**MINI PROJECT**

**(2020-21)**

**SMART GESTURE SYSTEM USING IOT**

**Institute of Engineering & Technology**

**Submitted by** –

* ISHA PARVEEN (181500278)
* RITIK LAMBA (181500577)
* JAYESH GUPTA (181500297)
* KAUSTUBH SISODIA (181500318)
* UTKARSH RATHORE (181500771)

**Submitted To-**

* Mr AMIR KHAN (TECHNICAL TRAINER)



**Department of Computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuhan,**

**Mathura – 281406**

***Declaration***

*We hereby declare that the work which is being presented in the Mini Project “****Hand Gesture System using IOT”,*** *in partial fulfillment of the requirements for Mini-Project LAB, is an authentic record of our own work carried under the supervision of* ***Amir Khan, Technical Trainer, GLA University, Mathura****.*

* ISHA PARVEEN (181500278)
* RITIK LAMBA (181500577)
* JAYESH GUPTA (181500297)
* KAUSTUBH SISODIA (181500318)
* UTKARSH RATHORE (181500771)



**Department of Computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuhan,**

**Mathura – 281406**

**CERTIFICATE**

This is to certify that the project entitled ***“****Hand Gesture System using IOT****”*** carried out in Mini Project Lab is a bonafide work done by *Kaustubh Sisodia (181500318), Ritik Lamba (181500577), Jayesh Gupta (181500297), Utkarsh Rathore (181500771) and Isha Parveen (181500278)* is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

**Signature of Supervisor:**

**Name of Supervisor: Amir Khan**

**ACKNOWLEDGEMENT**

*It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.*

*Our heartiest thanks to Dr. (Prof). Anand Singh Jalal, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.*

*We owe special debt of gratitude to Amir Khan, Technical Trainer, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.*

*We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.*

*Kaustubh Sisodia*

*Ritik Lamba*

*Jayesh Gupta*

*Utkarsh Rathore*

*Isha Parveen*

**ABSTRACT**

The purpose of gesture recognition in Computers has always been the minimization of the distance between the physical world and the digital world. The way humans interact among themselves could be implemented in communication with the digital world by interpreting gestures via mathematical algorithm. Numerous ways and algorithms have been proposed and implemented to achieve the goal of gesture recognition and its use in communicating with the digital world. Gestures can be tracked using hand movements, accelerometers and more. This paper deals with the design and implementation of a gesture controlled computer using Arduino Uno

**Content**

1. INTRODUCTION

1.1 General Introduction to the topic

1.2 Problem Introduction

1.3 Objective

1.4 Motivation

1.5 Proposed Work

2. Software Requirement Analysis

2.1 Methodology

2.2 Hardware Requirements

2.3 Software Requirements

2.4 Installation of Software and Packages

2.5 Design of the Project

3. IMPLEMENTATION DETAILS

3.1 Circuit diagram

4. SOFTWARE DESIGN

4.1 Flow Chart

4.2 Use Case Diagram

4.3 Data Flow Diagram

5. CONCLUSION

6. FUTURE SCOPE

7. APPLICATION

8. REFERENCE

9. APPENDICES

**1. INTRODUCTION**

**1.1 General Introduction:**

In this project, we are going to learn how to build gesture-controlled laptops or computers. 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, move left/right in a photo slideshow, Scroll up/down in a web page and many more. This is why I decided to control VLC Media Player as a hand gesture project.

The idea behind the project is quite easy by using 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.

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 .In this project, we have implemented a simple Arduino based hand gesture control where you can control few functions of your web browser like switching between tabs, scrolling up and down in web pages, shift between tasks (applications), play or pause a video and increase or decrease the volume (in VLC Player) with the help of hand gestures.

**1.2 Problem Introduction:**

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.

**1.3. Objective:**

The principle behind the Arduino based Hand Gesture Control of Computer is actually very simple. All you have to do is use two Ultrasonic Sensors with Arduino, place your hand in front of the Ultrasonic Sensor and calculate the distance between the hand and the sensor. Using this information, relevant actions in the computer can be performed.

