

**A**  
**B.E. PROJECT REPORT**  
**On**

**“JUST A RATHER VERY INTELLIGENT SYSTEM  
AI DESKTOP VOICE ASSISTANCE”**

Submitted as partial fulfillment for the award of Bachelor of Engineering with  
specialization in  
*“Electronics and Tele-Communication Engineering”*

**To**  
**North Maharashtra University, Jalgaon.**

**Submitted By:-**  
**Pratiksha Manhoar Narote**  
**Ganesh Manohar Sisode**  
**Chanchal Nanaji Gayakwad**  
**Chinmay Shantilal Khairnar**  
**BE(E&Tc)/2024-25**

Under the Guidance of  
**Prof. P. Y. Nikume**  
Assistant Professor, Department of E&TC Engineering, GCOE Nangaon



**DEPARTMENT OF**  
**ELECTRONICS AND TELECOMMUNICATION ENGINEERING**

**Nagaon Education Society's**  
**Gangamai College of Engineering, Nagaon**  
**B+ NACC Accredited – (1<sup>ST</sup> Cycle)**

## CERTIFICATE

This is to certify that the “JUST A RATHER VERY INTELLIGENT SYSTEM AI DESKTOP VOICE ASSISTANCE” has been carried out by “**Sisode Ganesh, Narote Pratiksha, Gaikwad Chanchal, Khairnar Chinmay** ” submitted as partial fulfillment for the award of Bachelor in Engineering in “*Electronics and Telecommunication Engineering*” degree from **North Maharashtra University, Jalgaon (M.S.)**. This report submitted by “**Sisode Ganesh, Narote Pratiksha, Gaikwad Chanchal, Khairnar Chinmay** ” includes the work carried by him during the academic year 2024-25. To the best of my knowledge and belief, this work has not been submitted elsewhere for the award of any other degree.

Date:

Place: Department of Electronics & Telecommunication Engineering Gangamai College of Engineering ,  
Nagaon , Dhule(M.S) 424005 (India)

**Prof. P. Y. Nikume**  
Guide

**Prof. C. V. Patil**  
BE Coordinator

**Prof. R. M. Patil**  
H.O.D.

**Dr. G. M . Podar**  
Principal  
Gangamai College of Engineering ,  
Nagaon , Dhule(M.S) 424005

## **DECLARATION**

I hereby declare that the project entitled, “**J.A.R.V.I.S., AI DESKTOP VOICE ASSISTANCE**” was carried out and written by me under the guidance of Prof. Mr. (R. M. Patil) (HOD), Department of Electronics & Telecommunication Engineering, Gangamai College of Engineering, Nagaon. This work has not been previously formed the basis for the award of any degree or diploma or certificate nor has been submitted elsewhere for the award of any degree or diploma or certificate.

Place : Nagaon

Date : 25<sup>th</sup> May 2025

**Pratiksha Manhoar Narote**

**Ganesh Manohar Sisode**

**Chanchal Nanaji Gayakwad**

**Chinmay Shantilal Khairnar**

## ACKNOWLEDGEMENT

We have a great pleasure in submitting this dissertation entitled “JUST A RATHER VERY INTELLIGENT SYSTEM AI DESKTOP VOICE ASSISTANCE”, which was in fulfillment of the requirement of the B.E. (Electronics and Telecommunication Engineering), North Maharashtra University, Jalgaon.

We would like to express my deepest gratitude to my guide respected **Prof. P.Y.Nikume** for his invaluable support, guidance, motivation and encouragement through-out the period this work was carried out. It was a great pleasure to work under his guidance.

We would like to acknowledge **Prof. R. M. Patil, HOD (E&TC Department)** & **Dr. V. M . Patil**, Principal of GCOE, Nagaon for his great cooperation and support.

We are sincerely like to express our gratitude to the entire teaching staff member, my colleague, my family and those who knowingly and unknowingly have contributed in completion of this project Stage I &II.

**Pratiksha Manhoar Narote**  
**Ganesh Manohar Sisode**  
**Chanchal Nanaji Gayakwad**  
**Chinmay Shantilal Khairnar**

## **ABSTRACT**

In today's era advanced technology is growing faster and faster day by day and Artificial Intelligent is one of among them so we have decided to take use of this technology in our project which name is "JARVIS" AI Desktop voice assistant. The purpose of this project is to complete our daily task with our voice command that's why we are using Python program containing modules like pyphirmata, pywhatkit, smtplib, pyttsx3, wolphramalpha, speech recognition, os module to perform different type of task on desktop eg. play music, open web browser, send emails, and making serial communication with arduino for operations on externally connected devices like lamps, fan etc. The main task of a voice assistant is to minimize the use of input devices like keyboard, mouse, touch pens, etc. This will reduce both the hardware cost and space taken by it.

The most famous application is the amazon "ALEXA" which helps the end user to communicate with the end user with voice and it also responds to the voice commands of the user. Same kind of application is also developed by Google that is "Google Voice Search" which is used for Android Phones.

JARVIS takes the user input in form of voice or text and processes it and returns the output in various forms like action to be performed or the search result is dictated to the end user. JARVIS is developed in Python which can also control any of the regular home appliances with the help of arduino and relays. Because of this it does not required smart devices like smart light, smart fan etc.

**Keywords:** Desktop Assistant, Python, Machine Learning, Text to Speech, Speech to Text, Language Processing, Voice Recognition, Virtual Assistant.

## CONTENTS

Chapter No.	Title	Page No.
	Certificate	i
	Declaration	ii
	Acknowledgement	iii
	Abstract	iv
	List of Figures	vii
	List of Tables	viii
1.	Introduction	1
	1.1 Introduction	1
	1.2 Objective	2
	1.3 Problem Statement	2
	1.4 Methodology	2
2.	Literature Review	3
	2.1 Literature Survey	3
3.	System Development	5
	3.1 System Description	5
	3.2 Circuit Description	8
	3.3 Hardware	9
	3.3.1 Arduino Uno R3	9
	3.3.2 Dual-Channel Relay Module	18
	3.3.3 Light bulb	23
	3.3.4 Arduino Jumper Cables	24
	3.3.5 PCB sample connection	25
	3.3.6 Project Image	25
	3.4 Software	26
	3.4.1 Vs Code Editor	26
	3.4.2 Arduino IDE	27
	3.4.3 Python Programming Language (V 3.8.10)	28
	3.4.4 Python Modules used in Project	31

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
	3.4.5 Arduino Protocol used in Project	39
	3.5 Algorithm	40
	3.6 Flow chart	41
4.	Results, Performance and Analysis	42
	4.1 System Testing	42
	4.2 Applications	43
	4.3 Advantages and Disadvantages	43
5.	Conclusion	44
	5.1 Conclusion	44
	5.2 Future Scope	44
	Reference	46

## LIST OF FIGURES

Figure No.	Title	Page No.
2.1	Literature view	3
3.1	Block Diagram	5
3.2	Class Diagram	6
3.3	Use Class Diagram	6
3.4	Sequence Diagram	7
3.5	Component Diagram	8
3.6	Circuit Diagram	8
3.7	Arduino Board Topology	12
3.8	Arduino Power Tree	13
3.9	Arduino Connector Pinouts	15
3.10	Board Outline & Mounting Holes	17
3.11	Pin Configuration of Atmega 328p IC	18
3.12	Dual-Channel Relay Module	19
3.13	Components present on a 5V Dual Channel Relay Module	21
3.14	Internal Circuit Diagram for Dual-Channel Relay Module	22
3.15	Light Bulb	23
3.16	Arduino Jumper Cables	24
3.17	PCB Sample connection diagram	25
3.18	Project Image	25
3.19	Flowchart of AI Desktop Voice Assistant	41



## LIST OF TABLES

<b>Table No.</b>	<b>Title</b>	<b>Page No.</b>
3.1	Recommended Operating Conditions	11
3.2	Power Consumption	11
3.3	Arduino Board Topology	12
3.4	Arduino Analog Pinouts	16
3.5	Arduino Digital Pinouts	16
3.6	Dual-Channel Relay Module Pinouts	20

## INTRODUCTION

### 1.1 Introduction :

In today's era almost all tasks are digitized. We have Smartphones in our hands and it is nothing less than having the world at your fingertips. These days we aren't even using fingers. We just speak of the task and it is done. There exist systems where we can say Text Dad, "I'll be late today." And the text is sent. That is the task of a Virtual Assistant. It also supports specialized tasks such as booking a flight or finding cheapest book online from various ecommerce sites and then providing an interface to book an order that helps automate search, discovery and online order operations. Virtual Assistants are software programs that help you ease your day to day tasks, such as showing a weather report, creating reminders, making shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. For my project the wake word is JARVIS. We have so many virtual assistants, such as Apple's Siri, Amazon's Alexa and Microsoft's Cortana. For this project, wake word was chosen JARVIS.

This system is designed to be used efficiently on desktops. Personal assistant Software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user. JARVIS is effortless to use. Call the wake word 'JARVIS' followed by the command. And within seconds, it gets executed. Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2022. Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses. This project was started on the premise that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

## **1.2 Objective**

The main objective of the work is to enhance Modern Technology on next level, design and build smart and simple automated desktop assistant which can play audio, control your home appliances, answer questions and engage your favorite services to keep you organized, informed, safe, connected and entertained.

## **1.3 Scope of Work**

To Provide easeful life for human-being especially to handicapped people is the important thing in our work. To reduce human effort in many works like go through walking and switch on or off some appliances are time consuming and irritating for aged and specially abled people. To recover such a condition the 'AI desktop voice assistance' system is used. The Software based system using Arduino in this we have tried to create this project in which voice commands are used to control some appliances and do some work for us.

## **1.4 Methodology**

In order to achieve the objective of the scope few tasks need to be done for the Software of the system and hardware. For the hardware of the system there are two parts which have to be considered. They are the Arduino circuit board and relay which are connected to the external appliances and for the software of the system Specially purposed program is created with Python 3.8.10 modern language of program which perform different types of operations specified in program according to voice commands given to the system.

The coding needed to be testing on simulator software, the purpose of this simulator program is to debug errors. The last part in order to achieve the objective is to test the output of the system. To test the Arduino is by giving appropriate power supply to it.

## LITERATURE REVIEW

### 2.1 Literature Survey :

1. **Abhishek Kumar, Rahul Adhikrao More, Nishant Kumar, and Prof. Amit Kumar Patil (2022):**

In their study titled "Voice Assistant with Home Automation", the authors developed a Python-based voice assistant capable of executing tasks such as sending emails, playing music, and controlling home appliances through Arduino integration. The system utilized speech recognition and text-to-speech modules to interpret user commands, aiming to enhance user convenience by reducing reliance on manual inputs.

