that a computer would be able to recognize. The gesture can be captured using distance measurement, camera, or a data glove. Gestures can also be captured via Bluetooth or infrared waves, Acoustic, Tactile, optical or motion technological means. The embedded systems designed for specific control functions can be optimized to reduce the size and cost of the device and increase the reliability and performance.

This project consists of mainly three components – Arduino Uno, Ultrasonic sensors, and a laptop. The ultrasonic sensors hooked to the Arduino are used to determine the gestures and the distance of the hand from the ultrasonic sensors. The code loaded in Arduino finds the respective keyword for the distance found and sends it to Windows OS. Python code that runs in the background recognizes the keywords and generates the corresponding virtualkeystrokes for Windows. The hotkeys then control particular function of the application of intend to run, that is VLC Media Player.

The Arduino coding is done in “Arduino IDE” and uploaded to Arduino Uno. To run the required python code, we need to complete a few more steps.

- (1) Python 2.7.14 is downloaded and installed.
- (2) “pip” function is upgraded, a tool to easily update python modules.
- (3) PyAutoGUI module is downloaded using the pip function and downloaded through internet in the Python command screen itself.
- (4) Then we open “IDLE (Python GUI)”.
- (5) We create a new file and type in the code.

(6) The file is then saved and run

With the above steps done, python will be able to receive feedback from the Arduino board and hence generate virtual keystrokes in Windows OS. The keystrokes then control the hotkeys of the VLC media player.

