CHAPTER 1
INTRODUCTION

1.1 Introduction:

Security is of utmost concern, whether it may be the security of data, assets, and human life. With the development of technology, the increasing use of IoT and biometrics authentication for security has gained popularity. Facial recognition is considered a universal, collectible, and accessible biometrics system. Biometric face recognition is based on the same identifier that humans use to distinguish one person from another. In this work, facial recognition has been considered to develop a highly reliable door lock system. Face-ID-based security system has been developed using ESP32-CAM. The system performs real-time face detection and recognition, rapidly when a face is in front of a camera. A prototype has been developed and tested, that evaluates the door lock system's working, efficiency, and reliability. The results show that it can be efficiently used to enhance human life and security and for commercial and residential purposes.

There are various methods to ensure human security, be it at home, in the office or anywhere else. Some methods include wired or wireless alarms, RFID keys, and biometrics recognition. However, the first two methods also have several disadvantages, such as losing keys and forgetting passwords. Password protection at the entrance using RFID between the door and the control panel is cheaper, but it can be readily defeated utilizing advanced technologies like intercepting data, deciphering commands, and replaying them to the control panel at will. This signal can also be jammed to prevent it from triggering an alert by sending radio noise to stop the signal from reaching the control panel from the sensors. Significant advancements in the field of digital sciences have boosted the use of control engineering systems in daily life in recent years. As a result of this advancement, systems capable of in-depth data processing, including photos and videos, are now available. Facial recognition is the most popular method in biometric recognition technologies besides fingerprint characteristics which uses this technology. The goal is to locate a face in the database with the greatest resemblance to the person's face. The emergence of commercial interest, and hence the creation of feasible technologies to enable the development of face recognition, is ascribed to the requirement for face recognition in security systems. Unlike many of the technologies listed above, face recognition does not necessitate physical touch. The system can detect a human face in the form of an image or video, then compare and analyse the face's pre-determined attributes

recorded in the database because of its ease of use, this technology has become quite popular for a variety of applications, including device locking systems and national security systems.

1.2 IOT (Internet of things)

The Internet of things (IoT) describes devices with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. The Internet of things encompasses electronics, communication, and computer science engineering. "Internet of things" has been considered a misnomer because devices do not need to be connected to the public internet; they only need to be connected to a network and be individually addressable. The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, and increasingly powerful embedded systems, as well as machine learning. Older fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with "smart home" products, including devices and appliances (lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcare systems.

There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently there have been industry and government moves to address these concerns, including the development of international and local standards, guidelines, and regulatory frameworks.

The extensive set of applications for IoT devices is often divided into consumer, commercial, industrial, and infrastructure spaces.

1.2.1 Home Automation

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems and camera systems. Long-term benefits could include energy savings by automatically ensuring lights and electronics are turned off or by making the residents in the home aware of usage.

A smart home or automated home could be based on a platform or hubs that control smart devices and appliances. For instance, using Apple's HomeKit, manufacturers can have their

home products and accessories controlled by an application in iOS devices such as the iPhone and the Apple Watch. This could be a dedicated app or iOS native applications such as Siri. This can be demonstrated in the case of Lenovo's Smart Home Essentials, which is a line of smart home devices that are controlled through Apple's Home app or Siri without the need for a Wi-Fi bridge. There are also dedicated smart home hubs that are offered as standalone platforms to connect different smart home products. These include the Amazon Echo, Google Home, Apple's Home Pod, and Samsung's SmartThings Hub. In addition to the commercial systems, there are many non-proprietary, open source ecosystems, including Home Assistant, OpenHAB and Domoticz. The use of IOT in home automation is shown in fig1.2.1.

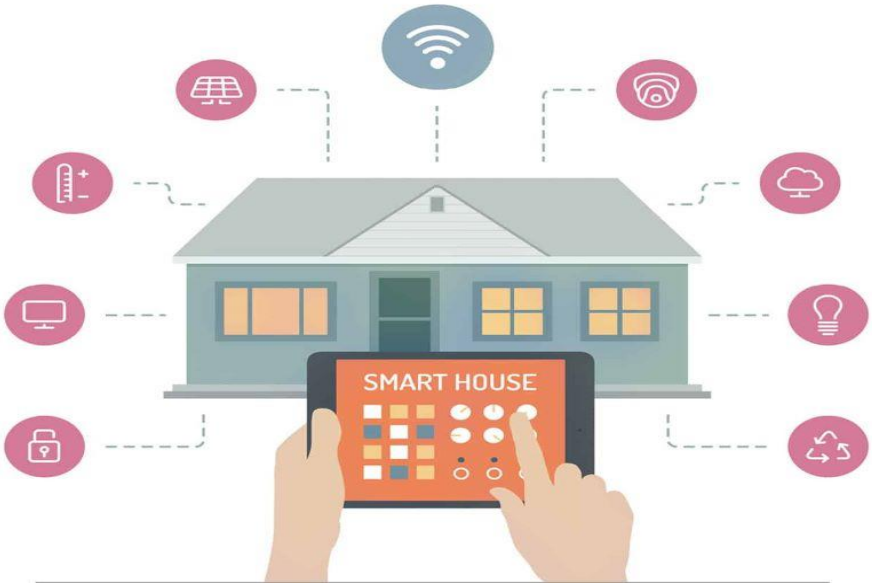


Fig 1.2.1 IOT use in home automation

1.2.2. Medical and Healthcare

The Internet of Medical Things (IoMT) is an application of the IoT for medical and health-related purposes, data collection and analysis for research, and monitoring. The IoMT has been referenced as "Smart Healthcare" as the technology for creating a digitized healthcare system, connecting available medical resources and healthcare services.

IoT devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fitbit electronic wristbands, or advanced hearing aids. Some hospitals have begun implementing "smart beds" that can detect when they are occupied and when a patient is

attempting to get up. It can also adjust itself to ensure appropriate pressure and support are applied to the patient without the manual interaction of nurses. A 2015 Goldman Sachs report indicated that healthcare IoT devices "can save the United States more than $300 billion in annual healthcare expenditures by increasing revenue and decreasing cost." Moreover, the use of mobile devices to support medical follow-up led to the creation of 'm-health', used analysed health statistics."

Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people to regain lost mobility via therapy as well. These sensors create a network of intelligent sensors that are able to collect, process, transfer, and analyse valuable information in different environments, such as connecting in-home monitoring devices to hospital-based systems. Other consumer devices to encourage healthy living, such as connected scales or wearable heart monitors, are also a possibility with the IoT. End-to-end health monitoring IoT platforms are also available for antenatal and chronic patients, helping one manage health vitals and recurring medication requirements.

Advances in plastic and fabric electronics fabrication methods have enabled ultra-low cost, use-and-throw IoMT sensors. These sensors, along with the required RFID electronics, can be fabricated on paper or e-textiles for wireless powered disposable sensing devices. Applications have been established for point-of-care medical diagnostics, where portability and low system-complexity is essential. The use of IOT in medical and health care is shown in fig 1.2.2.

Fig 1.2.2 IOT use in medical and Health care

1.2.3. Manufacturing

The IoT can connect various manufacturing devices equipped with sensing, identification, processing, communication, actuation, and networking capabilities. Network control and management of manufacturing equipment, asset and situation management, or manufacturing process control allow IoT to be used for industrial applications and smart manufacturing. IoT intelligent systems enable rapid manufacturing and optimization of new products and rapid response to product demands.

Digital control systems to automate process controls, operator tools and service information systems to optimize plant safety and security are within the purview of the IoT. IoT can also be applied to asset management via predictive maintenance, statistical evaluation, and measurements to maximize reliability. Industrial management systems can be integrated with smart grids, enabling energy optimization. Measurements, automated controls, plant optimization, health and safety management, and other functions are provided by networked sensors.

In addition to general manufacturing, IoT is also used for processes in the industrialization of construction. The use of IOT in manufacturing is shown in fig 1.2.3.

Fig 1.2.3 The use of IOT in manufacturing

1.2.4. Agriculture

There are numerous IoT applications in farming such as collecting data on temperature, rainfall, humidity, wind speed, pest infestation, and soil content. This data can be used to automate farming techniques, take informed decisions to improve quality and quantity, minimize risk and waste, and reduce the effort required to manage crops. For example, farmers can now monitor soil temperature and moisture from afar and even apply IoT-acquired data to precision fertilization programs. The overall goal is that data from sensors, coupled with the farmer's knowledge and intuition about his or her farm, can help increase farm productivity, and help reduce costs.

In August 2018, Toyota Tsusho began a partnership with Microsoft to create fish farming tools using the Microsoft Azure application suite for IoT technologies related to water management. Developed in part by researchers from Kindai University, the water pump mechanisms use artificial intelligence to count the number of fish on a conveyor belt, analyze the number of fish, and deduce the effectiveness of water flow from the data the fish provide. The Farm Beats project from Microsoft Research that uses TV white space to connect farms is also a part of the Azure Marketplace now. The use of IOT in agriculture is shown in fig 1.2.4.

Fig 1.2.4 Use of IOT in agriculture

1.2.5. Military

The Internet of Military Things (IoMT) is the application of IoT technologies in the military domain for the purposes of reconnaissance, surveillance, and other combat-related objectives. It is heavily influenced by the future prospects of warfare in an urban environment and involves the use of sensors, munitions, vehicles, robots, human-wearable biometrics, and other smart technology that is relevant on the battlefield. One of the examples of IOT devices used in the military is Xaver 1000 system. The Xaver 1000 was developed by Israel's Camero Tech, which is the latest in the company's line of "through wall imaging systems". The Xaver line uses millimeter wave (MMW) radar, or radar in the range of 30-300 gigahertz. It is equipped with an AI-based life target tracking system as well as its own 3D 'sense-through-the-wall' technology. The use of IOT in military is shown in fig 1.2.5

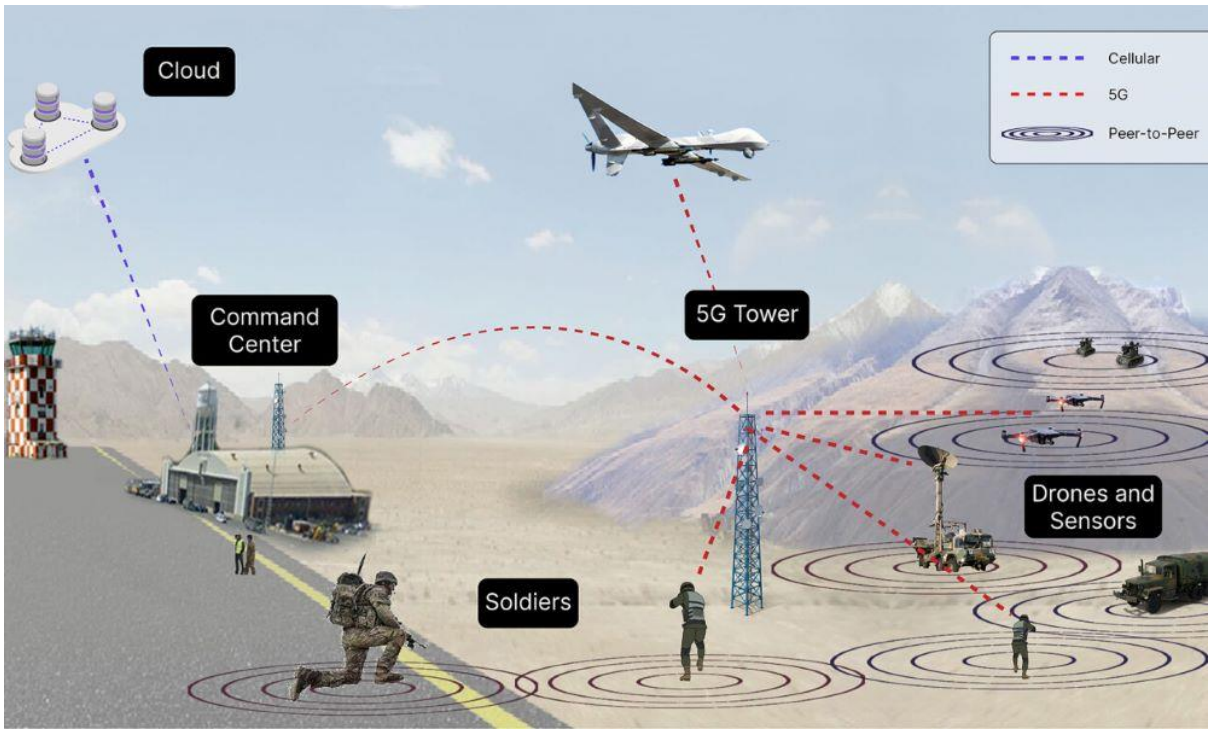


Fig 1.2.5 Use of IOT in military

1.3 Facial Recognition

A facial recognition system is a technology potentially capable of matching a human face from a digital image or a video frame against a database of faces. Such a system is typically employed to authenticate users through ID verification services, and works by pinpointing and measuring facial features from a given image. Development began on similar systems in the 1960s, beginning as a form of computer application. Since their inception, facial recognition

systems have seen wider uses in recent times on smartphones and in other forms of technology, such as robotics. Because computerized facial recognition involves the measurement of a human's physiological characteristics, facial recognition systems are categorized as biometrics. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognition, fingerprint image acquisition, palm recognition or voice recognition, it is widely adopted due to its contactless process. Facial recognition systems have been deployed in advanced human–computer interaction, video surveillance, law enforcement, passenger screening, decisions on employment and housing and automatic indexing of images. Facial recognition systems are employed throughout the world today by governments and private companies. Their effectiveness varies, and some systems have previously been scrapped because of their ineffectiveness. The use of facial recognition systems has also raised controversy, with claims that the systems violate citizens' privacy, commonly make incorrect identifications, encourage gender norms and racial profiling, and do not protect important biometric data. The appearance of synthetic media such as deepfakes has also raised concerns about its security. These claims have led to the ban of facial recognition systems in several cities in the United States. Growing societal concerns led social networking company Meta Platforms to shut down its Facebook facial recognition system in 2021, deleting the face scan data of more than one billion users. The change represented one of the largest shifts in facial recognition usage in the technology's history. IBM also stopped offering facial recognition technology due to similar concerns.

Techniques for Facial Recognition are

1.3.1. Traditional

Some face recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features.

Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation.

Recognition algorithms can be divided into two main approaches: geometric, which looks at distinguishing features, or photo-metric, which is a statistical approach that distils an image into values and compares the values with templates to eliminate variances. Some classify these algorithms into two broad categories: holistic and feature-based models. The former attempts to recognize the face in its entirety while the feature-based subdivide into components such as according to features and analyze each as well as its spatial location with respect to other features. The use of facial recognition with help of traditional method is shown in fig 1.3.1

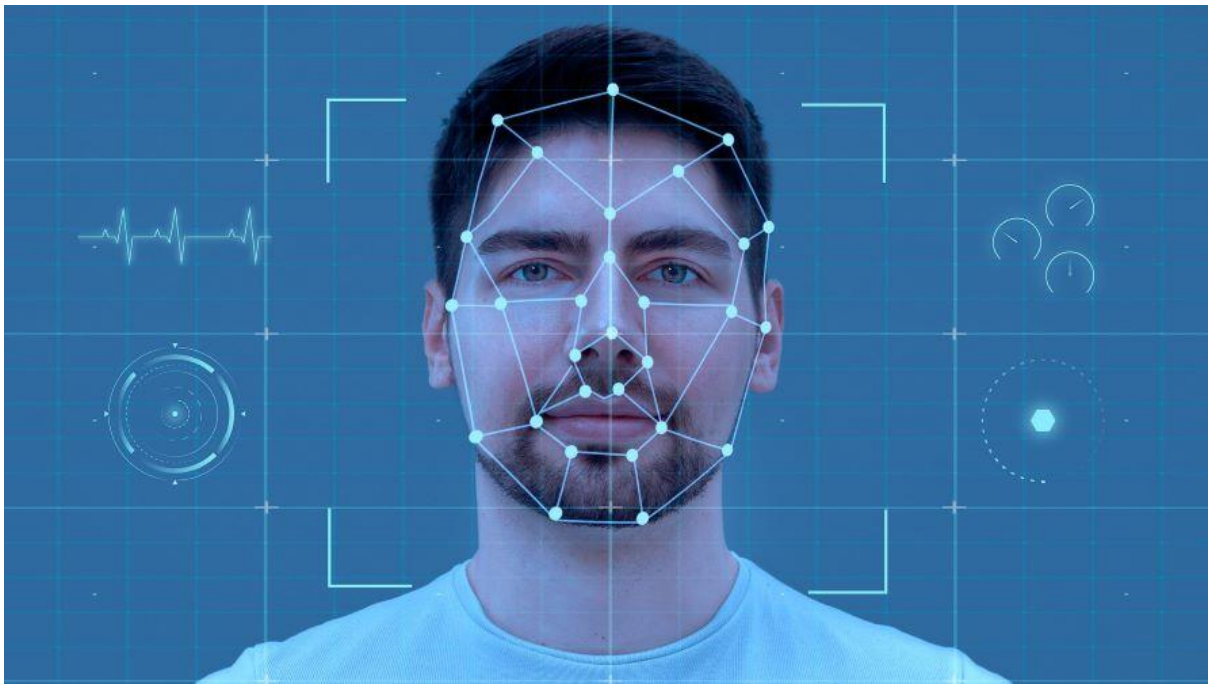


Fig 1.3.1 Use of facial recognition in Traditional method by using algorithms

1.3.2. Human Identification at distance

To enable human identification at a distance (HID) low-resolution images of faces are enhanced using face hallucination. In CCTV imagery faces are often very small. But because facial recognition algorithms that identify and plot facial features require high resolution images, resolution enhancement techniques have been developed to enable facial recognition systems to work with imagery that has been captured in environments with a high signal-to-noise ratio. Face hallucination algorithms that are applied to images prior to those images being submitted to the facial recognition system use example-based machine learning with pixel substitution or nearest neighbour distribution indexes that may also incorporate demographic and age related facial characteristics. Use of face hallucination techniques improves the performance of high-resolution facial recognition algorithms and may be used to overcome the

inherent limitations of super-resolution algorithms. Face hallucination techniques are also used to pre-treat imagery where faces are disguised. Here the disguise, such as sunglasses, is removed and the face hallucination algorithm is applied to the image. Such face hallucination algorithms need to be trained on similar face images with and without disguise. To fill in the area uncovered by removing the disguise, face hallucination algorithms need to correctly map the entire state of the face, which may be not possible due to the momentary facial expression captured in the low-resolution image. The use of facial recognition in HID is shown in fig 1.3.2

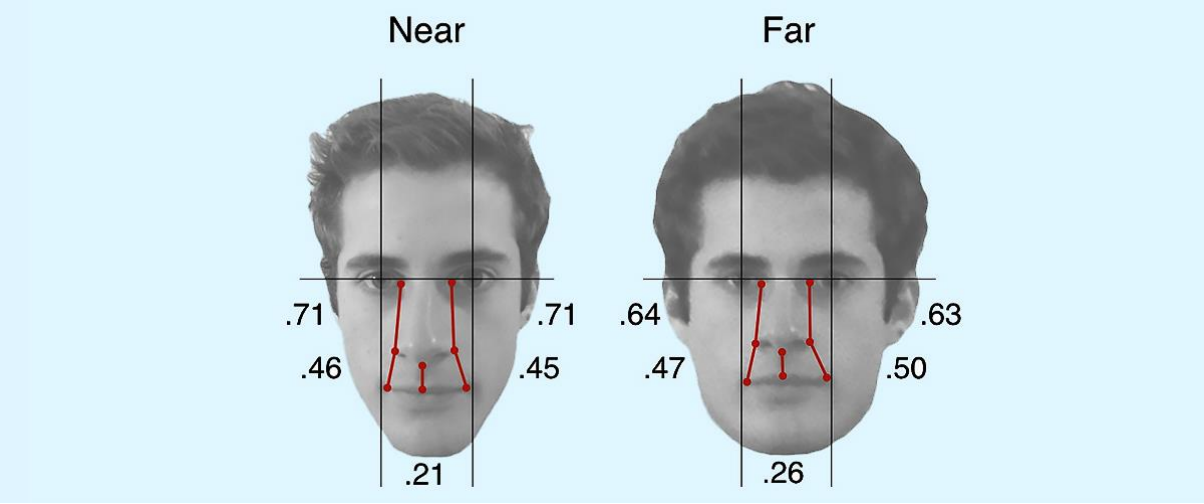


Fig 1.3.2 Use of facial recognition in HID

1.3.3. 3-Dimensional recognition

Three-dimensional face recognition technique uses 3D sensors to capture information about the shape of a face. This information is then used to identify distinctive features on the surface of a face, such as the contour of the eye sockets, nose, and chin. One advantage of 3D face recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view. Three-dimensional data points from a face vastly improve the precision of face recognition. 3D-dimensional face recognition research is enabled by the development of sophisticated sensors that project structured light onto the face. 3D matching technique are sensitive to expressions, therefore researchers at Technion applied tools from metric geometry to treat expressions as isometries. A new method of capturing 3D images of faces uses three tracking cameras that point at different angles; one camera will be pointing at the front of the subject, second one to the side, and third one at an angle. All these cameras will work together so it can track a

subject's face in real-time and be able to face detect and recognize. The use of facial recognition using 3-dimensional recognition is shown in fig 1.3.3.

