

A TRAINING REPORT
ON
PYTHON AND ARDUINO TO CREATE A GESTURE
CONTROLLED VIDEO PLAYER

SUBMITTED BY:

HARSHUL JAIN

41218002817

ECE, 2017-21

SUBMITTED TO: DR KUMAR SHUBHAM
(Head Of Department ECE)



Affiliated to GGSIP University, New Delhi
Approved by AICTE & Council of Architecture

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING

DELHI TECHNICAL CAMPUS

(AFFILIATED TO GGSIP UNIVERSITY)

28/1, KNOWLEDGE PARK-III, GREATER NOIDA,U.P.

CONTENTS

1. What is Python
2. How to install Python on Windows
3. Introduction of Project
4. Components
 - a) Arduino
 - b) Ultrasonic Sensor
 - c) Pyserial
 - d) PyAutoGui
 - e) Arduino IDE
 - f) Python IDE
5. Logic of the Project
6. Observation and Working
7. Resources

WHAT IS PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Can be used for :

- 1) Back end or server side web and mobile development
- 2) Desktop app and software development
- 3) Processing big data and performing mathematical computations
- 4) Writing system scripts(creating instructions that tell a computer system to do something)

Python knowledge Required for this Project :

- a) Syntax - The syntax of the Python programming language is the set of rules which defines how a Python program will be written
- b) A Python program is divided into a number of logical lines and every logical line is terminated by the token NEWLINE. A logical line is created from one or more physical lines.
A line contains only spaces, tabs, form feeds possibly a comment, is known as a blank line, and Python interpreter ignores it.
- c) Comments - A comment begins with a hash character(#) which is not a part of the string literal and ends at the end of the physical line. All characters after the # character up to the end of the line are part of the comment and the Python interpreter ignores them
- d) When you want to write a long code in a single line you can break the logical line in two or more physical lines using backslash character(\). Therefore when a physical line ends with a backslash characters(\) and not a part of a string literal or comment then it can join another physical line
- e) You can write two separate statements into a single line using a semicolon (;) character between two line.

- f) Indentation - Python uses whitespace (spaces and tabs) to define program blocks whereas other languages like C, C++ use braces ({}) to indicate blocks of codes for class, functions or flow control. The number of whitespaces (spaces and tabs) in the indentation is not fixed, but all statements within the block must be the indented same amount.

- g) Reserve Words -

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	el	if	or	yield
assert	else	import	pass	
break	except	in	raise	

- h) Print() function - The print statement has been replaced with a print() function, with keyword arguments to replace most of the special syntax of the old print statement.
- i) Variables - A variable is a memory location where a programmer can store a value
- j) Value is either string, numeric etc
- k) Int()- stores integer value
- l) Float()- stores decimal value
- m) Str()- stores string
- n) We can convert between different data types by using different type conversion functions like int(), float(), str(), etc.
- o) Loops- loops used in this are while and if, while runs continuously if the condition is true, whereas if loop runs once if the condition is valid and it rechecks, it will run if it is true
- p) To use libraries we need to install it in computer and then run it by importing it using import function
- q) Libraries Used
- 1) PyAutoGUI
PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications
 - 2) PySerial
This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython.

HOW TO INSTALL PYTHON ON WINDOWS

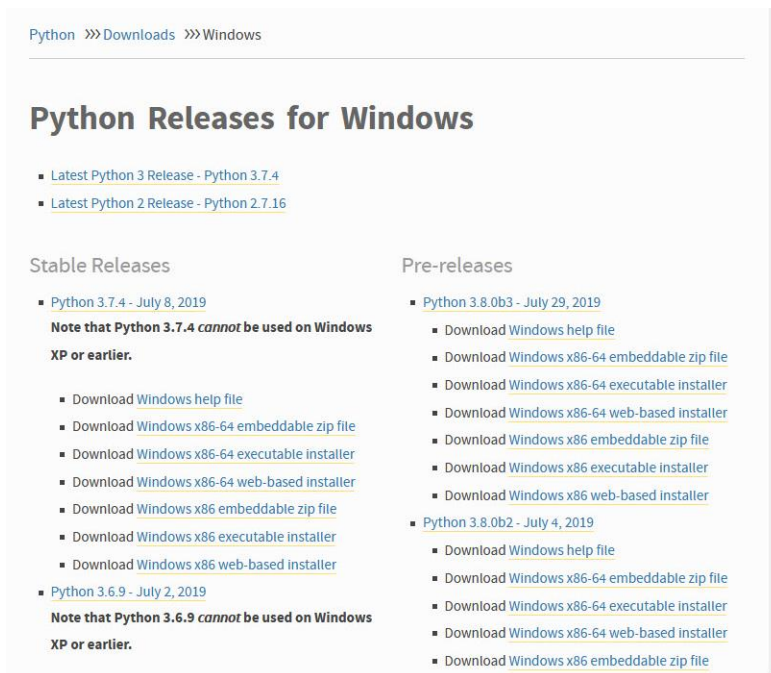
Python download requires about 25 Mb of disk space; keep it on your machine, in case you need to re-install Python. When installed, Python requires about an additional 90 Mb of disk space.

