

ADVANCED MILITARY SPYING AND BOMB DISPOSAL ROBOT

Submitted in partial fulfilment of the requirements for the award of
Bachelor of degree in Electronics and Communication Engineering

By

HARISH M (Reg. No – 39130290)
HARSHAVARDHAN M (Reg. No – 39130287)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRONICS AND ELECTRICAL ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHISALAI, CHENNAI - 600119

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work **MEKA.HARSHAVARDHAN (39130287) and METTU.HARISH (39130290)** who carried out the project entitled " **ADVANCED MILITARY SPYING AND BOMB DISPOSAL ROBOT** " under our supervision from November 2022 to April 2023.

Internal Guide

Dr. S.LALITHAKUMARI, M.E., Ph.D.,

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 27.04.2023

Internal Examiner

External Examiner

DECLARATION

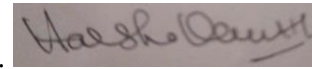
We, work **MEKA.HARSHAVARDHAN (39130287)** and **METTU.HARISH (39130290)** here by declare that the Project Report entitled " **ADVANCED MILITARY SPYING AND BOMB DISPOSAL ROBOT** " done by us under the guidance of **Dr. S . Dr. S.LALITHAKUMARI, M.E., Ph.D.**, Associate Professor, Dept. of Electronics and Communication Engineering, Sathyabama institute of science and technology , Chennai, submitted partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE :27.04.23

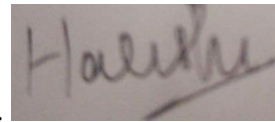
PLACE :Chennai

SIGNATURE OF CANDIDATES

1.



2.



ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N.M. NANDHITHA M.E., Ph. D, Dean**, School of Electronics, **Dr. T. RAVI M.E., Ph.D.**, Head of the Department of Electronics and Communication Engineering for providing us the necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.S. LALITHAKUMARI M.E., Ph.D**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our phase-2 project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This work proposed a model of wireless bomb disposal robot which helps to aid an automation in defence applications. In this work, it is focused to develop robotic hardware, which controls the movement of the robot. The embedded hardware is used to develop Node MCU microcontroller motor running default from program takes the data and controls the motors of the robot by the motor driver L293D.

In order to set the system that operates the arm forward, backward, left and right movements. A metal detector is connected to the robot to detect the metal. Through the smartphones are have been interfaced with the device by using the Bluetooth app, which it detects the metal, it will blink on the app. Thus, this proposed system makes use of robotic arm and robotic vehicle to enter a high-risk area.

This work finds its applications, where there is high risk for humans to enter hazard places. The system proposed robotic arm forward, backward, left and right directions as well as the robotic vehicle which helps not only to enter high risk areas but also it will pick the objects and destroys.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	ix
	LIST OF TABLES	x
	LIST OF ACRONYMS	xi-xii
CHAPTER 1	INTRODUCTION	1-13
	1.1 Introduction	1-3
	1.2 Literature Survey	3-5
	1.3 Existing System	5-6
	1.4 Proposed System	6-7
	1.5 Block Diagram	7
	1.6 Hardware Requirements	7
	1.7 Circuit Diagram	8
	1.8 Hardware Explanation	8
	1.8.1 IOT (internet Of Things)	8
	1.8.2 IOT	8-10
	1.8.3 IOT work	10
	1.9 Features of IOT	11-12
	1.10 Advantages and Disadvantages of IOT	12-13
CHAPTER 2	EMBEDDED DEVICES	13-20
	2.1 Embedded Devices(system) in IOT	13
	2.2 Embedded system Hardware	14
	2.3 Basic Embedded System	15
	2.4 IOT Ecosystem	15-17
	2.5 IOT Decision Framework	17-18
	2.6 Decision Area	18
	2.7 Decision Area of IOT Decision Framework	19-20
	2.8 IOT Architecture	20

CHAPTER 3	STAGES OF IOT SOLUTIONS	21-35
	3.1 Stages of IOT Solutions Architecture	21-22
	3.2 Smart Objects in IOT	22-24
	3.3 IOT Devices	24-25
	3.4 IOT Devices and Technologies	25-34
	3.4.1 NodeMCU	26
	3.4.2 ESP8266 Arduino Core	26-27
	3.4.3 Introduction of NodeMCU	27
	3.4.4 NodeMCU Development version 1	27-28
	3.4.5 NodeMCU Development version 2	28-29
	3.4.6 NodeMCU with ESPlorer IDE	29
	3.4.7 NodeMCU with Arduino IDE	29-30
	3.4.8 Metal Detector	30-31
	3.4.9 DC Motors	31-32
	3.4.10 Bluetooth Module	32-33
	3.4.11 Buzzer	34
	3.4.12 Robotic Arm	34-35
CHAPTER 4	CONSTRUCTION	36-44
	4.1 Construction	36-37
	4.2 DC Motor Operation	37-38
	4.3 DC Motor Equivalent Circuit	38-40
	4.4 Introduction to L293D IC	40-41
	4.5 L293D Connections	41-42
	4.6 Truth Table	42-43
	4.7 Motor Driver	43
	4.7.1 A Motor Driver	43
	4.7.2 Remote Controlled Robots	44
	4.7.3 Autonomous Robots	44

CHAPTER 5	SOFTWARE SPECIFICATIONS	45-58
	5.1 Arduino uno and its Programming	45
	5.2 Overview	45-47
	5.3 Arduino Software (IDE)	47-48
	5.3.1 Writing Sketches	48-49
	5.3.2 File	49-50
	5.3.3 Edit	50
	5.3.4 Sketch	51
	5.3.5 Tools	51-52
	5.3.6 Help	52
	5.3.7 Sketchbook	52-53
	5.3.8 Tabs, Multiple Files and Compilation	53
	5.3.9 Uploading	53-54
	5.3.10 Libraries	54
	5.3.11 Third-Party Hardware	54
	5.3.12 Serial Monitor	54-55
	5.3.13 Preferences	55
	5.3.14 Language Support	56-57
	5.3.15 Boards	57-58
CHAPTER 6	CONCLUSION & FUTURE SCOPE	59-62
	6.1 Conclusion	59
	6.2 Future Scope	60
	6.3 References	61-62

LIST OF FIGURES

Figure No	TITLE	PAGE NO
1.5	Block Diagram	7
1.7	Circuit Diagram	8
1.8.2	Internet of things	9
1.8.3	IOT Architecture	11
2.1	Embedded devices in IOT	14
2.3	Basic Embedded System	15
2.4	IOT Ecosystem	16
2.5	IOT DECISION Framework	18
2.7	Decision Area of the IOT Framework	19
2.8	IOT Architecture	20
3.1	The IOT 4 Stages Architecture	22
3.2	Smart Objects in IOT	23
3.3	IOT Devices	24
3.4	IOT Devices and Technologies	25
3.4.3	NodeMCU	27
3.4.5	NodeMCU Version 1	28
3.4.8	Metal Detector	30
3.4.10	Bluetooth Module	33
3.4.11	Buzzer	34
4.1	Construction	35
4.2	DC Motor Operation	36
4.3	DC Motor Equivalent Circuit	37

LIST OF TABLES

4.6	Truth table	41
-----	-------------	----

LIST OF ACRONYMS

PC	Personal Computer
UNIX	Uniplexed information computing system
OS	Operating System
CPU	Central processing Unit
DSP	Digital signal Processor
RAM	Random access memory
LCD	Liquid Crystal Display
LED	Light Emitting Diode
USB	Universal Serial Bus
IEEE	Institute of electrical and electronics engineers
DVD	Digital Video Disc
VCO	Voltage Controlled Oscillator
ECG	Electrocardiograph
EEG	Electro encephalogram
ISDN	Integrated Service Digital Network
ATM	Asynchronous Transfer Mode
ATM	Automatic Teller machine
MCU	Microcontroller
ADC	Analog to Digital Converter
DAC	Digital to Analog converter
ISP	In system Programmable
CMOS	Complementary Metal oxide Semiconductor
TTL	Transistor to transistor logic
ALE	Advance latch Enable
PSEN	Program store Enable

VPP	Value payable post
SFR	Special Function Register
MUL	Multiplication
DIV	Division
DPTR	Dynamic progressive transmission of resolution
DCEN	Down Counter Enable
PAN	Personal Area Network
GFSK	Gaussian frequency shift keying
GPS	Global Posting System
ISM	Information System Management
SIG	Special Interest Group
HCI	Host Controller Interface
UART	Universal Asynchronous Receiver Transmitter
DC	Direct Current
AC	Alternating Current
CD-ROM	Compact Disc Read only memory

CHAPTER 1

INTRODUCTION

1.1 Introduction

Different technological terms such as Telecommunications, Internet of Things (IoT), and robotics have been considered a vital part of our daily activities. Although its advances and limitations, innovative tech can be solved fundamental issues and save lives for many people for political or financial purposes. In the electronic era, speed, flexibility, and automation are major defiance that is enabling researchers to meet the challenges of the society against the quick development of the techs. Robotics have been becoming dynamically significant for several standard applications. Applications such as military, Salvage and Urban Hunt. Due to its human reduction activities in a severe environment. Effectiveness metal and landmine detections are two vital research areas that still considered more attractive to researchers due to investing tech. According to Landmine and Cluster Munition Monitor report in 2014, more than 110 million landmines in the ground scattered in 68 countries. So, landmines and metal detector robots are the advances innovation of the crucial and most dangerous part of the humanitarian detection process. Multi-sensor robot, path planning algorithm, and vehicle-mounted sensors are different strategies that used to search and detect mines. These applications are safer and more efficient due to they provide a safe route for the soldiers through minefields. Robotics, communication, and data analysis have been advanced with high-speed achieved in landmine detection domain. Data management, analysis, and archiving for the detection, navigating and mapping area will open the door to more deployed possibilities for furthermore mapping and detection areas. Automation mechanism presents easier and faster scanning process due to it provides the guarantee to more rapid the scanning processing and investigation automatically. As well, it fits more deployed capabilities based on the big data received to disclose and recognize metals and mines from other objects. Several attempts by authors have been made for research and development perspectives concern assist or automate human deminers in the humanitarian demining and scanning process to reduce efforts, time, cost, human risks and dangerous issues by proposing diverse efficient and suitable robotics

landmine detection with several sensors design. Recent robotics research involves various trends such as sensor technology, Ground-Penetrating Radar (GPR), Electromagnetic Induction (EMI), Nuclear Quadrupole Resonance (NQR) and so on to perform multifunction's such as detection, deactivating, extracting, and demining for explosive devices, cluster munitions, landmines as well avoiding obstacles; These trends can deploy in diverse soil kinds such as foliage, dry and desert soil based on Aerial and wheel techniques for landmine detection. Proposed Marwa which is a landmine detecting robot. Marwa is supported with a robotic arm whose payload a metal detector and provided visual information to be a visual serving system to benefit researchers with limited resources to work in this area. Also, presented a metal mine signals for detection purpose to estimate the depth of the metallic targets. The metal mine detector using robotic manipulator to take advantage of high precision scanning of the minefield; the proposed detector was fast and accurate in Humanitarian Demining; As well, its pre-build library containing data of many targets at different postures and depths. Whereas, a remote robot is conducted to identify personnel landmines in diverse domains. The detector was obtained (87.5%) accuracy from a set of 8 different materials in identifying mines. Also, the detection system can be implemented in different terrains. introduced three-way to avert the issues of using one technology by using data fusion of multi-sensor system based on developing decision level fusion to decrease false alarms. While, presented multi-sensor data-fusion approach by a high accuracy geo-referencing the field-data acquired by multiple platforms to localize and detect purposes in landmines domain. Likewise, introduced a novel landmine detection sensor that based on the principle of 2-DOF vibration absorber system. The proposed new sensor gives the sensitivity of 1559 Hz/ (MN/m) and linearity better than (95%). Besides, in 2016, Hank and Haddad presented a mobile robot for autonomous navigation problem based on a hybrid approach; the produced navigator subjected to perform an emergency task with shorter execution times. As alongside, designed and fabricated an efficient wireless controlled robot to detect landmine in defence fields as well avoiding the obstacles robustly; H-Bridge module used to control the robot wheels and wireless camera added to capture and located off the robot destination. Also, proposed a low-altitude autonomous flight to detect landmines; the system called Backstepping which is an integrated system architecture based on lightweight GPR.

