PROJECT COORDINATOR **:** NAVJOT RATHOUR

COURSE CODE **:** ECE878

COURSE NAME **:** PROGRAMMING WITH ARDUINO

SECTION **:** DOE02

SUBMITTED BY:

NAME	REGISTRATION ID	ROLL NUMBER
PADMANAV PHUKAN	11615896	53
SAFIUR RAHAMAN	11604538	50
SAGNIK PATRA	11716191	51
RAVI YADAV	11717020	52

Hand Gesture Controlled Laptop Using Arduino Uno And Ultrasonic sensor
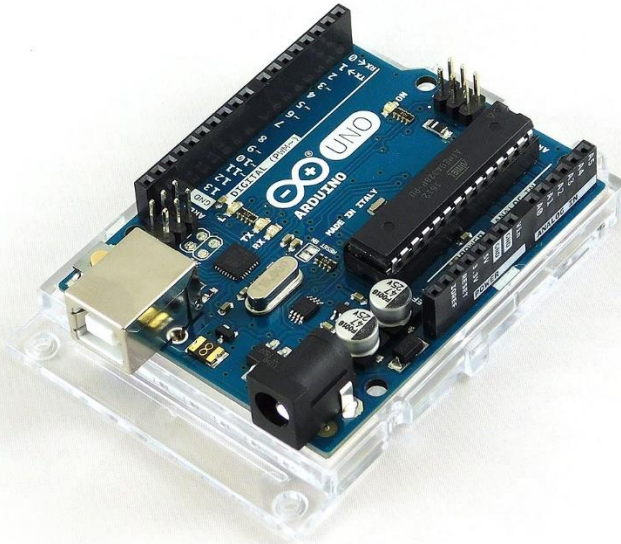
ACKNOWLEDGEMENT

This project consumed huge amount of work, research and dedication. Still, implementation would not have been possible if we did not have received support from the team members and our respected project guide, Mrs Navjot Rathour.

We would like to express our special thanks of gratitude to Mrs Navjot Rathour, who gave us the golden opportunity to do this wonderful project, which also helped us in doing a lot of Research and we came to know about a lot of new things. We are really thankful to her.

Secondly we would also like to appreciate the effort of the team members (Padmanav Phukan, Safiur Rahaman, Sagnik Patra, Ravi Yadav) by the support of which, the project was successfully completed within the limited time frame.

ABSTRACT

Since the time that the computer revolution started, human computer interaction has always been attempted to improve. Computers have now become an integral part of our lives and hence their usage should be as trouble-free as talking to someone is. Earlier the way humans interacted with this smart machine was either through keyboard or a mouse. But now attempts are being made to make the man-machine interaction as natural as possible. Fulfilling this requirement is the popular touch screen technology which is soon expected to be replaced by the **gesture recognition technology**

More and more electronic devices are being made with gesture recognition features. As the name suggests, gesture recognition is designed to recognize the physical movements, or "gestures," of humans. Waving your hand in a particular pattern in front of the device, for instance, may tell it to launch a specific app.

This paper deals with the design and implementation of a gesture controlled computer using Arduino Uno with ATMEGA328 processor and a laptop loaded with Windows 10 operating system along with low cost hardware requirements. The system can be broadly classified into two components: The Hardware part consisting of Arduino Uno Microcontroller, the ultrasonic sensors HC-SR04 and a laptop, and the software part consists of Arduino IDE and Python 3.7 IDLE with PyAutoGUI module installed.

CONTENTS

INTRODUCTION

Today's world is a high technological world; imagining life without computer is impossible. Now computer is an essential part of human's day-to-day life. Gesture is an important approach for communication between human beings just in the same way as keyboard and mouse play a role for interacting with computer. It involves the use of natural hand gestures to control devices. A gesture recognition system comprehends human gestures with mathematical algorithms. With gesture recognition, computers can become familiarized with the way humans communicate using gestures. Thus, machines and humans can interact freely with each other. The primary goal of gesture recognition is to create a system which understands human gestures uses them to control various devices.