The position of the Ultrasonic Sensors is very important. Place the two Ultrasonic Sensors on the top of a laptop screen at either end. The distance information from Arduino is collected by a Python Program and a special library called PyAutoGUI will convert the data into keyboard click actions. Our objective is to make this device simple as well as cheap so it can be produced and used for number of purposes. The objective of this project is to build a iot device that can be controlled by gesture wirelessly. In this project user is also able to control motions of the hand by wearing controller glove and performing predefined gestures.

**1.4 Motivation**

The principle behind the Arduino based Hand Gesture Control of Computer is actually very simple. All you have to do is use two Ultrasonic Sensors with Arduino, place your hand in front of the Ultrasonic Sensor and calculate the distance between the hand and the sensor. Using this information, relevant actions in the computer can be performed.

The position of the Ultrasonic Sensors is very important. Place the two Ultrasonic Sensors on the top of a laptop screen at either end. The distance information from Arduino is collected by a Python Program and a special library called PyAutoGUI will convert the data into keyboard click actions.

**1.5 Proposed Work**

Gesture controlling is based on specifying hand position from the ultrasonic sensor. For processing the raw data, a micro-controller is essential; for that we use Arduino UNO board. Via USB connection the microcontroller transfers the processed and calculated distance value which is provided by the sensor. The data which is send by the sensor is processed in the software in PC where all the calculations are performed and the data is matched with the predefined conditions (gesture resolution). In this model two ultrasonic sensors are used to detect hand position and are connected to the Arduino board. As we know ultrasonic sensor continuously emits sound and it gets reflected back from user’s hand. The distance between the sounds is send and detection of reflect back sound wave is calculated by the micro-controller.

**2. Software Requirement Analysis**

**2.1 Methodology:**

The design of the circuit is very simple, but the setup of the components is very important. The Trigger and Echo Pins of the first Ultrasonic Sensor (that is placed on the left of the screen) are connected to Pins 11 and 10 of the Arduino. For the second Ultrasonic Sensor, the Trigger and Echo Pins are connected to Pins 6 and 5 of the Arduino.

Now, coming to the placement of the Sensors, place both the Ultrasonic Sensors on top of the Laptop screen, one at the left end and the other at right. You can use double sided tape to hold the sensors onto the screen.Coming to Arduino, place it on the back of the laptop screen. Connect the wires from Arduino to Trigger and Echo Pins of the individual sensors. Now, we are ready for programming the Arduino. Once all work is done then we are ready to demonstrate our project of hand gestuire as our objective is to make this device simple as well as cheap so it can be produced and used for number of purposes.

**2.2 Hardware Requirements:**

* Arduino UNO x 1
* Ultrasonic Sensors x 2
* USB Cable (for Arduino)
* Few Connecting Wires
* A Laptop with internet connection