As well, a hybrid platform introduced by Gareeb et al. (2017), speed data transferring and transmission quality to improve central unit destination that is based on web server and a database server application to store data regarding the navigation field for current mapping and detection or future investigation purpose. In this research, highlights on metal detection issues. Also, the capability to detect mines, avoid obstacles. The proposing system using the concept of metal detection sensor and ultrasonic sensor, wireless communication via controlled robotics vehicle remotely for flat surfaces and dry, desert soils to design and fabricate detect metals without human harm. Also, proposed an integrated management and achieving system to control, manage and stored the navigation results with online structured query language databased registry on a server side to give high-security system. As well, conduct a comparative analysis of the proposed robot detection systems and focus on their main findings and systems features like detection, controlling, management, and archiving information in the database server for future mapping and navigation. The rest of the article is structured as follows: In Section 2, describes hardware and software that required to produce the controlling, management, and archiving system that includes three subsections are metal detector robot architecture, management and controlling the metal detector robot, and archiving database system. Section 3 presents all steps and processes to implement the proposed detection system. Besides, results and discussion will be included in Section 4. Whereas, Section 5 gives a comparison of the works that have been done to detect landmine detection. Finally, Section 6 concerns with conclusions and future directions.

1.2 Literature Survey

Arduino Military Spying and Bomb Detecting Robot

The work is designed to develop a War field robot which is capable of detecting bombs land mines in its path and which is wirelessly controlled through RF module. It is used to monitor the Warfield. The robot can be moved in all the directions using the remote wirelessly. The robot system is also used for bomb detection. The controlling device of the whole system is an Arduino. Due to that circuit complexity is reduced and performance speed is increased. Whenever, land mines or bombs are detected, it alerts through blinking LEDs of system.

The Arduino used in the project are programmed using Embedded C language.

Wireless Bomb Disposal Robot

The bomb detectors and disposal system work only with the presence of experts. But this way of analysing takes more time and make risk to life of experts. The Wireless Bomb Disposal Robot uses a control application, at the user end to control the robot remotely using Wireless technology. The bomb technician controls the robot using this application at control site. Input from the user is transmitted over ZigBee to the Receiver, where it is received, identified and given to the appropriate module (Robot) to act. The Robot consists of a Base, a robotic Arm and a wireless camera on it. We have used DC motors for the elbow and the gripper of the robotic arm. As we are not risking the life of a bomb expert or any other commando. Hence introducing the safest way for disposing the explosive to save life of common people.

Wireless Bomb Defusing Robot with camera interfacing

The robot provides a safe distance threat assessment and increased capacity for explosive ordinance disposal, improving the effectiveness of bomb disposal teams. The robot's low -cost, intuitive operation and ease-of-maintenance promote its widespread appeal, thereby saving the lives of both law enforcement personnel and civilians.

Military spying and bomb disposal robot using IOT

The Wireless Bomb Disposal Robot uses a control application, at the user end to control the robot via android application using Wireless technology. The whole system is controlled via android application. The bomb technician controls the robot using this application at control site. Input from the user is transmitted over to the Receiver, where it is received, identified and given to the appropriate module (Robot) to act.

Landmine Detection Using Autoencoders on Multiploidization GPR Volumetric Data

Paolo Bestagini, Federico, Lombardi, Maurizio Lualdi,

Buried landmines and unexploded remnants of war are a constant threat

for the population of many countries that have been hit by wars in the past years. The huge number of casualties has been a strong motivation for the research community toward the development of safe and robust techniques designed for landmine clearance.

Landmine detection using multispectral images

The conditions in which images are obtained to perform multispectral detection of landmines have a direct influence on the methods that are used to perform automatic detection of landmines. In this article, two methodologies are proposed, one using traditional classifiers and the other using Deep Learning, namely a Convolutional Neuronal Network (CNN). In the first methodology, classifier fusion techniques are also used.

Multiparameter Adaptive Target Classification Using Full-Polarimetric GPR: A Novel Approach to Landmine Detection.

Full-polarimetric ground penetrating radar (FPGPR) can measure the ability of an object to change the polarization of electromagnetic waves. Compared to the traditional GPR, it has a stronger capability to identify underground objects. In recent years, a series of polarization decomposition methods have been applied to the FPGPR data processing to obtain the polarimetric attributes and enhance the capability of targets identification.

1.3 Existing System

Existing system

- In army search operations and wars, soldiers become lost and get injured. There are many developments which give ability to track the location of soldier at any moment at any place.
- The main aim of the project is to provide medical monitoring for

soldiers in real time. In these existing developments or systems,

- Bluetooth technology etc. have been used for wireless transmission of position information and bio sensor's data of soldiers.
- Moreover, the landmines are detected by handheld detectors and only a particular soldier may have that device and the fellow soldiers follow that particular soldier's information regarding the buried landmines.

Disadvantages

- Not user friendly.
- Less range transmission.
- Cannot be implemented in all fields.
- More expensive.

1.4 Proposed System

- The given proposed is done by using NodeMCU controlled robot the project is mainly concerned on surveillance in border.
- Metal detector is used to detects the any land mines, disposal items. When the robot is detects the mines send an alert SMS through mail.
- The robot platform is control by using the internet streaming. If any object is detected.
- This project was about robotic movement control through smartphones. Many researchers have developed such robotic movement control system using smartphones.

Advantages

- user friendly.
- Long range transmission.
- Can be implemented in all fields.
- Less expensive.

1.5 Block Diagram

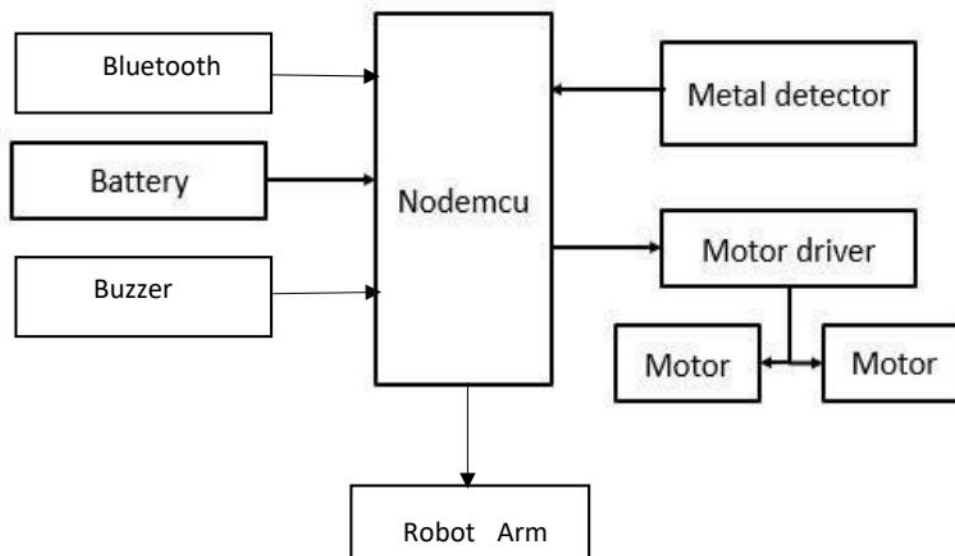


Fig: 1.5 Block Diagram

1.6 Hardware Requirements NodeMCU

Robotic Arm

Metal detector

Motor Driver

DC Motor Voltage

Regulator

Capacitors

Bluetooth Module

Buzzer

1.7 Circuit diagram

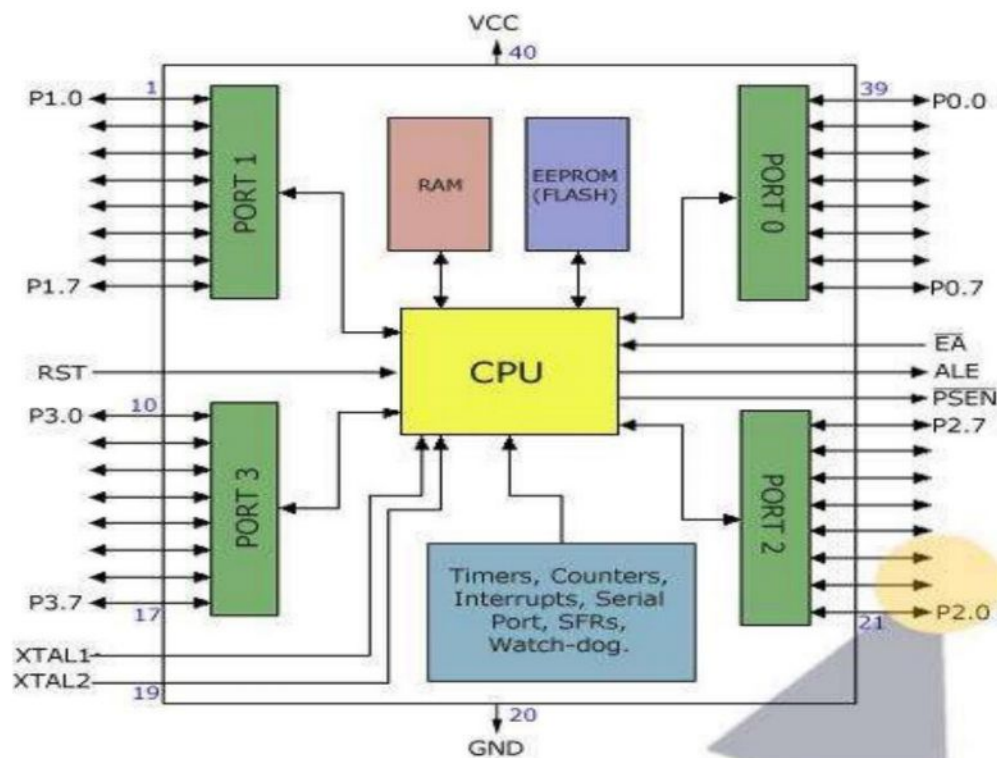


Fig: 1.7 Circuit Diagram

1.8 HARDWARE EXPLANTION:

1.8.1 IOT (Internet of Things):

IoT stands for Internet of Things, which means accessing and controlling daily usable equipment's and devices using Internet.

Our IoT tutorial includes all topics of IoT such as introduction, features, advantage and disadvantage, ecosystem, decision framework, architecture and domains, biometric, security camera and door unlock system, devices, etc.

1.8.2 Internet of Things (IoT)

Let's us look closely at our mobile device which contains GPS Tracking, Mobile Gyroscope, Adaptive brightness, Voice detection, Face detection etc. These components have their own individual features, but what about if these all communicate with each other to provide a better environment? For example, the phone brightness is adjusted based on my GPS location or my direction.

Connecting everyday things embedded with electronics, software, and sensors to internet enabling to collect and exchange data without human interaction called as the Internet of Things (IoT).

The term "Things" in the Internet of Things refers to anything and everything in day-to-day life which is accessed or connected through the internet.

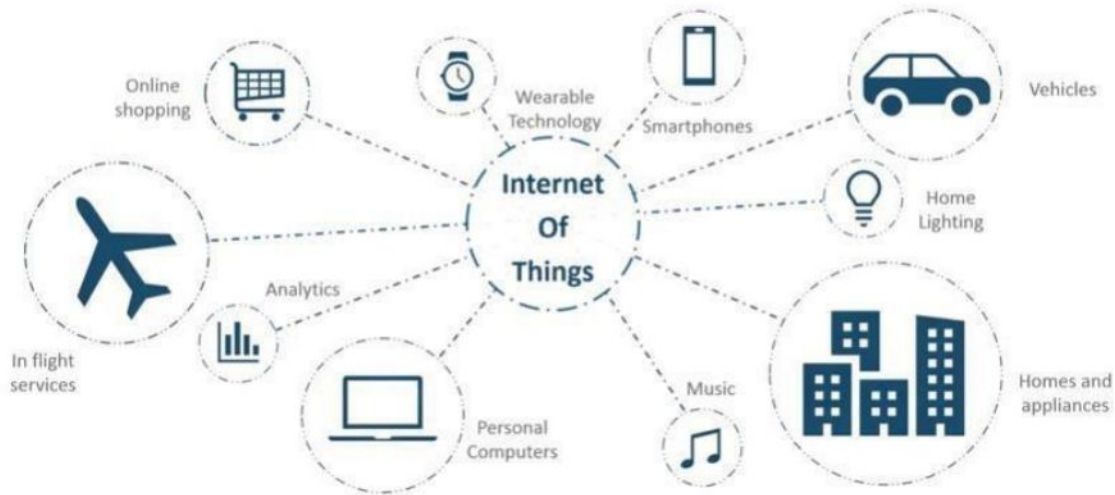


Fig: 1.8.2 INTERNET OF THINGS

IoT is an advanced automation and analytics system which deals with artificial intelligence, sensor, networking, electronic, cloud messaging etc. to deliver complete systems for the product or services. The system created by IoT has greater transparency, control, and performance.

