Hand Gesture Controlled Computer

Nirja Joshi, *Student, University of Regina*

***Abstract*—The project is focused on implementing the technique to control computer using hand gestures. The concept behind Arduino-based Gesture maintained computer is simple and is implemented using Python. The principle here uses 2 Ultrasonic sensors with Arduino and the central part is the position of the ultrasonic sensors. It works by placing our hands in front of the ultrasonic sensor and then calculating the distance between the hand and the sensor. The computer uses this information to perform relevant actions. The positioning of ultrasonic sensors is the most crucial part here. The project proposal focuses on making modifications to the existing project, implementing multiple gestures to control computer actions.**

***Index Terms*—project report, computer science, CS 807, Arduino, Hand gestures**

I. Introduction

IN this tech-advanced world Gesture controlled Laptops/-Computers are getting popular. The use of gesture-based interaction systems is becoming increasingly popular at work and at home. This project aims to create a system for recognizing hand gestures that can be used as input commands for interacting with a laptop or PC. The basic project base is implementing different gestures, specifically volume up and down, brightness low and high, and that is handled using hand gestures.

The implementation stage of such systems is one of the most important aspects to consider. For the implementation of the code, we will be using Python. Our goal is to contribute at least a minimal amount to the future of natural gesture-based interfaces if we succeed in meeting our goal. It looks very cool because it enables us to control certain functions on our computer/Laptop by simply waving our hand in front of it. The logic of the program is very simple, we just need to place two Ultrasonic sensors on top of a computer or laptop which will read the distance between the monitor and our hand using Arduino, based on this value of distance we will perform certain actions. I have implemented this using the distance reading concept, which means the volume on the computer will turn up if there is a specific distance between the hand and the sensors and the same implies to all the gestures.

A. Action 1

When a hand is at a specific distance from the sensors, the volume turns up. The Arduino uno will check if the distance is between 2cm and 10cm if yes then it will print "VOLUME-UP" in the serial.

B. Action 2

When a hand is at a specific distance from the sensors, the volume turns down. The Arduino uno will check if the distance is between 12cm and 20cm if yes then it will print "VOLUME-DOWN" in the serial.

C. Action 3

When a hand is at a specific distance from the sensors, the brightness turns high. The Arduino uno will check if the distance is between 22cm and 30cm if yes then it will print "BRIGHTNESS-UP" in the serial.

D. Action 4

When a hand is at a specific distance from the sensors, the brightness turns low. The Arduino uno will check if the distance is between 32cm and 50cm if yes then it will print "BRIGHTNESS-DOWN" in the serial.

To perform actions on our computer we use the Python pyAutoGUI library. The principle here uses 2 Ultrasonic sensors with sensors. It works by placing our hands in front of the ultrasonic sensor and then calculating the distance between the hand and the sensor. The commands from Arduino are sent to the computer through the serial port. 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 main purpose of this program is to recreate the Gesture controlled Laptops/Computers and increase their functionality by adding voice with hand gestures. Gesture-based computer control based on Arduino is based on a very simple principle. The distance between our hand and the Ultrasonic Sensor can simply be calculated using two Ultrasonic Sensors and Arduino. The computer can perform certain actions based on this information. Ultrasonic Sensors play an important role in the design process. It is necessary to place two Ultrasonic Sensors at either end of a laptop screen. PyAutoGUI, a library that converts keyboard click actions into distance information from Arduino, collects distance information from Arduino with a Python program.

II. Related Work

Sensors are used primarily in 13 types around the world. To accomplish different types of work, these 13 types of sensors can be used in different ways. Different sensors have different shapes, sizes, and costs. Sensors deployed on real-time devices, such as vehicles, crops, medical equipment, and weather forecasting, can be used to monitor any real-time device. An input device is also known as a sensor. Processors receive input from these input devices. Processors can control these sensors in different ways. Using Arduino, the project aims to create a system that can be used to monitor and trigger actions on operating systems via a user's hand gesture as input. The ultrasound sensor is being used in a similar manner in this project. By calculating hand gestures, the sensor is able to determine distance changes as well. A trigger will be activated

over the operating system using this data. Several applications have inputs for performing tasks such as scrolling up, next, pausing, rewinding, scrolling down, etc.