2. **Mondal, U. K. & Manedal, J. (2011)**

In their paper presented at the *International Conference on Recent Trends in Information Technology (ICRTIT)*, the authors explored various techniques for voice recognition and signal processing. Their research focused on optimizing the conversion of spoken commands into machine-readable formats and highlighted challenges related to ambient noise and accent variability in voice-based systems.

3. **Thangavel Bhuvaneswari, Venugopal Chitra, and Goh Chee Cheng (2023):**

Their work, "Voice Controlled Home Automation System Design", focuses on assisting individuals with disabilities by providing a voice-controlled system to manage household devices. Utilizing Google Cloud Speech API for voice recognition and Bluetooth for communication, the system allows users to control appliances securely and efficiently through a dedicated mobile application.

4. **Pathak, Pankaj (2012)**

The author, in the *International Journal on Advanced Research in Computer Science*, analyzed the development and practical applications of speech recognition technology. The paper detailed the evolution of speech-to-text engines and emphasized the role of machine learning in enhancing recognition accuracy. It also provided insights into real-time command execution through speech, which is crucial for voice assistant systems like JARVIS.

5. **Shivam Kashyap (2024):**

The project "Implementation of a Voice Controlled Home Automation System using Arduino" describes a portable and user-friendly system that enables users, especially the elderly and disabled, to control home appliances via voice commands. The system integrates voice recognition with Arduino to operate devices like lights and fans, aiming to enhance accessibility and ease of use.

6. **Shailesh D. Arya and Samir Patel (2020):**

Their research, "Implementation of Google Assistant & Amazon Alexa on Raspberry Pi", explores the integration of popular voice assistants on Raspberry Pi platforms. The study demonstrates how these assistants can be configured to control IoT devices, providing users with voice-activated control over various smart home appliances.

## SYSTEM DEVELOPMENTS

The voice module used this system is Google's API i.e. "speech\_recognition". This module is used to recognize the sound waves given by the user as input. This is a loose API this is supplied and supported by Google. This is a totally mild API that facilitates in decreasing the scale of our application.

The design consists of the following:

- Taking voice as a input from user.
- Conversion of the speech into text by the system
- The converted text is then processed to get the desired output.

The text contains some keywords that determine what queries are to be executed. If the keyword did not matched the queries in the program hence the assistant will search it on wolframalpha which is a knowledgebased engine and gives coputational results or assistant will ask the user to speak again. The output which is in the text form is converted to speech and is provided to the user. Speech recognition is used to convert the input voice to text. This text is then sent to the processor, it detects the nature of the command and calls the related code for execution process.

### 3.1 System Description :

The block diagram of "JARVIS" AI Desktop Voice Assistance is shown in figure 3.1.

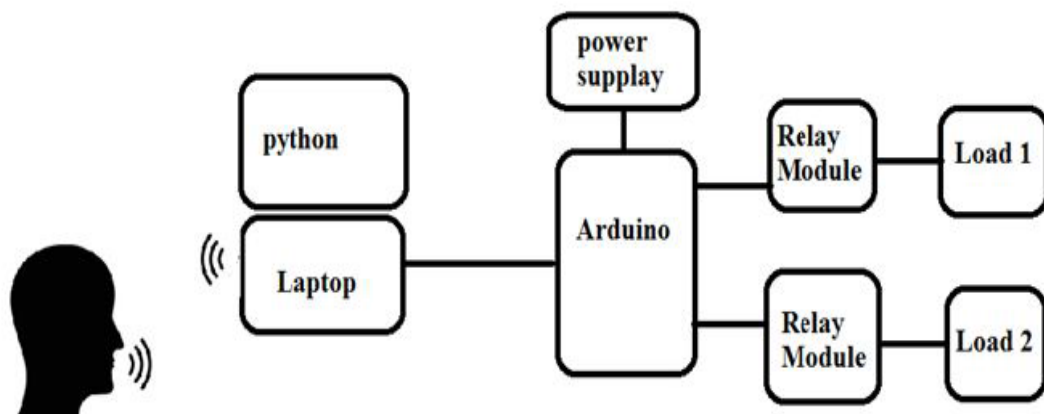


Figure 3.1 Block Diagram of JARVIS AI Desktop Voice assistant

This system receives voice commands from user through microphone and voice converted that signal into voice recognition module and interpret that command and send to the microcontroller and produce control signal according to command gives to the system.

In this project we use Arduino uno R3 in that circuit The Atmega 328PU ic is programmed to perform specific application. In this project we are using A separate laptop/desktop for creating program and run it on the system, Arduino Circuit board, Two relays for switching operation which are connected to the external home appliances like light, fan etc. Data cable for connection between laptop and Arduino circuit, connecting wires, and external dc power adapter for Arduino circuit.

In this project according to block diagram used create program with python language using Software “VS code editor” accordingly to specific purposes of AI desktop voice assistant with that system can receive voice commands from user through the microphone of Laptop device then voice get converted into text with the help of voice recognition module of system This VR is mainly software and hardware based system which is mainly based on Python 3.8.10 and Arduino. The conversion result were sent to the microcontroller of Arduino connected to laptop through data cable or response sent back to the user according to command given to the system. Then signal sent to the relays connected to the Arduino to control the devices.

When user gives voice commands to the system through microphone of laptop System try to listen that voice for recognition according to program created. If voice is recognised then voice is converted into text then system checks that given command is ‘query’ or ‘condition’ if the command is neither query or condition then system search it on web and gives something knowledgeable information about it. If command found query or condition then then system check for it is external or internal operation. If given command is for internal operation then system perform that task and if command is external operation then send it to Arduino to perform that task and control that specific device.

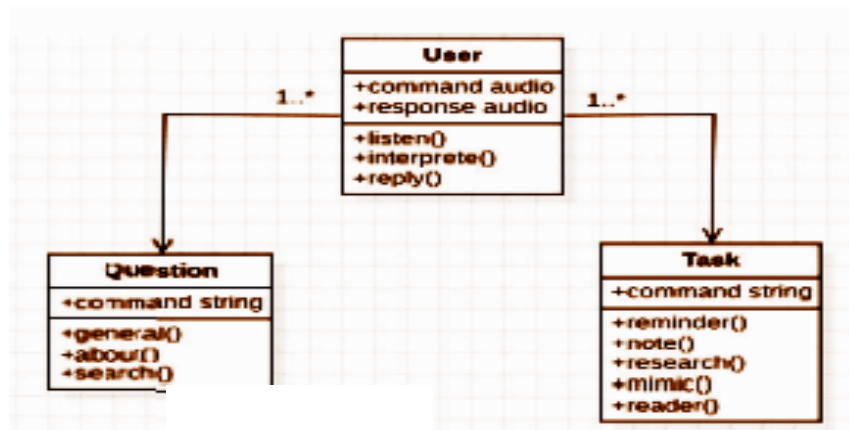


Figure 3.2 : Class Diagram

The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, research and reader. For class diagram refer figure 3.2

Use case diagram :

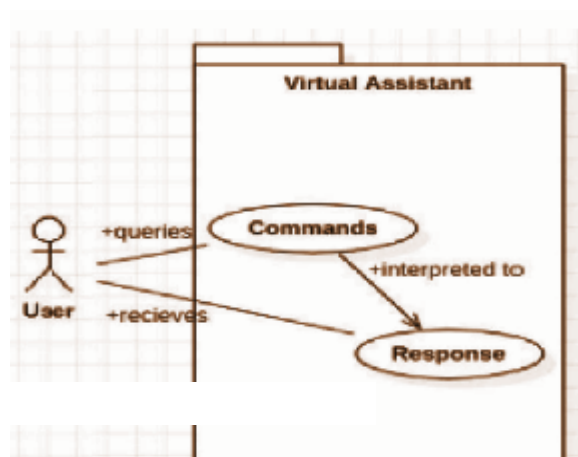


Figure 3.3 : Use Class Diagram

In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user as shown in figure 3.3.



Sequence diagram :

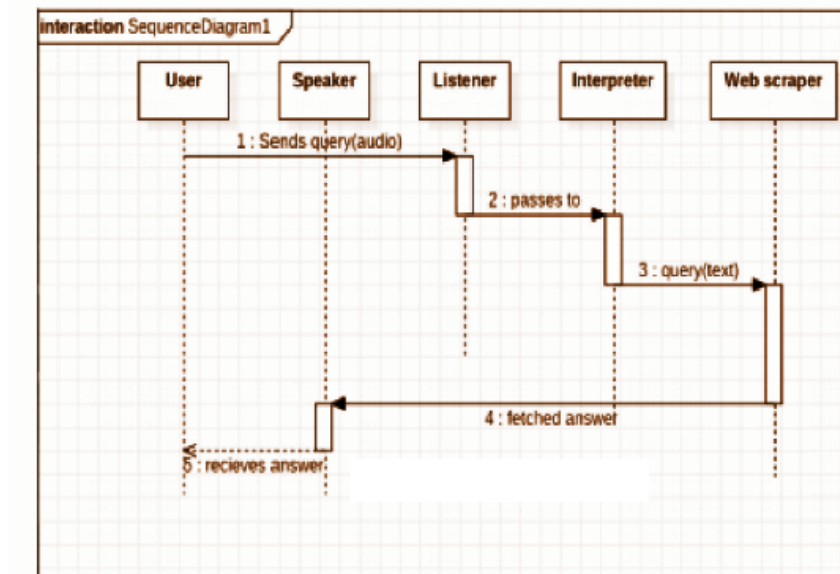


Figure 3.4 : Sequence Diagram

The figure 3.4 showing the sequence diagram that how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to Web scraper. The web scraper searches and finds the answer. It is then sent back to speaker, where it speaks the answer to user.

The user sends command to virtual assistant in audio form. The command is passed to the interpreter. It identifies what the user has asked and directs it to task executer. If the task is missing back about it. The received information is sent back to task and it is accomplished. After execution feedback is sent back to user.

The above diagram shows entities and their relationship for a virtual assistant system. We have a who can have their keys and values. It can be used to store any information about the user. Say, for key “name” value can be “Jim”. For some key’s user might like to keep secure. There he can enable lock and set a password (voice clip).Single user can ask multiple questions. Each question will be given ID to get recognized along with the query and its corresponding answer. User can also be having n number of tasks. These should have their own unique id and status i.e.their current state. A task should also have a priority value and its category whether it is a parent task or child task of an older task.

