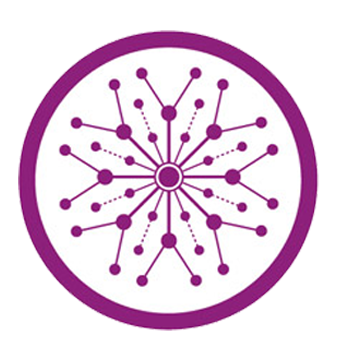
Multifunction Robot

Final Year Project

SP 2019-2023

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2023

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type (Multifunction Robot) | | | [ ✓ ] **D**evelopment [ ] **R**esearch [ ] **R**&**D** | | |
| Area of specialization | | |  | | |
| FYP ID(FYP-BCSM-S22-024) | | |  | | |
| **Project Group Members** | | | | | |
| Sr.# | Reg. # | Student Name | | Email ID | \*Signature |
| (i) |  | YASIR RAMZAN | | Bcsm-s19-056@superior.edu.pk |  |
| (ii) |  | UMER IDREES | | Bcsm-s19-020@superior.edu.pk |  |
| (iii) |  | FAKHIR ALI | | Bcsm-f18-457@superior.edu.pk |  |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

# Plagiarism Free Certificate

This is to certify that, I \_\_\_\_\_\_\_\_ S/D of Muhammad Siddique, group leader of FYP under registration no \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_at Software Engineering Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date:  Name of Group Leader: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Supervisor: Dr. ABC Co-Supervisor: Mr. XYZ

Designation: Lecturer Designation: Associate Professor

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HoD: Dr. Arfan Jaffar

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Report**

**[Multifunction Robot]**

**Change Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author(s)** | **Version** | **Date** | **Notes** | **Supervisor’s Signature** |
|  | 1.0 |  | <Original Draft> |  |
|  |  |  | <Changes Based on Feedback from Supervisor> |  |
|  |  |  | <Changes Based on Feedback From Faculty> |  |
|  |  |  | <Added Project Plan> |  |
|  |  |  | <Changes Based on Feedback from Supervisor> |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**APPROVAL**

|  |  |
| --- | --- |
| **Project Supervisor** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

|  |  |
| --- | --- |
| **Project Manager** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| **Head of the Department** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Dedication

I dedicated this project to my parents who have been a great source of inspiration and support this project is also dedicated to Senior Lecturer Ma’am Humaira Muqaddas who encourage us to build our motivation.

# 

# Executive Summary

**Robot**, any automatically operated  [machine](https://www.britannica.com/technology/machine) that replaces human effort, though it may not resemble human beings in appearance or perform functions in a humanlike manner. The Android controlled temperature sensing Robot Car. The working is based on Android OS.  The device can be controlled by any smart device with android. The movement aided by sensors enables the robot to avoid obstacles faster and quickly finding the direction of the object. It is used to connect the mobile camera to view in our pc to fixing in the car. It also used to view the location of the car. The Robotic car senses the temperature to view by the mobile app. All the controls of the vehicle on the app on that device.

# Table of Contents

Contents

[Chapter 1 11](#_Toc124248573)

[Introduction 11](#_Toc124248574)

[1.1. Background 12](#_Toc124248575)

[1.2. Motivations and Challenges 13](#_Toc124248576)

[1.3. Goals and Objectives 13](#_Toc124248577)

[1.4. Literature Review/Existing Solutions 14](#_Toc124248578)

[1.5. Proposed Solution 14](#_Toc124248579)

[1.6. Project Plan 14](#_Toc124248580)

[1.6.1. Work Breakdown Structure 15](#_Toc124248581)

[1.6.2. Roles & Responsibility Matrix 15](#_Toc124248582)

[1.6.3. Gantt Chart 16](#_Toc124248583)

[1.7. Empathy Map 17](#_Toc124248584)

[Chapter 2 18](#_Toc124248585)

[Software Requirement Specifications 18](#_Toc124248586)

[2.1. Introduction 18](#_Toc124248588)

[2.1.1. Purpose 19](#_Toc124248589)

[2.1.2. Document Conventions 19](#_Toc124248591)

[2.1.3. Intended Audience and Reading Suggestions 19](#_Toc124248592)

[2.1.4. Product Scope 20](#_Toc124248593)

[2.1.5. References 20](#_Toc124248594)

[2.2. Overall Description 21](#_Toc124248595)

[2.2.1. Product Perspective 21](#_Toc124248596)

[2.2.2. User Classes and Characteristics 22](#_Toc124248597)

[2.2.3. Operating Environment 22](#_Toc124248598)

[2.2.4. Design and Implementation Constraints 23](#_Toc124248599)

[2.3. External Interface Requirements 23](#_Toc124248600)

[2.3.1. Hardware Interfaces 23](#_Toc124248601)

[2.3.2. Software Interfaces 23](#_Toc124248602)

[2.3.3. Communications Interfaces 23](#_Toc124248603)

[2.4. System Features 23](#_Toc124248604)

[2.4.1. System Feature 1 23](#_Toc124248605)

[IOT Based Mobile Application User 23](#_Toc124248606)

[2.4.2. System Feature 2 23](#_Toc124248607)

[2.4.3. System Feature 3 24](#_Toc124248608)

[2.5. Other Nonfunctional Requirements 24](#_Toc124248609)

[2.5.1. Performance Requirements 24](#_Toc124248610)

[2.5.2. Safety Requirements 24](#_Toc124248611)

[2.5.3. Security Requirements 24](#_Toc124248612)

[2.5.4. Usability Requirements 25](#_Toc124248613)

[2.5.5. Reliability Requirements 25](#_Toc124248614)

[2.5.6. Maintainability/Supportability Requirements 26](#_Toc124248615)

[2.5.7. Portability Requirements 26](#_Toc124248616)

[2.5.8. Efficiency Requirements 26](#_Toc124248617)

[2.6. Domain Requirements 26](#_Toc124248618)

[Chapter 3 27](#_Toc124248619)

[Use Case Analysis 27](#_Toc124248620)

[3.1. Use Case Model 28](#_Toc124248621)

[3.2. Use Case Descriptions 29](#_Toc124248622)

[Chapter 4 30](#_Toc124248623)

[System Design 30](#_Toc124248624)

[4.1. Architecture Diagram 30](#_Toc124248625)

[4.2. Component Diagram 32](#_Toc124248626)

[Chapter 5 36](#_Toc124248627)

[Implementation 36](#_Toc124248628)

[5.1. Important Flow Control/Pseudo codes 36](#_Toc124248629)

[5.2. Components, Libraries, Web Services, and stubs 36](#_Toc124248630)

[5.3. Deployment Environment 38](#_Toc124248631)

[5.4. Tools and Techniques 38](#_Toc124248632)

[5.5. Best Practices / Coding Standards 39](#_Toc124248633)

[5.6. Version Control 40](#_Toc124248634)

[Chapter 6 41](#_Toc124248635)

[Testing and Evaluation 41](#_Toc124248636)

[6.1. Use Case Testing 41](#_Toc124248637)

[6.2. Equivalence partitioning 41](#_Toc124248638)

[6.3. Boundary value analysis 41](#_Toc124248639)

[6.4. Data flow testing 41](#_Toc124248640)

[6.5. Unit testing 42](#_Toc124248641)

[6.6. Integration testing 42](#_Toc124248642)

[6.7. Performance testing 42](#_Toc124248643)

[6.8. Stress Testing 42](#_Toc124248644)

[Chapter 7 43](#_Toc124248645)

[Summary, Conclusion and Future Enhancements 43](#_Toc124248646)

[7.1. Project Summary 43](#_Toc124248647)

[7.2. Achievements and Improvements 43](#_Toc124248648)

[7.3. Future Enhancements/Recommendations 44](#_Toc124248649)

[Appendices 45](#_Toc124248650)

[Code: 45](#_Toc124248651)

[Reference and Bibliography 54](#_Toc124248653)

# Chapter 1

# Introduction

**Chapter 1:** Introduction

Robotics is part of Todays communication. In today’s world Robotics is fast growing and interesting field. It is simplest way for latest technology modification. Now a day’s communication is part of advancement of technology, so I decided to work on Robotics field, and design something which will make human life simpler in today aspect. An autonomous robot is a robot that can move on its own in an unknown and unstructured environment. An autonomous robot is equipped with software intelligence to sense its environment, detect obstacles in its path and move around an unknown environment overcoming the obstacles. There are many robotic designs that are employed in designing autonomous robots. These designs are usually developed considering the physical environment in which the robot must be deployed. There are autonomous robots like snake robots, walking robots, autonomous drones and autonomous robotic cars or rovers. This robot has sufficient intelligence to cover the maximum area of provided space. It has an infrared sensor which is used to sense the obstacles coming in between the path of robot. It will move in a particular direction and avoid the obstacle which is coming in its path. And it also having other feature like line following, robot follow a line to reach its destination and the other one is it control through mobile application under certain condition. It has many uses in hospitals, hotels, grocery stores and shopping malls. For different functions in different environment for different purposes.