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 virtual keystrokes for Windows. The hotkeys then control particular function of the application of intend to run, that is VLC Media Player.

HARDWARE

REQUIREMENTS:

ARDUINO UNO:
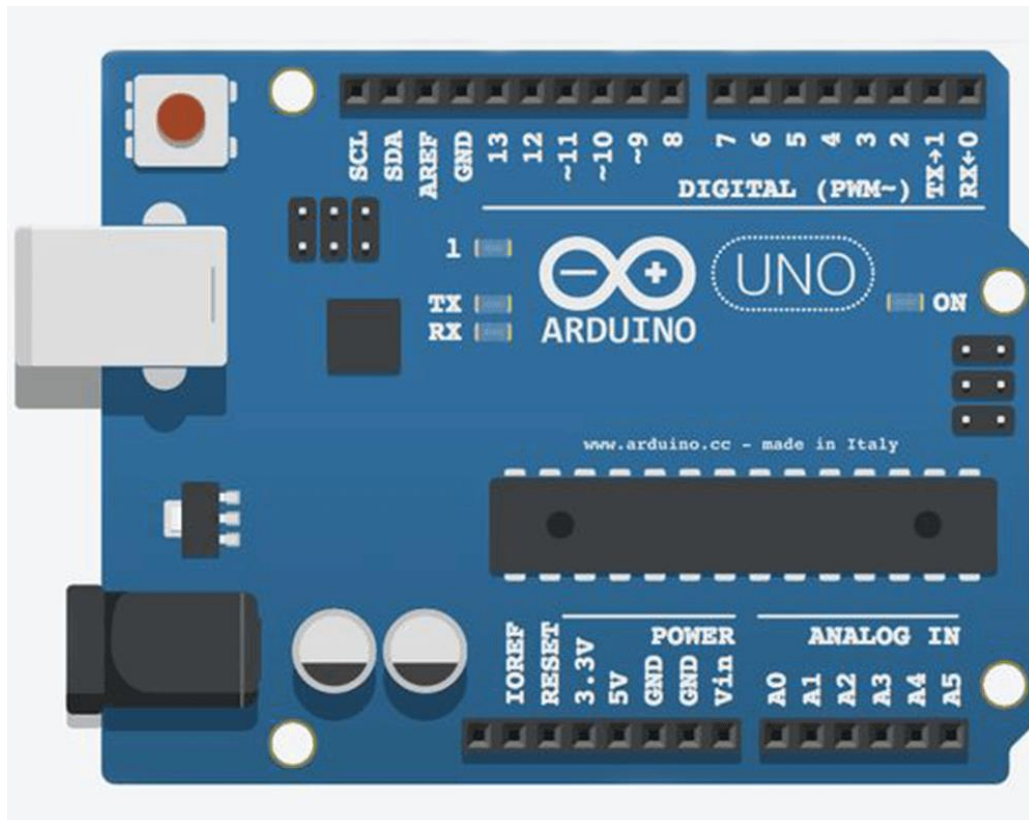
The **Arduino Uno** is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

ULTRASONIC SENSOR HC-SR04:

Ultrasonic sensors measure distance by using ultrasonic waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity. It measures the distance to the target by measuring the time between the emission and reception.

The ultrasonic sensor uses sonar to determine the distance to an object. Here's what happens:

1. The Trigger pin sends a signal: a high-frequency sound.
2. When the signal finds an object, it is reflected and…
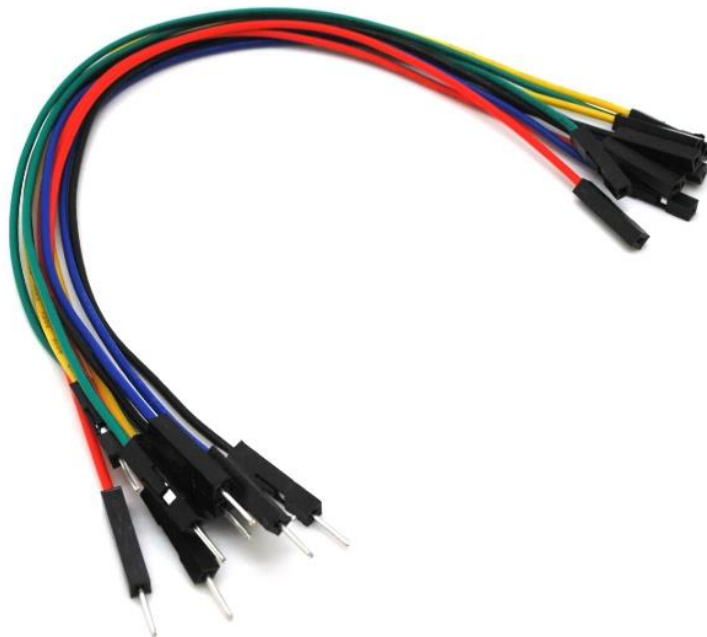3. The Echo pin receives it.

LAPTOP:

Specifications

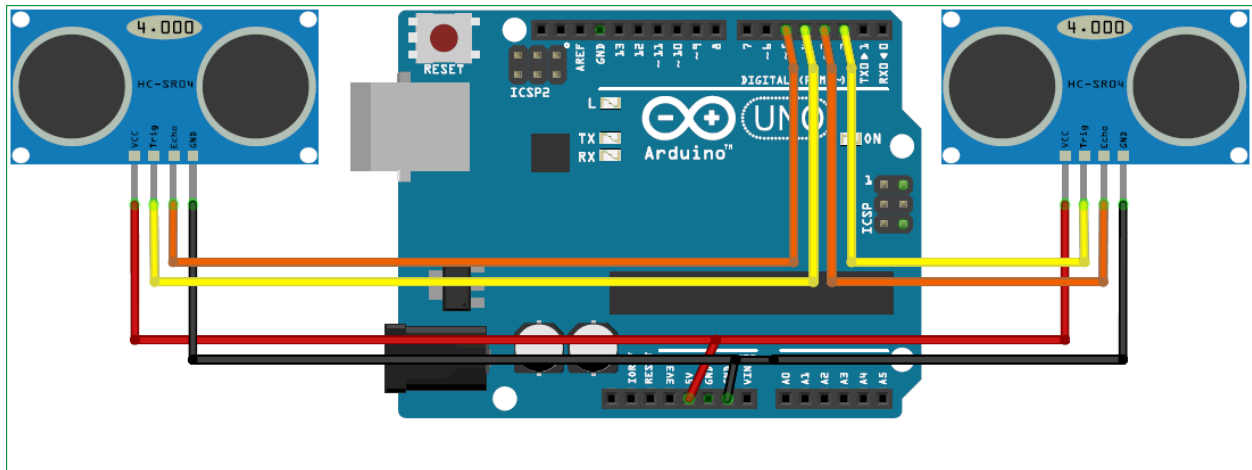| RAM	4GB
PROCESSOR	INTEL CORE i5
CPU ARCHITECTURE	64 BIT
GRAPHICS	AMD RADEON

USB CABLE FOR ARDUINO:

JUMPER WIRES:

IMPLEMENTATION:

The circuit diagram of Arduino part of the project is shown in the above image. It consists of an Arduino UNO board and two Ultrasonic Sensors. All these components can be powered up from the laptop's USB Port.

The Trigger and Echo Pins of the first Ultrasonic Sensor (which is placed on the right) are connected to Pins 2 and 3 of the Arduino. For the second Ultrasonic Sensor, the Trigger and Echo Pins are connected to Pins 4 and 5 of the Arduino.

After the connections are done, the components are placed on the laptop as shown below.

SOFTWARE

REQUIREMENTS

1. ARDUINO IDE

2. PYTHON IDLE

3. VLC MEDIA PLAYER

IMPLEMENTATION

To read hand gestures using the sensors, we will need to program an Arduino using Arduino IDE and design a Python code. 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 being played. There are three modules which we need for this project. PyAutoGUI, pyserial and time module, these three modules have been imported in the python program to perform the job.