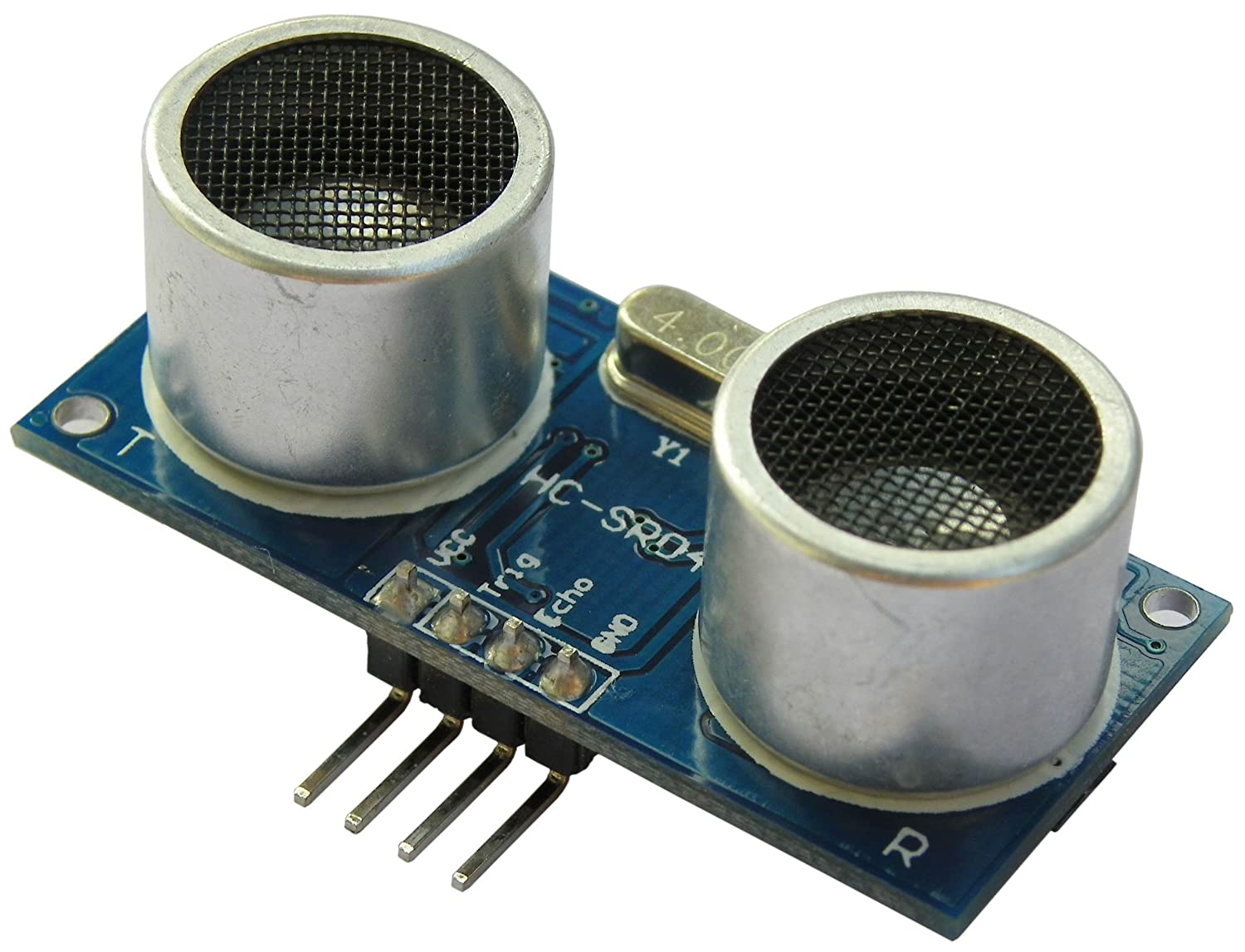
Arduino

**Arduino is** an open-source electronics platform based on easy-to-use hardware and software. **Arduino** boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online also **used for** building electronics projects. It consists of both a physical programmable circuit board and a software, or IDE (Integrated Development Environment) that runs on your computer, where you can write and upload the computer code to the physical board. The **Arduino Programming Language** is basically a framework built on top of C++. You can argue that it's not a real **programming language** in the traditional term, but I think this helps avoiding confusion for beginners. A program written in the **Arduino Programming Language** is called sketch.



Ultrasonic sensor

An **ultrasonic sensor** is an electronic device that measures the distance of a target object by emitting **ultrasonic** sound waves, and converts the reflected sound into an electrical signal. **Ultrasonic** waves travel faster than the speed of audible sound (i.e. the sound that humans can hear).**Human** Presence **Detection** with **Ultrasonic Sensors**. MaxBotix **ultrasonic sensors** solve the common problem of **sensing human** presence. Our **ultrasonic sensors** offer the ability to **detect** people over a wide range of distances with a high read rate and excellent reading to reading stability and **made** with piezoelectric crystals, use high frequency sound waves to resonate a desired frequency and convert electric energy into acoustic energy, and vice versa. **Ultrasonic** sensors work by sending out a sound wave at a frequency above the **range** of human hearing. ... The sensor determines the **distance** to a target by measuring time lapses between the sending and receiving of the **ultrasonic** pulse. The working principle of this module is simple.