## Background

One of the first instances of a mechanical device built to regularly carry out a particular physical task occurred around 3000 B.C.:  Egyptian water clocks used human figurines to strike the hour bells.  In 400 B.C., Archytus of Tarentum, inventor of the pulley and the screw, also invented a wooden pigeon that could fly.  Hydraulically operated statues that could speak, gesture, and prophecy were commonly constructed in Hellenic Egypt during the second century B.C.  
In the first century A.D., Petronius Arbiter made a doll that could move like a human being.  Giovanni Torriano created a wooden robot that could fetch the emperor’s daily bread from the store in 1557.  Robotic inventions reached a relative peak (before the 20th century) in the 1700s; countless ingenius, yet impractical, automata (i.e., robots) were created during this period.  The 19th century was also filled with new robotic creations, such as a talking doll by Edison and a steam-powered robot by Canadians.  Although these inventions throughout history may have planted the first seeds of inspiration for the modern robot, the scientific progress made in the 20th century in the field of robotics surpasses previous advancements a thousand-fold. The earliest robots as we know them were created in the early 1950s by George C. Devol, an inventor from Louisville, Kentucky.  He invented and patented a reprogrammable manipulator called "Unimate," from "Universal Automation."  For the next decade, he attempted to sell his product in the industry, but failed.  In the late 1960s, businessman/engineer Joseph Engleberger acquired Devol's robot patent and was able to modify it into an industrial robot and form a company called Unimation to produce and market the robots.  For his efforts and successes, Engleberger is known in the industry as "the Father of Robotics”. "Academia also made much progress in the creation of new robots.  In 1958 at the Stanford Research Institute, Charles Rosen led a research team in developing a robot called "Shakey."  Shakey was far more advanced than the original Unimate, which was designed for specialized, industrial applications.  Shakey could wheel around the room, observe the scene with his television "eyes," move across unfamiliar surroundings, and to a certain degree, respond to his environment.  He was given his name because of his wobbly and clattering movements.

## Motivations and Challenges

Robots are widely used in such industries as automobile manufacture to perform simple repetitive tasks, and in industries where work must be performed in [environments](https://www.merriam-webster.com/dictionary/environments) hazardous to humans. Many aspects of robotics involve [artificial intelligence](https://www.britannica.com/technology/artificial-intelligence), robots may be equipped with the equivalent of human senses such as vision, touch, and the ability to sense temperature. Some are even capable of simple [decision making](https://www.britannica.com/topic/decision-making), and current robotics research is geared toward devising robots with a degree of self-sufficiency that will permit mobility and decision-making in an unstructured [environment](https://www.merriam-webster.com/dictionary/environment). Today’s industrial robots do not resemble human beings, a [robot](https://www.britannica.com/technology/robot-technology) in human form is called an android.

In the Present Era from past two-year Covid ‘19’ spread all over the world and world face many challenges which disturb the individual life, Covid ‘19’ causes huge number of deaths all around the world. Why this was going to happen because we are not ready for such type of challenge. Covid ‘19’ motivative me to do something for human beings as a student in final year of computer science I used to develop a prototype of a robot which used for such type of circumstances.

Many challenges which we are facing and shall be face in the development of this project prototype. As a student are:

* Manufacturing procedures
* Facilitating Human-robot collaboration
* Creating Better Power Sources
* Mapping Environment
* Developing Reliable Artificial Intelligence
* Building Multifunctional Robot

## Goals and Objectives

The goals and objectives are as follows:

* Develop Prototype of a robot has Following Feature
* Automatically Obstacles Avoidance
* Line Following Function
* Control Through a Mobile App
* Interference of Sensor (uses for obstacle avoidance and line following) and Motors with controller.
* Mobile app Development for Robot control.
* Interface Software and Hardware Part.

## Literature Review/Existing Solutions

The search includes coming up with possible improvements in robotics which includes the multifunctional robot. Previous systems of multifunctional robots are either not as good as they were thought to be, or they are very expensive to make or buy. We are making this machine which will be easy to use, user friendly, more precise, with different functions and cost effective. As the focus of this study is on obstacle avoidance, line following and IOT Based Control robots, their function, their cost and results, we did a brief overview of past work in this line of work.

## Proposed Solution

Our proposed prototype is a multitasking robot which performs different sets of functions under the different condition with different functionalities. The goal/Reason behind to develop this prototype is to help people under certain challenging conditions for the improvement of Society and human beings. Our proposed solution helps peoples to do their daily tasks. Our proposed solution is used in hotels, hospitals, grocery stores, shopping malls and warehouses.

## Project Plan

The project plan will be based on a work breakdown structure, which will simplify the complex parts of this project, and the roles and responsibilities of each member will be defined in the roles and responsibility matrix.

