An Industry-Oriented Mini Project Report

On

ARDUINO-BASED HAND GESTURE CONTROL OF A PC

A Dissertation Submitted in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

K R RACHANA **16BD1A0442**

Under the esteemed guidance of
Mr. K YAKAIAH
Assistant Professor
Department of ECE

Department of Electronics and Communication Engineering

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by A.I.C.T.E Affiliated to JNTUH, Hyderabad

NARAYANAGUDA, HYDERABAD, TELANGANA-500029

2019-2020

CERTIFICATE

This is to certify that the project entitled **"ARDUINO BASED HAND GESTURE CONTROL OF A PC"** being submitted by **Ms. K R RACHANA (16BD1A0442)**student of **Keshav Memorial Institute of Technology, JNTUH** in partial fulfillment of requirements for the award of **"Bachelor of Technology"** in **"Electronics and Communication Engineering"** as specialization is a record of Bonafede work carried out by them under my guidance and supervision in the academic year 2019-2020.

SIGNATURE	**SIGNATURE**
Mr. K YAKAIAH	**Dr. S. JULIAN SAVARI ANTONY,**
PROJECT GUIDE,	**HEAD OF THE DEPARTMENT**
Assistant Professor,	Department of ECE,
Department of ECE,	Keshav Memorial Institute of Technology,
Keshav Memorial Institute of Technology,	Hyderabad – 500029.
Hyderabad - 500029.	

Submitted for the Project Viva Voce Examination held on ………………….

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work.

We render our thanks to **Dr. Maheshwar Dutta,** B.E, MTech., Ph.D., Principal who encouraged us to do the Project.

We are grateful to **Mr. NeilGogte,** Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. Nitin Kumar**, Director, and **Mrs. Deepa Abhishek Ganu,** Director for providing an excellent environment in the college.

We are also thankful to **Dr. S. Julian Savari Antony**, B.E, M.B.A, M.E, Ph.D., and Head of Department for providing us with its time and amenities to make this project a success within the given schedule.

We are also thankful to **Mr. K Yakaiah,** Assistant Professor, Department of ECE, as our Project Guide for his valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire **ECE Department** faculty who helped us directly and indirectly in the completion of the project.

K R RACHANA **16BD1A0442**

DECLARATION

I,

Hereby declare that the project report entitled **“ARDUINO BASED HAND GESTURE CONTROL OF A PC”** is done in partial fulfilment for the award of the Degree of Bachelor of Technology in Electronics and Communication Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. We have not submitted this report to any other University or organization for award of any other degree.

STUDENT NAME	**ROLL.NO**
KR RACHANA	**16BD1A0442**

INDEX

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Project Name: ARDUINO BASED HAND GESTURE CONTROL OF A PC

We normally use LED Indicators, Switches, Touch Screens and LCD Displays as a part of HMI devices to communicate with machines. 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, increase/decrease volume, 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.

CHAPTER 1

INTRODUCTION

1.1 Introduction

You might have seen Hand Gesture Controlled Robots, where the motion of a robot is controlled by the gestures of the hand. Another interesting project based on a similar principle is an Arduino based Hand Gesture Control of your computer or laptop. 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.

Humans interact in the physical world by the means of the five senses. However, gestures have been an important means of communication in the physical world from ancient times, even before the invention of any language. In this era of machines taking control of every complex works, interactions with machines have become more important than ever. Since this project deals with gesture-controlled PC, the primary focus will be on the use of hand gestures for specific applications only. 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. The project Arduino based Hand Gesture Control of Computer is implemented using Python.

There are several ways to capture a human gesture that a computer would be able to recognize. The gesture can be captured using distance measurement, camera, or a data glove. Gestures can also be captured via Bluetooth or infrared waves, Acoustic, Tactile, optical or motion technological means. The embedded systems designed for specific control functions can be optimized to reduce the size and cost of the device and increase the reliability and performance.

1.2 Principle

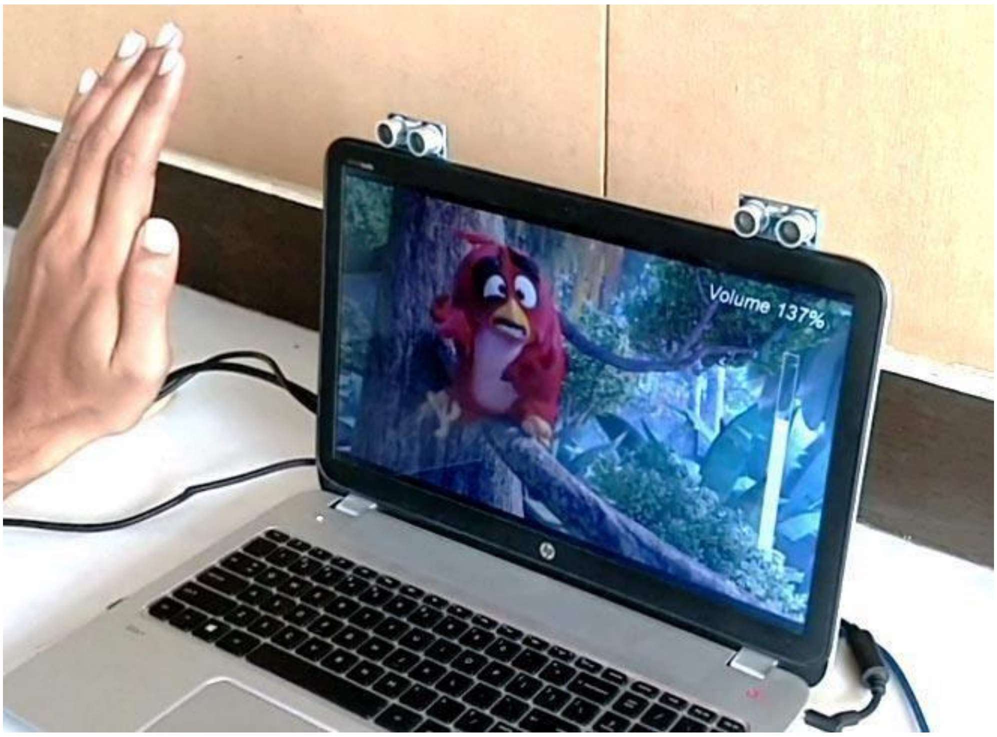


Figure 1.1 Hand Gesture Control of a Laptop

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 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 distance information from Arduino is collected by a Python Program and a special library called PyAutoGUI will convert the data into keyboard click actions.

CHAPTER 2
HARDWARE COMPONENTS

2.1 Arduino

2.1.1 What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

2.1.2 Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

2.1.3 Arduino UNO

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

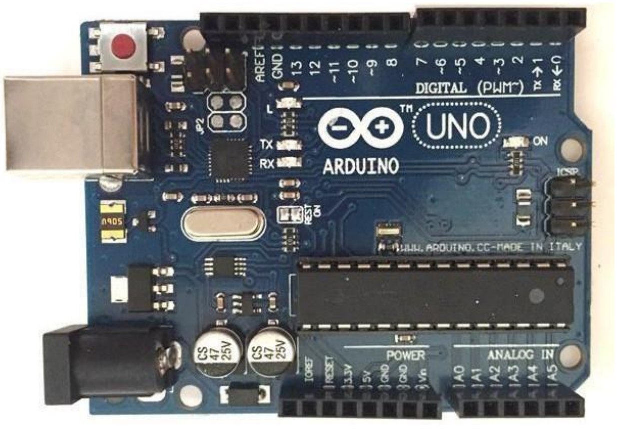


Figure 2.1 Arduino UNO

2.1.4 Automatic Software Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano-farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip.

This setup has other implications. When the Uno is connected to a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

2.1.5 Pin Description of Arduino UNO

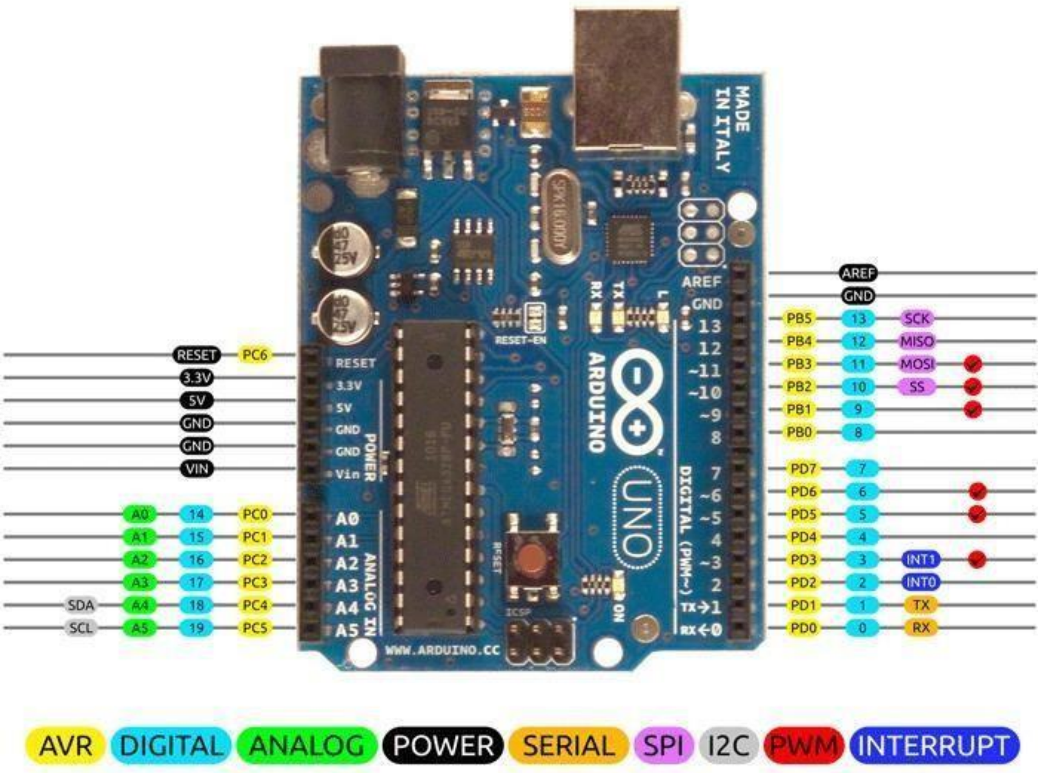


Figure 2.2 Pin description of Arduino UNO

Table 2.1 Pin description of Arduino UNO

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/ Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

2.1.6 Technical Specifications

Table 2.2 Technical Specifications of Arduino UNO

Microcontroller	**ATmega328P – 8-bit AVR family microcontroller**
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency(Clock Speed)	16 MHz

2.1.7 Applications of Arduino UNO

The applications of Arduino Uno include the following.

- Arduino Uno is used in Do-it-Yourself projects prototyping.
- In developing projects based on code-based control
- Development of Automation System
- Designing of basic circuit designs.
- Robotics
- Internet of things

2.1.8 Other Arduino Boards

There are several Arduino boards. Some of them are

- Arduino NANO
- Arduino RS232
- Arduino Mega
- Arduino Lilypad
- Arduino Pro
- Arduino Leonardo

2.1.9 Arduino Connection

First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

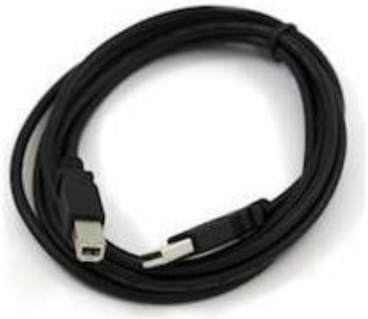


Figure 2.3 Arduino USB Cable

2.2 Ultrasonic Sensors

2.2.1 What is an Ultrasonic Sensor?

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

HC-SR04 is an ultrasonic sensor mainly used to determine the distance of the target object. It measures accurate distance using a non-contact technology - A technology that involves no physical contact between sensor and object.

Transmitter and receiver are two main parts of the sensor where former converts an electrical signal to ultrasonic waves while later converts that ultrasonic signals back to electrical signals. These ultrasonic waves are nothing but sound signals that can be measured and displayed at the receiving end.

It gives precise measurement details and comes with accuracy (resolution) around 3mm, terming there might be a slight difference in the calculated distance from the object and the actual distance.

