**CHAPTER 2**

**2.1**

**PROJECT PERSPECTIVE:**

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.

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.

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.

**2.1.1**

**PRINCIPLE BEHIND THE PROJECT:**

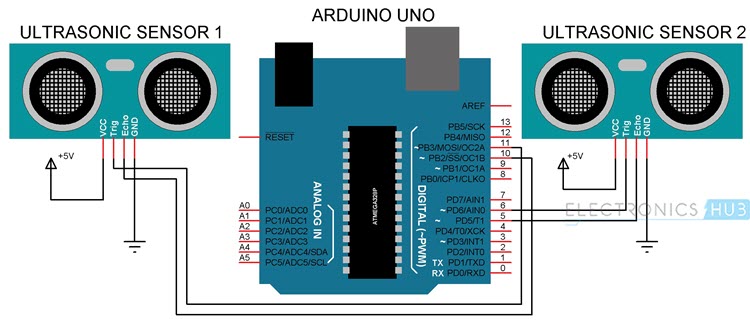
The principle behind the Arduino based Hand Gesture Control of Computer is actually very simple. All you have to do is use two Ultrasonic Sensors with Arduino, place your hand in front of the Ultrasonic Sensor and calculate the distance between the hand and the sensor. Using this information, relevant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

**2.1.2**

**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.



**2.2**

**PROJECT FUNCTIONS**

**2.2.1**

**WORKING:**

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.

We have already seen a project called [**PORTABLE ULTRASONIC RANGE METER**](https://www.electronicshub.org/portable-ultrasonic-range-meter/), where you can measure the distance of an object placed in front of an Ultrasonic Sensor with the help of Arduino.

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.

* 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.

**2.2.2**

**PROGRAMMING CONCEPTS**

**ARDUINO CODE CONCEPT**

In the Arduino Code, the gesture mentioned above have been converted into 5 Commands that are sent to the Serial Port. Using these 5 commands, you can write a Python Program to control certain Keyboard Functions in order to achieve the required task.

**PYTHON PROGRAMMING**

Writing Python Program for Arduino based Hand Gesture Control is very simple. You just need to read the Serial data from Arduino and invoke certain keyboard key presses. In order to achieve this, you have to install a special Python Module called PyAutoGUI.

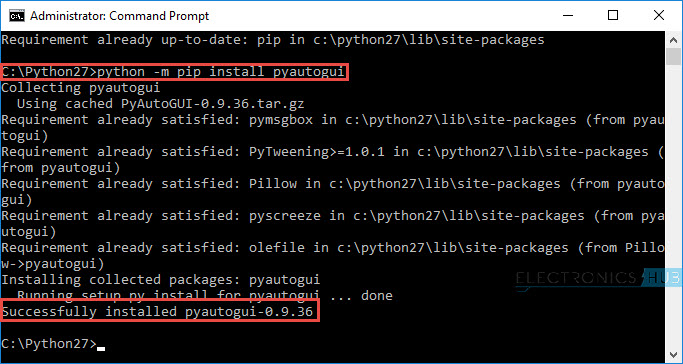
**INSTALLING PyAutoGUI**

The following steps will guide you through the installation of PyAutoGUI on Windows Computers. The module PyAutoGUI will help you to programmatically control the mouse and keyboard.

With the help of PyAutoGUI, we can write a Python Program to mimic the actions of mouse like left click, right click, scroll, etc. and keyboard like keypress, enter text, multiple key press, etc. without physically doing them. Let us install PyAutoGUI.

If you remember in the previous project, where we controlled an LED on Arduino using Python, we have installed Python in the directory “C:\Python27”.

Open Command Prompt with Administrator privileges and change to the directory where you have installed Python (in my case, it is C:\Python27).