Component Diagram :

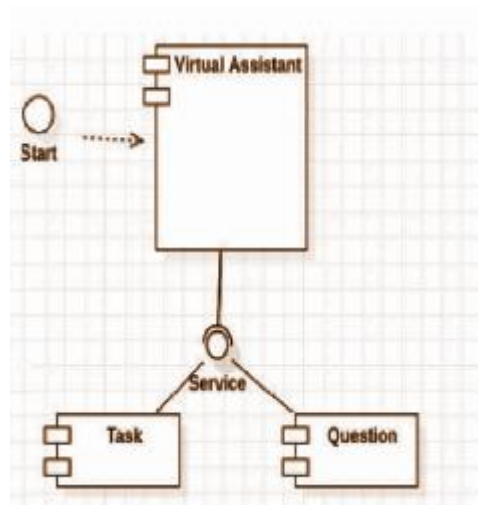


Figure 3.5 : Component Diagram

The main component here is the Virtual Assistant. It provides two specific service, executing Task or Answering your question as shown in figure 3.5.

### 3.2 Circuit Description :

The Circuit Diagram of JARVIS AI Desktop Voice Assistant is as shown in figure 3.6.

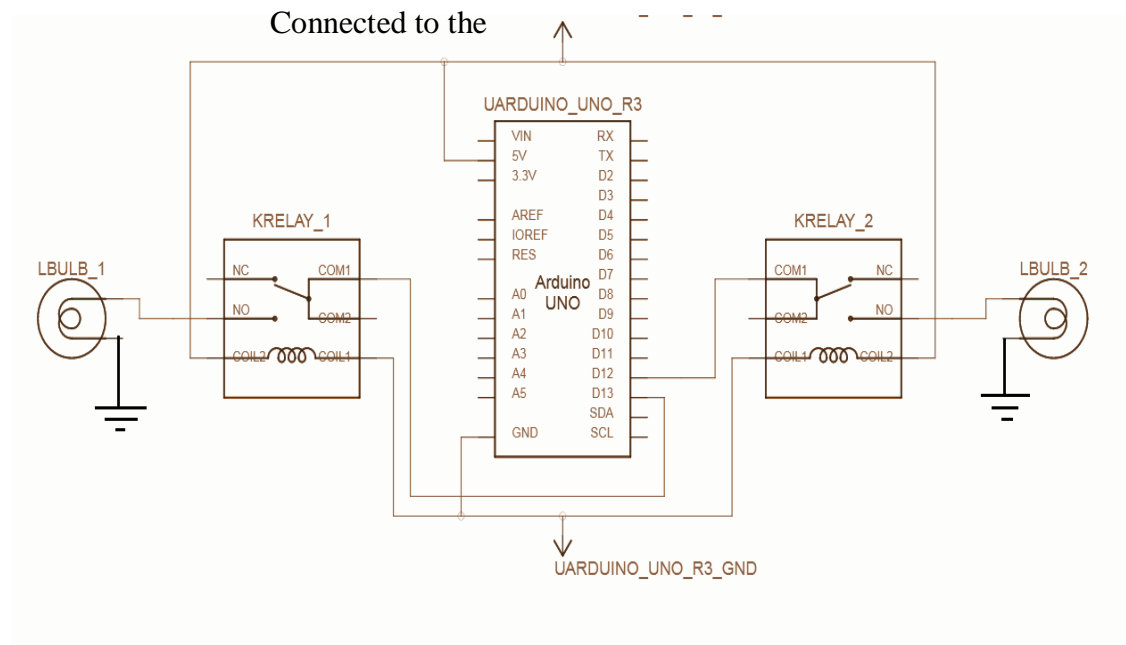


Figure 3.6 : Hardware Circuit Diagram Of ‘AI desktop Voice assistance’

This project is based on the Software Python Language program which control all the action of this project. Relays has a 4 pin one is connected to the +5 volt supply

coming from pin no. 4 of Arduino circuit and other pin no 5 ground also connected to relay circuit. Pin no. Ain 1 Coming from Arduino Connected to the common terminal of relay. So when voice command is given to the system so when command given to the system and command is query or question with external operation decided by program created for specific purpose then signal sent to relay from Arduino and normally closed relay switched to normally open position for turning on the light according to voice command.

In this project Arduino circuit is used for the external operation for externally connected devices like lite, fan and other home appliances. So for the operation of Arduino external 12 volt supply also provided from the adapter.

In this project according to block diagram used create program with python language using Software “VS code editor” accordingly to specific purposes of AI desktop voice assistant with that system can receive voice commands from user through the microphone of Laptop device then voice get converted into text with the help of voice recognition module of system This VR is mainly software and hardware based system which is mainly based on Python 3.8.10 and Arduino. The conversion result were sent to the microcontroller of Arduino connected to laptop through data cable or response sent back to the user according to command given to the system. Then signal sent to the relays connected to the Arduino to control the devices.

### **3.3 Hardware :**

#### **3.3.1 Arduino UNO R3:**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.<sup>[1]</sup> The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license

and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

- **Features of Arduino Uno R3 :**

- **ATMega328P Processor**

- **Memory**

- AVR CPU at up to 16 MHz
      - 32KB Flash
      - 2KB SRAM
      - 1KB EEPROM

- **Security**

- Power On Reset (POR)
      - Brown Out Detection (BOD)

- **Peripherals**

- 2x 8-bit Timer/Counter with a dedicated period register and compare channels
      - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
      - 1x USART with fractional baud rate generator and start-of-frame detection
      - 1x controller/peripheral Serial Peripheral Interface (SPI)
      - 1x Dual mode controller/peripheral I2C
      - 1x Analog Comparator (AC) with a scalable reference input

- Watchdog Timer with separate on-chip oscillator
- Six PWM channels
- Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
  - 8-bit AVR® RISC-based microcontroller
- **Memory**
  - 16 KB ISP Flash
  - 512B EEPROM
  - 512B SRAM
  - debugWIRE interface for on-chip debugging and programming
- **Power**
  - 2.7-5.5 volts

● **Ratings :**

**Recommended Operating Conditions :**

Table 3.1 : Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40°F)	85 °C ( 185°F)

**NOTE:** In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected due to the extreme temperature conditions

**Power Consumption :**

Table 3.2 : Power Consumption

Symbol	Description	Min	Type	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

**Functional Overview :**

- **Arduino Board Topology :**

Top View

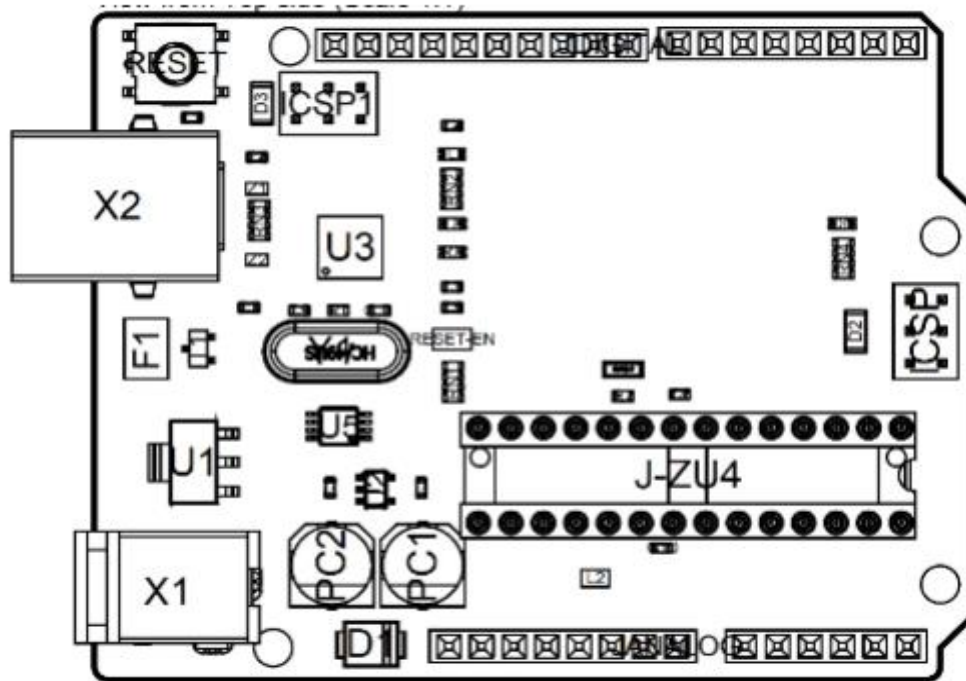


Figure 3.7 : Arduino Board Topology

Table 3.3 : Arduino Board Topology

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

- **Power Tree :**

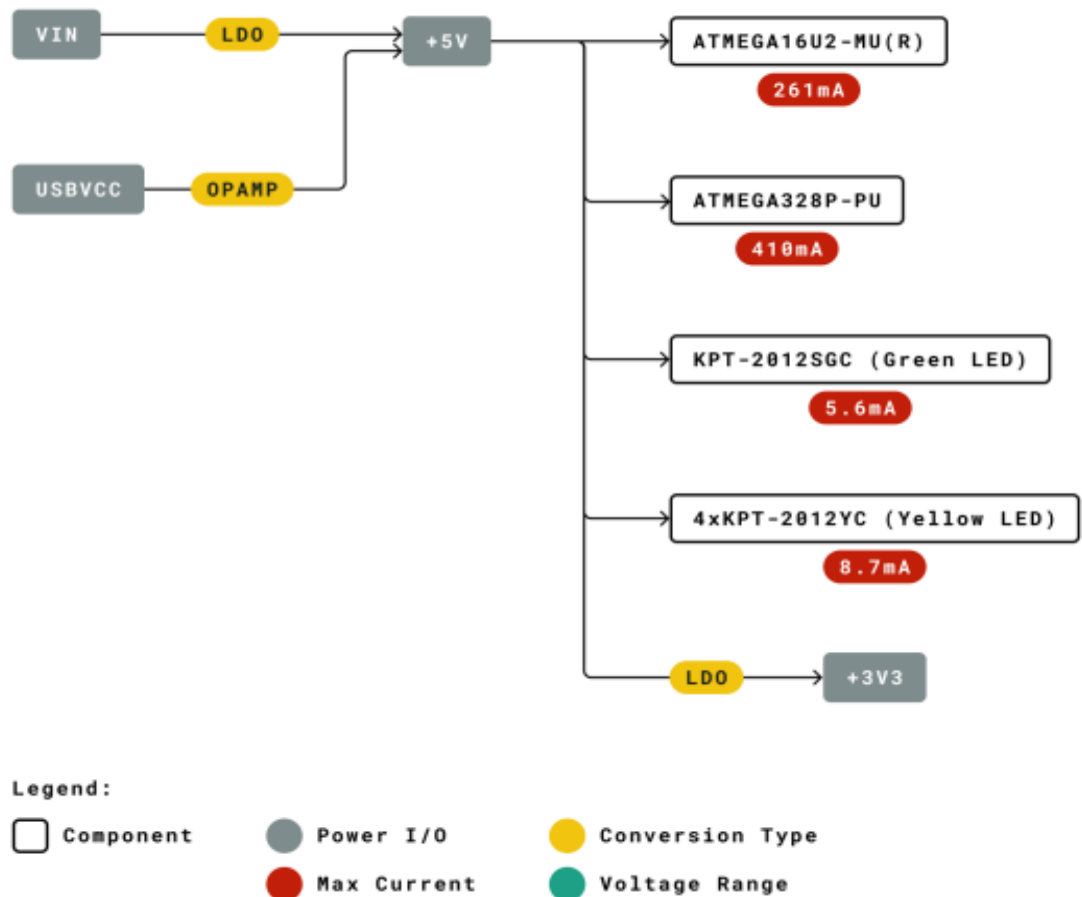


Figure 3.8 : Arduino Power Tree

- **Processor :**

The Main Processor is a ATmega328P running at up to 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

### Board Operation :

- **Getting Started - IDE :**

If you want to program your Arduino UNO while offline you need to install the Arduino Desktop IDE [1] To connect the Arduino UNO to your computer, you'll need a Micro-B USB cable. This also provides power to the board, as indicated by the LED.