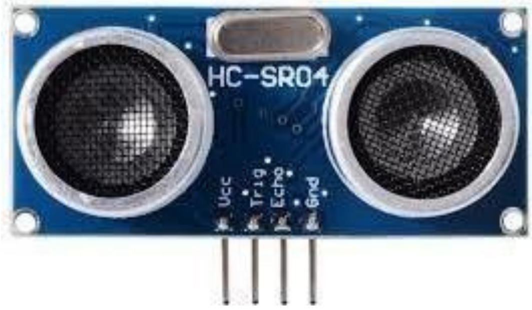


Figure 2.4 Ultrasonic Sensor

Table 2.3 Main Features of Ultrasonic Sensor

Parameter	**Value**
Main Parts	Transmitter & Receiver
Technology Used	Non-Contact Technology
Operating Voltage	5 V
Operating Frequency	4 MHz
Detection Range	2cm to 400cm
Measuring Angle	30º
Resolution	3mm
Operating Current	<15mA
Sensor Dimensions	45mm x 20mm x 15mm

2.2.2 Ultrasonic Sensor Pinout & Description

Ultrasonic Sensor contains 4 pins in total. I have labelled these HC-SR04 Pinout in below figure for better visualization:

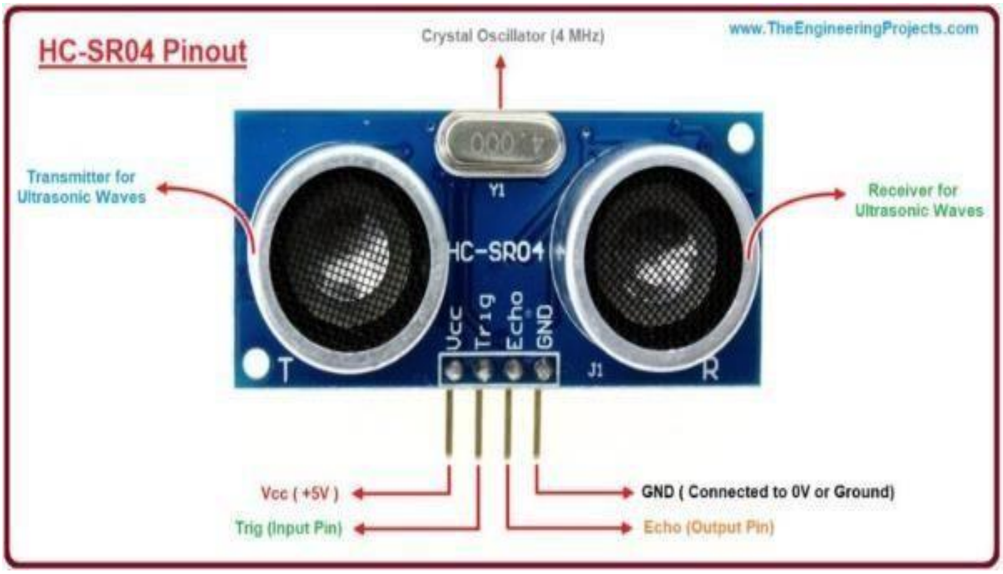

HC-SR04 Pinout
Crystal Oscillator (4 MHz)
www.TheEngineeringProjects.com
Transmitter for Ultrasonic Waves
Receiver for Ultrasonic Waves
4.000
Y1
HC-SR04
Vcc
Trig
Echo
GND
T
J1
R
Vcc (+5V)
Trig (Input Pin)
GND (Connected to 0V or Ground)
Echo (Output Pin)

Figure 2.5 Ultrasonic Sensor Pinout

Table 2.4 Pin Description of Ultrasonic Sensor

No.	**Pin Name**	**Pin Description**
1	VCC	The power supply pin of the sensor that mainly operates at 5V DC.
2	Trig Pin	It plays a vital role to initialize measurement for sending ultrasonic waves. It should be kept high for 10us for triggering the measurement.
3	Echo Pin	This pin remains high for short period based on the time taken by the ultrasonic waves to bounce back to the receiving end.
4	Ground	This pin is connected to ground.

2.2.2.1 Crystal Oscillator

A crystal oscillator is an electronic oscillator circuit which is used for the mechanical resonance of a vibrating crystal of piezoelectric material. It will create an electrical signal with a given frequency. This frequency is commonly used to keep track of time for example: wrist watches are used in digital integrated circuits to provide a stable clock signal and also used to stabilize frequencies for radio transmitters and receivers. Quartz crystal is mainly used in radio-frequency (RF) oscillators. Quartz crystal is the most common type of piezoelectric resonator, in oscillator circuits we are using them so it became known as crystal oscillators. Crystal oscillators must be designed to provide a load capacitance.

Figure 2.6 Crystal Oscillator

Generally bipolar transistors or FETs are used in construction of Crystal oscillator circuits. This is because operational amplifiers can be used in different low frequency oscillator circuits which are below 100KHz but operational amplifiers do not have the bandwidth to operate.

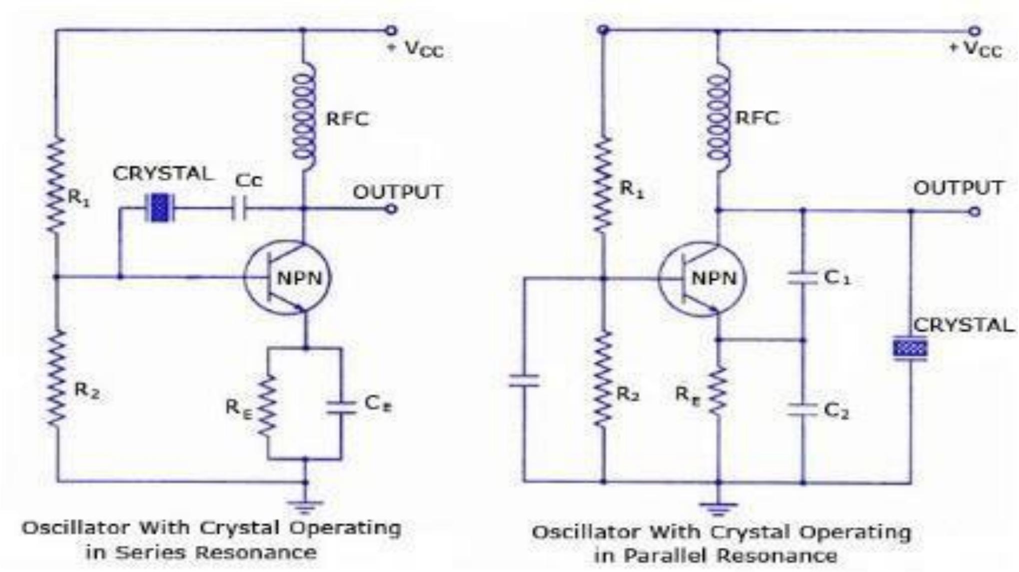

+ Vcc
RFC
CRYSTAL
Cc
OUTPUT
R_1
NPN
R_2
R_E
C_E
Oscillator With Crystal Operating in Series Resonance
+ Vcc
RFC
R_1
OUTPUT
NPN
C_1
CRYSTAL
R_2
R_E
C_2
Oscillator With Crystal Operating in Parallel Resonance

Figure 2.7 Crystal Oscillator Circuit Diagram

Usually quartz crystal oscillators are highly stable, consists of good quality factor(Q), they are small in size, and are economically related. Hence, quartz crystal oscillator circuits are more superior compared to other resonators like LC circuits, turning forks. Generally, in Microprocessors and Micro controllers we are using an 8MHz crystal oscillator.

2.2.2.2 Ultrasonic transmitter circuit

The transmitter circuit is built around two CD4017 decade counter ICs (IC1 and IC2), D-type flip-flop IC CD4013 (IC3) and a few discrete components. The arrangement generates stable 40kHz signals, which are transmitted by transducer TX.

The crystal-controlled radio-frequency (RF) oscillator built around transistor T 1 (BC549) generates an 8MHz signal, which serves as input to the first decade counter built around IC1. The decade counter divides the oscillator frequency to 800 kHz. The output of IC1 is fed to the second CD4017 decade counter (IC2), which further divides the frequency to 80 kHz.

The flip-flop (IC3) divides 80kHz signal by 2 to give 40kHz signal, which is transmitted by ultrasonic transducer TX.

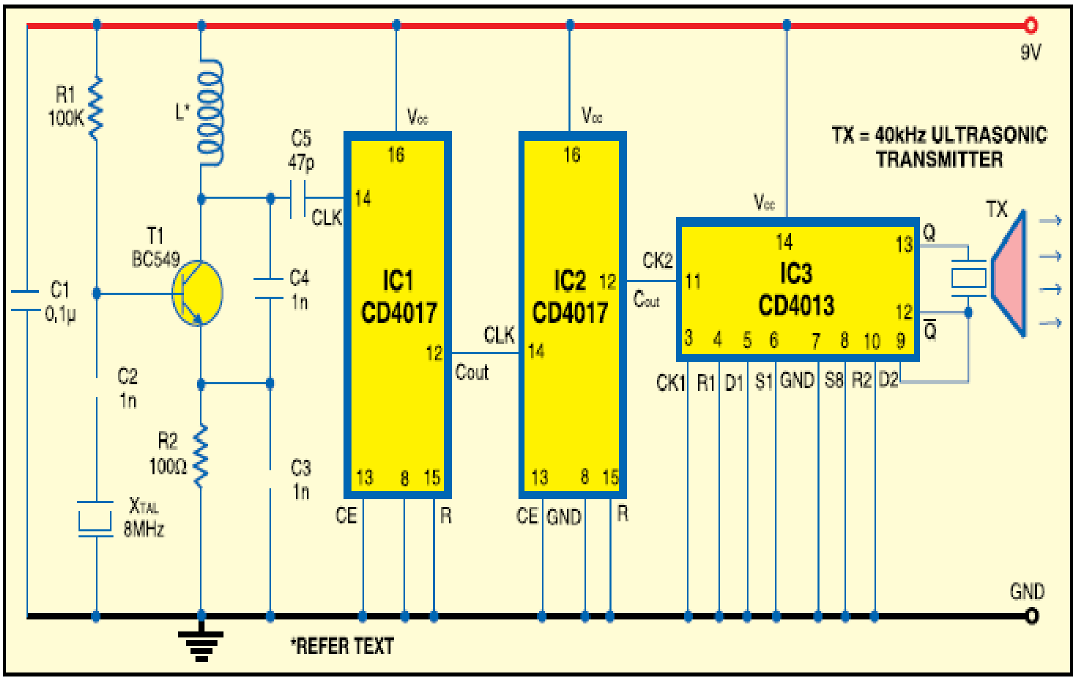


Figure 2.8 Ultrasonic transmitter circuit

Coil L is made with 36SWG enameled copper wire that is wound 15 times around an 8mm-diameter plastic former as used for radio oscillators, which has a ferrite bead. The transmitter circuit works off 9-12V DC.

2.2.2.3 Ultrasonic receiver circuit

The receiver circuit (Fig.2) is built around a single decade counter CD4017 (IC4) and a few discrete components. To check the working of the transmitter, it is necessary to down-convert the 40kHz signal into 4kHz to bring it in the audible range.

By using the receiver, the 40kHz ultrasonic transmitter can be tested quickly. The receiver's transducer unit (RX) is kept near the ultrasonic transmitter under test. It detects the transmitted 40kHz signal, which is amplified by the amplifier built around transistor BC549 (T2). The amplified signal is fed to decade counter IC4, which divides the frequency to 4 kHz. Transistor T3 (SL100) amplifies the 4kHz signal to drive the speaker. Use a 9V PP3 battery to power the receiver circuit.