As we have a platform such as a cloud that contains all the data through which we connect all the things around us. For example, a house, where we can connect our home appliances such as air conditioner, light, etc. through each other and all these things are managed at the same platform. Since we have a platform, we can connect our car, track its fuel meter, speed level, and also track the location of the car.



Fig :1.8.3 Internet of Thing (IoT) Works

If there is a common platform where all these things can connect to each other would be great because based on my preference, I can set the room temperature. For example, if I love the room temperature to be set at 25 or 26-degree Celsius when I reach back home from my office, then according to my car location, my AC would start before 10 minutes I arrive at home. This can be done through the Internet of Things (IoT)

Internet of Thing (IoT) Works

The working of IoT is different for different IoT echo system (architecture). However, the key concept of there working is similar. The entire working process of IoT starts with the device themselves, such as smartphones, digital watches, electronic appliances, which securely communicate with the IoT platform. The platforms collect and analyse the data from all multiple devices and platforms and transfer the most valuable data with applications to devices.

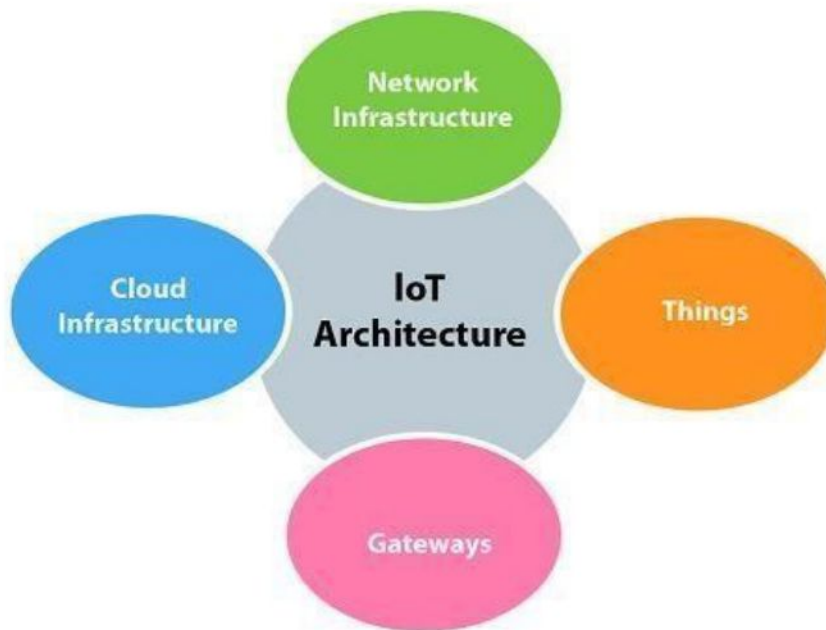


Fig: 1.8.4 IOT ARCHITECTURE

1.9 Features of IOT

The most important features of IoT on which it works are connectivity, analysing, integrating, active engagement, and many more. Some of them are listed below:

Connectivity: Connectivity refers to establish a proper connection between all the things of IoT-to-IoT platform it may be server or cloud. After connecting the IoT devices, it needs a high-speed messaging between the devices and cloud to enable reliable, secure and bi-directional communication.

Analysing: After connecting all the relevant things, it comes to real-time analysing the data collected and use them to build effective business intelligence. If we have a good insight into data gathered from all these things, then we call our system has a smart system.

Integrating: IoT integrating the various models to improve the user experience as well.

Artificial Intelligence: IoT makes things smart and enhances life through the use of data. For example, if we have a coffee machine whose beans have going to end, then the coffee machine itself order the coffee beans of your choice from the retailer.³

Sensing: The sensor devices used in IoT technologies detect and measure any change in the environment and report on their status. IoT technology brings passive networks to active networks. Without sensors, there could not hold an effective or true IoT environment.

Active Engagement: IoT makes the connected technology, product, or services to active engagement between each other.

Endpoint Management: It is important to be the endpoint management of all the IoT system otherwise, it makes the complete failure of the system. For example, if a coffee machine itself order the coffee beans when it goes to end but what happens when it orders the beans from a retailer and we are not present at home for a few days, it leads to the failure of the IoT system. So, there must be a need for endpoint management.

1.10 Advantages and Disadvantages of (IoT)

Any technology available today has not reached to its 100 % capability. It always has a gap to go. So, we can say that **Internet of Things** has a significant technology in a world that can help other technologies to reach its accurate and complete 100 % capability as well.

Let's take a look over the major, advantages, and disadvantages of the Internet of Things.

Advantages of IoT

Internet of things facilitates the several advantages in day-to-day life in the business sector. Some of its benefits are given below:

- **Efficient resource utilization:** If we know the functionality and the way that how each device works, we definitely increase the efficient resource utilization as well

as monitor natural resources.

- **Minimize human effort:** As the devices of IoT interact and communicate with each other and do lot of tasks for us, then they minimize the human effort.
- **Save time:** As it reduces the human effort then it definitely saves out time. Time is the primary factor which can save through IoT platform.
- **Enhance Data Collection:**
- **Improve security:** Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.

Disadvantages of IoT

As the Internet of things facilitates a set of benefits, it also creates a significant set of challenges. Some of the IoT challenges are given.

CHAPTER 2

2.1 Embedded Devices (System) in (IoT)

It is essential to know about the embedded devices while learning the IoT or building the projects on IoT. The embedded devices are the objects that build the unique computing system. These systems may or may not connect to the Internet. An embedded device system generally runs as a single application. However, these devices can connect through the internet connection, and able communicate through other network devices.

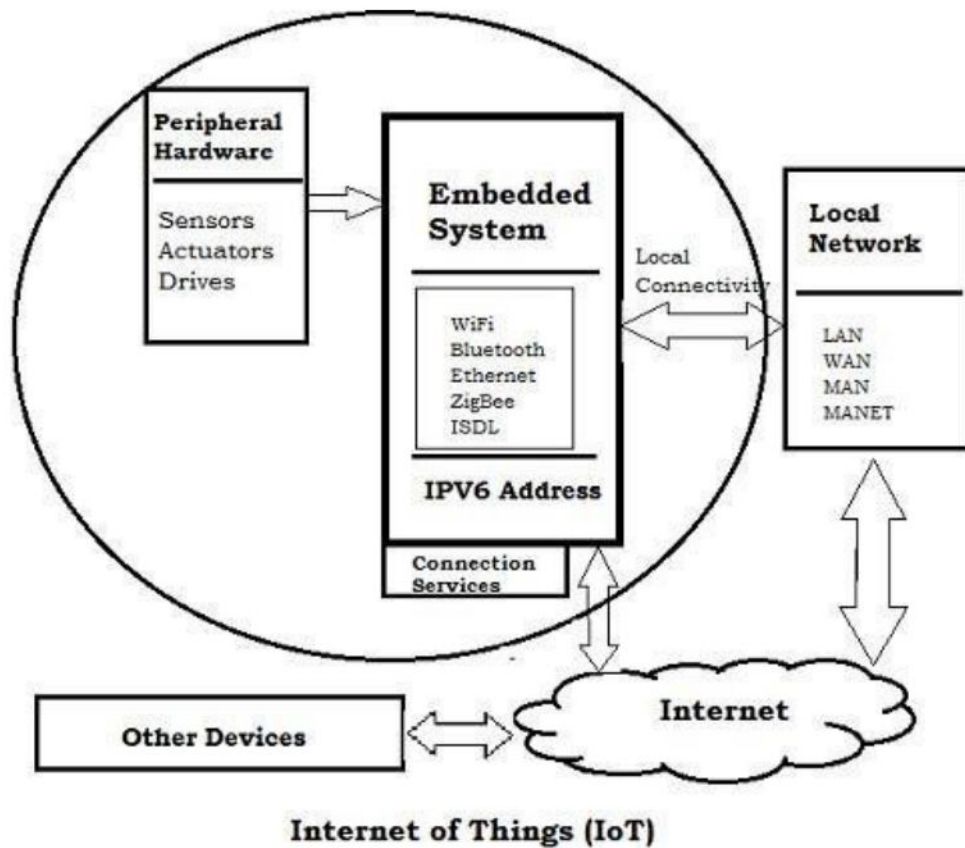
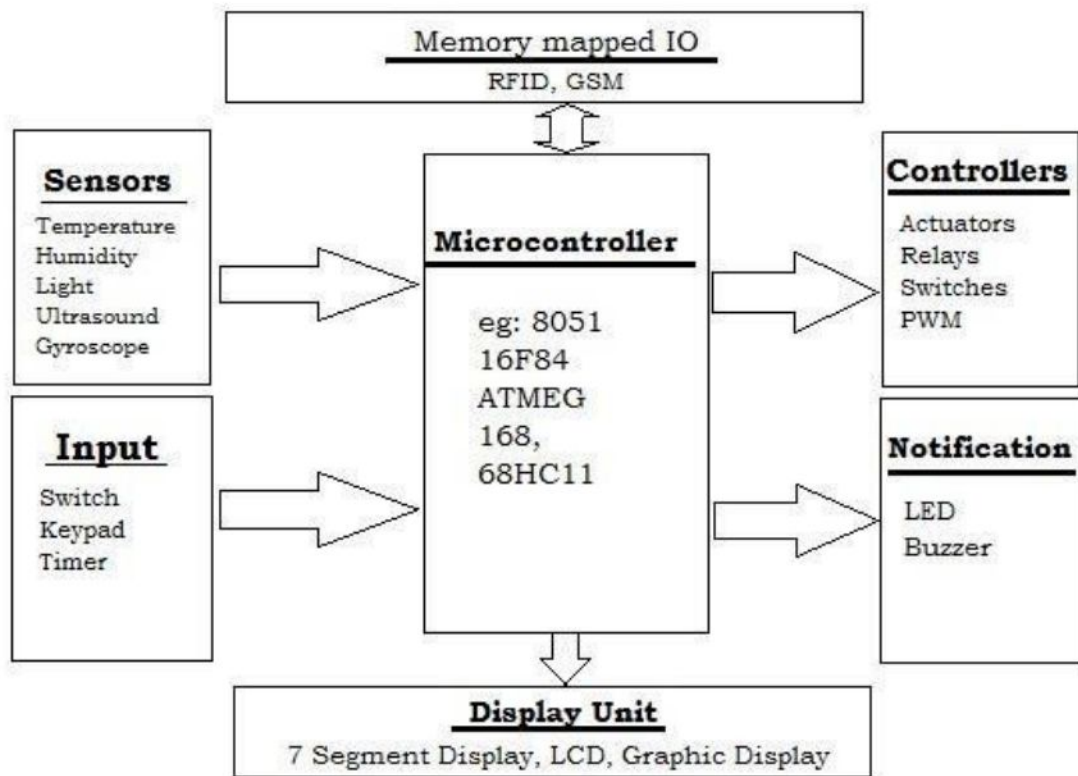


Fig: 2.1 Embedded devices in IOT

2.2 Embedded System Hardware

The embedded system can be of type microcontroller or type microprocessor. Both of these types contain an integrated circuit (IC).

