# AI BASED WEARABLE BLIND VISION ASSISTANT

A

Major Project Report

Submitted in the Partial fulfilment of the

Academic Requirements

For the Award of the Degree of

**Bachelor of Technology**

**In**

**Electronics and Communication Engineering**

**By**


**I. PRANAV DEEP** 15AG1A0455


Under the esteemed guidance of

**Dr. V.S.S.N. SRINIVASA BABA**

**Professor & HOD of ECE**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

## ACE Engineering College

(Sponsored by Yadala Satyanarayana Memorial Educational Society, Hyderabad)

**Approved by AICTE & Affiliated to JNTUH**

**NBA and NAAC 'A' grade Accredited B. Tech courses: ECE, EEE, CSE & MECH**

**Ankushapur (V), Ghatkesar (M), R.R. Dist. -501301**

2018-2019

# ACE Engineering College

Approved by AICTE & Affiliated to JNTUH

NBA and NAAC 'A' grade Accredited B. Tech courses: ECE, EEE, CSE & MECH

Ankushapur (V), Ghatkesar (M), R.R. Dist -501301

---

## CERTIFICATE

This is to certify that the major project entitled "AI BASED WEARABLE BLIND VISION ASSISTANT" done by

**I.PRANAV DEEP** **15AG1A0455**

Of Department **of Electronics and Communication Engineering**, is a record of bonafide work carried out by them. This major project is done as a partial fulfillment of obtaining **Bachelor of Technology** Degree to be awarded by **Jawaharlal Nehru Technological University, Hyderabad,** during the academic year 2018-2019.

**Dr. V. S. S. N. SRINIVASA BABA**
**Professor & HOD of ECE**

**(MAJOR PROJECT SUPERVISOR)**

**Dr. V. S. S. N. SRINIVASA BABA**
**Professor & HOD of ECE**

**EXTERNAL SUPERVISOR**

# ACKNOWLEDGEMENT

Hard work, commitment and team spirit are the basic ingredients for the success of any group task. Be it in education, sports, arena, battlefield or even in our lives, similarly this project is the result of contribution from the students and staff. While doing this project we have come across many difficulties but finally we could complete it with the guidance and suggestions from the following intelligentsia who gave generously of their time and expertise.

We take the opportunity to thank one and all who have helped in making this project possible. We are thankful to **JNTUH** for giving us this opportunity to work on a project as a part of our curriculum.

We would like to convey our heart full thanks to **Dr. B. L. RAJU, principal of the college**, for the wonderful encouragement given to us to move ahead the execution of this project.

We are grateful to thank **Dr. V. S. S. N. SRINIVASA BABA, Head of the Dept. of ECE,** for this thought, provoking suggestions and constant encouragement that led us to this success.

We are very happy for being guided by **Dr. V. S. S. N. SRINIVASA BABA, Professor and Head of the Dept. of ECE**, for his able guidance given to us to complete our technical work successfully.

We wish to express our deep sense of gratitude to our **project coordinator, Mr. B. GIRIRAJU** of ECE Department, for his valuable guidance throughout this project.

Above all, we are very much thankful to the management of ACE Engineering College which was established by the high profiled intellectuals for the cause of technical education in modern era. We wish that ACE sooner should become a deemed university and produce uncountable young engineers and present them to the modern technical world

With regards

**I.PRANAV DEEP (15AG1A0455)**

# Tables of contents

# List of Figures

# ABSTRACT


Visually impaired people face a lot of difficulties in their daily life. The proposed solution aims at reducing those difficulties. We intend to propose a solution in the form of a wearable device, that can be worn by a blind person in order to perceive their world in a better way using Computer Vision technology.

The evolution of AI and Computer Vision techniques is empowering the blind overcome their hurdles. This paper is based on a research project-“Headset for the blind”. It proposes a system for the visually impaired, in the form of a wearable device to assist them.

The proposed system aims at developing a VR like headset for visually impaired people. Its functionalities include identification of objects, people, environment etc. The device’s input and output are completely speech signals. Several computer vision techniques like Face Recognition, Object detection, Environment perception, Photo-OCR etc are proposed to be integrated with a voice based personal assistant that would understand the user’s commands, process and respond appropriately. This will help the visually impaired person to manage day-to-day activities better and to navigate his/her surroundings.

The hardware section of this solution is based on Raspberrypi- which is an onboard computer. It has a few input devices attached like Camera, Microphone, reset button and output device being headphones/speaker.

Our solution embeds emerging technologies like Internet of Things (IOT), Computer Vision, Artificial Intelligence, Speech Recognition etc. Using these technological features, we intend to develop a wearable headset.



***Keywords—Computer Vision, Artificial Intelligence, Object Recognition, Face Recognition, IOT, Wearable Headset, Speech Recognition, Indoor Navigation***

# CHAPTER-1

## 1.1. INTRODUCTION

Visually impaired people face a lot of difficulties in their daily life. A survey by WHO estimated that there are 39 million blind people across the globe, whose quality of life is limited since they need to depend on others for their navigation, of which over 15 million are from India [a]. Blind people require an escort all the time. Our proposed solution reduces the dependency of a blind person on his escort, thus improving his quality of life style. Independent mobility is one of the most pressing problems faced by the blind. blind and 246 million have low vision. However only an insignificant number of users use technological travel aids. There are many factors that contribute this as technology does not operate in isolation; it must be considered within the broader context. One of the most important drawbacks is that almost all the systems use sensor devices.

The objects can be identified using sensor components. The usage of sensors is expensive and unsuitable off late. The proposed work is an attempt to recognize objects, recognize people, perceive environment, plan their navigation using modern AI-Computer Vision techniques. Our solution intends to answer basic questions a visually impaired person gets, which are “Who is it”, “What is it”, “Where am I”, “How to move” etc. The solution proposes a low-cost wearable device (an embedded system designed with AI features) using an on-board computing device (Raspberry Pi). The concept of minimum hardware made it possible for us to integrate the embedded system into a wearable headset. The input and output are totally based on audio (speech signals).

Our solution embeds emerging technologies like Internet of Things (IOT), Computer Vision, Artificial Intelligence, Speech Recognition etc. Using these technological features, we intend to develop a wearable headset.The solution is

broadly classified into 3 main sections: Computer Vision, Embedded system design, mechanical design and their integrated implementation. This book also covers the implementation-both theoretical and experimental approaches of each sub-section and their relevant results, code snippets, flow-charts.

## 1.2. PROPOSED SYSTEM

This system works based on Internet of Things and Artificial Intelligence. It mainly consists of a camera, microphone, push button, speaker/headphone. The controller unit is based on an on-board computer- a Raspberrypi. Ultrasonic sensor detects the distance of an obstacle by sending sonar waves and calculating the duration of reflection of the sonar wave. Based on some mathematical computation this duration computes the distance. We set certain threshold values for each ultrasonic sensor to detect the presence of an obstacle for detecting obstacles.

The proposed system consists of three major components namely- mechanical, embedded and software.

### a) *Mechanical Design:*

**Fig1.1:**This is a CAD modelled on Solid Works inspired from a generic VR headset. Mr. Nikhil Nandepu converted our paper made design into a CAD model . It is a 3-D printable design with dimensions: 200*200*277 mm. The design citation can be viewed and downloaded from [14].

The proposed system advocates usage of lightweight materials like ABS, Resin, etc. However, the design can be fabricated using MDF wood, Plywood also.

### b) *Embedded Design:*

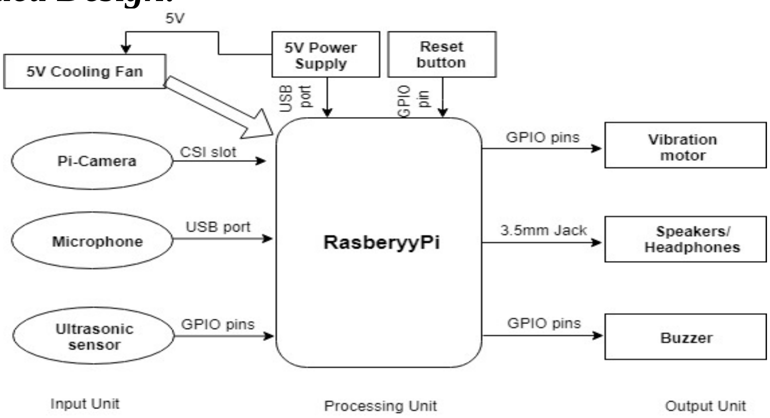



Fig1.2.: Embedded design

Like any system, the proposed system consists of three major parts namely:
**Input** – Microphone, Camera, Ultrasonic sensor, reset button
**Processor** – Raspberry-Pi (ARM based) (PI)
**Output** – Speakers/Headphones, Vibration Motor, Buzzer
Input and output of the system are in audio form. The AI processing requires high speed broadband and in cases where processing takes time, the on-board buzzer, vibration motor and Ultrasonic sensors will help the user and his surroundings in alerting about the obstacles in his/her pathway especially during navigation.

## c) Software Design

The proposed software design can further be divided into 2 sub categories: *embedded firmware* and AI based *Computer Vision algorithm*
The major parts of the firmware interfacing are:

*Speech-Text (STT) conversion*
*Text-Speech (TTS) conversion*
*OpenCV for camera interfacing*
*GPIO library for interfacing with transducers like motors, sensors etc*

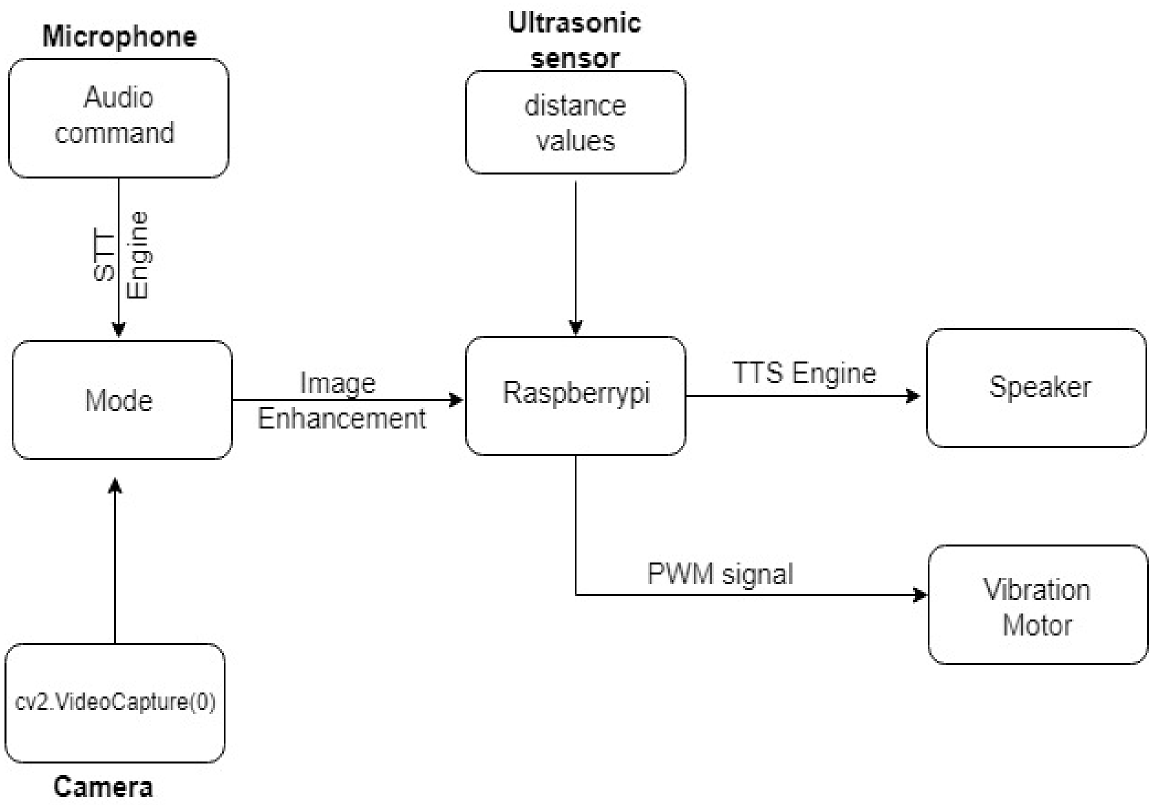



Fig1.3.: Software design

# CHAPTER 2

## 2.1. WORKING PRINCIPLE

Our system consists of sensors, camera and mic in order to take input from the user in various forms. To detect a person/object in front of it, an ultrasonic sensor is used. When there is an object/person in its radar, ultrasonic sensor sends a PWM pulse to the controller and controller sends a pulse to the buzzer. Based on the speech input, the mode of the algorithm is determined.

Eg: If input command==”face”, then face-recognition() will be executed

Similar is the case for object detection and other features as well. In order to detect obstacles an ultrasonic sensor is used.

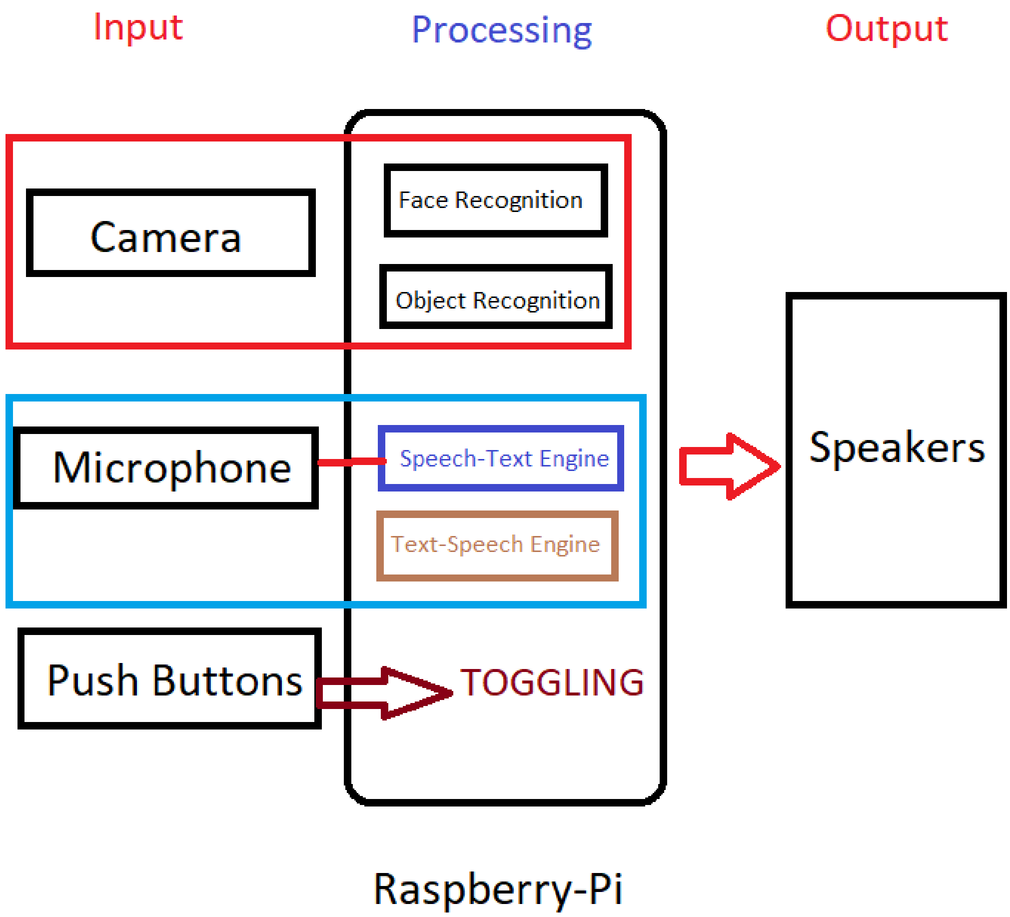
The ultrasonic sensor sends sonar waves and receives the reflected waves. Based on the duration of the reflected wave signal, the controlled is programmed to compute the distance of the obstacle. Then, we set a threshold level of distance.