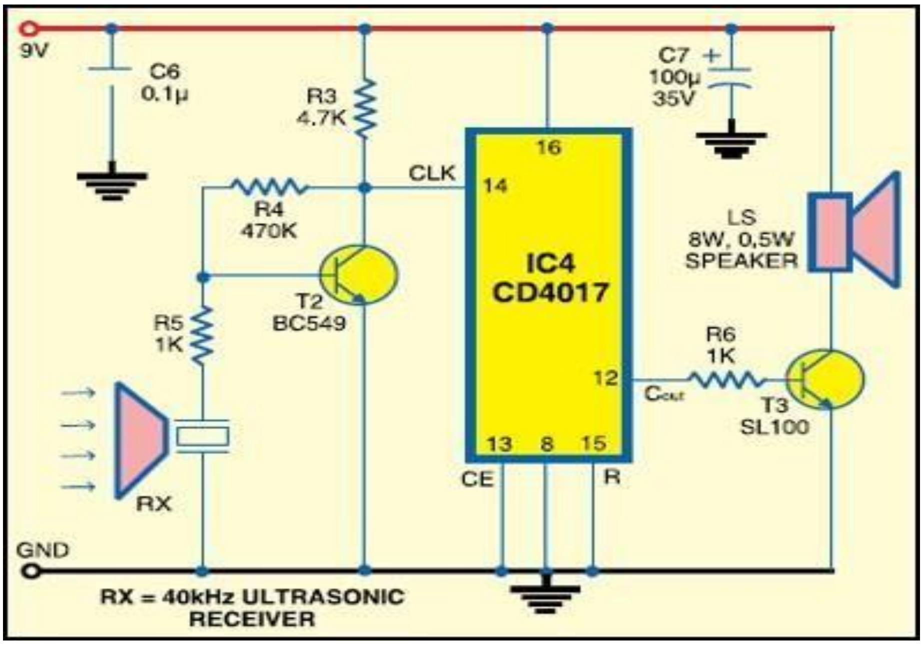


Figure 2.9 Ultrasonic receiver circuit

House the transmitter and receiver circuits in separate small cabinets. If the 40kHz transducer under test is working, the receiver circuit produces audible whistling sound

2.2.3 Working of Ultrasonic Sensor

The HC-SR04 Ultrasonic (US) sensor is an ultrasonic transducer that comes with 4 pin interfaces named as Vcc, Trigger, Echo, and Ground. It is very useful for accurate distance measurement of the target object and mainly works on the sound waves.

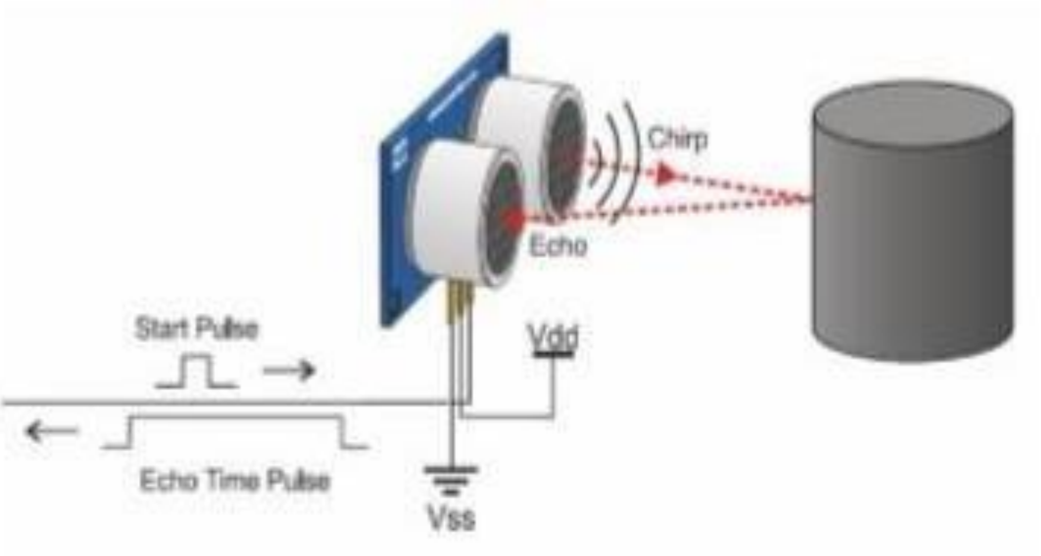


Figure 2.10 Working of ultrasonic sensor

A crystal oscillator is an electronic oscillator circuit which is used for the mechanical resonance of a vibrating crystal of piezoelectric material. It will create an electrical signal with a given frequency.

As we connect the module to 5V and initialize the input pin, it starts transmitting the sound waves which then travel through the air and hit the required object. These waves hit and bounce back from the object and then collected by the receiver of the module.

Distance is directly proportional to the time these waves require to come back at the receiving end. The more the time taken, more the distance will be. The waves will be generating if the Trig pin is kept High for 10 µs. These waves will travel at the speed of sound, creating 8 cycle sonic burst that will be collected in the Echo pin.

The echo pin remains turned on for the time these waves take to travel and bounce back to the receiving end. This sensor is mainly incorporated with Arduino to measure the required distance.

Following formula is used to calculate the distance of the object.

$$S = (V \times t)/2$$

Where S is the required distance,

V is the speed of sound and

t is the time sound waves take to come back after hitting the object.

We need to divide the value by 2 because time will be double as the waves travel and bounce back from the initial point. Dividing it by 2 will give the actual distance of the target object.

2.2.4 Interfacing Ultrasonic Sensor with Arduino

In order to get the precise distance measurement, HC-SR04 is mostly used in combination with different Arduino Modules like Arduino Uno and Arduino Mega. You can connect Arduino with this sensor in the following way.

First, you need to power up the sensor using 5V DC regulated input to the sensor. Connect the ground pin with the ground of the voltage source. You can also power the sensor module using the Arduino 5V pins as the current drawn by the sensor is less than 15mA, won’t be affecting the current ratings of the Arduino Module.

After setting up the initial arrangement, connect both Trig and Echo pins to the I/O pins of the Arduino Board. As mentioned earlier, in order to initialize the measurement process, the Trig pin must be kept high for 10us in the start. The sensor module will start generating sound waves with the frequency around 40,000 Hz per second from the transmitter.

As the waves bounce back, consequently, the Echo pin will turn on until the sound waves are received by the receiver. This time will be calculated using Arduino Module.

2.2.5 Applications

HC-SR04 comes with a wide range of applications mainly targeting distance and direction measurements. Some of the major applications are Speed and direction measurement, Wireless charging, Humidifiers, Medical ultrasonography, Burglar alarms, Embedded system, Depth measurement, Non-destructive testing.

2.3 Personal Computer

A personal computer (PC) is a multi-purpose computer whose size, capabilities, and price make it feasible for individual use. Personal computers are intended to be operated directly by an end user, rather than by a computer expert or technician. Every PC is dependent on microprocessor technology, which allows PC makers to set the entire central processing unit (CPU) on a single chip. Unlike large costly minicomputer and mainframes, time-sharing by many people at the same time is not used with personal computers. In this project we use a laptop.

2.3.1 Laptop

A laptop computer (also shortened to just laptop; or called a notebook computer) is a small, portable personal computer (PC) with a "clamshell" form factor, typically having a thin LCD or LED computer screen mounted on the inside of the upper lid of the clamshell and an alphanumeric keyboard on the inside of the lower lid. The clamshell is opened up to use the computer. Laptops are folded shut for transportation, and thus are suitable for mobile use. Its name comes from lap, as it was deemed to be placed on a person's lap when being used. Although originally there was a distinction between laptops and notebooks (the former being bigger and heavier than the latter), as of 2014, there is often no longer any difference. Laptops are commonly used in a variety of settings, such as at work, in education, for playing games, Internet surfing, for personal multimedia, and general home computer use.

Figure 2.11 Laptop

2.4 Jumper Wires

Jump wires (also called jumper wires) for solderless breadboarding can be obtained in ready-to-use jump wire sets or can be manually manufactured. The latter can become tedious work for larger circuits. Ready to-use jump wires come in different qualities, some even with tiny plugs attached to the wire ends. Jump wire material for ready-made or homemade wires should usually be 22 AWG (0.33 mm2) solid copper, tin-plated wire - assuming no tiny plugs are to be attached to the wire ends. The wire ends should be stripped 3⁄16 to 5⁄16 in (4.8 to 7.9 mm). Shorter stripped wires might result in bad contact with the board's spring clips . Longer stripped wires increase the likelihood of short-circuits on the board. Needle-nose pliers and tweezers are helpful when inserting or removing wires, particularly on crowded boards.

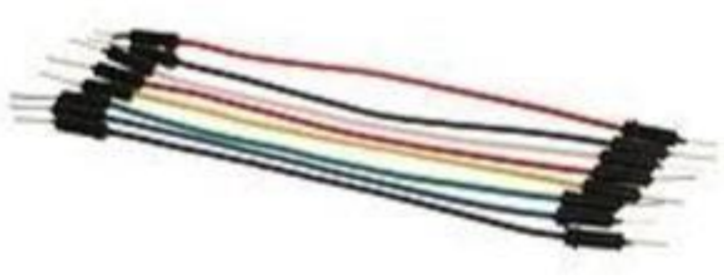


Figure 2.12 Jumper Wires

Differently colored wires and color-coding discipline are often adhered to for consistency. However, the number of available colors is typically far fewer than the number of signal types or paths. Typically, a few wire colors are reserved for the supply voltages and ground (e.g., red, blue, black), some are reserved for main signals, and the rest are simply used where convenient.

CHAPTER 3
SOFTWARE REQUIRED

3.1 Arduino IDE Software

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

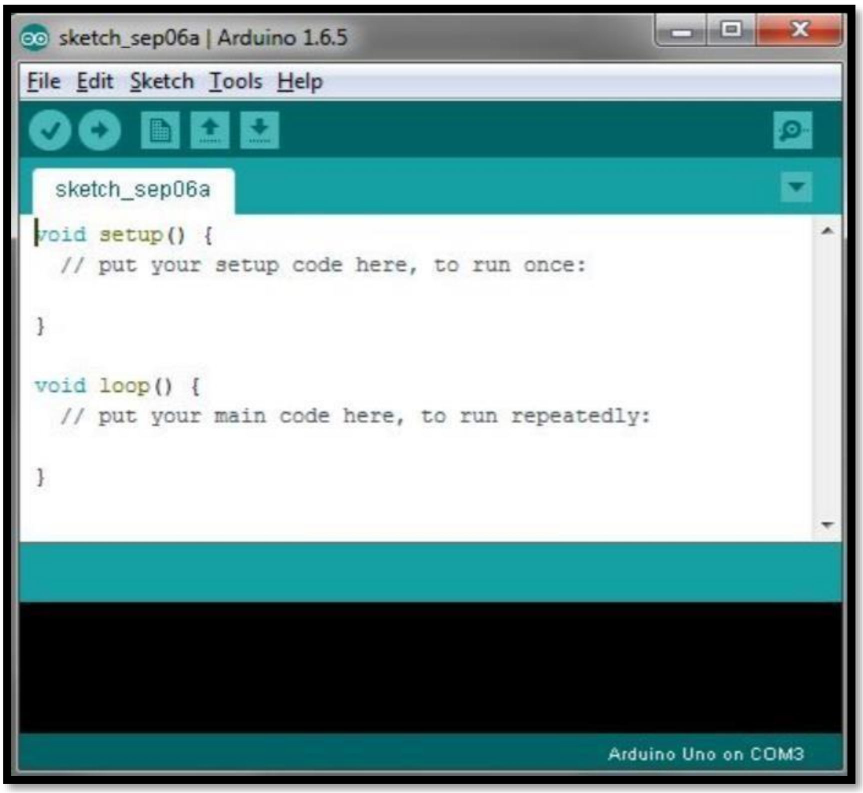
It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism to compile and load programs to an Arduino board. A program written with the IDE for Arduino is called a "sketch".

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

3.1.1 Arduino IDE: Initial Setup

We can download Arduino Integrated Design Environment (IDE) at https://www.arduino.cc/en/Main/Software.