WORKING OF THE LAPTOP

2.1 Basic Layout

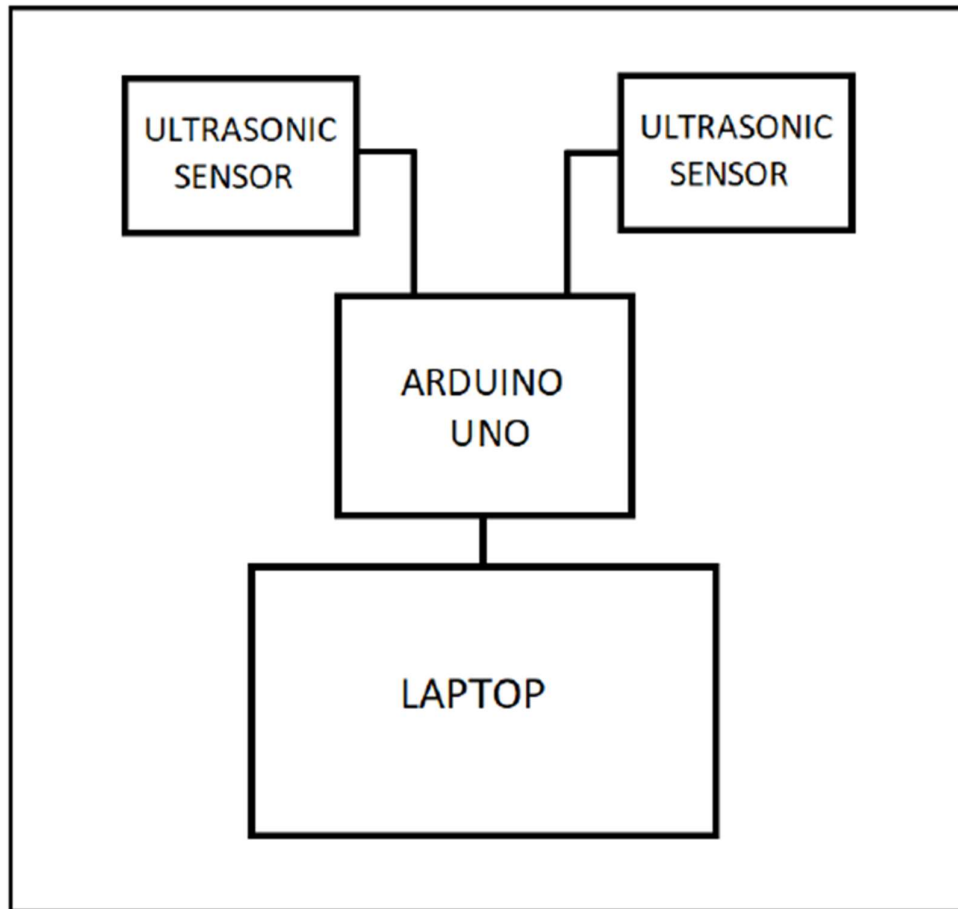


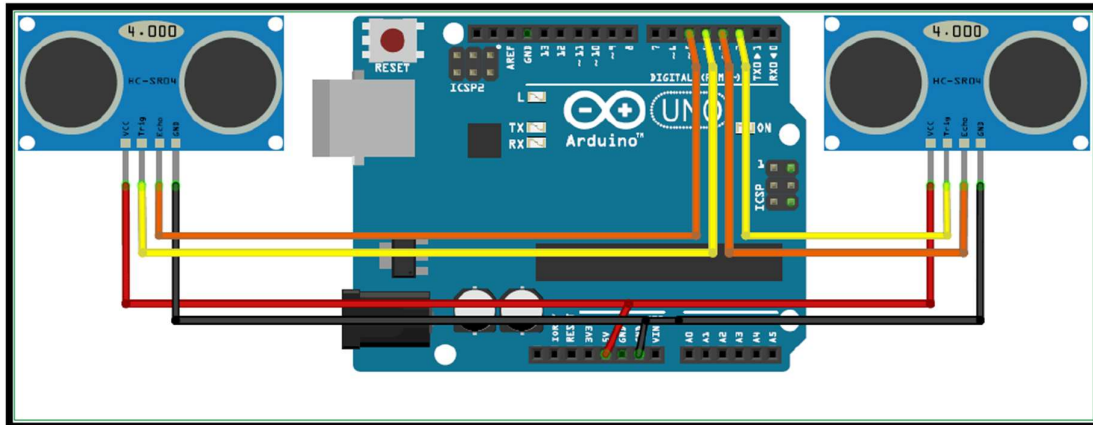
Fig 2.1 : Block Diagram

From the above figure 2.1, we see that two ultrasonic sensors are connected to Arduino Uno board and the board is connected to Laptop through USB. The ultrasonic sensors have a transmitter and a receiver where the transmitter emits ultrasonic waves. The waves hit a surface in front of the sensor and any reflected waves are picked up by the transmitter and based on the intensity the distance of the object is determined. This data is received by the Arduino running in the background of Windows. Python picks up the keywords and generates virtual keystrokes of hot keys for the VLC media player. This process is kept on a continuous loop to control media playback, either video or audio files.

The ultrasonic sensors are the components that read the hand gesture and the distance of the hand from the sensor. For the left sensor, bringing our hand closer to it will rewind the video

and moving away from it will fast forward the video. In the right sensor, the same actions would result in controlling the volume of the player. and looks for specific keywords within the code according to the distance that is to be sent to the Python code

Circuit Layout



As we see in Fig 2.2 here, two ultrasonic sensors are connected to Arduino Uno board and the board is connected to Laptop through USB. The Vcc pin of each sensor is provided by Arduino using the 5V output port. Grounding of both sensors is attached to GND0. The Trigger and Echo pins are connected as per the code written. The ultrasonic sensors are equipped with a transmitter and a receiver where the transmitter emits ultrasonic waves. The waves hit a surface in front of the sensor and any reflected waves are picked up by the transmitter and based on the intensity the distance of the object is determined. This data is received by the Arduino and looks for specific keywords within the code according to the distance that is to be sent to the Python code running in the background of Windows. Python picks up the keywords and generates virtual keystrokes of hot keys for the VLC media player. This process is kept on a continuous loop to control media playback, either video or audio files.

The ultrasonic sensors are the components that read the hand gesture and the distance of the hand from the sensor. For the left sensor, bringing our hand closer to it will rewind the video and moving away from it will fast forward the video. In the right sensor, the same actions would result in controlling the volume of the player.

2.3 Software Used

To read hand gestures, will need to program an Arduino and design a Python code. As mentioned earlier that

Windows does not recognize commands given by Arduino, hence we use Python to establish serial connection and turn the data received into virtual keystrokes. These virtual keystrokes control the Hotkeys of VLC media player which in turn will control playback of video or audio being played.

