**A Guide to creating**

**HAND GESTURE CONTROLLED LAPTOP**

**Using ARDUINO**

Modern society is extremely technological. It's difficult to envision an existence without digital devices and applications. Sight & motion is just as essential to us as the importance of mouse & keyboard for a desktop. However, the biggest obstacle is how these two key elements of interaction are connected to one and other.

HMI is a mechanism encompassing of H/W and S/W which aids in interaction and sharing of data between the man (human beings) and the instrument. Devices such as touch screens, switches, LED displays are the most popular HMI machines. The form of communication is via movements or speech methods to smart devices like robot, AI and computers.

Hand motion seems to have the benefit of becoming simple to utilize opposed to some other methods. In this scenario, the hand motions are utilized as the feedback to substitute the cursor and keypad sticky keys characteristics. Such a method has a swift, appealing & simpler communication with desktop. This allows the customer to interact effectively with the model and there's no need for a material customer-system linkage. Rather than utilizing a keypad, cursor or controller, we could use this experiment to influence other aspects of a Computer such as start / stop a vid, navigate through a picture display (towards left or right), swipe vertically on a site & further.

Robust computer vision techniques are needed to detect human movements through perceptual detectors, such as hand monitoring & hand position identification, or locating head motions, facial actions and angle of sight.

##### ABSTRACT

CHAPTER ONE INTRODUCTION 1-4

* 1. [Background **1**](#_TOC_250011)
  2. [Problem Statement **2**](#_TOC_250010)
  3. [Literature Survey **2**](#_TOC_250009)
  4. Objectives **4**

CHAPTER TWO EXPERIMENTAL DETAILS 5-12

* 1. [Hardware & Software Requirements **5**](#_TOC_250008)
  2. Methodology **7**
  3. Data Flow Diagram **9**
  4. Project Timeline **12**

CHAPTER THREE TESTING & ANALYSIS 13

[3.1 Execution of the Program **13**](#_TOC_250007)

CHAPTER FOUR RESULTS & DISCUSSION 14-17

* 1. [Results **14**](#_TOC_250006)
  2. [Observation **16**](#_TOC_250005)
  3. [Benefits & Applications **16**](#_TOC_250004)
  4. [Drawbacks **16**](#_TOC_250003)
  5. [Prospects **17**](#_TOC_250002)
  6. [Conclusion **17**](#_TOC_250001)

[REFERENCES 18-20](#_TOC_250000)

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Name of Figure** | **Page number** |
| 1 | Fig 1: Installing pip function and PyAutoGUI module | 5 |
| 2 | Fig 2: Arduino uno board | 6 |
| 3 | Fig 3: ultrasonic sensor | 6 |
| 4 | Fig 4: circuit diagram | 7 |
| 5 | Fig 5: Basic detection algorithm | 9 |
| 6 | Fig 6: Level 1 diagram for muting and forward/rewind | 9 |
| 7 | Fig 7: Level 1 diagram for minimizing video & volume  increase/decrease | 10 |
| 8 | Fig 8: Level 1 diagram for skipping video and switching tabs | 10 |
| 9 | Fig 9: Level 2 diagram for entire system | 11 |
| 10 | Fig 10: Serial communication takes place once connection  between Arduino and US is established | 13 |
| 11 | Fig 11: Snapshot of Python code being executed according to  the Arduino. | 13 |
| 12 | Fig 12: pause or play | 14 |
| 13 | Fig 13: forward | 14 |
| 14 | Fig 14: rewind | 14 |
| 15 | Fig 15: mute | 15 |
| 16 | Fig 16: switch tab | 15 |
| 17 | Fig 17: minimize youtube screen | 15 |
| 18 | Fig 18: next youtube video | 16 |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Abbreviation** | **Full Form** |
| 1 | H/W | Hardware |
| 2 | S/W | Software |
| 3 | HMI | Human Machine Interface |
| 4 | LED | light-emitting diode |
| 5 | AI | Artificial Intelligence |
| 6 | US | Ultrasonic Sensors |
| 7 | IR | Infrared rays |
| 8 | OS | Operating system |
| 9 | IDE | Integrated development  environment |
| 10 | HCI | Human-computer interaction |
| 11 | SVM | Support Vector Machine |
| 12 | PSO | Particle Swarm Optimization |
| 13 | FMCW | [Frequency Modulated Continuous](https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html)  [Wave](https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html) |
| 14 | CPU | Control Processing Unit |
| 15 | LCD | Liquid Crystal Display |

#### Background

**CHAPTER ONE**

**INTRODUCTION**

Modern society is extremely technological. It's difficult to envision an existence without digital devices and applications. Sight & motion is just as essential to us as the importance of mouse & keyboard for a desktop [1].

Through the 5 sensations, humans communicate in the material universe. But movements were from earlier ages, well before the creation of a dialect, an essential method of communication in the material universe. Experiences to devices became more essential than before in this age of devices trying to take power of all sophisticated acts. Because this paper deals with a motion driven computer, the main objective is on using hand movements for apps. There’re many methods in which a human motion could be seized by a laptop. With range measurement, lens, or information glove, the gesture could be seized. Motions may be seized utilizing technical means such as gaze angle, gestures, expressions, sounds (acoustic) as well as signals like IR and Bluetooth. Systems with embedded circuits and data for influence tasks could be tailored to decrease the instrument's dimensions and expense and boost flexibility and efficiency [2].