Eg: Front ultrasonic distance <15, then call the buzzer()

The camera view of the person can always be viewed on any screen by the escort using VNC server.

The wearable headset is a carefully designed solution such that it serves a major purpose of assisting a visually impaired person.

## 2.2. **BLOCK DIAGRAM**




The Block diagram shows the different components used in the smart dustbin system. The important blocks are:

1) Camera, Mic, Push Button
2) Ultrasonic Sensor
3) Raspberrypi
4) Speakers
5) Image Processing
6) AI- Speech Recognition

## 2.3 Power supply

The system we are using requires a 5V power supply. There are various ways we can supply power to our system namely:

The Raspberry Pi 3 is powered by a +5.1V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what you connect to it. We have found that purchasing a 2.5A power supply from a reputable retailer will provide you with ample power to run your Raspberry Pi

Typically, the model B uses between 700-1000mA depending on what peripherals are connected; the model A can use as little as 500mA with no peripherals attached. The maximum power the Raspberry Pi can use is 1 Amp. If you need to connect a USB device that will take the power requirements above 1 Amp, then you must connect it to an externally-powered USB hub.

The power requirements of the Raspberry Pi increase as you make use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the camera module requires 250mA, and keyboards and mice can take as little as 100mA or over 1000mA! Check the power rating of the devices you plan to connect to the Pi and purchase a power supply accordingly.

**Backpowering**

Backpowering occurs when USB hubs do not provide a diode to stop the hub from powering against the host computer. Other hubs will provide as much power as you want out each port. Please also be aware that some hubs will backfeed the Raspberry Pi. This means that the hubs will power the Raspberry Pi through its USB cable input cable, without the need for a separate micro-USB power cable, and bypass the voltage protection. If you are using a hub that backfeeds to the Raspberry Pi and the hub experiences a power surge, your Raspberry Pi could potentially be damaged.

## 2.4. PIN DIAGRAM OF RASPBERRYPI

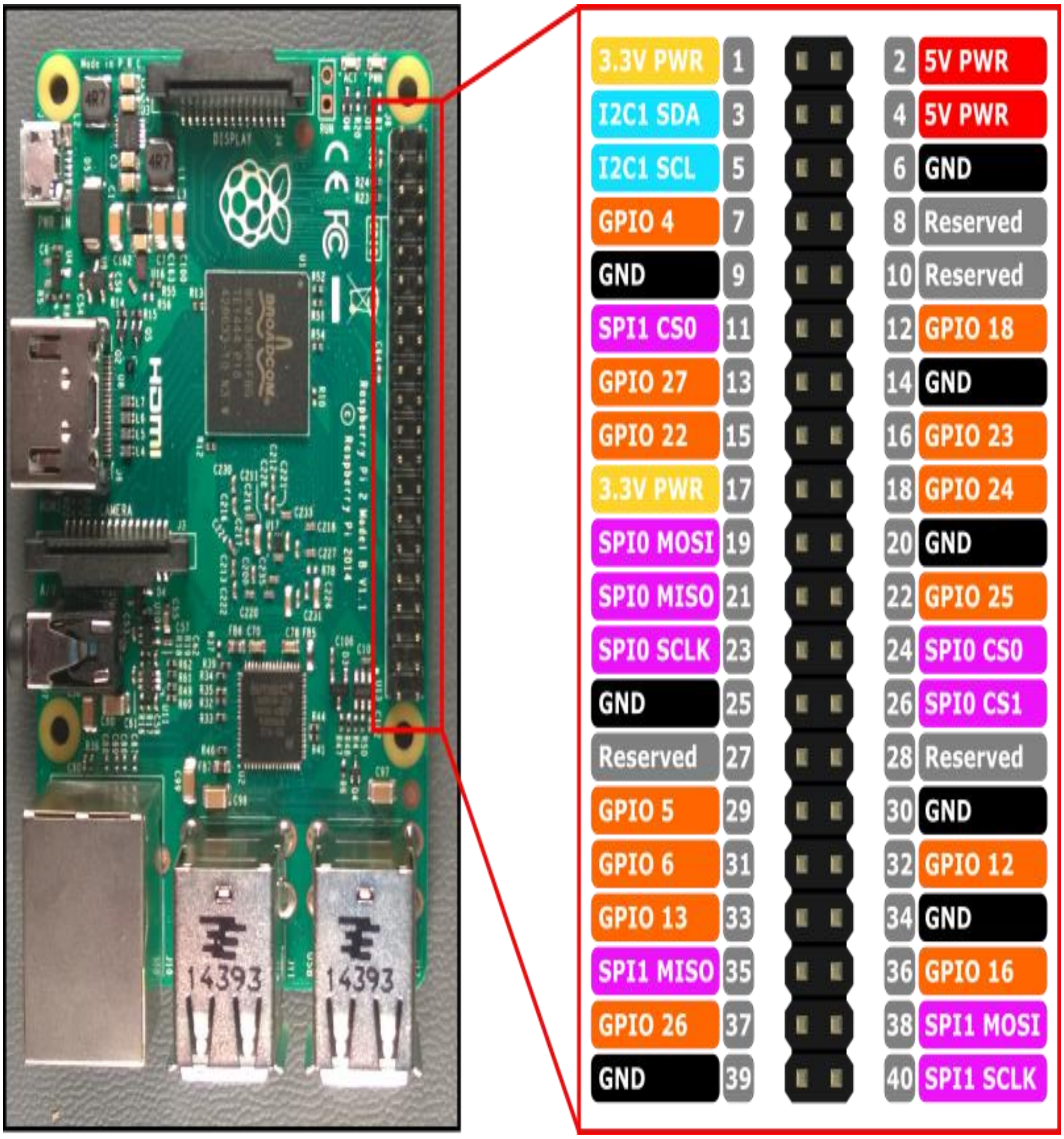



Fig 2.1. : Pin diagram of raspberry pi

# CHAPTER 3

## 3.1. RESISTORS

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element.

In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can dissipate many watts of electrical power as heat, may be used as part of motor controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity.

Resistors are common elements of electrical networks and electronic circuits and are ubiquitous in electronic equipment. Practical resistors as discrete components can be composed of various compounds and forms. Resistors are also implemented within integrated circuits.

The electrical function of a resistor is specified by its resistance: common commercial resistors are manufactured over a range of more than nine orders of magnitude. The nominal value of the resistance falls within the manufacturing tolerance, indicated on the component.

Fig.3.1. A Typical Resistor

Resistors are used for many purposes. A few examples include delimit electric current, voltage division, heat generation, matching and loading circuits, control gain, and fix time constants. They are commercially available with resistance values over a range of more than nine orders of magnitude. They can be used to as electric brakes to dissipate kinetic energy from trains, or be smaller than a square millimeter for electronics.

Fig.3.2. Standard Resistor Symbols

## Composition Types of Resistor

**Carbon Resistors** are the most common type of Composition Resistors. Carbon resistors are a cheap general-purpose resistor used in electrical and electronic circuits. Their resistive element is manufactured from a mixture of finely ground carbon dust or graphite (similar to pencil lead) and a non-conducting ceramic (clay) powder to bind it all together.

Fig.3.3. Carbon Resistor

The ratio of carbon dust to ceramic (conductor to insulator) determines the overall resistive value of the mixture and the higher the ratio of carbon, the lower the overall resistance. The mixture is moulded into a cylindrical shape with metal wires or leads are attached to each end to provide the electrical connection as shown, before being coated with an outer insulating material and colour coded markings to denote its resistive value.

## Carbon Resistor

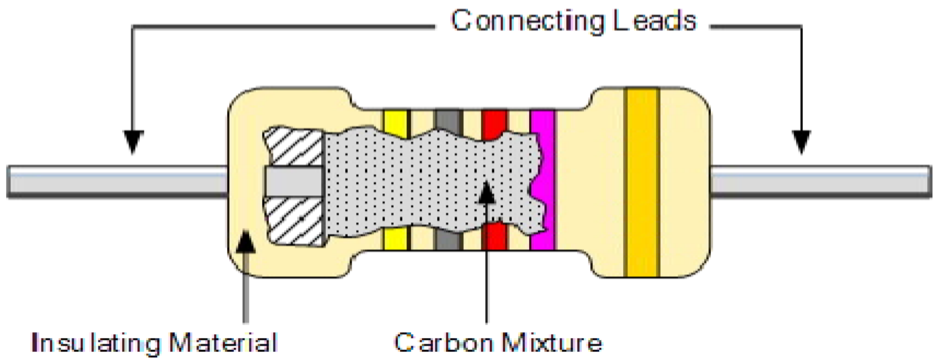



Fig.3.4. carbon resistor internal

The **Carbon Composite Resistor** is a low to medium type power resistor which has a low inductance making them ideal for high frequency applications but they can also suffer from noise and stability when hot. Carbon composite resistors are generally prefixed with a “CR” notation (e.g. CR10kΩ) and are available in E6 (± 20% tolerance (accuracy)), E12 (± 10% tolerance) and E24 (± 5% tolerance) packages with power ratings from 0.250 or 1/4 of a Watt up to 5 Watts.

Carbon composite resistor types are very cheap to make and are therefore commonly used in electrical circuits. However, due to their manufacturing process carbon type resistors have very large tolerances so for more precision and high value resistances, film type resistors are used instead.

## Film Type Resistors

The generic term “**Film Resistor**” consist of *Metal Film*, *Carbon Film* and *Metal Oxide Film* resistor types, which are generally made by depositing pure metals, such as nickel, or an oxide film, such as tin-oxide, onto an insulating ceramic rod or substrate.

Fig.3.5. Film Resistor

The resistive value of the resistor is controlled by increasing the desired thickness of the deposited film giving them the names of either “thick-film resistors” or “thin-film resistors”.

Once deposited, a laser is used to cut a high precision spiral helix groove type pattern into this film. The cutting of the film has the effect of increasing the conductive or resistive path, a bit like taking a long length of straight wire and forming it into a coil.

This method of manufacture allows for much closer tolerance resistors (1% or less) as compared to the simpler carbon composition types. The tolerance of a resistor is the difference between the preferred value (i.e. 100 ohms) and its actual manufactured value i.e. 103.6 ohms, and is expressed as a percentage, for example 5%, 10% etc, and in our example the actual tolerance is 3.6%. Film type resistors also achieve a much higher maximum ohmic value compared to other types and values in excess of 10MΩ (10 Million Ohms) are available. **Film Resistor**

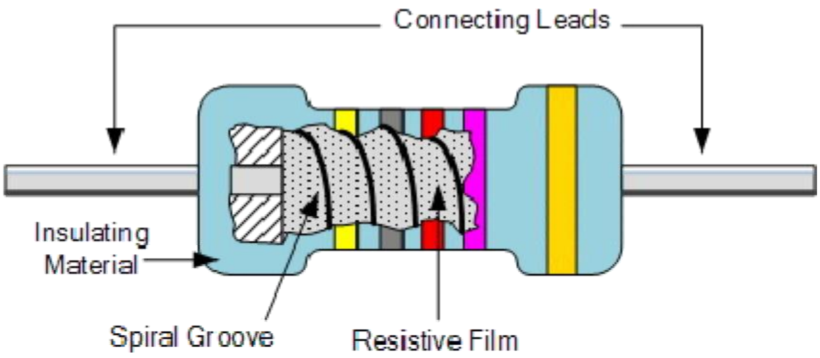



Fig.3.6. Film Resistor (internal)

**Metal Film Resistors** have much better temperature stability than their carbon equivalents, lower noise and are generally better for high frequency or radio frequency applications. **Metal Oxide Resistors** have better high surge current capability with a much higher temperature rating than the equivalent metal film resistors.

Another type of film resistor commonly known as a **Thick Film Resistor** is manufactured by depositing a much thicker conductive paste of Ceramic and Metal, called Cermet, onto an alumina ceramic substrate. Cermet resistors have similar properties to metal film resistors and are generally used for making small surface mount chip type resistors, multi-resistor networks in one package for pcb's and high frequency resistors. They have good temperature stability, low noise, and good voltage ratings but low surge current properties.
**Metal Film Resistors** are prefixed with a "MFR" notation (eg, MFR100kΩ) and a CF for Carbon Film types. Metal film resistors are available in E24 (±5% & ±2% tolerances), E96 (±1% tolerance) and E192 (±0.5%, ±0.25% & ±0.1% tolerances) packages with power ratings of 0.05 (1/20th) of a Watt up to 1/2

Watt. Generally speaking Film resistors and especially metal film resistors are precision low power components.

## **<u>Wire wound Types of Resistor</u>**

Another type of resistor, called a **Wire wound Resistor**, is made by winding a thin metal alloy wire (Nichrome) or similar wire onto an insulating ceramic former in the form of a spiral helix similar to the film resistor above.

Fig.3.7. Wire wound Resistor

These types of resistor are generally only available in very low ohmic high precision values (from 0.01Ω to 100kΩ) due to the gauge of the wire and number of turns possible on the former making them ideal for use in measuring circuits and Wheatstone bridge type applications.

They are also able to handle much higher electrical currents than other resistors of the same ohmic value with power ratings in excess of 300 Watts. These high power resistors are moulded or pressed into an aluminum heat sink body with fins attached to increase their overall surface area to promote heat loss and cooling.

These special types of resistor are called “Chassis Mounted Resistors” because they are designed to be physically mounted onto heatsinks or metal plates to further dissipate the generated heat. The mounting of the resistor onto a heatsink increases their current carrying capabilities even further.

Another type of wire wound resistor is the **Power Wire wound Resistor**. These are high temperature, high power non-inductive resistor types generally coated with a vitreous or glass epoxy enamel for use in resistance banks or DC motor/servo control and dynamic braking applications. They can even be used as low wattage space or cabinet heaters.

The non-inductive resistance wire is wound around a ceramic or porcelain tube covered with mica to prevent the alloy wires from moving when hot. Wirewound resistors are available in a variety of resistance and power ratings with one main use of power wire wound resistor is in the electrical heating elements of an electric fire which converts the electrical current flowing through it into heat with each element dissipating up to 1000 Watts, (1kW) of energy.

Because the wire of standard wire wound resistors is wound into a coil inside the resistors body, it acts like an inductor causing them to have inductance as well as resistance. This affects the way the resistor behaves in AC circuits by producing a phase shift at high frequencies especially in the larger size resistors. The length of the actual resistance path in the resistor and the leads contributes inductance in series with the "apparent" DC resistance resulting in an overall impedance path of Z Ohms.