The leap motion sensor was also developed by Leigh Ellen Potter and others. A view of sign language with the Leap Motion controller. This device is designed primarily to understand Australian signs. The system is very expensive and has a low level of accuracy.

Meenakshi Panwar developed a method for recognizing hand gestures based on shape parameters [8]. Gestures are detected using infrared sensors. There was a lot of slowness and lack of accuracy in the recognition.

Rishabh Agrawal and Nikita Gupta propose the IEEE paper titled "Real-Time Hand Gesture Recognition for Human-Computer Interaction"[14]. Currently, a majority of human-computer interaction interfaces that are designed are designed to require explicit user inputs in the form of keyboard taps or mouse clicks from user. By using computer vision and image processing techniques, this paper proposes a novel method of recognizing hand gestures for human-computer interaction. There are however some disadvantages to this type of input that do not exist for traditional inputs as well.

Minh Q. Nguyen and Changzhi Li [15] propose a radar and ultrasound hybrid system for human-computer interaction in their IEEE paper. Among the emerging technologies for human-computer interaction is touchless hand gestures. An FMCW radar and ultrasound sensor hybrid system are proposed in this paper in order to detect hand movement for controlling a computer using frequency modulated continuous wave (FMCW) radar and ultrasound sensors. In addition, this type of input raises issues that do not apply to traditional inputs. Users have problems learning, remembering, and executing gestures accurately.

An IEEE paper by Gergely Sziladi and Tibor Ujbanyi [16] examines the control of cursor position during an IT-related task based on hand gestures. In this article, a gesture control system is designed and implemented that determines gestures based on hand movements. Using distance measurement sensors, the control interface detects the hand's movement and then assigns the gesture based on it. In addition to ensuring that gestures are recognized quickly and correctly, the developer must also create a guide to help users learn these gestures quickly and easily.

An IEEE paper titled "CONTROLLING COMPUTER APPLICATIONS WITH HAND GESTURES USING ARDUINO" is presented by Andrea Attwenger [7]. Gesture-based interaction has become increasingly common in our daily lives, but its potential is not fully realized. In the real world, gestures mimic the way users interact. In addition to feeling natural, they do not require any interruptions or additional devices. Another aspect of gestures is their design, which should make them both memorable and comfortable to use. Gestures also raise issues that do not apply to traditional methods of input.

IEEE paper by Kaoru Yamagishi, Lie Jing, and Zixue Cheng proposes a wireless sensor device for controlling personal computers with hand gestures [17]. We live in a world filled with household appliances and computers. By using Arduino-based Hand Gesture Control of Computer Application as an interface, we developed a system that controls a personal computer by using the natural behavior of humans. This research is focusing on the definition of associations between PC operations and gestures, the recognition of hand gestures, and how to adjust the gesture errors.

III. Proposed Methodology

The concept behind Arduino-based Gesture maintained computer is simple and is implemented using Python. The commands from Arduino are sent to the computer through the serial port. 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 principle here uses 2 Ultrasonic sensors with Arduino and the central part is the position of the ultrasonic sensors. For instance if raising a right hand raises volume, the audio will say "Volume up". Likewise, the main purpose is to add audio and make the user more compatible with the computer/laptop.