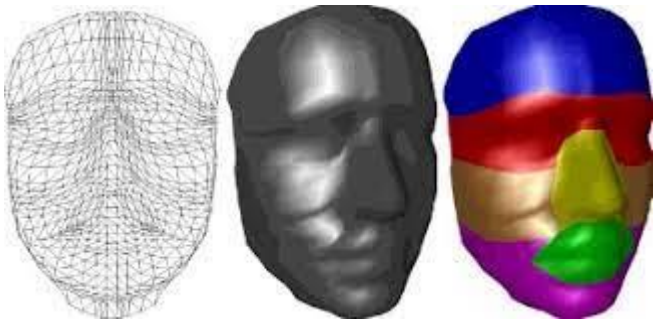


Fig 1.3.3 Use of facial recognition using 3-dimensional recognition

Our suggested system uses IoT and faces recognition to provide a cost and energy-efficient alternative for home protection. Many face recognition algorithms have been developed like using Eigenface, active appearance, Bayesian model, deep learning neural network, etc. Face recognition uses a web camera to capture a face image of the person and then the image captured is compared to the information in the database. They are then classified and saved in the database using recognized classes.

Considering the demand for an automatic security system (touchless lock and unlocking system) to avoid the manual operation that prevents touching the surfaces (prevent the spread of coronavirus). After going with an extensive literature survey, we found that there are various products available in the market, using Raspberry pi Camera Board, using Arduino and Arduino UNO and Android-based smartphones, ESP32 cam module. In this work, the AI-Thinker ESP32-CAM module has been considered as it is a low-cost development board with a very small OV2640 camera and a micro-SD card slot. The device's development includes various electronic components such as voltage regulator, switching module, microcontroller, power supply, and other electronic components. So, the developed door lock system can provide multiple benefits to society such as security, health and hygiene.

1.3 Proposed system:

In this work, a security system (door locking) is proposed based on facial recognition. The system consists of various hardware and software. The hardware includes the ESP32 cam module (which acts as the heart of the system) and multiple components, as mentioned in .

The process starts with uploading images by the user to the database. When an image arrives at the database, it gets analysed. Various facial parameters have been extracted

The testing and evaluation of the developed prototype has been started with an enrolment of face (store in database). This enrolment includes capturing the image by the camera, registering (users as authorized one), and storing (in the database). The implementation has been done using a facial recognition algorithm to authenticate the system. The performed experiment shows output with the trained face images in the database. The successful.

CHAPTER 2
LITERATURE SURVEY

2.1 Two Layer Security System using RFID and Face Detection

This project is mainly based on two-layer security system which includes the use of RFID (Radio Frequency Identification) and biometric detection which is facial detection in our project. With the use of RFID, we scan the tag (RFID card) in the hardware (RFID Reader) and the card id is generated for a particular user and then the webcam is opened for the facial detection, then the attendance is marked. The main purpose of our project is to ensure maximum security is provided for a particular student so that no other person can mark his attendance on his behalf. This two-layer security system is useful as it provides authentication in two steps. In first step, it will let person with authorized card to complete the first layer of security. After completion of first step, camera will open for facial recognition. After detection, attendance will be marked. RFID based recognition provides faster process while face detection provides security as security system is having database of only authorized person and detection will be done on this base only. So, by using these two steps, we can ensure better security for sure.

2.2 IOT based smart security system Face detection and Recognition

This paper proposes an IoT based smart security system using face detection and recognition. This system gives a smart approach for monitoring since all the present system just keep records of the events happening in front of the camera while this system alerts the owner. Once the PIR sensor detects any motion the raspberry pi camera module capture the image. This system uses the face recognition to reduce the errors occurred due to motion detection modules. System recognizes the face in the captured image and alerts the owner if unknown person is present in the image. Basically, in this paper, face detection and recognition algorithm based on LBPH approach is presented. LBPH (Local Binary Pattern Histogram) is a visual descriptor which is used for face recognition tasks. Also, CNN (Convolutional Neural Network) is another approach which is originally based on biological process and it is used for conveying the weight of particular image to recognize that image. In the proposed technique, the image is divided into multiple pixels and from that pixel threshold value is calculated. By using this value facial expressions, features of eyes, nose and lightning effects are calculated. Likewise, all the features are calculated and easily face is detected.

2.3 Face Recognition based smart attendance system using IOT

Attendance is a compulsory requirement of every organization. Maintaining attendance register daily is a difficult and time-consuming task. There are many automated methods for the same available like Biometric, RFID, eye detection, voice recognition, and many more. This paper provides an efficient and smart method for marking attendance. As it is known that primary identification for any human is its face, face recognition provides an accurate system which overcomes the ambiguities like fake attendance, high cost, and time consumption. This system uses face recognizer library for facial recognition and storing attendance. The absentee's supervisor or parents are informed through email regarding the absence of their employees or wards respectively. The objective of this project is to innovate existing projects with some added feature like large data storage and fast computing through less hardware cost.

2.4 RFID and Face Recognition based Security and access control

In the past, most of the companies were using a magnetic stripe card which is a type of card capable of storing data by modifying the magnetism of tiny iron-based magnetic particles on a band of magnetic material on the card. Card has a Personal Identification Number (PIN) allow to access account electronically. These days, RFID card has been used widely. In this paper the researcher used RFID card instead of a magnetic stripe card because it does not need a physical contact. The researcher combines a face recognition system with RFID system to support the security system. If the PIN in the RFID card matches the recognized face then a response is send to a remote station by a modem. This paper is important because it enhance security. In this paper the researcher mixes two type of security (FR and RFID) which make the security stronger and reliable. In turn, this gives the user more control. The MATLAB program is developed and it met the design criteria and solve the paper problem.

2.5 IOT based Facial Recognition Security System

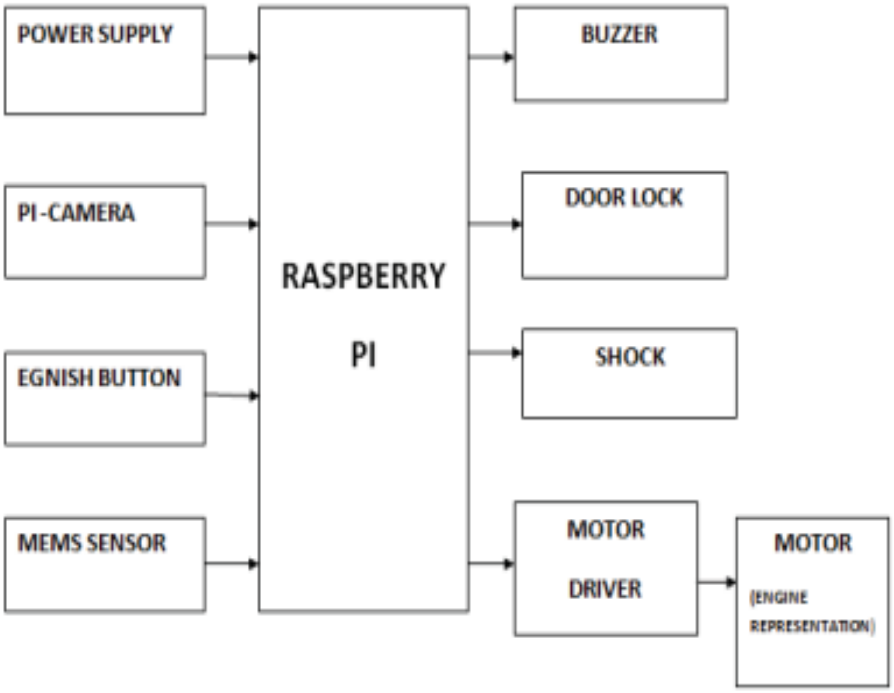
In recent years, with the demand for better security, computers have played a large role. Due to their precision, large memory banks and high computing power, considerable development has been made in face recognition. Computers now surpass humans in many face recognition tasks. A human being can remember limited number of faces. But a computer doesn't have any limits, and can hence be used where large databases of facial records are needed. Such a facial recognition system has many potential applications including crowd and airport surveillance, private security and improved human-computer interaction. Such a system is perfectly suited to fix security issues and offer flexibility to smart house control. This project is aimed to be a

complete system for face recognition: easy to build, cheap cost and effective. Main purpose is to be set as an alert for home visitors and provide information about the visitors in a dynamic website and phone application. It can also be used in other fields like industries, offices and even air-ports for identifying wanted people. Among the other bio-metric techniques, face recognition method offers one great advantage which is user friendliness.

CHAPTER 3

BLOCK DIAGRAM

3.1. BLOCK DIAGRAM



The list of hardware components present in above block diagram are:

1. Raspberry PI
2. PI-Camera
3. Egnish Button
4. Mems Sensor
5. Buzzer
6. Motor driver
7. Motor (Engine representation)

The List of Software components used are:

1. Python IDE
2. Fritzing

CHAPTER 4
HARDWARE COMPONENTS

4.1. Raspberry Pi

Raspberry Pi, developed by Raspberry Pi Foundation in association with Broadcom, is a series of small single-board computers and perhaps the most inspiring computer available today.

From the moment you see the shiny green circuit board of Raspberry Pi, it invites you to tinker with it, play with it, start programming, and create your own software with it. Earlier, the Raspberry Pi was used to teach basic computer science in schools but later, because of its low cost and open design, the model became far more popular than anticipated.

It is widely used to make gaming devices, fitness gadgets, weather stations, and much more. But apart from that, it is used by thousands of people of all ages who want to take their first step in computer science.

It is one of the best-selling British computers and most of the boards are made in the Sony factory in Pencoed, Wales.

Generations and Models

In 2012, the company launched the Raspberry Pi and the current generations of regular Raspberry Pi boards are Zero, 1, 2, 3, and 4.

Generation 1 Raspberry Pi had the following four options −

- Model A
- Model A +
- Model B
- Model B +

Among these models, the Raspberry Pi B models are the original credit-card sized format.

On the other hand, the Raspberry Pi A models have a smaller and more compact footprint and hence, these models have the reduced connectivity options.

Raspberry Pi Zero models, which come with or without GPIO (general-purpose input output) headers installed, are the most compact of all the Raspberry Pi boards types. The Raspberry pi is shown in below fig 4.1

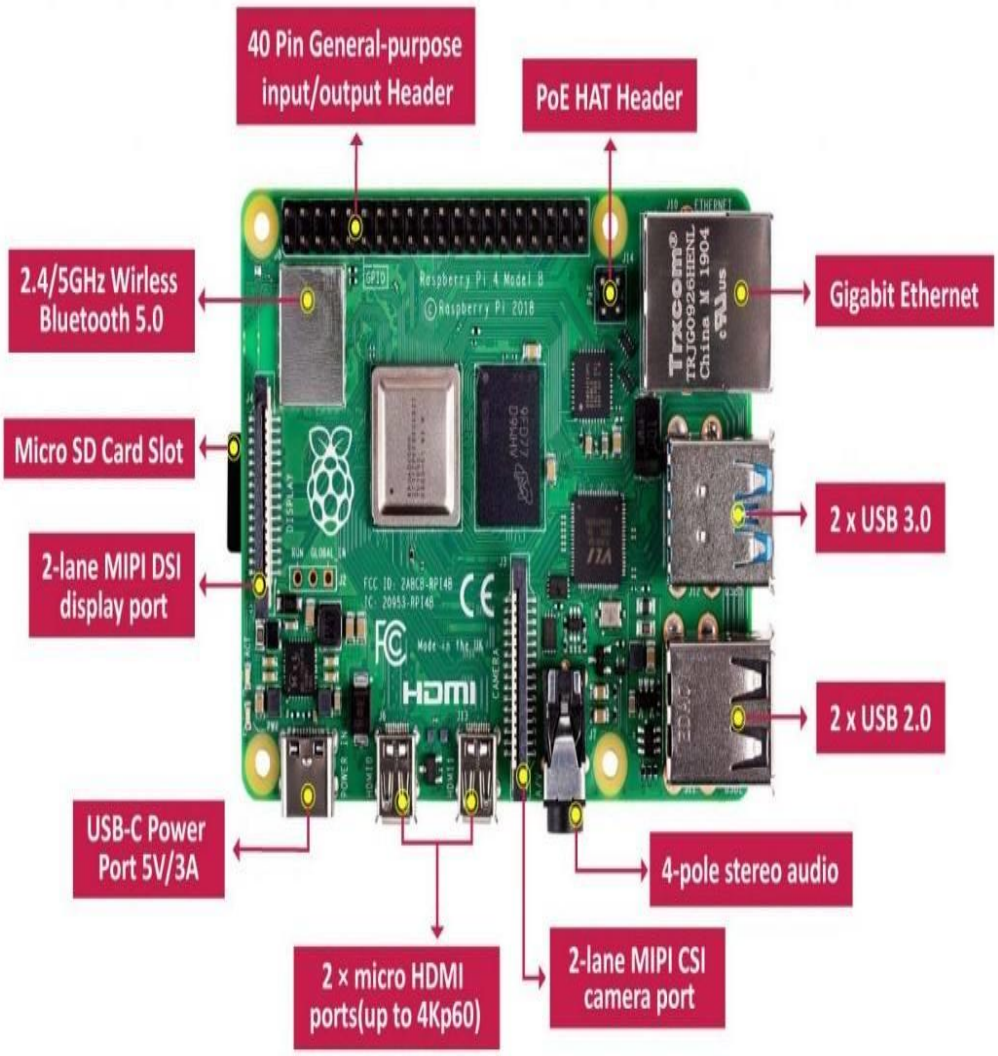


Fig 4.1 Raspberry Pi

4.1.1 HISTORY:

Software developer Eben Upton and Software Engineers Pete Lomas and David Braden formed the Raspberry Pi foundation in 2006. The main aim of this foundation was to devise a computer to inspire children. Hence, to reduce the cost, the early prototypes of the Raspberry Pi were based on the 8-bit Atmel AT mega microcontroller. On February 29th, 2012, the team started taking the orders for Model B and in the same year, they started its production run which

consisted of around 10,000 units. These models were manufactured by the founders in China and Taiwan. On February 4th, 2013, they started taking the orders for lower cost Model A. Similarly, on November 10th, 2014, the team launched even more low-cost Model A+. The cheapest Raspberry Pi Zero was launched on November 26th, 2015. The name Raspberry Pi was chosen with “Raspberry” as an ode to tradition of naming early computer companies after fruit. Here, "Pi" is for Python Programming Language.

In this modern age when computers are sleek, Raspberry Pi seems alien with tiny codes printed all over its circuit board. That is a big part of Raspberry Pi’s appeal. Let us have a look at what we can do with this appealing circuit board.

USES OF RASPBERRY PI

There are so many things you can make with a Raspberry Pi. And due to its extreme popularity, the number of third-party sensors, modules, and code libraries has grown to epic proportions. This has greatly expanded the possible applications for the Raspberry Pi. The applications really are

- Web servers
- Local network hubs for IoT
- Sensor control
- Robotics control
- Industrial equipment control
- Automated control systems
- Data processing

The table below gives the speed specifications of various Raspberry Pi models and generations focusing on the version’s weight, General Purpose Input/Output (GPIO), central processing unit (CPU) speed, Cores and Random-access memory (RAM).

Raspberry Pi Version	**Weight (in grams)**	**GPIO**	**CPU Speed**	**Cores**	**RAM**
Raspberry Pi 4 Model B	46	40 Pin	1.5 GHz	Quad	1,2,4, or 8 GB

Raspberry Pi 3 Model B+	50	40 Pin	1.4 GHz	Quad	1 GB
Raspberry Pi 3 Model B	40	40 Pin	1.2 GHz	Quad	1 GB
Raspberry Pi 3 Model A+	28	40 Pin	1.4 GHz	Quad	512 MB
Raspberry Pi Zero Wireless with Headers	10	40 Pin	1 GHz	Single	512 MB
Raspberry Pi Zero Wireless	10	40 Pin Unpopulated	1 GHz	Single	512 MB
Raspberry Pi Zero	8	40 Pin Unpopulated	1 GHz	Single	512 MB
Raspberry Pi 2 Model B	42	40 Pin	1.2 GHz	Quad	1 GB
Raspberry Pi 1 Model B +	42	40 Pin	700 MHz	Single	512 MB
Raspberry Pi 1 Model B	38	21 Pin (26 Pin Header)	700 MHz	Single	512 MB
Raspberry Pi 1 Model A+	23	40 Pin	700 MHz	Single	512 MB

Raspberry Pi 1 Model A	30	21 Pin (26 Pin Header)	700 MHz	Single	256

These are the list of various Raspberry pi version along with their weight, GPIO, CPU Speed, Cores and RAM. The new model of Raspberry Pi is Raspberry PI model 5. It has various key features like 2.4GHz quad-core 64-bit ARM cortex-A76 CPU, 4Kp60 HEVC decoder, real time clock, power button, Raspberry PI standard 40-pin GPIO header, High speed microSD card interface with SDR104 mode support, Dual band 802.11ac Wi-Fi, Bluetooth low energy.

Requirements

To use your Raspberry Pi board, you need to buy a few other bits and pieces. Following is the checklist of what else we might need −

Monitor

The Raspberry Pi uses a high-definition multimedia interface (HDMI) connection for video feed, and you can connect your monitor directly with this interface connection, if your monitor has an HDMI socket.

Television

In the similar way, if you have High-Definition Television (HD TV), you can also connect it to your Raspberry Pi using an HDMI socket. It will give you a crisper picture.

USB hub

Depending on the model, Raspberry Pi has 1, 2, or 4 Universal Serial Bus (USB) sockets. You should consider using powered USB to connect other devices to your Raspberry Pi at the same time.

Keyboard and Mouse

Raspberry Pi only supports the USB keyboards and mouse. If you are using keyboards and mouse with PS/2 connectors, you need to replace them with Raspberry Pi.

SD or MicroSD card

As we know that the Raspberry Pi does not have a hard drive, so we need to use SD cards or MicroSD cards (depending on the model) for storage. The Raspberry Pi boots up from this SD card, much like a PC boots up from a hard drive. There are many brands and sizes of microSD cards that are compatible with the Raspberry Pi. Some of the popular ones include:

- **SanDisk Extreme Pro**: Known for its speed, it's often considered the best overall[3].
- **SanDisk Ultra**: This is a cost-effective option that provides good performance[4].
- **Samsung EVO Plus**: This is another popular choice among Raspberry Pi users

USB Wi-Fi adapter

If you are going to use model A and A+ then, you need to buy a USB Wi-Fi adapter for connecting to the internet. This should be done because these Raspberry models do not have an Ethernet socket.

External hard drive

If you want to share your collection of music and movies, you need to use an external hard drive with your Raspberry Pi model. You can connect the same by using a powered USB cable.

Raspberry Pi Camera Module

The Raspberry Pi camera module originated at Raspberry Pi foundation. It is an 8MP (megapixel) fixed focus camera that can be used to shoot high-definition video and take still photos. For wildlife photography at night, it provides another version without an infrared filter.

Speakers

The Raspberry Pi has a standard audio out socket. This socket is compatible with headphones and speakers that use a 3.5mm audio jack. We can plug headphones directly to it.

Power supply

For power supply, it uses a Micro USB connector. Hence theoretically, it is compatible with a mobile phone and tablet charger.

4.2 PI CAMERA

The Raspberry Pi camera module can be used to take high-definition video, as well as still photographs. It's easy to use for beginners but has plenty to offer advanced users if you're looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries we bundle with the camera to create effects.

If you are interested in the nitty-gritty, you'll want to know that the module has a five-megapixel fixed-focus camera that supports 1080p30, 720p60, and VGA90 video modes, as well as stills capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Picamera Python library.

The camera module is very popular in-home security applications, and in wildlife camera traps. You can also use it to take snapshots.

Features

- 5MP sensor
- Wider image, capable of 2592x1944 stills, 1080p30 video
- 1080p video supported
- CSI
- Size: 25 x 20 x 9 mm

Camera Details

The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90. Installation involves connecting the ribbon cable to the CSI connector on the Raspberry Pi board. This can be a little tricky, but if you watch the videos that demonstrate how it is done, you should not have any trouble. When you purchase the camera, you will receive a small camera board and cable. You will want to devise some method of supporting the camera to use. The PI camera is shown in below fig 4.2.