Impedance (Z) is the combined effect of resistance ( R ) and inductance ( X ), measured in ohms and for a series AC circuit is given as, $Z^2 = R^2 + X^2$.

When used in AC circuits this inductance value changes with frequency (inductive reactance, $X_L = 2\pi fL$) and therefore, the overall value of the resistor changes. Inductive reactance increases with frequency but is zero at DC (zero frequency). Then, wire wound resistors must not be designed or used in AC or amplifier type circuits where the frequency across the resistor changes. However, special no inductive wire wound resistors are also available.

**Wire wound Resistor**

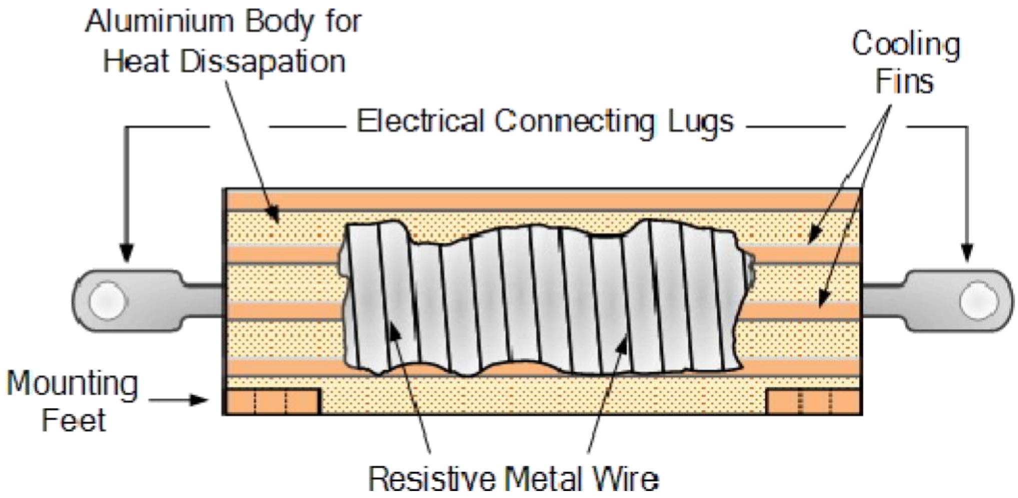



Fig.3.8. wire wound resistor

Wire wound resistor types are prefixed with a "WH" or "W" notation (eg WH10Ω) and are available in the WH aluminium clad package (±1%, ±2%, ±5% and ±10% tolerance) or the W vitreous enameled package (±1%, ±2% and ±5% tolerance) with power ratings from 1W to 300W or more.

## 3.2. LED

A **light-emitting diode** (**LED**) is a two-lead semiconductor light source. It is a p–n junction diode that emits light when activated. When a suitable current is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. LEDs are typically small (less than 1 $mm^2$) and integrated optical components may be used to shape the radiation pattern.

Fig.3.9. LED

Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity infrared light. Infrared LEDs are still frequently used as transmitting elements in remote-control circuits, such as those in remote controls for a wide variety of consumer electronics. The first visible-light LEDs were of low intensity and limited to red. Modern LEDs are available across the visible, ultraviolet, and infrared wavelengths, with very high brightness.

Early LEDs were often used as indicator lamps for electronic devices, replacing small incandescent bulbs. They were soon packaged into numeric readouts in the form of seven-segment displays and were commonly seen in digital clocks. Recent developments have produced LEDs suitable for environmental and task lighting. LEDs have led to new displays and sensors, while their high switching rates are useful in advanced communications technology.

LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. Light-emitting diodes are used in applications as diverse as aviation lighting, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper and medical devices. They are also significantly more energy efficient and, arguably, have fewer environmental concerns linked to their disposal.

Unlike a laser, the color of light emitted from an LED is neither coherent nor monochromatic, but the spectrum is narrow with respect to human vision, and for most purposes the light from a simple diode element can be regarded as functionally monochromatic.

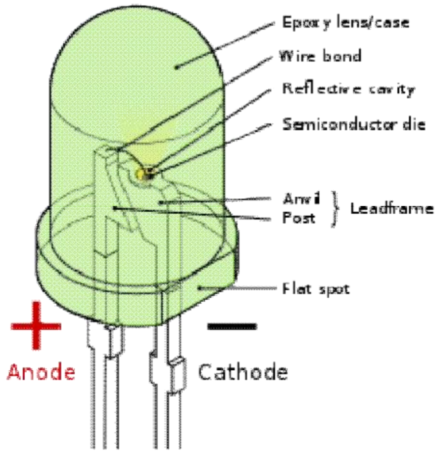



Fig.3.10. LED internal

Although LEDs will continue to be very widely used as small indicator lamps, the number of applications they can find is increasing as the technology improves. New very high luminance diodes are now available. These are even being used as a form of illumination, an application which they were previously not able to fulfil because of their low light output. New colours are being introduced. White and blue LEDs, which were previously very difficult to manufacture are now available. In view of the ongoing technology development, and their convenience of use, these devices will remain in the electronics catalogues for many years to come.

## 3.3. ULTRASONIC SENSORS

The device called ultra-sonic sensor is used to detect the distance from which the object is separated from it. The principle behind this is it sends out the sound waves and it waits for reflection of sound wave from the object under consideration. By noting down the time lag between the sent and received wave, it is actually possible to measure the distance of object from the sonar sensor.

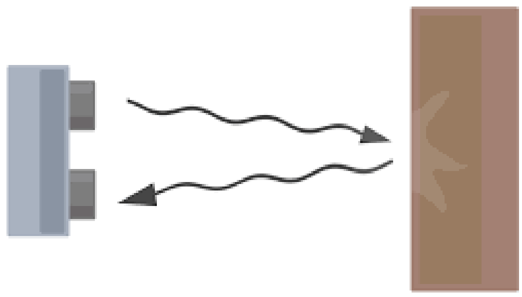


Fig 3.11. Ultrasonic Sensor functionality

Fig 3.12. Ultrasonic Sensor

As we know that sound wave travels through air at a speed of 344m/s, to find the round-trip distance of sound wave, the return time is multiplied by 344. Round trip distance refers to the twice the distance of the object from the sensor. The actual distance is calculated by diving round trip distance by 2. Distance= (speed of sound x time taken)/2 It should be noted that not all objects are detected by sensor due its shape wherein some waves get reflected back, size might be very small and the positioning angle also plays a role in detection of object. Some objects such as cloths, carpeting absorb the waves, where there is no way for detection of such objects. These are the factors to be noted.

## How Does it Work?

The ultrasonic sensor uses sonar to determine the distance to an object. Here's what happens:

1. The transmitter (trig pin) sends a signal: a high-frequency sound.
2. When the signal finds an object, it is reflected and …
3. …the transmitter (echo pin) receives it.

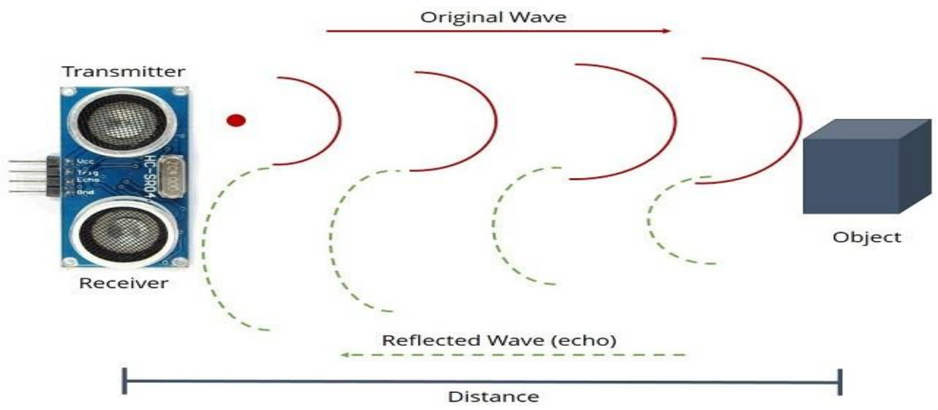



**Fig3.13. Concept of ultrasonic sensor**

## Internal circuit:

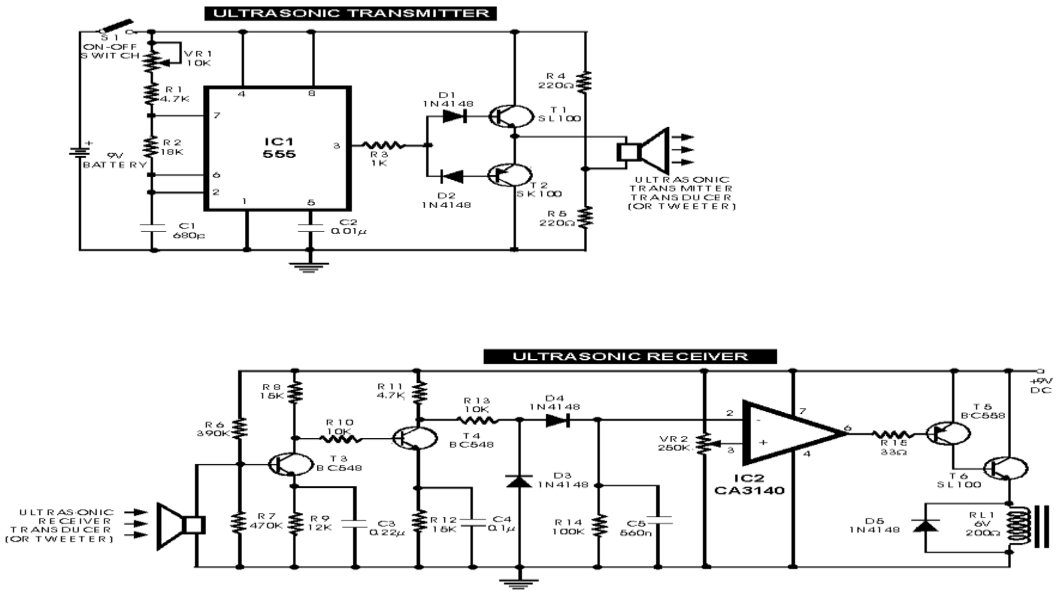



**Fig3.14. Internal circuit of ultrasonic sensor**

## Pins

- VCC: +5VDC
- Trig : Trigger (INPUT)
- Echo: Echo (OUTPUT)
- GND: GND

**Specifications:**

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1″ – 13ft
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

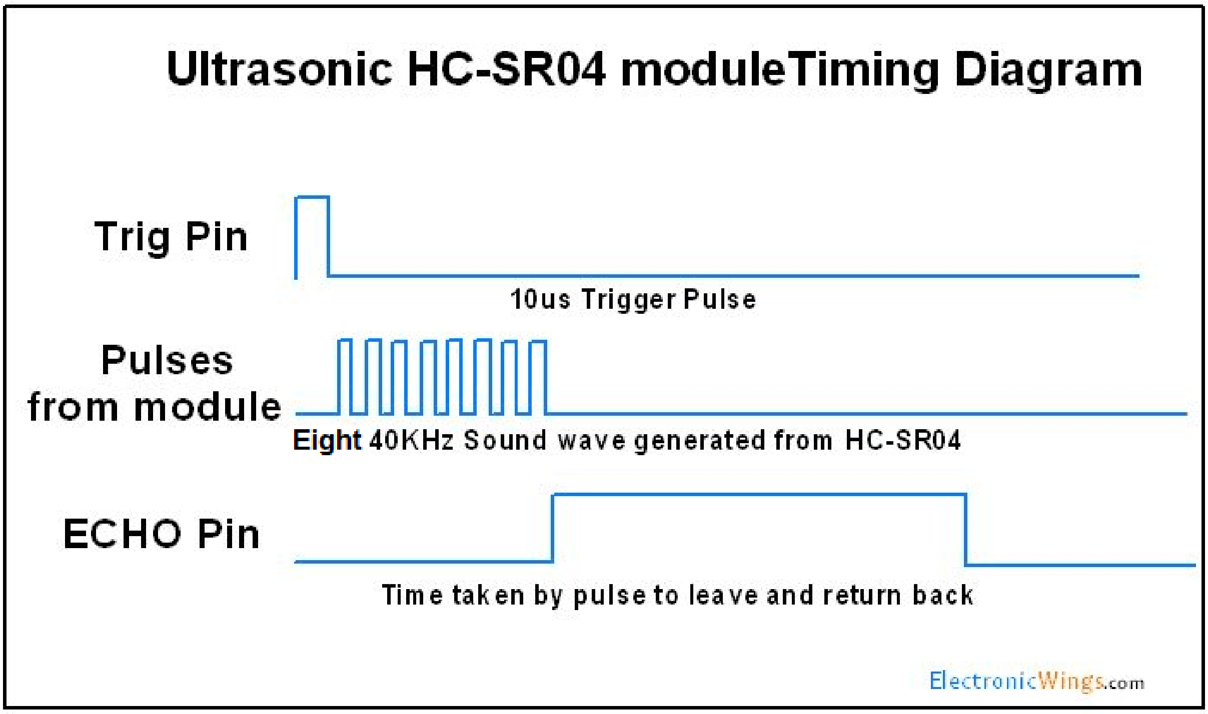



**Fig 3.15. timing diagram of ultrasonic sensor HC-SR04**

## 3.4. MICROPHONE

Fig 3.16. USB Microphone

A **microphone**, colloquially nicknamed **mic** or **mike**, is a transducer that converts sound into an electrical signal.

The USB microphone is a type of microphone that has become more and more popular over recent years. They're aimed more towards the **podcasting market** and are best suited to the **spoken voice**, such as in **radio** or **voiceover** work. But they can also be used for singing, or any sound source that is relatively loud and close to the mic.

### USB

USB stands for **Universal Serial Bus**, which is an industry standard method of **interconnecting** different electronic devices. A modern PC or Mac usually contains between one and four USB ports, sometimes even more.

One of the positive aspects of the **USB** mic is the **portability** available to you. You only need **one connection** – the USB cable runs straight from the microphone into the computer. There's no need for an external sound card, mixer, or any kind of preamp. This can make it a great **budget solution** for location recording of spoken-word interviews.

But one of the drawbacks is the fact that when a signal is low in volume, the digital USB circuitry can introduce a lot of **unwanted noise and distortion**. They can also be difficult to set up correctly inside a computer when you want to record inside your DAW program.

## USB Mic Features

Some USB mics have their **frequency response** tailored towards the spoken word, with a roll-off above 15-16 kHz and a presence peak around the 9-10 kHz mark to give the voice some "air" (a word used when higher frequencies of the voice are emphasized through the use of **equalization**).

The better USB mics usually feature a **cardioid polar pattern**, so they also display the **proximity effect** as well, where the lower bass frequencies get slightly louder the closer the sound source gets to the microphone.

The microphone itself contains a **preamp** along with an **analog-to-digital converter** (ADC), but alot of these devices don't feature separate gain controls which can be a problem if the source is on the quiet side. Latency can be very high with these mics, so to help combat this the microphone will sometimes have a **headphone jack** directly on it – this enables **zero-latency monitoring** directly from the microphone.