Downloading Python

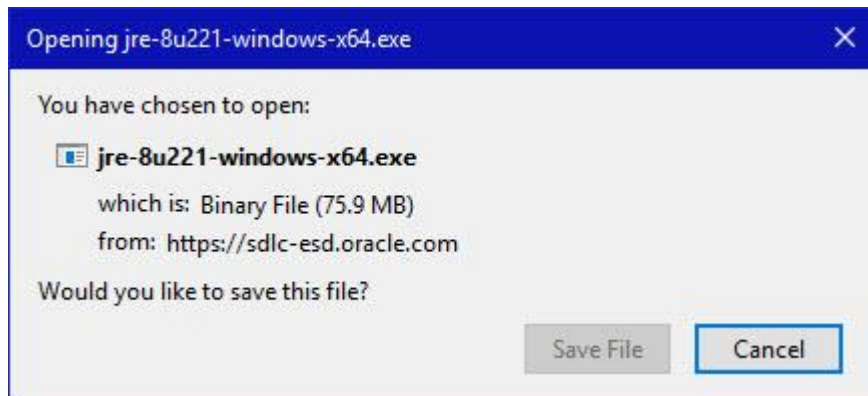
- 1) Visit <https://www.python.org/downloads/>



- 2) Click the **Windows** link (two lines below the **Download Python 3.7.4** button)



- 3) Click on the **Download Windows x86-64 executable installer** link under the top-left **Stable Releases**.
- 4) The following pop-up window titled **Opening python-3.7.4-amd64.exe** will appear



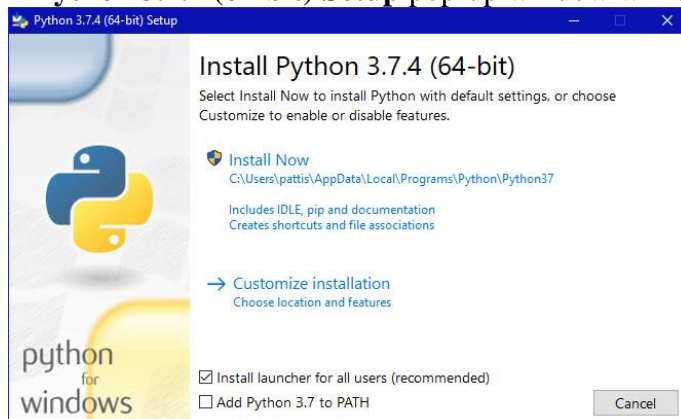
- 5) Click the **Save File** button.
- 6) The file named **python-3.7.4-amd64.exe** should start downloading into your standard download folder. This file is about 30 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem).
- 7) The file should appear as



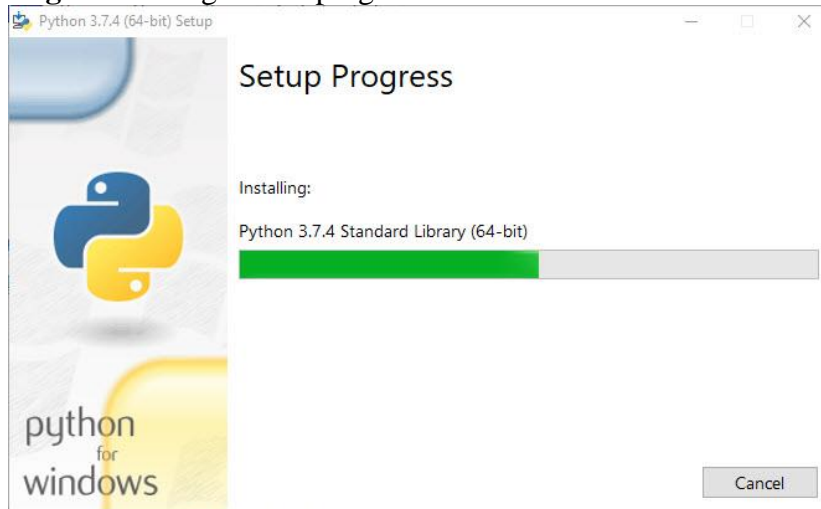
- 8) Move this file to a more permanent location, so that you can install Python (and reinstall it easily later, if necessary).
- 9) Feel free to explore this webpage further; if you want to just continue the installation, you can terminate the tab browsing this webpage.
- 10) Start the **Installing** instructions directly below.