The essential component of the embedded system is a RISC family microcontroller like Motorola 68HC11, PIC 16F84, Atmel 8051 and many more. The most important factor that differentiates these microcontrollers with the microprocessor like 8085 is their internal read and writable memory. The essential embedded device components and system architecture are specified below.



2.2 Fig: Basic Embedded System

2.3 Embedded System Software

The embedded system that uses the devices for the operating system is based on the language platform, mainly where the real-time operation would be performed. Manufacturers build embedded software in electronics, e.g., cars, telephones, modems, appliances, etc. The embedded system software can be as simple as lighting controls running using an 8-bit microcontroller. It can also be complicated software for missiles, process control systems, airplanes etc.

2.4 IoT Ecosystem

The IoT ecosystem is not easy to define. It is also difficult to capture its proper image due to the vastness and emerging possibility and the rapidity with which it is expanding in the entire sector. However, the IoT ecosystem is a connection of various kind of

IoT ecosystem is a connection of various kind of devices that sense and analyse the data and communicates with each other over the networks. In the IoT ecosystem, the user uses smart devices such as smartphones, tablet, sensors, etc. to send the command or request to devices for information over the networks. The device response and performs the command to send information back to the user through networks after analysed.

The typical IoT ecosystem is shown in below image, where the smarter devices send and receive data from the devices themselves in the environment that are integrate over network and Cloud Computing.

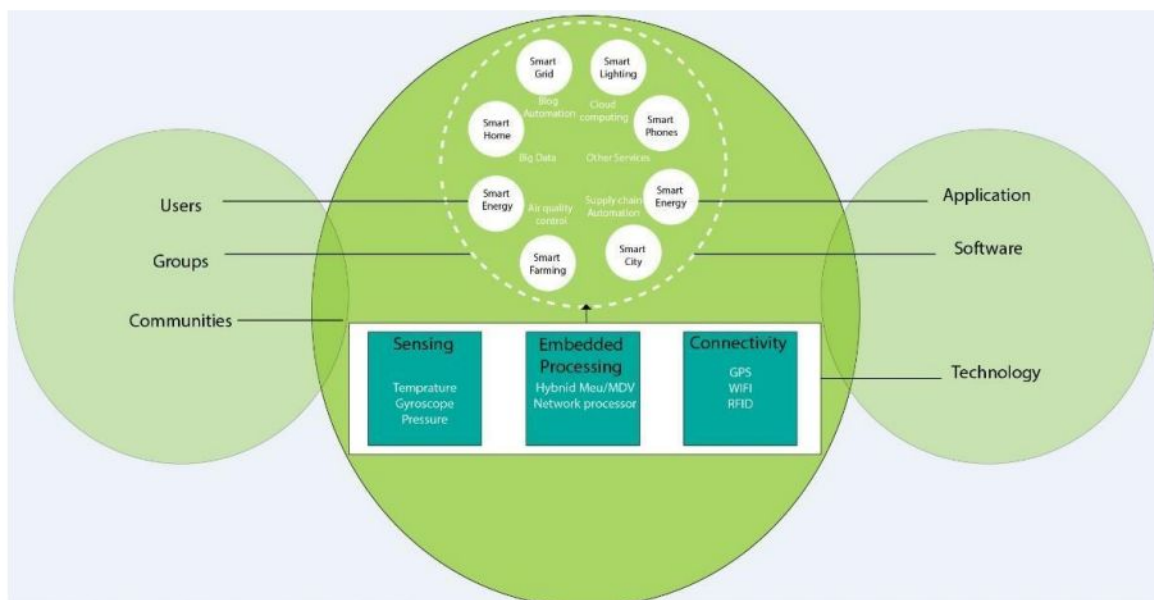


Fig: 2.4 IOT Ecosystem

The IoT is itself an ecosystem of network devices that transfer the data. It is also well interconnected with Big Data and Cloud Computing.

- **Sensing, Embedded processing, Connectivity:** The IoT ecosystem senses its surrounding like temperature, gyroscope, pressure, etc. and make the embedded processing using devices. These devices are connected through any type of devices such as GPS, WIFI, RFID, etc. over the networks.

- **Smart devices and environment, Cloud Computing, Big Data:** The data transfer or receive through smart devices and environments are communicated through Cloud Computing or others Servers and stored as Big Data.
- **Technology, Software, Application:** The IoT ecosystem uses any of different technologies, software and application to communicate and connect with smart devices and environment.
- **Users or groups of community:** The product or services generated by the IoT ecosystem are consumed by the users or the group of communities to serve the smart life.

2.5 IoT Decision Framework

The IoT decision framework provides a structured approach to create a powerful IoT product strategy. The IoT decision framework is all about the strategic decision making. The IoT Decision Framework helps us to understand the areas where we need to make decisions and ensures consistency across all of our strategic business decision, technical and more.

The IoT decision framework is much more important as the product or services communicates over networks goes through five different layers of complexity of technology.

1. Device Hardware
2. Device Software
3. Communications
4. Cloud Platform
5. Cloud Application

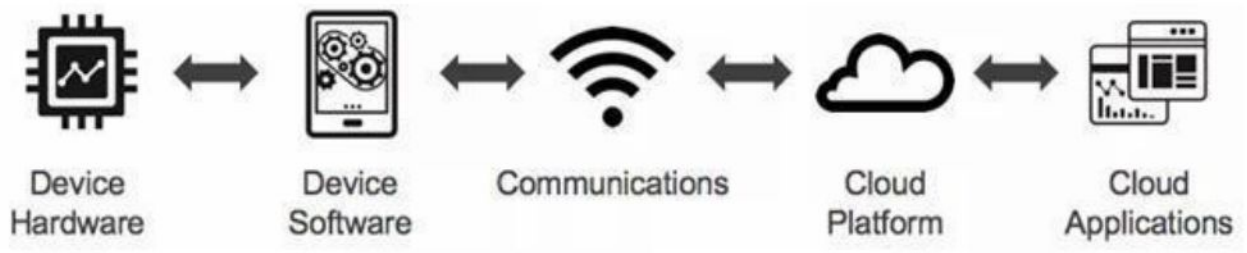


Fig: 2.5 IOT Decision Framework

2.6 Decision Area

The IoT decision framework pays attention to six key decision areas in any IoT product. These decision areas are:

1. User Experience (UX)
2. Data
3. Business
4. Technology
5. Security
6. Standards & Regulations

Each of these decision areas is evaluated at each of the IoT Technology Stack. The User Experience will be evaluated at Device Hardware, Device Software and so to provide the better user experience. Then at the next step Data Decision Area, we have to explore data considerations for all the stages of IoT Technology Stack.

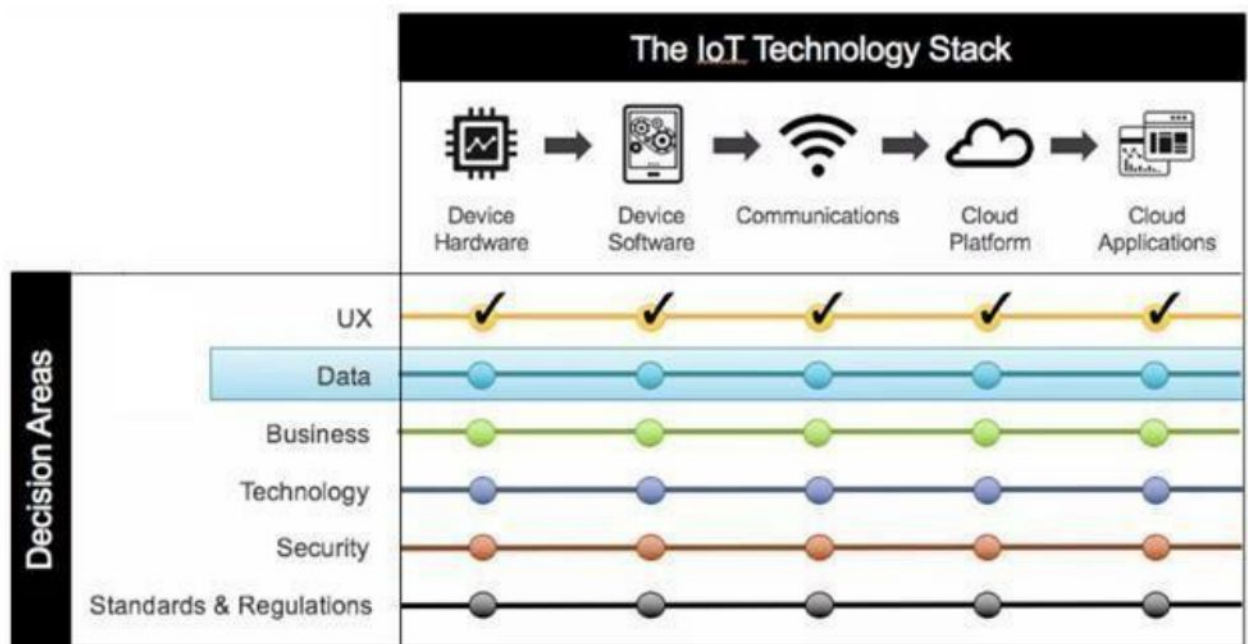


Fig: 2.7 Decision Area of the IoT Decision Framework

Let's see each of the Decision Area of IoT Decision Framework in detail:

1. **User Experience Decision Area:** This is the area where we concentrate about who are the users, what are their requirements and how to provide a great experience at each step of IoT stack without worrying about the technical details.
2. **Data Decision Area:** In this area, we make the overall data strategy such as the data flow over the entire IoT stack to fulfil the user's requirements.
3. **Business Decision Area:** Based on the previous decisions area, we make the decision how product or services will become financial potential. At each of the IoT Stack level are monetized about the costs of providing services.
4. **Technology Decision Area:** In this area, we work with the technology for each layer to facilitate the final solution.

5. **Security Decision Area:** After going through the implementation of technology it is important to decide and provide the security at each stage of the IoT Stack.
6. **Standards & Regulations Decision Area:** At the last stage of IoT Decision Area, we identify the standards and regulations of product or services that will affect your product at each layer of the IoT Stack.

2.8 IoT Architecture

There is not such a unique or standard consensus on the Internet of Things (IoT) architecture which is universally defined. The IoT architecture differs from their functional area and their solutions. However, the IoT architecture technology mainly consists of four major components:

Components of IoT Architecture ◦

Sensors/Devices ◦ Gateways and

Networks ◦ Cloud/Management

Service Layer ◦ Application Layer

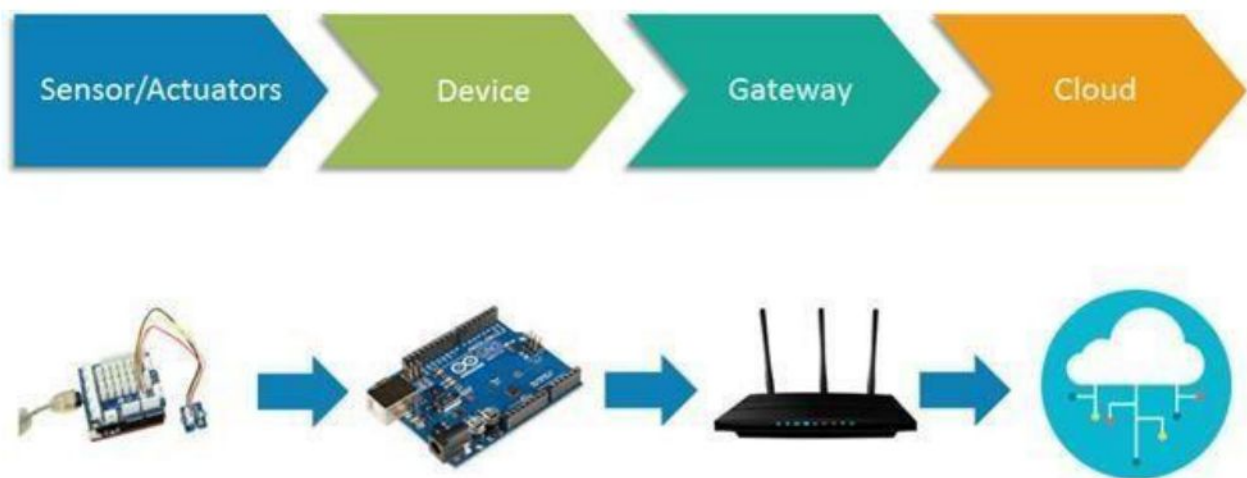


Fig: 2.8 IOT Architecture

CHAPTER 3

3.1 Stages of IoT Solutions Architecture

There are several layers of IoT built upon the capability and performance of IoT elements that provides the optimal solution to the business enterprises and end-users. The IoT architecture is a fundamental way to design the various elements of IoT, so that it can deliver services over the networks and serve the needs for the future.