**2.3 Software Requirements:**

* Arduino IDE
* Python IDE

Arduino IDE

The **Arduino Integrated Development Environment (**[**IDE**](https://en.wikipedia.org/wiki/Integrated_development_environment)**)** is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in functions from [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B_(programming_language)).It is used to write and upload programs to [Arduino](https://en.wikipedia.org/wiki/Arduino) compatible boards, but also, with the help of third-party cores, other vendor development boards. It is the main text editing program used for **Arduino** programming. ... Essentially, the **IDE** translates and compiles your sketches into code that **Arduino** can understand. Once your **Arduino** code is compiled it's then uploaded to the board's memory.

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on **Windows**, **Mac OS X**, and **Linux**. The environment is written in Java and based on Processing and other open-source software. **So** the **Arduino** hardware is easy, the **Arduino** IDE is simple, and the code itself is much easier to comprehend (than trying to program an off-the-shelf microcontroller). ... Another reason **Arduino** is **so popular** is because there are many people using it which means there's a lot of examples out there to work with.

Python IDE

**IDLE** (short for **Integrated Development and Learning Environment**) is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) for [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), which has been bundled with the default implementation of the language since 1.5.2b1.It is packaged as an optional part of the Python packaging with many [Linux distributions](https://en.wikipedia.org/wiki/Linux_distributions). It is completely written in Python and the [Tkinter](https://en.wikipedia.org/wiki/Tkinter" \o "Tkinter) GUI toolkit ([wrapper](https://en.wikipedia.org/wiki/Wrapper_function) functions for [Tcl](https://en.wikipedia.org/wiki/Tcl" \o "Tcl)/[Tk](https://en.wikipedia.org/wiki/Tk_(framework)" \o "Tk (framework))).We should use this IDE as because another good point to consider is your use of programming languages in general; if you expect **Python** to be your main language,

with little use of others, or you expect to use **Python** as your only non-**IDE**-bound language, an **IDE** is probably a good idea; you'll get good at using it and it will be an effective tool for programming.

Python

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language" \o "Interpreted language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) and [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.

Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library). Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a [semicolon](https://simple.wikipedia.org/wiki/Semicolon) at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

Python also does something called "dynamic variable assignment". This means that when a number or word is made in a program, the user does not have to say what type it is. This makes it easier to reuse variable names, making fast changes simpler. An example of this is shown below. This code will make both a number and a word, and show them both, using only one variable.

[Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support [functional programming](https://en.wikipedia.org/wiki/Functional_programming) and [aspect-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including by [meta programming](https://en.wikipedia.org/wiki/Metaprogramming) and [meta objects](https://en.wikipedia.org/wiki/Metaobject" \o "Metaobject) (magic methods)). Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [logic programming](https://en.wikipedia.org/wiki/Logic_programming).

Python uses [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing) and a combination of [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and a cycle-detecting garbage collector for [memory management](https://en.wikipedia.org/wiki/Memory_management).[[57]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-Reference_counting-57) It also features dynamic [name resolution](https://en.wikipedia.org/wiki/Name_resolution_(programming_languages)) ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution.

Python's design offers some support for [functional programming](https://en.wikipedia.org/wiki/Functional_programming) in the [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)) tradition. It has filter, map, and reduce functions; [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), [dictionaries](https://en.wikipedia.org/wiki/Associative_array), sets, and [generator](https://en.wikipedia.org/wiki/Generator_(computer_programming)) expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)) and [Standard ML](https://en.wikipedia.org/wiki/Standard_ML).