This experiment primarily comprises of 3 modules –Arduino Uno, US detectors, and a computer. The Arduino hooked ultrasonic detectors were used to assess the movements as well as the hand range from US detectors. The script written in Arduino discovers the corresponding key phrase for range discovered & directs to OS. Phrases are recognized by Python script running all the time and the correlating digital keystrokes are generated for OS. Now the shortcut keys influence the specific purpose of the intended app. Here we have used VLC media player. The Arduino programming takes place in "Arduino IDE" which is posted to Arduino Uno. We must finish a few further measures in order to operate the necessary python script [3]:

* + 1. download and install Python 2.7.14.
    2. Update the "pip" feature, an easy-to-update method for python nodes.
    3. Using the pip feature, the PyAutoGUI node is installed and accessed from the Python command screen.
    4. The "IDLE (Python GUI)" opens.
    5. A latest document is created & the script is entered.
    6. The document is then executed with above measures; python should be capable of receiving input from Arduino panel and thus produce digital keystrokes in OS. Keystrokes will then regulate the VLC media player shortcut keys.

#### Problem Statement

Common i/p devices such as the keypad and the cursor were not produced for ergonomic support or for the use of handicapped people or amputees. This proved to be a real disadvantage as digital devices and electronics were not accessible by everyone.

Current methods of communication do not facilitate interaction with the laptop beyond a minimum distance of 15cm. Majority require physical contact.

#### Literature survey

Several of modern HCI interfaces involve specific directions from customer in the manner of keypad presses or cursor snaps A novel technique for recognizing hand movements for HCI is suggested in this article by Nikita Gupta et al (2016) utilizing image processing and computer vision methods. This type of feedback, though, often leads to questions that aren't always pertinent to conventional feedback. Such issues on the customer's end are learning, remembering & performing movements correctly [5].

A hand motion identification model provides non-verbal interaction in an organic, creative & contemporary manner. It has a broad range of HCI and sign dialect applications. M. Panwar and

P. Singh Mehra (2014) proposed a model to evaluate a new method to hand motion identification premised on the identification of certain pattern-based characteristics. The issues are learning, remembering, and performing movements correctly. The programmer must supply a model that acknowledges such movements accurately [6].

W Zhi-heng et al (2017) proposed a HCI Control model premised on motion identification is constructed through LabVIEW. First, a gesture survey information is gathered utilizing 5 data glove bending sensors, then the information gathered is processed beforehand to enhance the acknowledged accuracy and the value obtained by the SVM kernel variable is discovered utilizing

the enhanced PSO algorithm to modify it. In addition, an enhanced PSO-SVM categorization method for hand-motion identification is suggested to address the current issues of reduced accuracy and bad real-time capability in the motion identification system [7].

We're surrounded by many household devices & laptops. Through this research K. Yamagishi et al (2014) proposes, as an easy-to-use interface, to create a model for controlling pcs utilizing Arduino established Hand Motion Control of Desktop App that applies man’s normal attitude. Study problems involve defining the connection among the processes & motions of the Computer, recognizing hand movements, adjusting the hand motions ' mistake, as well as how the model can be realized [8].

Throughout the area of computing tech, with new innovations & advancement, the dimensions of electronics are quickly diminishing. Such machines therefore necessitate a fresh feedback interface. By implementing sight tech & regulating the machines through normal hand movements, we may decrease the necessary workspace. D. K. Singh (2015) publication, suggests a unique approach utilizing movements utilizing a vid machine to regulate the computer. There are already straightforward interfaces, like built-in keypad, folder keypad and mini keypad. But these interfaces are immobile, that is they need a proper set up to use and cannot be used on the go [9].

One coming up industry for HCI is contactless hand movements. Throughout this article presented by M. Q. Nguyen et al (2018), we examine the viability of a mixed model utilizing FMCW radar and US detectors that has one transmitter and receiver to perceive manual motion for desktop command. This type of feedback often leads to questions with conventional feedback which are not pertinent. On the client side, such issues are learning, remembering, & executing movements correctly [10].

According to the publication by Y. V. Parkale (2012), HCI continues to move more towards organic & interfaces that are intuitive compared to conventional cursor and keypad. Hand movements are essential for HCI. Such interfaces require a certain level of room to utilize & could even not be utilized during motion [11].

The paper presented by G. Sziladi et al (2017) indicates just how a motion dictated model is designed & implemented that defines a motion from a hand action. The motion identified with range measuring detectors & the control interface discussed decides the motion attached as per the

motion of the hand. The programmer must not only make sure that motions will be identified promptly & accurately, but must still include a manual which enables fast & simple training of such movements [12].

Hand motion-controlled gadgets is receiving more exposure in current times. There’re many various methods for manipulating gestures. In U. Rajkanna et al (2014) work, the motion commands the mobile robot's motion. The constructed model is categorized in to mobile robot system & motion system. With higher performance this current structure is formed inexpensively. In several other projects, this could be incorporated, like holding sound systems to help people who are mute, toys etc. varied action & airborne movements are harder to learn than solo action movements are. In latter scenario the hand body language is immaterial - consumers just have to adopt an appropriate direction to execute an instruction [13].

According to Gaurav et al (2018) with a rising popularity of mobiles, hand motion-based communication may have landed in our daily lives, however we are often not taking total benefit of its capability. Movements supply the consumer with a special type of communication which reflects his natural-world involvement. They feel authentic & do not require disturbance or any extra instrument. But movements often address problems which are not applicable to conventional feedback techniques. An other element is the structure of movements themselves, that must render them meaningful and simple to perform and comfy to perform [14].