This is the Arduino IDE once it’s been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us to upload code. Connect your Arduino board to the PC via the USB cable.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Arduino Uno on COM3

Figure 3.1 Arduino IDE default window

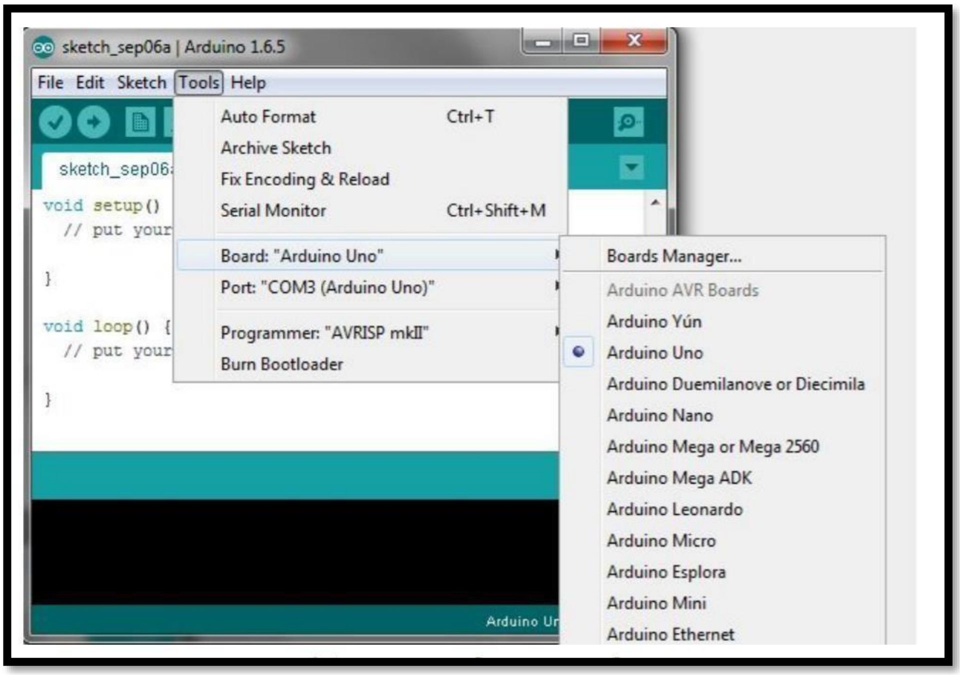
3.1.2 IDE: Board Setup

You have to tell the Arduino IDE what board you are uploading to. Select the Tools pull down menu and go to Board. This list is populated by default with the currently available Arduino Boards that are developed by Arduino. If you are using an Uno or an Uno-Compatible Clone (ex. Funduino, SainSmart, IEIK, etc.), select Arduino Uno. If you are using another board/clone, select that board.

3.1.3 IDE: COM Port Setup

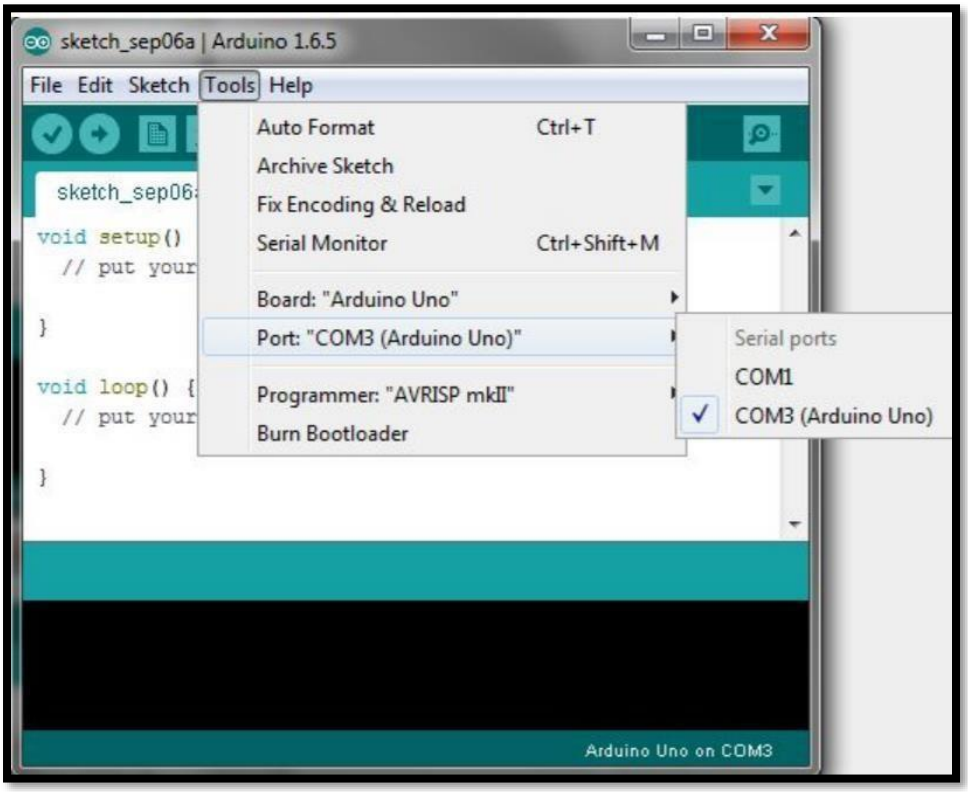
If you downloaded the Arduino IDE before plugging in your Arduino board, when you plugged in the board, the USB drivers should have installed automatically. The most recent Arduino IDE should recognize connected boards and label them with which COM port they are using. Select the Tools pulldown menu and then Port. Here it should list all open COM ports, and if there is a recognized Arduino Board, it will also give its name. Select the Arduino board that you have connected to the PC. If the setup was successful, in the bottom right of the Arduino IDE, you should see the board type and COM number of the board you plan to program.

NOTE: the Arduino Uno occupies the next available COM port; it will not always be COM3.



```
void setup()
  // put your

}

void loop() {
  // put your

}
```

Figure 3.2 Arduino IDE: Board setup procedure



```
void setup()
  // put your

}

void loop() {
  // put your

}
```

Figure 3.3 Arduino IDE: COM Port Setup

3.1.4 Uploading the program to your board

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

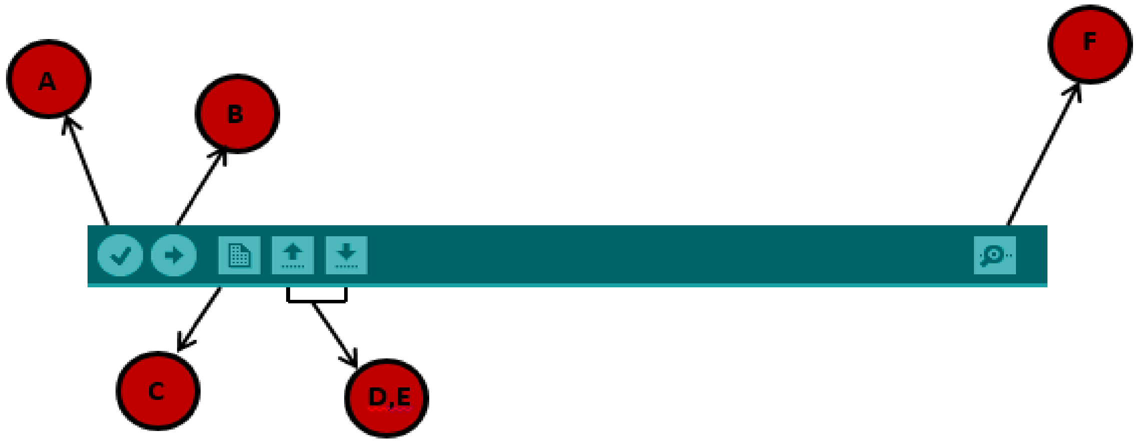


Figure 3.4 Uploading the program to Arduino

A – Used to check if there is any compilation error.

B – Used to upload a program to the Arduino board.

C – Shortcut used to create a new sketch.

D – Used to directly open one of the example sketch.

E – Used to save your sketch.

F – Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

3.2 Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics with high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development. It is used for: web development (server-side), software development, mathematics, system scripting.

3.2.1 Advantages of Python

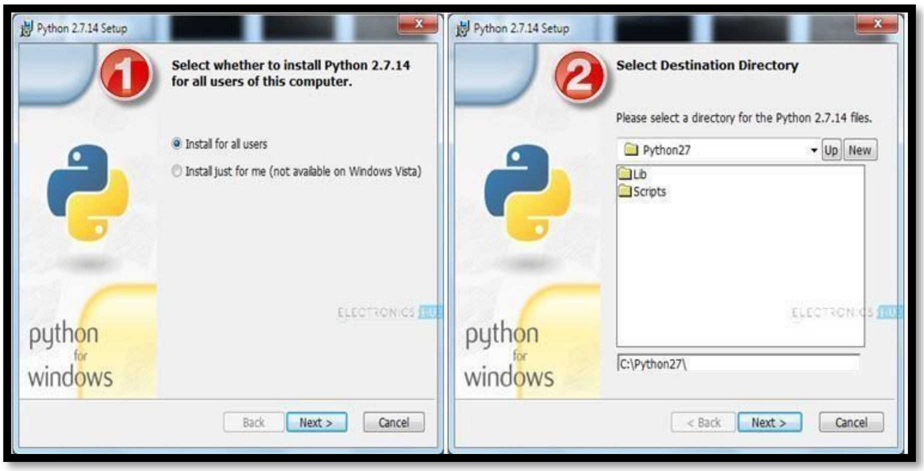
- Python can be used on a server to create web applications.
- It can be used alongside software to create workflows.
- It can connect to database systems.
- It can also read and modify files.
- It can be used to handle big data and perform complex mathematics.
- It can be used for rapid prototyping, or for production-ready software development.
- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.
- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- It is possible to write Python in an Integrated Development Environment, such as Thonny, PyCharm, NetBeans or Eclipse which are particularly useful when managing larger collections of Python files.
- Python Syntax compared to other programming languages
- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

3.2.2 Downloading Python

The first step is to download the Python IDE from the official website. The installation procedure and steps mentioned in this project are relevant only for Windows 10 Operating System. The setup procedure for Python on other operating systems like MAC OS and Linux will be different. Python can be downloaded from the Python's official website. Before downloading the Python IDE, we must know that there are two variants of Python: Python v2 and Python v3. Python v3 is the future version of Python and is still in its developing stage. Python v2, on the other hand, is a legacy version with a huge library support, communities and forums. So, we will be downloading the Python v2.7. Also, majority of the libraries are compiled for 32-bit version of the Windows. Hence, even if your system is a 64-bit one, we recommend installing a 32-bit version of Python v2.

3.2.3 Installing Python on Your Computer

After downloading the installation file, proceed with installation by double clicking it. Select "Install for all users" and click on next. Do not change the default installation directory, which is "C:\Python27\" and click on next.



Python 2.7.14 Setup

1 Select whether to install Python 2.7.14 for all users of this computer.

● Install for all users

○ Install just for me (not available on Windows Vista)

Python 2.7.14 Setup

2 Select Destination Directory

Please select a directory for the Python 2.7.14 files.

Python27

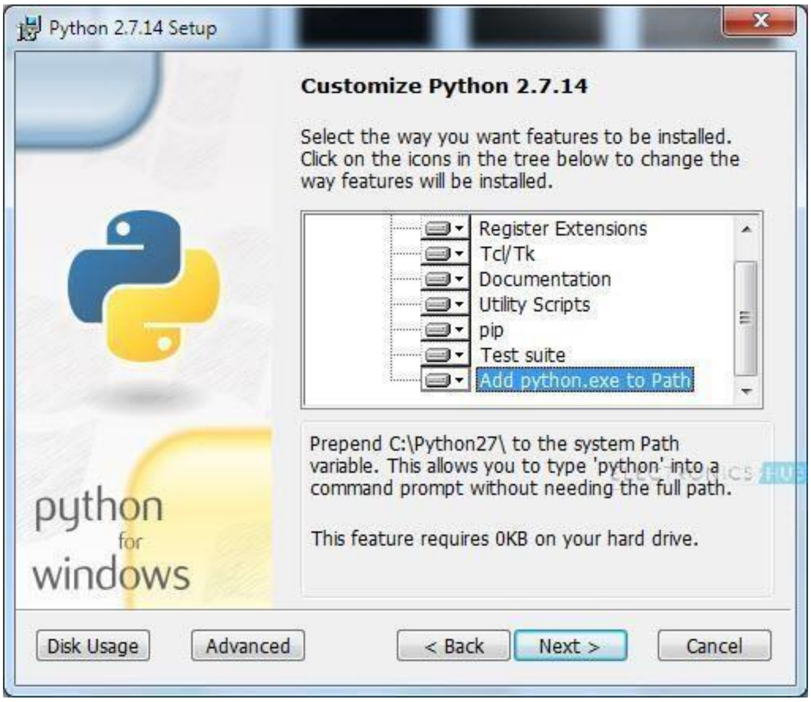
Lib

Scripts

C:\Python27\

Figure 3.5 Installing Python

Next, select the "Add python.exe to path" option in the Customize Python 2.7.14 area. This will automatically add the Python directory to the Windows System Path. If you skip this step, you can add the directory to the path manually. Proceed and finish the installation.