**2.4 INSTALLATION OF SOFTWARES AND PACKAGES**

A. Installation of Arduino IDE:- Arduino IDE is open-source software developed in order to program circuit boards easily and efficiently. We can perform simple operations like turning on LED to complex operations like controlling robots. We can tell our board to do whatever operation we want by our Arduino board using a set of programming instructions. So to perform these operations we use a software called Arduino IDE. Using this software we can perform coding for various projects like Iot, wearable devices, 3D printing, circuit boards etc. So in order to install the software we can go to the below mentioned link and select our operating system and download the software and install it. https://www.arduino.cc/en/Main/Software

B. Installation of Python IDE We have to install latest version of python i.e 3.83 (at the time paper was written). It can be installed from the below link based on your operating system: https://www.python.org/downloads/

**2.5 Design of the Project**

The design of the circuit is very simple, but the setup of the components is very important. The Trigger and Echo Pins of the first Ultrasonic Sensor (that is placed on the left of the screen) are connected to Pins 11 and 10 of the Arduino. For the second Ultrasonic Sensor, the Trigger and Echo Pins are connected to Pins 6 and 5 of the Arduino.

Now, coming to the placement of the Sensors, place both the Ultrasonic Sensors on top of the Laptop screen, one at the left end and the other at right. You can use double sided tape to hold the sensors onto the screen.



**3. IMPLEMENTATION DETAILS**

The important part of this project is to write a program for Arduino such that it converts the distances measured by both the sensors into the appropriate commands for controlling certain actions.

A similar concept is used here to measure the distance of your hand in front of both the Ultrasonic Sensors in this project. The fun part starts after calculating the distance.The hand gestures in front of the Ultrasonic sensors can be calibrated so that they can perform five different tasks on your computer. Before taking a look at the gestures, let us first see the tasks that we can accomplish.

* Switch to Next Tab in a Web Browser
* Switch to Next Tab in a Web Browser
* Scroll Down in a Web Page
* Scroll Up in a Web Page
* Switch between two Tasks (Chrome and VLC Player)
* Play/Pause Video in VLC Player
* Increase Volume
* Decrease Volume

The following are the 5 different hand gestures or actions that I’ve programmed for demonstration purpose.

**Gesture 1:** Place your hand in front of the Right Ultrasonic Sensor at a distance (between 15CM to 35CM) for a small duration and move your hand away from the sensor. This gesture will Scroll Down the Web Page or Decrease the Volume.

**Gesture 2:** Place your hand in front of the Right Ultrasonic Sensor at a distance (between 15CM to 35CM) for a small duration and move your hand towards the sensor. This gesture will Scroll up the Web Page or Increase the Volume.

**Gesture 3:** Swipe your hand in front of the Right Ultrasonic Sensor. This gesture will move to the Next Tab.

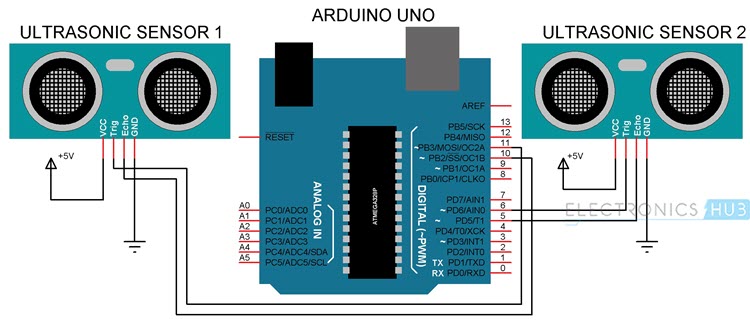
**Gesture 4:** Swipe your hand in front of the Left Ultrasonic Sensor. This gesture will move to the Previous Tab or Play/Pause the Video.

**Gesture 5:** Swipe your hand across both the sensors (Left Sensor first). This action will Switch between Tasks.

**3.1 Circuit Diagram**

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 you can power up all these components from the laptop’s USB Port.

For left sensor we connect VCC to 5V power pin in Arduino-Uno board, then we connect ground pin to first GND pin. Then we connect trigger pin to pin 2 and finally echo pin to pin 3 in Arduino board. For the right sensor, we connect VCC pin to 3.3V power pin, then we connect ground pin to second GND pin. Finally we connect trigger pin to pin 4and echo pin to pin 5 in Arduino board.