## Work Breakdown Structure

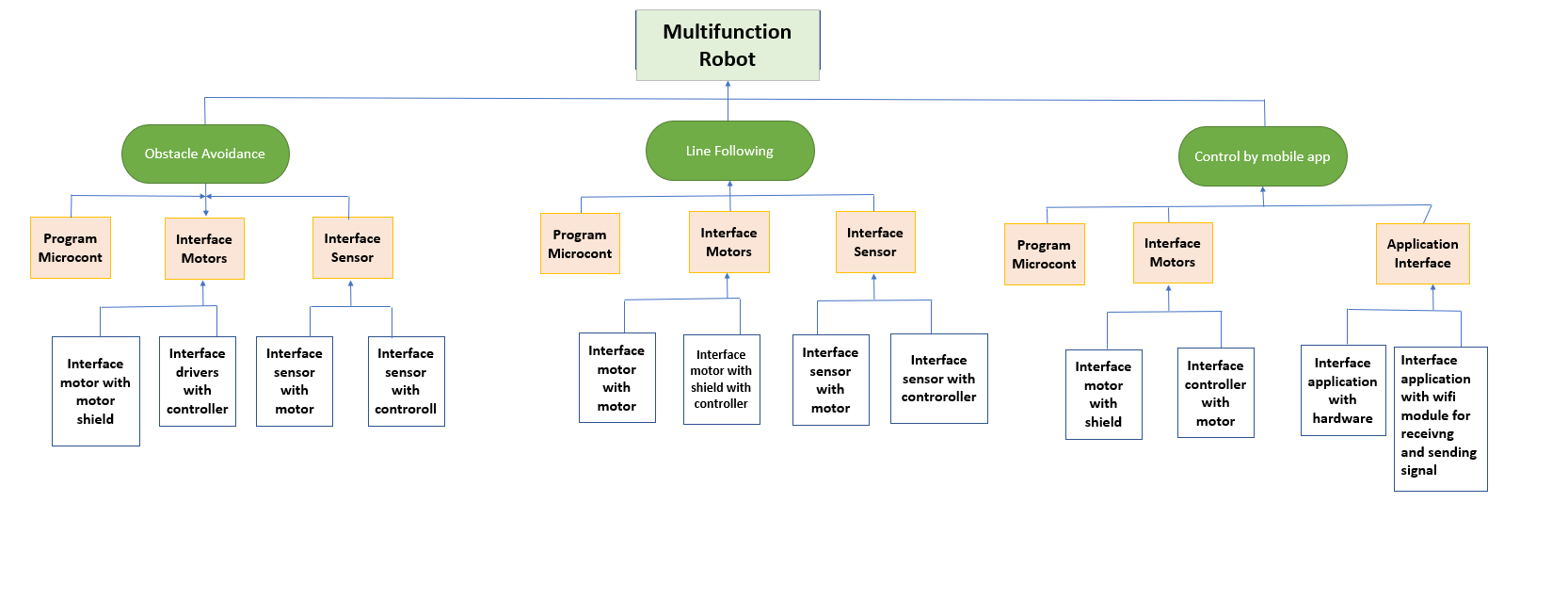


Figure 0‑1 WBS

## Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **WBS #** | **WBS Deliverable** | **Activity #** | **Duration**  **(# of Days)** | **Responsible Team Member(s) & Role(s)** |
|  | Documentation |  | 7 days | YASIR, UMER, FAKHIR |
|  | Hardware Interfacing |  | 2 months | YASIR, UMER |
|  | Software Coding |  | 7 days | YASIR |
|  | Interfacing of software with hardware |  | 3 months | YASIR, UMER, FAKHIR |

## Gantt Chart

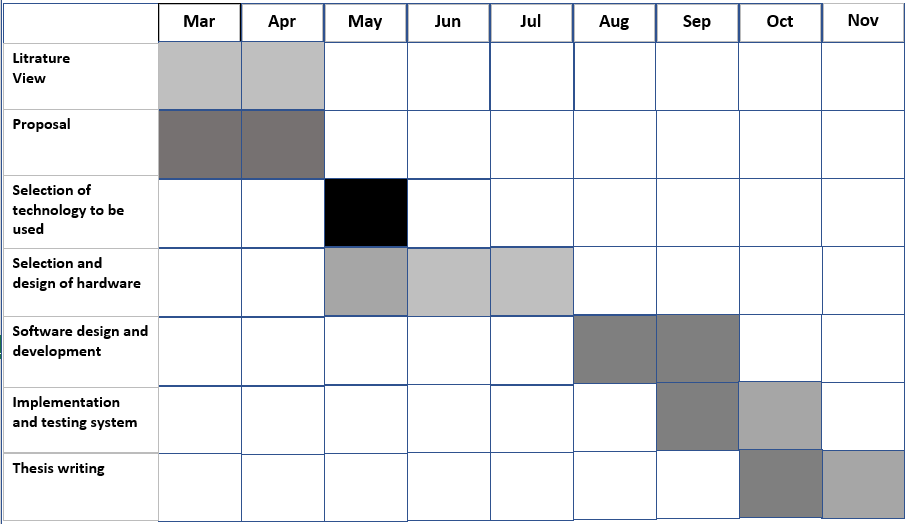
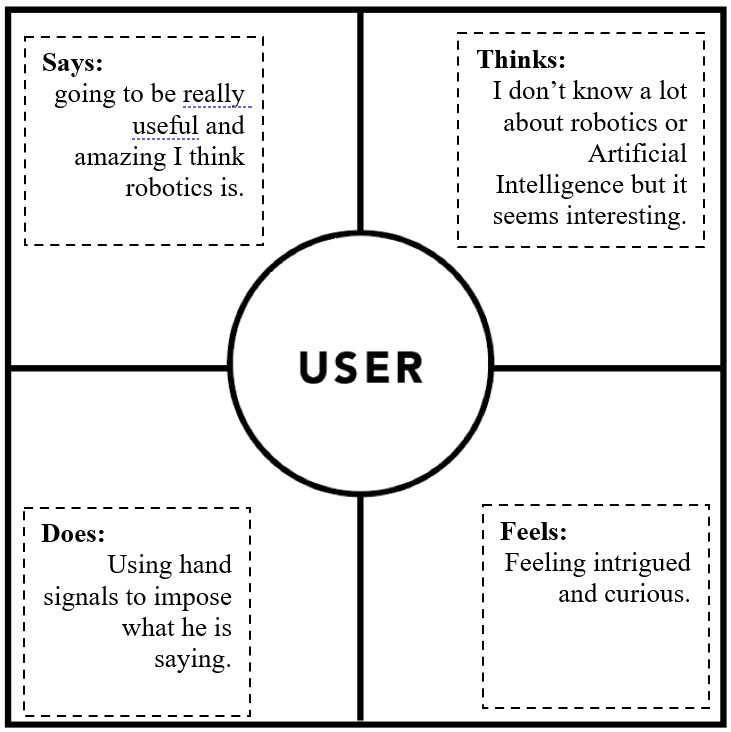


Figure 0‑2 TABLE OF GANTT CHART

## Empathy Map



# Chapter 2

# Software Requirement Specifications

**Chapter 2:** Software Requirement Specifications



## Introduction

Robotics is part of Todays communication. In today’s world Robotics is a fast growing and interesting field. It is the simplest way for latest technology modification. Now a day’s communication is part of the advancement of technology, so I decided to work on Robotics field, and design something which will make human life simpler in today aspect. An autonomous robot is a robot that can move on its own in an unknown and unstructured environment. An autonomous robot is equipped with software intelligence to sense its environment, detect obstacles in its path and move around an unknown environment overcoming the obstacles. There are many robotic designs that are employed in designing autonomous robots. These designs are usually developed considering the physical environment in which the robot must be deployed. There are autonomous robots like snake robots, walking robots, autonomous drones and autonomous robotic cars or rovers. This robot has sufficient intelligence to cover the maximum area of provided space. It has an infrared sensor which is used to sense the obstacles coming in between the path of robot. It will move in a particular direction and avoid the obstacle which is coming in its path. And it’s also having other feature like line following, robot follow a line to reach its destination and the other one is it control through mobile application under certain condition. It has many uses in hospitals, hotels, grocery stores and shopping mall. For different function in different environment for different purposes.

## Purpose

## Robots are made for Doing Repeated Task and For the Development and growth of an organization. In this project we are going to make a multifunctional robot car which perform different function under different environment for different purpose. In the advancement of our project our ideas are very beneficial for the society and environment. Main purpose of making this robot prototype is handle the un- circumstances situations like Covid 19 now days. It uses as Nurse in hospitals, waiter as in hotels, worker as warehouses and delivery car at small town or 1-2km of surrounding

## Document Conventions

* C Programming
* HTML Hyper Text Markup Language
* **DB** Data Base and Management
* **IP** Internet Protocol
* **API** Application Programming Interface
* **CSS** Cascading Style Sheets
* Circuit Designing
* **EMs** Energy System Management
* Resources Management

## Intended Audience and Reading Suggestions

This project is a prototype for Robotics Process Automation which will automating the ongoing processes in Different Sector. This requires the people who work on such processes manually by handling day to day tasks within the organization.

Such people include

• Records staff

• Accounts staff

• Reception staff

## Product Scope

Artificial intelligence (AI) and robotics are digital technologies that will have significant impact on the development of humanity soon. Our Product Uses in Different Sectors of society and it has huge Scope.