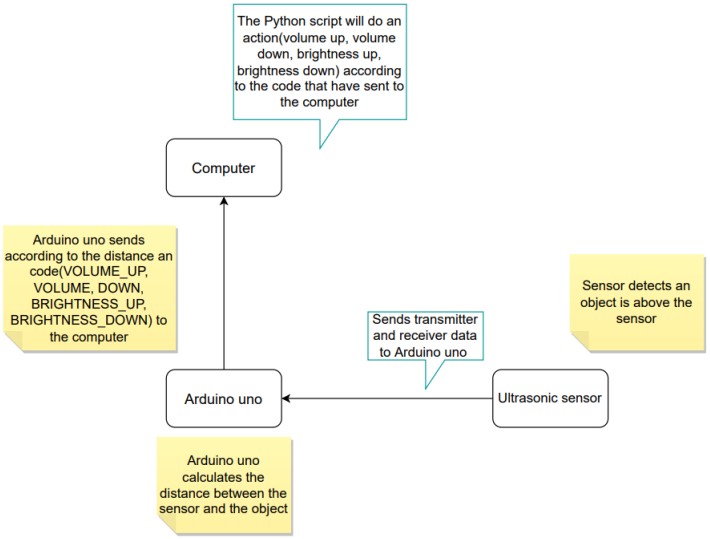

Fig. 1. Block diagram of the circuit

IV. Detailed Description

A. Arduino Uno

The Arduino Uno is a microcontroller based on AT mega 328, which is having 14 digital input/output pins, six analog input pins, a USB connection facility, a power jack, and a reset button facility

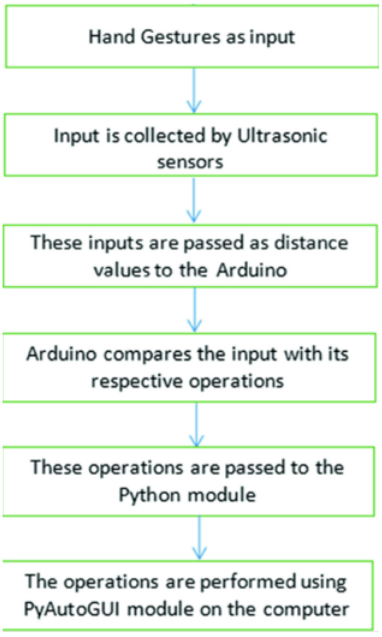

Fig. 2. [Process map

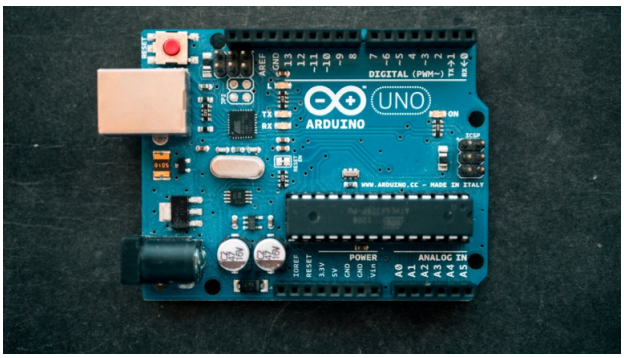

Fig. 3. Block diagram of the circuit

B. Ultrasonic sensor

The HC-SR04 Ultrasonic Distance Sensor is a sensor used for detecting the distance to an object using sonar. It's ideal for any robotics projects your have which require you to avoid objects, by detecting how close they are you can steer away from them.

C. PyAutoGUI library

PyAutoGUI is used to control the hardware like a mouse, keyboard, etc., using programs. GUI automation for human beings can be achieved through a cross platform Python module provided by PyAutoGUI. It is with sensible defaults and is possibly designed to be very simple. It can simulate various functions of a keyboard as well as a mouse. Pressing hot keys combination is one among them.

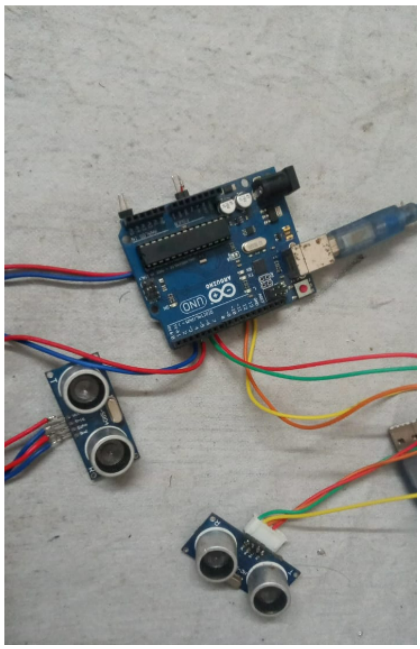

Fig. 4. sensors

D. Serial library

The Serial library reads and writes data to and from external devices one byte at a time. It allows two computers to send and receive data. This library has the flexibility to communicate with custom micro controller devices and to use them as the input or output to processing program.