Python 2.7.14 Setup

Customize Python 2.7.14

Select the way you want features to be installed. Click on the icons in the tree below to change the way features will be installed.

- Register Extensions
- Tcl/Tk
- Documentation
- Utility Scripts
- pip
- Test suite
- Add python.exe to Path

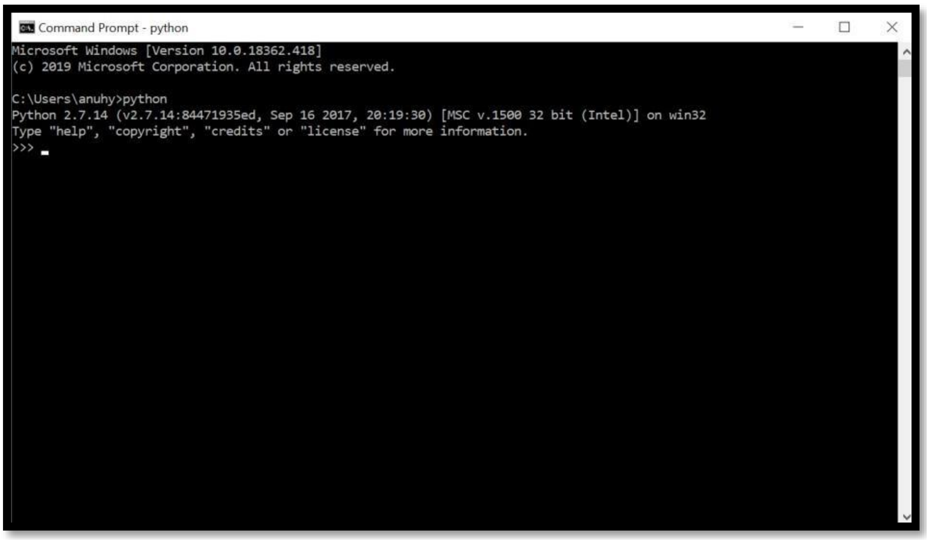
Prepend C:\Python27\ to the system Path variable. This allows you to type 'python' into a command prompt without needing the full path.

This feature requires 0KB on your hard drive.

Figure 3.6 Python Setup

Python is successfully installed on your computer. If you have set the system path (either manually or while installing path), you can check if the Python is successfully installed or not.

For this, run the Command Prompt with Administrator rights and just enter “python”. If python is successfully installed on your computer, you will get a response corresponding to the Python Version and type of installation (32-bit version).



Command Prompt - python

```
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\anuhy>python
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 3.7 Verifying Python Version

At this point, you can start using the Python IDE by searching for “Python” in the Start Menu of Windows 10. You will get IDLE (Python GUI) option. IDLE stands for Integrated Development and Learning Environment, a GUI based Python IDE.

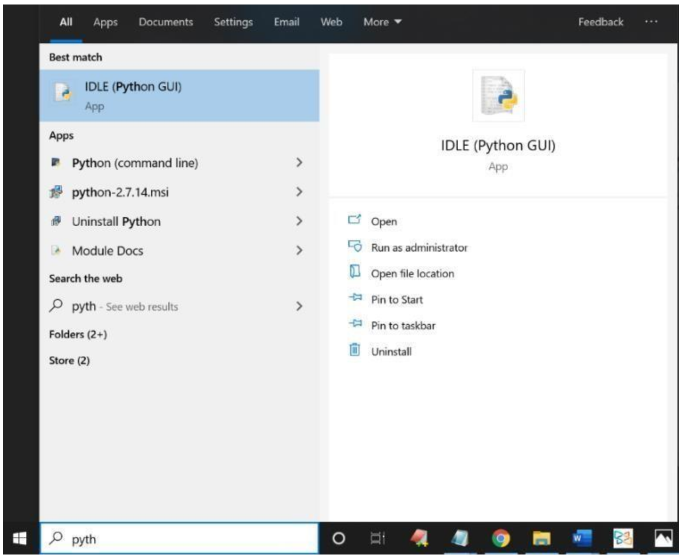
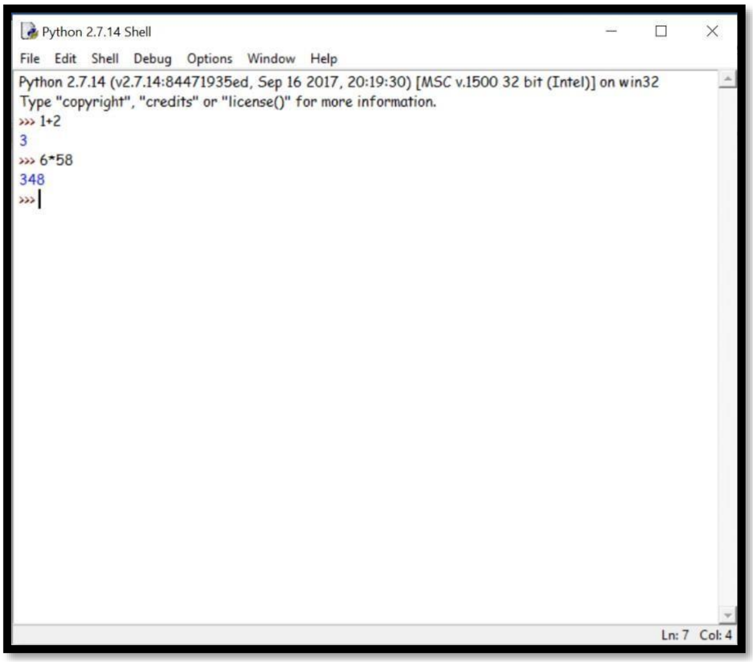


Figure 3.8 Python IDLE

Selecting it will open the Python Shell, which lets you use the Python interpreter in an interactive mode i.e. you can type commands and see the results instantaneously. For example, simple math operations can be performed just like that.



```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 1+2
3
>>> 6*58
348
>>> 
```

Figure 3.9 Python IDLE Shell

3.3 Interfacing Arduino with Python

Arduino is one of the most powerful open source electronics prototyping platform built around AVR Microcontrollers. Python, on the other hand, is one of the most widely used open source high level programming languages. Combining Arduino and Python will open doors to a wide range of ideas, projects and combinations. One such application is the Internet – of – things or IoT, which requires features like Communication Interfaces like Serial, graphical user interfaces, web interfaces, data storage and many other. Python has a huge collection of libraries that are open source and easy to use, which makes Python Programming Language the most suitable language for IoT Applications.

3.3.1 Getting pySerial

In order to access the Serial Port of the Computer through Python, we are going to need a library called pySerial. pySerial is a Python Serial Port Extension that provides access to the Serial Port for Python running on different Operating Systems like Windows, Linux, Mac OS (OSX), etc.

In our project, we will be using pySerial to Read and Write data to Arduino through Python. In order to download the pySerial from "***pyserial*-3.4.tar.gz**"
Extract the contents of the file to any folder like "C:\pyserial-3.4".

NOTE: Location is important, as we have to navigate to this location and install the pySerial.

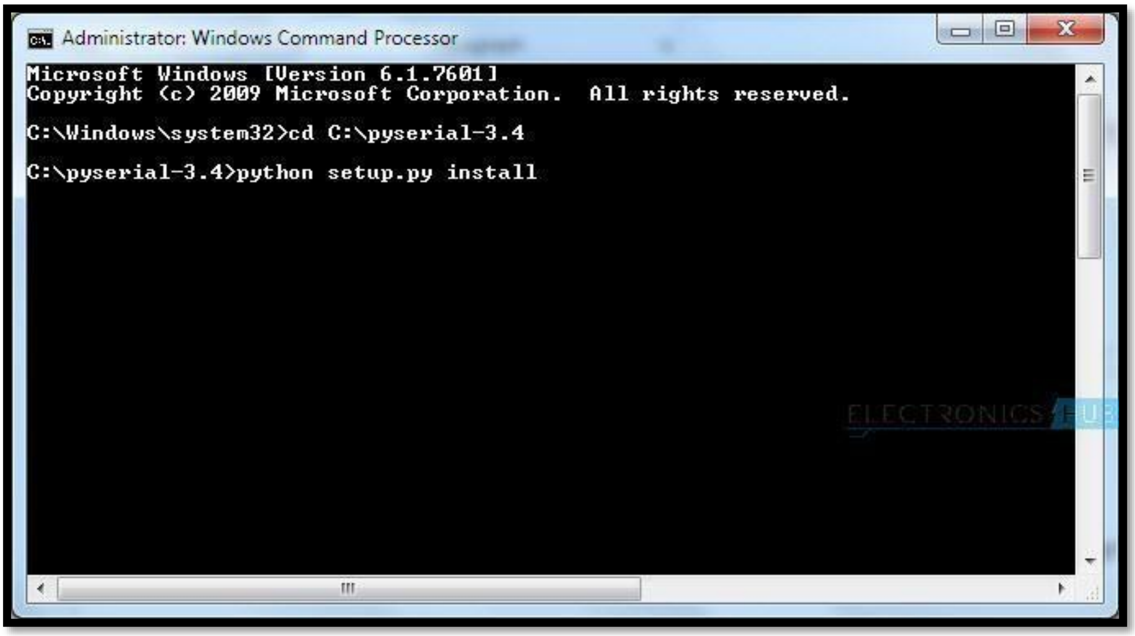
Now, run the Windows Command Prompt with Administrator privileges and navigate to the folder where pySerial is extracted to. (Here we extracted it to C:\pyserial-3.4). In order to install pySerial, type the following command in the command prompt and hit enter.

```
python setup.py install
```

After successful installation of pySerial, you can check if it is integrated to Python or not by entering the following command in the Python IDLE.

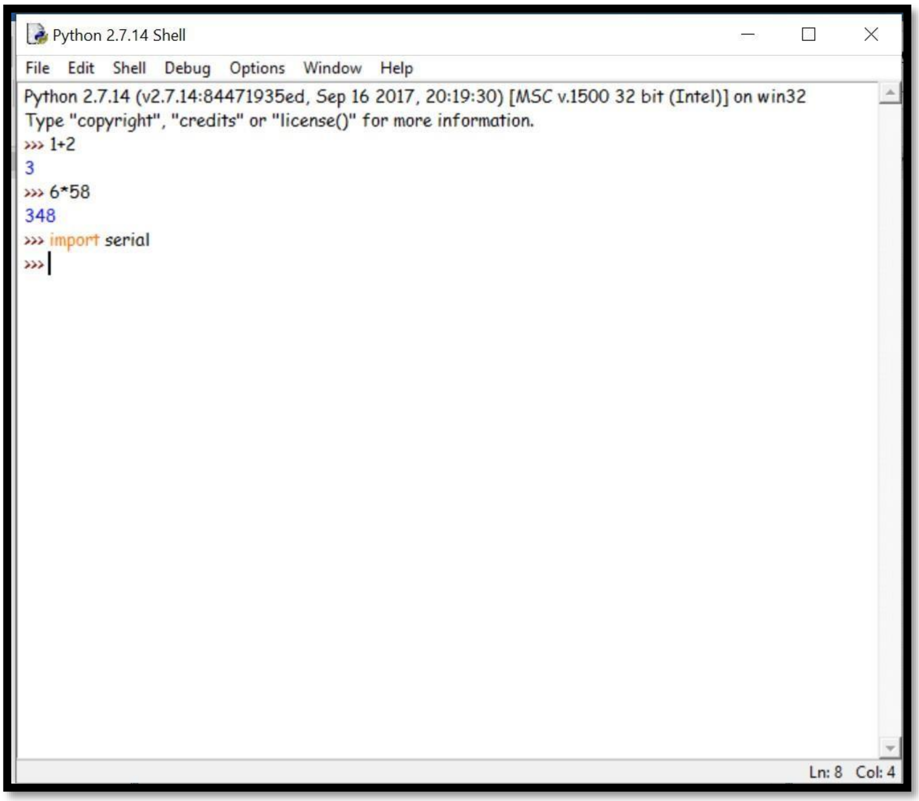
```
import serial
```

If everything goes well, you will not get any error.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd C:\pyserial-3.4

C:\pyserial-3.4>python setup.py install
```

Figure 3.10 Installing pySerial



```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 1+2
3
>>> 6*58
348
>>> import serial
>>>
```

Figure 3.11 Importing serial library in python shell

Now that we have successfully installed Python and pySerial, we will continue with interfacing Arduino with Python.