In the Arduino Code, the Arduino sends out five different texts or commands through Serial Port upon detecting appropriate hand gestures. These commands are

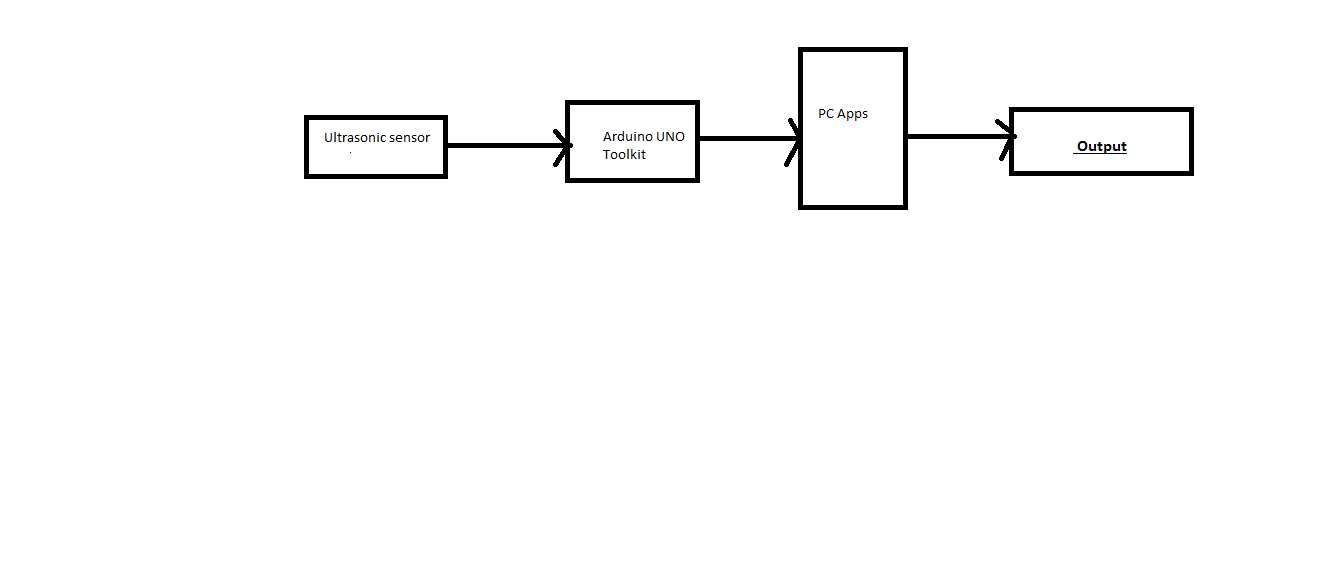
* Next
* Previous
* Down
* Up
* Change

Using these commands along with few functions in PyAutoGUI (like hotkey, scroll, keyDown, press and keyUp), you can write a simple Python Code that will execute the following tasks of keyboard and mouse.

* Data = “next” – – > Action = Ctrl+PgDn
* Data = “previous” – – > Action = Ctrl+PgUp
* Data = “down” – – > Action = Down Arrow
* Data = “up” – – > Action = Up Arrow
* Data = “change” – – > Action = Alt+Tab

**2.3**

**Block Diagram**

****

The above is a block diagram for showing the steps of execution of the hand gesture control system using arduino toolkit and a couple of ultrasonic sensors

**2.4.1**

**PROJECT ASSUMPTIONS**

An assumption is a belief of what you assume to be true in the future. You make assumptions based on your knowledge, experience or the information available on hand. These are anticipated events or circumstances that are expected to occur during your project’s life cycle.

Assumptions are supposed to be true but do not necessarily end up being true. Sometimes they may turn out to be false, which can affect your project significantly. They add risks to the project because they may or may not be true.

The assumptions made by us in our project is that we cover a lot of actions by mapping buttons to actions within a limited distance which is equal to 60 cm. This in turn sometimes leads to overlapping of actions giving it a slight accuracy issue.

**2.4.2**

**CONSTRAINTS**

Constraints are limitations imposed on the project, such as the limitation of cost, schedule, or resources, and you have to work within the boundaries restricted by these constraints. All projects have constraints, which are defined and identified at the beginning of the project.

Constraints are mainly of two types:

* **Business constraints**
* **Technical constraints**

In our project currently there are no business constraints since our project is not working on a global scale, so we have to deal with only technical constraints.

Our main technical constraints were:

* No previous knowledge of arduino
* No past project experience with arduino
* Ultrasonic sensors not working with full accuracy
* Project funding sources are limited, with no contingency.