## Audio Interface Class

The Audio Interface class groups all functions that can interact with USB-compliant audio data streams. All functions that convert between analog and digital audio domains can be part of this class. In addition, those functions that transform USB-compliant audio data streams into other USB-compliant audio data streams can be part of this class. Even analog audio functions that are controlled through USB belong to this class. In fact, for an audio function to be part of this class, the only requirement is that it exposes one Audio Control interface. No further interaction with the function is mandatory, although most functions in the audio interface class will support one or more optional AudioStreaming interfaces for consuming or producing one or more isochronous audio data streams. The Audio Interface class code is assigned by the USB. For details, see Section A.1, “Audio Interface Class Code.”

## Inter Channel Synchronization

An important issue when dealing with audio, and 3-D audio in particular, is the phase relationship between different physical audio channels. Indeed, the virtual spatial position of an audio source is directly related to and influenced by the phase differences that are applied to the different physical audio channels used to reproduce the audio source. Therefore, it is imperative that USB audio functions respect the phase relationship among all related audio channels. However, the responsibility for maintaining the phase relation is shared among the USB host software, hardware, and all of the audio peripheral devices or functions. To provide a manageable phase model to the host, an audio function is required to report its internal delay for every Audio Streaming interface.

This delay is expressed in number of frames (ms) and is due to the fact that the audio function must buffer at least one frame worth of samples to effectively remove packet jitter within a frame. Furthermore, some audio functions will introduce extra delay because they need time to correctly interpret and process the audio data streams (for example, compression and decompression). However, it is required that an audio function introduces only an integer number of frames of delay. In the case of an audio source function, this implies that the audio function must guarantee that the first sample it fully acquires after SOFn (start of frame n) is the first sample of the packet it sends over USB during frame (n+d). d is the audio function’s internal delay expressed in ms.

The same rule applies for an audio sink function. The first sample in the packet, received over USB during frame n, must be the first sample that is fully reproduced during frame (n+d). By following these rules, phase jitter is limited to ±1 audio sample. It is up to the host software to synchronize the different audio streams by scheduling the correct packets at the correct moment, taking into account the internal delays of all audio functions involved.

**Microphone Internal Circuit:**

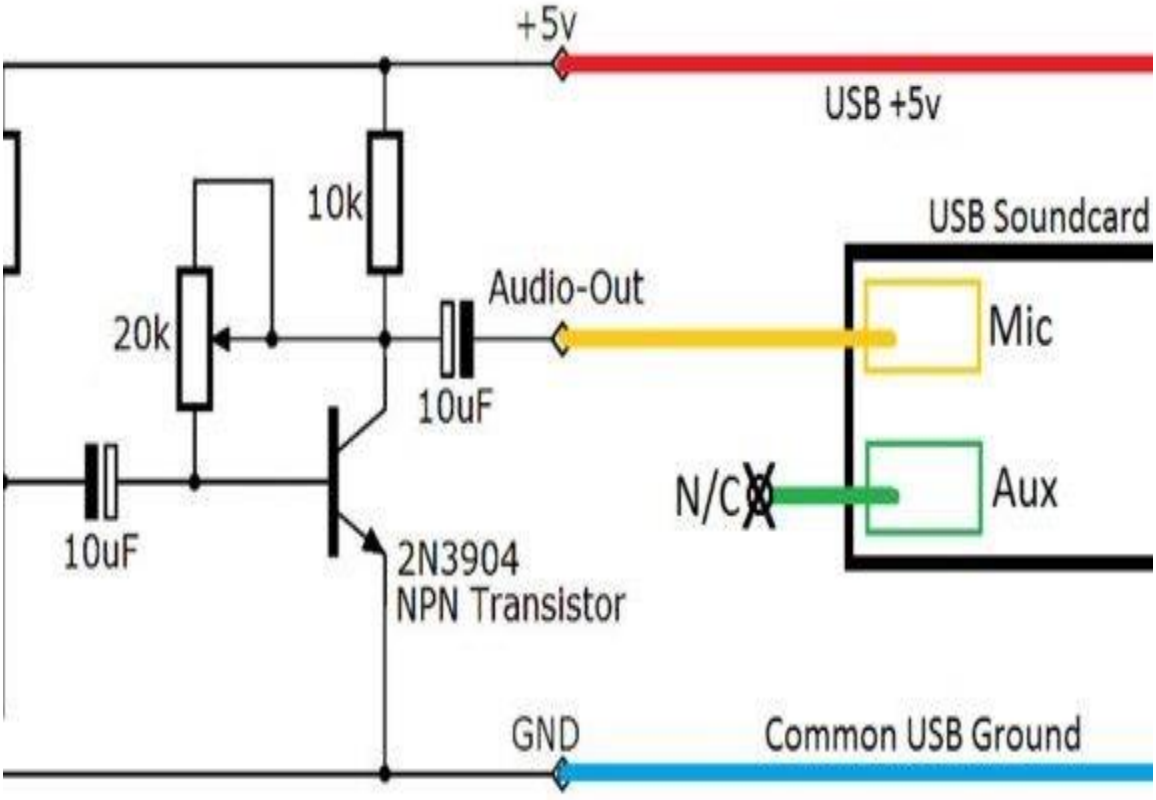



**Fig 3.17. internal circuit of USB microphone**

**Interfacing Mic with Raspberrypi:**

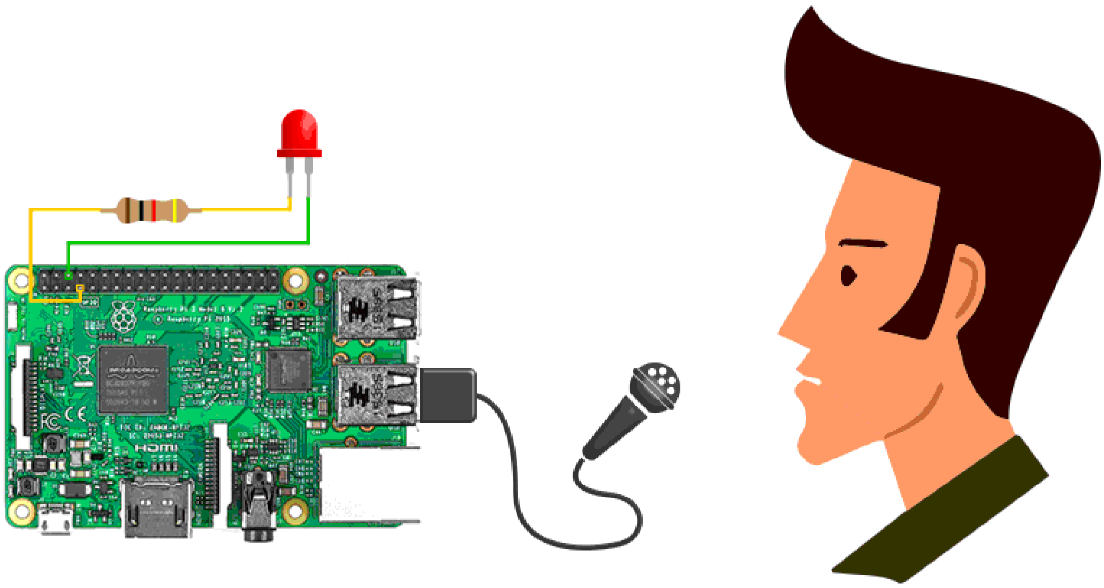


**Fig 3.18. Interfacing microphone with raspberry pi**

## Microphone Specifications

- Microphone and performance
- Power Required/Consumption: 5V 150mA
- Sample Rate: 48 kHz
- Bit Rate: 16bit
- Capsules: 3 Blue-proprietary 14mm condenser capsules
- Polar Patterns: Cardioid, Bidirectional, Omnidirectional, Stereo Frequency
- Response: 20Hz – 20kHz Sensitivity: 4.5mV/Pa (1 kHz)
- Max SPL: 120dB (THD: 0.5% 1kHz)
- Headphone Amplifier Impedance: >16 ohms
- Power Output (RMS): 130 mW THD: 0.009%
- Frequency Response: 15 Hz – 22 kHz
- Signal to Noise: 100dB

**Specifications Dimensions** (extended in stand): 4.72” (12cm) x 4.92”(12.5cm) x 11.61”(29.5cm)

Weight (microphone): 1.2 lbs (.55 kg)
Weight (stand): 2.2 lbs (1 kg)
System Requirements PC: Windows 7, Windows Vista, Linux, XP Home Edition or XP Professional USB 1.1/2.0;
64 MB RAM (minimum) Macintosh: Mac OSX ( 10.4.11 or higher ) USB 1.1/2.0 64 MB RAM (minimum)

## Applications

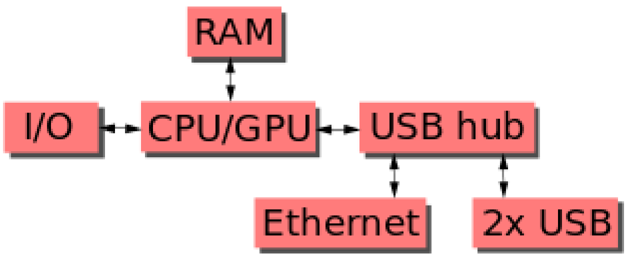
- Telephones
- Hearing aids
- Public Address Systems for concert halls and public events
- Motion Picture production, live and recorded audio engineering, sound recording, two-way radios, megaphones, radio and television broadcasting, and in computers for recording voice
- Speech Recognition
- VoIP, and for non-acoustic purposes such as ultrasonic sensors or knock sensors

## 3.5. RASPBERRYPI

The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards and mice) and cases. However, some accessories have been included in several official and unofficial bundles. The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with Eben Upton as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.


Fig 3.19. Raspberry pi

| | |
|---|---|
| **Operating system** | Linux<br>FreeBSD<br>NetBSD<br>OpenBSD<br>Plan 9<br>RISC OS<br>Windows 10 IoT Core<br>Windows 10 ARM64[2] |
| **System-on-chip used** | Broadcom BCM2837B0 |
| **CPU** | 1.4 GHz 64/32-bit quad-core ARM Cortex-A53 |
| **Memory** | 1 GB LPDDR2 RAM at 900 MHz[3] |
| **Storage** | MicroSDHC slot |
| **Graphics** | Broadcom VideoCore IV 300 MHz/400 MHz |
| **Power** | 1.5 W (average when idle) to 6.7 W (maximum under stress)[4] |




*HARDWARE ARCHITECTURE*

## Processor

The Raspberry Pi 2B uses a 32-bit 900 MHz quad-core ARM Cortex-A7 processor. The Broadcom BCM2835 SoC used in the first generation Raspberry Pi includes a 700 MHz ARM11 76JZF-S processor, VideoCore IV graphics processing unit (GPU), and RAM. It has a level 1 (L1) cache of 16 KB and a level 2 (L2) cache of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible. The 1176JZ(F)-S is the same CPU used in the original iPhone, although at a higher clock rate, and mated with a much faster GPU.

The earlier V1.1 model of the Raspberry Pi 2 used a Broadcom BCM2836 SoC with a 900 MHz 32-bit, quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache. The Raspberry Pi 2 V1.2 was upgraded to a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, the same SoC which is used on the Raspberry Pi 3, but underclocked (by default) to the same 900 MHz CPU clock speed as the V1.1. The BCM2836 SoC is no longer in production as of late 2016.

The Raspberry Pi 3+ uses a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

## Performance

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997–99. The GPU provides 1 Gpixel/s or 1.5 Gtexel/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphical capabilities of the Raspberry Pi are roughly equivalent to the performance of the Xbox of 2001.

Raspberry Pi 2 V1.1 included a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It was described as 4–6 times more powerful than its predecessor. The GPU was identical to the original. In parallelised benchmarks, the Raspberry Pi 2 V1.1 could be up to 14 times faster than a Raspberry Pi 1 Model B+.

The Raspberry Pi 3, with a quad-core ARM Cortex-A53 processor, is described as having ten times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelised tasks.

## Overclocking

Most Raspberry Pi systems-on-chip could be overclocked to 800 MHz, and some to 1000 MHz. There are reports the Raspberry Pi 2 can be similarly overclocked, in extreme cases, even to 1500 MHz (discarding all safety features and over-voltage limitations). In the Raspbian Linux distro the overclocking options on boot can be done by a software command running "sudo raspi-config" without voiding the warranty. In those cases the Pi automatically shuts the overclocking down if the chip temperature reaches 85 °C (185 °F), but it is possible to override automatic over-voltage and overclocking settings (voiding the warranty); an appropriately sized heat sink is needed to protect the chip from serious overheating.

Newer versions of the firmware contain the option to choose between five overclock ("turbo") presets that when used, attempt to maximise the performance of the SoC without impairing the lifetime of the board. This is done by monitoring the core temperature of the chip and the CPU load, and dynamically adjusting clock speeds and the core voltage. When the demand is low on the CPU or it is running too hot the performance is throttled, but if the CPU has much to do and the chip's temperature is acceptable, performance is temporarily increased with clock speeds of up to 1 GHz, depending on the board version and on which of the turbo settings is used.

The seven overclock presets are:

- none; 700 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 overvolting
- modest; 800 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 overvolting,
- medium; 900 MHz ARM, 250 MHz core, 450 MHz SDRAM, 2 overvolting,
- high; 950 MHz ARM, 250 MHz core, 450 MHz SDRAM, 6 overvolting,
- turbo; 1000 MHz ARM, 500 MHz core, 600 MHz SDRAM, 6 overvolting,
- Pi 2; 1000 MHz ARM, 500 MHz core, 500 MHz SDRAM, 2 overvolting,
- Pi 3; 1100 MHz ARM, 550 MHz core, 500 MHz SDRAM, 6 overvolting. In system information the CPU speed will appear as 1200 MHz. When idling, speed lowers to 600 MHz. In the highest (*turbo*) preset the SDRAM clock was originally 500 MHz, but this was later changed to 600 MHz because 500 MHz sometimes causes SD card corruption. Simultaneously in *high* mode the core clock speed was lowered from 450 to 250 MHz, and in *medium* mode from 333 to 250 MHz.

The Raspberry Pi Zero runs at 1 GHz.

The CPU on the first and second generation Raspberry Pi board did not require cooling, such as a heat sink or fan, even when overclocked, but the Raspberry Pi 3 may generate more heat when overclocked.

## RAM

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with only a 1080p framebuffer, and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC). Comparatively the Nokia 701 uses 128 MB for the Broadcom VideoCore IV.

For the later Model B with 512 MB RAM, new standard memory split files (arm256_start.elf, arm384_start.elf, arm496_start.elf) were initially released for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM) respectively. But a week or so later the RPF released a new version of start.elf that could read a new entry in config.txt (gpu_mem=*xx*) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start.elf worked the same for 256 MB and 512 MB Raspberry Pis.

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

## Networking

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly connected to a network using an external user-supplied USB Ethernet or Wi-Fi adapter. On the Model B and B+the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip. The Raspberry Pi 3 and Pi Zero W (wireless) are equipped with 2.4 GHz WiFi 802.11n(150 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on the Broadcom BCM43438 FullMAC chip with no official support for monitor mode but implemented through unofficial firmware patching and the Pi 3 also has a 10/100 Mbit/s Ethernet port. The Raspberry Pi 3B+ features dual-band IEEE 802.11b/g/n/ac WiFi, Bluetooth 4.2, and Gigabit Ethernet (limited to approximately 300 Mbit/s by the USB 2.0 bus between it and the SoC).