Following are the primary stages (layers) of IoT that provides the solution for IoT architecture.

1. **Sensors/Actuators:** Sensors or Actuators are the devices that are able to emit, accept and process data over the network. These sensors or actuators may be connected either through wired or wireless. This contains GPS, Electrochemical, Gyroscope, RFID, etc. Most of the sensors need connectivity through sensors gateways. The connection of sensors or actuators can be through a Local Area Network (LAN) or Personal Area Network.
2. **Gateways and Data Acquisition:** As the large numbers of data are produced by these sensors and actuator need the high-speed Gateways and Networks to transfer the data. This network can be of type Local Area Network (LAN such as WIFI, Ethernet, etc.), Wide Area Network (WAN such as GSM, 5G, etc.).
3. **Edge IT:** Edge in the IoT Architecture is the hardware and software gateways that analyse and pre-process the data before transferring it to the cloud. If the data read from the sensors and gateways are not changed from its previous reading value then it does not transfer over the cloud, this saves the data used.
4. **Data centre/ Cloud:** The Data Centre or Cloud comes under the Management Services which process the information through

analytics, management of device and security controls. Beside this security controls and device management the cloud transfers the data to the end user's application such as Retail, Healthcare, Emergency, Environment, and Energy, etc.

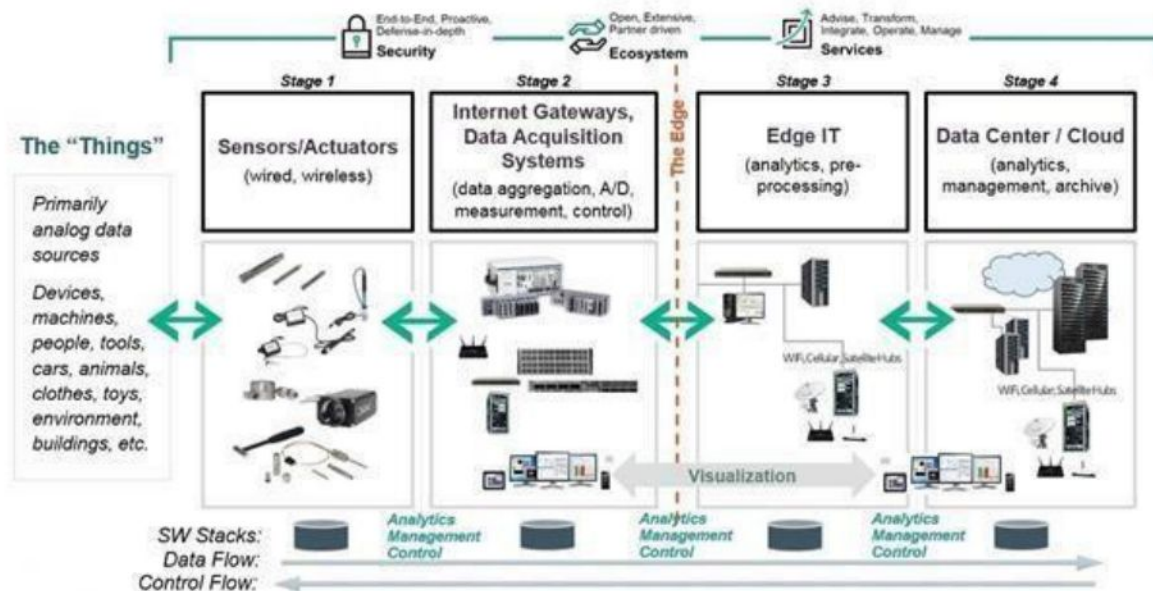


Fig: 3.1 THE IOT 4 STAGE ARCHITECTURE

3.2 Smarts Objects in IoT

The concept of smart in IoT is used for physical objects that are active, digital, networked, can operate to some extent autonomously, reconfigurable and has local control of the resources. The smart objects need energy, data storage, etc.

A **smart object** is an object that enhances the interaction with other smart objects as well as with people also. The world of IoT is the network of interconnected heterogeneous objects (such as smart devices, smart objects, sensors, actuators, RFID, embedded computers, etc.) uniquely addressable and based on standard communication protocols. In a day-to-day life, people have a lot of objects with internet or wireless or wired connection. Such as:

- **Smart phones**
- **Tablets**
- **TV's**

These objects can be interconnected among them and facilitate our daily life (smart home, smart cities) no matter the situation, localization, accessibility to a sensor, size, scenario or the risk of danger.



Fig: 3.2 Smart Objects in IOT

Smart objects are utilized widely to transform the physical environment around us to a digital world using the Internet of things (IoT) technologies.

A smart object carries blocks of application logic that make sense for their local situation and interact with human users. A smart object sense, log, and interpret the occurrence within themselves and the environment, and intercommunicate with each other and exchange information with people. The work of smart object has focused on technical aspects (such as software infrastructure, hardware platforms, etc.) and

application scenarios. Application areas range from supply-chain management and enterprise applications (home and hospital) to healthcare and industrial workplace support. As for human interface aspects of smart-object technologies are just beginning to receive attention from the environment.

3.3 IoT Devices

Internet of Things Devices is non-standard devices that connect wirelessly to a network with each other and able to transfer the data. IoT devices are enlarging the internet connectivity beyond standard devices such as smartphones, laptops, tablets, and desktops. Embedding these devices with technology enable us to communicate and interact over the networks and they can be remotely monitored and controlled.

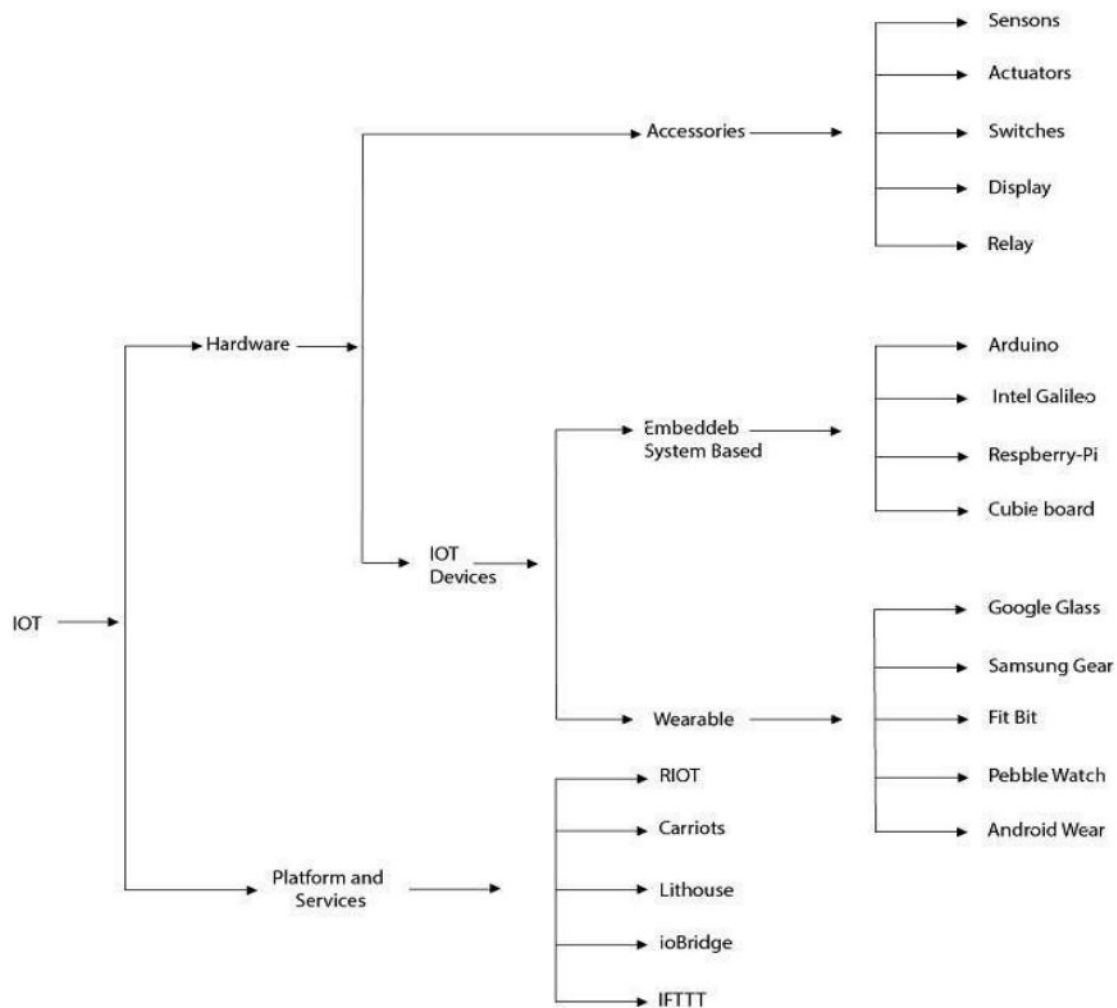


Fig: 3.3 IOT DEVICES

There are large varieties of IoT devices available based on IEEE 802.15.4 standard. These devices range from wireless motes, attachable sensor-boards to interface-

board which are useful for researchers and developers. IoT devices include computer devices, software, wireless sensors, and actuators. These IoT devices are connected over the internet and enabling the data transfer among objects or people automatically without human intervention.

Some of the common and popular IoT devices are given below:



IoT Devices and Technologies

Fig:3.4 IOT Devices and Technologies

3.4 IOT DEVICES AND TECHNOLOGIES

3.4.1 NODE MCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Ten silica Xtensa LX106 core,^[citation needed] widely used in IoT applications (see related projects). NodeMCU started in 2014, when Hong committed the first file of nodemcu firmware to GitHub. Two months later, the project expanded to include an openhardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made in 2015, when Devsaurus ported the u8glib to NodeMCU project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

3.4.2 ESP8266 Arduino Core^[edit]

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WIFI SoC, popularly called the "ESP8266 Core for the Arduino IDE". This has become a leading software development platform for the various ESP8266-based modules and development boards, including Node MCUs.

3.4.3 INTRODUCTION

Node MCU is an open source LUA based firmware developed for ESP8266 Wi-Fi chip. By exploring functionality with ESP8266 chip, Node MCU firmware comes with ESP8266 Development board/kit i.e., Node MCU Development board.



Fig: 3.4.3 NODEMCU

3.4.4 Node MCU Development Board/kit v0.9 (Version1)

Since Node MCU is open-source platform, their hardware design is open for edit/modify/build.

Node MCU Dev Kit/board consist of ESP8266 WIFI enabled chip. The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 WIFI Module.

There is Version2 (V2) available for Node MCU Dev Kit i.e., **Node MCU Development Board v1.0 (Version2)**, which usually comes in black colored PCB.



Fig 3.4.5 NodeMCU Version 1

3.4.5 NodeMCU Development Board/kit v1.0 (Version2)

For more information about NodeMCU Boards available in market refer NodeMCU Development Boards

NodeMCU Dev Kit has Arduino **like** Analog (i.e., A0) and Digital(D0-D8) pins on its board.

It supports serial communication protocols i.e., UART, SPI, I2C etc.

Using such serial protocols, we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc. start with NodeMCU

NodeMCU Development board is featured with wifi capability, analog pin, digital pins and serial communication protocols.

To get start with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

There is online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement.

To know more about how to build custom NodeMCU firmware online and download it refer [Getting started with NodeMCU How to write codes for NodeMCU?](#)

After setting up ESP8266 with Node-MCU firmware, let's see the IDE (Integrated Development Environment) required for development of NodeMCU

3.4.6 NodeMCU with ESPlorer IDE

Lua scripts are generally used to code the NodeMCU. Lua is an open source, lightweight, embeddable scripting language built on top of C programming language.