- **Getting Started - Arduino Web Editor :**

All Arduino boards, including this one, work out-of-the-box on the Arduino Web Editor [2], by just installing a simple plugin. The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow [3] to start coding on the browser and upload your sketches onto your board.

- **Getting Started - Arduino IoT Cloud :**

All Arduino IoT enabled products are supported on Arduino IoT Cloud which allows you to Log, graph and analyze sensor data, trigger events, and automate your home or business.

- **Sample Sketches :**

Sample sketches for the Arduino XXX can be found either in the “Examples” menu in the Arduino IDE or in the “Documentation” section of the Arduino Pro website [4]

- **Online Resources :**

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on ProjectHub [5], the Arduino Library Reference [6] and the online store [7] where you will be able to complement your board with sensors, actuators and more

- **Board Recovery :**

All Arduino boards have a built-in bootloader which allows flashing the board via USB. In case a sketch locks up the processor and the board is not reachable anymore via USB it is possible to enter bootloader mode by double-tapping the reset button right after power up.

## **Connector Pinouts :**



It comprises 14-digit I/O pins. From these pins, 6-pins can be utilized like PWM outputs. This board includes 14 digital input/output pins, Analog inputs-6, a USB connection, quartz crystal-16 MHz, a power jack, a USB connection, resonator-16Mhz, a power jack, an ICSP header and a RST button.

The figure 3.9 shows the connector pinouts of Arduino Uno R3.

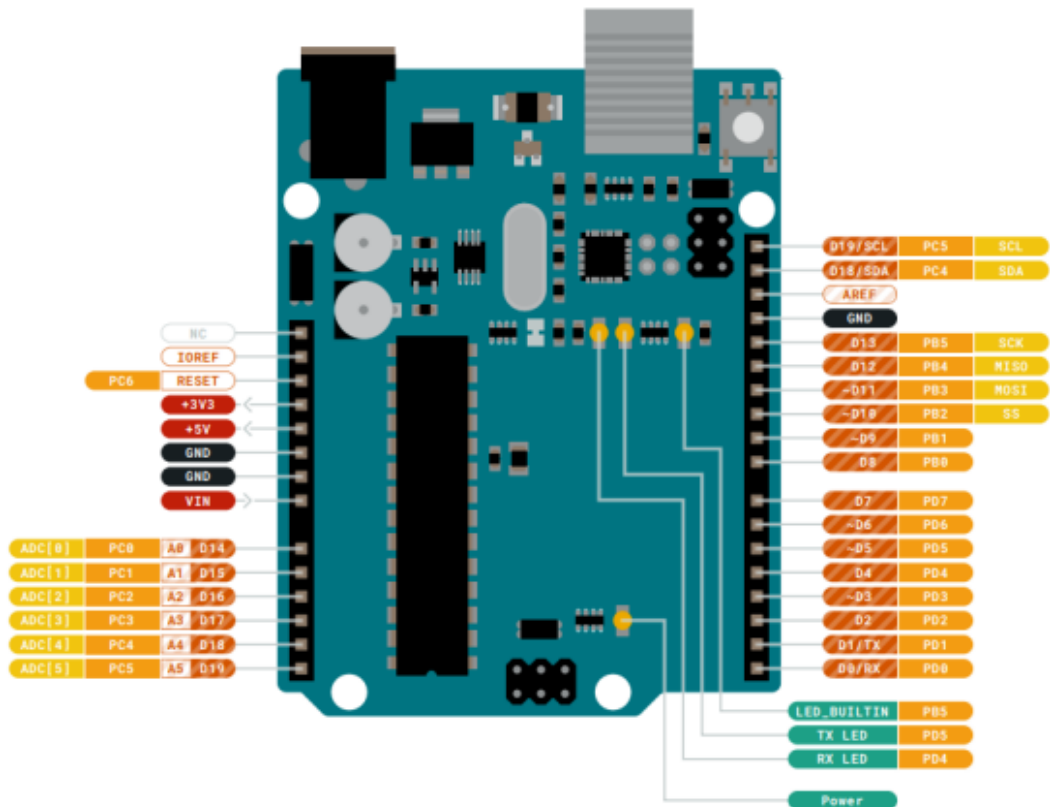


Figure 3.9 : Arduino Connector Pinouts

Description of analog and digital pinouts of Arduino Uno R3 is given below. Refer table 3.4 for analog pinout description and table 3.5 for digital pinout description.

- **Analog Pinouts :**

The following table 3.4 shows analog pinouts of Arduino Uno R3.

Table 3.4 : Arduino Analog Pinouts

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

- **Digital Pinouts :**

The following table 3.5 shows digital pinouts of Arduino Uno R3.

Table 3.5 : Arduino Digital Pinouts

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select

12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Digital	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

### Board Outline & Mounting Holes :

The following figure 3.10 shows the board outline and mounting hole of the Arduino Uno R3.

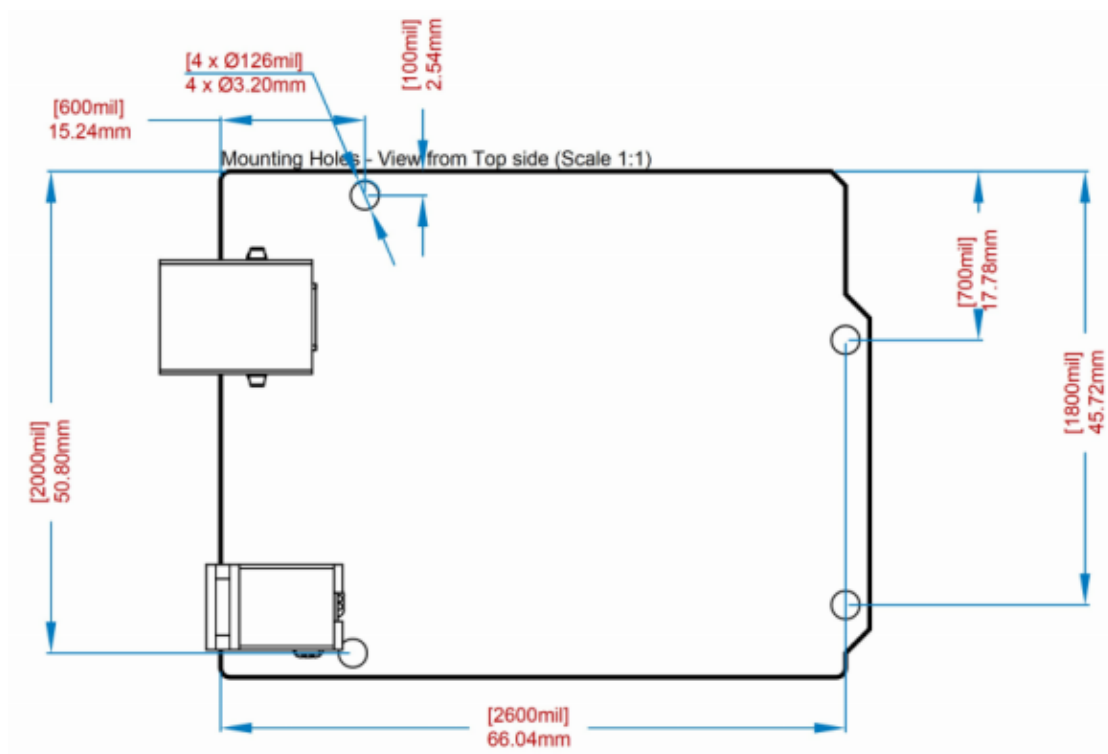


Figure 3.10: Board Outline & Mounting Holes

### Atmega 328P IC :

A microcontroller (MCU for microcontroller unit) is a small computer on a single metal-oxide-semiconductor (MOS) integrated circuit (IC) chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips.

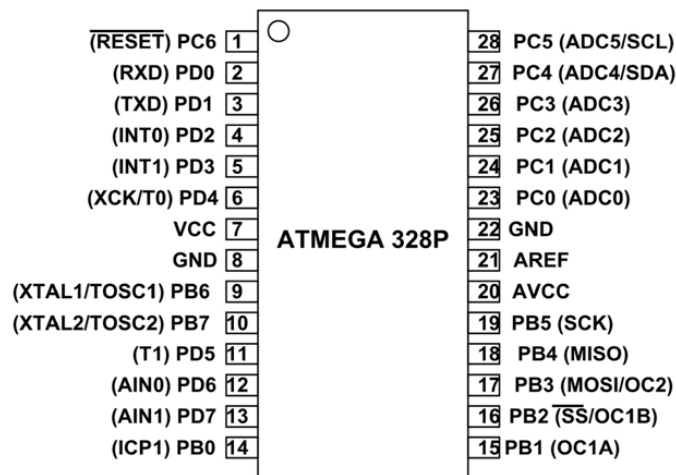


Figure 3.11 : Pin Configuration of Atmega 328p IC

The ATmega328 is a single-chip microcontroller. ATmega328 is commonly used in many projects and autonomous systems where a simple, low-powered, low-cost microcontroller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino Uno, Arduino Pro Mini and Arduino Nano models.

### 3.3.2 Dual-Channel Relays Module :

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were

used extensively in telephone exchanges and early computers to perform logical operations.