## Peripherals

The Raspberry Pi may be operated with any generic USB computer keyboard and mouse. It may also be used with USB storage, USB to MIDI converters, and *virtually* any other device/component with USB capabilities.

Other peripherals can be attached through the various pins and connectors on the surface of the Raspberry Pi.

## Real-time clock

None of the current Raspberry Pi models have a built-in real-time clock, so they are unable to keep track of the time of day independently. Instead, a program running on the Pi can retrieve the time from a network time server or from user input at boot time, thus knowing the time while powered on. To provide consistency of time for the file system, the Pi automatically saves the current system time on shutdown, and re-loads that time at boot.

A real-time hardware clock with battery backup, such as the DS1307, may be added (often via the I²C interface). Note however that this conflicts with the camera's CSI interface, effectively disabling the camera.

## Operating systems

Various operating systems for the Raspberry Pi can be installed on a MicroSD, MiniSD or SD card, depending on the board and available adapters; seen here is the MicroSD slot located on the bottom of a Raspberry Pi 2 board.

The Raspberry Pi Foundation provides Raspbian, a Debian-based Linux distribution for download, as well as third-party Ubuntu, Windows 10 IoT Core, RISC OS, and specialised media centre distributions It promotes Python and Scratch as the main programming languages, with support for many other languages. The default firmware is closed source, while an unofficial open source is available. Many other operating systems can also run on the Raspberry Pi, including the formally verified microkernel, seL4. Other third-party operating systems available via the official website include Ubuntu MATE, Windows 10 IoT Core, RISC OS and specialised distributions for the Kodi media centre and classroom management.

## Firmware

The official firmware is a freely redistributable binary blob, that is closed-source. A minimal proof-of-concept open source firmware is also available, mainly aimed at initializing and starting the ARM cores as well as performing minimal startup that is required on the ARM side. It is also capable of booting a very minimal Linux kernel, with patches to remove the dependency on the mailbox interface being responsive. It is known to work on Raspberry Pi 1, 2 and 3, as well as some variants of Raspberry Pi Zero. While it is in a working state, it is not actively developed, with last significant commits made around mid-2017.

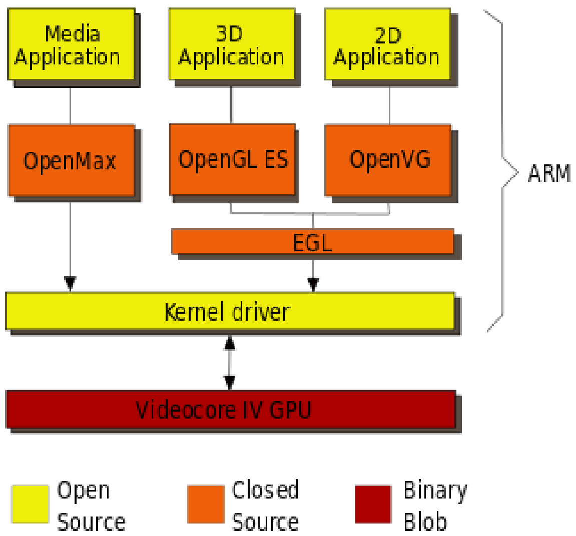
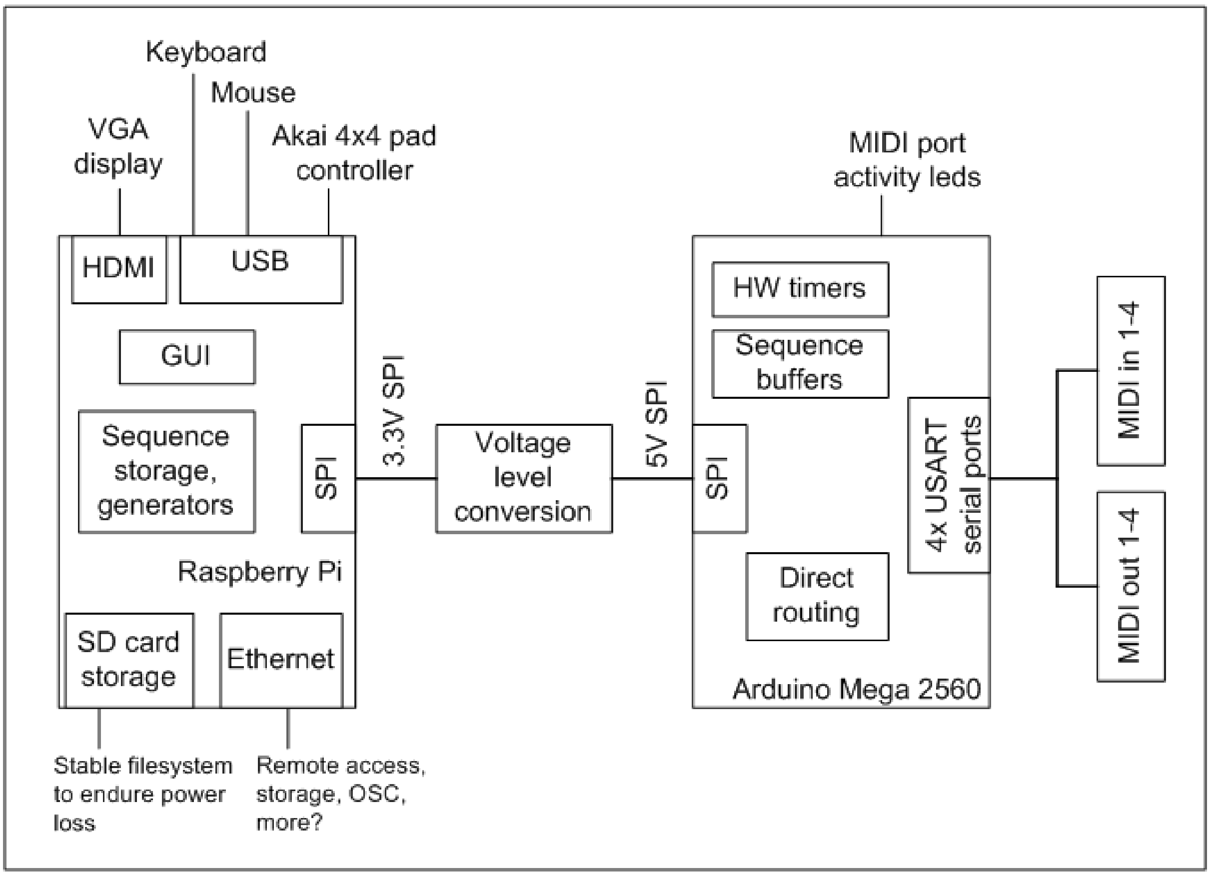



Fig 3.20. Firmware

# Specifications

| | |
|---|---|
| System on Chip: | Broadcom BCM2837 |
| CPU: | 4X ARM Cortex-A53, 64-bit, 1.2 GHz |
| GPU: | Broadcom VideoCore IV |
| RAM: | 1 GB LPDDR2 (900 MHz) |
| Networking: | 10/100 Ethernet, 2.4 GHZ 802.11n |
| Bluetooth: | 4.1 Classic, Low Energy |
| microSD: | Sandisk Ultra 8GB Class 10 microSDHC |
| GPIO: | 40-pin header, populated |
| Ports: | HDMI. 3.55mm analogue AV jack, 4xUSB 2.0, Ethernet, CSI (Camera serial interface), DSI (Digital serial Interface) |

## Internal Architecture:




## 3.6. BUZZER

A **buzzer** or **beeper** is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Fig.3.21. Buzzer

## Piezoelectric

Piezoelectric buzzers, or piezo buzzers, as they are sometimes called, were invented by Japanese manufacturers and fitted into a wide array of products during the 1970s to 1980s. This advancement mainly came about because of cooperative efforts by Japanese manufacturing companies. In 1951, they established the Barium Titanate Application Research Committee, which allowed the companies to be "competitively cooperative" and bring about several piezoelectric innovations and inventions.

## Types of buzzers:

### Electromechanical:

Early devices were based on an electromechanical system identical to an electric bell without the metal gong. Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

*Mechanical*:

A joy buzzer is an example of a purely mechanical buzzer and they require drivers. Other examples of them are doorbells.

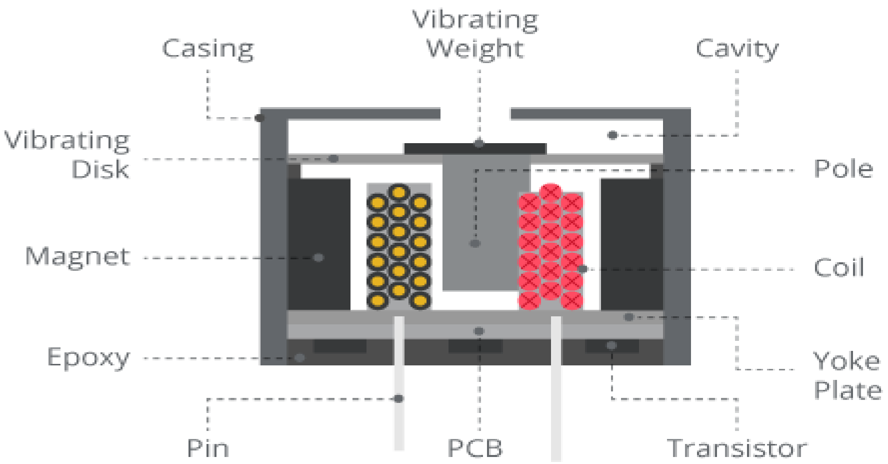
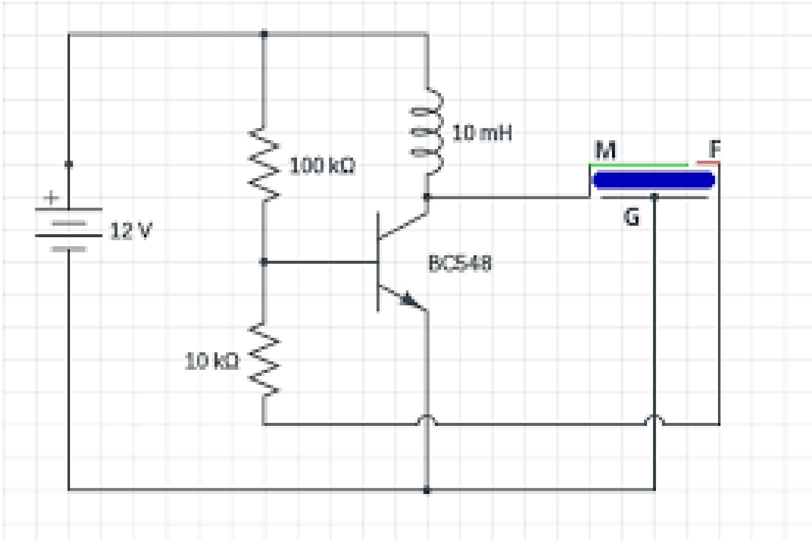



Fig 3.22. **Internal structure of a buzzer**

## Simple circuit diagram of a buzzer:




### *Uses:*

- Educational purposes
- Annunciator panels
- Electronic metronomes
- Game show lock-out device
- Microwave ovens and other household appliances
- Sporting events such as basketball games
- Electrical alarms

## 3.7. BREADBOARD

A **breadboard** is a construction base for prototyping of electronics. Originally it was literally a bread board, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

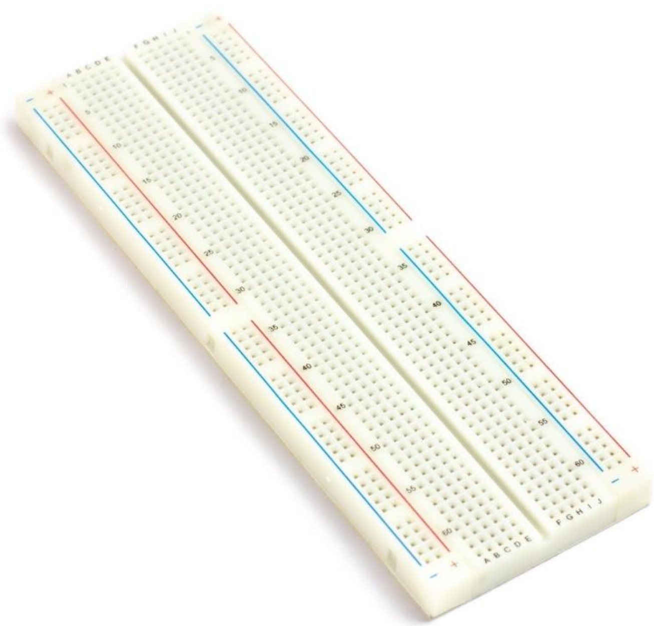


Fig.3.23. Breadboard

**Breadboard Function and Uses:**

A breadboard is an inexpensive, easy-to-use piece of hardware for wiring electrical circuits. Breadboards acquired their name because they are similar in shape to cutting boards used for cutting unsliced bread. In the past, hobbyists and engineers nailed metal spikes into cutting boards when wiring circuits.

A breadboard is usually covered with holes lined with metal, in which wires and electrical components such as resistors, diodes and capacitors can be plugged. The holes are divided into rows, and holes within particular rows are wired together on the underside of the breadboard so that an electric current can flow down the rows.

Breadboards usually are plugged in to a standard power supply that either connects to a wall outlet or a battery. Certain holes in the breadboard are connected to positive or negative voltage so that when a circuit is correctly wired and the breadboard is plugged in, current flows through the circuit. Usually, they're used with low levels of voltage and current so that components are safe to touch even while the breadboard is plugged in, but it's a good idea to keep the breadboard unplugged and, if it has a power switch, turned off until a circuit is complete to avoid shocks or damaged components.

It's easier and faster to work with breadboards to lay out circuits than it would be to try to wire components without one, so they're useful for quickly testing and prototyping electronics ideas. Breadboards can also be used without soldering, which makes it easy to remove components and reuse them when you're done with an experiment. If you're building a simple electronic project for home use, you may be able to leave it on the breadboard while it's in use, but a commercial project would need to be remade on a permanent platform.

## 3.8 Picamera

*The Raspberry Pi camera module is capable of taking full HD 1080p photo and video and can be controlled programmatically.*

### Connecting the camera

The flex cable inserts into the connector situated between the Ethernet and HDMI ports, with the silver connectors facing the HDMI port. The flex cable connector should be opened by pulling the tabs on the top of the connector upwards then towards the Ethernet port. The flex cable should be inserted firmly into the connector, with care taken not to bend the flex at too acute an angle.

The top part of the connector should then be pushed towards the HDMI connector and down, while the flex cable is held in place.

## QUICK OVERVIEW

1. Resolution: 5 MP
2. Interface Type: CSI(Camera Serial Interface)
3. Dimensions: 25x23x8 (LxWxH) mm
4. Supported Video Formats: 1080p @ 30fps, 720p @ 60fps and 640x480p 60/90 video
5. Fully Compatible with Raspberry Pi 3 Model B.
6. Plug-n-Play camera for Raspberry Pi 3 Model B.