3.4 Controlling Arduino with Python

Upload the Arduino Code to the Arduino UNO. Note the COM Port to which it is connected. The same COM Port number should be given in the Python Code.
After uploading the code to Arduino, launch the Python Program. The Python Shell will be opened and the program starts executing.

3.5 Python Programming for the Project

Writing Python Program for Arduino based Hand Gesture Control is very simple. We 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.

3.5.1 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. Open Command Prompt with Administrator privileges and change to the directory where Python is installed.
In the latest version of Python, then pip (a tool for installing packages in Python) will already be installed. To check if pip is installed or not, type the following command.

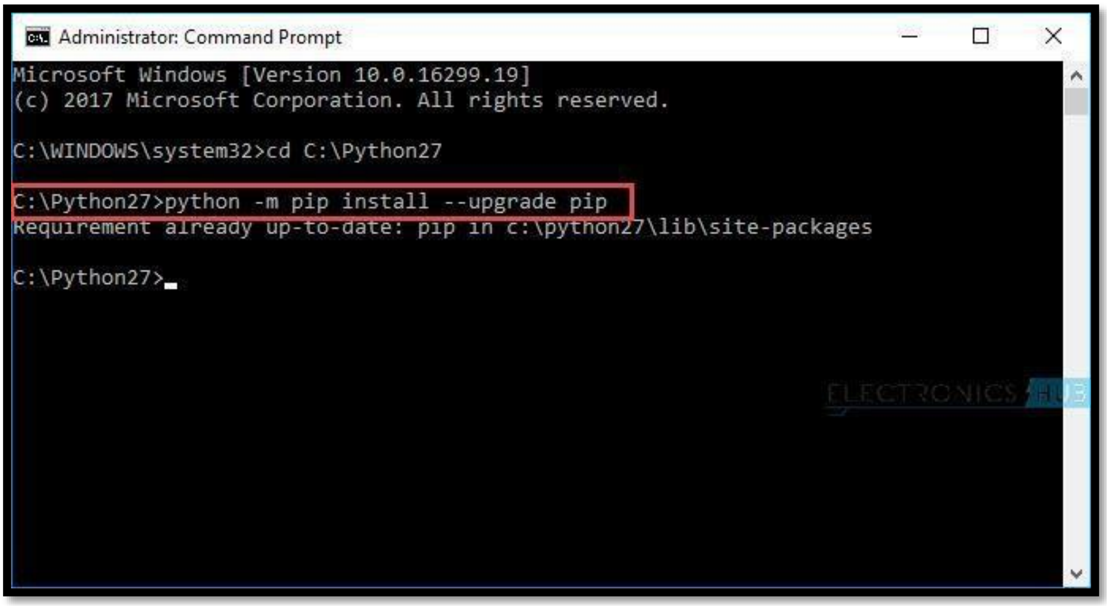
```
pip -V
```

You should upgrade to the latest package of pip using the following command. If pip is already in its latest version, then ignore this step.

```
python -m pip install -U pip
```

Or

```
python -m pip install –upgrade pip
```



```
Microsoft Windows [Version 10.0.16299.19]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Python27

C:\Python27>python -m pip install --upgrade pip
Requirement already up-to-date: pip in c:\python27\lib\site-packages

C:\Python27>
```

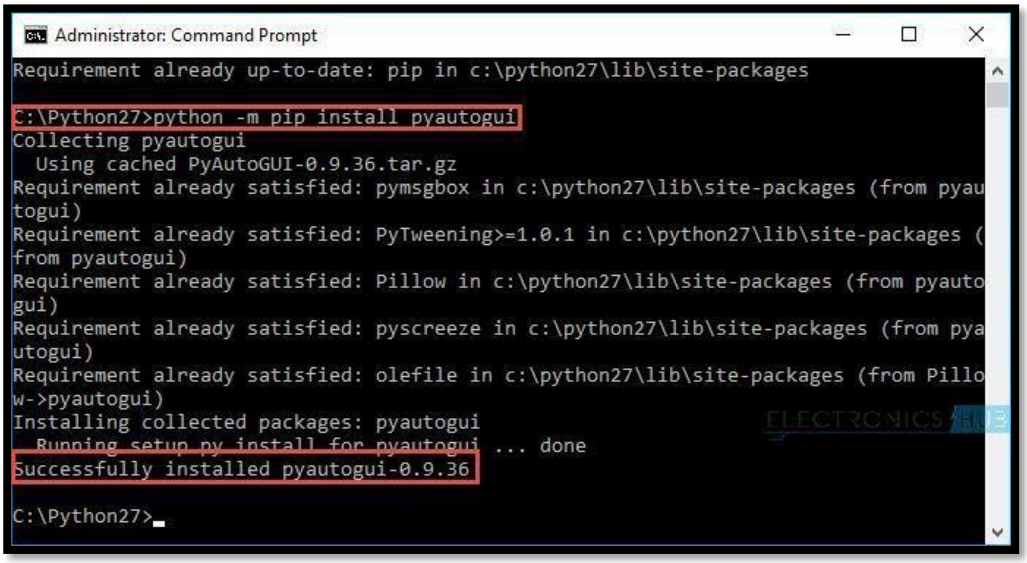
Figure 3.12 Upgrading pip

After upgrading pip, you can proceed to install PyAutoGUI. In order to install PyAutoGUI, type the following command.

Python -m pip install pyautogui

Or

pip install pyautogui



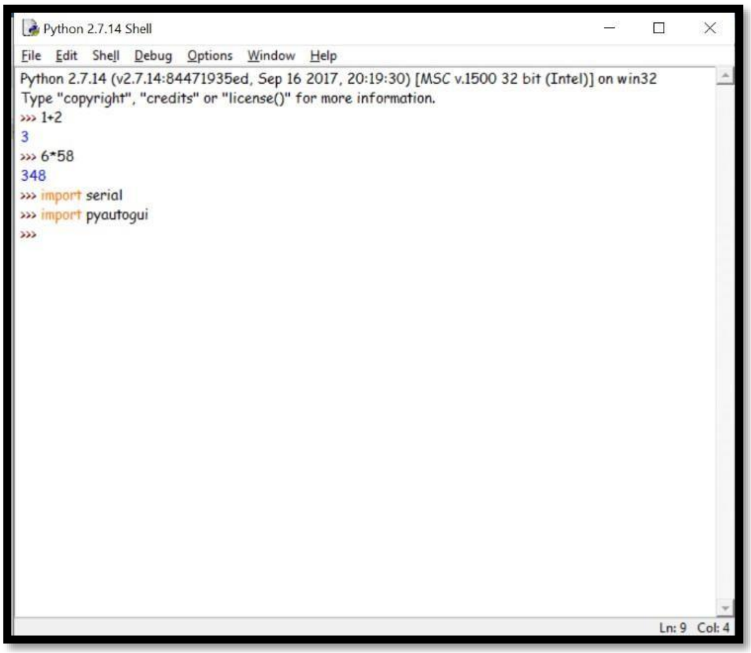
```
Requirement already up-to-date: pip in c:\python27\lib\site-packages

C:\Python27>python -m pip install pyautogui
Collecting pyautogui
  Using cached PyAutoGUI-0.9.36.tar.gz
Requirement already satisfied: pymsgbox in c:\python27\lib\site-packages (from pyau
togui)
Requirement already satisfied: PyTweening>=1.0.1 in c:\python27\lib\site-packages (
from pyautogui)
Requirement already satisfied: Pillow in c:\python27\lib\site-packages (from pyauto
gui)
Requirement already satisfied: pyscreeze in c:\python27\lib\site-packages (from pya
utogui)
Requirement already satisfied: olefile in c:\python27\lib\site-packages (from Pillo
w->pyautogui)
Installing collected packages: pyautogui
  Running setup.py install for pyautogui ... done
Successfully installed pyautogui-0.9.36

C:\Python27>
```

Figure 3.13 Installing pyautogui

After successful installation of pyautogui, you can check if it is integrated to Python or not by entering the following command in the Python IDLE.

```
import pyautogui
```

If everything goes well, you will not get any error.



```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 1+2
3
>>> 6*58
348
>>> import serial
>>> import pyautogui
>>>
```

Figure 3.14 Importing pyautogui

3.6 Python Programming

Now, we will take a look at the Python Code. First, open the Python IDLE and create a new file by clicking on File – >New File. This will open an empty text editor. Start writing the code in the and when you are done save the file using File – > Save or Ctrl+S. This will ask you to enter a name for the file. Give an appropriate name and save the file at a convenient location. The extension for Python Programs is “.py”. This will be automatically assigned to the file when you are saving it. After saving the file, we can run the file by selecting Run – > Run Module in the editor window.

CHAPTER 4

IMPLEMENTATION

4.1 Hardware Requirement

The hardware components used in the system are:

A. Arduino UNO with USB cable

B. Ultrasonic Sensors

C. Jumper wires

D. Laptop

4.2 Software Requirement

The main software's required for this project is

A. Arduino IDE

B. Python

4.3 Block Diagram

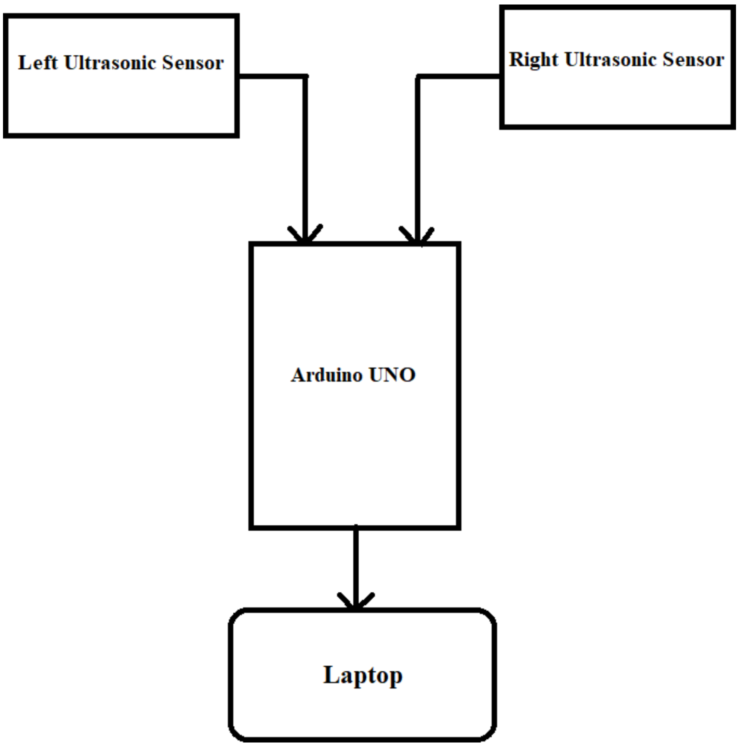


Figure 4.1 Block Diagram

4.4 Circuit Diagram and Design of the project

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.