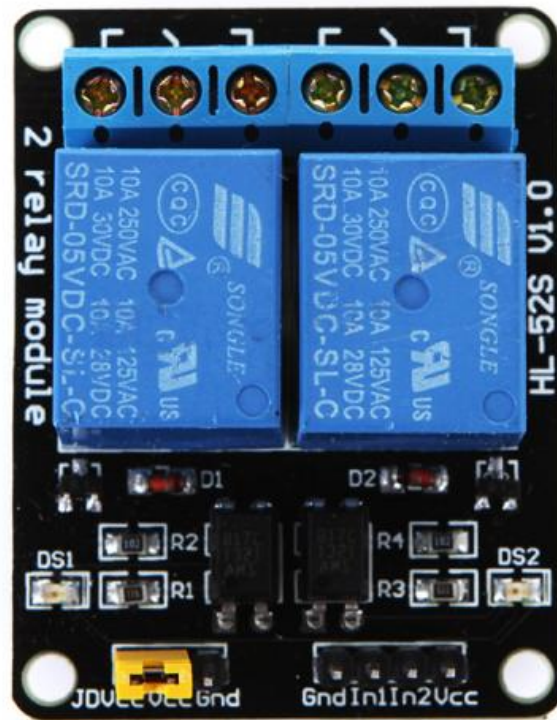


Figure 3.12 : Dual-Channel Relay Module

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays". Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts. Magnetic latching relays can have either single or dual coils. On a single coil device, the relay will operate in one direction when power is applied with one polarity, and will reset when the polarity is reversed. On a dual coil device, when polarized voltage is applied to the reset coil the contacts will transition. AC controlled magnetic latch relays have single coils that employ steering diodes to differentiate between operate and reset commands.

**Dual-Channel Relay Module Pinouts :**

The pinouts description of dual channel relay module is as shown in table 3.6.

Table 3.6 : Dual-Channel Relay Module Pinouts

Pin Number	Pin Name	Description
1	JD-VCC	Input for isolated power supply for relay coils
2	VCC	Input for directly powering the relay coils
3	GND	Input ground reference
4	GND	Input ground reference
5	IN1	Input to activate the first relay
6	IN2	Input to activate the second relay
7	VCC	VCC to power the optocouplers, coil drivers, and associated circuitry

**Dual-Channel Relay Module Specifications :**

- Supply voltage – 3.75V to 6V
- Trigger current – 5mA
- Current when relay is active - ~70mA (single), ~140mA (both)
- Relay maximum contact voltage – 250VAC, 30VDC
- Relay maximum current – 10A

**Components present on a 5V Dual Channel Relay Module :**

The following are the major components as shown in figure 3.13 that would be found on a Dual channel Relay module. 5V Relay, Optocoupler, Diodes, Transistors, Resistors, LEDs, Male Headers, 3-pin Terminal connectors, etc.

The dual-channel relay module contains switching relays and the associated drive circuitry to make it easy to integrate relays into a project powered by a microcontroller.

On the left are two terminal blocks, which are used to connect mains wires to the module without soldering.

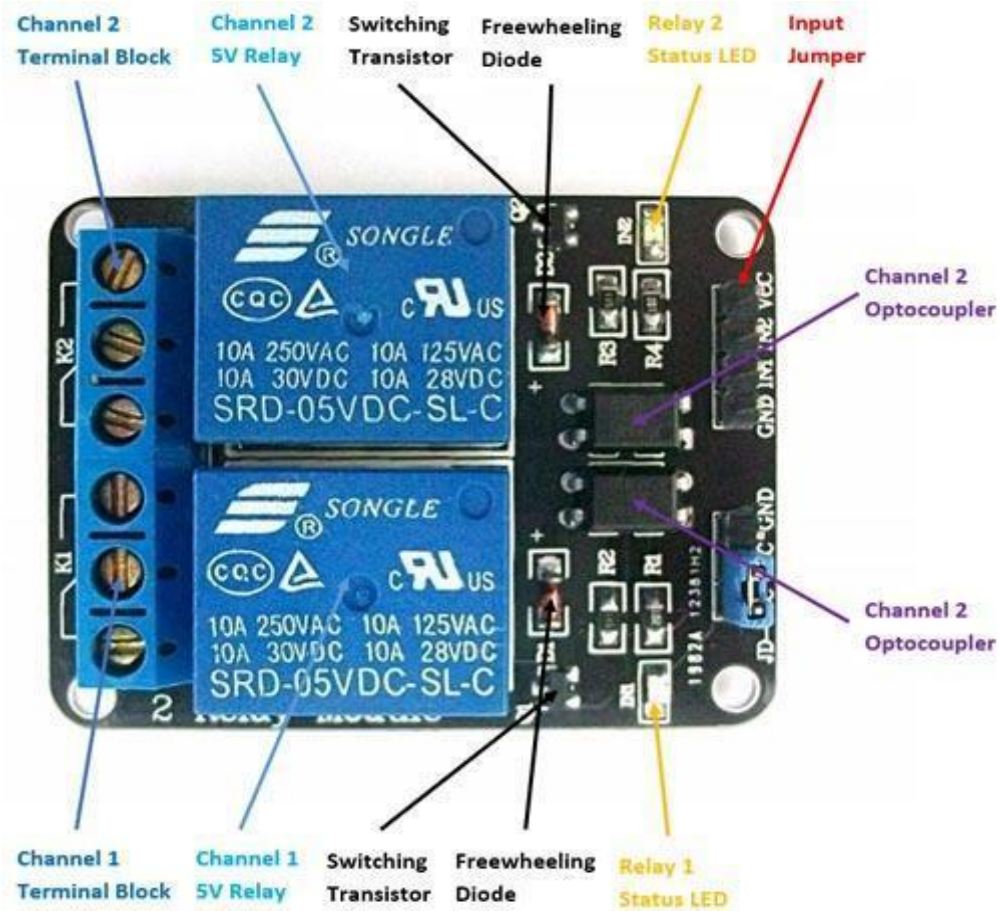


Figure 3.13 : Components present on a 5V Dual Channel Relay Module

Next, come to the two relays. As marked on the body of the relay, the relay coil is rated for 5VDC, and the contacts are rated for 10A at 250VAC or 30VDC, or 125VAC or 28VDC.

The switching transistors amplify the signal from the inputs enough to drive the relay. The freewheeling diodes prevent voltage spikes across the switching transistors. The status LEDs turn on when the relay is active and indicate switching.

The optocouplers are used to provide additional isolation between the input and the relays. The isolation can be selected using the VCC/JD VCC jumper.

The input jumper has two input and two power pins and can be easily used to connect to jumper wires and other microcontrollers and sensors.

### Internal Circuit Diagram for Dual-Channel Relay Module :

Following figure 3.14 shows the internal circuit diagram of dual channel relay module.

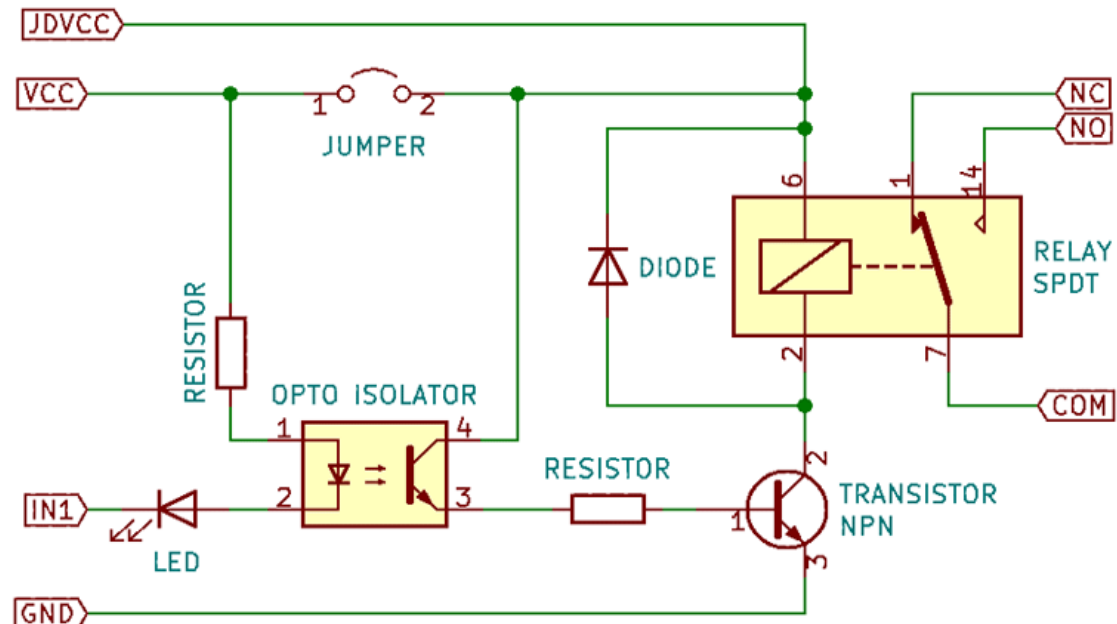


Figure 3.14 : Internal Circuit Diagram for Dual-Channel Relay Module

This same circuit is replicated twice on the module. There are several differences when compared to the single-channel relay module. The first is that the optocoupler and relay can have a separate power supply from the input. This is selected using the JD-VCC jumper – if the jumper is shorted, then the relay coil and optocoupler output are connected to the signal VCC. If the jumper is left open, a separate power supply can be provided to the relay coil through the JD-VCC pin.

The second is that the input is active low, meaning that the relay coil is activated only when the input is low. This is because the optocoupler input and the indicator LED are chained to VCC. When the input is high, the voltage difference across the chain is 0V and no current can flow, but when the input is low, supply voltage appears across the chain and the optocoupler output conducts, supplying current to the base of the drive transistor, turning the relay on.

### 3.3.3 Light Bulbs :



Light bulbs have a very simple structure. At the base, they have two metal contacts, which connect to the ends of an electrical circuit. The metal contacts are attached to two stiff wires, which are attached to a thin metal filament. The filament sits in the middle of the bulb, held up by a glass mount. The wires and the filament are housed in a glass bulb, which is filled with an inert gas, such as argon. When the bulb is hooked up to a power supply, an electric current flows from one contact to the other, through the wires and the filament. Electric current in a solid conductor is the mass movement of free electrons (electrons that are not tightly bound to an atom) from a negatively charged area to a positively charged area. As the electrons zip along through the filament, they are constantly bumping into the atoms that make up the filament. The energy of each impact vibrates an atom -- in other words, the current heats the atoms up.