For more information about how to write Lua script for NodeMCU refer [Getting started with NodeMCU using ESPlorerIDE](#)

3.4.7 NodeMCU with Arduino IDE

Here is another way of developing NodeMCU with a well-known IDE i.e., Arduino IDE. We can also develop applications on NodeMCU using Arduino development environment. This makes easy for Arduino developers than learning new language and IDE for NodeMCU.

For more information about how to write Arduino sketch for NodeMCU refer [Getting started with NodeMCU using Arduino IDE](#)

Difference in using ESPlorer and Arduino IDE

Well, there is a programming language difference we can say while developing application for NodeMCU using ESPlorer IDE and Arduino IDE.

We need to code in C\C++ programming language if we are using Arduino IDE for developing NodeMCU applications and Lua language if we are using ESPlorer IDE. Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executes sequentially. It will not build binary firmware file of code for

NodeMCU to write. It will send Lua script as it is to NodeMCU to get execute. In Arduino IDE when we write and compile code, ESP8266 toolchain in background creates binary firmware file of code we wrote. And when we upload it to NodeMCU then it will flash all NodeMCU firmware with newly generated binary firmware code.

In fact, it writes the complete firmware.

That's the reason why NodeMCU not accept further Lua scripts/code after it is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be no more Lua interpreter and we got error if we try to upload Lua scripts. To again start with Lua script, we need to flash it with NodeMCU firmware.

Since Arduino IDE compile and upload/writes complete firmware, it takes more time than ESPlorer IDE.

3.4.8 Metal detector



Fig: 3.4.8 Metal Detector

- A **metal detector** is an instrument that detects the nearby presence of metal. Metal detectors are useful for finding metal objects on the surface, underground, and under water. The unit itself, consist of a control box, and an adjustable shaft, which holds a pickup coil, which can vary in shape and size. If the pickup coil comes near a piece of metal, the control box will register its presence by a changing tone, a flashing light, and or by a needle moving on an indicator. Usually, the device gives some indication of distance; the closer the metal is, the higher the tone in the earphone or the higher the needle goes. Another common type are stationary "walk through" metal detectors used at access points in prisons, courthouses, airports and psychiatric hospitals to detect concealed metal weapons on a person's body.
- The simplest form of a metal detector consists of an oscillator producing an alternating current that passes through a coil producing an alternating magnetic field. If a piece of electrically conductive metal is close to the coil, eddy currents will be induced (inductive sensor) in the metal, and this produces a magnetic field of its own. If another coil is used to measure the magnetic field (acting as a magnetometer), the change in the magnetic field due to the metallic object can be detected.
- The first industrial metal detectors were developed in the 1960s and were used extensively for mineral prospecting and other industrial applications. Uses include detecting land mines, the detection of weapons such as knives and guns (especially in airport security), geophysical prospecting, archaeology and treasure hunting. Metal detectors are also used to detect foreign bodies in food, and in the construction industry to detect steel reinforcing bars in concrete and pipes and wires buried in walls and floors.

3.4.9 DC MOTORS

A DC motor is used to drive a mechanical load. In this lab, a separately excited

DC generator provides the load. The load on the motor is adjusted by varying the generator field current. By increasing the field current of the DC generator, the load on the DC motor increases and thus the armature current increases. In general, DC motors are characterized by their torque-speed curves as shown in Figure 9.1. Since the measuring equipment for shaft torque is not available in the lab it is necessary to use alternative means of characterizing the DC motor. One alternative is to plot shaft speed versus armature current since torque is directly proportional to the armature current ($T = K_a \phi I_a$) with a constant field current supplied to the motor. Shaft speed is also a function of the field current in a DC motor while maintaining a constant armature voltage ($E_a = K_a \phi$) as field current is directly proportional to the direct axis flux produced in the machine.

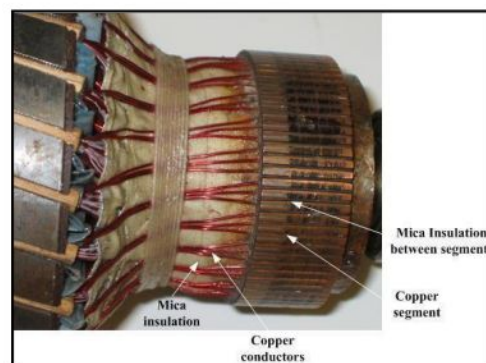


Fig: 3.4.9 DC Motors

3.4.10 BLUETOOTH MODULE

Bluetooth is a wireless protocol utilizing short-range communications technology facilitating data transmission over short distances from fixed and/or mobile devices, creating wireless personal area networks (PANs). The intent behind the development of Bluetooth was the creation of a single digital wireless protocol, capable of connecting multiple devices and overcoming issues arising from synchronization of these devices. Bluetooth uses a very robust radio technology called frequency hopping spread spectrum. It chops up the data being sent and transmits chunks of it on up to 75 different frequencies. In its basic mode, the modulation is Gaussian frequency shift keying (GFSK). It can achieve a gross data rate of 1 Mb/s. Bluetooth provides a way

to connect and exchange information between devices such as mobile phones, telephones, laptops, personal computers, printers, GPS receivers, digital cameras, and video game consoles over a secure, globally unlicensed Industrial, Scientific, and Medical (ISM) 2.4 GHz short-range radio frequency bandwidth.

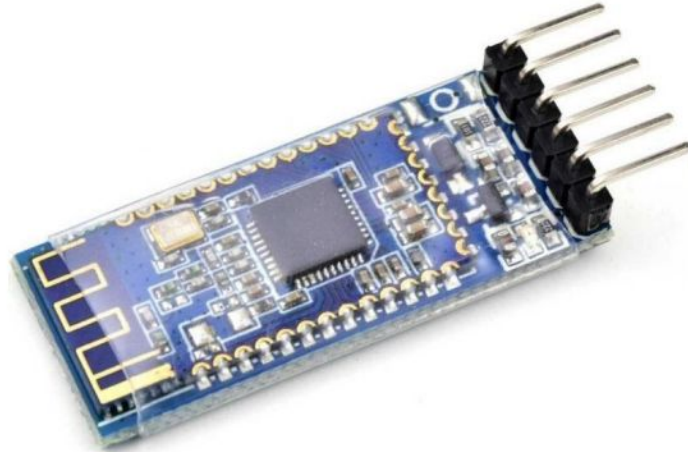


Fig: 3.4.10 Bluetooth Module

The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group (SIG). The Bluetooth SIG consists of companies in the areas of telecommunication, computing, networking, and consumer electronics. Bluetooth is a standard and communications protocol primarily designed for low power consumption, with a short range (power-class-dependent: 1 meter, 10 meters, 100 meters) based on low-cost transceiver microchips in each device.

Bluetooth enables these devices to communicate with each other when they are in range. The devices use a radio communications system, so they do not have to be in line of sight of each other, and can even be in other rooms, as long as the received transmission is powerful enough. Bluetooth device class indicates the type of device and the supported services of which the information is transmitted during the discovery process.

3.4.11 BUZZER

The piezo buzzer produces sound based on reverse of the piezoelectric effect. The generation of pressure variation or strain by the application of electric potential across a piezoelectric material is the underlying principle.



Fig: 3.4.11 Buzzer

These buzzers can be used alert a user of an event corresponding to a switching action, counter signal or sensor input. They are also used in alarm circuits. The buzzer produces a same noisy sound irrespective of the voltage variation applied to it. It consists of piezo crystals between two conductors. When a potential is applied across these crystals, they push on one conductor and pull on the other.

3.4.12 Robotic Arm



Fig: 3.4.12 Robotic Arm

In this proposed project robot was totally depend on the Robotic Arm

Commands of the Robotic Arm:

u(small): Robotic Arm up

U(capital): Robotic Arm down

p(small): open hand

P(capital): close hand

CHAPTER 4

CONSTRUCTION

4.1 Construction

The stator of the dc motor has poles, which are excited by dc current to produce magnetic fields. In the neutral zone, in the middle between the poles, commutating poles are placed to reduce sparking of the commutator.

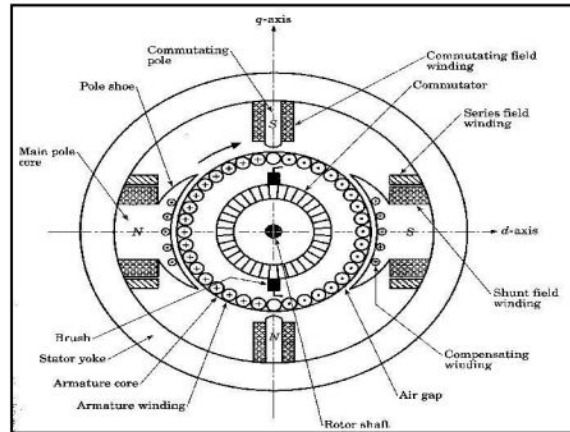


Fig: 4.1 Construction

The commutating poles are supplied by dc current. Compensating windings are mounted on the main poles. These short-circuited windings damp rotor oscillations. The poles are mounted on an iron core that provides a closed magnetic circuit. The motor housing supports the iron core, the brushes and the bearings. The rotor has a ring-shaped laminated iron core with slots. Coils with several turns are placed in the slots. The distance between the two legs of the coil is about 180 electric degrees. The coils are connected in series through the commutator segments. The ends of each coil are connected to a commutator segment. The commutator consists of insulated copper segments mounted on an insulated tube. Two brushes are pressed to the commutator to permit current flow. The brushes are placed in the neutral zone, where the magnetic field is close to zero, to reduce arcing. The rotor has a ring-shaped laminated iron core with slots. The commutator consists of insulated copper segments mounted on an insulated tube. Two brushes are pressed to the commutator to permit current flow. The brushes are placed in the neutral zone, where the magnetic field is close to zero, to reduce arcing. The commutator switches the current from one rotor coil to the adjacent coil, the switching requires the interruption of the coil

current. The sudden interruption of an inductive current generates high voltages. The high voltage produces flashover and arcing between the commutator segment and the brush.

4.2 C. DC Motor Operation

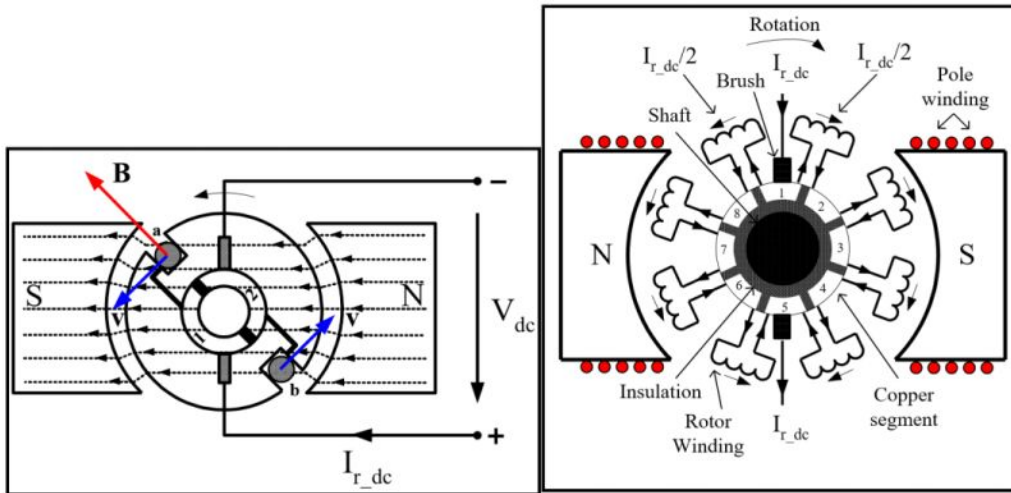
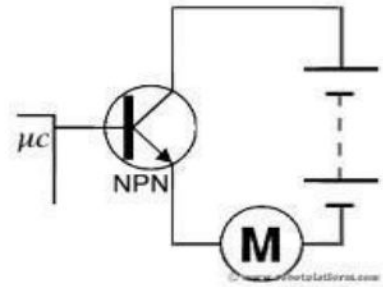


Fig: 4.2. DC Motor Operation

In a dc motor, the stator poles are supplied by dc excitation current, which produces a dc magnetic field. The rotor is supplied by dc current through the brushes, commutator and coils. The interaction of the magnetic field and rotor current generates a force that drives the motor. The magnetic field lines enter into the rotor from the north pole (N) and exit toward the south pole (S). The poles generate a magnetic field that is perpendicular to the current carrying conductors. The interaction between the field and the current produces a Lorentz force; the force is perpendicular to both the magnetic field and conductor. The generated force turns the rotor until the coil reaches the neutral point between the poles. At this point, the magnetic field becomes practically zero together with the force. However, inertia drives the motor beyond the neutral zone where the direction of the magnetic field reverses. To avoid the reversal of the force direction, the commutator changes the current direction, which maintains the counter clockwise rotation. Before reaching the neutral zone, the current enters in segment 1 and exits from segment 2. Therefore, current enters the coil end at slot a and exits from slot b during this stage. After passing the neutral zone, the current enters segment 2 and exits from segment 1, this reverses the current direction through the rotor coil, when the coil passes the neutral zone. The result of this current reversal is the maintenance of the rotation.