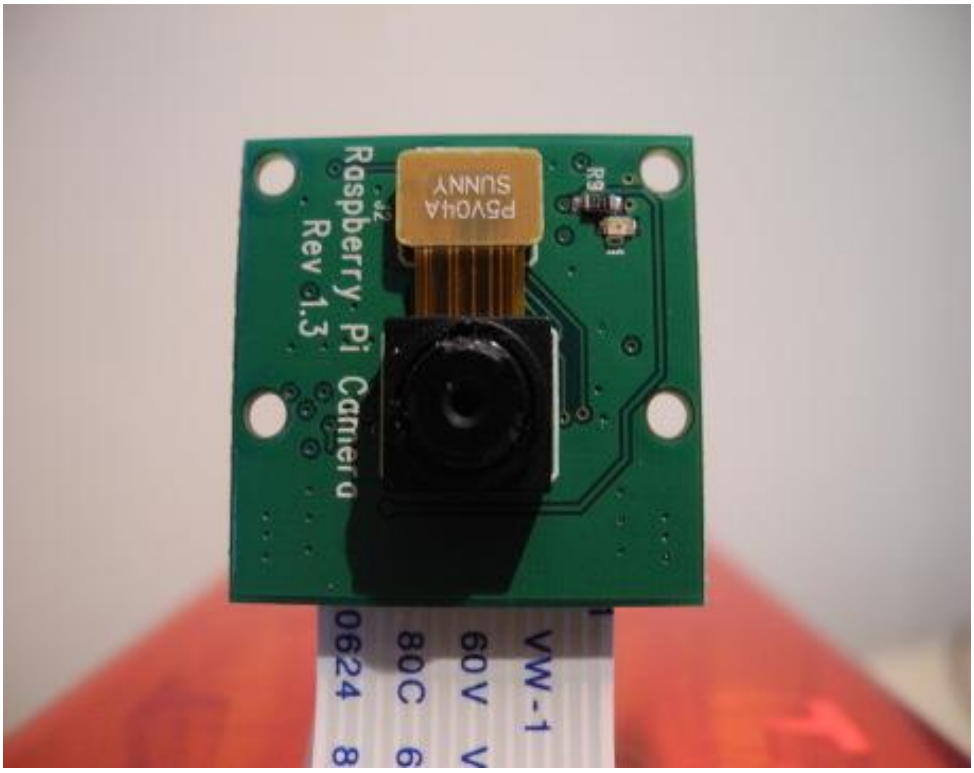


Fig 4.2 PI camera

We want to devise some method of supporting the camera to use it. Some camera stands and Raspberry Pi cases are now available. We can also rig up something simple yourself if you wish. It is attached to a case using a small piece of plastic and double-sided tape, as shown. The Pi camera using a small piece of plastic and double-sided tape is shown in fig 4.3

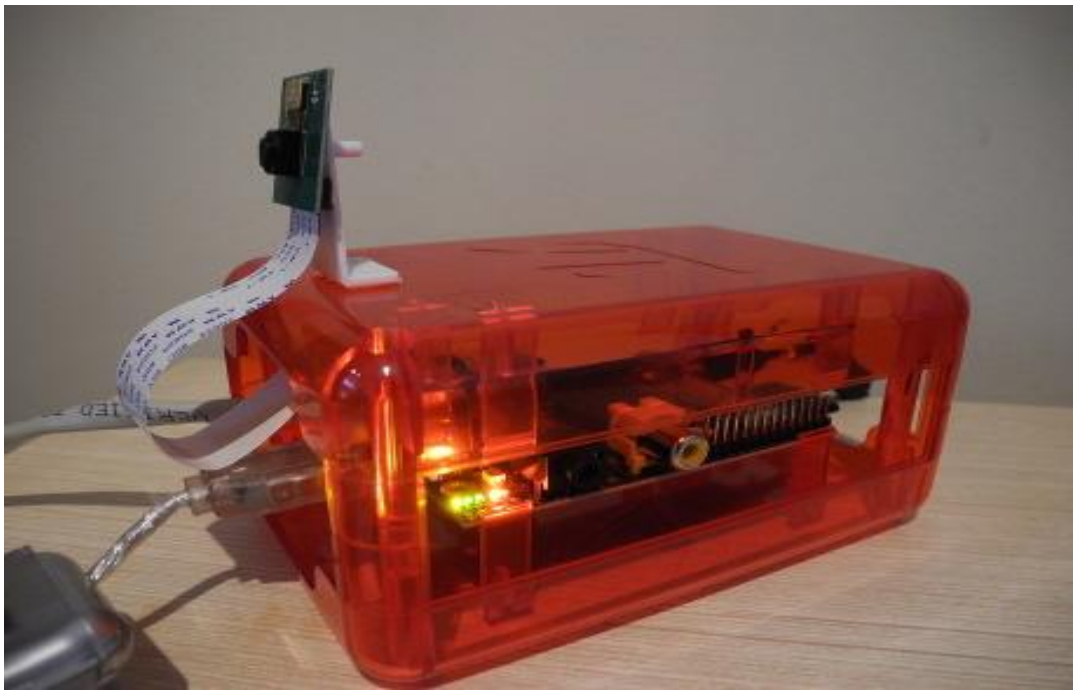


Fig 4.3 Pi camera using a small piece of plastic and double sided tape

Connect to 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.
- Update the SD card

In order to use the camera, you must be using a recent operating system that knows that the camera exists. The easiest way to do this is to grab the latest Raspbian image from the RaspberryPi.org site and create a fresh SD card.

- Enable camera in raspi-config settings

Reboot. If you are using a fresh image the raspi-config utility should load. If it doesn't then you can run it manually using: sudo raspi-config Select the "Camera" option and press "Enter". The configuration of Pi camera is shown in fig 4.4.

Fig 4.4 Configuration of PI camera

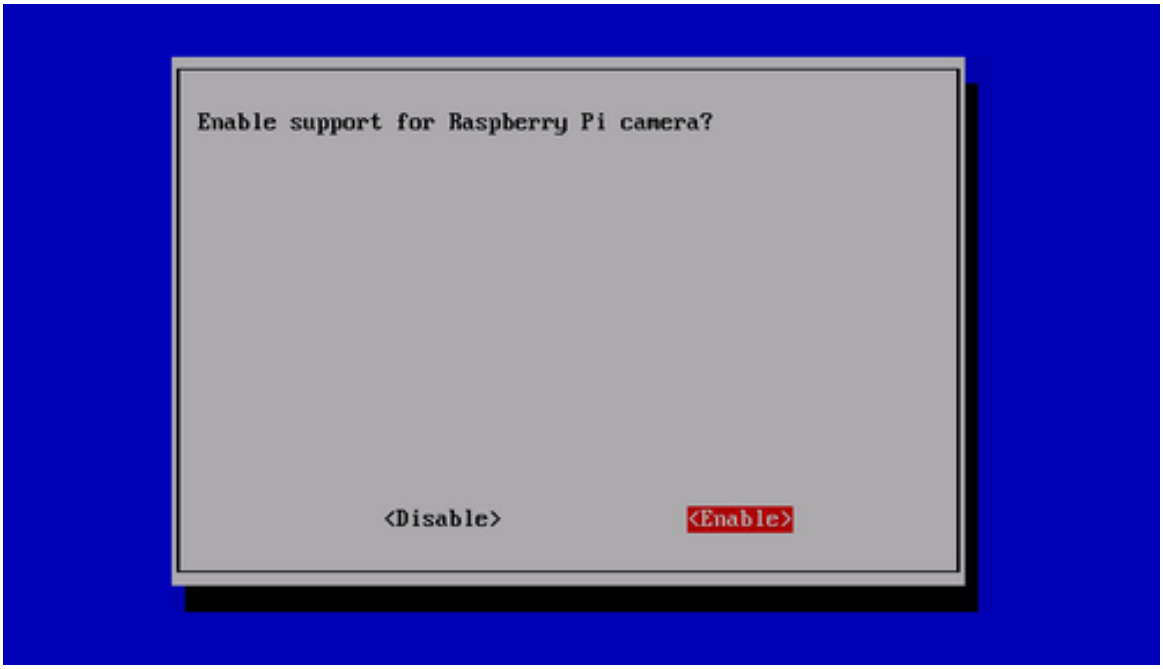


Fig 4.5 Support of PI camera to Raspberry PI

Select “Yes” and press “Enter”. Your Pi will reboot.

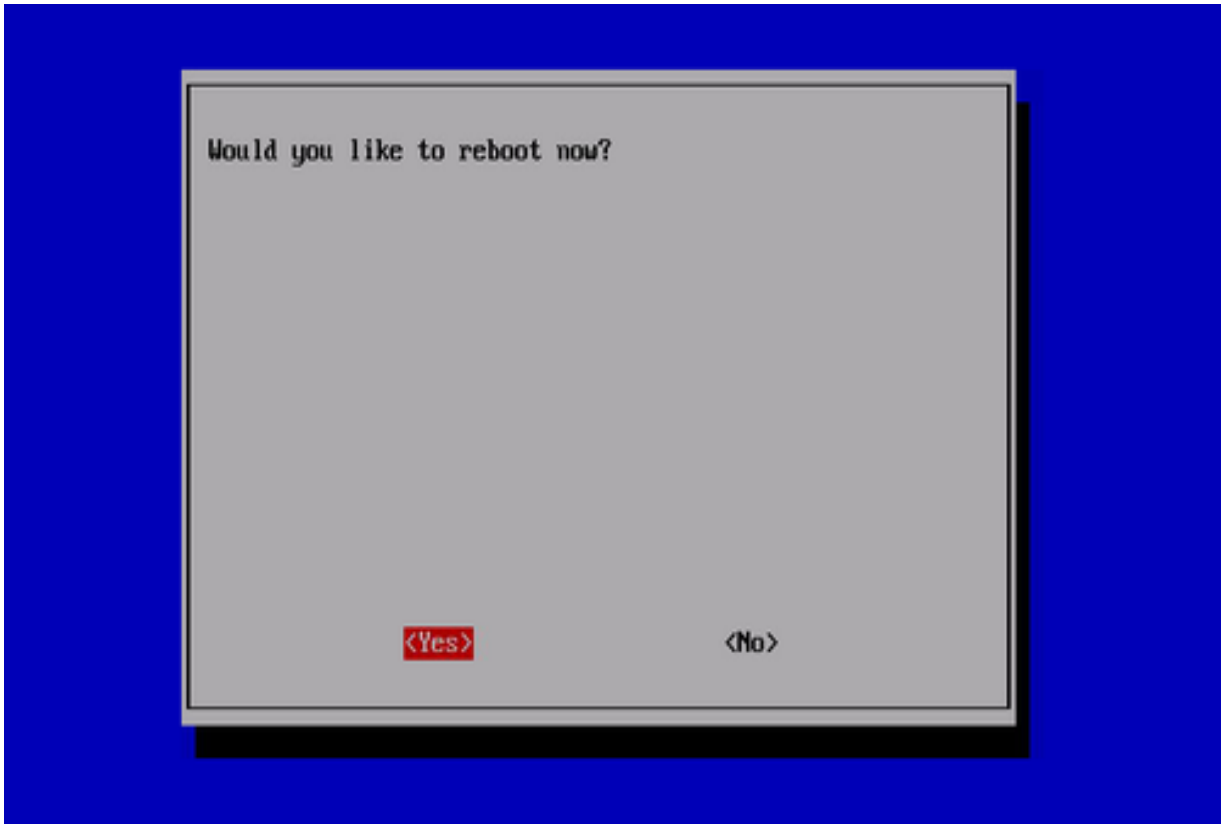


Fig 4.6 Reboot process of PI camera

Updating your operating and enabling the camera using Raspi-config did two things. It told your Pi that there is a camera attached and it added two command line utilities. raspistill

raspivid These allow you to capture still photos and HD video respectively. Fig 4.5 shows the support of PI camera to Raspberry Pi. Fig 4.6 shows the after the support Pi reboot occurs.

Software

Since its inception, the camera is supported in the latest version of Raspbian, the preferred operating system for Raspberry Pi. The instructions in this blog post assume you are running Raspbian. The first step is to get the latest Raspberry Pi firmware, which supports the camera. You can do that from a console by running:

- sudo apt-get update
- sudo apt-get upgrade

You then need to enable the camera from the Raspberry Pi configuration program by running:

- sudo raspi-config

Choose "camera" from the program and then select "Enable support for Raspberry Pi camera". You should then reboot when prompted by the raspi-config program. The camera will be enabled on subsequent boots of the Raspberry Pi.

Several applications should now be available for the camera: the rapistill program captures images, raspivid captures videos, and raspiyuv takes uncompressed YUV format images. These are command line program. They accept a number of options, which are documented if you run the commands without options. That reference also describes some more sophisticated things you can do, like streaming the video over the network and viewing it on another computer.

The following shell command runs the video capture program with a preview showing all the built-in camera effects and makes for an interesting demonstration:

do

    echo$effect

    raspivid -d -ifx$effect

done

If you want to examine the source code for the programs, report bugs or compile them yourself, they are maintained at the project on github. You can either cross-compile or build the tools natively on the Raspberry Pi.

- User Space V4L2 Driver

The camera drivers are proprietary in the sense that they do not follow any standard APIs. That means that applications have to be written specifically for the Raspberry Pi camera. Under Linux, the standard API for cameras (including web cams) is V4L (Video for Linux), and a number of applications have been written that support any camera with a V4L driver. An independent developer has now written a user space V4L driver for the Raspberry Pi camera. With that driver, you can use generic Linux applications written for cameras. The driver has a few limitations: it is closed sourced, and can be a little slow because it runs as a user program rather than a kernel driver. The program worked reasonably well when I tested it and it is expected to continue to improve.

- Official V4L2 Driver

Recognizing that a V4L driver is needed, the Raspberry Pi Foundation reported that they were working with Broadcom to develop an official kernel V4L driver. As a kernel driver, it should be faster than the user space driver. The official driver became available in December 2013. The driver

is still quite new and not many people appear to have tried it yet. The latest Raspbian distribution and latest Raspberry Pi boot firmware is required for use. You will also need to build some code yourself. If you want to try it, some brief instructions showing the commands to build it are listed below (these commands should be run from a shell).

#Get the latest Raspbian packages

sudo apt-get update

sudo apt-get upgrade

#Get the latest firmware

Sudo rpi-update

#Get the source code for the v4L utilities

git clone git

cd v4l-utils

#Install some packages needed to build it libjpeg62-dev

#Configure and build it

autoreconf -vfi

. /Configure

Make

sudo make install

Building the software should take about fifteen minutes. You need to have the camera enabled and sufficient Graphics Processing Unit (GPU) memory configured. Here are some example commands to get started:

#Load the module

Sudo modprobe bcm2835-v4l2

#Control the viewfinder

v4l2-ctl --overlay=1 #enable viewfinder

v4l2-ctl --overlay=0 #disable viewfinder

#Record a video

v4l2-ctl --set-fmt-video=width=1920, height=1088, pixelformat=4

v4l2-ctl --stream-mmap=3 –stream-count=100 –stream-to=somefile.264

#Capture a JPEG image

v4l2-ctl --set-fmt-video=width=2592, height=1944, pixelformat=3

v4l2-ctl --stream-mmap=3 –stream-count=1 –stream-to=somefile.jpg

Set the video bitrate

v4l2-ctl –set-ctrl video bitrate=10000000

#List the supported formats

v4l2-ctl –list-formats

Connect the Camera Module

Ensure your Raspberry Pi is turned off.

1. Locate the Camera Module port
2. Gently pull up on the edges of the port’s plastic clip
3. Insert the Camera Module ribbon cable; make sure the connectors at the bottom of the ribbon cable are facing the contacts in the port.

4. Push the plastic clip back into place

Now, switch on the Raspberry Pi and foll0w the below steps.

- Start up your Raspberry Pi.
- Go to the main menu and open the Raspberry Pi Configuration tool.

The configuration of Raspberry is selected is shown in fig 4.7

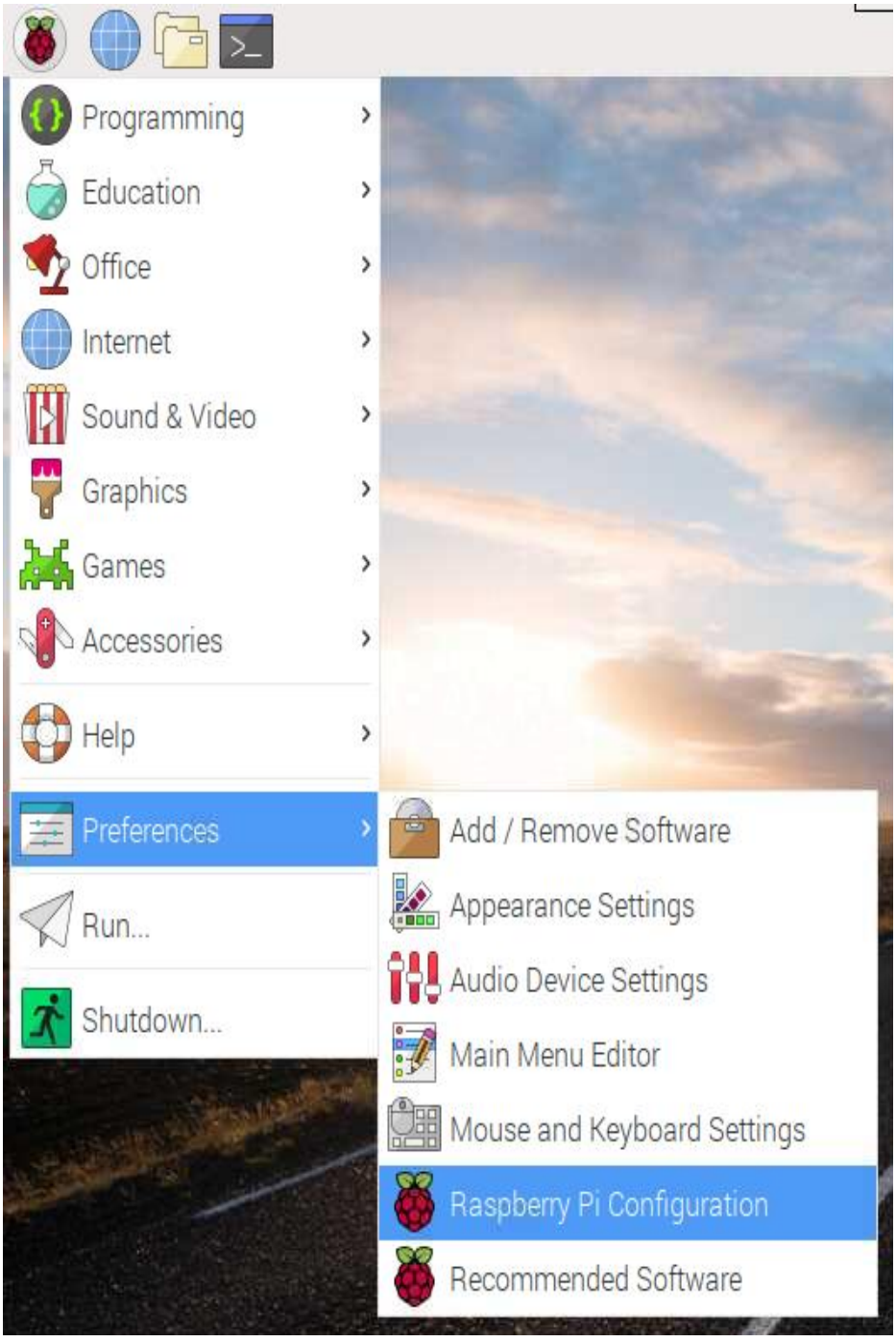


Fig 4.7 Raspberry Pi configuration

- Select the Interfaces tab and ensure that the camera is enabled. The selection of configuration is shown in fig 4.8

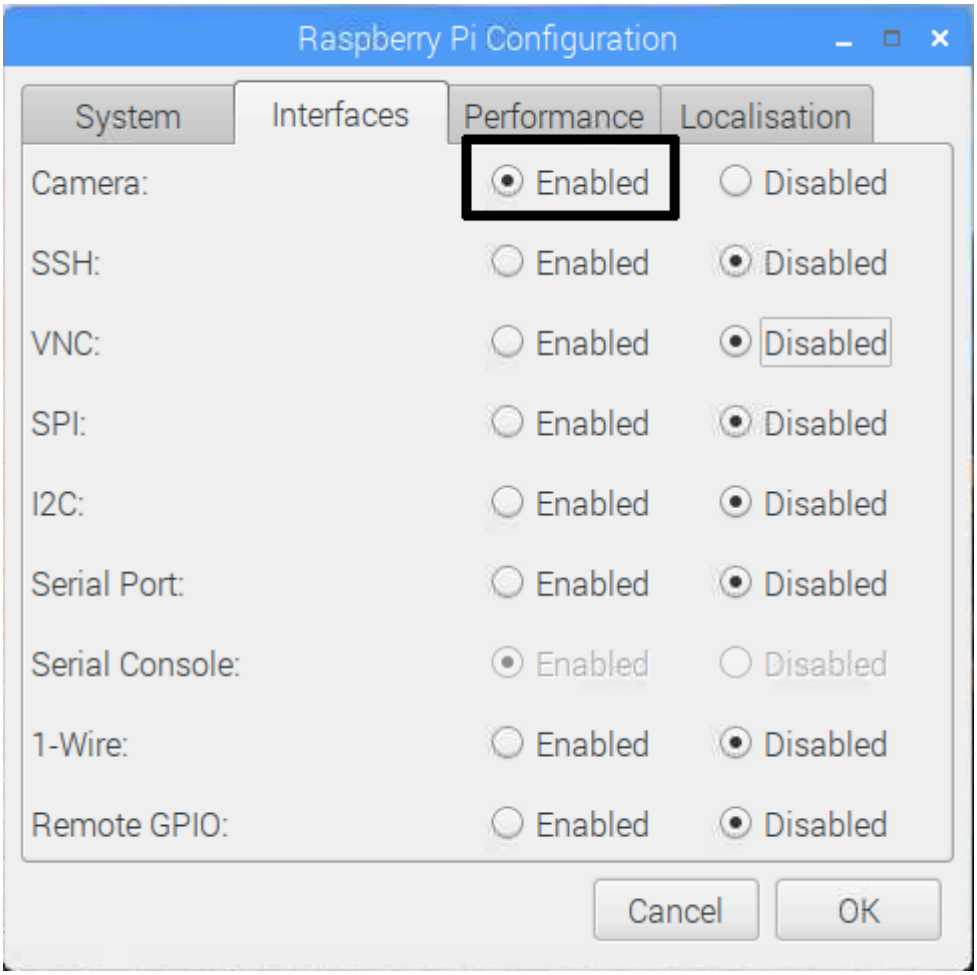


Fig 4.8 Selection of configuration

- Reboot your Raspberry Pi.