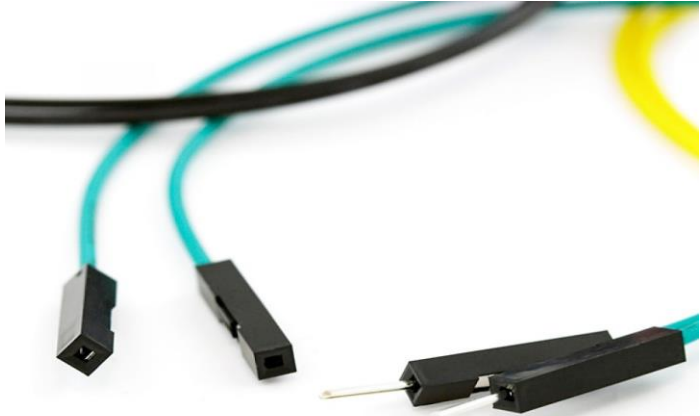
Figure 3.15 : Light Bulb

A thinner conductor heats up more easily than a thicker conductor because it is more resistant to the movement of electrons. Bound electrons in the vibrating atoms may be boosted temporarily to a higher energy level. When they fall back to their normal levels, the electrons release the extra energy in the form of photons. Metal atoms release mostly infrared light photons, which are invisible to the human eye. But if they are heated to a high enough level -- around 4,000 degrees Fahrenheit (2,200 degrees C) in the case of a light bulb -- they will emit a good deal of visible light. The filament in a light bulb is made of a long, incredibly thin length of tungsten metal. In a typical 60-watt bulb, the tungsten filament is about 6.5 feet (2 meters) long but only one-hundredth of an inch thick. The tungsten is arranged in a double coil in order to fit it all in a small space. That is, the filament is wound up to make one coil, and then this coil is wound to make a larger coil. In a 60-watt bulb, the coil is less than an inch long. Tungsten is used in nearly all incandescent light bulbs because it is an ideal filament material.

### **3.3.4 Arduino Jumper Cables :**

In most cases thermocouples are connected to other process instruments by connecting cables. Connecting cables are divided into compensating cables and

extension cables. These can again be divided into solid and stranded conductors. Extension cables are connecting cables with conductors made of thermocouple material, i.e. of the same material as the thermocouple itself. Extension cables therefore are subject to the same accuracy limits as the thermocouples.



Extension cables are marked with an „X“ behind the identification letter for the thermocouple. For example, „KX“ means that the cable is an extension cable / insulated thermocouple wire for the thermocouple type K (NiCr-Ni). Compensation cables are connecting cables with conductors made of materials which only in a limited temperature range have the same thermoelectric properties as the thermocouple itself. Compensating cables are marked with a „C“ behind the identification letter for the thermocouple. For example, „KC“ means that the cable is a compensating cable for the thermocouple type K. The insulation of the conductors as well as of the sheaths can be made of a variety of materials. The following criteria must be taken into consideration when selecting the appropriate material: chemical and mechanical resistance against the ambient conditions, temperature range, insulation resistance, flexibility. Apart from that there is a great number of cable types available as compensating cables or extension cables as well as those with more than two inner conductors. Many special designs are available upon request, some of them on a short term basis.

### **3.3.5 PCB Sample Connection :**

Figure 3.17 shows the sample connections between bulbs, relay and arduino together.

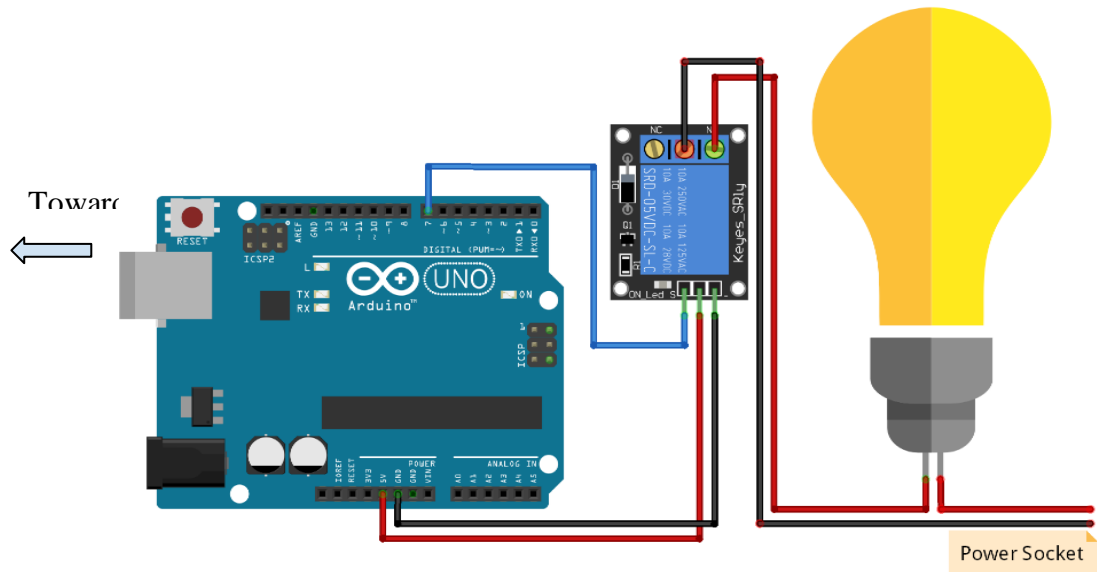


Figure 3.17 : PCB Sample connection diagram

### 3.3.6 Project Image :

The following figure 3.18 is a captured image showing the hardware of JARVIS.

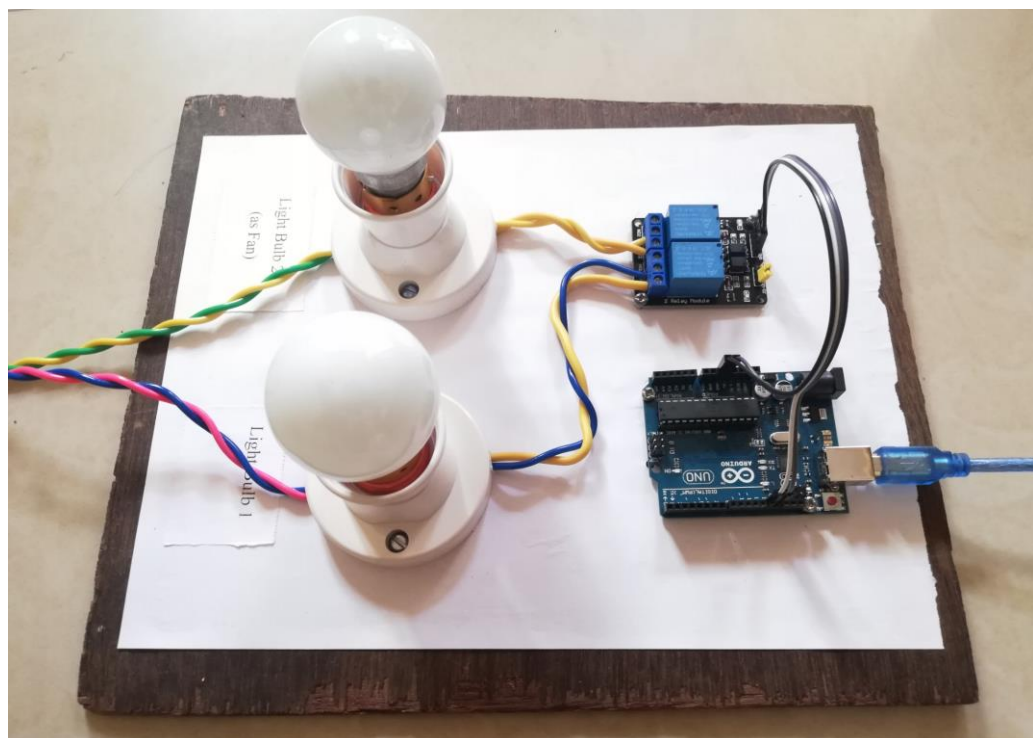


Figure 3.18 : Project Image

### 3.4 Software :

#### 3.4.1 VS Code Editor :

Study about VSCode tells us Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include Services for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter. On November 18, 2015, the source of Visual Studio Code was released under the MIT License, and made available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web. Microsoft has released most of Visual Studio Code's source code on GitHub under the permissive MIT License, while the releases by Microsoft are proprietary freeware. In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++ and Fortran. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports many programming languages and a set of features that differs per language. Unwanted files and folders can be excluded

from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where you can access version control settings and view changes made to the current project. To use the feature you must link Visual Studio Code to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows you to create repositories as well as to make push and pull requests directly from the Visual Studio Code program.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. Due to the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected.

### **3.4.2 Arduino IDE:**

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on our computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board we can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package. The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even our smart-phone or our TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a huge variety of Arduino-based projects. Recently, arduino is compatible for almost all AVR microcontrollers. For programming ESP8266 (32 bit microcontroller with embedded WI-FI) just need to update the existing arduino IDE, it installs the additional boards as nodemcu. Rest of the commands and programming method is the same as we did with UNO.

### **3.4.3 Python Programming Language (Version 3.8.10) :**

The programming language Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to ABC capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL). (However, van Rossum stepped down as leader on July 12, 2018.). Python was named after the BBC TV show Monty Python's Flying Circus.

Python 2.0 was released on October 16, 2000, with many major new features, including a cycle-detecting garbage collector (in addition to reference counting) for memory management and support for Unicode. However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process.

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008 after a long period of testing. Many of its major features have also been backported to the backwards-compatible, though now-unsupported, Python 2.6 and 2.7.

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages. Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, Data science etc. Python has a lot of libraries for every need of this project. For JARVIS, libraries used are speech recognition to recognize voice, Pyttsx3 for text to speech, selenium for web automation etc.

Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

- **Features of Python :**

There are many features in Python, some of which are following :

i) Easy To Code: Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.

ii) Free and Open Source: Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword. Since it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

- iii) Object-Oriented Language: One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.
- iv) GUI Programming Support: Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.
- v) High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.
- vi) Extensible feature: Python is a Extensible language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language.
- vii) Python is Portable language: Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.
- viii) Python is Integrated language: Python is also an Integrated language because we can easily integrated python with other languages like c, c++, etc.
- ix) Interpreted Language: Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.
- x) Large Standard Library: Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.
- xi) Dynamically Typed Language: Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.



