Amrita Vishwa Vidyapeetham
Centre for Excellence in Computational Engineering and Networking
Amrita School of Engineering, Coimbatore

Arduino based Gesture Controlled Computer

Prepared By:
M.D.S. Rama Saran
Mohitha.V
J. Ajay Surya
P. Sai Ravula

Supervised by:
Dr. Ambika P.S
Asst. Professor

An End Semester Project submitted to the CEN department as a part of course evaluations of **"Principles of Measurements and Sensors"** for B. tech in **Computer Science Engineering** – **Artificial Intelligence.**

ACKNOWLEDGMENT

We are deeply thankful to **Centre for Excellence in Computational Engineering and Networking** (CEN) at **Amrita Vishwa Vidyapeetham**, **Coimbatore** for providing us such a wonderful environment to peruse our research. We would like to express our sincere gratitude to **Dr. Ambika PS, Asst. Professor, Department of Centre for Excellence CEN**, Amrita Vishwa Vidyapeetham. We have completed our research under her guidance. We found the research area, topic, and problem with her suggestions. She guided us with our study, and supplied us many research papers and academic resources in this area. She is patient and responsible. When we had questions and needed her help, she would always find time to meet and discuss with us no matter how busy she was.

We would also like to extend our gratitude to **Vinod CK**, Asst. Professor **Department of Centre for Excellence** CEN, Amrita Vishwa Vidyapeetham, who provided us with the base knowledge of this field of research, without him this research work would not exist in its present form. We would also like to acknowledge our team members for supporting each other and be grateful to our university for providing this opportunity for us. Lastly special thanks to Centre for Excellence in CEN for providing this opportunity to research in this field.

DECLARATION

We do hereby declare that the research works presented in this project entitled, "**Arduino based Gesture controlled Computer**" are the results of our own works. We further declare that the project has been compiled and written by us under the guidance of our supervisor. The materials that are obtained from other sources are duly acknowledged in this project.

Authors:

M. Devi Siva Rama Saran
RN: CB.EN.U4AIE21034

Mohitha Velagapudi
RN: CB.EN.U4AIE21032

J.R.S. Ajay Surya
RN: CB.EN.U4AIE21018

P. Sai Ravula
RN: CB.EN.U4AIE21041

TABLE OF CONTENTS

LIST OF TABLES

BIBLIOGRAPHY

Arduino based Gesture Controlled Computer

1.1 Abstract

In this project, we are developing a gesture – controlled laptop. This build is based on usage of information exchange between Arduino and python.

Instead of using external input/output devices such as keyboard, mouse or touchpad etc, we can use sensors which uses our hand gestures to manipulate some actions in our device. Some of these actions include forwarding and rewinding videos in VLC media player, volume controlling in VLC, capturing a screenshot, scrolling in webpages etc.

1.2 Introduction

With change in time, new technologies are being developed. One such development in electronic world is Gesture controlled robots. These robots are operated solely using hand gestures. Also, people nowadays are keen on using hands – free operating electronic gadgets. These hand gestures are given by means of a sensor called Ultrasonic sensors.

This project deals with a similar type of real-time usage of the ultrasound sensor. The sensor is used to calculate the change in hand gesture, which will be further used to calculate the change in distance. This data will be used to

perform the triggered action over the operating system. Then, Arduino will request the system processor to perform the requested action by the user. The time taken to perform these basic operations play/pause video, screenshot capturing etc operations are reduced because of this proposed model.

Also, our model is very cheaper compared to complex systems employing the same usage of sensors since they have to produce accurate results.

1.3 Objective

Our main objective by proposing this model is to minimize the usage of external Input/output devices to interact with the computer. Instead, we use sensors that maps our hand gestures to perform certain actions in the device.

The primary benefit of employing the sensors for interacting with the electronic devices is that they can be very affordable and helps in development of touchless systems.