4.3. MEMS SENSOR

Micro-Electro-Mechanical Systems, or MEMS, is a technology that in its most general form can be defined as miniaturized mechanical and electro-mechanical elements (i.e., devices and structures) that are made using the techniques of microfabrication. The critical physical dimensions of MEMS devices can vary from well below one micron on the lower end of the dimensional spectrum, all the way to several millimetres. Likewise, the types of MEMS devices can vary from relatively simple structures having no moving elements, to extremely complex electromechanical systems with multiple moving elements under the control of integrated microelectronics. The one main criterion of MEMS is that there are at least some elements

having some sort of mechanical functionality whether or not these elements can move. The term used to define MEMS varies in different parts of the world. In the United States they are predominantly called MEMS, while in some other parts of the world they are called "Microsystems Technology" or "micromachined devices".

While the functional elements of MEMS are miniaturized structures, sensors, actuators, and microelectronics, the most notable (and perhaps most interesting) elements are the microsensors and micro actuators. Microsensors and micro actuators are appropriately categorized as "transducers", which are defined as devices that convert energy from one form to another. In the case of microsensors, the device typically converts a measured mechanical signal into an electrical signal. The components of MEMS are shown in fig 4.9.

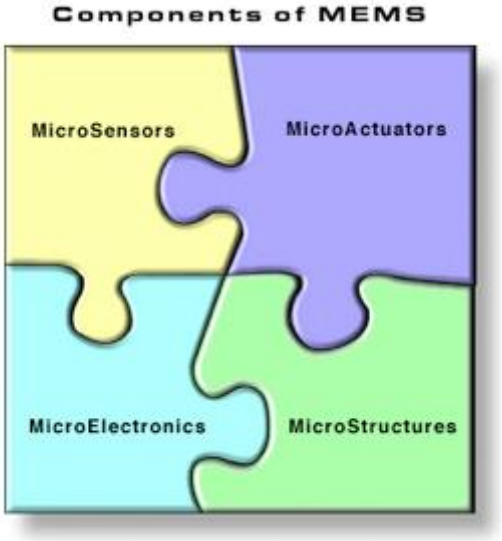


Fig 4.9 Components of MEMS

Over the past several decades MEMS researchers and developers have demonstrated an extremely large number of microsensors for almost every possible sensing modality including temperature, pressure, inertial forces, chemical species, magnetic fields, radiation, etc. Remarkably, many of these micromachined sensors have demonstrated performances exceeding those of their macroscale counterparts. That is, the micromachined version of, for example, a pressure transducer, usually outperforms a pressure sensor made using the most precise macroscale level machining techniques. Not only is the performance of MEMS devices exceptional, but their method of production leverages the same batch fabrication techniques used in the integrated circuit industry – which can translate into low per-device production costs, as well as many other benefits. Consequently, it is possible to not only achieve stellar

device performance, but to do so at a relatively low-cost level. Not surprisingly, silicon based discrete microsensors were quickly commercially exploited and the markets for these devices continue to grow at a rapid rate.

More recently, the MEMS research and development community has demonstrated a number of micro actuators including: microvalves for control of gas and liquid flows; optical switches and mirrors to redirect or modulate light beams; independently controlled micromirror arrays for displays, micro resonators for a number of different applications, micropumps to develop positive fluid pressures, micro flaps to modulate airstreams on air foils, as well as many others. Surprisingly, even though these micro actuators are extremely small, they frequently can cause effects at the macroscale level; that is, these tiny actuators can perform mechanical feats far larger than their size would imply. For example, researchers have placed small micro actuators on the leading edge of air foils of an aircraft and have been able to steer the aircraft using only these microminiaturized devices. The MEMS sensor is shown in fig 4.10. The MEMS based micro actuator is shown in fig 4.11

Fig 4.10 MEMS Sensor

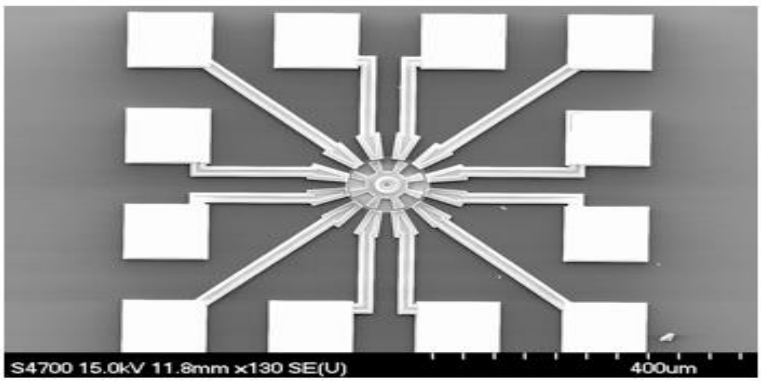


Fig 4.11 MEMS-based micro actuator.

The real potential of MEMS starts to become fulfilled when these miniaturized sensors, actuators, and structures can all be merged onto a common silicon substrate along with integrated circuits (i.e., microelectronics). While the electronics are fabricated using integrated circuit (IC) process sequences (e.g., CMOS, Bipolar, or BICMOS processes), the micromechanical components are fabricated using compatible "micromachining" processes that selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices. It is even more interesting if MEMS can be merged not only with microelectronics, but with other technologies such as photonics, nanotechnology, etc. This is sometimes called “heterogeneous integration.” Clearly, these technologies are filled with numerous commercial market opportunities.

While more complex levels of integration are the future trend of MEMS technology, the present state-of-the-art is more modest and usually involves a single discrete microsensor, a single discrete micro actuator, a single microsensor integrated with electronics, a multiplicity of essentially identical microsensors integrated with electronics, a single micro actuator integrated with electronics, or a multiplicity of essentially identical micro actuators integrated with electronics. Nevertheless, as MEMS fabrication methods advance, the promise is an enormous design freedom wherein any type of microsensor and any type of micro actuator can be merged with microelectronics as well as photonics, nanotechnology, etc., onto a single substrate. The device used as both microsensor as well as micro actuator is shown in fig 4.12

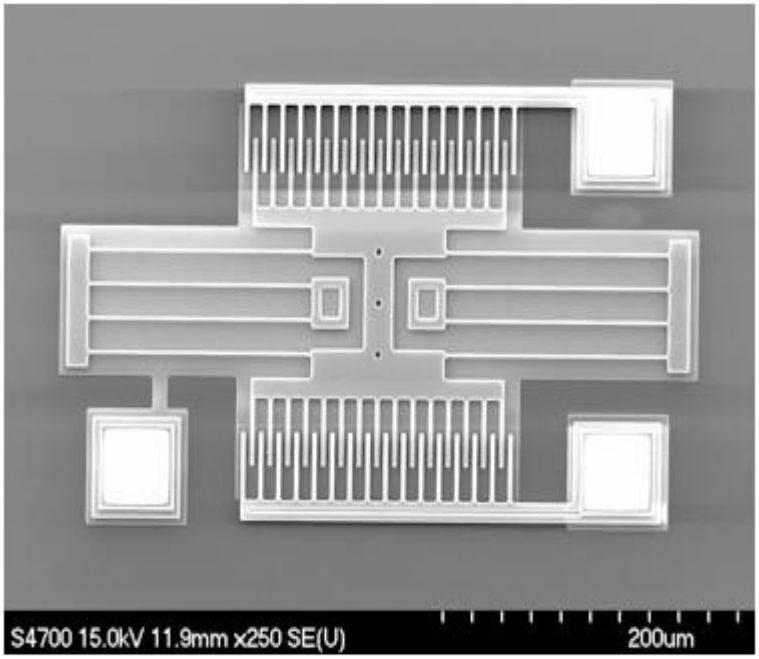


Fig 4.12 device can be used as both a microsensor as well as a micro actuator.

This vision of MEMS whereby microsensors, micro actuators and microelectronics and other technologies, can be integrated onto a single microchip is expected to be one of the most important technological breakthroughs of the future. This will enable the development of smart products by augmenting the computational ability of microelectronics with the perception and control capabilities of microsensors and micro actuators. Microelectronic integrated circuits can be thought of as the "brains" of a system and MEMS augments this decision-making capability with "eyes" and "arms", to allow microsystems to sense and control the environment. Sensors gather information from the environment through measuring mechanical, thermal, biological, chemical, optical, and magnetic phenomena. The electronics then process the information derived from the sensors and through some decision-making capability direct the actuators to respond by moving, positioning, regulating, pumping, and filtering, thereby controlling the environment for some desired outcome or purpose. Furthermore, because MEMS devices are manufactured using batch fabrication techniques, like ICs, unprecedented levels of functionality, reliability, and sophistication can be placed on a small silicon chip at a relatively low cost. MEMS technology is extremely diverse and fertile, both in its expected application areas, as well as in how the devices are designed and manufactured. Already, MEMS is revolutionizing many product categories by enabling complete systems-on-a-chip to be realized. Nanotechnology is the ability to manipulate matter at the atomic or molecular level to make something useful at the nano-dimensional scale. Basically, there are two approaches in implementation: the top-down and the bottom-up. In the top-down approach, devices and structures are made using many of the same techniques as used in MEMS except they are made smaller in size, usually by employing more advanced photolithography and etching methods. The bottom-up approach typically involves deposition, growing, or self-assembly technologies. The advantages of nano-dimensional devices over MEMS involve benefits mostly derived from the scaling laws, which can also present some challenges as well. Fig 4.13 shows array of sub-micron using nanotechnology.

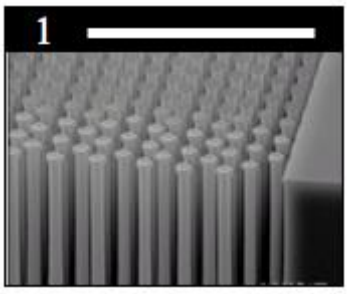


Fig 4.13 An array of sub-micron posts made using nanotechnology

Some experts believe that nanotechnology promises to: a). allow us to put essentially every atom or molecule in the place and position desired – that is, exact positional control for assembly, b). allow us to make almost any structure or material consistent with the laws of physics that can be specified at the atomic or molecular level; and c). allow us to have manufacturing costs not greatly exceeding the cost of the required raw materials and energy used in fabrication (i.e., massive parallelism). Although MEMS and Nanotechnology are sometimes cited as separate and distinct technologies, in reality the distinction between the two is not so clear-cut. In fact, these two technologies are highly dependent on one another. The well-known scanning tunnelling-tip microscope (STM) which is used to detect individual atoms and molecules on the nanometre scale is a MEMS device. Similarly, the atomic force microscope (AFM) which is used to manipulate the placement and position of individual atoms and molecules on the surface of a substrate is a MEMS device as well. In fact, a variety of MEMS technologies are required in order to interface with the nano-scale domain.

Different Types of MEMS Sensors

One of the major implementors of MEMS technology is the automotive industry. Modern cars use a lot of sensors and most of them are MEMS-based devices. The following is a list of MEMS Sensors that are used in a modern car.

- Accelerometers – For Electronic Stability Control and Airbag deployment.
- Inertial Measurement Units or IMU (they are a combination of MEMS Accelerometer and MEMS Gyroscope) – For measuring yaw, pitch, and roll for autonomous driving.
- Magnetometer – For direction used in navigation.
- Pressure and Inertial Sensor – For braking control.
- Pressure Sensor – Tire Pressure Monitoring System.
- Airflow Sensor – Air intake monitoring.
- Fuel Sensor – Fuel Level Indicator.
- Impact and Crash Sensor – Impact detection and Airbag deployment.
- MEMS Microphone – For communication and Noise Cancellation.
- Temperature Sensor – For Automatic Climate Control and Engine Temperature Monitoring.
- Gyro Sensor – Detects travelling direction
- Magnetic Sensor – Navigation, tracking
- Gas Sensor – Monitoring gases like Carbon monoxide, Nitrogen Oxide.
- There are many other MEMS-based devices for various applications.

MEMS Applications

MEMS technology can be incorporated into a wide variety of electronic components. The companies that make these components would presumably claim that a MEMS implementation is superior to whatever was used before the MEMS version became available. It would be difficult to verify enough of these claims to justify a generalized statement along the lines of "MEMS devices offer significantly better performance than non-MEMS devices." However, my general impression is that in many situations MEMS is indeed a significant step forward and, if performance or ease of implementation is a priority in your design, I would look at MEMS devices first.

In the context of electrical engineering, MEMS technology has been incorporated into four product categories:

- Audio
- Sensors
- Switches
- Oscillators

There might be some less-common products that don't fit into one of these categories; if you're aware of something that I overlooked, feel free to let us know in the comments.

Audio

In the audio domain, we have MEMS microphones and MEMS speakers. The basic characteristics of a MEMS mic are conveyed by the following diagram. The use of MEMS sensor in audio is shown in fig 4.14.

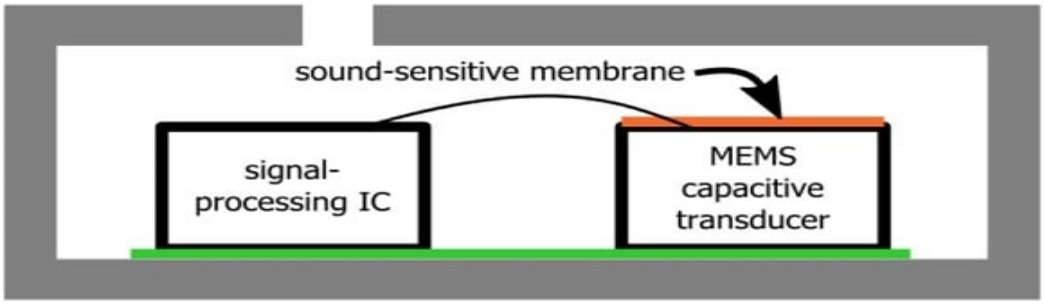


Fig 4.14 MEMS Sensor in audio

Sensors

Sensors are the dominant application of MEMS techniques; there are MEMS gyroscopes, inclinometers, accelerometers, flow sensors, gas sensors, pressure sensors, and magnetic field sensors.

Switches

Electrically controlled switches are particularly interesting application of MEMS technology. The ADGM1004 is easy to control, works with signal frequencies from 0 Hz to over 10 GHz, has less than 1 nA of leakage current in the off state, and provides an actuation lifetime of at least one billion cycles.

Oscillators

Combining a micromachined resonator with excitation circuitry and sustaining circuitry results in a MEMS oscillator. If you'd like to investigate an actual MEMS component, you can check out a news article from 2017 in which I discussed the SiT2024B MEMS oscillator from SiTime. The use of MEMS in oscillator is shown in fig 4.15

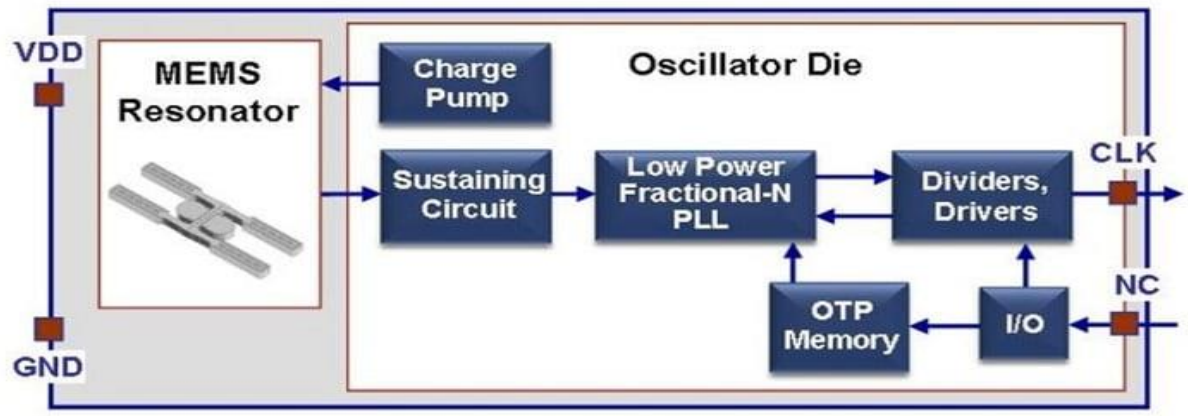


Fig 4.15 MEMS Sensor in Oscillator

MEMS Advantages

The advantages of MEMS sensor include the following.

- The manufacturing of MEMS is semiconductor IC manufacturing like low-cost mass invention, consistency is also essential to MEMS devices.

- The size of sensor sub-components will be within 1 to 100 micrometres range as well as the MEMS device size will determine 20 micro-meter to a millimetre range.
- Power consumption is very low.
- Simple to incorporate into systems or change
- The thermal constant is small
- These can be highly opposed to shock, radiation, and vibration.
- Better thermal development tolerance
- Parallelism

4.4 MOTOR DRIVER

Motor drive means a system that includes a motor. An adjustable speed motor drive is a system that includes a motor that has multiple operating speeds. A variable speed motor drive is a system that includes a motor and is continuously variable in speed. If the motor is generating electrical energy rather than using it – this could be called a generator drive but is often still referred to as a motor drive. A variable frequency drive (VFD) or variable speed drive (VSD) describes the electronic portion of the system that controls the speed of the motor. More generally, the term drive describes equipment used to control the speed of machinery. Many industrial processes such as assembly lines must operate at different speeds for different products. Where process conditions demand flow adjustment from a pump or fan, varying the speed of the drive may save energy compared with other techniques for flow control. Where speeds may be selected from several different pre-set ranges, usually the drive is said to be adjustable speed. If the output speed can be changed without steps over a range, the drive is usually called variable speed. Adjustable and variable speed drives may be purely mechanical (termed variations), electromechanical, hydraulic, or electronic. Sometimes motor drive refers to a drive used to control a motor and therefore gets interchanged with VFD or VSD. The Motor driver is shown in fig 4.16

Fig 4.16 Motor driver

Electric Motors

AC electric motors can be run in fixed-speed operation determined by the number of stator pole pairs in the motor and the frequency of the alternating current supply. AC motors can be made for "pole changing" operation, reconnecting the stator winding to vary the number of poles so that two, sometimes three, speeds are obtained. For example, a machine with eight physical pairs of poles, could be connected to allow running with either four or eight pole pairs, giving two speeds - at 60 Hz, these would be 1800 RPM and 900 RPM. If speed changes are rare, the motor may be initially connected for one speed then re-wired for the other speed as process conditions change, or, magnetic contactors can be used to switch between the two speeds as process needs fluctuate. Connections for more than three speeds are uneconomic. The number of such fixed-speed-operation speeds is constrained by cost as number of pole pairs increases. If many different speeds or continuously variable speeds are required, other methods are required. Direct-current motors allow for changes of speed by adjusting the shunt field current. Another way of changing speed of a direct current motor is to change the voltage applied to the armature. An adjustable-speed motor drive might consist of an electric motor and controller that is used to adjust the motor's operating speed. The combination of a constant-speed motor and a continuously adjustable mechanical speed-changing device might also be called an "adjustable speed motor drive". Power electronics-based variable frequency drives are rapidly making older technologies redundant. Some prime movers (internal_combustion engines, reciprocating or turbine steam engines, water wheels, and others) have a range of operating speeds which can be varied continuously (by adjusting fuel rate or similar means). However, efficiency may be low at extremes of the speed range, and there may be system reasons why the prime mover speed cannot be maintained at very low or very high speeds. Before electric motors were invented, mechanical speed changers were used to control the mechanical power provided by water wheels and steam engines. When electric motors came into use, means of controlling their speed were developed almost immediately. Today, various types of mechanical drives, hydraulic drives and electric drives compete with one another in the industrial drives market.

Mechanical drives

There are two types of mechanical drives, variable-pitch drives, and traction drives.

Variable-pitch drives are pulley and belt drives in which the pitch diameter of one or both pulleys can be adjusted.

Traction drives transmit power through metal rollers running against mating metal rollers. The input-output speed ratio is adjusted by moving the rollers to change the diameters of the contact path. Many different roller shapes and mechanical designs have been used.

Hydraulic adjustable speed drives

There are three types of hydraulic drives, those are: hydrostatic drives, hydrodynamic drives and hydro viscous drives.

A hydrostatic drive consists of a hydraulic pump and a hydraulic motor. Since positive displacement pumps and motors are used, one revolution of the pump or motor corresponds to a set volume of fluid flow that is determined by the displacement regardless of speed or torque. Speed is regulated by regulating the fluid flow with a valve or by changing the displacement of the pump or motor. Many different design variations have been used. A swash plate drive employs an axial piston pump or motor in which the swash plate angle can be changed to adjust the displacement and thus adjust the speed.

Hydrodynamic drives or fluid couplings use oil to transmit torque between an impeller on the constant-speed input shaft and a rotor on the adjustable-speed output shaft. The torque converter in the automatic transmission of a car is a hydrodynamic drive.

A hydro viscous drive consists of one or more discs connected to an input shaft pressed against a similar disc or discs connected to an output shaft. Torque is transmitted from the input shaft to the output shaft through an oil film between the discs. The transmitted torque is proportional to the pressure exerted by a hydraulic cylinder that presses the discs together. This effect may be used as a clutch, such as the Hele-Shaw clutch, or as a variable-speed drive, such as the Beier variable-ratio gear.

Continuously variable transmission (CVT)

Mechanical and hydraulic adjustable speed drives are usually called "transmissions" or "continuously variable transmissions" when they are used in vehicles, farm equipment and some other types of equipment.

Electric adjustable speed drives

Types of control

Control can mean either manually adjustable - by means of a potentiometer or linear hall effect device, (which is more resistant to dust and grease) or it can also be automatically controlled, for example, by using a rotational detector such as a Gray code optical encoder.