Installing Python

- 1) Double-click the icon labeling the file **python-3.7.4-amd64.exe**.
- 2) A **Python 3.7.4 (64-bit) Setup** pop-up window will appear.



- 3) Ensure that the **Install launcher for all users (recommended)** and the **Add Python 3.7 to PATH** checkboxes at the bottom are checked.
- 4) If the Python Installer finds an earlier version of Python installed on your computer, the **Install Now** message may instead appear as **Upgrade Now** (and the checkboxes will not appear).
- 5) Highlight the **Install Now** (or **Upgrade Now**) message, and then click it.
- 6) When run, a **User Account Control** pop-up window may appear on your screen. I could not capture its image, but it asks, **Do you want to allow this app to make changes to your device**.
- 7) Click the **Yes** button.
- 8) A new **Python 3.7.4 (64-bit) Setup** pop-up window will appear with a **Setup Progress** message and a progress bar



- 9) During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new **Python 3.7.4 (64-bit) Setup** pop-up window will appear with a **Setup was successfully** message



- 10) Click the **Close** button. Python should now be installed.

INTRODUCTION OF PROJECT

The Project is about using python and arduino Using HC SR04 ultrasonic sensor making a gesture controlled device through which we can perform specific tasks such as volume increase or decrease or forward or rewind in VLC video player using software like pyautogui, pyserial

Python can be used for writing scripts here we use pyserial to communicate with serial port on which our arduino will be giving outputs, it will be read using pyserial and the pyautogui will give outputs to computer in terms of keys of mouse and keyboard which computer will execute and the VLC volume can be increased or decreased, and the video can be forwarded or rewinded, ie automation using gestures which we do near ultrasonic sensor

COMPONENTS USED

Hardware :

1.	Computer
2.	Arduino Microcontroller
3.	Ultrasonic sensor HC SR04 Quantity- 2

Software :

1	Python IDE
2	Arduino IDE
3	PyAutoGui
4	PySerial

COMPONENTS

ARDUINO MICROCONTROLLER:

The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, 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 a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g



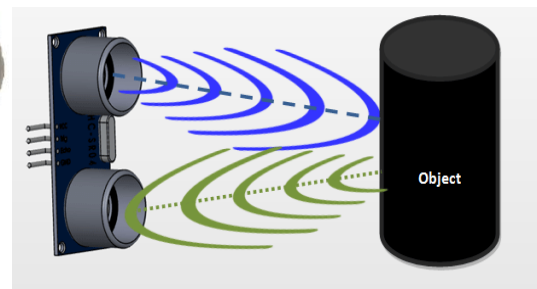
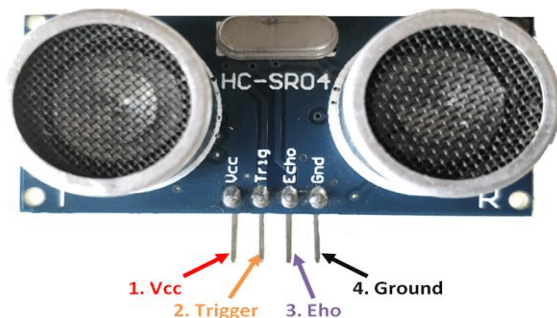
ULTRASONIC SENSOR HC SR04:

HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

$$\text{Distance} = \text{Speed} \times \text{Time}$$

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module

Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.