1.4 Human Machine Interface

Human Machine Interfaces (HMIs) are modern software which is a user interface that connects a person to an electronic device. They are capable of performing various tasks and act as bridges between the human operator and the complex internal working systems of a device allowing the user to control functions of that device.

HMIs comes in various forms such as built-in screens to the devices, monitors but irrespective of their form, their purpose is to provide insight on how to interact with that particular device.

Our project is entirely based on the HMI since hand gestures that we employ are a way in which we are controlling the actions in our computer. Also, there are different interfaces like computer screen used in our project also.

1.5 Hardware Requirements

We need some hardware equipment for assembling our project which are listed as shown.

- Arduino UNO board (1) – for processing the code for our desired implementation.

Fig – 1.5.1 (Arduino UNO board)

- HC SR- 04 Ultrasonic sensors (1) – for capturing the hand gestures and perform its functionality.

Fig – 1.5.2 (HC SR -04 Ultrasonic sensor)

- Jumper wires male to female type (4) – To connect the ultrasonic sensor to the Arduino board.

Fig – 1.5.3 (Male to female jumper wires)

- Type - B USB cable (1) - To connect our Arduino board to laptop / computer.

Fig – 1.5.4 (USB cable type - B)

- A computer / laptop (1) - To test our program.

Fig – 1.5.5 (Laptop)

1.6 Software requirements

Some software has to be installed in the device before testing our model. There are 2 software that have to be installed on our laptop.

- **Arduino IDE –** This is the platform where we load our Arduino code, compile and upload it to the Arduino board. It is basically an interface to interact with the ARDUINO board.

- **Python IDLE –** Python is another software platform where we load our python code which is used to communicate with the serial ports. We also need some extra libraries to be downloaded in python.

1.7 Python Libraries

For our experiment to work, we have to install another 2 packages for python which are necessary for information exchange between Arduino and computer. The two packages are as follows;

Pyserial – This package is used to access the serial ports of the microcontrollers which also works as backend for Python. Python commands can be used to access serial port settings. It is used to read and write serial data into the microcontroller which is Arduino in our case.

After installing Python from the official website of Python, we can install Pyserial library from the command prompt by using the command "**pip install pyserial**" which installs this package for every environment of computer.

PyautoGUI – This package is a cross – platform GUI automation module that works with all the Python versions and is used to control the mouse and keyboard as well as to perform some basic image recognition for automating tasks on our computer. It contains several functions related to keyboard, mouse, message box, alerts etc. It is helpful in asserting a task to each gesture.

Pyautogui can be installed in the same way as pyserial from the command prompt by using the command "**pip install pyautogui**".

1.8 Architecture of model

Our model after complete assembling of all the components will be as shown in the picture.

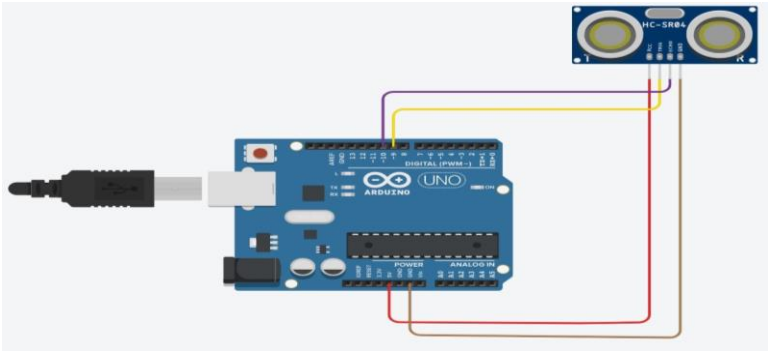


Fig – 1.8.1 (Schematic diagram)

Fig – 1.8.2 (Assembled model)

As we know, our HC SR04 ultrasonic sensor has 4 different pins. **Vcc** pin is connected to 5V port of Arduino board, **TRIG** pin is connected to port 9 of digital pins, **ECHO** pin is connected to port 10 of digital PWMs of Arduino board and **ground** pin is connected to ground port of Arduino board.