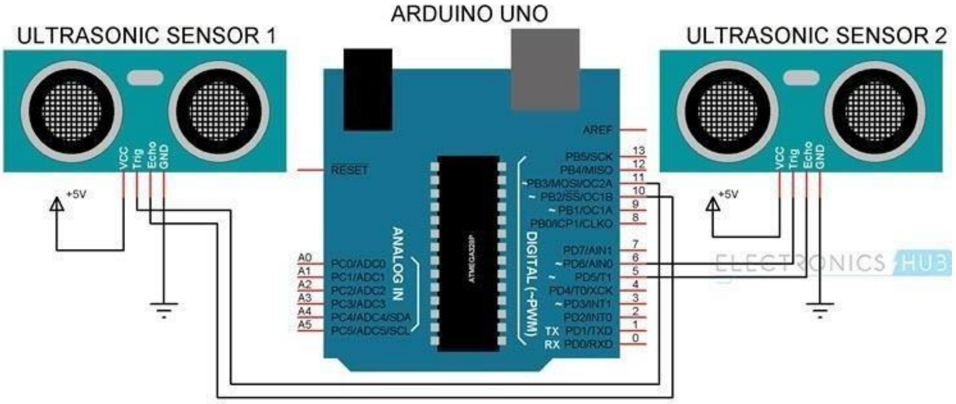


Figure 4.2 Circuit Diagram

The design of the circuit is very simple, but the setup of the components is very important. The Trigger and Echo Pins of the first Ultrasonic Sensor (that is placed on the left of the screen) are connected to Pins 11 and 10 of the Arduino. For the second Ultrasonic Sensor, the Trigger and Echo Pins are connected to Pins 6 and 5 of the Arduino.

Now, coming to the placement of the Sensors, place both the Ultrasonic Sensors on top of the Laptop screen, one at the left end and the other at right. You can use double sided tape to hold the sensors onto the screen.

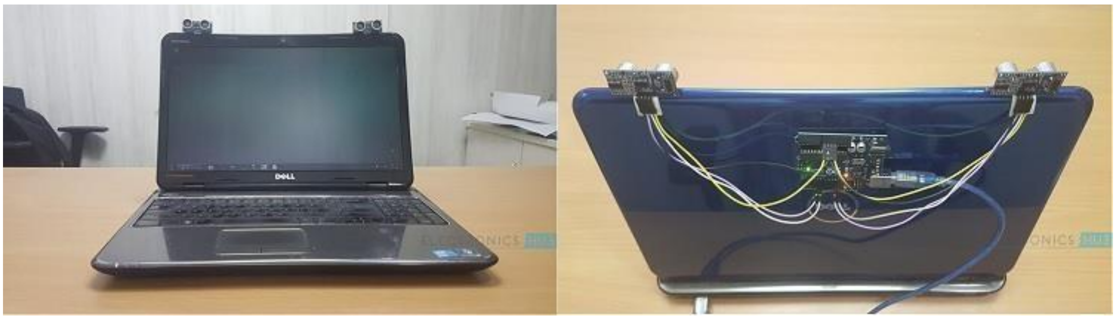


Figure 4.3 Circuit after connecting to a laptop

Coming to Arduino, place it on the back of the laptop screen. Connect the wires from Arduino to Trigger and Echo Pins of the individual sensors. Now, we are ready for programming the Arduino.

4.5 Programming Your Arduino to Detect Gestures

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.

The 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. Before taking a look at the gestures, let us first see the tasks that we can accomplish.

- 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 we’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.

Based on the above-mentioned gesture, the following Arduino Program has been written.

4.6 Arduino Code

```
const int trigPin1 = 11; // the number of the trigger output pin ( sensor 1 )
const int echoPin1 = 10; // the number of the echo input pin ( sensor 1 )
const int trigPin2 = 6; // the number of the trigger output pin ( sensor 2 )
const int echoPin2 = 5; // the number of the echo input pin ( sensor 2 )

long duration;
int distance1, distance2;
float r;
unsigned long temp=0;
int temp1=0;
int l=0;

void find distance (void);

// this function returns the value in cm.
/*we should not trigger the both ultrasonic sensor at the same time.
it might cause error result due to the intraction of the both sounds waves.*/
void find distance (void)
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
```

```
 duration = pulseIn(echoPin1, HIGH, 5000);// here this pulse in function won’t wait
more then 5000us for the ultrasonic sound to came back. (due to this it won’t measure
more than 60cm)
                                          // it helps this project to use the gesture control in the
defined space.
                                          // so that, it will return zero if distance greater than 60m. (
it helps usually if we remove our hands in front of the sensors ).

  r = 3.4 * duration / 2;                 // calculation to get the measurement in cm using
the time returned by the pulse in function.
  distance1 = r / 100.00;
//upper part for left sensor and lower part for right sensor
  digitalWrite(trigPin2, LOW);
  delay Microseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);

  duration = pulseIn(echoPin2, HIGH, 5000);
  r = 3.4 * duration / 2;
  distance2 = r / 100.00;
  delay(100);
}

void setup()
{
  Serial.begin(9600);
  pinMode(trigPin1, OUTPUT); // initialize the trigger and echo pins of both the
sensor as input and output:
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  delay (1000);
```

```
}
void loop()
{
  find_distance(); // this function will store the current distance measured by the
ultrasonic sensor in the global variable "distance1 and distance2"
                    // no matter what, the program has to call this "find_distance" function
continuously to get the distance value at all time.

  if(distance2<=35 && distance2>=15) // once if we placed our hands in front of the
right sensor in the range between 15 to 35cm this condition becomes true.
  {
    temp=millis();                   // store the current time in the variable temp. (" millis "
Returns the number of milliseconds since the Arduino board began running the
current program )
    while(millis()<=(temp+300)) // this loop measures the distance for another 300
milliseconds. ( it helps to find the difference between the swipe and stay of our hand
in front of the right sensor )
    find_distance();
    if(distance2<=35 && distance2>=15)
// this condition will true if we place our hand in front of the right sensor for more
//then 300 milli seconds.
    {
     temp=distance2;                         // store the current position of our hand in the
variable temp.
     while(distance2<=50 || distance2==0)    // this loop will run until we remove our
hand in front of the right sensor.
     {
       find_distance();                      // call this function continuously to get the live data.
       if((temp+6)<distance2)                // this condition becomes true if we move our
hand away from the right sensor (**but in front of it ). here " temp+6 " is for
calibration.
       {
       Serial.println("down");               // send "down" serially.
       }
    else if((temp-6)>distance2)      // this condition becomes true if we move our
hand closer to the right sensor.
    {
     Serial.println("up");          // send "up" serially.
    }
   }
  }
  else                          // this condition becomes true, if we only swipe in front
of the right sensor .
  {
   Serial.println("next");          // send "next" serially.
  }
 }

 else if(distance1<=35 && distance1>=15) // once if we placed our hands in front of
the left sensor in the range between 15 to 35cm this condition becomes true.
 {

  temp=millis();

  while(millis()<=(temp+300))
  {
    find_distance();
    if(distance2<=35 && distance2>=15) // if our hand detects in the right sensor
before 300 milli seconds this condition becomes true. ( usually it happens if we swipe
our hand from left to right sensor )
    {
     Serial.println("change");     // send "change" serially.
     l=1;                        // store 1 in variable l. ( it avoids the program to enter into
the upcoming if condition )
     break;                      // break the loop.
    }
  }
```

```
    if(l==0)                            // this condition will become true, only if we swipe our
hand in front of left sensor.
    {
    Serial.println("previous");            // send "previous" serially.
    while(distance1<=35 && distance1>=15)
 // this loop will rotate until we remove our hand in front of the left sensor. this will
//avoid not to enter this if condition again.
    find_distance();
    }
    l=0;                                // make l=0 for the next round.
   }
}
```

4.7 Python Programming for the project

If everything goes well till now, you can proceed to write the Python Code. If you observe the Arduino Code given above, 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

The Python Code for Arduino based Hand Gesture Control of Computer is given below.

NOTE: We have used Google Chrome as Web Browser and VLC Player as Media Player. Also, we modified the hotkeys of VLC Player to suit our Python Program. The modifications are as follows.

- Keypress = Up Arrow – – > Action = Increase Volume
- Keypress = Down Arrow – – > Action = Decrease Volume
- Keypress = Ctrl+PgUp – – > Action = Play/Pause

4.8 Python Code

```
import serial                          # add Serial library for serial communication
import pyautogui                          # add pyautogui library for programmatically
controlling the mouse and keyboard.
Arduino_Serial = serial.Serial('com12',9600)      # Initialize serial and Create Serial
port object called Arduino_Serial

while 1:
   incoming_data = str (Arduino_Serial.readline()) # read the serial data and print it as
line
   print incoming_data                      # print the incoming Serial data



   if 'next' in incoming_data:                # if incoming data is 'next'
      pyautogui.hotkey('ctrl', 'pgdn')        # perform "ctrl+pgdn" operation which
moves to the next tab

   if 'previous' in incoming_data:             # if incoming data is 'previous'
      pyautogui.hotkey('ctrl', 'pgup')        # perform "ctrl+pgup" operation which
moves to the previous tab

   if 'down' in incoming_data:                # if incoming data is 'down'
```

```
        #pyautogui.press('down')                   # performs "down arrow" operation which
scrolls down the page
        pyautogui.scroll(-100)

    if 'up' in incoming_data:                     # if incoming data is 'up'
        #pyautogui.press('up')                    # performs "up arrow" operation which
scrolls up the page
        pyautogui.scroll(100)

    if 'change' in incoming_data:                  # if incoming data is 'change'
        pyautogui.keyDown('alt')                   # performs "alt+tab" operation which
switches the tab
        pyautogui.press('tab')
        pyautogui.keyUp('alt')

    incoming_data = "";                           # clears the data
```

4.9 Execution

As the code for Arduino and Python are written, upload the Arduino code to Arduino UNO board. See through that the Arduino code is dumped into the Arduino UNO Board without any errors. Connect the Arduino board and Ultrasonic sensors to Laptop. Note the COM Port to which it is connected. The same COM Port number should be given in the Python Code.

After uploading the code to Arduino, launch the Python Program. The Python Shell will be opened. In case of errors or exceptions, check the connections and relaunch the python module.

4.10 Algorithm

The following is the algorithm of the project.

Step 1: First the ultrasonic sensor will capture the gesture.

Step 2: The captured gesture will be compared with the stored gestures.

Step 3: If the gesture matches with the stored gesture then a particular action will be executed by the system.

Step 4: If the gesture did not match stored gesture then system will not perform any action.

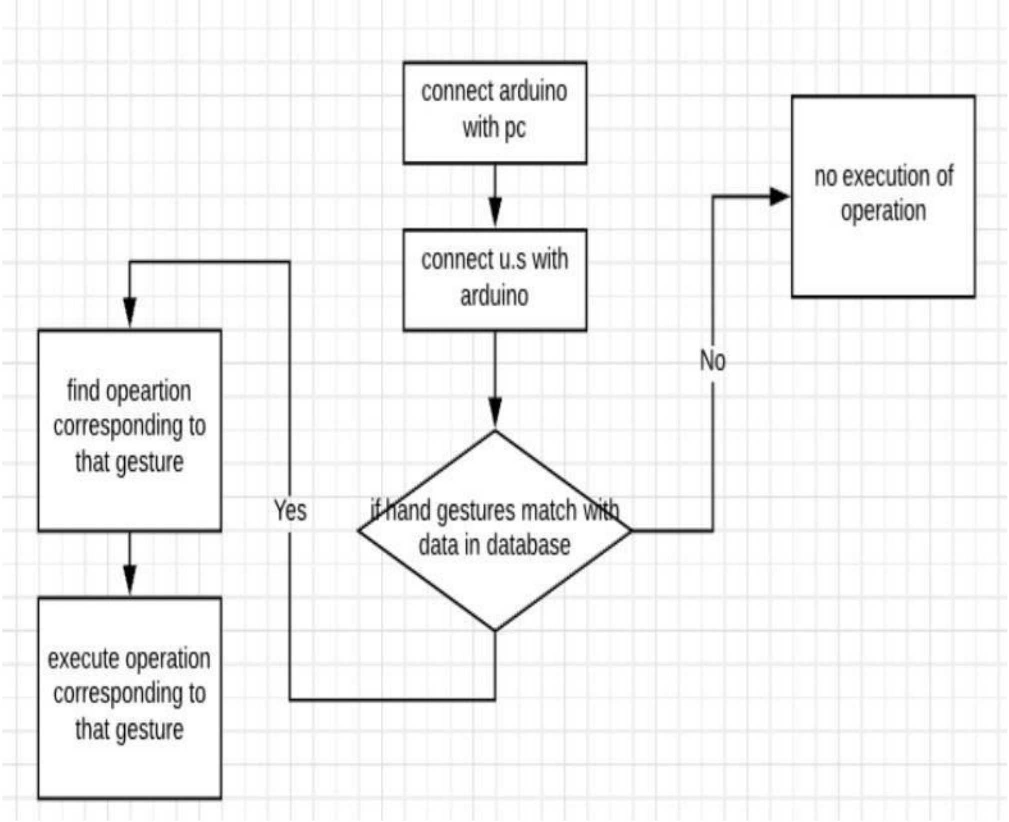

connect arduino with pc
connect u.s with arduino
no execution of operation
No
find opeartion corresponding to that gesture
Yes
if hand gestures match with data in database
execute operation corresponding to that gesture