E. Screen-Brightness-Control library

A Python tool for controlling the brightness of your monitor. Supports Windows and most flavours of Linux.

V. Device Information Circuit design

A. Device parts

The device used:

- 1 bread board,
- 1 ultrasonic sensor,
- 1 Arduino uno,
- 4 male to male jumper wires,
- 1 USB Cable A male to B male.

B. Circuit design

I have connected the VCC (Common Collector Voltage) in the sensor to the 5v pin in the Arduino Uno, connected the trig pin in the sensor to the 5 digital pins in the Arduino Uno, and connected the echo pin in the sensor to the 6 digital pins in the Arduino Uno and connected the GND(Ground) pin in the sensor to the GND(Ground) pin in the Arduino Uno.

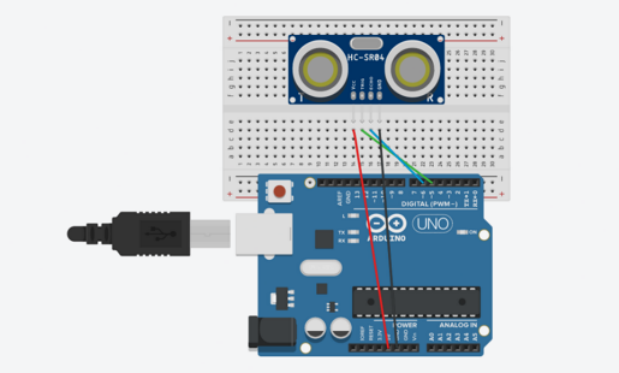

Fig. 5. Circuit design from Tinkercad

C. Detailed Description

Firstly, the device takes power by connecting 5v pin in Arduino uno to the VCC pin in the ultrasonic sensor and connecting GND pin in Arduino uno to the GND pin in the ultrasonic sensor.

The trig pin data will be sent to the (5) digital pin in the Arduino uno, and the echo pin data will be sent to the (6) digital pin in the Arduino uno. The transmitter in the sensor will send ultrasonic waves and the waves will stop at an object and come back to the receiver, so then the Arduino uno will calculate the time between sending the waves and receiving the waves, and then we will use this equation to find the distance between the sensor and the object (duration * 0.034 / 2). The

Arduino uno will check if the distance is between 2cm and 10cm if yes then it will print “VOLUME-UP” in the serial, if no then it will check if the distance is between 12cm and 20cm if yes then it will print “VOLUME-DOWN” in the serial, if no then it will check if the distance is between 22cm and 30cm if yes then it will print “BRIGHTNESS-UP” in the serial, if no then it will check if the distance is between 32cm and 50cm if yes then it will print “BRIGHTNESS-DOWN” in the serial.

The Python script will get the serial of the device by the port number (e.g. com1, com2, com3, come4… etc.), and it will check if it prints “VOLUME-UP” then it will volume up the computer audio, if no it will check if it prints “VOLUME-DOWN” if yes it will volume down the computer audio, if no it will check if it prints “BRIGHTNESS-UP” if yes It will increase computer brightness, if no then it will check if it prints “BRIGHTNESS-DOWN” if yes It will decrease computer brightness.

My sensor is connected to the Arduino uno via the 5v pin, and the trig pin has been connected to the 5 digital pins of the Arduino uno. I have also connected the VCC pin of the sensor to the Arduino uno’s VCC pin. In addition, the sensor’s echo pin is connected to the Arduino uno’s six digital pins, and the sensor’s GND (Ground) pin is connected to the Arduino uno’s GND pin.