#### Objective

* + 1. To recognize the various gestures performed by the user
    2. To convert motion into commands and to communicate the data in real time

#### CHAPTER TWO

**EXPERIMENTAL DETAILS**

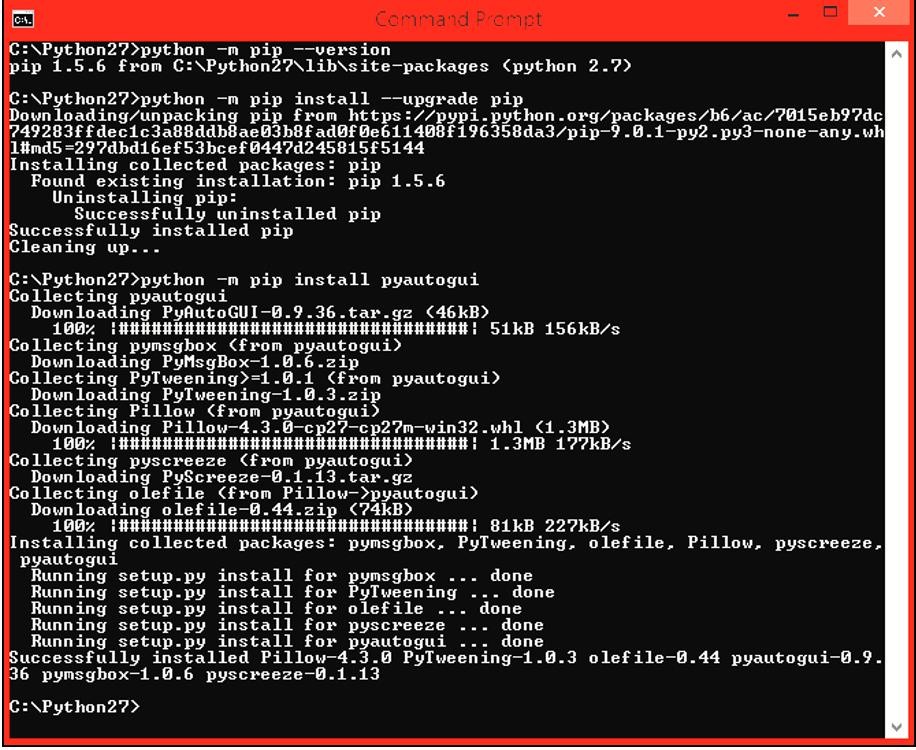
#### Hardware & Software Requirements

##### Software Used

You must code an Arduino & style a Python script to perceive hand movements. Operating system doesn't acknowledge Arduino instructions, and we use Python to link serially and switch the obtained information into digital keystrokes. Such digital keystrokes regulate the VLC media player's shortcut keys that in effect regulate the recording of the vid and sound.

The Arduino IDE software runs well enough to code the Arduino, but upon setting up Python 2.7 as seen in Fig. 1, we must conduct a few further measures for Python. The measures involved were [15]:

* + - 1. Update the python pip feature
      2. using the pip feature download the PyAutoGUI node.



*Figure 1: Installing pip function and PyAutoGUI module*

#### Hardware used

##### Arduino UNO

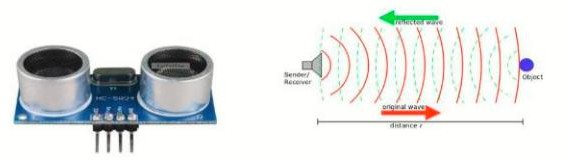
The panel of Arduino Uno is premised on an ATmega328. Arduino UNO is a free software framework that is straightforward for novices to understand. Its regular connectors, that allows the CPU panel to link to multiple complementary components referred to as shields, perform an essential position in Arduino. Shields offer Ethernet, GPS, breadboard, motor control or LCD. Arduino IDE is utilized for coding Arduino panels across a serial association in C & C++ scripts. [16]



##### Ultrasonic sensor

*Figure 2: Arduino UNO board*

To evaluate range to an item, the US detector is utilized. The ultrasound is emitted & the noise moves throughout the atmosphere. As it reaches an item, it is directed back to the US detector. There are four rods in the US detector: voltage at the common collector, utilized to control the detector, Trigger, to deliver the Ultrasonic signals, & Echo, the o/p rod. The ultrasonic signal is sent towards the detector by the echo rod and the ground is connected to the GND of the Arduino panel [17].



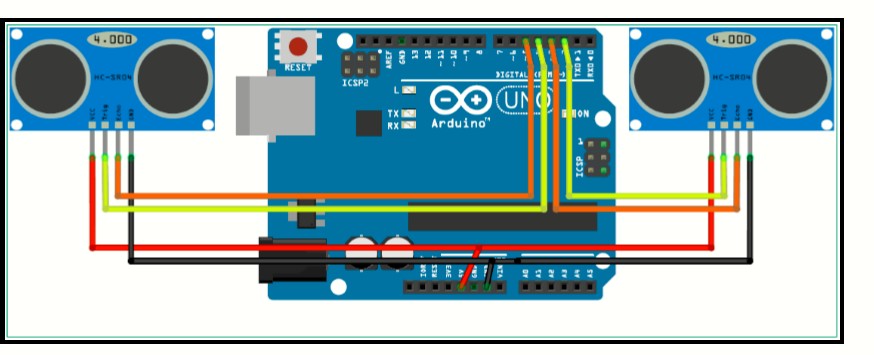
*Figure 3: Ultrasonic sensor*

##### USB Cable (for Arduino)

* + - 1. **Few Connecting Wires**
  1. **Methodology**

With Arduino, you utilize 2 US detectors, position the hand next to detector, & determine the range among the hand & the detector. utilizing this data, it is possible to conduct appropriate desktop acts. The US detectors ' placement is quite essential. Position the 2 US detectors at each tail edge of computer monitor. A Python code collects the range data from Arduino and a unique library named PyAutoGUI converts the information in to tap behavior on a keypad.