### **3.4.4 Python Modules Used In Project JARVIS :**

#### **i) Smtplib :**

An SMTP instance encapsulates an SMTP connection. It has methods that support a full repertoire of SMTP and ESMTP operations. If the optional host and port parameters are given, the SMTP connect() method is called with those parameters during initialization. If specified, local\_hostname is used as the FQDN of the local host in the HELO/EHLO command. Otherwise, the local hostname is found using socket.getfqdn(). If the connect() call returns anything other than a success code, an SMTPConnectError is raised. The optional timeout parameter specifies a timeout in seconds for blocking operations like the connection attempt (if not specified, the global default timeout setting will be used). If the timeout expires, TimeoutError is raised. The optional source\_address parameter allows binding to some specific source address in a machine with multiple network interfaces, and/or to some specific source TCP port. It takes a 2-tuple (host, port), for the socket to bind to as its source address before connecting. If omitted (or if host or port are "" and/or 0 respectively) the OS default behavior will be used. For normal use, you should only require the initialization/connect, sendmail(), and SMTP.quit() methods. An example is included below.

#### **ii) Pyttsx3 :**

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male which is provided by "sapi5" for windows.

It supports three TTS engines :

- sapi5 – SAPI5 on Windows
- nsss – NSSpeechSynthesizer on Mac OS X
- espeak – eSpeak on every other platform

Pyttsx3 stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech

engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

**iii) Speech Recognition :** Speech Input Using a Microphone and Translation of Speech to Text Allow Adjusting for Ambient Noise: Since the surrounding noise varies, we must allow the program a second or too to adjust the energy threshold of recording so it is adjusted according to the external noise level.

Speech to text translation: This is done with the help of Google Speech Recognition. This requires an active internet connection to work. However, there are certain offline Recognition systems such as PocketSphinx, but have a very rigorous installation process that requires several dependencies. Google Speech Recognition is one of the easiest to use. This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

**iv) WolframAlpha :** WolframAlpha is a computational knowledge engine and answer engine developed by Wolfram Research. It answers factual queries directly by computing the answer from externally sourced data.

WolframAlpha was released on May 18, 2009, and is based on Wolfram's earlier product Wolfram Mathematica, a computational platform for calculation, visualization, and statistics capabilities. Additional data is gathered from both academic and commercial websites such as the CIA's The World Factbook, the United States Geological Survey, a Cornell University Library publication called All About

Birds, Chambers Biographical Dictionary, Dow Jones, the Catalog of Life, CrunchBase, Best Buy, and the FAA.

Users submit queries and computation requests via a text field. WolframAlpha then computes answers and relevant visualizations from a knowledge base of curated, structured data that come from other sites and books. It is able to respond to particularly phrased natural language fact-based questions, or more complex questions. It displays its "Input interpretation" of such a question, using standardized phrases. Mathematical

symbolism can be parsed by the engine, which responds with numerical and statistical results.

WolframAlpha is written in the Wolfram Language, a general multi-paradigm programming language, and implemented in Mathematica and ran on more than 10,000 CPUs as of 2009.

WolframAlpha has been used to power some searches in the Microsoft Bing and DuckDuckGo search engines but is not currently used. For factual question answering, it is sometimes queried by Apple's Siri and Amazon Alexa for math and science queries.

#### **v) OS module :**

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see `open()`, if you want to manipulate paths, see the `os.path` module, and if you want to read all the lines in all the files on the command line see the `fileinput` module. For creating temporary files and directories see the `tempfile` module, and for high-level file and directory handling see the `shutil` module.

#### **Notes on the availability of these functions:**

- The design of all built-in operating system dependent modules of Python is such that as long as the same functionality is available, it uses the same interface; for example, the function `os.stat(path)` returns stat information about path in the same format (which happens to have originated with the POSIX interface).
- Extensions peculiar to a particular operating system are also available through the `os` module, but using them is of course a threat to portability.
- All functions accepting path or file names accept both bytes and string objects, and result in an object of the same type, if a path or file name is returned.
- On VxWorks, `os.popen`, `os.fork`, `os.execv` and `os.spawn*p*` are not supported.

In Python, file names, command line arguments, and environment variables are represented using the string type. On some systems, decoding these strings to and from bytes is necessary before passing them to the operating system. Python uses the filesystem encoding and error handler to perform this conversion (see `sys.getfilesystemencoding()`).

The filesystem encoding and error handler are configured at Python startup by the `PyConfig_Read()` function: see `filesystem_encoding` and `filesystem_errors` members of `PyConfig`.

Changed in version 3.1: On some systems, conversion using the file system encoding may fail. In this case, Python uses the `surrogateescape` encoding error handler, which means that undecodable bytes are replaced by a Unicode character `U+DCxx` on decoding, and these are again translated to the original byte on encoding.

The file system encoding must guarantee to successfully decode all bytes below 128. If the file system encoding fails to provide this guarantee, API functions can raise `UnicodeError`.

#### **vi) Pyfirmata :**

PyFirmata is basically a prebuilt library package of python program which can be installed in Arduino to allow serial communication between a python script on any computer and an Arduino. This python package can give access to read and write any pin on the Arduino.

#### **vii) Pywhatkit :**

Python offers numerous inbuilt libraries to ease our work. Among them `pywhatkit` is a Python library for sending WhatsApp messages at a certain time, it has several other features too.

**Following are some features of `pywhatkit` module:**

- Sending Message to a WhatsApp Group or Contact
- Sending Image to a WhatsApp Group or Contact
- Converting an Image to ASCII Art
- Converting a String to Handwriting

- Playing YouTube Videos
- Sending Mails with HTML Code
- Install and Use

#### 1. Send Whatsapp Messages:

pywhatkit module which utilises the WhatsApp webpage to automate messages sending to any number on WhatsApp. But make sure that you have logged into your WhatsApp in your browser.

#### 2. Play a YouTube video:

Function `pywhatkit.playonyt()`, opens the YouTube in your default browser and plays the video you mentioned in the function. If you pass the topic name as parameter, it plays the random video on that topic. On passing the URL of the video as the parameter, it open that exact video.

#### 3. Perform Google Search:

You can perform a Google search using the following simple command. It opens your browser and searches for the topic you have given in your code.

#### 4. Get information on particular topic:

We can get brief information on a particular topic. We can also limit the number of lines to be printed. Also, make sure that you are searching for the topics that are available on Wikipedia.

#### **viii) Wikipedia :**

The Internet is the single largest source of information, and therefore it is important to know how to fetch data from various sources. And with Wikipedia being one of the largest and most popular sources for information on the Internet.

Wikipedia is a multilingual online encyclopedia created and maintained as an open collaboration project by a community of volunteer editors using a wiki-based editing system.

Python's Wikipedia module to fetch a variety of information from the Wikipedia website. Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia.

#### **ix) Request :**

Requests allows you to send HTTP/1.1 requests extremely easily. There's no need to manually add query strings to your URLs, or to form-encode your POST data. Keep-alive and HTTP connection pooling are 100% automatic, thanks to urllib3.

#### **Features of Request Module:**

- Keep-Alive & Connection Pooling
- International Domains and URLs
- Sessions with Cookie Persistence
- Browser-style SSL Verification
- Automatic Content Decoding
- Basic/Digest Authentication
- Elegant Key/Value Cookies
- Automatic Decompression
- Unicode Response Bodies
- HTTP(S) Proxy Support
- Multipart File Uploads
- Streaming Downloads
- Connection Timeouts
- Chunked Requests
- .netrc Support

#### **x) Fast2sms API (SMS Gateway API) :**

API or Application Programming Interface is a set of rules and specifications through which software programs communicate with each other. Basically API is used for sending different messages to everyone not manually but automatically.

#### **Features of Fast2SMS :**

- Bulk SMS – Bulk SMS refers to business sending SMS to one or more recipients and can scale up to millions of persons at the same time. It refers to sending large number of messages to a predefined set of customers.
- Quick SMS feature – Fast2SMS provides a very unique and useful feature which is not available in any other bulk SMS service provider. You can send SMS to DND and Non DND numbers even if you are not registered in the DLT portal.
- API SMS – API refers to Application Programming Interface. SMS API integration is the fastest and simplest way to send automated messages directly from your platform. Fast2SMS provides an API for bulk SMS, which ensures security and it is a very reliable source of sending data.
- Add contacts with QR – Fast2SMS offers the facility of adding contacts with QR. QR stands for quick response. The main benefit of QR code is it takes very less space for storing information. It is a form of barcode and has become quite popular these days. In this feature you can add contacts by creating a QR code. Simply you need to add your group name and group URL name and then click on create QR. The QR code gets created.

#### **xi) OpenWeatherMap API :**

OpenWeatherMap provides a range of weather-related products in a variable combination of depth and steps of measurement to millions of clients globally. The product range includes current, historical and forecasted weather data with the granularity as high as 1 minute. The length of the nowcast reaches 2 hours, short-term forecast reaches 16 days and long-term forecast can reach up to 1 year length. Historical weather data goes over 40 years deep. OpenWeather also provides a range of weather maps and weather alert services.

In 2015, Google chose OpenWeatherMap as a weather data provider for its bid-by-weather script in Google Ads, which serves the ads based on the local weather conditions, such as temperature, humidity, and cloudiness. Same year, Google published documentation on how to use OpenWeather data to display weather conditions on Google Maps.

In 2020, Samsung included OpenWeatherMap into their Galaxy Watch Studio as a weather data provider for those willing to develop applications for Galaxy Watch. In 2020, OpenWeatherMap has released its weather application for iOS and Android.

OpenWeather provides data for weather risk management on the individual agreement basis to the industries like energy, agriculture, transportation, construction, municipalities, travel, food processors, retail sales and real estate. OpenWeather also operates under the terms of Creative Commons Attribution-ShareAlike license providing free access to the APIs that include current weather, a minutely forecast for 1 hour, hourly forecast for 48 days, 3-hour forecast for 5 days, daily forecast for 7 days, short-term history, weather maps, alerts, geocoding, air quality weather triggers and weather widgets. The projects with a higher demand of loading, may obtain an extended service on the basis of paid subscription.

Access current weather data for any location on Earth including over 200,000 cities! We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations. Data is available in JSON, XML, or HTML format.

### **3.4.5 Arduino Protocol Used In Project JARVIS :**

#### **i) Firmata protocol :**

Firmata is a protocol for communicating with microcontrollers from software on a computer (or smartphone/tablet, etc). The protocol can be implemented in firmware on any microcontroller architecture as well as software on any computer software package.

Firmata is based on the midi message format in that command bytes are 8 bits and data bytes are 7 bits. For example the midi Channel Pressure (Command: 0xD0) message is 2 bytes long, in Firmata the Command 0xD0 is used to enable reporting for a digital port (collection of 8 pins). Both the midi and Firmata versions are 2 bytes long, but the meaning is obviously different. In Firmata, the number of bytes in a message must conform with the corresponding midi message. Midi System Exclusive (Sysex) messages however, can be any length and are therefore used most prominently throughout the Firmata protocol.