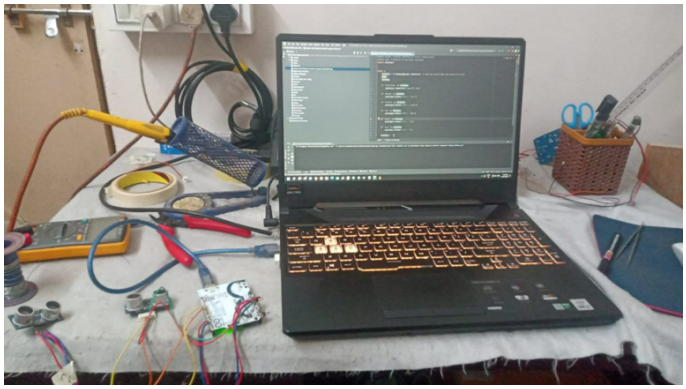

Fig. 6. Final Setup

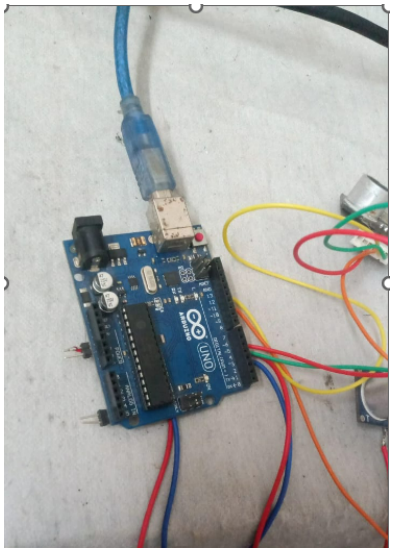

Fig. 7. Final Setup

VI. Results and Conclusion

I have successfully implemented hand gestures controlled computer system using the distance reading concept, which means the volume on the computer will turn up if there is a specific distance between the hand and the sensors and the same implies to all the gestures.

A. Action 1

When a hand is at a specific distance from the sensors, the volume turns up. The Arduino uno will check if the distance is between 2cm and 10cm if yes then it will print “VOLUME-UP” in the serial.

B. Action 2

When a hand is at a specific distance from the sensors, the volume turns down. The Arduino uno will check if the distance is between 12cm and 20cm if yes then it will print “VOLUME-DOWN” in the serial.

C. Action 3

When a hand is at a specific distance from the sensors, the brightness turns high. The Arduino uno will check if the

distance is between 22cm and 30cm if yes then it will print "BRIGHTNESS-UP" in the serial.

D. Action 4

When a hand is at a specific distance from the sensors, the brightness turns low. The Arduino uno will check if the distance is between 32cm and 50cm if yes then it will print "BRIGHTNESS-DOWN" in the serial.

VII. Milestone Discussion

Below are the given milestones that would be met once when the project is completed.

- Milestone 1: Design of the circuit and components setup:

Firstly, understanding the ultrasonic sensors and the whole concept took some time for me. My next priority was to design the circuit as that was the most difficult part for me, therefore I connected Pins 11 and 10 of the Arduino (Trigger and Echo Pins) and Ultrasonic sensor connected to Pins 6 and 5 of the Arduino. Estimated completion date was July 03 but his milestone was successfully achieved to start designing the circuit. It took more time than I thought and I initially designed it on Tinkercad and then designed a circuit with sensors. The estimated completion date was July 03, but I completed it on July 24.

- Milestone 2: Placement of the Sensors and Arduino:

The second milestone was to complete the design of the circuit - both ultrasonic sensors are mounted on the top of the screen and Arduino on the back side. I did start working on writing a report with the circuit design. Estimated completion date was July 08 but this milestone was achieved on July 28 after I designed the respective circuit.

- Milestone 3: Programming Arduino:

To calculate distance such that it converts the distances measured by both the sensors into the appropriate commands for controlling certain actions. I added modifications to the project as discussed above. Estimated completion date: July 28. This was the major challenge as I was not able to implement the technique or methodology to calculate the distance such that it converts the distances measured by both the sensors into the appropriate commands for controlling certain actions. This was implemented on August 02 after multiple failed attempts.