Types of drives

There are three general categories of electric drives: DC motor drives, eddy current drives and AC motor drives. Each of these general types can be further divided into numerous variations. Electric drives generally include both an electric motor and a speed control unit or system. The term drive is often applied to the controller without the motor. In the early days of electric drive technology, electromechanical control systems were used. Later, electronic controllers were designed using various types of vacuum tubes. As suitable solid state electronic components became available, new controller designs incorporated the latest electronic technology.

DC drives

DC drives are DC motor speed control systems. Since the speed of a DC motor is directly proportional to armature voltage and inversely proportional to motor flux (which is a function of field current), either armature voltage or field current can be used to control speed.

Eddy current drives

An eddy current drive (sometimes called a "Dynamitic drive", after one of the most common brand names) consists of a fixed-speed motor (generally an induction motor) and an eddy current clutch. The clutch contains a fixed-speed rotor and an adjustable-speed rotor separated by a small air gap. A direct current in a field coil produces a magnetic field that determines the torque transmitted from the input rotor to the output rotor. The controller provides closed loop speed regulation by varying clutch current, only allowing the clutch to transmit enough torque to operate at the desired speed. Speed feedback is typically provided via an integral AC tachometer. Eddy current drives are slip-controlled systems the slip energy of which is necessarily all dissipated as heat. Such drives are therefore generally less efficient than AC/DC-AC conversion based drives. The motor develops the torque required by the load and operates at full speed. The output shaft transmits the same torque to the load, but turns at a slower speed. Since power is proportional to torque multiplied by speed, the input power is proportional to motor speed times operating torque while the output power is output speed times operating torque. The difference between the motor speed and the output speed is called the slip speed. Power proportional to the slip speed times operating torque is dissipated as heat in the clutch. While it has been surpassed by the variable-frequency drive in most variable-speed applications, the eddy current clutch is still often used to couple motors to high-inertia loads that are frequently stopped and started, such as stamping presses, conveyors, hoisting

machinery, and some larger machine tools, allowing gradual starting, with less maintenance than a mechanical clutch or hydraulic transmission.

AC drives

AC drives are AC motor speed control systems. A slip-controlled wound-rotor induction motor (WRIM) drive controls speed by varying motor slip via rotor slip rings either by electronically recovering slip power fed back to the stator bus or by varying the resistance of external resistors in the rotor circuit. Along with eddy current drives, resistance-based WRIM drives have lost popularity because they are less efficient than AC/DC-AC-based WRIM drives and are used only in special situations.

Slip energy recovery systems return energy to the WRIM's stator bus, converting slip energy and feeding it back to the stator supply. Such recovered energy would otherwise be wasted as heat in resistance-based WRIM drives. Slip energy recovery variable-speed drives are used in such applications as large pumps and fans, wind turbines, shipboard propulsion systems, large hydro-pumps and generators and utility energy storage flywheels. Early slip energy recovery systems using electromechanical components for AC/DC-AC conversion (i.e., consisting of rectifier, DC motor and AC generator) are termed Kramer drives, with more recent systems using variable-frequency drives (VFDs) being referred to as static Kramer drives.

In general, a VFD in its most basic configuration controls the speed of an induction or synchronous motor by adjusting the frequency of the power supplied to the motor. When changing VFD frequency in standard low-performance variable-torque applications using Volt-per-Hertz (V/Hz) control, the AC motor's voltage-to-frequency ratio can be maintained constant, and its power can be varied, between the minimum and maximum operating frequencies up to a base frequency.

Constant voltage operation above base frequency, and therefore with reduced V/Hz ratio, provides reduced torque and constant power capability. Regenerative AC drives are a type of AC drive which have the capacity to recover the braking energy of a load moving faster than the motor speed (an overhauling load) and return it to the power system.

Reasons for using adjustable speed drives

Process control and energy conservation are the two primary reasons for using an adjustable-speed drive. Historically, adjustable-speed drives were developed for process control, but

energy conservation has emerged as an equally important objective. The graph between frequency and Voltage power torque is shown in fig 4.17

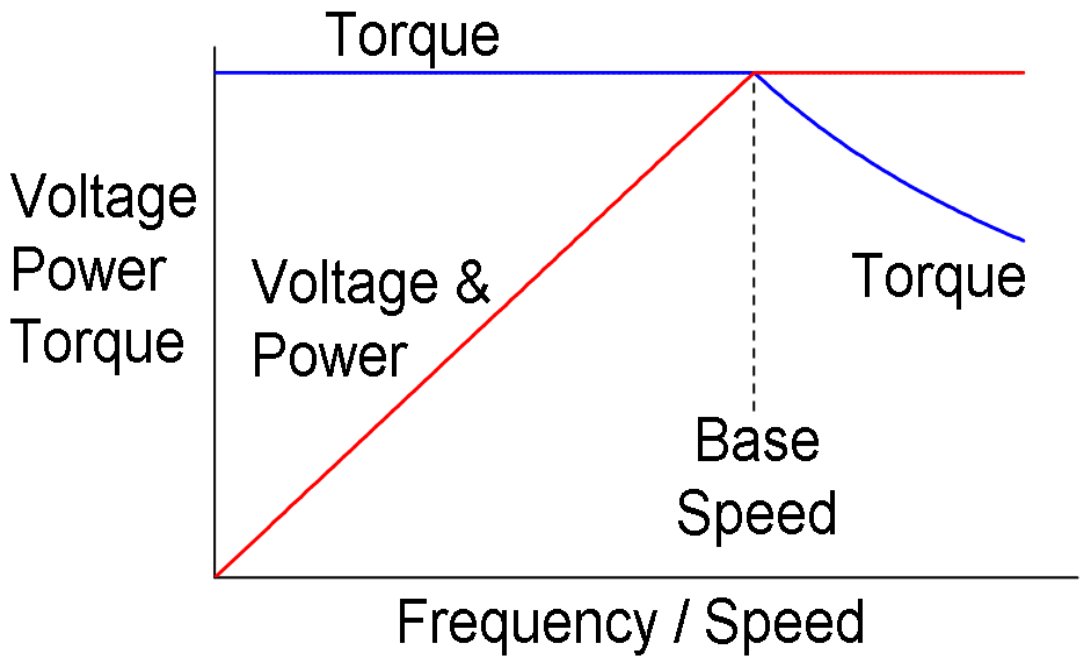


Fig 4.17 Graph between Frequency and Voltage Power torque

Saving energy by using efficient adjustable-speed drives

Fans and pumps consume a large part of the energy used by industrial electrical motors. Where fans and pumps serve a varying process load, a simple way to vary the delivered quantity of fluid is with a damper or valve in the outlet of the fan or pump, which by its increased pressure drop, reduces the flow in the process. However, this additional pressure drop represents energy loss. Sometimes it is economically practical to put in some device that recovers this otherwise lost energy. With a variable-speed drive on the pump or fan, the supply can be adjusted to match demand and no extra loss is introduced.

For example, when a fan is driven directly by a fixed-speed motor, the airflow is designed for the maximum demand of the system, and so will usually be higher than it needs to be. Airflow can be regulated using a damper but it is more efficient to directly regulate fan motor speed. Following the affinity laws, for 50% of the airflow, the variable-speed motor consumes about

20% of the input power (amps). The fixed-speed motor still consumes about 85% of the input power at half the flow.

Acceleration control

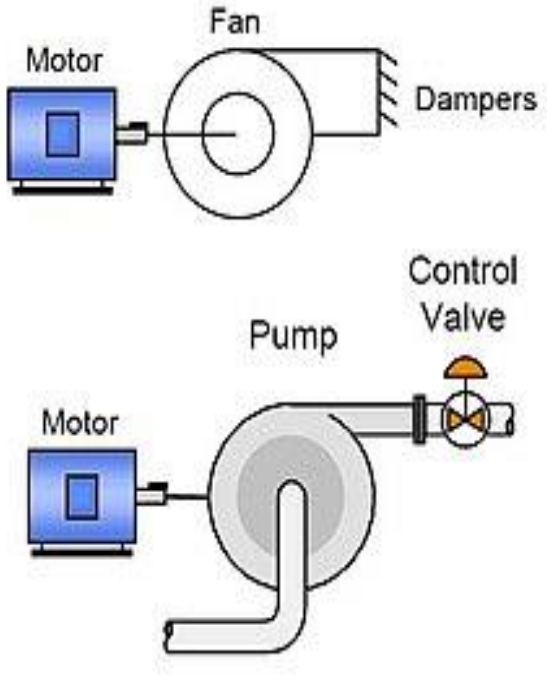


Fig 4.18 Acceleration control

An adjustable-speed drive can often provide smoother operation compared to an alternative fixed-speed mode of operation. Fig 4.18 shows acceleration control. For example, in a sewage lift station sewage usually flows through sewer pipes under the force of gravity to a wet well location. From there it is pumped up to a treatment process. When fixed-speed pumps are used, the pumps are set to start when the level of the liquid in the wet well reaches some high point and stop when the level has been reduced to a low point. Cycling the pumps on and off results in frequent high surges of electric current to start the motors that results in electromagnetic and thermal stresses in the motors and power control equipment, the pumps and pipes are subjected to mechanical and hydraulic stresses, and the sewage treatment process is forced to accommodate surges in the flow of sewage through the process. When adjustable speed drives are used, the pumps operate continuously at a speed that increases as the wet well level increases. This matches the outflow to the average inflow and provides a much smoother operation of the process. The L298N Motor Driver Module. is a high-power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control upto 4 DC motors, or 2 DC motors with directional and speed control. Fig 4.19 shows L298N motor driver module and its module point

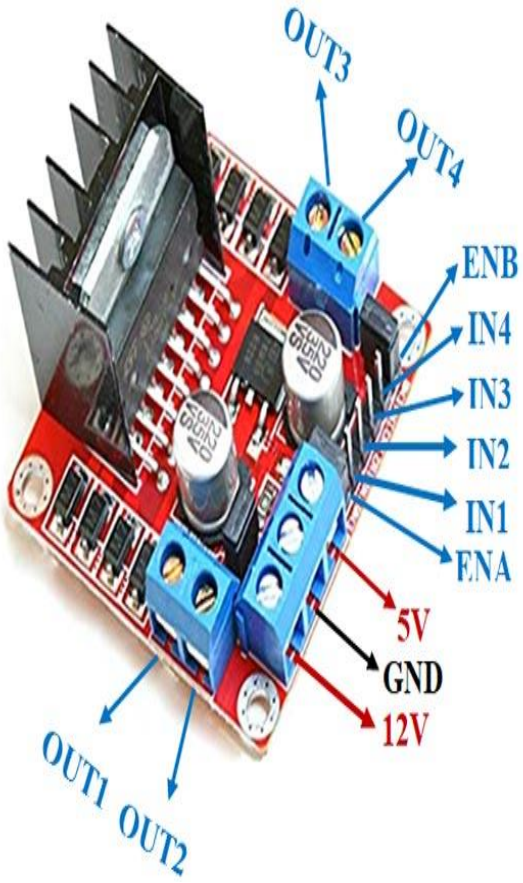


Fig 4.19 L298N Motor Driver Module Pinout

Features & Specifications

- Driver Model: L298N 2A
- Driver Chip: Double H Bridge L298N
- Motor Supply Voltage (Maximum): 46V
- Motor Supply Current (Maximum): 2A
- Logic Voltage: 5V
- Driver Voltage: 5-35V
- Driver Current:2A
- Logical Current:0-36mA
- Maximum Power (W): 25W
- Current Sense for each motor
- Heatsink for better performance
- Power-On LED indicator

L298N Module Pinout Configuration

Table 4.1 L298N Module Pinout Configuration

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

L298N Circuit Diagram

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output. This depends on the voltage used at the motors VCC. The module have an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V we must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly. The Enable A and Enable B pins are used for enabling and controlling the speed of the motor. If a jumper is present on this pin, the motor will be enabled and work at maximum speed, and if we remove the jumper, we can connect a PWM input to this pin and in that way control the speed of the motor. If we

connect this pin to a Ground the motor will be disabled. Fig 4.20 shows connection of L298N circuit module.

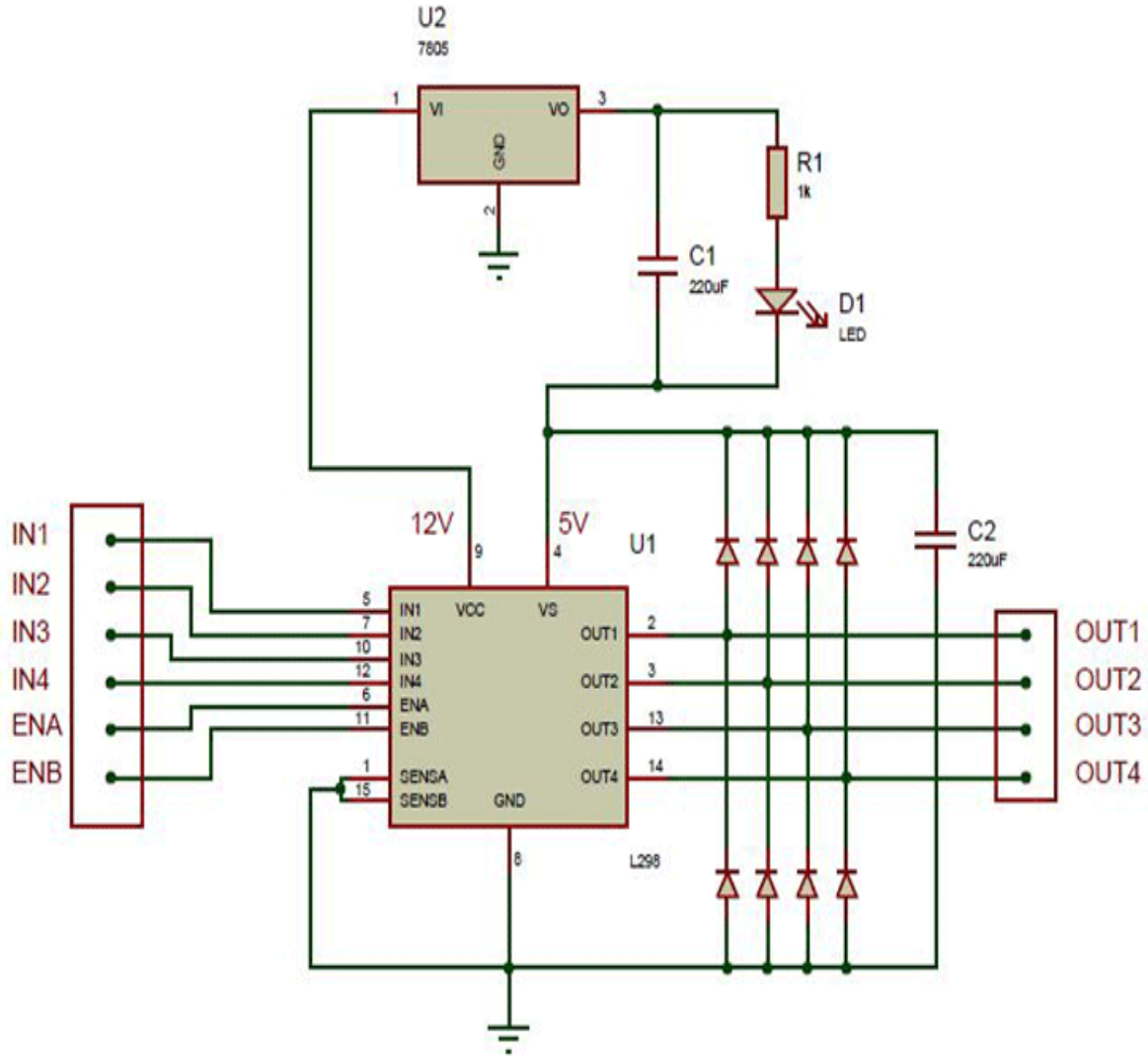


Fig 4.20 L298N Circuit Connection

4.5 BUZZER

There are many ways to communicate between the user and a product. One of the best ways is audio communication using a buzzer IC. So, during the design process, understanding some technologies with configurations is very helpful. So, this article discusses an overview of an audio signalling device like a beeper or a buzzer and its working with applications. An audio signalling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren. Fig 4.21 shows buzzer pin configuration.

Fig 4.21 Buzzer Pin Configuration

The pin configuration of the buzzer is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '- 'symbol or short terminal and it is connected to the GND terminal.

History

There are two main types of Buzzers, they are:

1. Electromechanical Buzzer
2. Piezoelectric Buzzer

Electromechanical

This buzzer was launched in the year 1831 by an American Scientist namely Joseph Henry but, this was used in doorbells until they were eliminated in 1930 in support of musical bells, which had a smooth tone.

Piezoelectric

These buzzers were invented by manufacturers of Japanese & fixed into a broad range of devices during the period of 1970s – 1980s. So, this development primarily came due to

cooperative efforts through the manufacturing companies of Japanese. In the year 1951, they recognized the Application Research Committee of Barium Titanate that allows the corporations to be cooperative competitively & bring about numerous piezoelectric creations.

Specifications

The specifications of the buzzer include the following.

- Colour is black
- The frequency range is 3,300Hz
- Operating Temperature ranges from – 20° C to +60°C
- Operating voltage ranges from 3V to 24V DC
- The sound pressure level is 85dBA or 10cm
- The supply current is below 15mA

Types of Buzzers

A buzzer is available in different types which include the following.

- Piezoelectric
- Electromagnetic
- Mechanical
- Electromechanical
- Magnetic

Piezoelectric

As the name suggests, the piezoelectric type uses the piezoelectric ceramic's piezoelectric effect & pulse current to make the metal plate vibrate & generate sound. This kind of buzzer is made with a resonance box, multi resonator, piezoelectric plate, housing, impedance matcher, etc. Some of the buzzers are also designed with LEDs. The multi resonator of this mainly includes ICs and transistors. Once the supply is given to this resonator, it will oscillate and generates an audio signal with 1.5 to 2kHz. The impedance matcher will force the piezoelectric plate to produce sound.

Electromagnetic

This type of buzzer is made with a magnet, solenoid coil, oscillator, housing, vibration diaphragm, and magnet. Once the power supply is given, the oscillator which produces the audio signal current will supply throughout the solenoid coil to generate a magnetic field. Sometimes, the vibration diaphragm will vibrate & generates sound under the magnet & solenoid coil interaction. The frequency range of this ranges from 2 kHz to 4kHz. Fig 4.22 shows Electromagnetic buzzer.

Fig 4.22 Electromagnetic Buzzer

Mechanical

These types of buzzers are subtypes of electromagnetic, so the components used in this type are also similar. But the main difference is that the vibrating buzzer is placed on the outside instead of the inside.

Electromechanical

The designing of these types of buzzers can be done with a bare metal disc & an electromagnet. The working principle of this is similar to magnetic and electromagnetic. It generates sound throughout the disc movement & magnetism.

Magnetic

Like a piezo type, magnetic is also used to generate a sound but they are different due to core functionality. The magnetic type is more fixed as compared to the piezo type because they

work through a magnetic field. Magnetic buzzers utilize an electric charge instead of depending on piezo materials to generate a magnetic field.

Working Principle

The working principle of a buzzer depends on the theory that, once the voltage is given across a piezoelectric material, then a pressure difference is produced. A piezo type includes piezo crystals among two conductors.

Once a potential disparity is given across these crystals, then they thrust one conductor & drag the additional conductor through their internal property. So, this continuous action will produce a sharp sound signal.

Mounting Configurations

The mounting configurations of buzzers include the following.

- Panel Mount
- Wire Leads
- Screw Terminals
- Through Hole
- Spring Contact
- Surface Mount

How to use a Buzzer?

A buzzer is an efficient component to include the features of sound in our system or project. It is an extremely small & solid two-pin device thus it can be simply utilized on breadboard or PCB. So in most applications, this component is widely used.

There are two kinds of buzzers commonly available like simple and readymade. Once a simple type is power-driven then it will generate a beep sound continuously. A readymade type looks heavier & generates a Beep. Beep. Beep. This sound is because of the internal oscillating circuit within it.

This buzzer uses a DC power supply that ranges from 4V – 9V. To operate this, a 9V battery is used but it is suggested to utilize a regulated +5V/+6V DC supply. Generally, it is connected through a switching circuit to switch ON/OFF the buzzer at the necessary time interval.

Buzzer Circuit Diagram

The circuit diagram of the water level indicator using the buzzer is shown below. This circuit is used to sense or detect the water level within the tank or washing machine or pool, etc. This circuit is very simple to design using few components such as a transistor, buzzer, 300K

variable resistor, and power supply or 9V battery. The internal circuit of buzzer is shown in fig 4.23

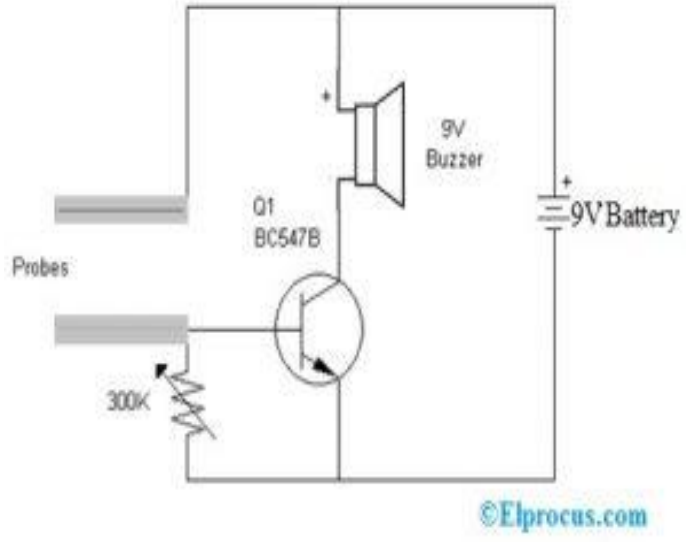


Fig 4.23 Internal circuit of buzzer

Water Level Circuit using Buzzer

Once the two probes of the circuit are placed in the tank, it detects the level of water. Once the water level exceeds the fixed level, then it generates a beep sound through a buzzer connected to the circuit. This circuit uses a BC547B NPN transistor however we can also use any general-purpose transistor instead of using 2N3904/2N2222.