## Specifications:

| | |
|---|---|
| **Resolution** | 5 MP |
| **Lens Focus** | Fixed Focus |
| **Image Size(Pixels)** | 2592×1944 |
| **Interface Type** | CSI(Camera Serial Interface) |
| **Sensors** | Omnivision 5647 fixed-focus |
| **Aperture** | 2.9 |
| **Focal Length** | 3.29 |
| **FOV** | 72.4° |
| **Length (mm)** | 25 |
| **Width (mm)** | 23 |
| **Height (mm)** | 8 |
| **Weight (gm)** | 3 |
| **Shipment Weight** | 0.03 kg |
| **Shipment Dimensions** | 12 x 5 x 3 cm |

## *Hardware connection:*

1. Soft cable, 90-degree vertical connector, HDMI port next to it. When connecting the contact side facing the HDMI interface.

2. Tear protective film on the lens
3. Bare boards, pay attention to ESD damage, beware of static electricity!

***Software:***

1. RPi firmware and raspi-config has been updated to the camera, do an apt-get update; apt-get upgrade;
2. Raspi-config, select the camera, start RPi firmware camera driver, and then restart
3. Using the command-line program raspivid and raspistill operate a camera to capture video clips or images
4. To capture video clips need to play with mplayer
5. How will the data through the network camera live broadcast out:
6. By nc command (ncat – Concatenate and redirect sockets) of the input data of the camera, directly to the output ports of the network redirection.

Fig 3.24. raspberry pi with Picamera

***Features:***

1. Supported Video Formats: 1080p @ 30fps, 720p @ 60fps and 640x480p 60/90 video
2. Fully Compatible with Raspberry Pi 3 Model B.
3. Small and lightweight camera module.
4. Plug-n-Play camera for Raspberry Pi 3 Model B.

**Applications**:

- CCTV security camera

- Motion detection
- Time lapse photography

# CHAPTER 4

## 4.1 EMBEDDED FIRMWARE

Embedded firmware is the flash memory chip that stores specialized software running in a chip in an embedded device to control its functions. Firmware in embedded systems fills the same purpose as a ROM but can be updated more easily for better adaptability to conditions or interconnecting with additional equipment. Hardware makers use embedded firmware to control the functions of various hardware devices and systems much like a computer's operating system controls the function of software applications. Embedded firmware exists in everything from appliances so simple you might not imagine they had computer control, like toasters, to complex tracking systems in missiles. The toaster would likely never need updating but the tracking system sometimes does. As the complexity of a device increases, it often makes sense to use firmware in case of design errors that an update might correct.

Embedded firmware is used to control the limited, set functions of hardware devices and systems of greater complexity but still gives more appliance-like usage instead of a series of terminal commands. Embedded firmware functions are activated by external controls or external actions of the hardware. Embedded firmware and ROM-based embedded software often have communication links to other devices for functionality or to address the need for the device to be adjusted, calibrated or diagnosed or to output log files. It is also through these connections that someone might attempt embedded device hacking.

Embedded software varies in complexity as much the devices it is used to control. Although *embedded software* and *embedded firmware* are sometimes used synonymously, they are not exactly the same thing. For example, embedded software may run on ROM chips. Also, embedded software is often the only computer code running on a piece of hardware while firmware can also refer to the chip that houses a computer's EFI or

BIOS, which hands over control to an OS that in turn launches and controls programs.

## 4.2 OPERATING SYSTEM

# Raspbian

Operating system

**Raspbian** is a Debian-based computer operating system for Raspberry Pi. There are several versions of Raspbian including Raspbian Stretch and Raspbian Jessie. Since 2015 it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the family of Raspberry Pi single-board computers. Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012. The operating system is still under active development. Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs.

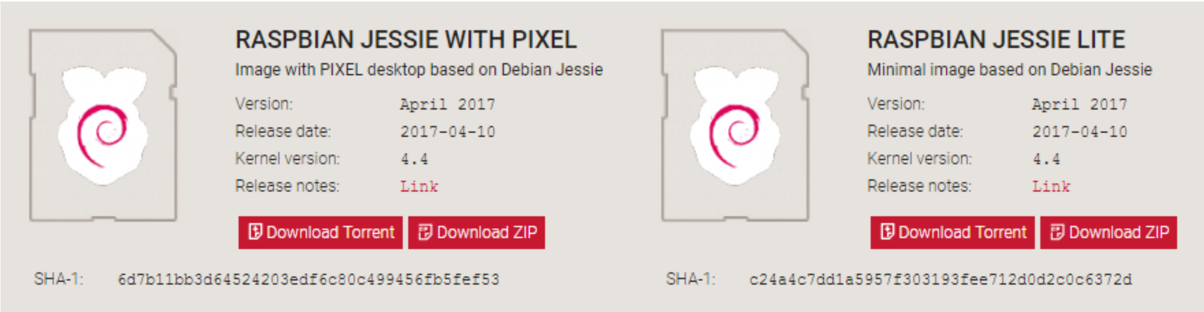
Raspbian uses PIXEL, **P**i **I**mproved **X**-Window **E**nvironment, **L**ightweight as its main desktop environment as of the latest update. It is composed of a modified LXDE desktop environment and the Openbox stacking window manager with a new theme and few other changes. The distribution is shipped with a copy of computer algebra program Mathematica and a version of Minecraft called Minecraft Pi as well as a lightweight version of Chromium as of the latest version.

***About Raspberrypi Community:***

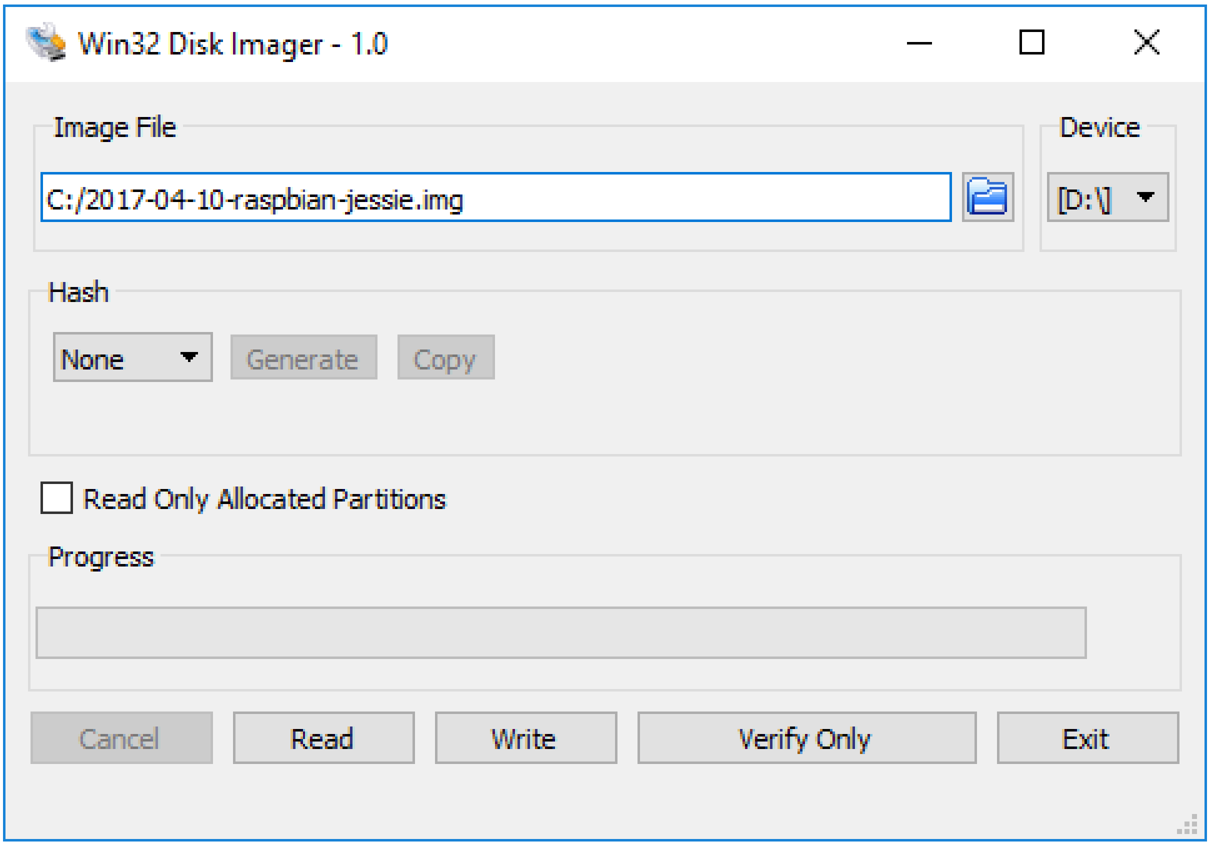
*The Raspberry Pi Foundation is a UK-based charity that works to put the power of computing and digital making into the hands of people all over the* *world. We do this so that more people are able to harness the power of computing and digital technologies for work, to solve problems that matter to them, and to express themselves creatively.*

## 4.3 Installing Raspbian OS on Pi

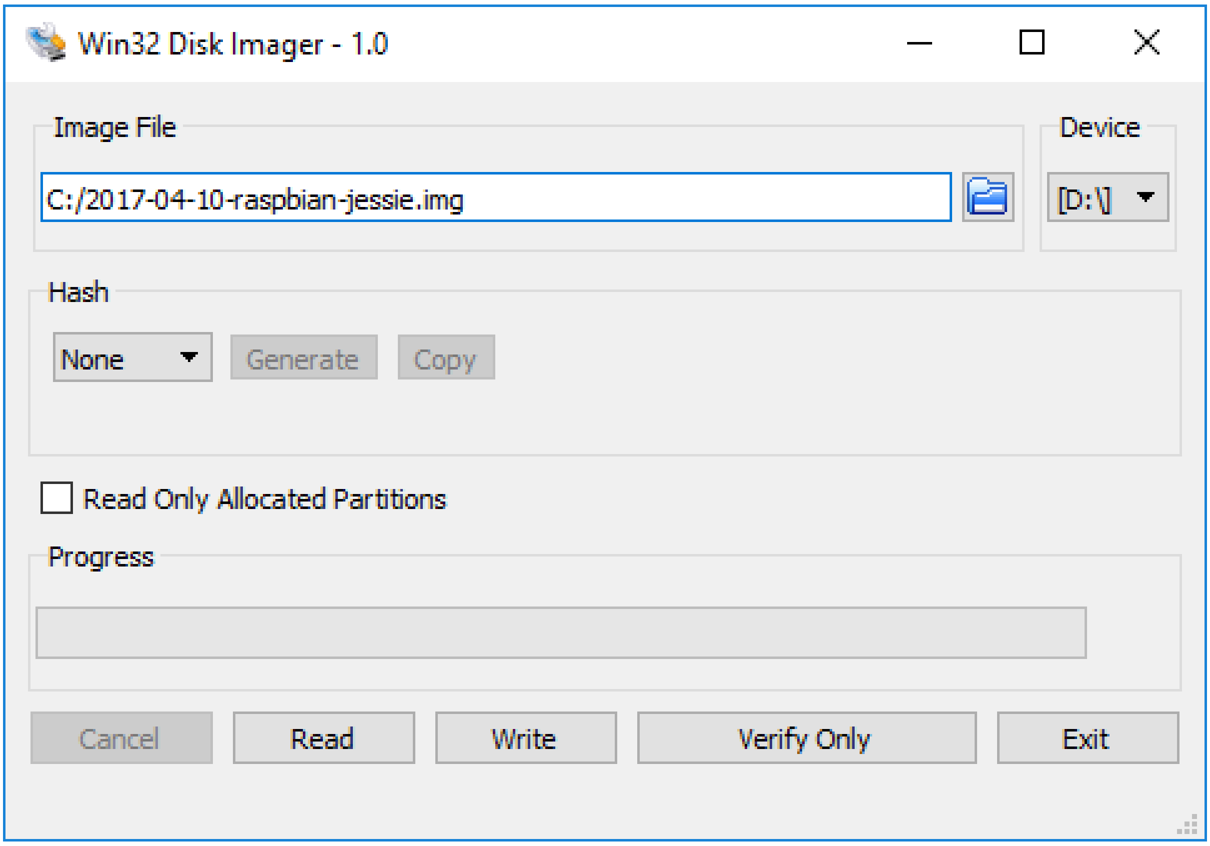
### Step 1: Download Raspbian




### Step 2: Unzip the file



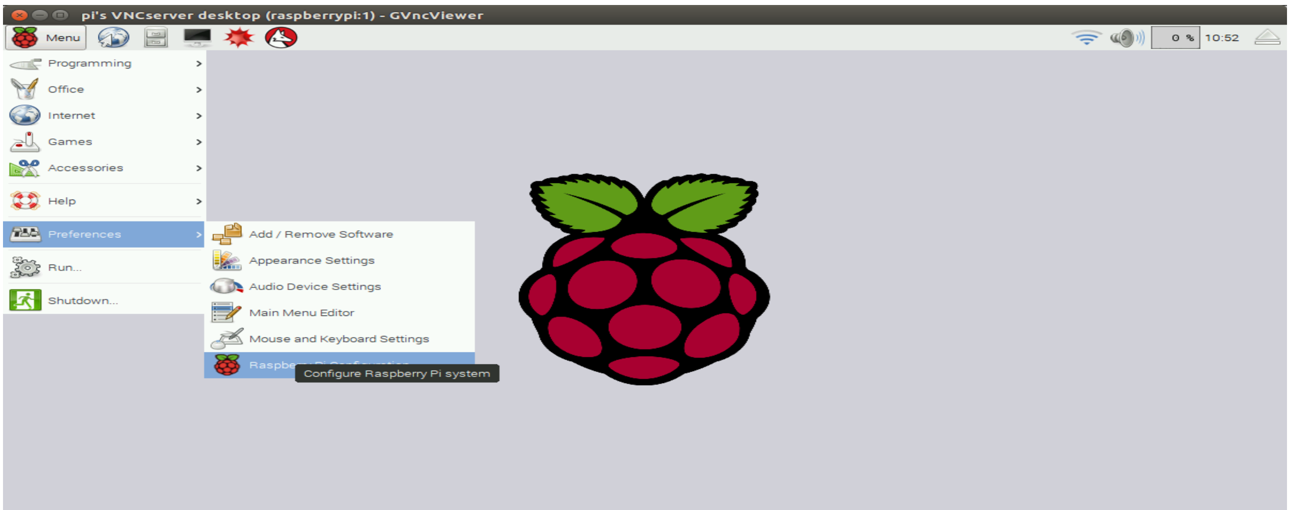
## Step 3: Write the disc image to your microSD card




## Step 4: Put the microSD card in your Pi and boot up

Once the disc image has been written to the microSD card, you’re ready to go! Put that sucker into your Raspberry Pi, plug in the peripherals and power source, and enjoy. The current edition to Raspbian will boot directly to the desktop. Your default credentials are username **pi** and password **raspberry.**

## Step 5: Verify the installation



## 4.4 PYTHON IDLE

IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit.