The Arduino IDE software works well for programming the Arduino but for Python we need to perform a few more

steps after installing Python 2.7 as shown in Fig 2.3. The steps are

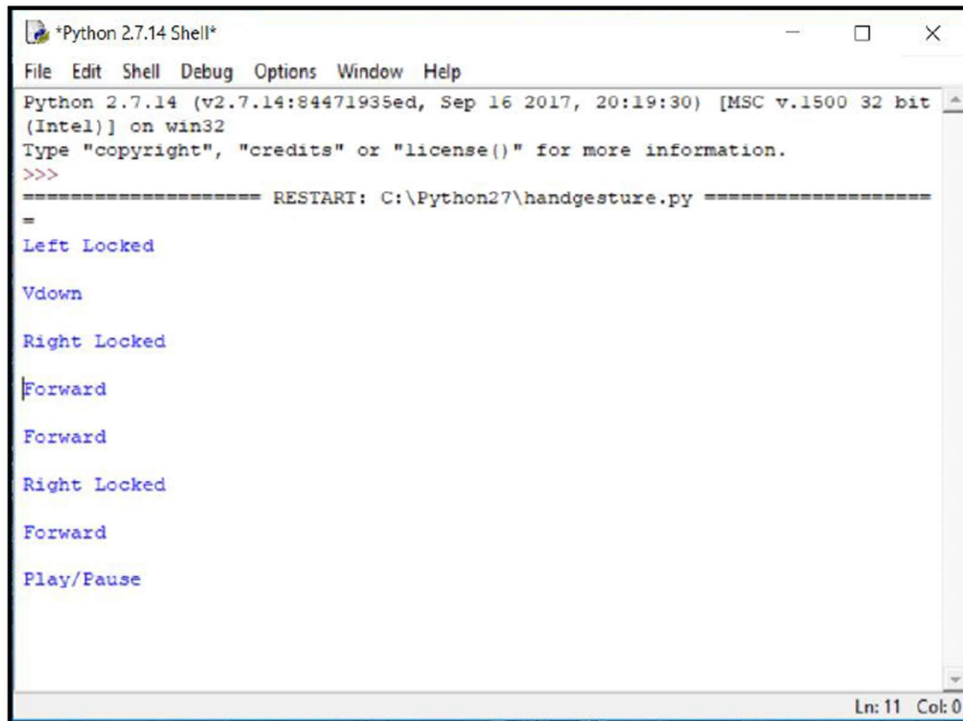
- (1) Upgrade pip function of Python
- (2) Install the PyAutoGUI module using pip function.

Hardware Used

For this, we use an Arduino Uno connected to an Asus laptop using the USB port as shown in Fig 2.4.

Execution of the Program.

When the setup is completely done, we run the python code. Fig 2.5 is how the screen would appear once we click on Run Module. Now we can use our hand gestures near the ultrasonic sensors where necessary actions will be executed.



```
Python 2.7.14 Shell*
File Edit Shell Debug Options Window Help
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Python27\handgesture.py =====
=
Left Locked
Vdown
Right Locked
Forward
Forward
Right Locked
Forward
Play/Pause
Ln: 11 Col: 0
```

ig. 2.5: Python code beng executed according to inputs from Arduino

The left ultrasonic sensor is coded such that bringing our hand closer to the sensor will Rewind and farther from it will Fast Forward the movie or video we play in VLC. Similarly, the right ultrasonic sensor is used to decrease the volume when we bring our hand closer to the sensor and increase the volume when we bring our hand away from it. The same is read by the python code and displayed in blue font.

When we place our hand in front of the left sensor, the input gets activated for the left sensor only and the screen displays “Left Locked” and reads the input as Vdown or Vup, in short for Volume up and Volume down. The same process happens for the right hand. Only when we want to pause, we bring both our hands in front of each of the sensors. Now we can use hand gestures to control VLC Media Player as in

III. HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENT	SOFTWARE REQUIREMENT
<ol style="list-style-type: none">1. HC-SR04 Ultrasonic Sensor2. Arduino Uno3. Laptop (Windows 7/8/10 OS)	<ol style="list-style-type: none">1. Arduino IDE2. Python 2.7.143. VLC Media Player

Table 3.1 Hardware and Software requirements

Algorithms used:

step 1:

First the ultrasonic sensor will capture the gesture.

step 2:

The captured gesture will be compared with the stored gestures.

step 3:

If the gesture matches with the stored gesture then a particular action will be executed by the system.