This water level sensor circuit working is very simple and the transistor used within the circuit works as a switch. Once the two probes notice the water level within the tank, then the transistor turns ON & the voltage begins flowing throughout the transistor to trigger the buzzer.

How to Choose a Buzzer?

While choosing a buzzer or speaker, many principles need to consider like the following.

- Size of the product
- Consumption of Current
- Type of terminal
- Frequency Voltage
- Volume
- Type
- AC/DC Voltage
- The tone is Continuous/Pulsed
- Fixing – Pins, Leads/Surface Mount

- Output of Sound
- Feedback Option
- Piezo Elements

Advantages

The advantages of a buzzer include the following.

- Simply Compatible
- Frequency Response is Good
- Size is small
- Energy Consumption is less
- The Range of Voltage usage is Large
- Sound Pressure is high

Disadvantages

The disadvantages of the buzzer include the following.

- Controlling is a little hard
- Generates Annoying Sound
- Training is necessary to know how to repair the condition without just turning off.

Applications

The applications of the buzzer include the following.

- Communication Devices
- Electronics used in Automobiles
- Alarm Circuits
- Portable Devices
- Security Systems
- Timers
- Household Appliances
- Electronic Metronomes
- Sporting Events
- Annunciator Panels
- Game Shows

Thus, this is all about an overview of a buzzer data sheet that includes its working principle, pin configuration, specifications, circuit, advantages, disadvantages & its applications. It is an electromechanical, electromagnetic, mechanical, piezoelectric, electro-acoustic audio

signalling device. This buzzer works through an audio signal source or oscillating circuit. A ring or beep or click indicates that a switch has been pushed.

4.6 PUSH BUTTON

A pushbutton is a type of switch. It has two separate conductive pins that prevent a complete circuit by being separate from each other. When you press on a pushbutton, you're actually pushing the two pins together, completing the circuit. But if you let go, there's a spring-like mechanism that separates the pins again.

How to Use Pushbuttons

- Open your choice of code editor and paste the following code:

```
import Rpi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.IN, pull up down=GPIO.PUD_DOWN)
while True:
        if GPIO.input(7) ==GPIO.HIGH:
                print ("Pin 7 is High")
        elif GPIO.input(7) ==GPIO.LOW:
                print ("Pin 7 is LOW")
        sleep (0.15)
```

- Save as "rpi-pushbutton.py" (or any name you want as long as the file extension is the same)
- Build the circuit. On one pin of the pushbutton, wire it up to pin 7 and a resistor in parallel. Attach a jumper wire to a GND pin (pins 6, 7, 14, 20, 25, 30, 34, or 39) at the other side of this resistor, then attach another jumper wire to a 3.3V pin (pins 1 or 17) on a separate pushbutton pin
- In fig 4.24 left circuit is schematic diagram, right is live pushbutton on breadboard. Jumper wire colour designation: red =3.3V, brown= pin7 and black = GND.
- To find the right pin number, hold Raspberry Pi in a way that the GPIO pins sit in the upper-right corner. The top-left pin is pin 1, and to the right of it is pin2. Below pin1 is pin3 to the right is pin4 and so on.

- Power up your Raspberry Pi and open the terminal. Use cd to move to the Python script's directory, then enter python3 rpi-pushbutton.py. If you use a different filename, use that instead of "rpi-pushbutton".
- Fig 4.25 shows about pinout of raspberry pi

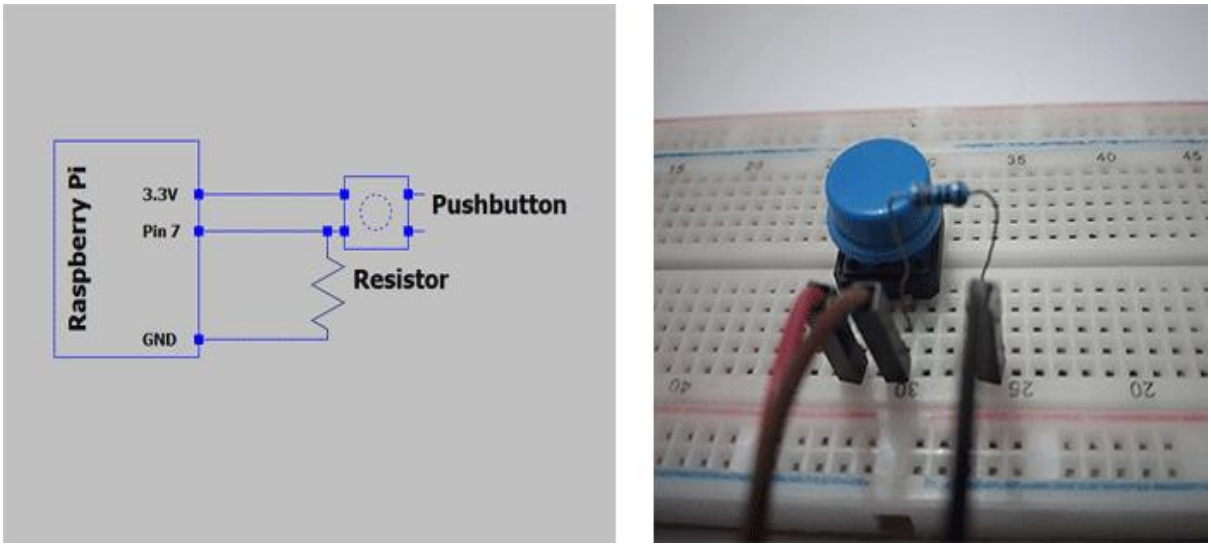


Fig 4.24 Circuit and connection of pushbutton

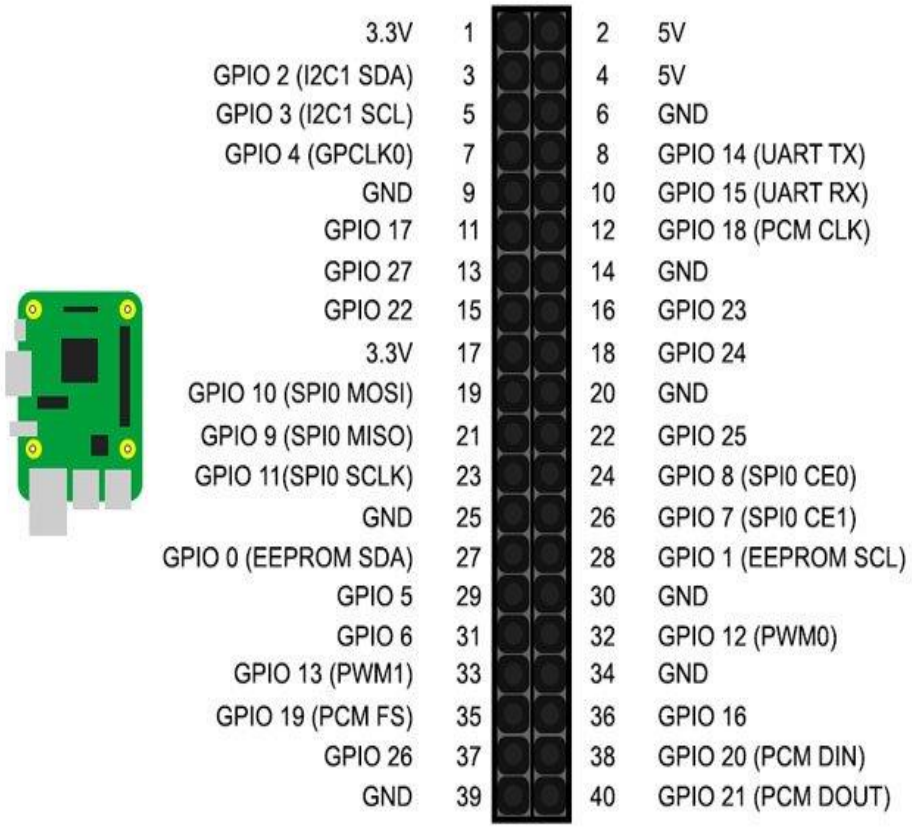


Fig 4.25 Pinout of the Raspberry Pi

- You should see a new line of text saying Pin 7 is low every 0.15 seconds on the terminal. If you press the button, the new line will be Pin 7 is high.

Hardware on Pushbuttons

Push buttons use two kinds of resistors: pull up and pull down. The one with 3.3V connected to the resistor is a pull-up resistor. It pulls the voltage upward. Meanwhile, pull-down resistors pull voltage down by having a GND pin connected to them

You can still use a pushbutton without a resistor, but doing that leaves your GPIO pin on float. A floating GPIO pin receives no direct electric charge, so it looks for charges over its surroundings. If there is a strong electromagnetic field near it, for example, it will just measure that instead. Fig 4.26 shows pin7 becomes a floating pin when 3.3V is separated from it.

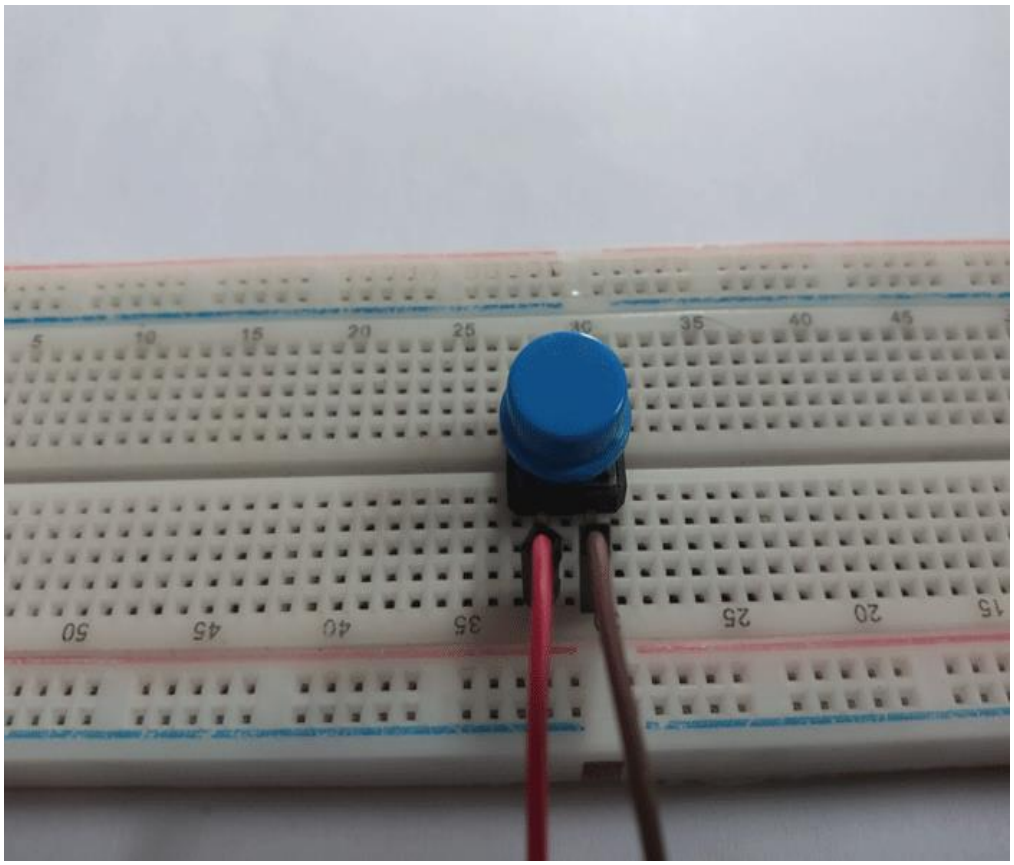


Fig 4.26 Pin 7 (brown) becomes a floating pin when 3.3V (red) is separated from it.

That is why you need a reference point. If you hook the GPIO pin to 0V (GND) by default then it will measure 0V while the button is unpressed. But if you do not, the GPIO pin's value can be anywhere – even negative volts. If we hook the GPIO pin to 0v by default then it will measure 0V while the button is unpressed. But if you do not, the GPIO pins value can be anywhere even negative volt. Floating pins can do something if we leave a pin floating it can sense the voltage difference in the air, measuring even the effect of having finger move near the pin itself. It is like an electromagnetic presence sensor or something.

Ways to Add a Button to Raspberry PI

There are two ways to add Raspberry Pi

- The RPi. GPIO Library
- GPIO Zero Library

Method 1: The RPi. GPIO Library

Once the Pi has booted, head to the menu and select Programming > Thonny Python IDE. A new Python script will open. If you are totally new to Python, it's a great language for beginners and there are many great places to learn more about Python after you are done with this tutorial!

Fig 4.27 shows interface of RPI.GPIO library.

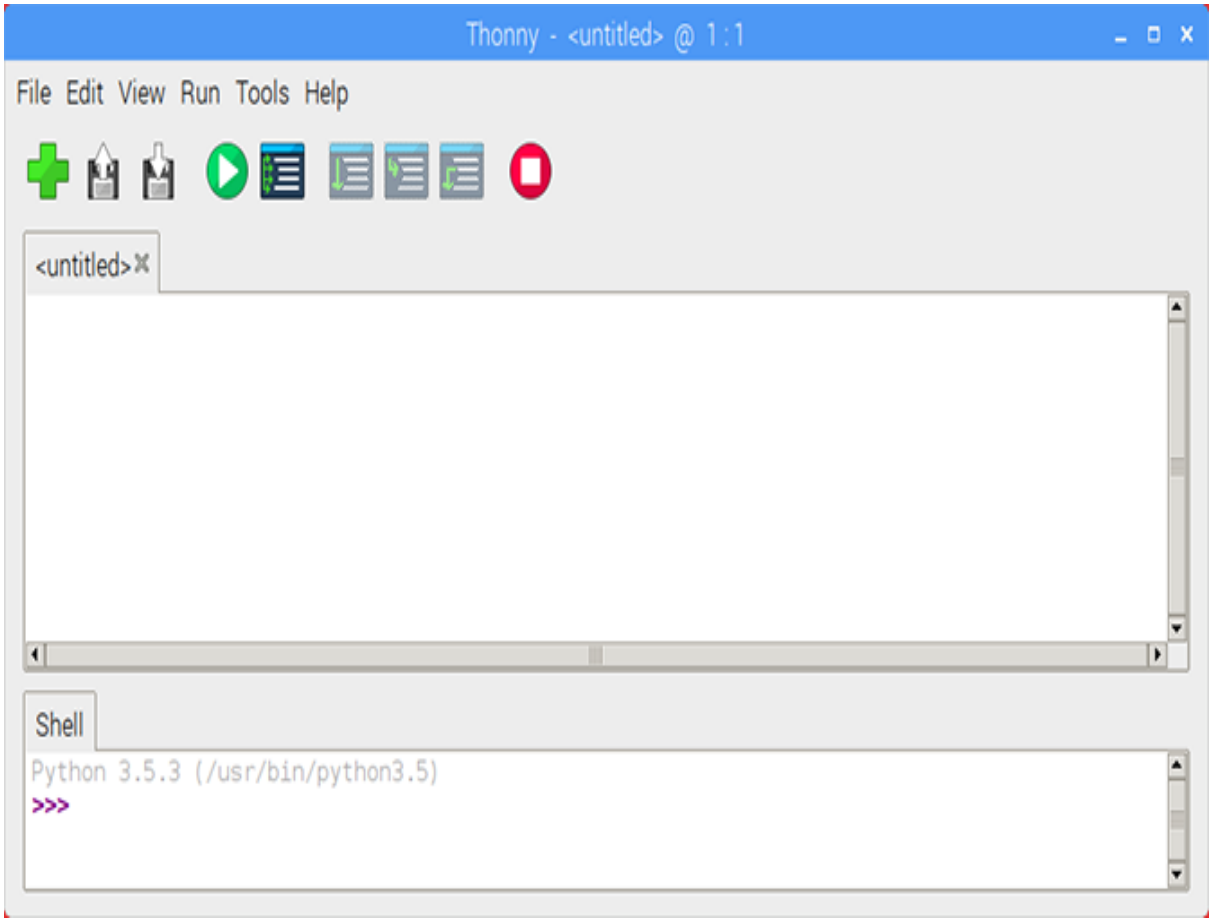


Fig 4.27 Interface of RPI.GPIO library

Start by importing the Rpi.GPIO library, and setting the board mode.

Import RPI.GPIO as GPIO

GPIO.setmode (GPIO.BOARD)

Now declare the variables for the LED and button pin numbers.

Led pin=12

Led pin=16

Note that since we have the board mode set to BOARD, we are using the pin numbers rather than the GPIO numbers. If that is confusing to you, a Raspberry Pi pinout chart can help clear it up.

It is time to set up the GPIO pins. Set the LED pin to output, and the button pin to input with a pull-up resistor

GPIO.setup(Led pin GPIO.OUT)

The text after GPIO.IN refers to the internal pull-up resistor of the Raspberry Pi. You need to enable this to get a clean reading from the button. Since the button is going to the ground pin, we need a pull-up resistor to hold the input pin HIGH until you press it. Before we go on, let's look at pull-up and pull-down resistors.

Intermission: Pull Up/Pull Down Resistors

When you configure a GPIO pin to input, it reads that pin to determine its state. In this circuit, you need to read whether a pin is HIGH or LOW to trigger the LED when the button is pressed. This would be simple if those were the only states a pin can have, but unfortunately, there is a third state: FLOATING.

A floating pin has a value between high and low, causing the input to act unpredictably. Pull-up/pull-down resistors solve this. Fig 4.28 shows example of Pull Up resistor

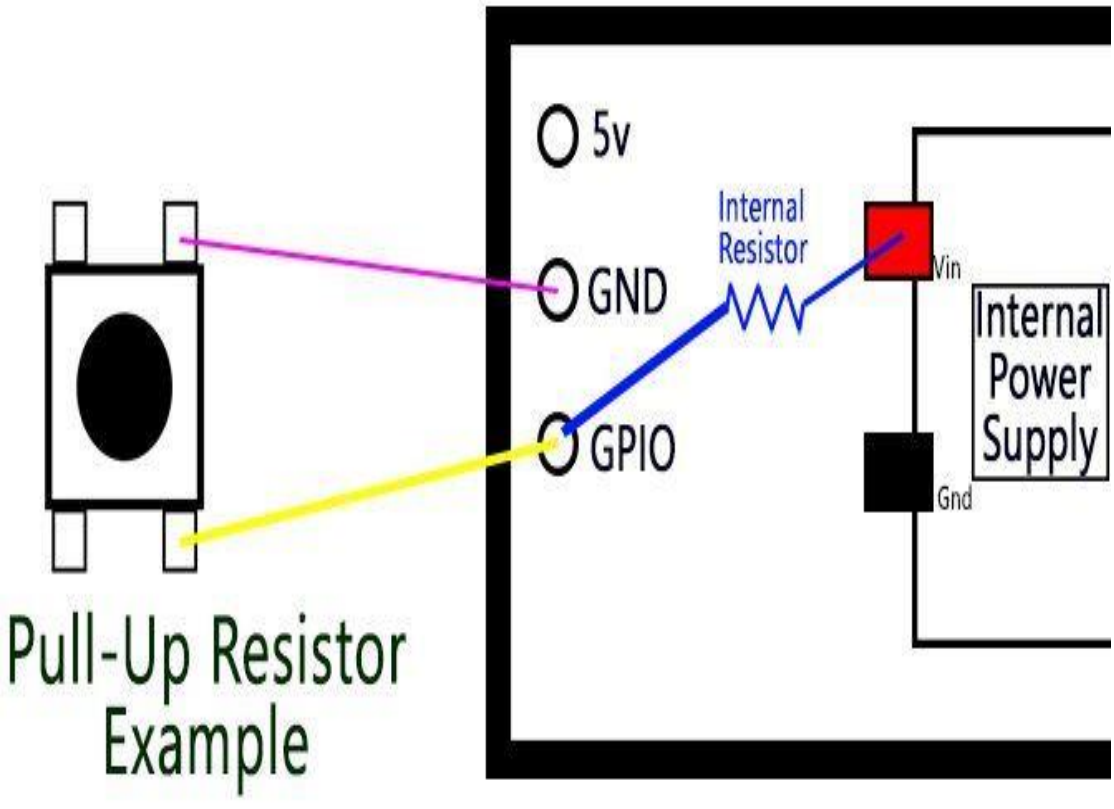


Fig 4.28 Example of Pull up resistor

The above image is a simplified diagram of a button and a Raspberry Pi. The GPIO pin connects to ground through the button. The internal pull-up resistor attaches the GPIO pin to the internal Pi power supply. These current flows and the pin are safely pulled up to HIGH.

When you press the button, the GPIO pin connects directly to the ground pin, and the button reads low. Fig 4.29 shows example of Pull-down resistor.