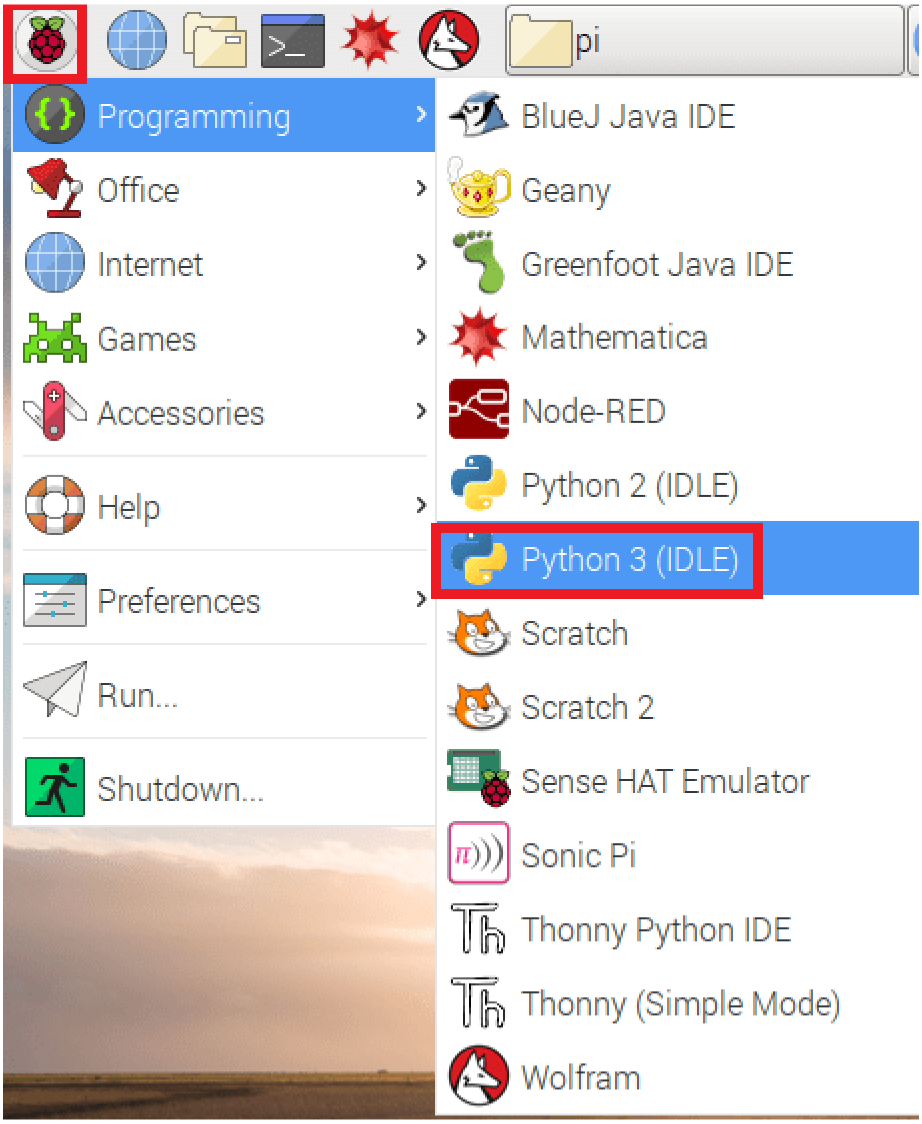
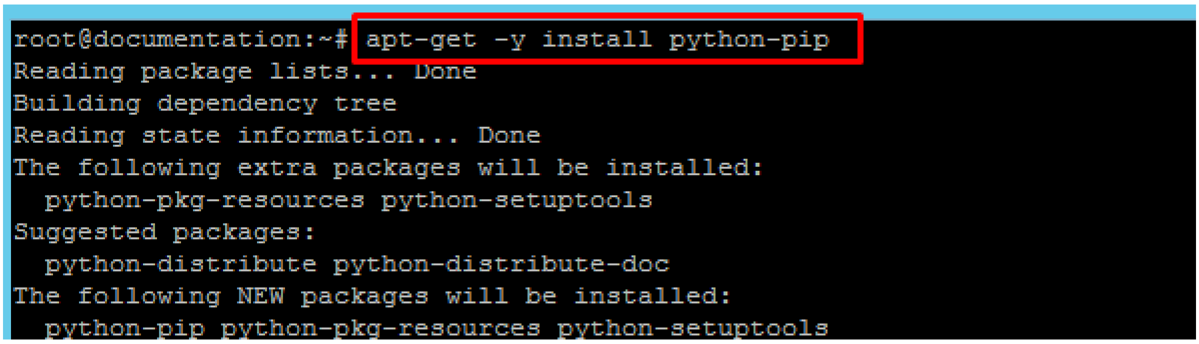


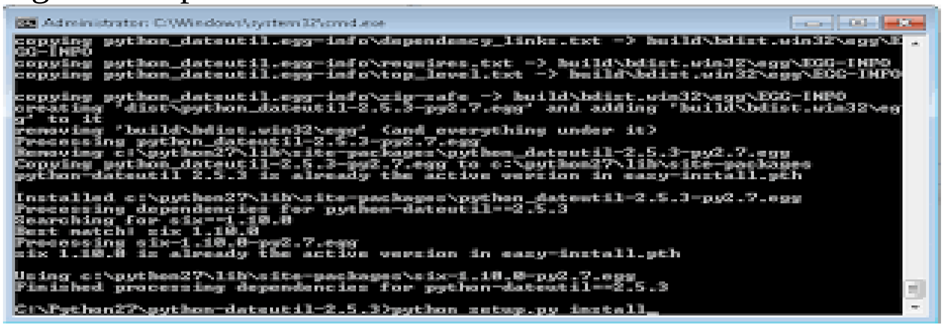

Fig 4.1. Python IDLE

## 4.5 INSTALLING DEPENDENCIES

On terminal, install pip by using :




Installing other dependencies



## Commands for installing other dependencies:

```
▪ apt install python3-pip
```

Other Dependencies:

1) pip install picam
2) pip install opencv
3) pip install opencv-contrib
4) pip install SpeechRecognition
5) pip install imutils
6) pip install scikit
7) pip install tensorflow
8) pip install matplotlib
9) pip install numpy
10) pip install scipy

# CHAPTER 5

## SOFTWARE DEVELOPMENT

## 5.1 Computer Vision algorithm:

The AI-Computer Vision part basically has 6 major components:

1) Image enhancement
2) Face Detection
3) Face Recognition
4) Object Recognition
5) Photo-OCR for reading
6) Environment perception

### A) TensorFlow

The next set of features namely- Object Recognition, Environment Understanding, Indoor Navigation, Path Planning etc use TensorFlow and Keras.

In our prototype, we used “TensorFlow object detection API” for object detection and it was built over TensorFlow framework. The dataset used here is COCO (Common Object Context) dataset. Tensors are nothing but vectors which represent pixels of the data model in any data structure. Since we are dealing with Computer Vision i.e., images here tensors represents vectors of various images, i.e., pixel descriptions. In our prototype, a pre trained model was used.

### B) Speech Recognition

The AI-Speech part includes two major components:

1) Speech-Text Conversion (STT)
2) Text-Speech Conversion (TTS)

In our prototype, we implemented the STT conversion using an STT engine named “Speech Recognizer”, using Google’s Speech Recognition API and the TTS conversion was done using an offline TTS engine named “pyttsx3”. Our prototype and the intended solution take the entire input from the blind user’s voice. We’ve created a data set that has some key-words which would map the user’s command. For example: To recognize the face of the person in front of him, he can simply say “Who” or “Recognize” and the STT engine with the help of Google’s API would understand the command, search for the relevant keyword in the pre-defined lists and call the corresponding face_recognizer.py file. After getting the desired output, the algorithm would send the string to the TTS engine and the TTS engine would give the output of the name of the person, something like “*PRANAV is in front of you*”

## 5.2 IMPLEMENTATION

We've developed a prototype of our project with certain key features like:

1) Face Detection
2) Face Recognition
3) Object Recognition
4) Speech Recognition
5) Speech Output

## Face Recognition:

Face Detection and Face Recognition is implemented using OpenCV-the most powerful library for Computer Vision. These features are implemented using HAAR Cascade. In our prototype we've achieved an accuracy of over 70%, without using any neural networks. The reason for choosing OpenCV over other available options like Neural Nets or TensorFlow API's for face recognition is the low computational power of HAAR over the other available options.

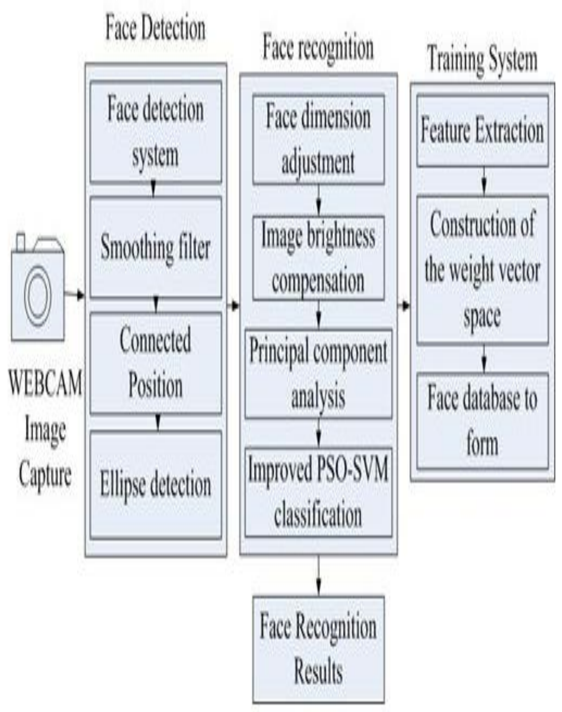



Fig 5.1. Implementation of face detection and recognition

# Object recognition:

The working of the system starts by suitably powering the raspberry pi processor. Thus, the web camera interfaced through one of the USB ports of pi is initialized. Real time video is captured using the NoIR camera which in turn is converted to a set of frames using python command. Here, we are using a simplest and fastest pre trained object detection model ‘ssd_mobilenet_v1_coco’ offered by TensorFlow to detect various objects present within our image. Identification of various objects present in the image is done using detection graphs and weights. The output contains a box representing a part of the image where a particular object was detected, score representing level of confidence for each of the objects and class label. This can be displayed when the raspberry pi is interfaced with a display system. By using the text to speech converter software, pyttsx3, the text documents like class label, scores etc are converted to voice output. The earphones connected to audio jack of raspberry pi provides voice description corresponding to the objects present in the image.

TensorFlow Object Detection API The TensorFlow API is widely used in the field of object detection. It is an open source software library for numerical computation using data flow graphs [8]. Second generation of Google artificial intelligence learning system got much attention and affirmation in the field of machine learning all over the world [2]. The API is trained by using the COCO dataset (Common Objects in Context). This is a dataset which contains 300k images of 90 most commonly found objects. Examples of objects includes:

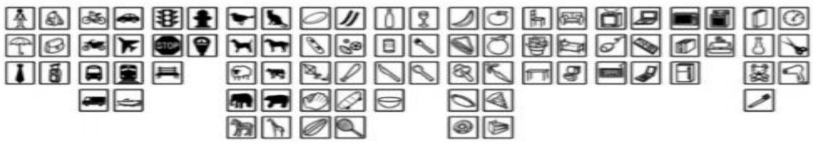


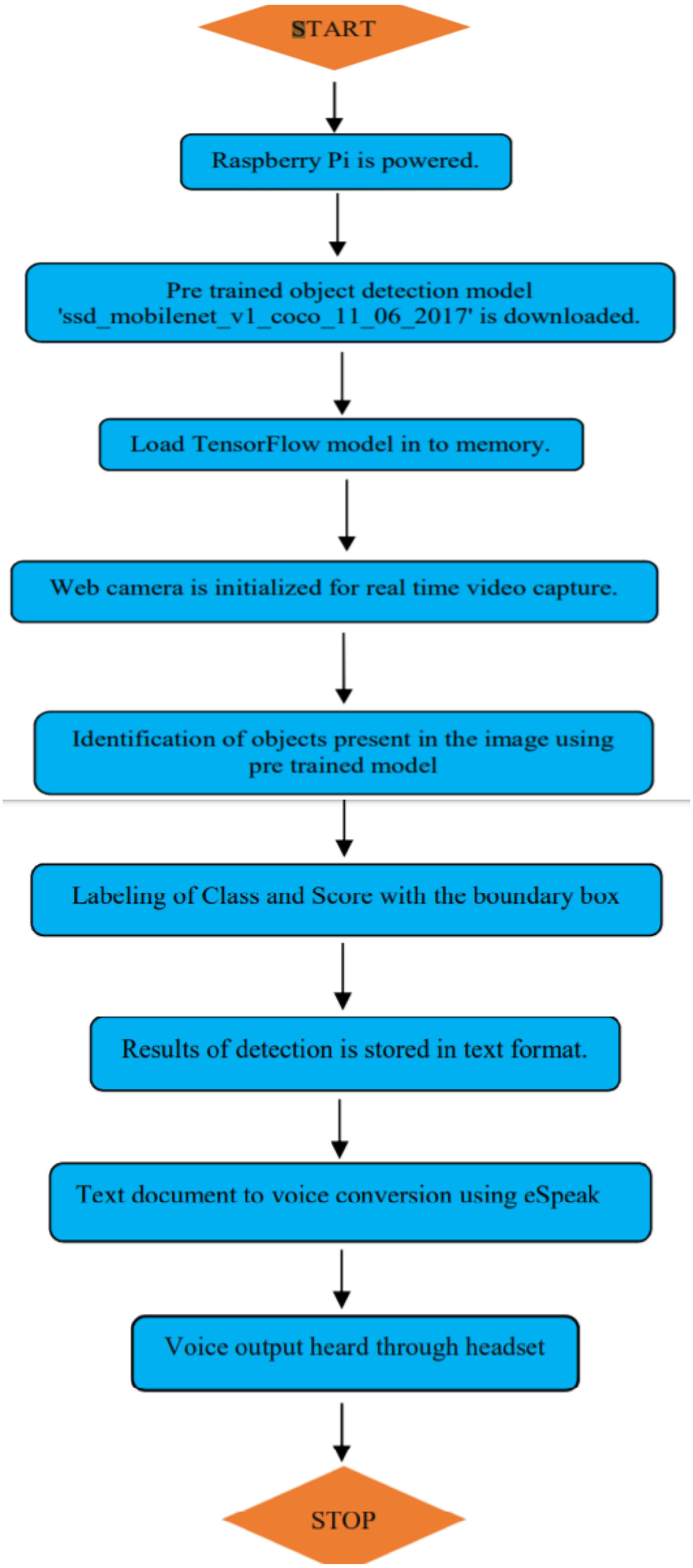
Fig.3. Objects in COCO model

The COCO model is achieved by gathering images of complex everyday scenes containing common objects in their natural context. The API provides 5 different models for object detection [4].

| Model name | Speed | COCO mAP | Outputs |
|---|---|---|---|
| ssd_mobilenet_v1_coco | fast | 21 | Boxes |
| ssd_inception_v2_coco | fast | 24 | Boxes |
| rfcn_resnet101_coco | medium | 30 | Boxes |
| faster_rcnn_resnet101_coco | medium | 32 | Boxes |
| faster_rcnn_inception_resnet_v2_atrous_coco | slow | 37 | Boxes |