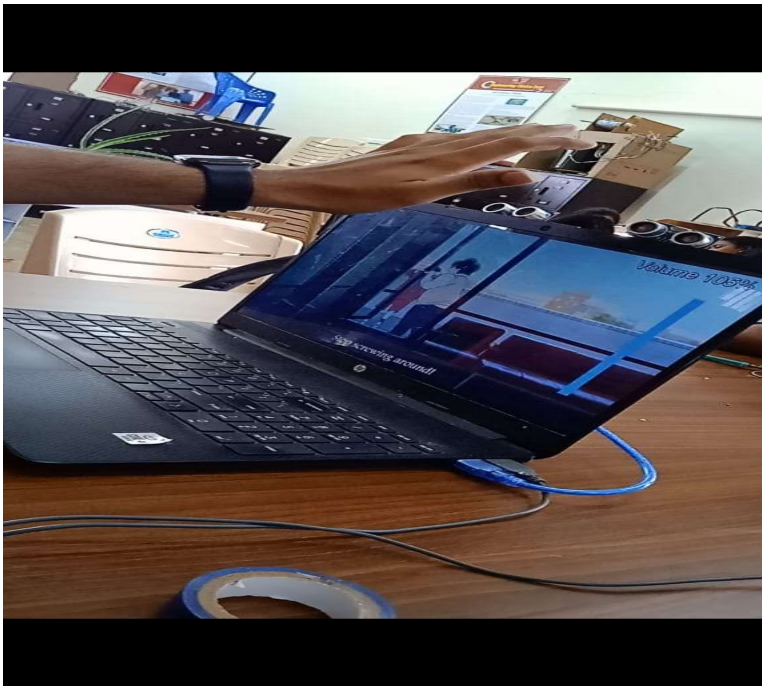
step 4:

If the gesture did not match stored gesture then system will not perform any action



The above figure shows the fast forwarding of video when left hand is placed at a particular distance less than 20cm from the left sensor. The above figure shows how the video is paused

or played when both the hands are placed at a particular distance between 40 to 50 cm from the sensor



The above figure shows how the volume of video is decreased when hand is pushed away from the right sensor at a distance greater than 20cm. The above figure shows how the volume of video is increased when hand is pushed towards the right sensor at a distance less than 20cm.

ARDUINO CODE:

```
/*
```

```
* Program for gesture control VLC Player
```

```
* Controlled using Python
```

```
* Code by B.Aswinth Raj
```

```
* Dated: 11-10-2017
```

```
* Website: www.circuitdigest.com
```

```
*/
```

```
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 Sesnor
```

```
const int echo2 = 5; //Echo pin of 2nd Sesnor
```

```
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>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 debudding

/*Serial.print("L=");

Serial.println(distL);

Serial.print("R=");

Serial.println(distR);

*/

//Pause Modes -Hold

if ((distL >40 && distR>40) && (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>=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 ("Vup"); delay (300);}

            if (distL>20) //Hand pulled out

```

```

        {Serial.println ("Vdown"); 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);
```

```
}
```

Python Code:

```
import serial #Serial imported for Serial communication
```

```
import time #Required to use delay functions
```

```
import pyautogui
```

```
ArduinoSerial = serial.Serial('com18',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 'Vup' in incoming:
```

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

```
if 'Vdown' in incoming:
```

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

```
incoming = "";
```

CONCLUSION:

Hand gesture laptop uses an Arduino Uno, Ultrasonic sensors and a laptop to carry out the operation of controlling media playback and volume. It is mainly aimed at reducing the effort of interaction with computers through input devices using simple gestures. It is also done to increase the interactivity with computers. This type of technology

REFERENCES:

<https://www.instructables.com/PC-Apps-Control-Using-Arduino/>

<https://circuitdigest.com/microcontroller-projects/control-your-computer-with-hand-gestures>