Our Product Scope in Following Fields:

• Hotels

• Warehouses

• Delivery Agencies

• Hospitals

• Shopping Malls

• Grocery Stores

## References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley, USA (1974)

2. Ben-Ari, M.: Principles of Concurrent and Distributed Programming, 2nd end. Addison-Wesley, USA (2006)

3. Dudek, G., Jenkin, M.: Computational Principles of Mobile Robotics, 2nd edn. Cambridge University Press, UK (2010)

4. Kumar, D.: Learning Computing with Robots. Lulu (2011). Download from http://calicoproject. org/Learning\_Computing\_With\_Robots

5. Shin, J., Siegwart, R., Magnenat, S.: Visual programming language for Thymio II robot. In: Proc. of the 2014 Conference on Interaction Design and Children (IDC) (2014)

6. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots, 2nd edn. MIT Press, USA (2011)

7. Trobaugh, J.J., Lowe, M.: Winning LEGO MINDSTORMS Programming. Apress (2012)

8. Winfield, A.: Robotics: A Very Short Introduction. Oxford University Press, USA (2012)

## Overall Description

## Product Perspective

Like automation, robotics has multiple definitions associated with it, which can lead to misunderstandings when it comes to statistics. An Industrial Robot, as defined by ISO 8373:2012 is, ‘an automatically controlled, reprogrammable, multipurpose manipulator programmable in 3 or more axes, which can be either fixed in place or mobile for use in industrial automation applications.’ Robots vary in their autonomy, with some able to work completely independently from human intervention and others requiring continuous instruction to complete the task.

**Industrial robots** are ‘automatically controlled, reprogrammable multipurpose manipulator programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications’, as defined by the IFR (International Federation of Robotics, and International Organization for Standardization). Service robots can perform ‘useful tasks for humans or equipment excluding industrial automation applications’, according to the International Organizations for Standardization. They require a degree of autonomy and generally include systems based on some degree of human-robot interaction to perform tasks in both professional and personal uses. Service robots can be applied in many different application areas and have some ability to sense their surroundings to complete their tasks.

**Cabot’s** are ‘collaborative robots’, designed for ‘direct physical interaction with a human operator, within a shared workspace.’ They are produced to be safe for use in environments where humans are present, and therefore have certain constraints built into their design. Cabot’s open an interactive workspace, which is very suited to collaboration and can be arranged in a very adaptable manner due to humans being able to interact with the robots.

**Autonomous Mobile Robots (AMRs)** can make their own decisions based on their surroundings using machine learning and artificial intelligence. They can act independently without human intervention and are mobile in terms of being able to move around in a controlled manner.

Robotic Process Automation (RPA) is ‘software technology that’s easy for anyone to use to automate digital tasks.’ As a software tool it can partially or fully automate repetitive or rule-based human activities such as data entry or simple customer service queries.

**Robotics as a Service** has the potential to be significant in terms of easy adoption, whereby robotics equipment and maintenance services are leased to companies for use. This business model provides the opportunity for scaling up and down dependent on requirements and allows versatility to smaller companies who cannot feasibly invest in large, expensive equipment. Loughborough University is embarking on a three-year initiative to improve the ease with which robotics are hired and deployed.

## User Classes and Characteristics

Robots are now being used in many applications including: welding, sealing, and painting, material handling, assembly, and inspection; and in non-automotive industries such as electronics, consumer products, pharmaceutical, and service

• User Classes in Our Product/Project

• Our Product is multi-functional, so it has multiple user

**IOT Based Mobile Application User**

* Users use mobile application to control the robot for different functional
* Delivery Inside Warehouse
* Transformation of Material in shopping malls and Grocery Stores

**Line Following Based User**

* Users use it for repeated task on same route
* Users use it for transfer of material from one part to another of a factory on a specific route (like Breakfast & Dinner delivery in hotels)

**Automatically obstacle Avoidance User**

• Users use it for both things which explains above

## Operating Environment

Based on the old Product, Operating Environment

• Controller as Mobile Application

• Different Industries Sector

• Hospitals, Malls, Grocery Store, and factory Outlets Etc.

## Design and Implementation Constraints

A robotic constraint is a restriction placed on a robotic system that narrows its achievable motion possibilities. Constraints can include both the mechanical constraints of the system, such as a joint that is at its limit, and the constraints that are applied by the user, such as the desired position of the tool. While the term “constraint” often has a negative connotation, robotic constraints are necessary and allow the user to determine the motion of the system. For instance, when following a toolpath, respecting the surface normal vector and any desired execution offsets relative to that vector is usually critical to the robot task, so it’s necessary to constrain the robot link that is following the toolpath to match those “normal” vectors.

## External Interface Requirements

## Hardware Interfaces

• Interface of Sensors with Micro-Controller

• Interface of motors with micro-controller

• Power connection with sensor, motors, and Controller

## Software Interfaces

• Interface of Mobile Application with WIFI/Bluetooth Module

• Establish Connection for Sending Signals

## Communications Interfaces

• Bluetooth uses of Communication

• WIFI uses for Communication

## System Features

## System Feature 1

## IOT Based Mobile Application User

* Users use mobile application to control the robot for different functional
* Delivery Inside Warehouse
* Transformation of Material in shopping malls and Grocery Stores

## System Feature 2

**Line Following Based User**

* Users use it for repeated task on same route
* Users use it for transfer of material from one part to another of a factory on a specific route (like Breakfast & Dinner delivery in hotels)

## System Feature 3

**Automatically obstacle Avoidance User**

* Users use it for both things which are in Line Following as well as In IOT control
* Users use it in case of Emergency
* Users use it for different purpose for finding area

## Other Nonfunctional Requirements

• Scalability.

• Reliability.

• Regulatory.

• Maintainability.

• Serviceability.

• Utility.

• Security.

• Manageability

## Performance Requirements

• Micro-Controller Performances

• Connection Between User and Product (like application and hardware part)

• Motors Power

• Signals Sending and Receiving Time

## Safety Requirements

• Speed Control

• Battery Health

• External Guard Sheet/Body on Product

• SOPs in Case of Emergency

## Security Requirements

In our case Security Requirement are very because two function are only on based on hardware.

• Sending and Receiving Signals with Proper Encryption.

• Connection Establishment only one Device at a time.

## Usability Requirements

The following five essential qualities characterize robots as we have come to know them today.

* Intelligence. Human intelligence is derived from the elaborate and interconnected network of neurons within the human brain
* Sense Perception
* Dexterity
* Power
* Independence
* These All Present in our Presented Product

## Reliability Requirements

One of the most essential aspects of a reliability program is defining the reliability goals that a product needs to achieve. This article will explain the proper ways to describe a reliability goal and highlight some of the way’s reliability requirements are commonly defined improperly.

Designs are usually based on specifications. Reliability requirements are typically part of a technical specifications document. They can be requirements that a company sets for its product and its own engineers or what it reports as its reliability to its customers. They can also be requirements set for suppliers or subcontractors. However, reliability can be difficult to specify. It is easy to use "qualitative" language such as, "our product needs to exceed customer expectations" or "our product should be more reliable than its competition." Joseph Juran, a famous quality pioneer, said, "If you don't measure it, you don't manage it." If an organization does not specify reliability goals numerically, it loses control over managing its products' reliability improvements**.**

For Our Product Reliability Requirements are

* Quality Of sells using in Battery
* Motor Performance and Reliability Parameter
* Controller Performance and Reliability Parameter
* Under which Circumstances and Environment Product is used Reliability Requirements Varies.

## Maintainability/Supportability Requirements

* Motors Maintenance
* Sensor Performance and Maintenance
* Micro-Controller Performance and supportability
* Life Cycle of all Material use in the Development of this Product

## Portability Requirements

Portability is the ease with which a software system can be transferred from its current hardware or software environment to another environment.

ELICITATION: Portability requirements address the user concern for how easy it is to transport the system.

## Efficiency Requirements

Efficiency requirements Describe the extent to which capture software makes optimal use of resources the speed with which payment system executes and the care it consumes for its operation for is the system should still be able to site at round three times faster than the existing system.

## Domain Requirements

All Requirement are Covered in SRS

# Chapter 3

# Use Case Analysis

**Chapter 3:** System Analysis

Use case analysis is a technique used to identify the requirements of a system (normally associated with software/process design) and the information used to both define processes used and classes (which are a collection of actors and processes) which will be used both in the use case diagram and the overall use case in the development or redesign of a software system or program. The use case analysis is the foundation upon which the system will be built.

In this chapter we discuss and elaborate the use case model and check our system requirements and processes involved.