1.9 HC SR04 Ultrasonic sensor

As already said before, an HC – SR04 Ultrasonic sensor is a 4-pin module which is used in many applications where measuring distance or sensing objects is necessary. The module has two eyes like projections in the front which forms ultrasonic transmitter and receiver.

The features of an average HC – SR04 Ultrasonic sensor are

- Operating voltage - +5V

- Theoretical measuring distance – 2cm to 450cm
- Practical measuring distance – 2cm to 80cm
- Accuracy – 3mm
- Measuring angle covered – less than 15^{o}
- Operating current – less than 15mA
- Operating frequency – 40Hz

HC – SR04 has 4 different pins which has different functionalities as shown

- **VCC pin –** Connected to 5V of Arduino board for supplying power to the sensor
- **Trig pin –** Used to initialize measurement by transmitting ultrasonic waves by making this pin high for 10us.
- **Echo pin –** An output pin which goes high for a specific time and it will be equivalent to the duration of the time for the wave to return back to the sensor.
- **Ground pin –** Connected to the ground port of the Arduino.

Working of this ultrasonic sensor is very simple. At first, give the power supply to the sensor to turn on and connect the GND pin of this sensor to the GND pin of the Arduino board. And the sensor module can be powered up with the voltage supply of the Arduino board when the current which is drawn through the sensor is below 15mA. So, the Arduino current ratings will not affect the sensor.

Once the set-up is complete, connect both the pins of sensor (Trig & Echo) to the Arduino board's input/output pins. As we discussed earlier, the Trig pin in the sensor must be kept 10us in the beginning in order to start the measurement method. So, this sensor module will generate sound waves by the 40,000 Hz frequency around for every second from the source.

When the sound waves return back, the Echo pin will activate until these waves are obtained by the receiver. The time will be measured with the help of an Arduino board. Also, the distance can be calculated using the formula

$$S = (V \times t) / 2$$

Where, **S** is the required object distance

V is the speed of sound in room conditions (340m/s)

t is the time for the sound waves to return back after striking the object.

1.10 Methodology

We followed a unique logic for implementing our model. In Arduino code, we set the required pins as high and calculate the distance between our hand and the sensor. We will not take a single value of distance, rather we will take a continuous data of distance values. Then, we will print this distance value in the Arduino code.

Now, in Python code we will import both the python libraries. Then, we will take this data of distance values from the serial ports and append it into an empty list created. Before

appending, we will convert the float values of those distance values into integer values.

From this data, we will calculate standard deviation and mean of the data of a particular range. Then, based on the mean obtained we have associated different operation in our computer for different ranges of distances.

1.11 Hand Gestures

We have created 3 different python files each corresponding for different functions.

- **vlc_control.py** – For controlling action in VLC media player
- **browser_control.py** – For controlling actions in a browser
- **ppt_control.py** – For controlling actions in Microsoft PowerPoint

Each python file will have different functions and hence they will also have different gesture commands. Hand must be kept within the range of mean with respect to that function.

For VLC controlling –

S. No	Range of mean	Assigned hotkey	Function
1	[0,10]	Space	Pause the video
2	(10,20]	Ctrl + up	Volume up
3	(20,30]	Ctrl + down	Volume down
4	(30,40]	Ctrl + right	Forward video
5	(40,50]	Ctrl + left	Rewind video

Table – 1.11.1 (VLC controlling)

So, our hand has to placed according to the mean range.