This repository contains documentation of the Firmata protocol. The core of the protocol is described in the `protocol.md` file. Feature-specific documentation is described in individual markdown files (`i2c.md`, `accelStepperFirmata.md`, `servos.md`, etc). Files added to the `proposals` directory are proposals for new features that have not yet been finalized. See `feature-registry.md` for the full list of documented firmata features.

The Firmata protocol could theoretically be implemented for any microcontroller platform. Currently however, the most complete implementation is for Arduino (including Arduino-compatible microcontrollers). Here are the known Firmata microcontroller platform implementations:

- Firmata for Arduino
- Firmata for Spark.io

Firmata is a generic protocol for communicating with microcontrollers from software on a host computer. It is intended to work with any host computer software package. Right now there is a matching object in a number of languages. It is easy to add objects for other software to use this protocol. Basically, this firmware establishes a protocol for talking to the Arduino from the host software. The aim is to allow people to completely control the Arduino from software on the host computer.

There are several client libraries. These are libraries that implement the Firmata protocol in order to communicate (from a computer, smartphone or tablet for example) with Firmata firmware running on a microcontroller platform. The following is a list of Firmata client library implementations:

- python - pymata4, pymata-express, pyduino, python-firmata, pyFirmata
- perl - perl-firmata, rx-firmata
- ruby - firmata, rufinol
- javascript - firmata, johnny-five
- java - Firmata, firmata4j, FiloFirmata, diozero-provider-firmata
- .NET - Arduino, iot
- PHP - carica-firmata, phpmake\_firmata
- iOS - iosfirmata
- Android/Kotlin - android-firmata

### **3.5 Algorithms :**

Step 1. Start and Initializing the program

Step 2. Main infinite loop for taking voice commands continuously.

Step 3. Listening commands

Step 4. Voice to text conversion

Step 5. Command check in queries and conditions.

Step 6. Perform specific task according to the commands or search on web

Step 7. After performing the tasks, go back/returns to the listening function again.

Step8. End of the program.

### 3.6 Flowchart :

The following figure 3.19 is the flowchart of JARVIS AI Desktop Voice Assistant.

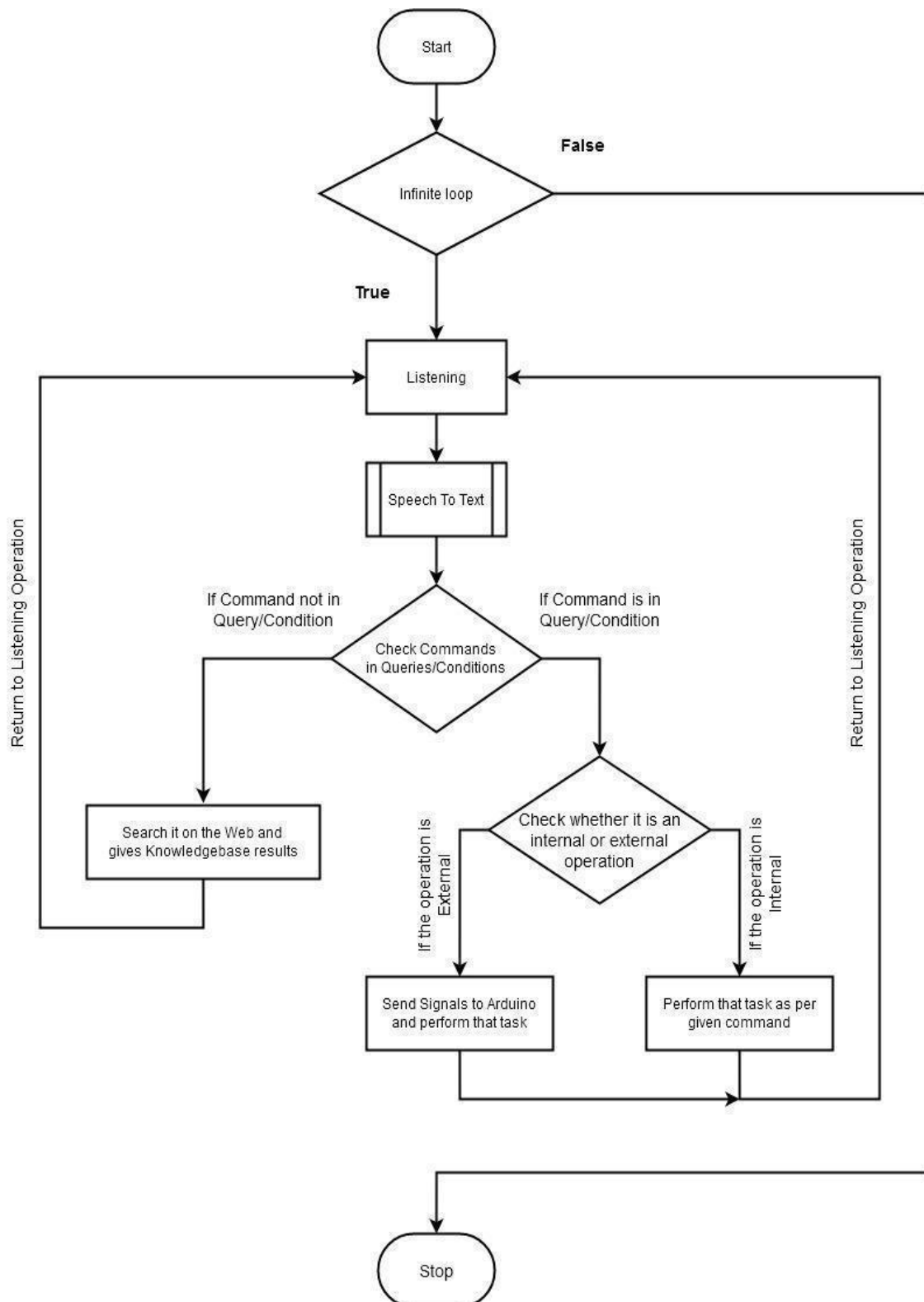


Figure 3.19 : Flowchart of AI Desktop Voice Assistant

## RESULTS, PERFORMANCE AND ANALYSIS

Virtual assistant is a less time consuming. Virtual assistant is a software that understands verbal or written commands and complete task assigned by client. Virtual assistant use natural language processing to match user voice or text input with executable commands. With the help of virtual assistant you able to run your machine like laptop or PC's on your own command. It is the fast process, therefore it saves time. Virtual assistant is working for you at set times, so always available to you and able to adapt to changing needs quickly. Virtual assistant will be available to you and, should their workload enable, help others too, such as family and colleagues.

### 4.1 System Testing :

We did software and hardware testing on this system and we found that the system is working normally and giving appropriate results.

In the hardware testing we have tested the system using various devices like light bulbs and fan. The Arduino, Relays Module and connected devices are working perfectly fine for the user commands and giving the appropriate results.

In software testing we have tested the system as on following parameters.

**Test Title:** Response Time

**Description:** Time is very critical in a voice-based system. As we are not typing inputs, we are speaking them. The system must also reply in a moment. User must get instant response of the query made. But in some cases we found the system taking little bit more time to give response, the reason is this system works online so it requires a good internet connection to get instant response of the query. In some cases as per the network speed, the response time may vary.

**Test Title:** Accuracy

**Description:** A virtual assistant system is mainly used to get precise answers to any question asked. Getting answer in a moment is of no use if the answer is not correct. Accuracy is of utmost importance in a virtual assistant system and we found that the system gives accurate response for the clear voice inputs.

## **4.2 Applications :**

What can this A.I. Assistant do for you?

1. It can control your devices like Light, Fan, and any other home appliances.
2. It can send emails on your behalf.
3. It can play music for you.
4. It can do Wikipedia searches for you.
5. It is capable of opening websites like Google, Youtube, etc., in a web browser.
6. It is capable of opening your code editor or IDE with a single voice command.

## **4.3 Advantages And Disadvantages :**

### **4.3.1 Advantages :**

1. Easy to use
2. Reduce Human Efforts
3. Works Efficiently
4. We can control so many things only by using voice commands
5. This is future

### **4.3.2 Disadvantages :**

1. Sometimes Lack of Accuracy and Misinterpretation
2. Background Noise Interference

## CONCLUSION

The project we have worked on is “JARVIS” AI Desktop voice assistant. Jarvis is a wake up word which is abbreviation of ‘Just A Rather Very Intelligent System’. This Project based on the artificial intelligent system which receives voice commands from user and perform specific task according to python program created on desktop. In python program different types of modules are used for different types of application user want to execute on desktop, arduino uno r3 with relays is used to connect external home appliances for their switching purposes with voice command.

### 5.1 CONCLUSION

In this project report we have discussed AI Desktop support voice Assistant developed using python. It was a wonderful learning experience for us while working on this project this project took us through the various phases of project development and take us a real insight into the world of software Engineering. Our purpose behind working on this project was to make human life more easeful for aged and physically enabled peoples who can do many work without help of others only with speaking some commands and our dream come in true with completion of this project which fulfilled all our ideas which was discussed by us before some months. This assistant currently works online and performs basic tasks like weather updates, stream music, search Wikipedia, open desktop applications, etc. The functionality of the current system is limited to working online only. The upcoming updates of this assistant will have machine learning incorporated in the system which will result in better suggestions with IoT to control the nearby devices similar to what Amazon’s Alexa does. Now just with his voice user can search web pages, open youtube, check daily weather condition , play music, send emails or messages to others without using hands.

### 5.2 FUTURE SCOPE

The virtual assistants which are currently available are fast and responsive but we still have to go a long way. The understanding and reliability of the current systems need to be improved a lot. The assistants available nowadays are still not reliable in critical scenarios. The future of these assistants will have the virtual assistants

incorporated with Artificial Intelligence which includes Machine Learning, Neural Networks, etc. and IoT. With the incorporation of these technologies, we will be able to achieve new heights. What the virtual assistants can achieve is much beyond what we have achieved till now. Most of us have seen Jarvis, that is a virtual assistant developed by iron man which is although fictional but this has set new standards of what we can achieve using voice-activated virtual assistants.

## REFERENCES

1. MONDAL, U. K. MANEDAL, J. Voice recognition technique and voice processing. Recent trends in information technology (ICRTIT)2011.
2. Pankaj Pathak, "Speech recognition technology, International journal on advanced research computer science,2012"
3. Davis, S. Mermelstein, P.(1980) "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences".
4. <https://www.python.org/downloads/release/python-3810/>
5. <https://docs.arduino.cc/hardware/uno-rev3>