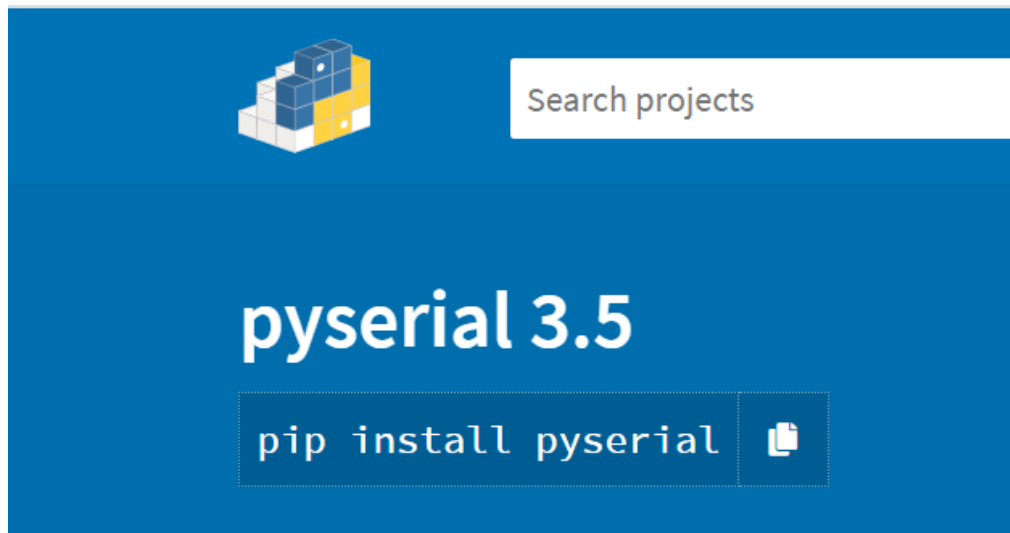
Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

PySerial:

- This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named “serial” automatically selects the appropriate backend.
- PySerial is a library which provides support for serial connections over a variety of different devices

How To Install PySerial:

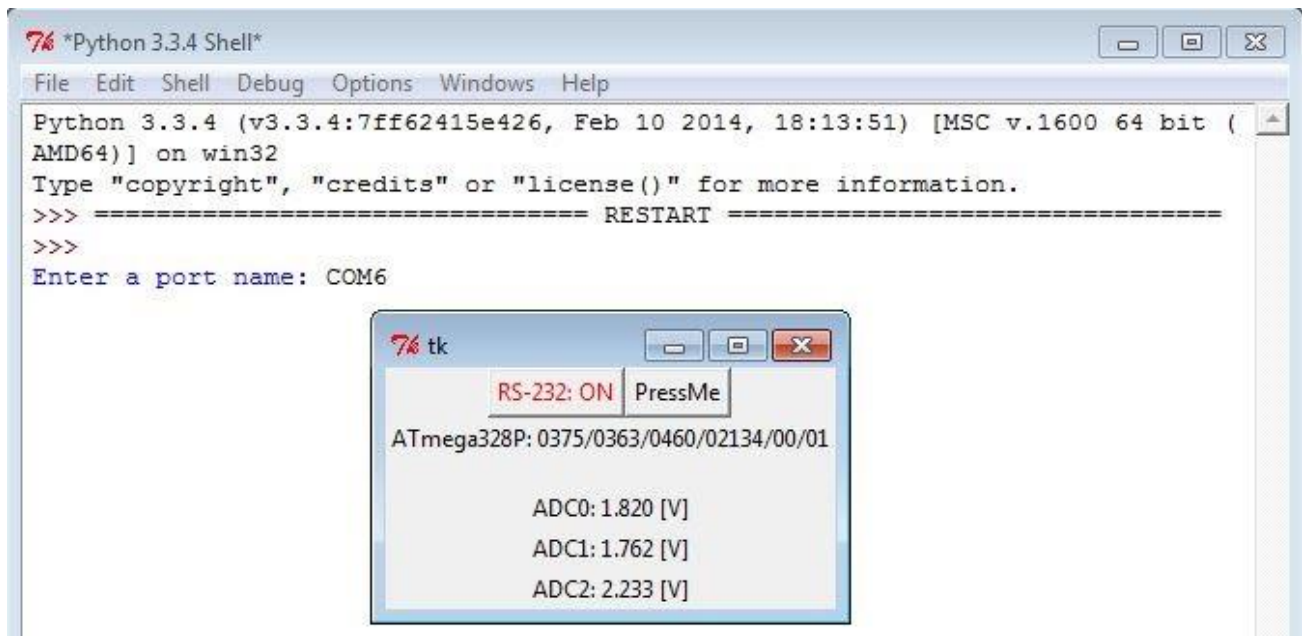
 pypi.org/project/pyserial/



By Typing this command in Terminal ^

Code :

Import serial (in python shell)