Generally, even the simplest robot requires a motor to rotate a wheel or performs particular action. Since motors require more current than the microcontroller pin can typically generate, you need some type of a switch (Transistors, MOSFET, Relay etc.,) which can accept a small current, amplify it and generate a larger current, which further drives a motor. This entire process is done by what is known as a **motor driver**.

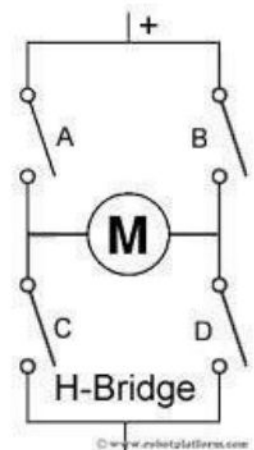


Motor driver is basically a current amplifier which takes a low-current signal from the microcontroller and gives out a proportionally higher current signal which can control and drive a motor. In most cases, a transistor can act as a switch and perform this task which drives the motor in a single direction.

Turning a motor ON and OFF requires only one switch to control a single motor in a single direction. What if you want your motor to reverse its direction? The simple answer is to reverse its polarity. This can be achieved by using four switches that are arranged in an intelligent manner such that the circuit not only drives the motor, but also controls its direction. Out of many, one of the most common and clever design is a H-bridge circuit where transistors are arranged in a shape that resembles the English alphabet "H".

As you can see in the image, the circuit has four switches A, B, C and D. Turning these switches ON and OFF can drive a motor in different ways.

1. Turning on Switches **A** and **D** makes the motor rotate clockwise
2. Turning on Switches **B** and **C** makes the motor rotate anticlockwise
3. Turning on Switches **A** and **B** will stop the motor (Brakes)
4. Turning off all the switches gives the motor a free wheel drive
5. Lastly turning on **A & C** at the same time or **B & D** at the same time shorts your entire circuit. So, do not attempt this.



H-bridges can be built from scratch using relays, mosfets, field effect transistors (FET), bipolar junction transistors (BJT), etc. But if your current requirement is not too high and all

you need is a single package which does the job of driving a small DC motor in two directions, then all you need is a L293D IC. This single inexpensive package can interface not one, but two DC motors. L293, L293B and few other versions also does the same job, but pick the L293D version as this one has an inbuilt fly back diode which protects the driving transistors from voltage spikes that occur when the motor coil is turned off.

4.4 Introduction to L293D IC

L293D IC generally comes as a standard 16-pin DIP (dual in line package). This motor driver IC can simultaneously control two small motors in either direction; forward and reverse with just 4 microcontroller pins (if you do not use enable pins).

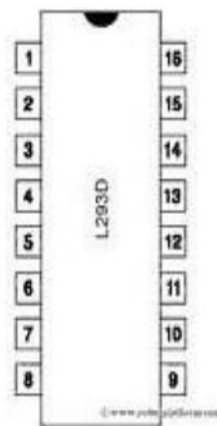


Fig: 4.4 L293D IC

Some of the features (and drawbacks) of this IC are:

1. Output current capability is limited to 600mA per channel with peak output current limited to 1.2A (non-repetitive). This means you cannot drive bigger motors with this IC. However, most small motors used in hobby robotics should work. If you are unsure whether the IC can handle a particular motor, connect the IC to its circuit and run the motor with your finger on the IC. If it gets really hot, then beware... Also note the words "non-repetitive"; if the current output repeatedly reaches 1.2A, it might destroy the drive transistors.
2. Supply voltage can be as large as 36 Volts. This means you do not have to worry much about voltage regulation.

3. The datasheet also mentions an "over temperature protection" built into the IC. This means an internal sensor senses its internal temperature and stops driving the motors if the temperature crosses a set point. Another major feature of **L293D** is its internal clamp diodes. This flyback diode helps protect the driver IC from voltage spikes that occur when the motor coil is turned on and off (mostly when turned off)
5. The logical low in the IC is set to 1.5V. This means the pin is set high only if the voltage across the pin crosses 1.5V which makes it suitable for use in high frequency applications like switching applications (up to 5KHz)
6. Lastly, this integrated circuit not only drives DC motors, but can also be used to drive relay solenoids, stepper motors etc.

4.5 L293D Connections

The circuit shown to the right is the most basic implementation of L293D IC. There are 16 pins sticking out of this IC and we have to understand the functionality of each pin before implementing this in a circuit

1. Pin1 and Pin9 are "Enable" pins. They should be connected to +5V for the drivers to function. If they pulled low (GND), then the outputs will be turned off regardless of the input states, stopping the motors. If you have two spare pins in your microcontroller, connect these pins to the microcontroller, or just connect them to regulated positive 5 Volts.
2. Pin4, Pin5, Pin12 and Pin13 are ground pins which should ideally be connected to microcontroller's ground.
3. Pin2, Pin7, Pin10 and Pin15 are logic input pins. These are control pins which should be connected to microcontroller pins. Pin2 and Pin7 control the first motor (left); Pin10 and Pin15 control the second motor(right).
4. Pin3, Pin6, Pin11, and Pin14 are output pins. Tie Pin3 and Pin6 to the first motor, Pin11 and Pin14 to second motor
5. Pin16 powers the IC and it should be connected to **regulated +5Volts**

6. Pin8 powers the two motors and should be connected to positive lead of a secondary battery. As per the datasheet, supply voltage can be as high as 36 Volts.

4.6 Truth table

I have shown you were to connect the motors, battery and the microcontroller. But how do we control the direction of these motors? Let us take an example:

Suppose you need to control the left motor which is connected to Pin3 (O1) and Pin6 (O2). As mentioned above, we require three pins to control this motor - Pin1 (E1), Pin2 (I1) and Pin7 (I2). Here is the truth table representing the functionality of this motor driver.

Pin 1	Pin 2	Pin 7	Function
High	High	Low	Turn Anti-clockwise (Reverse)
High	Low	High	Turn clockwise (Forward)
High	High	High	Stop
High	Low	Low	Stop
Low	X	X	Stop

High ~+5V, Low ~0V, X=Either high or low (don't care)

In the above truth table, you can observe that if Pin1 (E1) is low then the motor stops, irrespective of the states on Pin2 and Pin7. Hence it is essential to hold E1 high for the driver to function, or simply connect enable pins to positive 5 volts.

With Pin1 high, if Pin2 is set high and Pin7 is pulled low, then current flows from Pin2 to Pin7 driving the motor in anti-clockwise direction? If the states of Pin2 and Pin7 are

flipped, then current flows from Pin7 to Pin2 driving the motor in clockwise direction. The above concept holds true for other side of the IC too. Connect your motor to Pin11 and Pin14; Pin10 and Pin15 are input pins, and Pin9 (E2) enables the driver.

4.7 Motor Driver

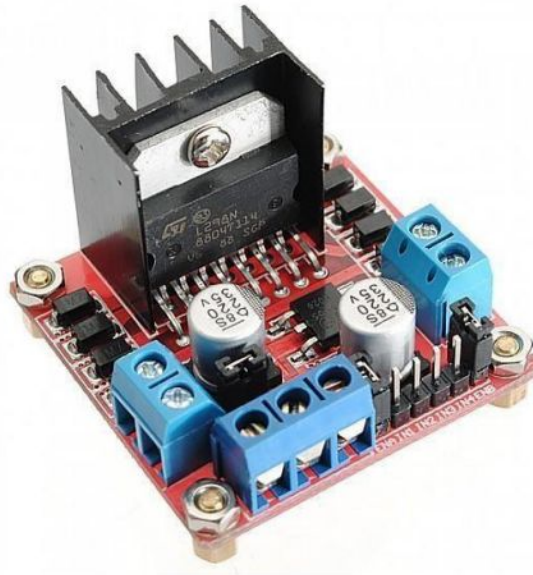


Fig: 4.7 Motor Drive

In the field of Robotics, we have different types of motors such as DC motors, Servo motors, stepper motors etc. Selection of motors depends on your application and there are various ways to operate them. Some motors can be controlled directly by various methods such as connecting the power supply directly to them and make it work. But what if we want to control these motors using other controller devices such as Wireless systems, microcontroller, etc. we need something known as MOTOR DRIVERS

4.7.1 Motor Driver

Motor drivers acts as an interface between the motors and the control circuits. Motor requires high amount of current whereas the controller circuit works on low current signals. So, the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor.

4.7.2 Different motor drivers for Remote Controlled Robots

To control the robot wirelessly through a remote controller we need to interface the motors with the wireless systems such as Bluetooth, 2.4 GHz Rf modules, etc. Here is the detailed comparison between the suggested motor drivers which will help you to select the combination of motor and motor driver

Important Info:

The Fire blade series cannot be interfaced directly with the 2.4 GHz Remote controllers as the signals generated by the 2.4 GHz receiver is not recognized by the Fire blade if interfaced directly. For this we need a board known as Phoenix Board which converts these signals and acts as translator.

Whereas the Viper Board can be interfaced directly with both Microcontrollers as well as 2.4 GHz Remote controllers. Along with it, this board is also a speed controller board that means it allows you to control the speed of motors while the Fire blade series is not having this feature.

4.7.3 Different motor drivers for Autonomous Robots

Autonomous robots are controlled basically by a Micro-controller. As mentioned above the Fire blade and Viper Motor drivers can be used to interface with Microcontrollers. Apart from these there are a few more low current motor drivers such as L293D,

L298N. L293D is used in various shields like Auton Shield, Xbee Shield, Starter Shield as it is best suitable for low current motors such as BO motors commonly used in small autonomous robots.

L298N board is a dual motor driver which can help you interface motors which draw up to 2 Ampere of current such as BO motors and Basic Gear motors. For interfacing motors which draws HIGH current we can choose Fire blade series – Fire blade 10A and Fire blade 30A (Non Speed Controlled) or Viper motor driver (Speed Controller) according to your application.

CHAPTER 5

SOFTWARE SPECIFICATIONS

5.1 ARDUINO UNO AND ITS PROGRAMMING

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand- alone, or they can communicate with software running on your computer. The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

5.2 Overview

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.

This guide covers the Arduino Uno board (Spark fun DEV-09950, \$29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation

This is what the Arduino ide looks like.



Fig: 5.2 overview Arduino

The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins

and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors.

The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you

push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

5.3 Arduino Software (IDE)

- Writing Sketches
- File
- Edit
- Sketch
- Tools
- Help
- Sketchbook
- Tabs, Multiple Files, and Compilation
- Uploading
- Libraries
- Third-party Hardware
- Serial Monitor

- Preferences
- Language Support
- Boards

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

5.3.1 Writing Sketches

Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension. `.ino`. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension. `.pde`. It is possible to open these files

with version 1.0, you will be prompted to save the sketch with the. `ino` extension on save.



Verify

Checks your code for errors compiling it.



Upload

Compiles your code and uploads it to the configured board.

See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"



New

Creates a new sketch.



Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.