##### Circuit Diagram



*Figure 4: Circuit Diagram [16]*

Fig. 4 shows the circuit diagram used for the project. It has 2 US and 1 Arduino UNO board.

All these parts are given power form the laptop’s USB port.

##### Algorithms

**Step 1: Designing the Project:**

The circuit layout is straightforward, although it is very necessary to set up the elements. The 1st US detectors Trigger and Echo Prongs (the one on the left of the monitor) are linked to the Arduino's Rods 11 & 10. For the 2nd US detector, the Trigger and Echo Prongs are linked to rods 6 & 5 of the Arduino. Position the US Detectors at the edge of the Computer monitor, on the ends of the monitor. To keep the detectors onto monitor with the use of scotch-tape

##### Stage 2: Programming the Arduino such that it can Detect Gestures:

The main portion of this experiment is to compose a script for Arduino so the ranges calculated from both detectors can be converted into the suitable instructions to regulate certain behavior. It is possible to measure the hand movements beside the US detectors so they may conduct 5 various functions in your PC. Let's see few of the functions we may achieve before we take a glance at the movements:

1. swapping tabs on the net
2. increasing/decreasing motion of vid in VLC Player
3. start/stop vid in VLC Player
4. boost sound
5. reduce sound, etc.

##### Stage 3: Programming in Python for the Project

We must study Arduino's Serial information and invoke some keypad taps. To do this, a unique Python unit termed PyAutoGUI must be installed. Few of the distinct hand movements are as follows:

**Motion1:** position the hand at a range (40-50 centimeters) across from both US detectors for a short period of time. This motion starts the vid or stops it (controls the space bar).

**Motion 2:** position the hand at a range (13-17 centimeters) next to the Right US detector for a short period of time and shift the hand toward detector. This motion should boost the sound although we can reduce the sound when we move the hand away from the detector.

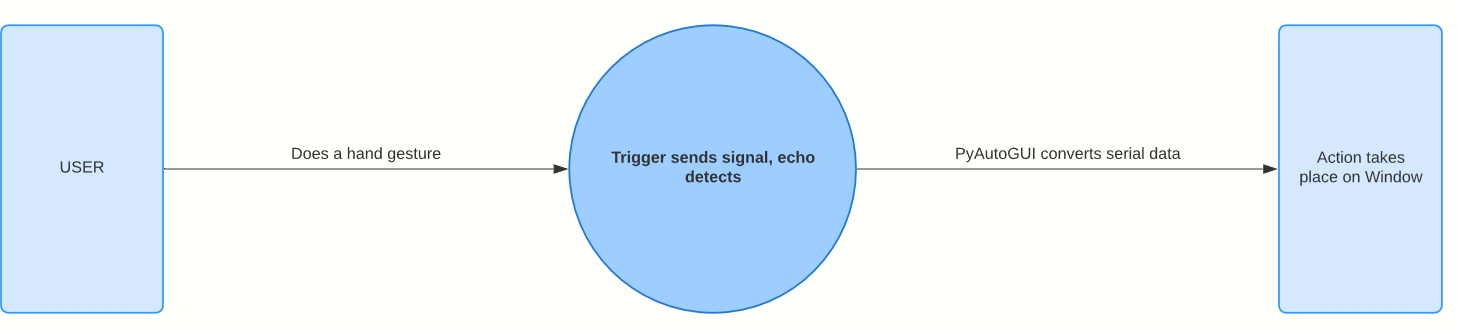
**Motion 3:** Flick the hand at a range of more than 30 centimeters next to the Left US detector. This motion should swap among tasks.

**Motion4:** Position the hand at a range (13-17 centimeters) beside the Left US detector and shift the hand toward detector. This motion should quickly forward the Clip when moving your arm away from the detector at a range ( 20 -30 centimeters) should slow vid down.

**Motion 5:** If both hands are positioned from the two detectors at a range of 10-20 centimeters, then we can close the tabs.

