Introduction:

The evolution of technologies over time has led to the introduction of open source platforms which provide various opportunities to approach a solution. The transformation from a room size computer to wearable compatible computers has motivated towards making of touch sensor systems which has been enhanced to absolutely no-touch required systems. This remarkable transformation to no-touch based systems i.e. sensor operating system has inspired to build this project. In this project, the effective translation between gestures and the type of operation to be performed in the computer is purely decided on the distance between the hand and the sensor.

Instead of using human touch required devices like Keyboard, Mouse or Joystick, the new approach of sensor-based systems will open wide opportunities to operate the computers. Nowadays, we see most of the LCD Screens are supporting the Finger-Touch Technology. The idea behind this project is to build a Human Machine Interface (HMI) by reducing the Touch-Based Systems and replacing them with Sensor- Based Systems. The idea of integrating this project with Python has unveiled a new dimension to the Arduino Programming. With the support of sensors, more features can be added to improve a human interaction with the computer.

This project is an amalgamation of Arduino and Python platforms. Arduino is considered as one of the most powerful and compatible electronics prototyping platforms. On the other hand, Python is the most advanced and widely used open source platform for high-end programming. The interaction bridge built between Arduino and Python gives the privilege to understand the distance of hand from the sensor to reflect the respective operation in the computer.

This project incorporates twelve different operations in the computer on different applications which are pre-installed. To perform these operations, we require 2 Ultrasonic sensors, each sensor has its own set of distance ranges and operations to perform. The moment any object or hand interferes the waves of the sensor, the Arduino program calculates the distance at which the interference has been encountered and hands over to Python for finding out which operation to perform in the computer based on the value of the distance.

Out of the twelve operations, seven operations are being performed on the VLC Media Player, 3 operations on the Chrome Web Browser, one operation for recording a live video from the webcam installed in the laptop and finally to shutdown the laptop.

Background:

Initially to put pen to paper to list down the ideas to integrate Arduino and Python which opens wide doors for supporting new functionalities, we browsed for project which supports these two platforms. One of the best results found was to gesture control the computer. The publisher of this project has made a video which acts as a manual for activating the features using gestures.

The idea of this existing project is to implement five features on VLC Media Player. These features are unique and demonstrate their own importance and purpose. The following is the description of each feature.

1. Play/Pause: By placing both the hands as near as possible to the Ultrasonic sensor will cause either Pause or play action.
2. Rewind the video: Place the hand slightly far from the left most Ultrasonic sensor caused the 'Rewind' action.
3. Forward the video: Place the hand in front of right sensor at a specific distance.
4. Increase volume: Position the hand far from left sensor and pull it away from the sensor.
5. Decrease volume: Position the hand at a distance far from the right sensor and push the hand toward the sensor.

We must appreciate the design of minimal use of the sensors and yet to manage to bring out these features is impeccable. The author of the existing project has given in-detail list of the libraries which are required to start the development process.

There is a speciality of each library in supporting the features. They provide user level access to the program to access and interact with the kernel libraries in the computer. The following are the pre-requisite libraries in Python that are suggested by the author to make the existing project to work

1. The version of python should be 2.7.14 - 2.3.x.
2. Installation of pySerial, which enabled the python to read the Serial Monitor of Arduino.

The idea is to capture the distance between the hand and the sensor. The author of the existing project has come up with the thought process of capturing the distance, when the distance meets the required distance which is set up in the Arduino program, print the necessary action to be made to the Serial Monitor.

The library pySerial in Python will be listening to the Serial Monitor, as soon as the action name is printed on the Serial Monitor matches with one of the actions in the Python program, that corresponding code will be executed to perform the action.

To perform the above 5 operations, the hand must interfere the Ultrasonic sensor waves at a distance which is registered in the program. Each feature has the distance value in unique.

Using the same methodology, we have included the eight additional operations as an extension to this project.

Design Process:

The design process has started by keeping in mind about the number of features that are going to take place in the project. Initially, there were twenty features listed by holding on to three Ultrasonic sensors. Unfortunately, due to the scarcity of the availability of those sensors, I had to stick with two Ultrasonic sensors with the supporting features of twelve.

Since, the sensors are overloaded with equal number of responsibilities to do, a detailed classification of which operation must be done at what distance of hand from the sensor to trigger the functionalities.

One of the major uncertainties of the Ultrasonic sensor is the accuracy of the distance. This project is designed in such a way that it requires a huge perimeter of open space. Thus, assumption is made that the perimeter is big enough such that the waves don't repel.