**4. SOFTWARE DESIGN**

This is the designing portion of the project which defines software solutions to one or more sets of problem. One of the main component of software design is the software requirement analysis.

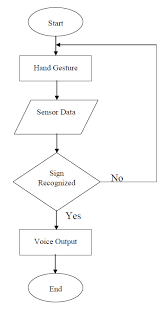
**4.1 FLOW CHART**

A **flowchart** is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan.

After That all Parts are control with Human

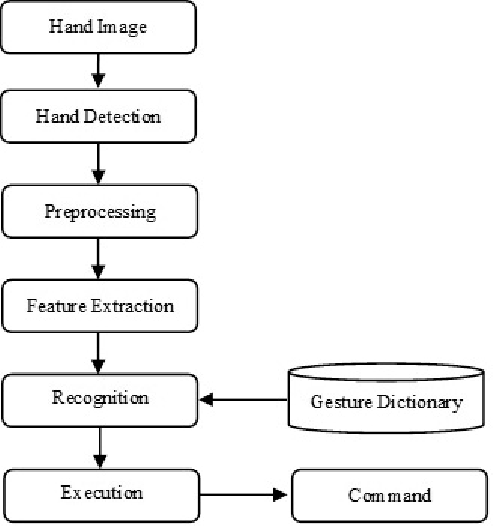
**4.2 USE CASE DIAGRAM**

A **use case** is a list of actions or event steps typically defining the interactions between a role (known in the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) as an [actor](https://en.wikipedia.org/wiki/Actor_(UML))) and a system to achieve a goal. The actor can be a human or other external system. [Use case analysis](https://en.wikipedia.org/wiki/Use-case_analysis) is an important and valuable [requirement analysis](https://en.wikipedia.org/wiki/Requirement_analysis) technique that has been widely used in modern software engineering.

****

**4.3 DATA FLOW DIAGRAM**

**Data flow diagrams** are used to graphically represent the **flow** of **data** in a business information system. DFD describes the processes that are involved in a system to transfer **data** from the input to the file storage and reports generation. **Data flow diagrams** can be divided into logical and physical.



**5. CONCLUSION**

This article presents one of the solutions among various others, for operating a computer using hand gestures. It is one of the easiest ways of interaction between human and computer. It is a cost effective model which is only based on Arduino UNO and ultrasonic sensor. The python IDE allows a seamless integration with Arduino UNO in order to achieve different processing and controlling methods for creating new gesture control solutions.

**6. FUTURE SCOPE**

Hand gesture technique is not only limited to gaming, using basic function of computer it can be useful for medical applications. Hand gesture technique can work as input method between medical instruments and human body as proposed. It can be used for operating each and every functions of computer.

**7. APPLICATION**

* In this project, we 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.
* This type of hand gesture control of computers can be used for VR (Virtual Reality), AR (Augmented Reality), 3D Design, Reading Sign Language, etc.

**8. REFERENCES**

1. Software Engineering by McGraw Hill Publication

2. www.youtube.com/articles.tvac.in.faqs

3. www.udemy.com

4. https://stackoverflow.com/questions/38217414

5. Amazing-control-computer-using-hand-motion-and-arduino-d933f1

**9. APPENDICES**

**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 loopy

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 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 ("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);

}

**PYTHON CODE**

const int trigger1 = 2; //Trigger pin of 1st Sesnor

const int echo1 = 3; //Echo pin of 1st Sesnor

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>60)

dist = 60;

}

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

//Pause Modes -Hold

if ((distL >40 && distR>40) && (distL <60 && distR<60)) //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 ("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);

}

import serial #Serial imported for Serial communication

import time #Required to use delay functions

import pyautogui #Required to to perform actions

ArduinoSerial = serial.Serial('com15',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 Incresaed' in incoming:

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

if 'Volume Decreased' in incoming:

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

incoming = "";