## Use Case Model

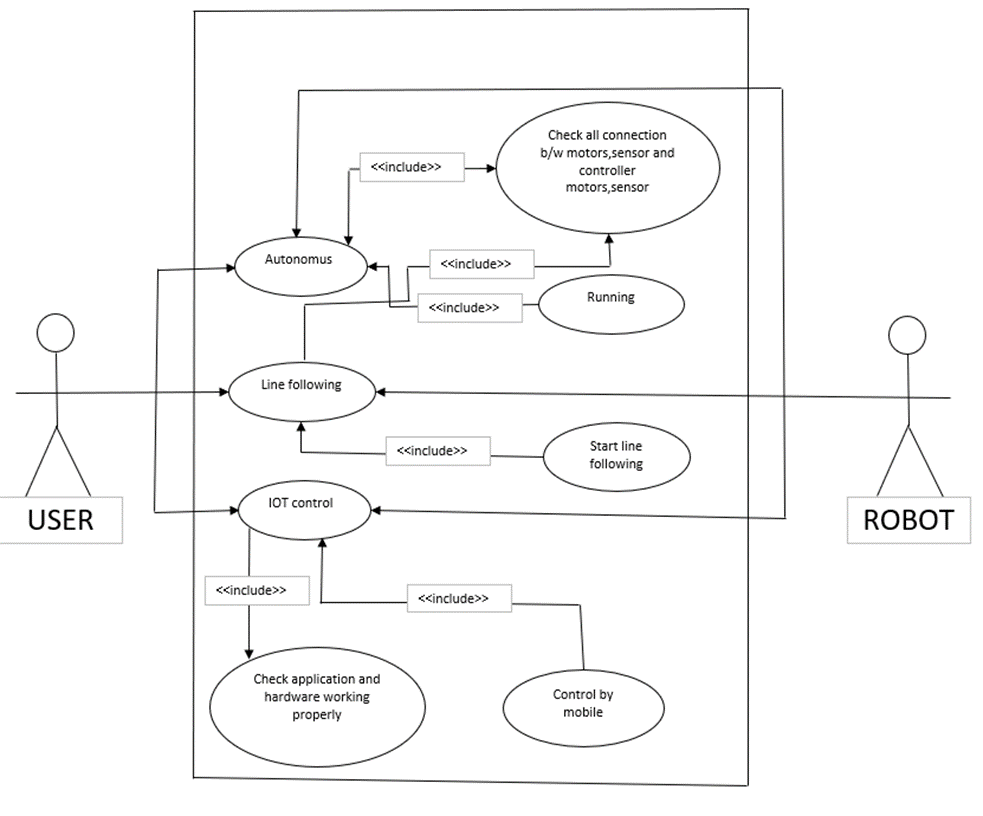
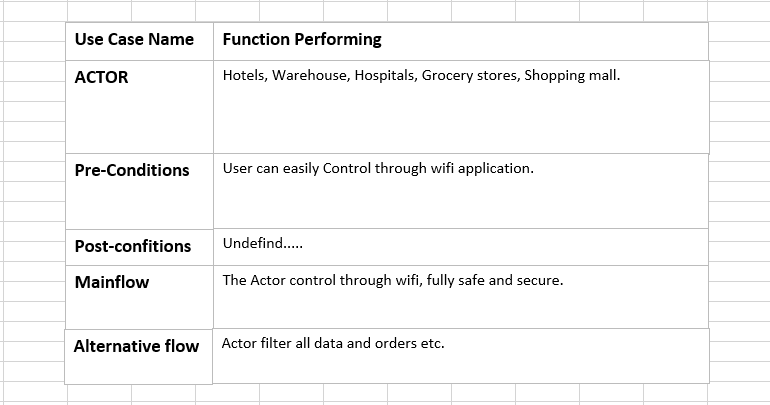


Figure 0-3 MODEL OF USE CASE

## Use Case Descriptions



# Chapter 4

# System Design

**Chapter 4:** System Design

This is chapter is going to define behavior of the project in the form of UML diagrams. In this chapter we can easily understand the functionality of the whole system and not even one stakeholder designer, developer, tester etc. equally can get advantage while developing designing and testing also. This chapter plays a vital role in the whole documentation due to his mutually beneficial for everyone.

## Architecture Diagram

Following its the architecture diagram of system as our system is IoT Based project. So, the whole architecture is given below.

**Line Following:**

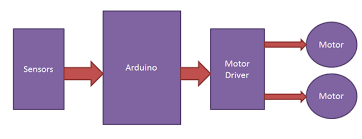


Figure 0‑4 LINE FOLLOWING

## Class Diagram

OBSTACLE AVOID:

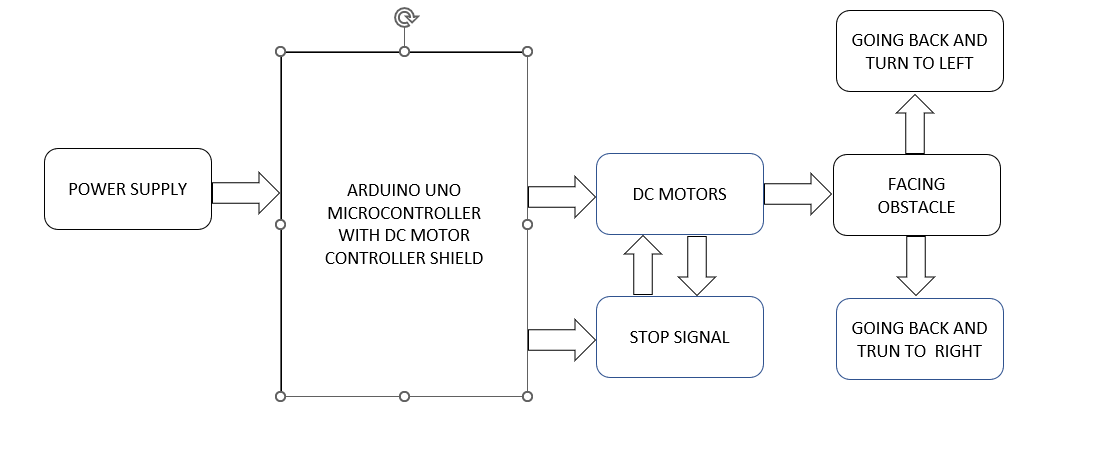


Figure 0‑5 OBSTACLE AVOID

LINE FOLLOWING:

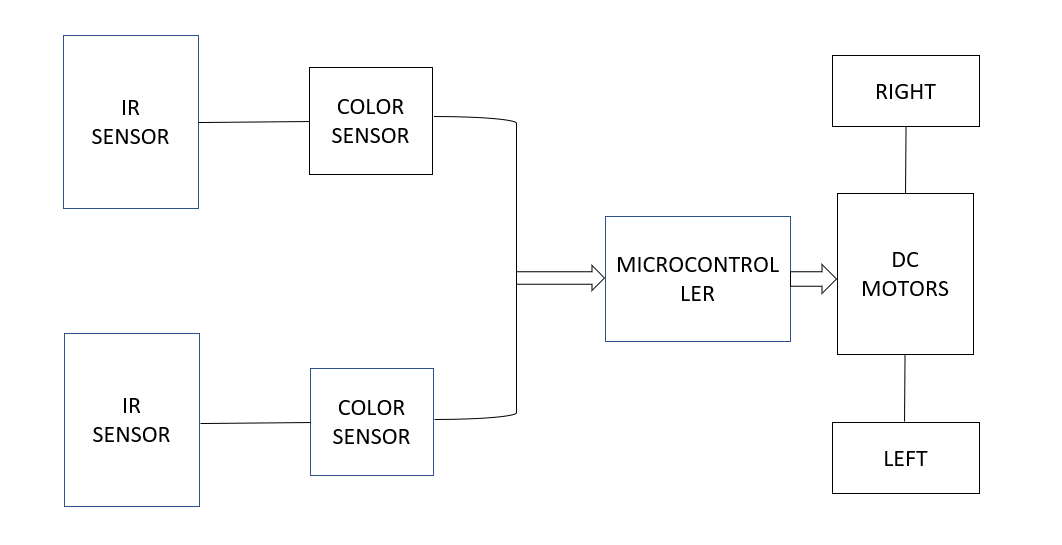


Figure 0‑6 LINE FOLLOWING

IOT BASED CAR:

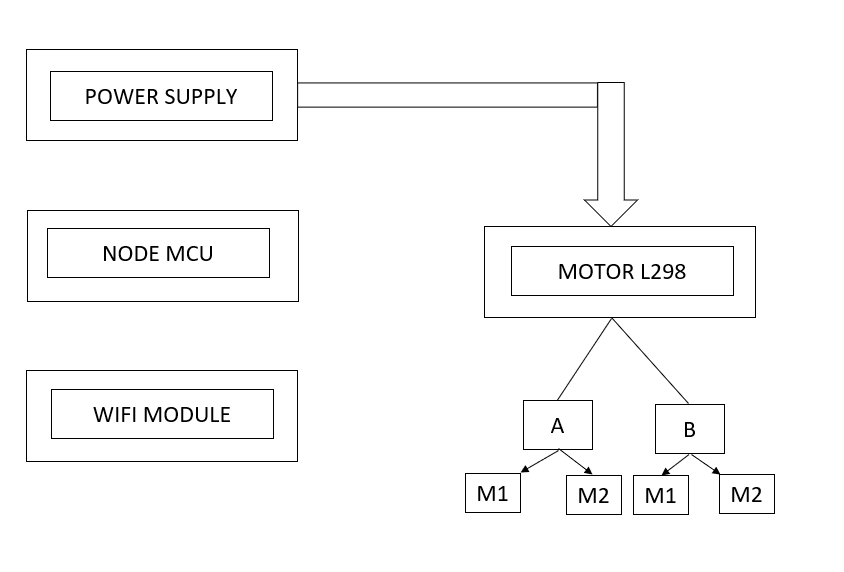


Figure 0‑7 IOT BASED CAR

## Data Flow diagram

OBSTACLE AVOID:

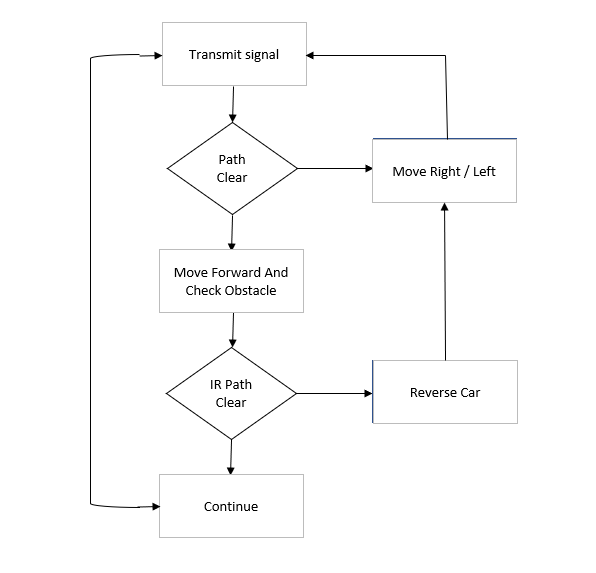


Figure 0‑8 DFD OBSTACLE AVOID

LINE FOLLOWING:

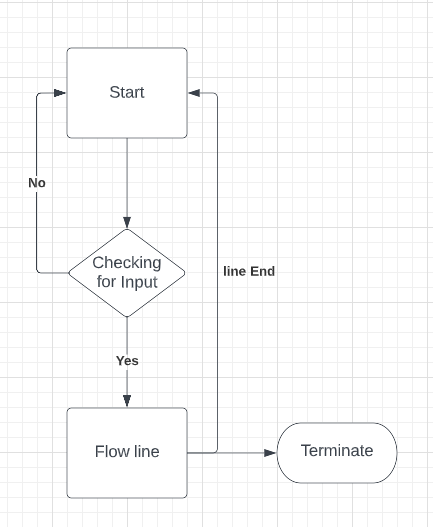


Figure 0‑9 DFD LINE FOLLOWING

# IOT BASED CAR:

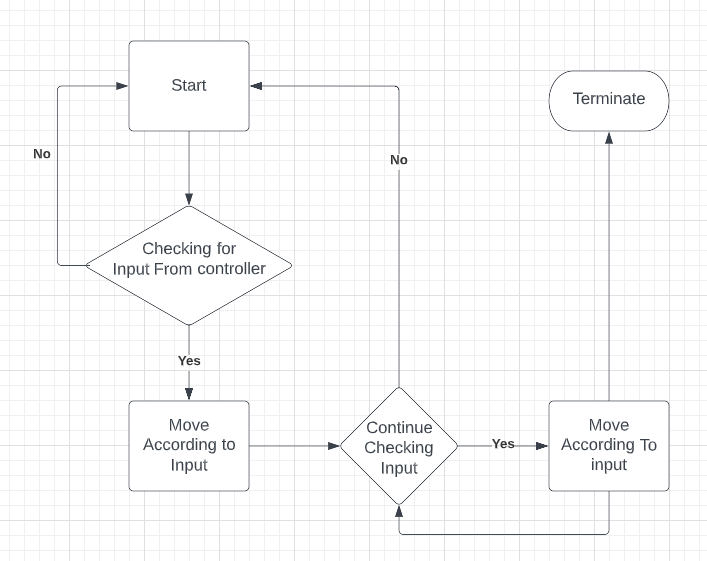


Figure 0‑10 DFD IOT BASED CAR

## ERD DIAGRAM

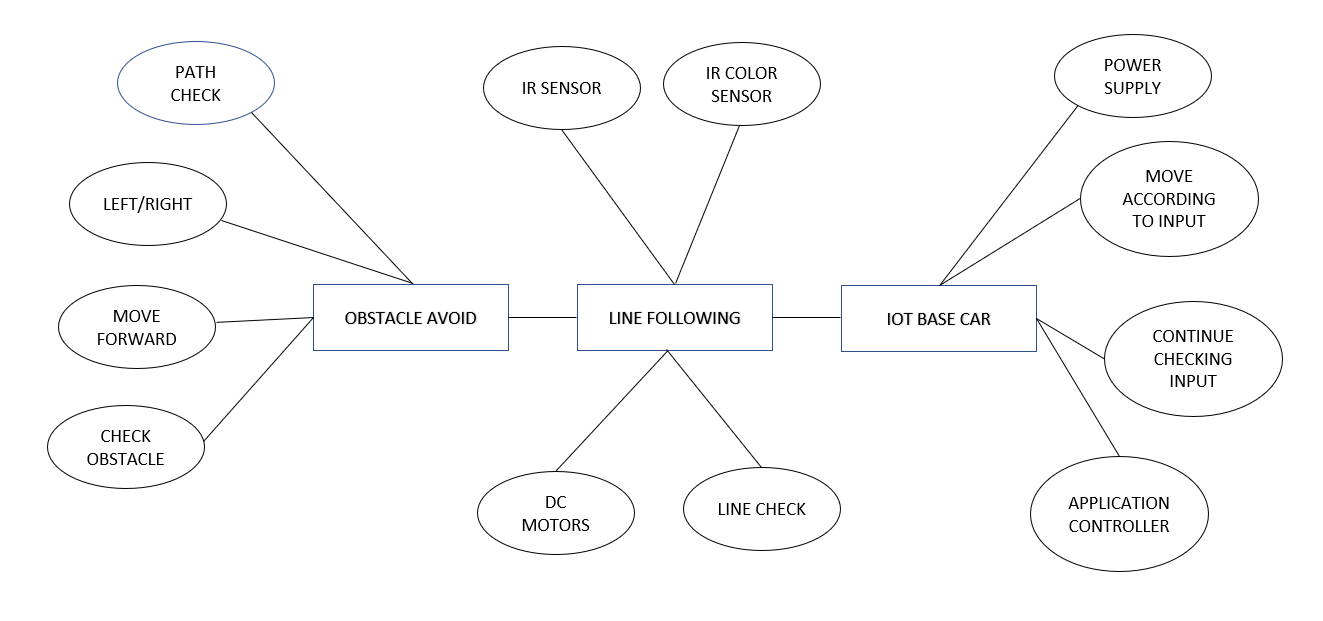


Figure 0‑11 ERD DIAGRAM

# Chapter 5

# Implementation

**Chapter 5:** Implementation

This is chapter is going to define implementation of the project in the form of Step. In this chapter we can easily understand the functionality of the whole system and not even one stakeholder designer, developer, tester etc. equally can get advantage while developing designing and testing also. This chapter plays a vital role in the whole documentation due to his mutually beneficial for everyone. We can easily understand working and controlling of the project.