Python plays an import role in triggering the actions on the VLC Media Player, Google Chrome and Camera applications. The research has been conducted to find out the libraries that supports user level actions on the applications.

After gathering and installing required libraries, the build process has started according to the milestones set during the project proposal.

Materials Required:

According to the above design process, the following materials are needed:

•Arduino UNO

•Ultrasonic sensors (HC-SR04)-2

•Connecting Wires

•The USB cable for connecting Arduino

•A Laptop

Build Process:

The building process of this project was very simple. All it needed was two Ultrasonic sensors, Arduino UNO Board and a power source to make them work. The Arduino board and Bread board are connected to the backscreen of my laptop conveniently such that the Arduino Board USB cable connection could reach the USB port of Laptop. the two Ultrasonic sensors are placed to the right-hand side and left-hand side of the Webcam. The position of these sensors is determined in such a way that the Ultrasonic waves will not be interfered by the circuit connection. All the connections are fixed to their position with the help of transparent tape.

The left-hand side of the ultrasonic sensor's Trigger and Echo pins are connected to the Digital Pin 11 and Digital Pin 13 respectively. On the other hand, the right-hand side of the ultrasonic sensor's Trigger and Echo pins are connected to the Digital Pin 5 and Digital Pin 3 respectively. The power source of 5V and GND is utilized from the Arduino Board. Both the Ultrasonic sensors share the power source with the help of Bread Board.

These sensors are not soldered into the Bread Board, they were connected using Male to Female jumper wires. This design is inspired from the existing project work. The published author has designed this format for the user's convenience. Since the signals generated from Ultrasonic sensors repel for minute interference.

The following are the list of Python libraries that must be installed:

- Numpy: It supports the numerical operations which are used for computing the frames while using opencv library.
- Opencv: This library helps to record a video. It has in built functions such as VideoCapture() and VideoWriter() which records the video and save it on our local file system.
- Os: This library helps to give commands to the Operating System such as shut down or restart the application/laptop at kernel level.
- Pyautogui: The short-hand keys such as 'cntrl','pageup','pagedown' and other hot keys can be executed through this library.
- Pyserial: From this package, the Serial module will help us to read the Serial Monitor of the Arduino. Set the port number and the baud rate at which the Serial Monitor is running.
- Time: Using this library we make a note of the time to stop the video recording after two minutes. The counter is set until it meets the requirement of time.

The above libraries must be downloaded with the latest version from python.org.

The following are the features which are which are implemented in this project. These features operate at various distances between the left and right ultrasonic sensors. The short key operations are performed using Python Programming.

1. Play/Pause video: These two operations are performed on the VLC Media player by placing the hand very close to the left most Ultrasonic sensor. These operations are performed using the short key 'space'.
2. Rewind video: Rewinding the video using the short key operation 'ctrl'+' left'. The hand must be slightly away from the sensor when compared to the hand position in feature 1.
3. Forward video: Forwarding the video using the short key operation 'ctrl'+' right'. The position of the hand must be slightly away from the position compared with feature 2.
4. Increase volume: Increase the volume of the video by placing the hand slightly away from the left sensor. The short key used for this operation is 'ctrl '+'up'.
5. Decrease volume: Decrease the volume of the video by pushing the hand slightly towards the left sensor. The short key used for this operation is 'ctrl '+'down'.
6. Close the VLC Media Player: This operation is called as the "quit" operation. It is performed by placing the hand far away from the right sensor.

7. Scroll up the webpage: For this operation, it is assumed that the webpage is already open. It is not based in the short key, instead we use scroll(100) function. This operation is done using left sensor.
8. Scroll up the webpage: This operation is performed when the webpage is open readily. The scroll(-100) function is used to perform page up operation. This operation is doe using left sensor.
9. Switch between webpage tabs: To navigate between various tabs open in Google Chrome/ any web browser, we use the short hand key 'ctrl '+'tab'.
10. Record Video: This operation is not made through short hand keys instead, VideoCapture(), VideoWriter(), cvtColor() and write() functions from cv2 module of opencv package help us to record and save the video on local file system.
11. Shutdown Laptop/Computer: Since this operation should directly communicate the command to the operating system. The library os will support this operation. The function os.system("Shutdown /s /t 1") will initiate the shutdown command with user level access.

These features are heavily depending upon the distance maintained between the hand and the sensor. If their respective distance is not maintained, the functionalities may overlap with each other. The accuracy of calculating the distance between the hand and sensor is utmost important.

The