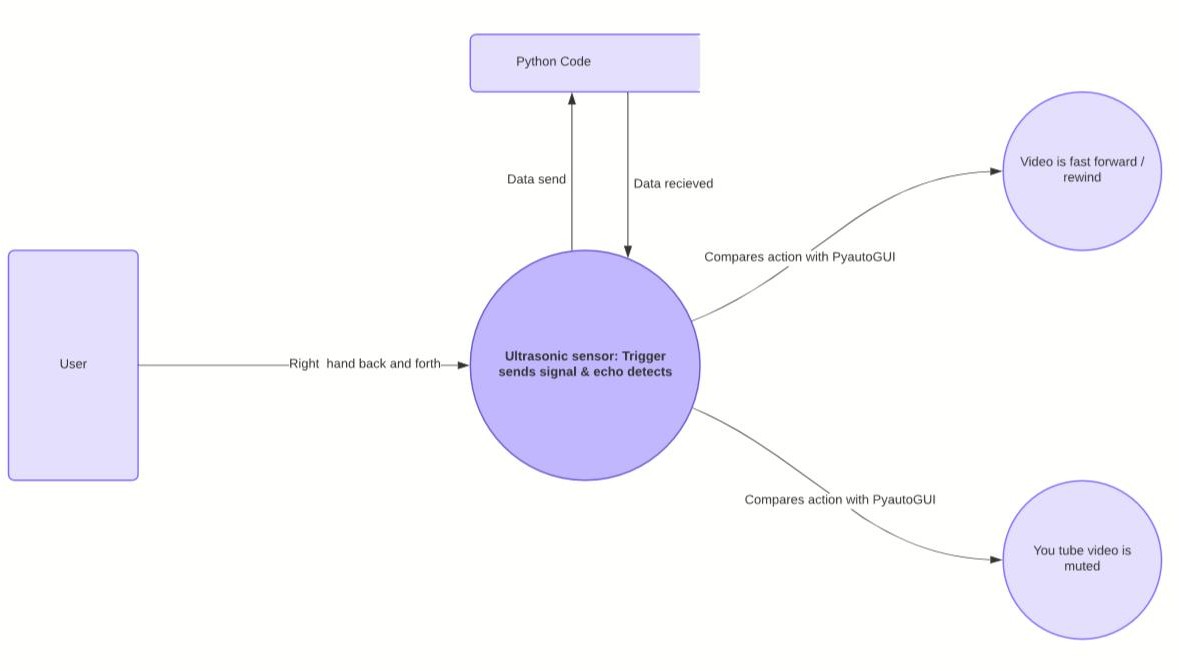
#### Data Flow Diagram

1. **Level 0**

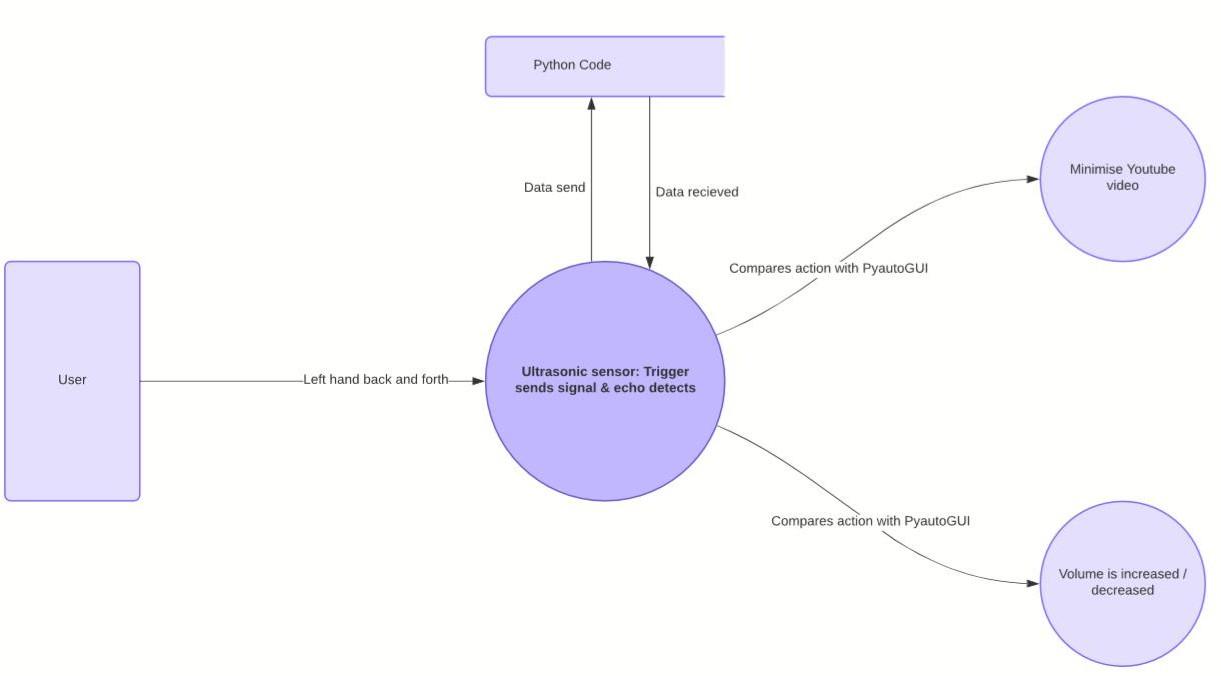


1. **Level 1**

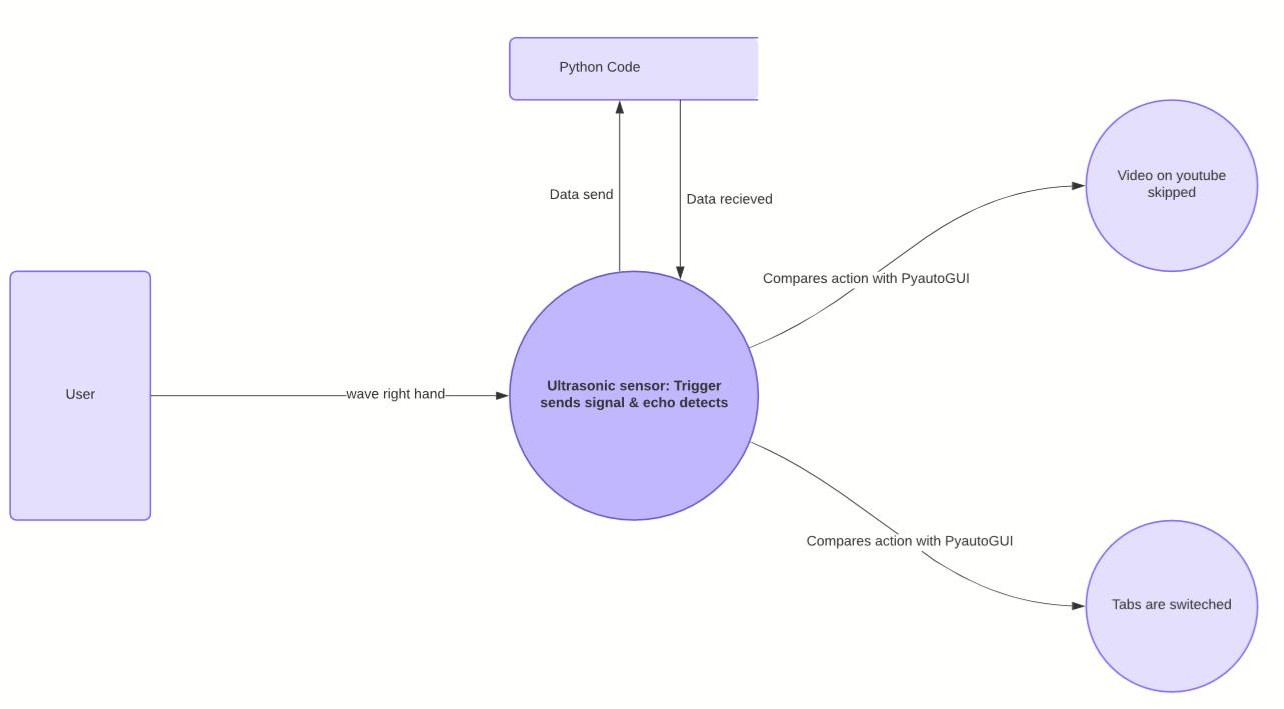
*Figure 5: Basic detection algorithm*



*Figure 6: Level 1 diagram for muting and forward/rewind*

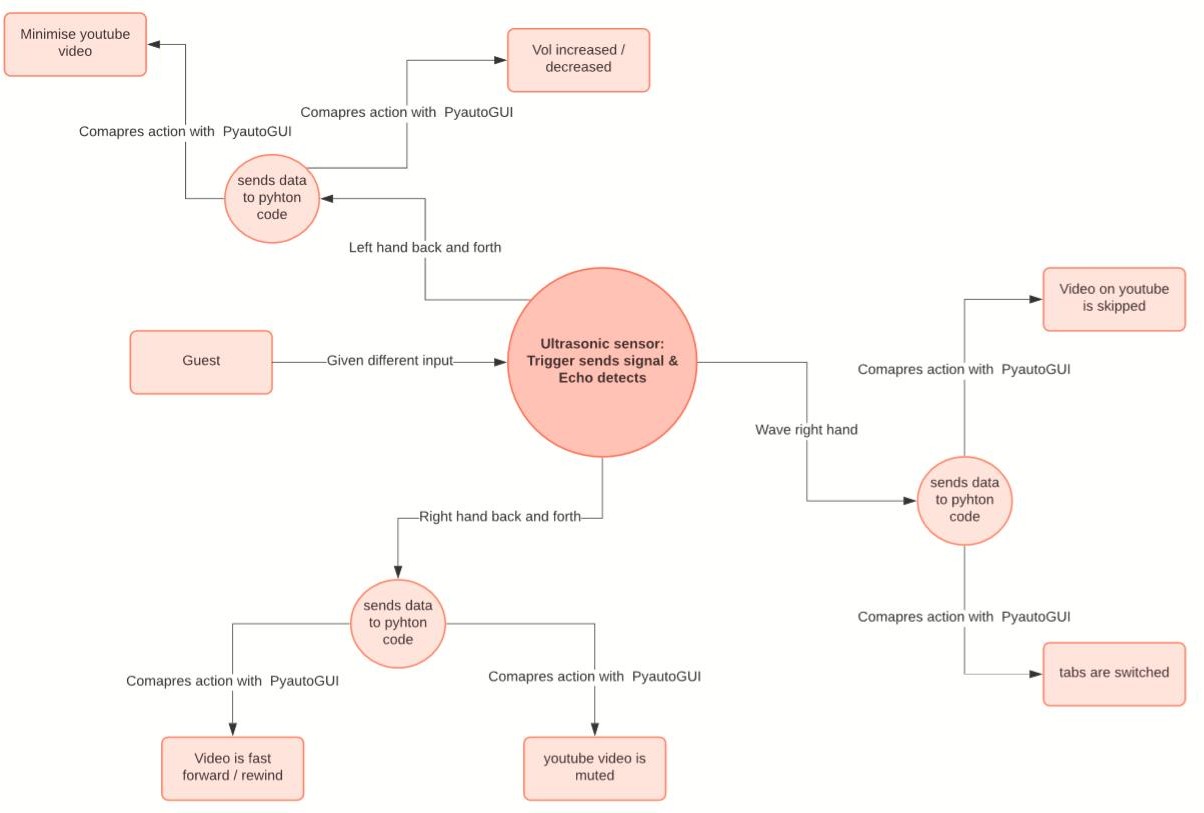


*Figure 7: Level 1 diagram for minimizing video & volume increase/decrease*



*Figure 8: Level 1 diagram for skipping video and switching tabs*

#### Level 3



*Figure 9: Level 2 diagram for entire system*

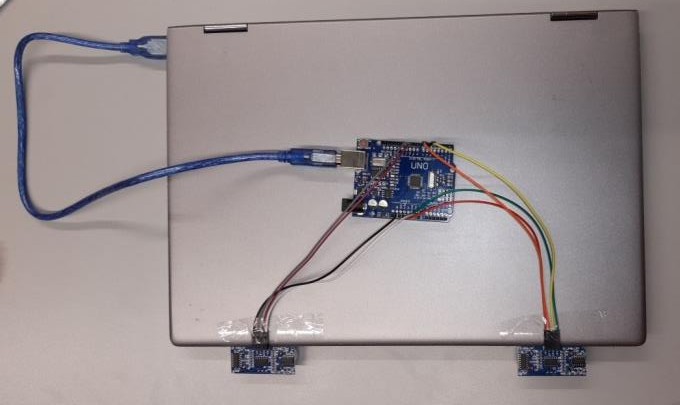
#### Project Timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Stages** | **Week 1** | **Week 2** | **Week 3** | **Week 4** | **Week 5** | **Week 6** | **Week 7** | **Week 8** | **Week 9** | **Week 10** | **Week 11** | **Week 12** |
| **Buying materials** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Installing software** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Learning the languages** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Coding** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Execution and debugging** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Finalization and vid of working mode** |  |  |  |  |  |  |  |  |  |  |  |  |

**CHAPTER THREE**

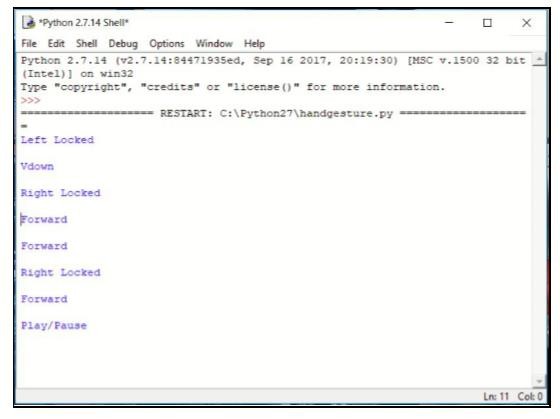
**TESTING & ANALYSIS**

### 3.1 Execution of The Program



*Figure 10: Serial communication takes place once connection between Arduino and US is established*

We operate the python script once the configuration is finished (Fig. 6). Figure 7 is how monitor should occur when the Run Unit is clicked. Then we can use our hand movements in the vicinity of the US detectors in which movement is needed.