For browser controlling –

S. No	Range of mean	Assigned hotkey	Function
1	[0,10]	Fn+win+prtsc	Screenshot
2	(10,20]	Pgup	Scrolls page up
3	(20,30]	Pgdn	Scroll page down
4	(30,40]	Alert box	Message box

Table – 1.11.1 (browser controlling)

For PowerPoint controlling –

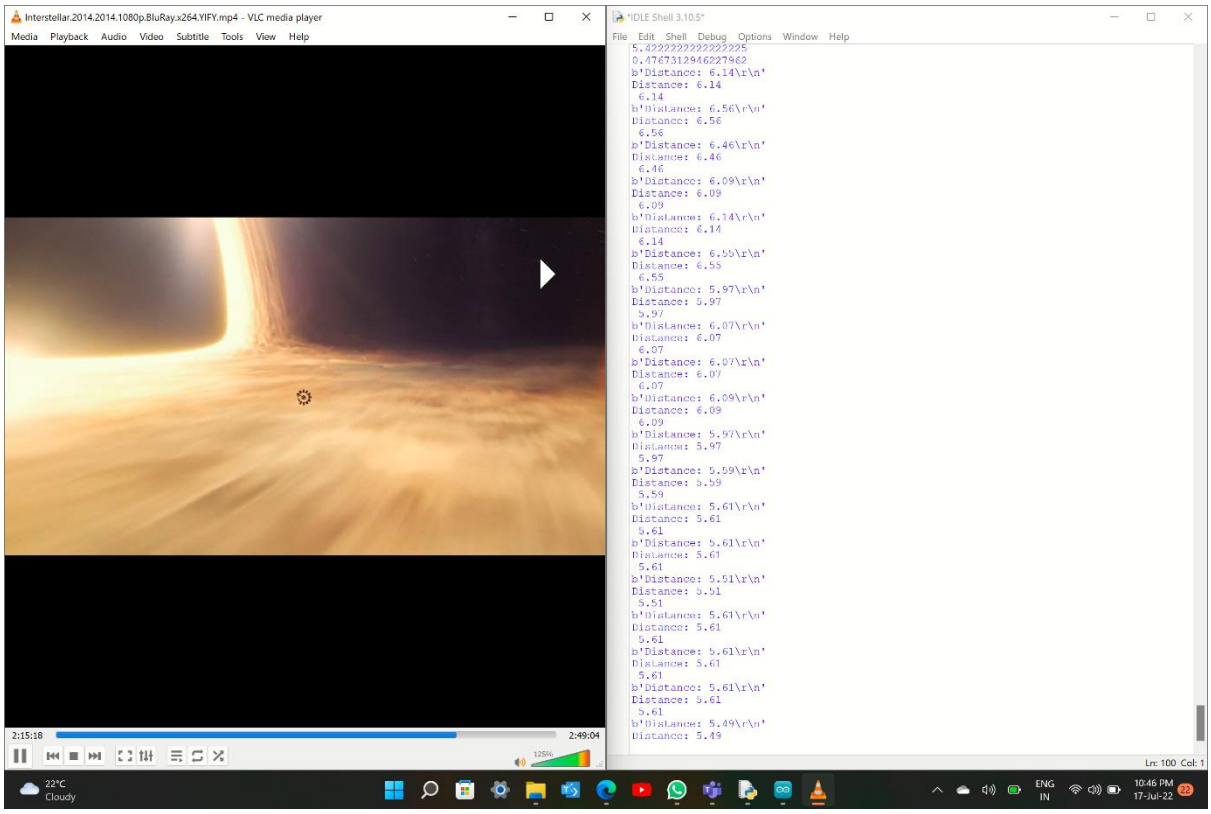
S. No	Range of mean	Assigned hotkey	Function
1	[0,10]	down	Next slide
2	(10,20]	up	Previous slide
3	(20,30]	Ctrl + l	On laser pointer
4	(30,40]	home	Go to 1st slide

Table – 1.11.1 (PowerPoint controlling)

