Table of Contents

List of figures

List of tables

CHAPTER 1

INTRODUCTION

CHAPTER 1

1.1 INTRODUCTION

Internet of Things (IOT) is system of related sensors, computing and digital devices spread across the globe over the internet which can communicate amongst them to share and transfer data and information using unique id which is assigned to each and every device, as UIDs (Unique Identifiers) [1]. The expression “Internet of Things” (IoT), coined back in 1999 by Kevin Ashton, the British technology pioneer who cofounded the Auto-ID Center at the Massachusetts Institute of Technology [2].The idea of adding sensors and intelligence to basic objects was discussed throughout the 1980s and 1990s (and there are arguably some much earlier ancestors), but apart from some early projects -- including an internet-connected vending machine -- progress was slow simply because the technology wasn't ready. Chips were too big and bulky and there was no way for objects to communicate effectively [3].

The number of online connected devices increased 31% from 2016 to 6.4 billion in 2017. In 2018 there were 7 billion IoT devices connected, this number increases to 26.6 billion in 2019, this means 127 new IoT devices are connected to web ever second. During 2020, experts estimate the installation of 31 billion IoT devices. They also forecast, 35 billion IoT devices will be installed worldwide by 2021 and at the end of 2025, more than 75 billion will be connected. The global market value was predicted to be $7.1 trillion by 2020. In 2018 alone the North American IoT market generated $83.9 billion in revenue. During 2020 global spending on the IoT should reach $1.29 trillion. By 2021 the industrial IoT market size should reach $124 billion and by 2024, the global IoT healthcare market should reach $140 billion. Experts estimate that the IoT device market will reach $1.1 trillion at the end
0f 2026 [4]. IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention.

Barrier detection and recognition systems that detect and recognize the presence of Barrier, pedestrians and vehicles play an important role in DASs. They can greatly reduce the incidence of accidents due to careless driving. Therefore, the development of a robust and real-time detection and recognition system is heavily desirable. However, the development of such a method is still challenging to the research community due to various conditions in real-world driving. On continuing the trend, we decided to build a voice-controlled robot which can avoid the barrier in real time.

Barrier detection is the process of using sensors, data structures, and algorithms to detect objects or terrain types that impede motion. Barrier detection is applicable to anything that moves, including robot manipulators and manned or unmanned vehicles for land, sea, air, and space; for brevity, these are all called vehicles here. Barrier detection is a system problem that encompasses sensors that perceive the world, world models that represent the sensor data in a convenient form, mathematical models of the interaction between objects and the vehicle, and algorithms that process all of this to infer Barrier locations. Barrier detection algorithms use the world model and the interaction model to locate Barriers, to characterize the degree to which the Barriers impede motion, and to produce an Barrier map for use by path planning algorithms. The complexity of Barrier detection systems varies greatly, depending on the domain of operation, the size and cost of the vehicle, and the degree of reliability required. Because sensor data is noisy and incomplete, often it is necessary to combine many sensor measurements into the world model to reduce noise and in gaps in the world model before applying Barrier detection algorithms; this makes Barrier detection related to mapping. Barriers may completely block motion of the vehicle or just make it more difficult to move, such as having to move slowly over rough ground; this makes Barrier detection related to terrain classification [5].

Barrier detection algorithms implicitly or explicitly use a model of how the vehicle interacts with the world to determine where Barriers exist. In the simplest case, the model may be just a binary decision based on raw sensor data, such as when single-axis scanning radars are used for Barrier detection indoors and any object that produces a range measurement counts as a Barrier. For outdoor navigation, the size of an object affects whether or not it is considered an obstacle; for example, small bumps on the ground may be ignored, but large rocks are Barriers. This distinction depends on characteristics of the vehicle, such as wheel diameter, ground clearance, velocity, and suspension stiffness. In this context, interaction modeling can take into account just the geometry of the vehicle and world, or varying levels of fidelity in the dynamics of the interaction [6].