Steps:

1. Python 3.7 is downloaded and installed.

2. Open Command Prompt and navigate to the python directory and use the command **python -m pip install --upgrade pip** to upgrade pip. Pip is a tool in python which helps us to install python modules easily.

3. Use the command **python -m pip install pyautogui** to install the pyautogui module.

4. Install pyserial module by using the command **python pip install pyserial**.

5. Open Arduino IDE and type in the required code and save the sketch.

6. Open the Python IDLE and create a new file and type in the required code and save the program.

7. Now, upload the Arduino Sketch and just after that, run the program.

Interaction between python and Arduino program happens by writing arduino port number into python program, which in our case in port COM6.

ARDUINO SKETCH CODE:

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


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>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 debugging
/*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:

```python
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 = "";
```

OBSERVATION

The sensor which is used in this system consumes very less power. The module is developed in low cost. In this system, ultrasonic sensor is used to detect hand gesture or distance of hand and according to condition operation is performed on the laptop. After the execution of both the Arduino Sketch and the Python program, we can use our hand gestures near the ultrasonic sensors where necessary actions will be executed. A function named calculate_distance() is defined, to calculate the distance between the Sensors and the hand each time, we move our hands in front of the laptop, before concluding on any action. This function returns us the distance between the sensor and the hand.

The left ultrasonic sensor is coded such that bringing our hand closer to the sensor will increase the volume and farther from it will decrease the volume of the video which is played in the VLC. Similarly, the right ultrasonic sensor is used to rewind the video when we bring our hand closer to the sensor and fast forward the video when we move our hand away from it. Only when we want to pause or play video, we bring both our hands in front of each of the sensors.

HAND GESTURE	OUTPUT PRODUCED BY SYSTEM
Left hand push in	Volume Increase
Left hand pull out	Volume Decrease
Right hand push in	Video Rewind
Right hand pull out	Video Fast Forward
Both hands in front of the sensor	Play/Pause

CONCLUSION

Hand gesture controlled 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. It is one of the easiest way 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 integration with Arduino UNO in order to achieve different processing and controlling methods for creating new gesture control solutions.

This project shows how HCI (Human Computer Interaction) has changed in the last few years. Numerous research are being done to facilitate easy and flexible HCI. Leap Motion and Sixth Sense Technology are such examples.

REFERENCES

[1] Cost effective hand gesture computer control interface , Gergely Sziládi; Tibor Ujbányi; JózsefKatona2016 7th IEEE International Conference on Cognitive Infocommunications

 [2] Vision based computer mouse control using hand gestures, Sandeep Thakur; Rajesh Mehra; Buddhi Prakash, 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)

[3] A basic hand gesture control system for PC applications C. J. Cohen; G. Beach; G. Foulk Proceedings 30th Applied Imagery Pattern Recognition Workshop (AIPR 2001). Analysis and Understanding of Time Varying Imagery

[4] A system for controlling personal computers by hand gestures using a wireless sensor vice, Kaoru magishi; Lei Jing; Zixue Cheng2014 IEEE International Symposium on Independent Computing (ISIC)

 [5] CMSWVHG control MS Windows via hand gesture, Hafiz Mohsin Abdul-Rashid; Lehmia Kiran; M. Danish Mirrani; M. Noman Maraaj2017 International Multi-topic Conference (INMIC)

[6] Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh, Mo Nours Arab, (2008). "Human Computer Interaction: Overview on State of the Art", International Journal on Smart Sensing and Intelligent Systems, Vol. 1(1).

[7] Joseph J. LaViola Jr., (1999). "A Survey of Hand Posture and Gesture Recognition Techniques and Technology", Master Thesis, Science and Technology Center for Computer Graphics and Scientific Visualization, USA

[8] John M.Zelle, Watburg College.,(2016). "Python Programming: An Introduction to Computer Science"