Figure 4.4 Flowchart of Algorithm

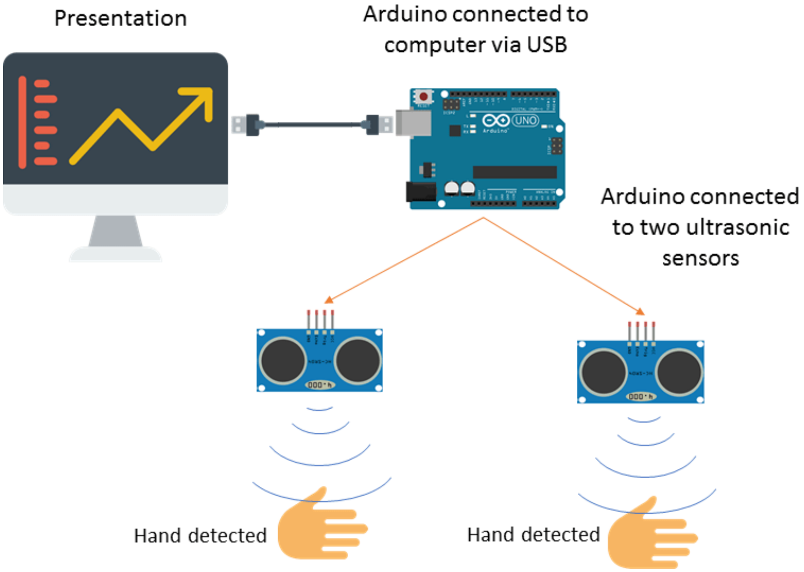
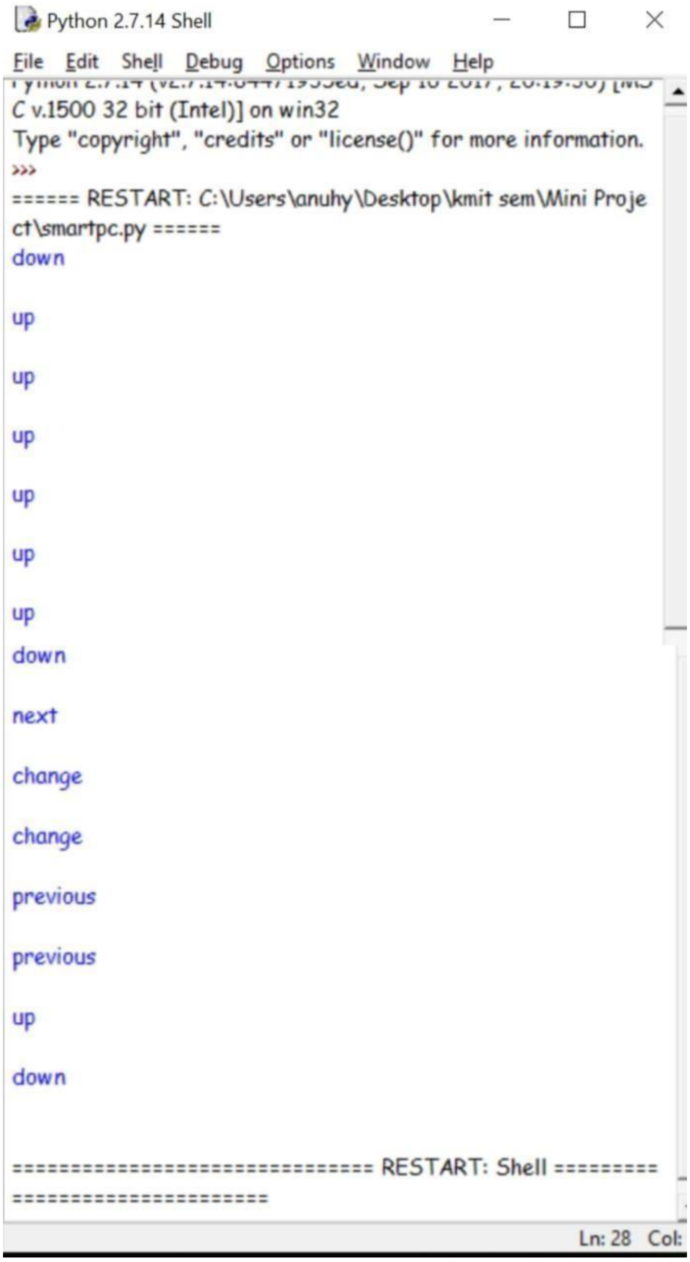

Presentation
Arduino connected to computer via USB
Arduino connected to two ultrasonic sensors
Hand detected
Hand detected

Figure 4.5 Pictorial representation of algorithm

4.11 Output in Python Shell

The output command of the project is printed in python IDLE shell.



```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MS
C v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
====== RESTART: C:\Users\anuhy\Desktop\kmit sem\Mini Proje
ct\smartpc.py ======
down

up

up

up

up

up

up
down

next

change

change

previous

previous

up

down


=============================== RESTART: Shell =========
======================
```

Figure 4.6 Output in Python Shell

CHAPTER 5

ADVANTAGES AND APPLICATIONS

5.1 Advantages

- For this system there is no need of sound to be created so no interruption of background noise.
- A number of functions of computer can be operated by using ultrasonic sensor.
- This technique may be very useful for those who does not know functionally of computer.
- This technique decreases the learning time required.
- Using this technique it is easy to interact with the computer and there is no language barrier.
- By using this system we can control our laptop from a small distance and it can help to control laptop in conference room presentation.

5.2 Applications

This type of technology can be used

- in classroom for easier and interactive learning,
- Immersive gaming,
- Interacting with virtual objects on screen,
- AR(Augmented Reality),
- 3D Design.

5.2.1 Immersive Gaming

The idea is just that a game (or any other media from books to movies) creates spatial presence when the user starts to feel like he is "there" in the world that the game creates. People who experience immersion tend to only consider choices that make sense in the context of the imaginary world. Immersion into virtual reality (VR) is a perception of being physically present in a non-physical world. The perception is created by surrounding the user of the VR system in images, sound or other stimuli that provide an engrossing total environment. Immersive virtual reality is a hypothetical future technology that exists today as virtual reality art projects, for the most part. It consists of immersion in an artificial environment where the user feels just as immersed as they usually feel in everyday life.

5.2.2 AR (Augmented Reality)

Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information including visual, auditory, haptic, somatosensory and olfactory. AR can be defined as a system that fulfils three basic features: a combination of real and virtual worlds, real-time interaction, and accurate 3D registration of virtual and real objects.

5.2.3 3D Design

3D modeling is the process of developing a mathematical representation of any surface of an object (either inanimate or living) in three dimensions via specialized software. The product is called a 3D model. 3D modeling is used in various industries like film, animation and gaming, interior design and architecture. They are also used in the medical industry to create interactive representations of anatomy. A wide number of 3D software are also used in constructing digital representation of mechanical models or parts before they are actually manufactured. CAD/CAM related software are used in such fields, and with this software, not only can you construct the parts, but also assemble them, and observe their functionality. 3D modeling is also used in the field of Industrial Design, wherein products are 3D modeled before representing them to the clients. In Media and Event industries, 3D modeling is used in Stage/Set Design.

CHAPTER 6
CONCLUSION

In this project, we have implemented Arduino based Hand Gesture Control of Your Computer, where few hand gestures made in front of the computer will perform certain tasks in the computer without using mouse or keyboard. It presents one of the solutions among various others, for operating a computer using hand gestures. It is one of the easiest ways of interaction between human and computer. It is a cost-effective model which is only based on Arduino UNO and ultrasonic sensor. 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 will become more effective if merged with completely developed hologram technology. We will be able to interact with virtual 3D objects in the physical world in real time. Though gesture recognition is still in its infancy state, the innovative market scenario which has seen Virtual Reality being a global phenomenon pushes us to admit that its application areas could keep on growing.

By applying this methodology we conclude that, no matter how powerful and complex, we always have to be near a machine and somehow in physical contact with it to interact with it. But gesture recognition technology could change all that. If perfected and used correctly, it could actually render traditional input devices like keyboard, mouse and touch screens redundant. Besides being really functional, it cannot be argued against that gesture control that looks really cool. While it might seem like a technology that will only increase our lethargy, truth is that other than making life easier, it also has a vast array of applications in almost all fields.

CHAPTER 7
FUTURE SCOPE

Leap Technology is a breakthrough in computer interaction. Through this technology we can say good bye to mouse and keyboard. Leap motion is a start-up developing advanced motion sensing technology for HCI (Human- Computer Interface). This is a new way of interaction using Motion detection – a process of detecting changes in position of an object relative to surrounding.

Leap is a tiny device which looks like any old USB, but has the power to change how we interact with computers. This device detects the motion and accordingly interacts with computer.

Figure 7.1 Leap Motion Device

Applications of Leap Technology are

- Robotics : Self navigate, Mimic Human Movement
- Gaming : More intuitive compared to mouse or even touch screen.
- Music and Video : Electronic Music, Flexible to create and edit videos.
- For surgeons : Contactless, maintain sterilization, Viewing reports

REFERENCES

[1] https://www.electronicshub.org/arduino-based-hand-gesture-control-computer/#Circuit_Diagram

[2] https://www.electronicshub.org/controlling-arduino-led-python/

[3] https://circuitdigest.com/microcontroller-projects/control-your-computer-with-hand-gestures

[4] https://en.wikipedia.org/wiki/Arduino_Uno

[5] Nidhi Gupta, Ramandeep Singh , Sidharth Bhatia. Hand gesture recognition using ultrasonic sensor and atmega128 microcontroller; June 2014 [online] http://esatjournals.net/ijret/2014v03/i06/IJRET20140306107.pdf

[6] Dong-Ik Go, Gaurav Agarwal. Gesture Recognition: Enabling natural interactions with electronics; April 2012 [online] URL: http://www.ti.com/lit/wp/spry199/spry199.pdf

[7] Huang, Pavlovic, Sharma. Visual Interpretation of Hand Gestures for Human-Computer Interaction

[8] Rishabh Agrawal and Nikita Gupta published in IEEE.(2010) “Real Time Hand Gesture Recognition for Human Computer Interaction”

[9] Meenakshi Panwar and Pawan Singh Mehra published in IEEE.(2014) “hand gesture recognition for human computer interaction”

[10] Wang Zhi-heng, Cao Jiang-Tao and Liu Jin-guo published in IEEE.(2009) “Design of human-computer interaction control system based on hand-gesture recognition”

[11] Kaoru Yamagishi, Lie Jing And Zixue Cheng Published In IEEE. “A System for Controlling Personal Computers by Hand Gestures Using A Wireless Sensor Device”

[12] Dushyant Kumar Singh published in IEEE.(2010) “Recognizing hand gestures for human computer interaction”

[13] Minh Q. Nguyen, Changzhi Li published in IEEE.(2010) “Radar and ultrasound hybrid system for human computer interaction”

[14] Yuvraj V Parkale published in IEEE “Gesture Based Operating System Control”

[15] Gergely Sziladi and Tibor Ujbanyi published in IEEE “The analysis of hand gesture-based cursor position control during solve an IT related task” on 2-4 April 2015 Melmaruvathur,India.

[16] U. Rajkanna, M. Mathankumar and K. Gunasekaran published in IEEE “Hand gesture-based computer control using microcontroller” on 7-8 Jan 2012,Rohtak, Haryana ,India.

[17] Andrea Attwenger published in IEEE “Controlling Computer Application Using Hand Gestures Using Arduino” .(2012)

[18] http://en.wikipedia.org/wiki/Leap_Motion

[19] https://www.leapmotion.com/technology/