Save

Saves your sketch.



Serial Monitor

Opens the serial monitor.

Additional commands are found within the five menus: **File**, **Edit**, **Sketch**, **Tools**, **Help**. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

5.3.2 File

New Creates a new instance of the editor, with the bare minimum structure of a sketch already in place. Open Allows to load a sketch file browsing through the computer drives and folders. Open Recent Provides a short list of the most recent sketches, ready to be opened. Sketch book Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance. Examples Any example provided by the Arduino Software (IDE) or library

shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library. Close Closes the instance of the Arduino Software from which it is clicked. Save Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as..." window. Save as... Allows to save the current sketch with a different name. Page Setup It shows the Page Setup window for printing. Print Sends the current sketch to the printer according to the settings defined in Page Setup. Preferences Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface. Quit Closes all IDE windows. The same sketches open when quit was chosen will be automatically reopened the next time you start the IDE.

5.3.3 Edit

Undo/Redo Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo. Cut Removes the selected text from the editor and places it into the clipboard. Copy Duplicates the selected text in the editor and places it into the clipboard. Copy for Forum Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax colouring. Copy as HTML Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages. Paste Puts the contents of the clipboard at the cursor position, in the editor. Select All Selects and highlights the whole content of the editor. Comment/Uncomment Puts or removes the // comment marker at the beginning of each selected line. Increase/Decrease Indent Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning. Find Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options. Find Next Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position. Find Previous Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

5.3.4 Sketch

Verify/Compile

Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area. Upload Compiles and loads the binary file onto the configured board through the configured Port. Upload Using Programmer This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed. Export Compiled Binary Saves a .hex file that may be kept as archive or sent to the board using other tools.

Show Sketch Folder Opens the current sketch folder. Include Library

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files. Add File... Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right-side of the toolbar.

5.3.5 Tools

AutoFormat This formats your code nicely: i.e., indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more. Archive Sketch Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch. Fix Encoding & Reload Fixes possible discrepancies between the editor char map encoding and other operating systems char maps. Serial Monitor Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening. Board Select the board that you're using. See below for descriptions of the various boards. Port

This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu. Programmer For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this. Burn Bootloader The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new AT mega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

5.3.6 Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

Find in Reference

This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

5.3.7Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the. pde extension. You may still open. pde named files in version 1.0 and later, the software will automatically rename the extension to. ino.

5.3.8 Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

5.3.9 Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the sports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx, /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to

the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e., when the board resets).

5.3.10 Libraries

Libraries provide extra functionality for use in sketches, e.g., working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more `#include` statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its `#include` statements from the top of your code. There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a thirdparty library.

To write your own library, see this tutorial.

5.3.11 Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "Arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory. For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

5.3.12 Serial Monitor

This displays serial sent from the Arduino or Genuino board over USB or serial

connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial. Begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

5.3.13 Preferences

Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

5.3.14 Language Support

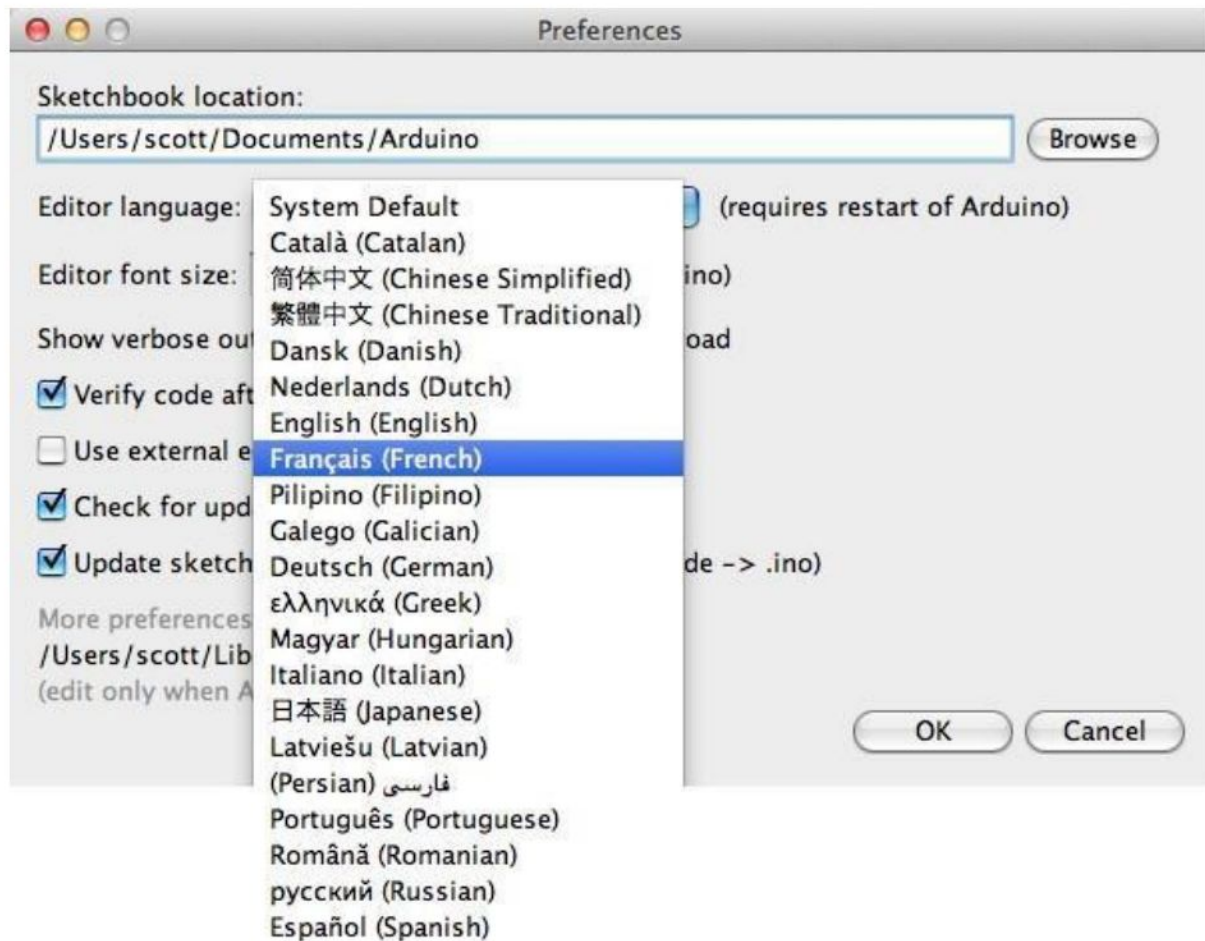


Fig: 5.3.14 Language Support

Since version 1.0.1, the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If you would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English. You can operating system by selecting System Default from the Editor Language drop-

down. This setting will take effect when you restart the Arduino Software (IDE). Similarly, after changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

5.3.15 Boards

The board selection has two effects: it sets the parameters (e.g., CPU speed and baud rate) used when compiling and uploading sketches; and sets the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection, you'll want to check it before burning the bootloader. You can find a comparison table between the various boards [here](#).

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

Arduino Uno An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM. Genuino Uno an ATmega328P running at 16 MHz with autoreset, 6 Analog In, 14 Digital I/O and 6 PWM. Arduino Diecimila or w/ ATmega168 An ATmega168 running at 16 MHz with auto-reset. Arduino Nano w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset. Has eight analog inputs. GenuinoMega2560 An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM. Arduino Mega an ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM. Arduino Mega ADK An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM. Arduino Leonardo An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM. Genuino Micro an ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM. Arduino Esplora An ATmega32u4 running at 16 MHz with auto-reset.

Arduino Mini w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 PWM. Arduino Ethernet Equivalent to Arduino UNO with an Ethernet shield: An ATmega328P running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM. Arduino Fio An ATmega328P running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328P, 6 Analog In, 14 Digital I/O and 6 PWM. Arduino BT w/ ATmega328P ATmega328P running at 16 MHz the bootloader burned (4 KB) includes codes to initialize the onboard Bluetooth module, 6 Analog In, 14 Digital I/O and 6 PWM. LilyPad Arduino USB An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM. LilyPad Arduino An ATmega168 or ATmega132 running at 8 MHz with autoreset, 6 Analog In, 14 Digital I/O and 6 PWM.

Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset. Equivalent to Arduino Duemilanove or Nano w/ ATmega328P; 6 Analog In, 14 Digital I/O and 6 PWM. Arduino NG or older w/ ATmega168 An ATmega168 running at 16 MHz without auto-reset. Compilation and upload are equivalent to Arduino Diecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset); 6 Analog In, 14 Digital I/O and 6 PWM. Arduino Robot Control an ATmega328P running at 16 MHz with auto-reset. Arduino Robot Motor an ATmega328P running at 16 MHz with auto-reset. Arduino Gemma An ATtiny85 running at 8 MHz with auto reset, 1 Analog In, 3 Digital I/O and 2 PWM. For instructions on installing support for other boards, see third-party hardware above.

The text of the Arduino getting started guide is licensed under a Creative Commons Attribution-Share Alike 3.0 License. Code samples in the guide are released into the public domain.

CHAPTER 6

CONCLUSION

6.1 Conclusion

This work proposed a design of a robot for bomb detection. It can detect a signal which is transmitted and according to that robot can follow the forward, backward, left turn, right turn movements. Metal/Bomb detector can detect the metals and alert with beep sound to notify the Metal/Bomb. In this manner this project plays a crucial role in Military as well as in police department.

The advanced Bomb Disposal Robot has been designed in such a way that it is capable of detecting and diffusing the bomb and even marking their exact location. It performs three different tasks like detecting, spying, diffusing.

FUTURE SCOPE

6.2 Future scope

This spying robot can be modified and made it for prolonged ranged and can be make it more useful by consuming more operational procedures and modules like Wi-Fi module, raspberry pi. Future scope of this robot is very efficient it may have gas sensors to detect the harmful or hazardous gases in the surroundings. It can also be used as bomb diffuser and bomb disposal team can also use this type of robot in many ways and reduces the risk factor of human loss. Further, a terminating framework can be set on the robot, to fire any foe when he is spotted.

The innovation can be enhanced further by offering directions to accepting circuit and control it by utilizing satellites correspondence.

REFERENCES

6.3 References

- [1] Chaitrali Jadhav, Shamli gibile, Snehal Gaikwad, Neelam Dave, "military spying and bomb disposal robot using IOT". International research journal of engineering and technology ISSN (e): 2250-3021, ISSN (p): 2278-8719 PP 09-11-2018.
- [2] Senthamizh, Subbu Lakshmi, Shubhashree, Prof. M. Priyadarshini, "Advanced Military Spying and Bomb Disposal Robot". IOSR Journal of Engineering 05 Issue: 04 | Apr-2018.
- [3] Sarmad Hameed, Muhammad Hamza Khan, Naqi Jafri, Adeel Azfar Khan, Muhammad Bilal Taak, "Military Spying Robot". International Journal of Innovative Technology and Exploring Engineering ISSN: 2278-3075, Volume-8, Issue7C2, May 2019.
- [4] Priyanka Yadav, Leena Chaudhari, Swati Gawhale, "War Field Spying Robot with Wireless Night Vision Camera", International Journal for Research in Applied Science & Engineering Technology ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887 Volume 5 Issue XII December 2017.
- [5] Nandagopal Rathod, Puneeth. G, P. Brahmaiah, Sandur Gangadhar, Gururaja Sharma, "Advanced Spying and Bomb Disposal Robot", International Journal of Scientific & Engineering Research Volume 11, Issue 6, June-2020 ISSN 2229- 5518.

[6] Patil Shailendra D., Wagh Shubham B., Tirmakhe Dipak K., Gavate Swati P., Prof. Dhakane V.N, "Android Based Advanced Military Spying and Bomb Disposal Robot", Resincap Journal of Science and Engineering Volume 3, Issue 11 November 2019 ISSN: 2456-9976.

[7] Rajat Desai, Pankaj Naik, Arun chawadi, Virendra Naik, "Advanced Military Spy Robot and Bomb Disposal", International Journal of Science Technology & Engineering | Volume 4 | Issue 9 | March 2018 ISSN (online): 2349-784X.