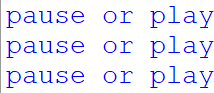
*Figure 11: Snapshot of Python code being executed according to the Arduino.*

#### CHAPTER FOUR

**RESULTS & DISCUSSION**

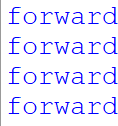
* 1. **Results**

Here are few of the many actions that take place in this project:



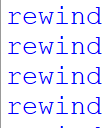
*Figure 12: pause or play*

In fig. 12 we can see that when both the hands are raised at 30 cm from the US sensor the video is paused/ played.

*Figure 13:forward*

In fig. 13 we can see that when the right hand moves backwards between 15 -30 cm the video is fast forwarded.

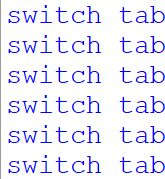
*Figure 14:rewind*

In fig. 14 we can see that when the right hand moves forward between 30 -15 cm the rewinding takes place.

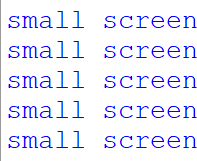
*Figure 15:mute*

In fig. 15 we can see that when the left hand is at 15 cm the muting of video takes place.



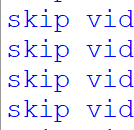
*Figure 16:switch tab*

In fig. 16 we can see that when the left hand moves at 10 cm the switching of tabs takes place.



*Figure 17:minimize youtube screen*

In fig. 17 we can see that when the right hand moves at 25 -50 cm the minimizing of youtube screen takes place.

*Figure 18:next youtube video*

In fig. 18 we can see that when the right hand is waved in front of the US detector the YouTube video is skipped.

### Observation

Much less energy is consumed by the detector in use in this model. The sensor required is inexpensive. US detectors are utilized in this model to identify hand movements or hand ranges and operate on a desktop depending on the situation. The alternative seen in the paper is user- friendly and feasible. [18]

### Benefits & Applications

Few of the benefits and Applications of this project is as follows[19-21]:

* + 1. No need to generate noise for such a model.
    2. Using an US detector, many desktop tasks could be performed.
    3. Such a method can be quite helpful for anyone that doesn't understand the desktop technically.
    4. Such a method cuts the time needed for learning.
    5. It is simple to communicate with the desktop utilizing this method and there's no cultural difference.
    6. Utilizing this model, we could regulate our computer by a short range & aid guide the computer in a presentation.
    7. Such a technology may be used in classrooms for simpler and more immersive studying, interactive gaming and on-screen interaction with virtual objects.
    8. Once combined with fully formed hologram tech, it can become more powerful.

### Drawbacks

* + 1. The proximity sensors could be damaged easily and thus produce a malfunction.
    2. there is a limit to the number of actions that could be applied per program as the action range is dependent on the range of the sensor.
    3. The HC- SRO4 node could be affected by the noise present in the surroundings.

### Prospects

Hand motion / gestures strategy is not restricted to games, it may be effective for clinical apps with simple machine control. The method of hand motion could function as a technique of feedback among surgical equipment & the man as suggested. it could be utilized to operate any & all desktop functionalities [20].