**Fig 5.2. Objects of COCO model**

## Flow chart of object recognition



## Image Enhancement:

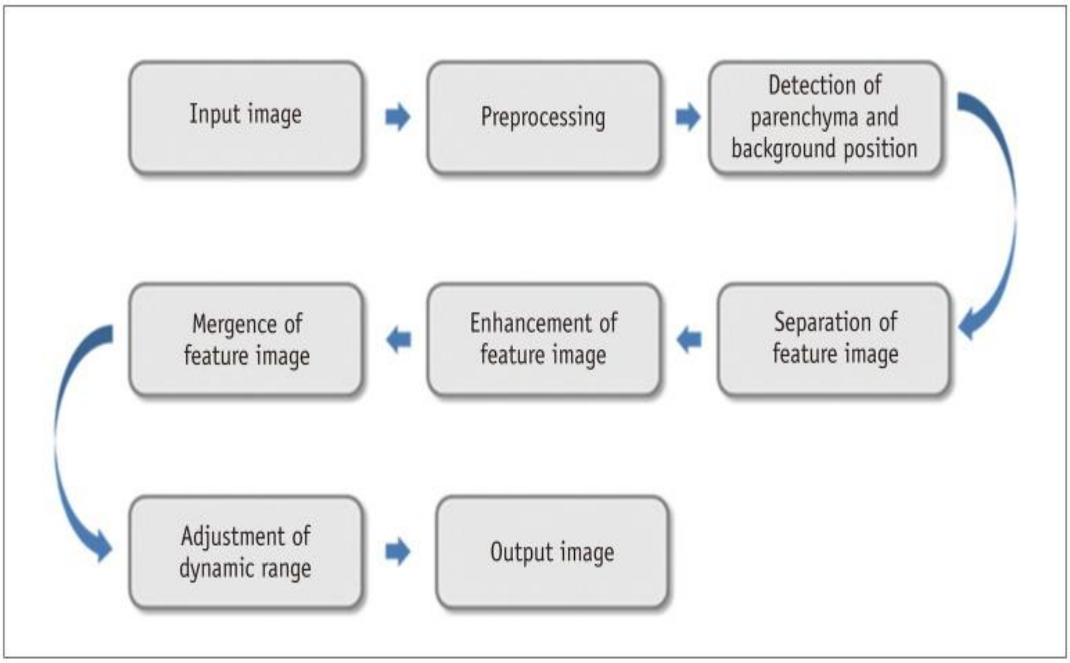



**Fig 5.3. image enhancement**

In short, following steps are involved in image processing:

1) We take input as an image.
2) Analyse and manipulate the image
3) Output is a processed image

$ sudo pip install pillow

Also, PIL can be used for the image enhancement and the development of the Python based image processing application so that it becomes easy for the beginners to learn and understand the complex tasks of the image processing using Python based image processing.

PIL provides some modules for performing operations on images. Image module is one of them which provides some basic functions like open, save, show, etc. The most important thing is the functions present in the Image module has the same names as the operations to be performed on that image. For using this module, first initialize the following things: ≫ import PIL ≫ from PIL import Image

# Speech Recognition

Automatic Speech Recognition (ASR) requires three main components for further analysis: Preprocessing, feature extraction, and post-processing. Feature extraction, in an abstract meaning, is extracting descriptive features from raw signal for speech classification purposes (Fig. 1). Due to the high dimensionality, the raw signal can be less informative compared to extracted higher level features. Feature extraction comes to our rescue for turning the high dimensional signal to a lower dimensional and yet more informative version of that for sound recognition and classification [1, 2, 3]

Feature extraction, in essence, should be done considering the specific application at hand. For example, in ASR applications, the linguistic characteristics of the raw signal are of great importance and the other characteristics must be ignored [4, 5]. On the other hand, in Speaker Recognition (SR) task, solely voice-associated information must be contained in extracted feature. So the feature extraction goal is to extract the relevant feature from the raw signal

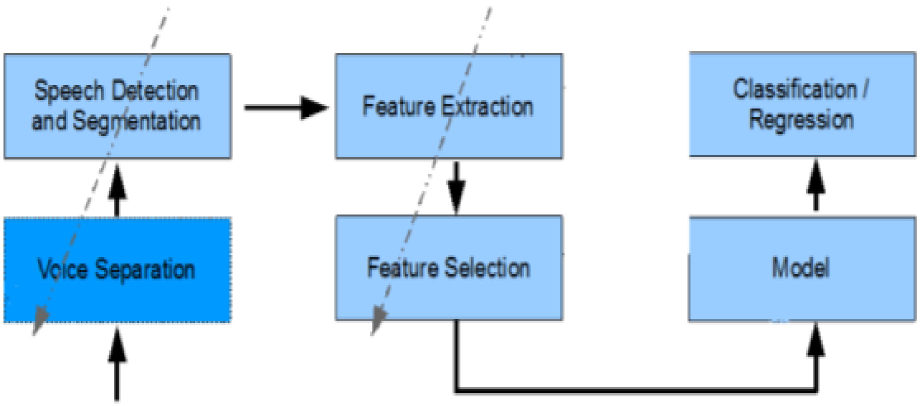
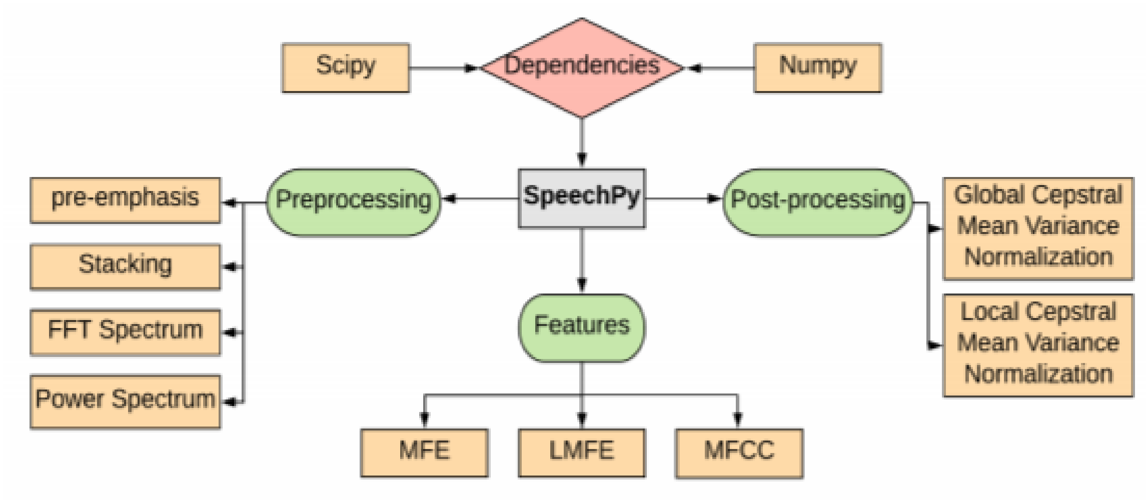



Fig 5.4. Speech recognition

and map it to a lower dimensional feature space. The problem of feature extraction has been investigated in pattern classification aimed at preventing the curse of dimensionality. There are some feature extraction approaches based on information theory [7, 8] applied to multimodal signals and demonstrated promising results.

- Google STT is the speech-to-text system by Google. If you have an Android smartphone, you might already be familiar with it, because it's basically the same engine that performs recognition if you say *OK, Google*. It can only transcribe a limited amount of speech a day and needs an active internet connection.
- An application invokes the **pyttsx3.init()** factory function to get a reference to a **pyttsx3.Engine** instance. During construction, the engine initializes a **pyttsx3.driver.DriverProxy** object responsible for loading a speech engine driver implementation from the **pyttsx3.drivers** module. After construction, an application uses the engine object to register and unregister event callbacks; produce and stop speech; get and set speech engine properties; and start and stop event loops.

## Flowchart of Speech Recognition:



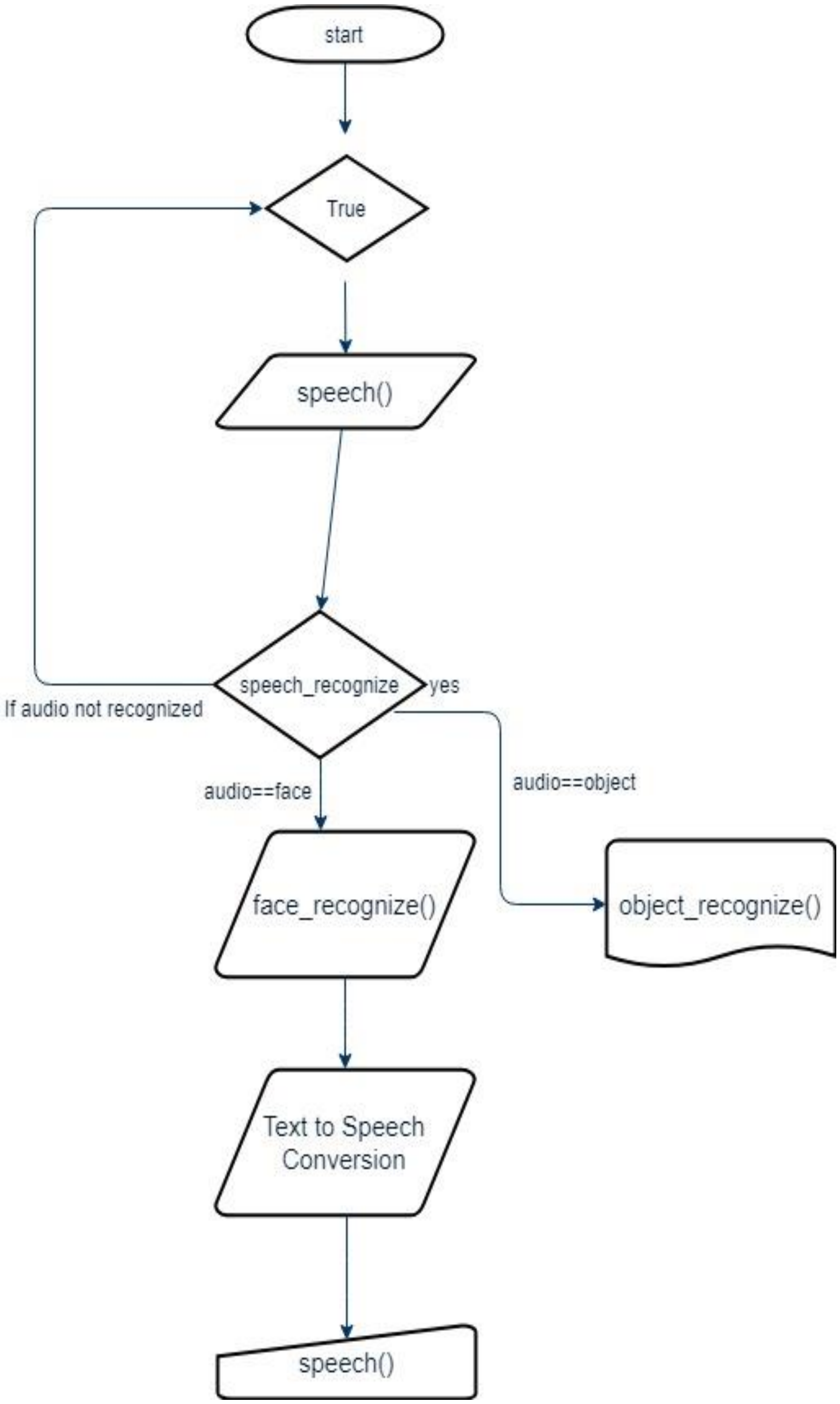

The Engine factory

pyttsx3.**init**([*driverName : string*, *debug : bool*]) → pyttsx3.Engine

| | |
|---|---|
| **Parameters:** | **driverName** –<br><br>Name of the **pyttsx3.drivers** module to load and use. Defaults to the best available driver for the platform, currently:<br><br>*sapi5* - SAPI5 on Windows<br>*nsss* - NSSpeechSynthesizer on Mac OS X<br>*espeak* - eSpeak on every other platform |

| | |
|---|---|
| | **debug** – Enable debug output or not. |
| **Raises:** | **ImportError** – When the requested driver is not found<br>**RuntimeError** – When the driver fails to initialize |

## 5.3 Flow-Chart



# CHAPTER- 6

## 6.1. RESULT

Finally, the solution provided by this system can be effective in managing large numbers of trash bins over a small-scale location.

Fig 6.1. Snap of the project

The wearable device we developed can be used to recognize faces, objects around it. The total input and output is completely based on audio signals in the form of speech.

The system has a simple architecture that transforms the visual information captured using a camera to voice information using Raspberry Pi. Unlike other systems available in the market, the subject needs only to wear the cap and doesn't require any particular skills to operate it. The proposed system is cheap and configurable. The person who uses it does not require any particular skill to operate it.

Any blind or visually impaired person can use it simply since he/she has to only power up the device. The system helps in clear path indication and environment recognition. The device is a real-time system that monitors the environment and provides audio information about the environment making his/her navigation more safe and secure.

## 6.2. ADVANTAGES

- Cost reduction
- Simple Hardware
- Cloud computation
- Less complex, but more effective
- Wearable Techonology
- Can be remotely monitored by escorts or relatives/friends

## 6.3. CHALLENGES

- Heavy weight of the physical hardware
- Required high speed internet connectivity
- The processing is slower at times due to instability of models

## 6.4. APPLICATIONS

- Can be used by blind or visually impaired people in real time

# CHAPTER 7

## 7.1. CONCLUSION

This solution presents a novel technique for assisting visually impaired people. The proposed system has a simple architecture and makes it user friendly thus, making the subject independent in his/her home. The system also aims at helping blind to navigate in his/her surroundings by detecting obstacles, locate his basic necessities, read sign boards and texts. Preliminary experiments show promising results as the user can freely navigate in his surroundings safely. The system is made much more user friendly by accepting speech as the input to access his basic necessities.

The desired output was achieved with the following accuracies:

1) Face Detection – 97%
2) Face Recognition – 67%
3) Object Detection – 62%
4) Speech Recognition – 59%
5) Speech Output – 98%

Future work includes more improvement in processed image using different algorithm or combination of algorithm techniques which best suits the system. And object detection can also be implemented, for the blind people to detect the exact obstacle. Initially simple object detection like stair case detection, vehicle detection etc. can be done along with depth information. Also, the complete set-up of the navigation system can be made using pair of glasses with camera fitted on the two sides and an ear-piece connected to the device, to alert the blind person about any obstacle present in the environment.

## 7.2. **FUTURE SCOPE**

- Using Computer Vision technology, various other modes and algorithms can be developed for blind person's indoor and outdoor navigation.
- Improvement of Object Detection algorithm.

# REFERENCES


- https://www.ijariit.com/manuscripts/v4i2/V4I2-1339.pdf
- https://www.researchgate.net/profile/Rammohan_Mallipeddi/publication/286331302_Smart_Cane_Face_Recognition_System_for_Blind/links/5667e32b08ae8905db8bced2.pdf
- http://sci-hub.tw/https://ieeexplore.ieee.org/document/8597327
- http://sci-hub.tw/https://ieeexplore.ieee.org/document/7434990
- http://sci-hub.tw/https://ieeexplore.ieee.org/document/8524251