## Components, Libraries, Web Services, and stubs

**Obstacle Avoiding**

* Arduino Uno
* DC Motors
* Ultrasonic Sensor
* Motor Drivers
* Battery 5V
* Toy Car Kit

**Line Following**

* Arduino Uno
* DC Motors
* Line Follow Sensor
* Motor Drivers
* Battery 5V
* Toy Car Kit

**IoT Based Car**

* Arduino Uno
* DC Motors
* Motor Drivers
* Wi-Fi Module
* Battery 5V
* Toy Car Kit

**Libraries:**

IDE Arduino will use for Programming of Project all modules and mostly Libraries use will

* AFMotor.h
* NewPing.h
* Servo.h

## Deployment Environment

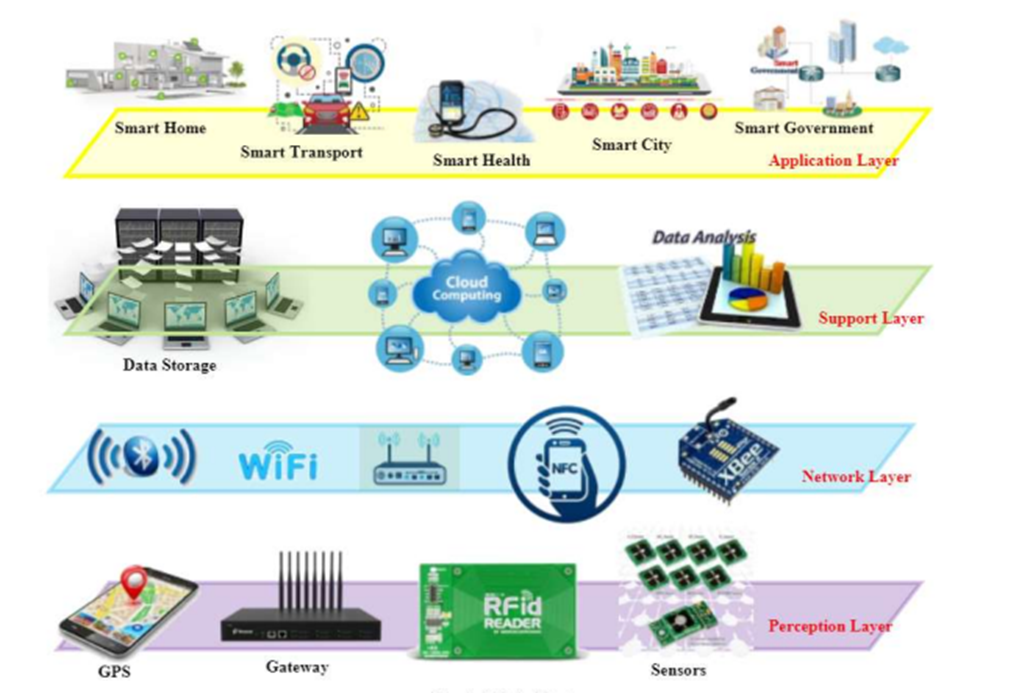


Figure 0‑12 DEVELOPMENT ENVIRONMENT

## Tools and Techniques

**Tools:**

* Data Collection
* Signal Processing
* Hardware Design
* Software Implementation
* Chassis design and fabrication

**Technique:**

* Bug Algorithm
* Artificial Potential Field Method
* Bubble Band Technique
* Fixed Sonar Method

## Best Practices / Coding Standards

**Programming Language:**

C language used for all Project

* Module 1 C programming language use
* Module 2 C programming language use
* Module 3 C programming language used But MT Platform will use for IoT application Development

**Coding Standards:**

Coding rules and guidelines ensure that software is:

* **Safe:** It can be used without causing harm.
* **Secure:**It can’t be hacked.
* **Reliable:** It functions as it should, every time.
* **Testable:** It can be tested at the code level.
* **Maintainable:** It can be maintained, even as your codebase grows.
* **Portable:**It works the same in every environment which is required by Product.

**As our Project Domain is Embedded System So, we ensure Our Coding Standards**

This is especially true for [functional safety standards](https://www.perforce.com/resources/qac/what-is-functional-safety), including:

* [IEC 61508](https://www.perforce.com/resources/qac/how-comply-iec-61508-standard): “Functional safety of electrical/electronic/programmable electronic safety-related systems”
* [ISO 26262](https://www.perforce.com/blog/qac/what-iso-26262-overview): “Road vehicles — functional safety”
* [EN 50128](https://www.perforce.com/resources/qac/how-achieve-en-50128-compliance): “Railway applications — Communication, signaling, and processing systems Software for railway control and protection systems”
* [IEC 62061](https://www.perforce.com/resources/qac/why-functional-safety-critical-embedded-software): "Safety of machinery: Functional safety of electrical, electronic and programmable electronic control systems

## Version Control

Software developers should have a rudimental understanding of what VCS is and which type of VCS suits them. The adoption of a VCS is a must in software development. It helps software developers manage their codes easily because it is common to have a lot of changes involving addition or deletion of features. To adopt a VCS, a software developer must know and perfectly understand which approach should be used as it will affect the whole project and team. It is also important for them to have the knowledge of different approaches of VCS because the various approaches will affect their software development process differently.

VCS Explains These Terms

• Software Merging

• Collaboration Modelling

• Software Changes

• Software Branching

• Open Software Projects

• Curriculum Development

I will be using the Sloeber Eclipse plugin for writing project Arduino code lately. Eclipse interacts nicely with Git by itself. You get to a point and then you make a commit, and it asks if you want to push to GitHub or somewhere and you can if you want.

# 

# Chapter 6

# Testing and Evaluation

**Chapter 6:** Testing and Evaluation

In this chapter we will discuss about the results after testing what we expect and what are the actual results in every type of testing. First, we will perform Use Case testing and after that Equivalence, Boundary value analysis, Data flow testing, Unity testing, then integration, performance and on last Stress testing then our testing phase will complete and if the testing result are positive then we launch our product otherwise we will do improvement to achieve our expected result.

## Use Case Testing

All cases were tested successfully, and we got our desired result.

## Boundary value analysis

Check all sensors and wires but mostly issues in wires and ultrasonic sensor.

## Data flow testing

Data successfully flowed and we tested.

## Unit testing

In unit testing we check all the single individual modules for their functionality. Means we test each single module that it is working perfectly or not.

## Integration testing

In integration testing we integrate the individual single modules and allow them to work together and test them for. We test the communication between the module is working perfectly and all the modules are performing as expected in a common environment or not.

## Performance testing

Performance index of our prototypes are moderate due to lack of material and battery efficiency.

Chapter 7

# Summary, Conclusion and Future Enhancements

**Chapter 7:** Summary, Conclusion & Future Enhancements

## Project Summary

Obstacle Avoiding Robot is an intelligent device that can automatically sense the obstacle in front of it and avoid them by turning itself in another direction. This design allows the robot to navigate in an unknown environment by avoiding collisions, which is a primary requirement for any autonomous mobile robot.