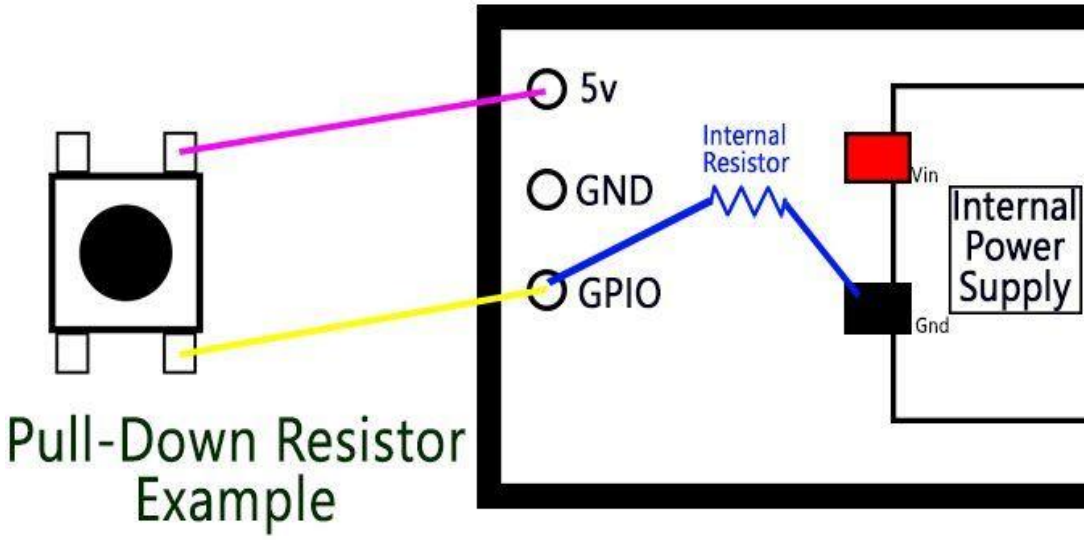


Fig 4.29 Example of Pull-down resistor

Pull-down resistors are for when the switch is connected to the power pin. This time, the internal resistor attaches the GPIO pin to ground, holding in LOW until you press the button.

Pull-up and Pull-down resistor theory is confusing at first glance, but important knowledge to have when working with microcontrollers.

The Program Loop

Next, set up the program loop:

- while True:

  button state= GPIO.input(button pin)
- if button state =false:

  GPIO.output(Led Pin, GPIO.High)
- else:

  GPIO.output(Led Pin, GPIO.LOW)

The while True loop continually runs the code inside it until we end the program. Every time it loops it updates the button State by reading the input from the button Pin. While the button is not being pressed, it stays HIGH. Once the button is pressed, buttonState becomes LOW. This triggers the if statement, since False is the same as LOW, and the LED turns on. The else statement turns the LED off whenever the buttonPin is not False.

Save and Run Your Script

Save your script by clicking File > Save As and choosing a file name. You can run the sketch by clicking the green Play button in the Thonny toolbar. Now press the button, and your LED should light up! Press the red Stop button at any time to stop the program. Fig 4.30 shows push button connection and configuration

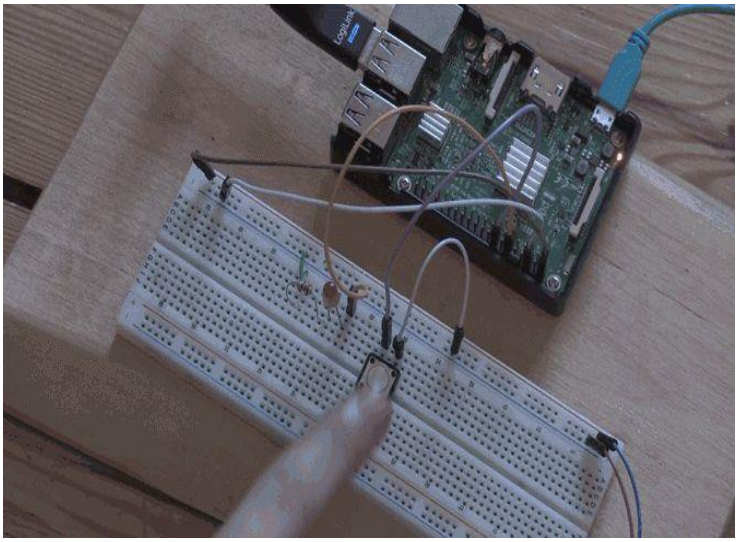


Fig 4.30 shows push button connection and configuration

Method 2: GPIO Zero Library

The RPi.GPIO library is fantastic, but there is a new kid on the block. The GPIO Zero Library was created by Raspberry Pi community manager Ben Nuttall with the intention of making code simpler, and easier to read and write.

You will notice you did not import the whole library. Since you are only using an LED and button, you require only those modules in the script. We also import Pause from the signal library, which is a Python library for event management.

Setting up the pins is much easier with GPIO Zero:

- Led = LED Button (18)

Since the GPIO Zero library has modules for the LED and button, you do not need to set up inputs and outputs like before. You will notice that though the pins haven't changed, the numbers here are different from above. That is because GPIO Zero only uses the GPIO pin numbers (also known as Broadcom or BCM numbers).

The rest of the script is only three lines:

- Button, when_pressed = LED.on
- Button, when_released=LED.off
- pause ()

The pause () call here simply stops the script from exiting when it reaches the bottom. The two-button events get triggered whenever the button is pressed and released. Save and run your script and the same result will be obtained.

Code for Pushbuttons

Knowing that, you should understand that pin 7 senses whether 3.3V or 0V passes through it. If it senses 3.3V, then it reports itself as HIGH. But if it senses 0V, then it is LOW. Let us divide the code into three parts: import commands, setup commands, and looped commands.

Import Commands

We are using two import commands:

- import RPI.GPIO as GPIO
- from time import sleep

import RPI.GPIO as GPIO, imports the RPi.GPIO module, which lets you do stuff with your Raspberry Pi's GPIO pins. By adding in GPIO at the end, you are telling Python to say that typing GPIO is equivalent to typing RPI.GPIO. You can even replace it with other strings, and the code should still work if you format it properly.

On the other hand, from time import sleep, imports only a part of Python's time module. It lets you use the sleep () function.

Setup Commands

We are working with the three commands from the RPi.GPIO module on the setup commands to fix some settings.

- GPIO.setwarnings(False)
- GPIO.setmode(GPIO.BOARD)
- GPIO.setup(7, GPIO.IN, pull up down= GPIO.PUD_DOWN)

The RPi.GPIO module normally shows a message that warns you about using the GPIO pins as soon as you start up the Python script. GPIO.setwarnings(False), prevents that from happening. GPIO.setmode(GPIO.BOARD) is another command from the RPi.GPIO module. It tells Python that you are using the "BOARD" pinout. There are two kinds of pinout in RPi.GPIO: BOARD and BCM. BOARD lets you pick pins by using the pin numbers. BCM (short for "Broadcom") lets you pick pins with their individual Broadcom SOC Channel. BOARD is much easier to use, as it's always the same no matter what kind of Raspberry Pi board you use. The BCM pinout can change depending on which model you use. Lastly, GPIO.setup(7, GPIO.IN, pull up down= GPIO.PUD_DOWN), lets you set pin 7 as an input pin. It uses the setup function and reads 7 as the pin you are trying to choose. GPIO.IN means

you are trying to set that as an input pin. Embedded systems normally just use a few lines of code and loop them indefinitely. Different programming languages use different ways to do it. But the concept is the same: they use a loop.

- while True:
  if GPIO.input(7) == GPIO.HIGH:
    print (“Pin 7 is HIGH”)
  elif GPIO.input(7) == GPIO.LOW:
    print (“Pin 7 is LOW”)
  sleep (0.15)

while True, lets loop code indefinitely. Everything you place in it will run forever as long as there’s electricity on the board.

if GPIO.input(7) == GPIO.HIGH, is an if statement. It says that if pin 7, which is an input pin, reads as HIGH, then it should do everything inside it.

print (“Pin 7 is HIGH”) is inside an if statement. All it does is print out Pin 7 is HIGH on the console. We can replace that with any string, number, or variable that contains those.

Next is elif GPIO.input(7) == GPIO.LOW: It’s basically the same as if GPIO.input(7) == GPIO.HIGH except for the first part: it uses elif instead of it. The code elif stands for Else If. What it says is that if all the other code above it returns false, then Python should run this else-if statement.

Lastly, sleep (0.15) pauses the code for 0.15 seconds. It is mostly for performance issues. The Raspberry Pi will send output code so fast that it is going to make your GUI lag a bit.

CHAPTER 5
SOFTWARE COMPONENTS

5.1 PYTHON IDE

In the world of software development, Python has emerged as a dominant force, owing to its simplicity, versatility, and extensive ecosystem of libraries and frameworks. However, the effectiveness and efficiency of Python development depend significantly on the tools available to developers. Among these tools, Integrated Development Environments (IDEs) play a pivotal role in shaping the coding experience, facilitating tasks ranging from code writing to debugging and deployment. Python Integrated Development Environments (IDEs) have revolutionized the way developers interact with the Python programming language, providing comprehensive platforms tailored to streamline the development process. From code writing to debugging and deployment, IDEs offer a plethora of features aimed at enhancing productivity and code quality. In this comprehensive introduction, we delve into the significance, evolution, and key features of Python IDEs, shedding light on their indispensable role in the programming landscape. This comprehensive exploration aims to delve deep into the realm of Python IDEs, shedding light on their evolution, features, significance, and the diverse array of options available to developers. From the early days of rudimentary text editors to the sophisticated, feature-rich IDEs of today, the journey of Python development tools reflects the evolution of the language itself and the ever-changing needs of developers.

5.1.1 EVOLUTION OF PYTHON IDE

The evolution of Python IDEs can be traced back to the nascent stages of the language's development. In the early days, developers relied on basic text editors such as Emacs, Vim, and Notepad for writing Python code. These editors offered minimal features, lacking even syntax highlighting or code completion. However, as Python gained popularity and matured as a language, the demand for more sophisticated development environments grew. The first generation of Python IDEs emerged in the late 1990s and early 2000s, offering features like syntax highlighting, code folding, and rudimentary debugging capabilities. Tools like IDLE (Integrated Development and Learning Environment), which shipped with the standard Python distribution, provided a simple yet functional environment for beginners and casual developers. As Python continued to gain traction in various domains, including web development, scientific computing, and data analysis, the need for more powerful IDEs

became apparent. This led to the rise of second-generation IDEs, such as PyCharm, which offered advanced features like intelligent code completion, integrated version control, and support for web frameworks like Django and Flask. These IDEs were designed to cater to the diverse needs of professional developers and teams working on complex projects.

5.1.2 SIGNIFICANCE OF PYTHON IDE

Python Integrated Development Environments (IDEs) stand as pillars in the software development landscape, offering a comprehensive suite of tools and features that significantly enhance the productivity, efficiency, and quality of Python programming. In this discourse, we delve into the profound significance of Python IDEs, elucidating their pivotal role in empowering developers, fostering collaboration, and catalysing innovation across various domains.

1. **Empowering Developers**

   Python IDEs empower developers by providing a unified environment that streamlines the entire development workflow, from code creation to deployment. By consolidating essential tools such as code editors, debuggers, and version control systems into a single platform, IDEs eliminate the need for developers to juggle multiple applications, thus minimizing context switching and cognitive overload. This seamless integration fosters a conducive environment for focused, uninterrupted coding sessions, enabling developers to fully unleash their creativity and productivity.

Features Empowering Developers

- **Code Editor**: Python IDEs offer advanced code editing features such as syntax highlighting, code completion, and automatic indentation, which facilitate rapid code creation and ensure syntactic correctness.
- **Debugging Tools:** IDEs provide robust debugging tools, including breakpoints, watch expressions, and interactive debug consoles, enabling developers to identify and rectify errors efficiently, thereby enhancing code quality and reliability.
- **Version Control Integration:** IDEs seamlessly integrate with version control systems like Git, allowing developers to manage code changes, collaborate with team members, and track project history within the familiar IDE environment.

2. Foresting Collaboration

Python IDEs serve as catalysts for collaboration, enabling developers to work cohesively in both local and distributed environments. By facilitating seamless communication, code sharing, and project management capabilities, IDEs empower teams to collaborate effectively, iterate rapidly, and deliver high-quality software products within tight deadlines. Moreover, the integration of collaborative tools and plugins within IDEs fosters a culture of knowledge sharing, mentorship, and continuous learning among team members, thereby fostering a dynamic and innovative development ecosystem.

Collaborative Features

- **Project management:** IDEs provide project management features such as project structuring, task tracking, and milestone management, which facilitate coordination and alignment among team members working on complex software projects.
- **Real time collaboration:** Some IDEs offer real-time collaboration features, allowing multiple developers to edit code simultaneously, review changes, and provide feedback in real-time, thereby promoting synergy and collective ownership of codebases

3.Catalyzing information

Python IDEs play a pivotal role in catalysing innovation by providing developers with access to cutting-edge tools, libraries, and frameworks that enable them to push the boundaries of what's possible in software development. By fostering an ecosystem of innovation and experimentation, IDEs empower developers to explore new ideas, prototype solutions, and iterate rapidly, thereby driving continuous improvement and evolution within the Python community. Moreover, the extensibility of IDEs through plugins and customizations empowers developers to tailor their development environment to suit their unique preferences, workflows, and project requirements, thereby unleashing their full creative potential.

Innovation Catalyst

- **Access to Libraries and Frameworks:** IDEs provide seamless integration with popular Python libraries and frameworks, such as TensorFlow for machine learning, Django for web development, and Pandas for data analysis, enabling

developers to leverage state-of-the-art tools and technologies to solve complex problems effectively.

- **Community and Ecosystem:** IDEs serve as hubs for the Python community, facilitating knowledge sharing, collaboration, and mentorship among developers worldwide. By providing access to community forums, tutorials, and educational resources, IDEs empower developers to learn from each other, exchange ideas, and collectively push the boundaries of Python programming.

5.1.3 POPULAR PYTHON IDE

- **PyCharm:** PyCharm, developed by JetBrains, stands as one of the most prominent and powerful Integrated Development Environments (IDEs) for Python programming. With its extensive feature set, intuitive user interface, and robust performance, PyCharm has garnered widespread acclaim and adoption among developers worldwide. In this comprehensive exploration, we delve deep into the myriad facets of PyCharm, elucidating its key features, advantages, use cases, and the profound impact it has had on Python development.

  #### Key Features of PyCharm

  1. **Intelligent Code editor:** PyCharm boasts an intelligent code editor that provides advanced features such as syntax highlighting, code completion, and code analysis. The code editor offers context-aware suggestions, helping developers write code faster and with fewer errors. Additionally, PyCharm's code analysis engine detects potential issues in the codebase, such as syntax errors, unresolved references, and code smells, allowing developers to address them proactively.
  2. **Powerful debugging tools:** Debugging is a critical aspect of software development, and PyCharm offers powerful debugging tools that facilitate the identification and resolution of errors in Python code. With features such as breakpoints, watch expressions, and interactive debug consoles, developers can inspect the state of their program at runtime, trace execution flow, and diagnose complex issues with ease. PyCharm's debugger integrates seamlessly with popular Python frameworks, enabling developers to debug web applications, server-side scripts, and other types of Python projects effortlessly.
  3. **Seamless Project Management:** PyCharm provides robust project management capabilities, allowing developers to organize their code into projects,

manage dependencies, and collaborate with team members effectively. The IDE supports various project types, including standalone Python scripts, web applications, and data science projects, and offers features such as version control integration, task tracking, and project templates to streamline the development process.

4. **Intelligent Code Navigation:** Navigating large codebases can be challenging, but PyCharm simplifies this task with its intelligent code navigation features. Developers can quickly jump to declarations, usages, and implementations of symbols within their codebase, making it easier to understand code structure and relationships. PyCharm also provides context-sensitive navigation options, allowing developers to explore code hierarchies, search for specific symbols, and navigate through files and directories effortlessly. The interface of PyCharm is shown in below fig 5.1

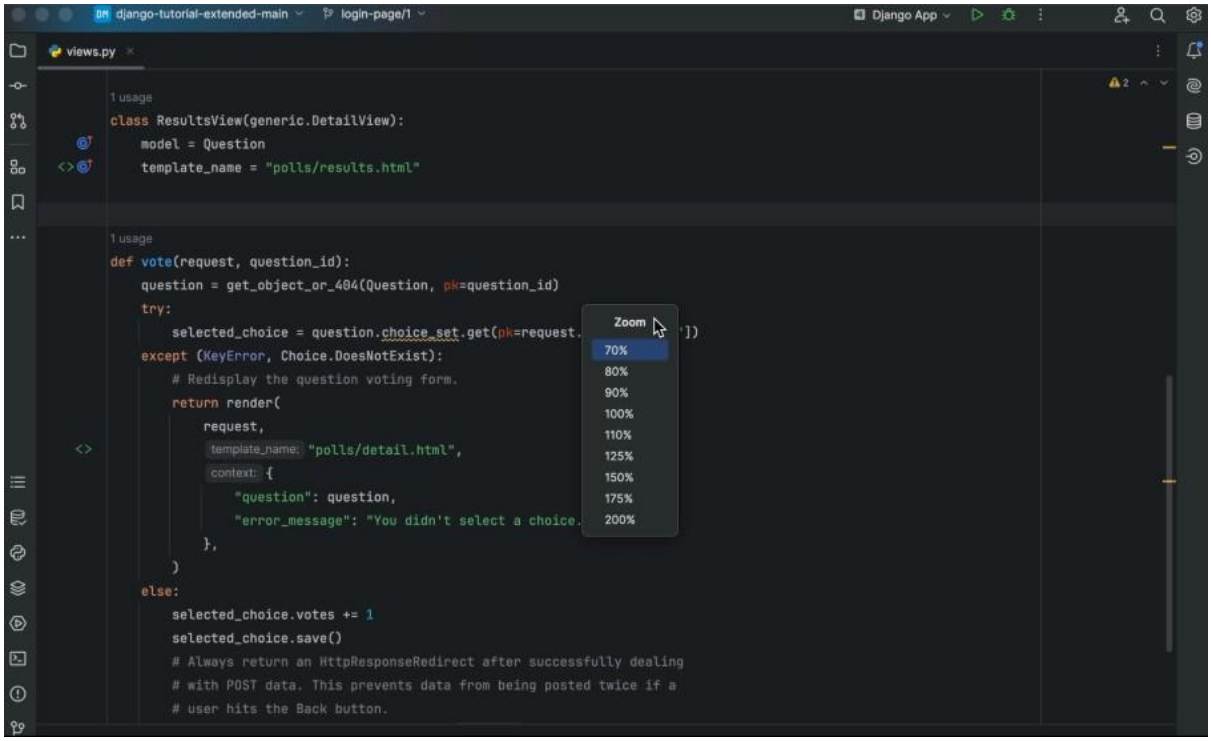


Fig 5.1 Interface of PyCharm

- **Visual Studio Code (Vs Code):** Visual Studio Code (VS Code), developed by Microsoft, stands as a paradigm shift in the landscape of code editors and Integrated Development Environments (IDEs). With its lightweight yet powerful architecture, extensive customization options, and vast ecosystem of extensions, VS Code has

garnered widespread adoption and acclaim among developers worldwide. In this comprehensive exploration, we delve into the myriad facets of Visual Studio Code, elucidating its key features, advantages, use cases, and the transformative impact it has had on the world of code development.

Keys Features of Visual Studio Code

1. **Lightweight and Fast:** Visual Studio Code is renowned for its lightweight and fast performance, making it ideal for developers working on projects of any size or complexity. Unlike traditional IDEs that may consume significant system resources and slow down over time, Visual Studio Code delivers a snappy and responsive editing experience, enabling developers to focus on writing code without distractions or delays.
2. **Extensibility and Customization:** One of the defining features of Visual Studio Code is its extensibility and customization options. The editor comes with built-in support for a wide range of programming languages and frameworks, but its true power lies in its vast ecosystem of extensions. Developers can choose from thousands of extensions available in the Visual Studio Code Marketplace, which provide additional features, tools, and language support tailored to their specific needs and preferences
3. **IntelliSense and Code Navigation:** Visual Studio Code offers IntelliSense, a powerful code completion feature that provides context-aware suggestions as you type, helping you write code faster and with fewer errors. Additionally, the editor includes robust code navigation capabilities, allowing you to quickly jump to definitions, find all references, and navigate through your codebase with ease, thereby enhancing productivity and code comprehension.
4. **Integrated terminal and Debugging:** Visual Studio Code includes an integrated terminal that allows you to run commands, execute scripts, and interact with your development environment without leaving the editor. Moreover, the editor provides seamless debugging capabilities with support for breakpoints, watch variables, call stacks, and interactive debugging sessions, enabling you to identify and fix errors in your code efficiently.