- Milestone 4: Final working project

There were multiple challenges with the code and my design had some issues, the breadboard was working perfectly but it was not controlling the computer for some reason. I had to redesign the whole thing as I was not able to find the problem. This took very long, and this milestone was partially achieved as I was not only able to implement 4 gestures: volume up and down, brightness up and down.

- Milestone 5: Preparing Project Report:

Project report with full working project with all the modifications of the gestures and if coding for audio works and adding it to the Github repository. The estimated completion date: August 13 but the final completion date got extended to August 14. I was preparing part of the report every day so the report was successfully completed on August 14.

VIII. Team Roles

This project is done only by me - Nirja Joshi and no teammates were involved in this project. To ensure the smooth functioning of the prototype, all project selection, research, hardware design, prototype testing, writing the paper, and GitHub documentation were entirely done by me.

IX. Conclusion

The main motivation for choosing this program was, the increase in usage of Gesture controlled Laptops/Computers, and it was relevant to an area of interest to work. The future laptops will be based on the technique called leap motion, keeping all the information in my I have reached the conclusion of building Gesture controlled Laptops/Computers with audio. Human-computer interaction is made easier by this method. An ultrasonic sensor and Arduino UNO are the only components required to create this cost-effective model. In order to create new gesture control solutions, the Python IDE can be seamlessly integrated with Arduino UNO to achieve different processing and controlling methods. It is also possible to recognize gestures in medical applications where nurses and doctors cannot touch a screen or track pad for health and safety reasons. Medical professionals may also need to manipulate the content shown on displays even if they are not within reach of the display. The best way to control the device is with appropriate gestures, such as hand swipes or the use of a finger as a virtual mouse.

X. Source Code

```
1
2  const int trigger1 = 12; //Trigger pin of 1st Sesnor
3  const int echo1 = 13; //Echo pin of 1st Sesnor
4  const int trigger2 = 3; //Trigger pin of 2nd Sesnor
5  const int echo2 = 4;//Echo pin of 2nd Sesnor
6
7  long time_taken;
8  int dist, distL, distR;
9
10
11
12
13
14 long duration;
15 float r;
16 unsigned long temp = 0;
17 int temp1 = 0;
18 int l = 0;
19 void distance_measurement (void);
20
```

```
//
    ////////////////////////////////////////////////
    _distance_measurement
    ----------////////////////////////////////////////


void distance_measurement (void)
{
  digitalWrite(trigger1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger1, LOW);
  duration = pulseIn(echo1, HIGH, 5000);
  r = 3.4 * duration / 2;
  distL = r / 100.00;
  digitalWrite(trigger2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger2, LOW);

  duration = pulseIn(echo2, HIGH, 5000);
  r = 3.4 * duration / 2;
  distR = r / 100.00;
  delay(100);
}



//
    ////////////////////////////////////////////////
    Hardware_setup_file
    --------------////////////////////////////////////


void setup() {
  Serial.begin(9600);

  pinMode(trigger1, OUTPUT);
  pinMode(echo1, INPUT);
  pinMode(trigger2, OUTPUT);
  pinMode(echo2, INPUT);
}

//
    ////////////////////////////////////////////////
    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;
}



//
    ////////////////////////////////////////////////
    main_file
    -------------------////////////////////////////////


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 > 25 && distR > 25) && (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 >= 10 && distL <= 20)
  {
    delay(50); //Hand Hold Time
    calculate_distance(trigger1, echo1);
    distL = dist;
    if (distL >= 10 && distL <= 20)
    {
      Serial.println("Left Locked");
      while (distL <= 40)
      {
        calculate_distance(trigger1, echo1);
        distL = dist;
        if (distL < 15) //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 >= 10 && distR <= 20)
```

```
  {
    delay(50); //Hand Hold Time
    calculate_distance(trigger2, echo2);
    distR = dist;
    if (distR >= 10 && distR <= 20)
    {
      Serial.println("Right Locked");
      while (distR <= 40)
      {
        calculate_distance(trigger2, echo2);
        distR = dist;
        if (distR < 15 ) //Right hand pushed in
        {
          Serial.println ("Rewind");
          delay (300);
        }
        if (distR > 20) //Right hand pulled out
        {
          Serial.println ("Forward");
          delay (300);
        }
      }
    }
  }



  //
    ////////////////////////////////----------------------------
    Swipe Next - Control Mode
    -------------------------////////////////////////////////

  if (distR <= 8 && distR >= 0)
  {
     temp = millis();
     while (millis() <= (temp + 300))
      distance_measurement();

    {

      Serial.println("next");

    }

  }


  delay(200);
}
```