The line follower robot senses a black line by using a sensor and then sends the signal to Arduino. Then Arduino drives the motor according to sensors' output. Here in this project, we are using two IR sensor modules namely the left sensor and the right sensor.

The proposed concept of this modified Wi-Fi controlled robot or node MCU Robot used for navigation or location identification. By Introducing this navigation and location identification concepts in robots can save human beings from a risky and harmful natural disaster especially that happens in power plants.

## Achievements and Improvements

We Successfully Developed our Multi-functional robot prototype and tested it under various circumstances and got our desired results. After fulfilling of our proposed objective, we analyzed many improvements are indeed some are following:

* Accuracy
* Processing Time
* Structure
* Cost Efficiency

## Future Enhancements/Recommendations

The line follower developed also sensing any type of obstacles in its way and can also control speed with the help of speed regulator. Further improvements can be made in the robot by using a greater number of IR sensors or an array of IR sensors.

In autonomous USV can fulfill a variety of missions and applications that are of increasing interest for the US Navy and other Department of Defense and Department of Homeland Security organizations. The USV obstacle avoidance package is being developed first by accurately creating a world model based on various sensors such as vision, radar, and nautical charts. Then, with this world model the USV can avoid obstacles with the use of a far-field deliberative obstacle avoidance component and a near-field reactive obstacle avoidance component.

# 

# Appendices

# Code:

# IoT Based

define ENA 14 // Enable/speed motors Right GPIO14(D5)

#define ENB 12 // Enable/speed motors Left GPIO12(D6)

#define IN\_1 15 // L298N in1 motors Right GPIO15(D8)

#define IN\_2 13 // L298N in2 motors Right GPIO13(D7)

#define IN\_3 2 // L298N in3 motors Left GPIO2(D4)

#define IN\_4 0 // L298N in4 motors Left GPIO0(D3)

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

String command; //String to store app command state.

int speedCar = 800; // 400 - 1023.

int speed\_Coeff = 3;

const char\* ssid = "NodeMCU Car";

ESP8266WebServer server(80);

void setup() {

pinMode(ENA, OUTPUT);

pinMode(ENB, OUTPUT);

pinMode(IN\_1, OUTPUT);

pinMode(IN\_2, OUTPUT);

pinMode(IN\_3, OUTPUT);

pinMode(IN\_4, OUTPUT);

Serial.begin(115200);

// Connecting WiFi

WiFi.mode(WIFI\_AP);

WiFi.softAP(ssid);

IPAddress myIP = WiFi.softAPIP();

Serial.print("AP IP address: ");

Serial.println(myIP);

// Starting WEB-server

server.on ( "/", HTTP\_handleRoot );

server.onNotFound ( HTTP\_handleRoot );

server.begin();

}

void goAhead(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, HIGH);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, HIGH);

analogWrite(ENB, speedCar);

}

void goBack(){

digitalWrite(IN\_1, HIGH);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, HIGH);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar);

}

void goRight(){

digitalWrite(IN\_1, HIGH);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, HIGH);

analogWrite(ENB, speedCar);

}

void goLeft(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, HIGH);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, HIGH);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar);

}

void goAheadRight(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, HIGH);

analogWrite(ENA, speedCar/speed\_Coeff);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, HIGH);

analogWrite(ENB, speedCar);

}

void goAheadLeft(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, HIGH);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, HIGH);

analogWrite(ENB, speedCar/speed\_Coeff);

}

void goBackRight(){

digitalWrite(IN\_1, HIGH);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar/speed\_Coeff);

digitalWrite(IN\_3, HIGH);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar);

}

void goBackLeft(){

digitalWrite(IN\_1, HIGH);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, HIGH);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar/speed\_Coeff);

}

void stopRobot(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar);

}

void loop() {

server.handleClient();

command = server.arg("State");

if (command == "F") goAhead();

else if (command == "B") goBack();

else if (command == "L") goLeft();

else if (command == "R") goRight();

else if (command == "I") goAheadRight();

else if (command == "G") goAheadLeft();

else if (command == "J") goBackRight();

else if (command == "H") goBackLeft();

else if (command == "0") speedCar = 400;

else if (command == "1") speedCar = 470;

else if (command == "2") speedCar = 540;

else if (command == "3") speedCar = 610;

else if (command == "4") speedCar = 680;

else if (command == "5") speedCar = 750;

else if (command == "6") speedCar = 820;

else if (command == "7") speedCar = 890;

else if (command == "8") speedCar = 960;

else if (command == "9") speedCar = 1023;

else if (command == "S") stopRobot();

}

void HTTP\_handleRoot(void) {

if( server.hasArg("State") ){

Serial.println(server.arg("State"));

}

server.send ( 200, "text/html", "" );

delay(1);

}

**Obstacle Avoidance:**

#include <Servo.h> //Servo motor library. This is standard library

#include <NewPing.h> //Ultrasonic sensor function library. You must install this library

//our L298N control pins

const int LeftMotorForward = 7;

const int LeftMotorBackward = 6;

const int RightMotorForward = 5;

const int RightMotorBackward = 4;

//sensor pins

#define trig\_pin A1 //analog input 1

#define echo\_pin A2 //analog input 2

#define maximum\_distance 200

boolean goesForward = false;

int distance = 100;

NewPing sonar(trig\_pin, echo\_pin, maximum\_distance); //sensor function

Servo servo\_motor; //our servo name

void setup(){

pinMode(RightMotorForward, OUTPUT);

pinMode(LeftMotorForward, OUTPUT);

pinMode(LeftMotorBackward, OUTPUT);

pinMode(RightMotorBackward, OUTPUT);

servo\_motor.attach(10); //our servo pin

servo\_motor.write(115);

delay(2000);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

}

void loop(){

int distanceRight = 0;

int distanceLeft = 0;

delay(50);

if (distance <= 20){

moveStop();

delay(300);

moveBackward();

delay(400);

moveStop();

delay(300);

distanceRight = lookRight();

delay(300);

distanceLeft = lookLeft();

delay(300);

if (distance >= distanceLeft){

turnRight();

moveStop();

}

else{

turnLeft();

moveStop();

}

}

else{

moveForward();

}

distance = readPing();

}

int lookRight(){

servo\_motor.write(50);

delay(500);

int distance = readPing();

delay(100);

servo\_motor.write(115);

return distance;

//delay(100);

}

int lookLeft(){

servo\_motor.write(170);

delay(500);

int distance = readPing();

delay(100);

servo\_motor.write(115);

return distance;

delay(100);

}

int readPing(){

delay(70);

int cm = sonar.ping\_cm();

if (cm==0){

cm=250;

}

return cm;

}

void moveStop(){

digitalWrite(RightMotorForward, LOW);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

digitalWrite(LeftMotorBackward, LOW);

}

void moveForward(){

if(!goesForward){

goesForward=true;

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

}

void moveBackward(){

goesForward=false;

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

void turnLeft(){

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

# 

# Reference and Bibliography

**Reference and Bibliography**

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley, USA (1974)

2. Ben-Ari, M.: Principles of Concurrent and Distributed Programming, 2nd end. Addison-Wesley, USA (2006)

3. Dudek, G., Jenkin, M.: Computational Principles of Mobile Robotics, 2nd edn. Cambridge University Press, UK (2010)

4. Kumar, D.: Learning Computing with Robots. Lulu (2011). Download from http://calicoproject. org/Learning\_Computing\_With\_Robots

5. Shin, J., Siegwart, R., Magnenat, S.: Visual programming language for Thymio II robot. In: Proc. of the 2014 Conference on Interaction Design and Children (IDC) (2014)

6. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots, 2nd edn. MIT Press, USA (2011)

7. Trobaugh, J.J., Lowe, M.: Winning LEGO MINDSTORMS Programming. Apress (2012)

8. Winfield, A.: Robotics: A Very Short Introduction. Oxford University Press, USA (2012)