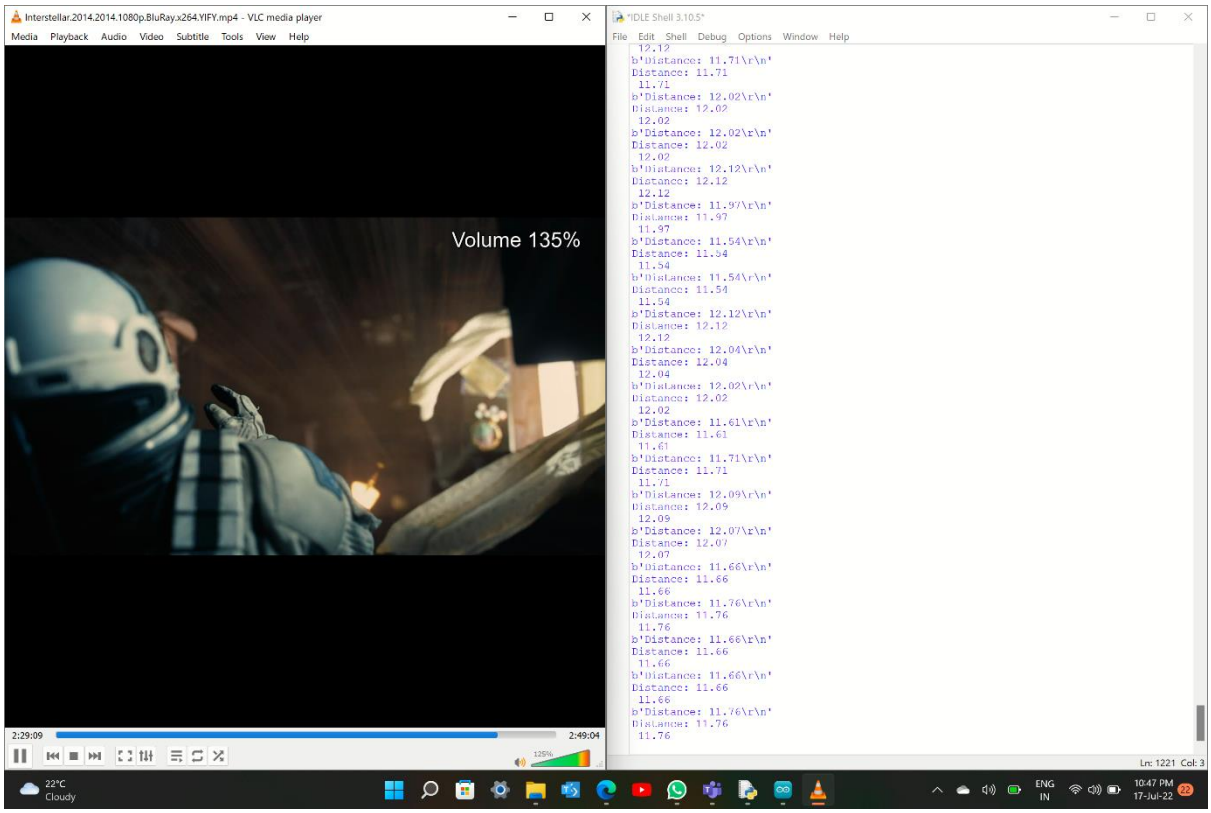
1.12 Proposed implementations

Now let us see some implementations of our Arduino based gesture control computer.