A screenshot of a Python 3.3.4 Shell window. The window title is "Python 3.3.4 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The main text area shows the following output:

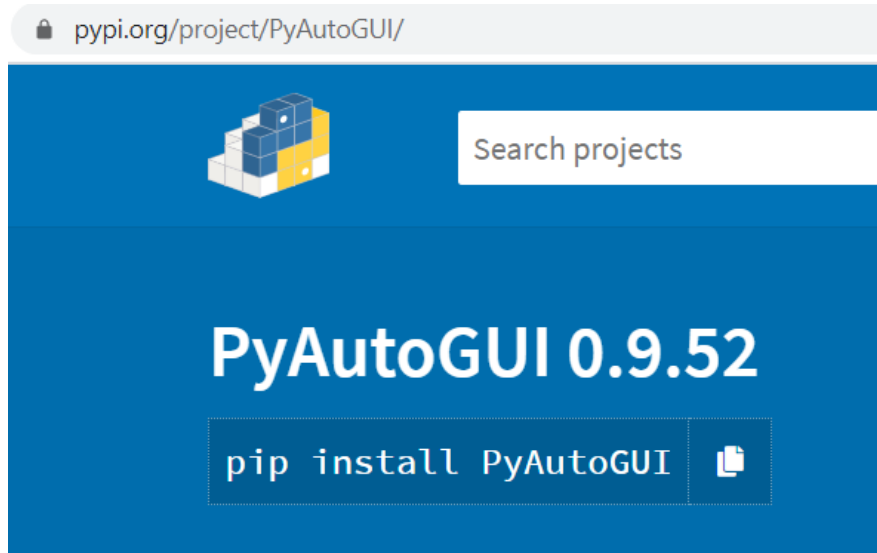
```
Python 3.3.4 (v3.3.4:7ff62415e426, Feb 10 2014, 18:13:51) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter a port name: COM6
```

An inset window titled "tk" is shown in the foreground. It contains a label "RS-232: ON" in red, a button labeled "PressMe", and the text "ATmega328P: 0375/0363/0460/02134/00/01". Below this, three lines of ADC readings are displayed: "ADC0: 1.820 [V]", "ADC1: 1.762 [V]", and "ADC2: 2.233 [V]".

PyAutoGUI:

- **PyAutoGUI** lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be as simple. **PyAutoGUI** works on Windows, macOS, and Linux, and runs on Python 2 and 3.

How to Install:



By Typing this command in Terminal ^

Code :

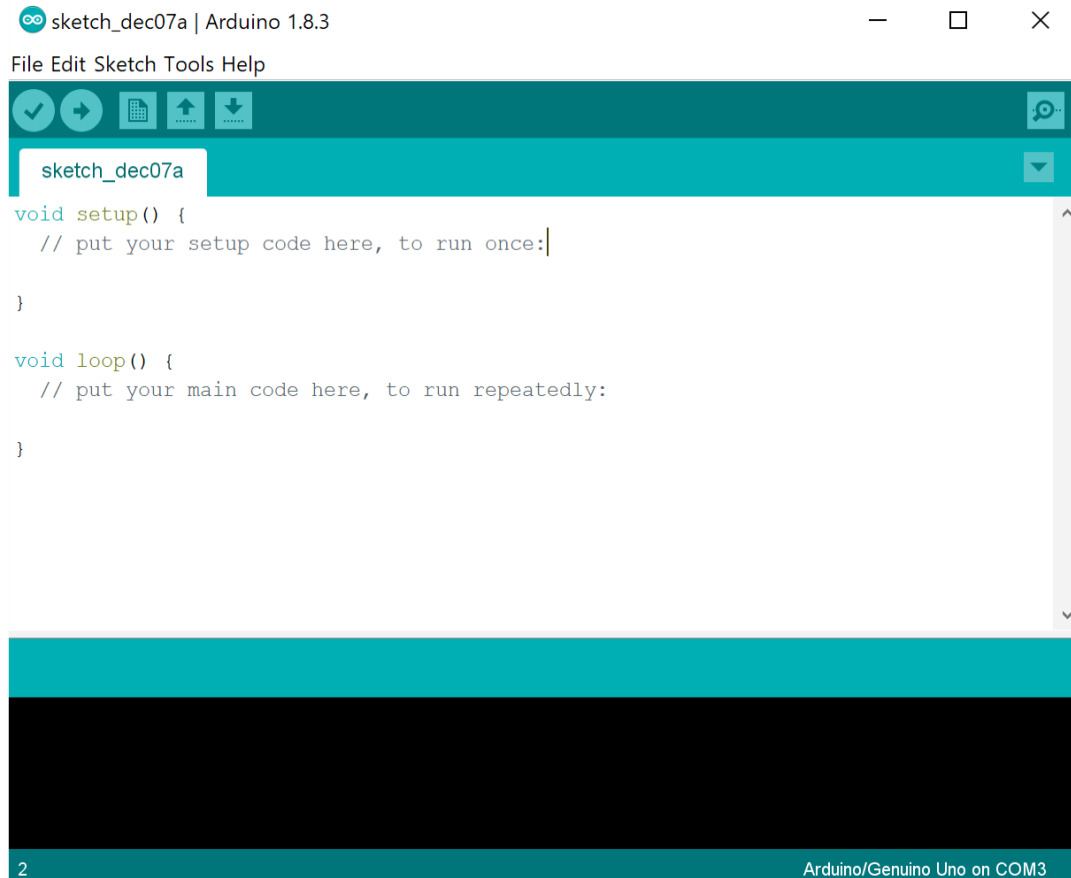
Import pyautogui (in python shell)