REFERENCES

[1] R. Mukherjee, P. Swethen, R. Pasha and S. Singh Rawat“ Hand Gesture Controlled Laptop Using Arduino”,International Journal of Management, Technology And Engineering, Vol. 8, pp. 1037-1043, Oct. 2018.

[2] Hafız Mohsin Abdul-Rashid,Lehmia Kiran, M. Danish Mirrani, M. Noman Maraaj,“CMSWVHG control MS Windows via hand gesture”, in: proc. of the IEEE Int. Multi-topic Conference (INMIC), Nov. 2017.

[3] Gergely Sziládi, Tibor Ujbányi and JózsefKatona, “Cost effective hand gesture computer control interface”,in: proc. of the IEEEInt. Conference on Cognitive Info communications, Oct. 2016.

[4] A. K. Shrivastava, A. Verma and S. P. Singh,“Distance Measurement of an Object or Obstacle by Ultrasound Sensors”, Int. Journal of Computer Theory and Engineering, Vol. 2, No. 1, pp. 64-68, Feb. 2010.

[5] Sawardekar, Gaurav and Thaker, Parthil and Singh, Rishiraj and Gaikwad (Mohite), Vaishali, Arduino Based Hand Gesture Control of Computer Application (October 29, 2018). Available at SSRN: https://ssrn.com/abstract=3274460 or http://dx.doi.org/10.2139/ssrn.3274460

[6] U. Rajkanna, M. Mathankumar and K. Gunasekaran published in IEEE “Hand gesture based computer control using microcontroller” on 7-8 jan 2012,Rohtak, Harayana ,India.

[7] Andrea Attwenger published in IEEE “CONTROLLING COMPUTER APPLICATION USING HAND GESTURES USING ARDUINO” (2012)

[8] Meenakshi panwar and pawan singh mehra published in ieee.(2014) “hand gesture recognition for human computer interaction”

[9] Aziz, Muhammad Zulkifal, Daniyal Adeeb Khan, and Asma Mushtaq. ”A review: Study of various techniques of Hand Gestures Recognition.”

[10] U. Rajkanna, M. Mathankumar and K. Gunasekaran published in IEEE “Hand gesture based computer control using microcontroller” on 7-8 jan 2012,Rohtak, Harayana ,India.

[11] Gowri, S., and G. Divya. ”Automation of garden tools monitored using mobile application.” International Confernce on Innovation Information in Computing Technologies. IEEE, 2015.

[12] 12 https://youtu.be/05jDQV4$_u XU$

[13] 13 https://youtu.be/xlmJDOvI5fA

[14] Agrawal, Rishabh, and Nikita Gupta. ”Real time hand gesture recognition for human computer interaction.” 2016 IEEE 6th International Conference on Advanced Computing (IACC). IEEE, 2016.

[15] Nguyen, Minh Q., and Changzhi Li. ”Radar and ultrasound hybrid system for human computer interaction.” 2018 IEEE Radar Conference (RadarConf18). IEEE, 2018.

[16] Sziladi, Gergely, et al. ”The analysis of hand gesture based cursor position control during solve an IT related task.” 2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom). IEEE, 2017.

[17] Sawardekar, Gaurav, Parthil Thaker, and Rishiraj Singh. ”Arduino Based Hand Gesture Control of Computer Application.” Vaishali, Arduino Based Hand Gesture Control of Computer Application (October 29, 2018) (2018).