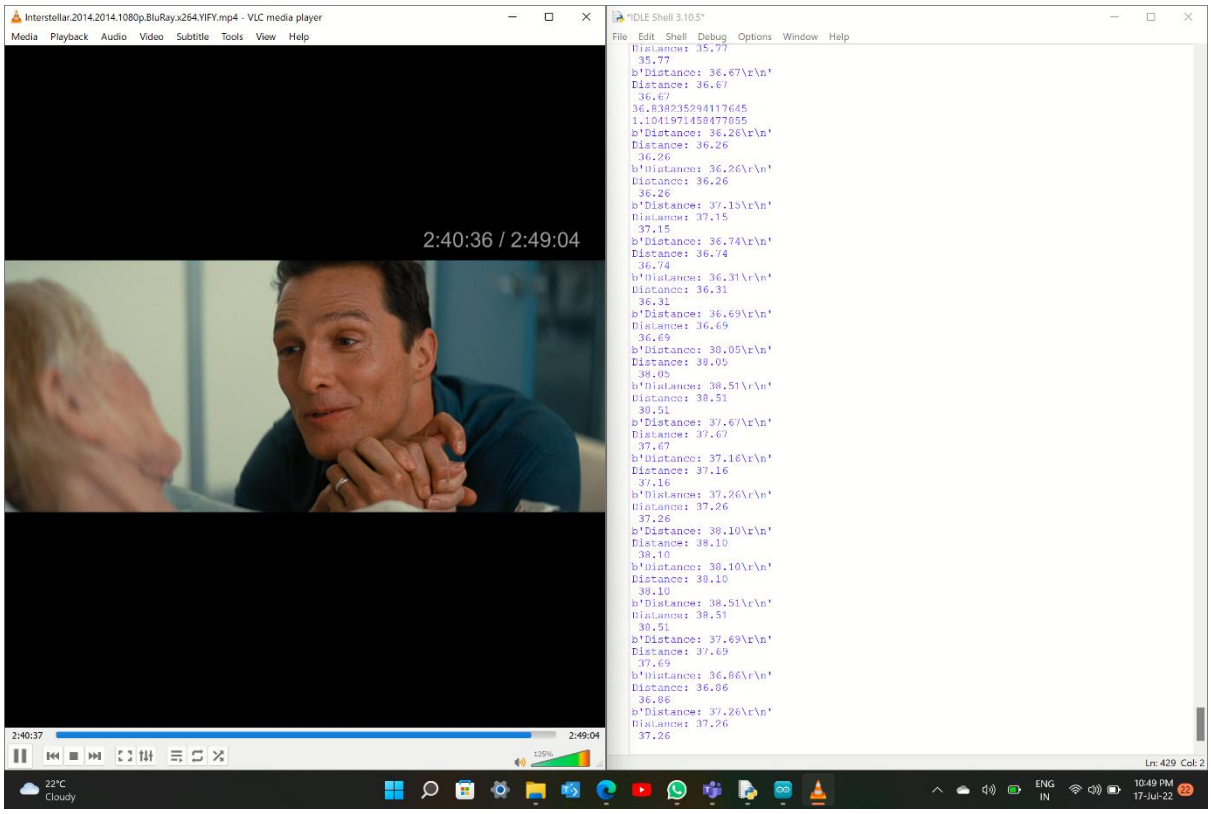
Playing/ pausing a video in VLC media player – **(Fig – 1.12.1)**



Volume controlling of a video in VLC media player – **(Fig – 1.12.2)**



Forwarding a video in VLC media player – **(Fig – 1.12.3)**



1.13 Analysis

The project is designed to reduce the use of physical buttons to an extent and use hand gestures to control the computer actions. The project is reliable and sustainable. Users can use hand gestures like keep a hand near the sensors to controlling actions in VLC media player, scrolling through webpages, capturing a screenshot, displaying a message box, controlling actions in Microsoft PowerPoint etc. This entire process is the work of ultrasonic sensors which is also supported by Arduino and Pyautogui library.

1.14 Conclusions

By applying our methodology, we conclude the following, no matter how powerful and complex, we always have to be near a machine and somehow in physical contact with it to interact with it. But gesture recognition technology could change all that. If perfected and used correctly, it could actually render traditional input devices like keyboard, mice and touch screens redundant. Besides being really functional, it cannot be argued against that gesture control looks really cool. Like in the 2002 hit movie Minority Report, there will probably come a time where everything is controlled by gestures. While it might seem like a technology that will only increase our lethargy, truth is that other than making life easier, it also has a vast array of applications in almost all fields.

BIBLIOGRAPHY

Testing — PyAutoGUI documentation

HC-SR04 Ultrasonic Sensor : Pin Configuration, Working and Applications (elprocus.com)