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import pyautogui as pg
>>> pg.size()
(1366, 768)
>>> pg.position()
(907, 35)
>>> pg.position()
(0, 0)
>>> pg.position()
(1365, 0)
>>> pg.position()
(1365, 767)
>>> pg.position()
(911, 31)
>>> pg.moveTo(907, 35, 4)
>>> |
```



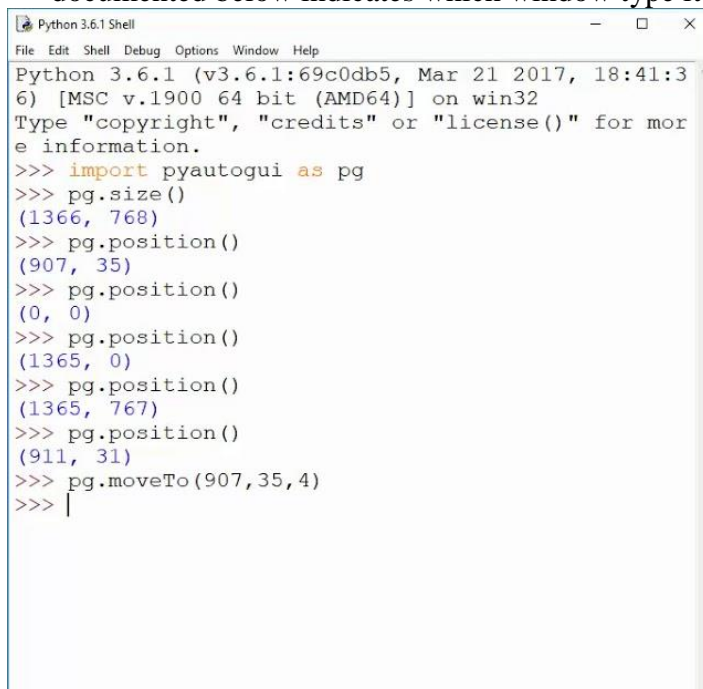
ARDUINO IDE:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.



PYTHON IDLE:

IDLE is Python's Integrated Development and Learning Environment, IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously. On Windows and Linux, each has its own top menu. Each menu documented below indicates which window type it is associated with.



LOGIC OF PROJECT

- In this program we use 2 ultrasonic sensors through which we can determine the distance of the hands the left ultrasonic sensor helps us determine if we want to move forward or rewind the movie