5. **Version Control Integration:** Visual Studio Code integrates seamlessly with version control systems such as Git, providing essential features for source code management, branch visualization, conflict resolution, and collaboration with team members. Whether you are working on a solo project or as part of a distributed team, Visual Studio Code's version control integration simplifies the process of tracking changes and managing code repositories effectively. The interface of Visual Studio Code is shown in fig 5.2

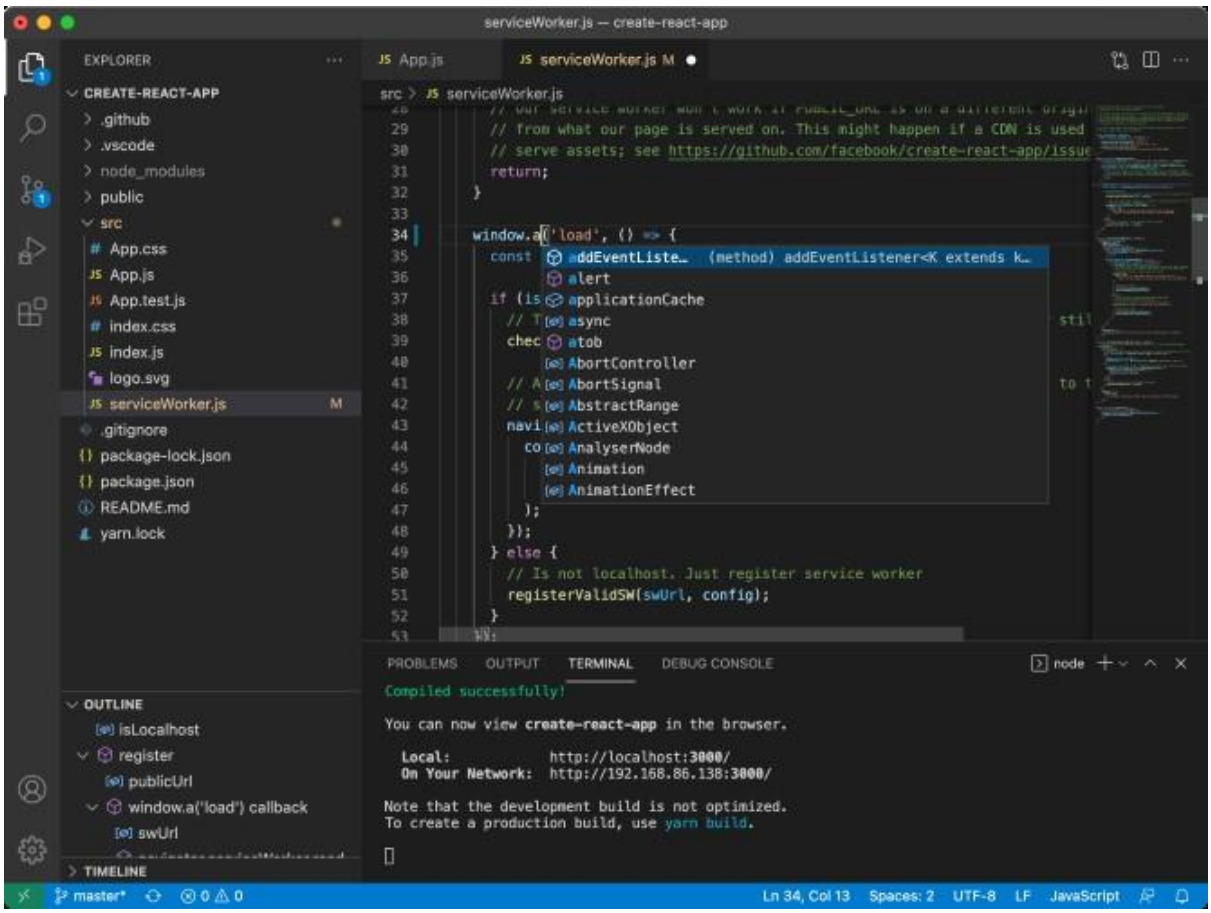


Fig 5.2 Interface of Vs Code

- **JupyterLab:** JupyterLab stands as a cutting-edge web-based interactive development environment (IDE) that has revolutionized the way data scientists, researchers, and developers work with code, data, and visualizations. Built on top of the Jupyter Notebook technology, JupyterLab offers a flexible, extensible, and user-friendly environment for interactive computing, exploration, and collaboration. In this comprehensive exploration, we delve into the myriad facets of JupyterLab, elucidating

its key features, advantages, use cases, and the transformative impact it has had on the world of interactive computing

Key Features of JupyterLab

1. **Flexible Notebooks:** JupyterLab provides support for Jupyter notebooks, which are interactive documents that combine live code, equations, visualizations, and narrative text. With Jupyter notebooks, users can write, execute, and share code in a collaborative and reproducible manner, making it ideal for data exploration, prototyping, and communication of results
2. **Multi-document Interface:** Unlike traditional Jupyter notebooks, which are limited to a single document interface, JupyterLab offers a multi-document interface that allows users to work with multiple notebooks, text files, and other documents simultaneously. This enables users to organize their work more efficiently, switch between different documents seamlessly, and compare results across multiple analyses
3. **Integrated Development Environment:** JupyterLab functions as a full-fledged integrated development environment (IDE) with features such as syntax highlighting, code completion, and code linting for various programming languages, including Python, R, Julia, and more. Users can write, edit, and execute code directly within JupyterLab, without the need for additional tools or environments
4. **Extensible Architecture:** JupyterLab's extensible architecture allows users to customize and extend the IDE with additional functionality through plugins and extensions. Users can install pre-built plugins from the JupyterLab extensions registry or develop custom plugins to meet their specific needs. This flexibility enables users to tailor JupyterLab to their workflow, preferences, and use cases
5. **Rich output and Visualization:** JupyterLab supports rich output and visualization capabilities, allowing users to generate interactive plots, charts, and dashboards directly within their notebooks. Users can leverage libraries such as Matplotlib, Plotly, and Bokeh to create stunning visualizations that enhance data exploration and communication. The interface of JupyterLab is shown in fig 5.3

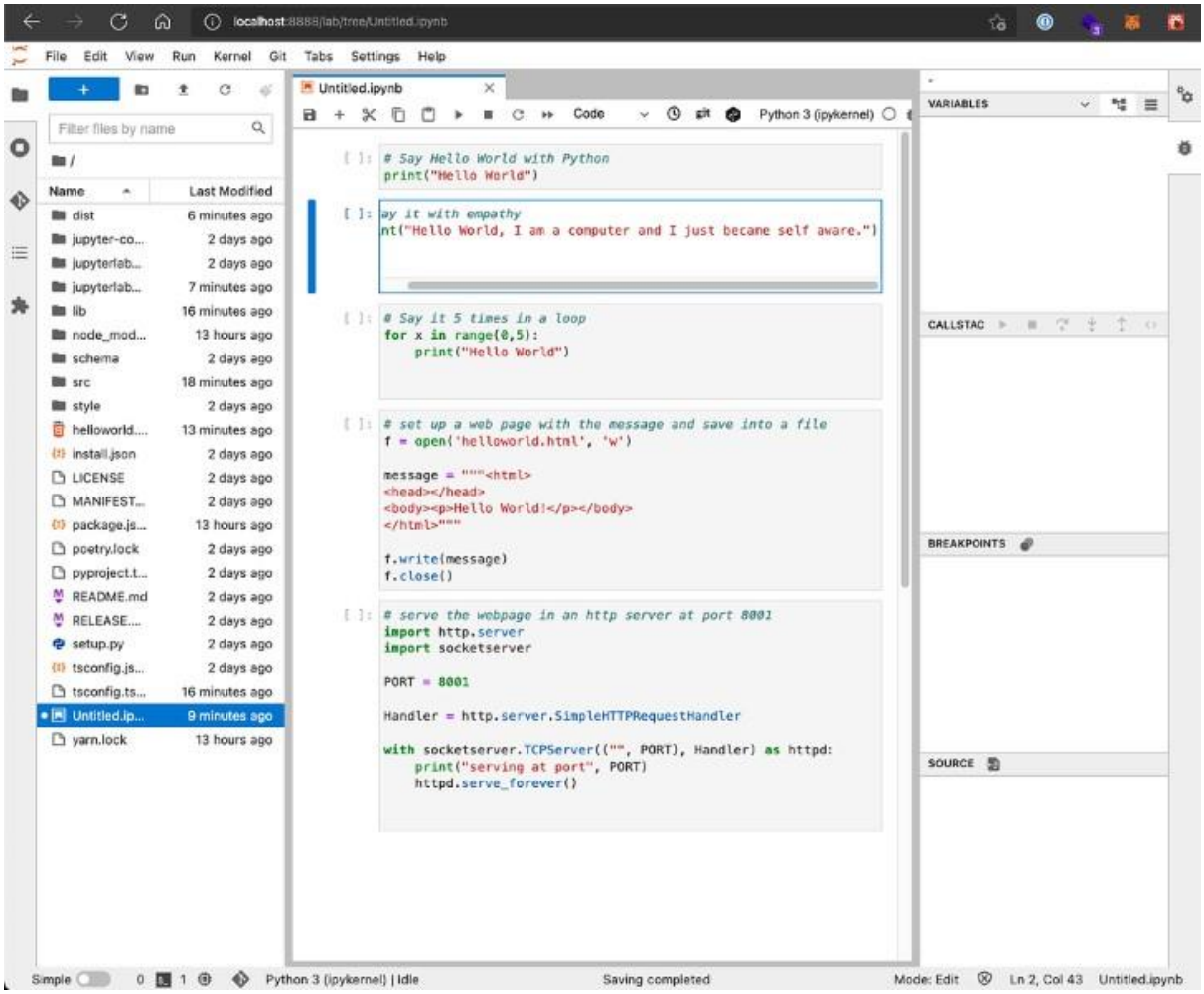


Fig 5.3 Interface of JupyterLab

- **Spyder:** Spyder stands as a dedicated Integrated Development Environment (IDE) designed specifically for scientific computing and data analysis tasks in Python. With its intuitive interface, powerful features, and seamless integration with scientific libraries, Spyder has become a go-to choose for researchers, scientists, and data analysts worldwide. In this comprehensive exploration, we delve into the myriad facets of Spyder, elucidating its key features, advantages, use cases, and the transformative impact it has had on the world of scientific computing.

Key Features of Spyder

1. **Integrated Console:** One of the standout features of Spyder is its integrated console, which allows users to execute Python code interactively and view the results in real-time. The console provides a familiar environment similar to the Python REPL (Read-Eval-Print Loop), enabling users to test code snippets, explore data, and prototype algorithms effortlessly

2. **Variable Explorer:** Spyder includes a powerful Variable Explorer tool that allows users to inspect and manipulate variables in memory. With the Variable Explorer, users can view the contents of variables, arrays, and data frames, as well as filter, sort, and edit data directly within the IDE. This feature is particularly useful for data analysis tasks, as it enables users to explore datasets and perform data manipulation operations with ease
3. **Interactive Help:** Spyder provides interactive help functionality that allows users to access documentation, function signatures, and examples for Python libraries and modules directly within the IDE. Users can simply hover over a function or method call to view its documentation, or use the built-in help pane to search for specific topics and keywords. This feature enhances productivity and code comprehension, especially for users who are new to Python or unfamiliar with certain libraries
4. **Code Editor and Syntax Highlighting:** Spyder's code editor offers advanced features such as syntax highlighting, code folding, and automatic indentation, making it easy to write and edit Python code. The code editor supports multiple programming languages, including Python, R, and Julia, and provides features like code completion, code linting, and code analysis to help users write clean, efficient code. The interface of spyder is shown in fig 5.4

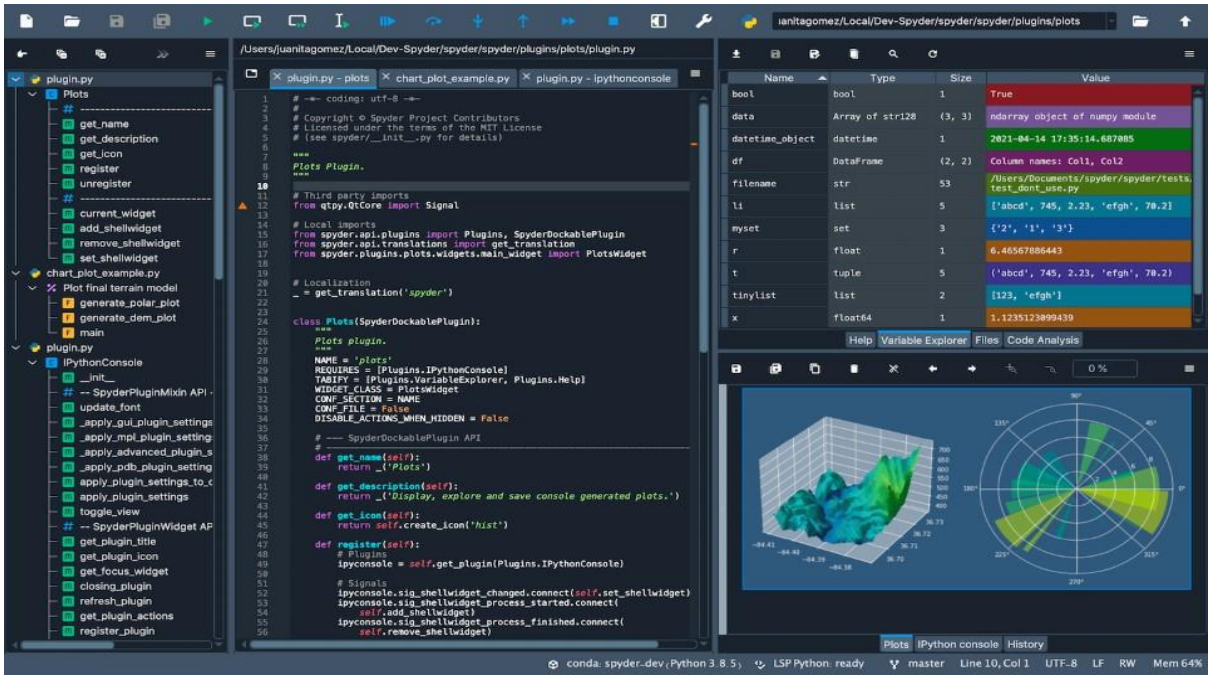


Fig 5.4 Interface of Spyder

5.2 Fritzing

Fritzing is a user-friendly software tool and open-source initiative designed to support designers, artists, researchers, and hobbyists in the realm of electronics prototyping and development. Renowned for its intuitive interface and comprehensive features, Fritzing serves as a versatile platform for creating, sharing, and documenting electronic circuits and projects. In this comprehensive exploration, we delve into the various facets of Fritzing, elucidating its key features, advantages, use cases, and its significance in the domain of electronics prototyping and education.

5.2.1 Evolution of Fritzing

Fritzing was initially conceived in 2007 as a project by the Interaction Design Lab at the University of Applied Sciences Potsdam, Germany. The primary objective was to develop a tool that simplifies the process of documenting and sharing electronic projects, particularly for non-engineers and beginners. Over the years, Fritzing has evolved into a mature and widely adopted software solution, offering a rich set of features and functionalities tailored to the needs of electronics enthusiasts and educators worldwide.

5.2.2 Key Features of Fritzing

- **Breadboard view:** Fritzing provides a virtual breadboard view that allows users to design and simulate electronic circuits using a familiar breadboard interface. Users can drag and drop components such as resistors, capacitors, LEDs, and microcontrollers onto the breadboard canvas and connect them using virtual wires. This intuitive interface enables users to quickly prototype and experiment with circuit designs without the need for physical components
- **Schematic view:** Fritzing offers a schematic view that provides a visual representation of the connections and components used in the circuit design. Users can switch between the breadboard and schematic views to gain a deeper understanding of the circuit topology and relationships between components. The schematic view also supports features such as zooming, panning, and component labelling, making it easier to analyze and document complex circuits. Fritzing's intuitive interface and user-friendly design make it accessible to users of all skill levels, from beginners to experienced electronics enthusiasts.

- **PCB view:** Fritzing includes a PCB (Printed Circuit Board) view that enables users to design custom PCB layouts for their electronic projects. Users can transfer their circuit designs from the breadboard and schematic views to the PCB view and arrange components on a virtual circuit board. Fritzing provides tools for routing traces, adding vias, and customizing board dimensions, allowing users to create professional-quality PCB designs for fabrication.
- **Parts Library:** Fritzing boasts an extensive parts library containing a wide range of electronic components, modules, and microcontrollers from various manufacturers. Users can browse the parts library, search for specific components, and add them to their circuit designs with ease. The parts library is continuously updated and expanded by the Fritzing community, ensuring that users have access to the latest and most used components
- **Projects sharing and Documentation:** Fritzing facilitates project sharing and documentation through its built-in features for exporting circuit designs as images, PDFs, or SVG files. Users can annotate their designs with text, labels, and notes, and generate comprehensive documentation for their projects. Additionally, Fritzing provides an online platform for users to share their projects, collaborate with others, and contribute to the growing repository of open-source electronic designs.

5.2.3 Advantage of Fritzing

- **Accessibility and Ease of use:** Fritzing's intuitive interface and user-friendly design make it accessible to users of all skill levels, from beginners to experienced electronics enthusiasts. The drag-and-drop functionality, visual feedback, and real-time simulation capabilities enable users to quickly prototype and experiment with circuit designs without requiring specialized knowledge or training
- **Educational Value:** Fritzing serves as a valuable educational tool for teaching electronics concepts, circuit design principles, and prototyping techniques in schools, universities, and maker spaces. The software's interactive nature, combined with its comprehensive documentation and project sharing features, facilitates hands-on learning and collaborative experimentation among students and educators
- **Rapid Prototyping:** Fritzing accelerates the prototyping process by enabling users to create and test electronic circuits virtually before building them physically. The

software's breadboard simulation and PCB layout capabilities allow users to iterate on their designs, troubleshoot potential issues, and refine their prototypes without the need for costly materials or equipment

- **Community and Collaboration:** Fritzing benefits from a vibrant and active community of users, contributors, and enthusiasts who share their knowledge, projects, and resources on the Fritzing platform. Users can collaborate on projects, provide feedback, and support to fellow users, and contribute to the ongoing development and improvement of the software. This collaborative ethos fosters creativity, innovation, and knowledge sharing within the Fritzing community
- **Open-Source Philosophy:** Fritzing embraces the principles of open-source software development, making its source code freely available to the public under the GNU General Public License (GPL). This open-source philosophy encourages transparency, community participation, and continuous improvement, ensuring that Fritzing remains accessible, adaptable, and responsive to the evolving needs of its users. The interface of Fritzing software simulator is shown in fig 5.5

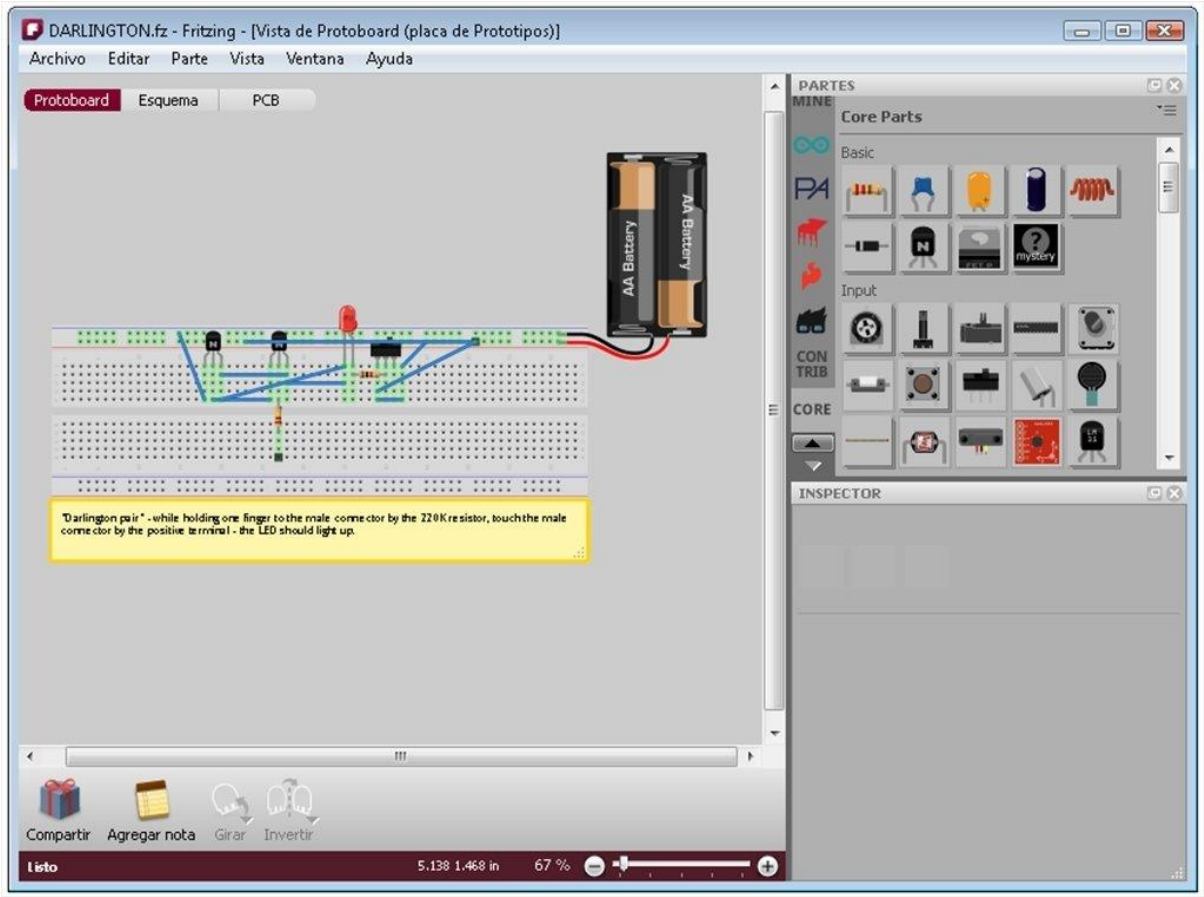


Fig 5.5 Interface of Fritzing Software simulator