When mastered & properly utilized, conventional feedback machines such as keypad, mouse, & screens may simply be pointless. besides the facts motion control appears quite neat apart from being practical. As in the ‘Minority Report’ hit film in 2002, a moment may likely arrive when all is regulated by movements. though it may appear like a tech which will only boost our tiredness, the reality being that, besides making things better, it has a range of uses in nearly every profession. [21-22]

### Conclusion

We draw the conclusion by implementing our analysis as follows: - regardless how complicated and effective, we always must be near a machine and somehow in physical contact with us to interact with it. But the tech of motion identification may alter everything. Our paper introduces amongst many others one of the solutions. This is a budget-effective system based on Arduino UNO & US detectors. The python IDE enables smooth convergence to Arduino UNO to accomplish separate techniques of extraction & regulating to create new alternatives for motion control.

### REFERENCES

1. Khasale, Mayuri, and Aparna Gurjar. “Computer Control with Hand Gestures Using Ultrasonic Sensor.” *International Journal of Innovations & Advancement in Computer Science,* vol. 7, no. 5, May 2018, pp. 361–367.
2. Mukherjee, Rohit Sachin, et al. “Hand Gesture Controlled Laptop Using Arduino.” *International Journal of Management, Technology And Engineering, vol. 8, no. 10,* Oct. 2018, pp. 1037–1043.
3. Sziladi, Gergely, et al. “Cost-Effective Hand Gesture Computer Control Interface.” 2016 *7th IEEE International Conference on Cognitive Info communications (CogInfoCom)*, 2016.
4. Sawardekar, Gaurav, et al. “Arduino Based Hand Gesture Control of Computer Application.” *SSRN Electronic Journal,* 2018.
5. Agrawal, Rishabh, and Nikita Gupta. “Real Time Hand Gesture Recognition for Human Computer Interaction.” *2016 IEEE 6th International Conference on Advanced Computing (IACC),* 2016.
6. M. Panwar and P. Singh Mehra, "Hand gesture recognition for human computer interaction," *2014 International Conference on Image Information Processing,* Shimla, 2014, pp. 1-7.
7. W. Zhi-heng, C. Jiang-tao, L. Jin-guo and Z. Zi-qi, "Design of human-computer interaction control system based on hand-gesture recognition," 2017 *32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC),* Hefei, 2017, pp. 143-147.
8. K. Yamagishi, L. Jing and Z. Cheng, "A system for controlling personal computers by hand gestures using a wireless sensor device," *2014 IEEE International Symposium on Independent Computing (ISIC),* Orlando, FL, 2014, pp. 1-7.
9. D. K. Singh, "Recognizing hand gestures for human computer interaction," 2015 *International Conference on Communications and Signal Processing (ICCSP),* Melmaruvathur, 2015, pp. 03790382.
10. M. Q. Nguyen and C. Li, "Radar and ultrasound hybrid system for human computer interaction," 2018 *IEEE Radar Conference (RadarConf18),* Oklahoma City, OK, 2018, pp. 14761480.
11. Y. V. Parkale, "Gesture Based Operating System Control," 2012 *Second International Conference on Advanced Computing & Communication Technologies,* Rohtak, Haryana, 2012, pp. 318-323
12. G. Sziladi, T. Ujbanyi, J. Katona and A. Kovari, "The analysis of hand gesture based cursor position control during solve an IT related task," 2017 *8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom),* Debrecen, 2017, pp. 000413-000418.
13. U. Rajkanna, M. Mathankumar and K. Gunasekaran, "Hand gesture based mobile robot control using PIC microcontroller," 2014 *International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]*, Nagercoil, 2014, pp. 1687-1691
14. Sawardekar, Gaurav, et al. “Arduino Based Hand Gesture Control of Computer Application*.” SSRN Electronic Journal*, 2018.
15. Karray, Fakhreddine, et al. “Human-Computer Interaction: Overview on State of the Art.” *International Journal on Smart Sensing and Intelligent Systems, vol. 1, no. 1*, 2012, pp. 137–159
16. Soni, Neeraja, et al. “Airborne Ultrasonic Sensor Node for Distance Measurement.”

*International Journal of Engineering Science and Computing,* Mar. 2017, pp. 5991–5991.

1. S. Szeghalmy, M. Zichar and A. Fazekas, "Comfortable mouse control using 3D depth sensor," 2013 *IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom*), Budapest, 2013, pp. 219-222.
2. X. Li, Hand gesture recognition Technology in Human-computer interaction, *School of Computer Science, University of Birmingham,* 2016.
3. Thakur, Sandeep, et al. “Vision Based Computer Mouse Control Using Hand Gestures.” 2015 *International Conference on Soft Computing Techniques and Implementations (ICSCTI),* 2015
4. Joseph J. LaViola.. “A Survey of Hand Posture and Gesture Recognition Techniques and Technology”. *Technical Report. Brown University*, USA.1999.
5. N. Anju Latha, B. Rama Murthy, K. Bharat Kumar. “Distance Sensing with Ultrasonic Sensor and Arduino”, *International Journal of Advance Research, Ideas and Innovations in Technology,*2015, pp.1-5.
6. Salunke, Tejashree P., and S. D. Bharkad. “Power Point Control Using Hand Gesture Recognition Based on Hog Feature Extraction and k-Nn Classification.” 2017 *International Conference on Computing Methodologies and Communication (ICCMC),* 2017

ORIGINALITY REPORT

2%

SIMILARITY INDEX

1%

INTERNET SOURCES

1%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

# ux.dailyrecord.com

1

1

Internet Source %

# Submitted to University of Warwick

Student Paper %

2

1

# Submitted to Trinity College Dublin

Student Paper %

3

1

Exclude quotes Off Exclude bibliography On

Exclude matches Off