- Where the distance of hands determines the action which is to be performed, if the distance increases from 20 and keeps on increasing then hand is moving away, If the distance keeps decreasing then hand is moving towards the sensor

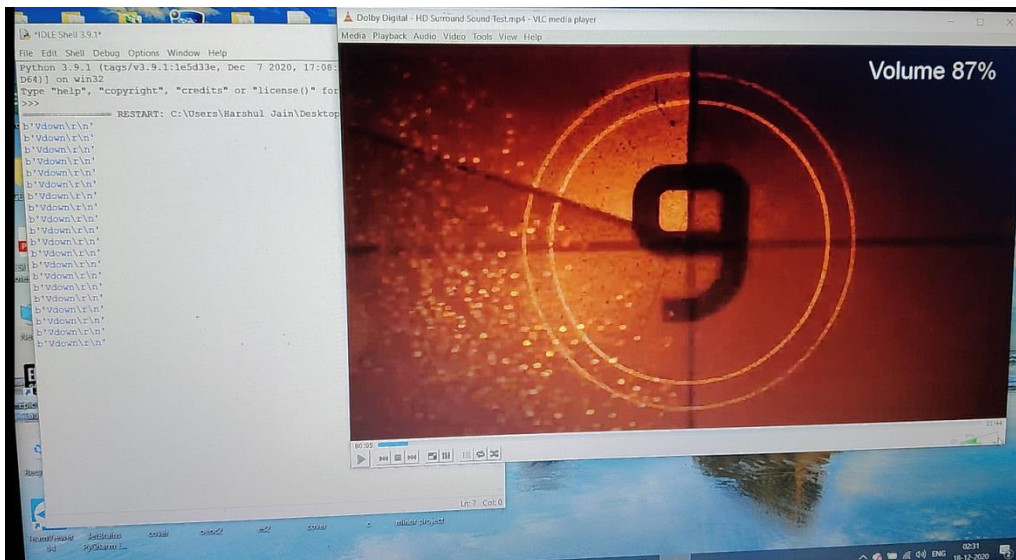
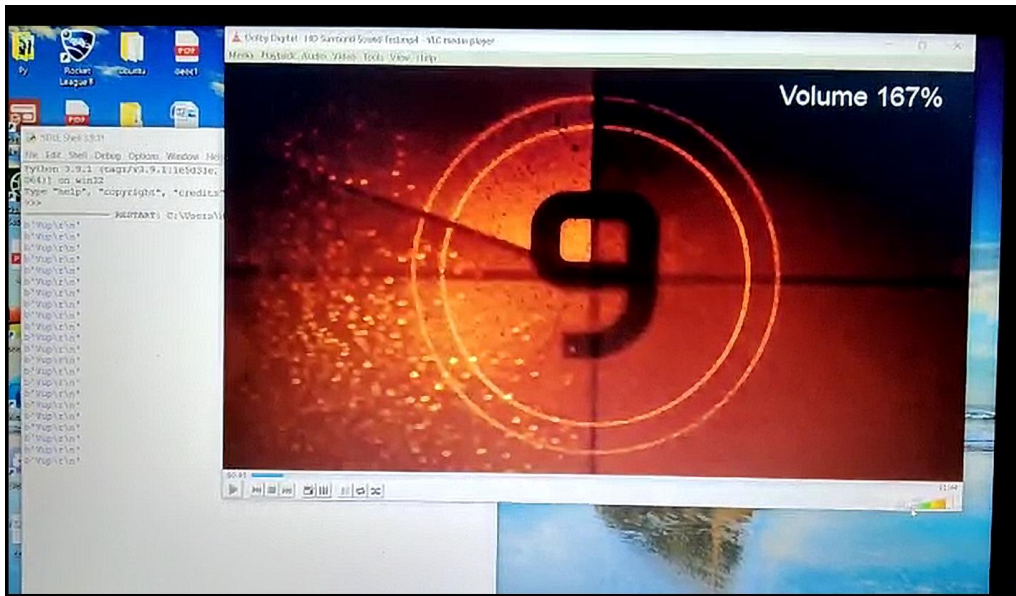
Sensor	Distance	Meaning
Left	D<10(hands pulled in)	Forward
	D>20(hands pulled out)	Rewind
Right	D<10(hands pulled in)	Volume Up
	D>20(hands pulled out)	Volume down

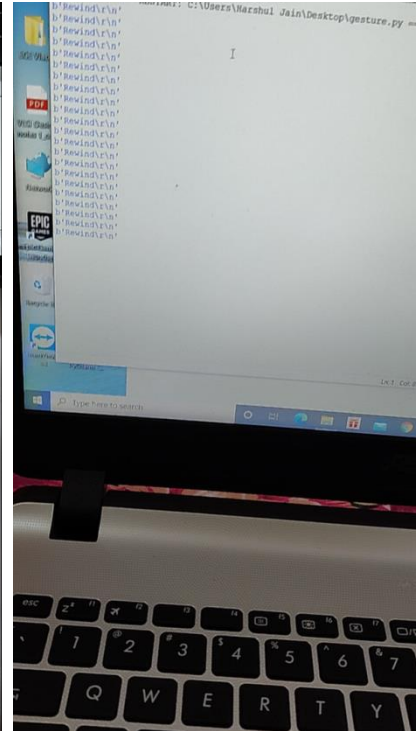
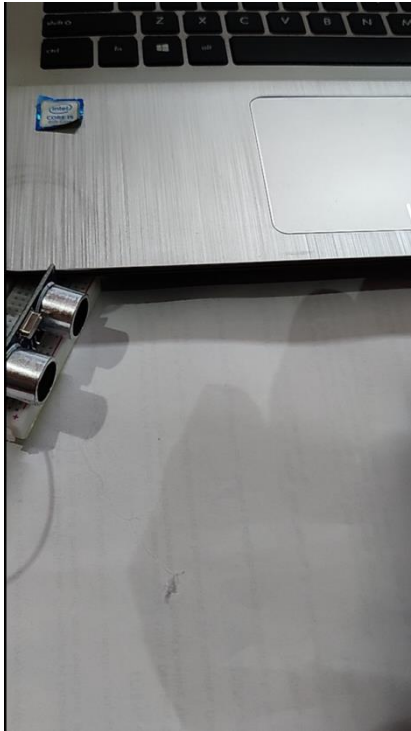
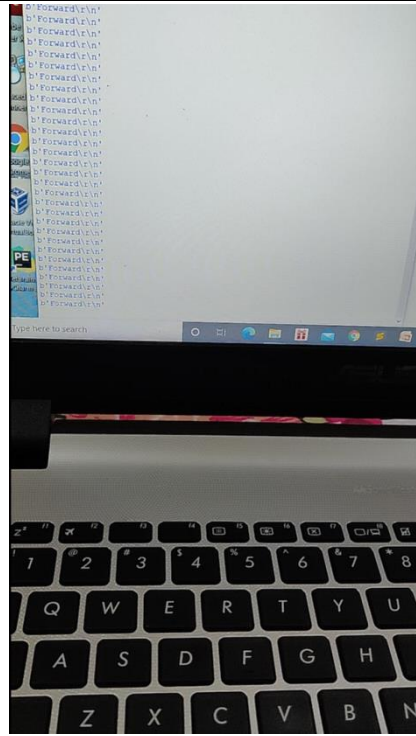
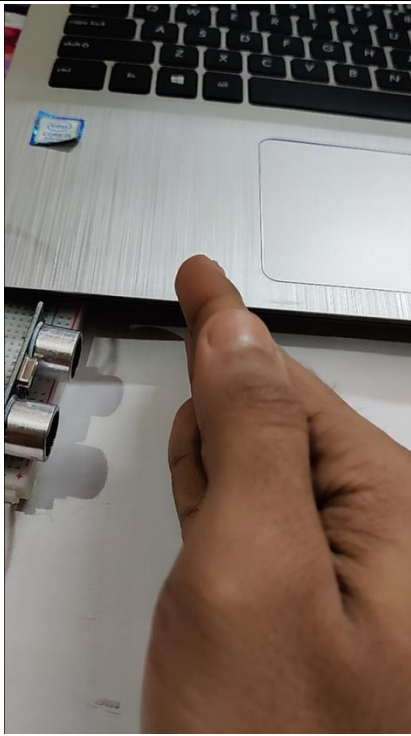
- By importing pyserial and pyautogui, we check the output at COM port of the system, the pyserial reads it if the output is same as one mentioned in our program then it will give command that those keys are pressed using pyautogui.typeWrite function

OUTPUT & KEYS

S.No	OUTPUT	KEY
1	Play/Pause	Space
2	Forward	Control + Right
3	Rewind	Control + Left
4	Volume Up	Control + Up
5	Volume Down	Control + Down

WORKING





RESOURCES

Links:

<https://docs.python.org/3/library/idle.html>

<https://support.microsoft.com/en-us/office/uploading-board-code-and-arduino-ide-a9723765-1314-49e0-a69b-bb5c3e1f628d>

<https://pypi.org/project/PyAutoGUI/>

<https://pyautogui.readthedocs.org>

<https://pypi.org/project/pyserial/>

<https://www.w3resource.com/python/python-data-type.php>

https://www.tutorialspoint.com/python/python_basic_syntax.htm

<http://pythonhosted.org/pyserial/>

<https://pypi.python.org/pypi/pyserial>

<https://components101.com/ultrasonic-sensor-working-pinout-datasheet>

https://www.w3schools.com/python/python_variables.asp

<https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/python.html>

<https://sites.google.com/site/greenmechatronics/code-garage/rs-232-pyserial-in-python>

<https://www.programiz.com/python-programming/variables-datatypes>