This robotic is designed to control vehicle by using human voice command through Bluetooth module. Voice Control Robot is used to complete specific commands like Forward, Backward, Stop, Left, Right and dancing (or rotation of robot) etc. Voice Control Robot is based on Speech Recognition. The commands are given to robot using Android application. The Android application (AMR – Voice) is connected to Bluetooth Module (HC – 05), which is directly connected to Arduino Uno R3. We give command to the robot and it performs work according to the given command. Voice Control Robot is much useful for those areas where humans can't reach. Robot can work in all type of situations like toxic area, in fire situations, polluted area and also on hills. This robot is very useful for those who is physically handicapped. This robot is very small in size so we can use this project for spying or espial. If we implement in this project so we can use this robot in military application, agriculture purpose, industrial purpose and also for surveillance device.

Speech is an ideal method for robotic control and communication. The speech recognition circuit we will outline, functions independently form the robot's main intelligence [central processing unit (CPU)]. This a good thing because it doesn't takes any of the robots main CPU processing power for word recognition. The CPU must merely poll the speech circuit's recognition lines occasionally to check if a command has been issued to the robot. We can even improve upon this by connecting the recognition line to one of the robot's CPU interrupt lines. By doing this, a recognized word would cause an interrupt, letting the CPU know a recognized word had been spoken. The advantage of using an interrupt is that polling the circuit's recognition line occasionally would no longer be necessary, further reducing any CPU overhead.

In this project, the robot basically works on human speech command. The Voice Control Robot is controlled by using voice command which is directly given by user to the robot. We can say, this is a wireless robot. The android application is installed in smartphone which works as a transmitter. The commands are given by this android application. The android application AMR Voice is use to recognize the Arduino using a Bluetooth link. The Bluetooth module (HC-05) which is connected to the Arduino. As we know that Arduino is programmable, so we have to do the programming using

C or Java Language. When the programming of Arduino is done, we connect all the connection as required for the robot. Hence we connect Android application (AMR Voice) and Bluetooth module (HC-05) using Bluetooth link. The commands are given by the AMR Voice by the user. These commands are received by Bluetooth module and Arduino perform the operation according to the given commands. The given command by the user is converted into digital form. These commands can be Left, Right, Backward, Forward or Rotation of Robot etc. The range of this robot is up to 100 meters. If we want to make this for a certain purpose the range can be increase.

1.2 PROBLEM STATEMENT:

Safety is a cornerstone of both connected vehicles and advanced driver assistance systems (ADAS). More complex safety problems are expected in mixed traffic of automated and manual vehicle as a result of distraction [7]. Distracted driving is one of the most dangerous reasons increasing crash risk. Although people are aware of risks of sending messages or making phone calls while driving, they still fail to focus on driving all the time anyway. Though India is home to just 1 per cent of the world's vehicles, it still account for 11 per cent of the global road accident deaths, the highest in the world. With 53 road crashes per hour, road accidents in India continue to be highest in world [8].Traffic risks can be intuitively understood as the probability and severity of accidents that drivers have a chance to encounter in the future. According to this concept, in order to decrease the risk in a particular driving scenario, it is necessary to create an accurate vehicle motion detection system (VMDS) that can detect real time barrier status and avoid the obstacle with voice as well as through automatic methods during the case of emergency. With VMDS, a kind of ADAS, drivers are able to have more time and space to respond to the potential risk. It is reported that 80% of existing crashes could have been prevented if the risk could be sensed and drivers could communicate the problem in time.

1.3 PROPOSED SYSTEM:

The whole system is built upon Arduino with Bluetooth model, C/C++ as processing library, MIT App inventor for android application and QT as editor. The purpose of this project is to build a robotic car which could be controlled using voice commands, can manually control the robot through use of android application, as well as to detect the barrier automatically and try to avoid it. So, the proposed system is the hybrid of three functionality which are mentioned below:

1.3.1 Voice Controlled System:

For the first stage we have prototype of Speech Controlled Automation Systems (SCAS). We are not aiming to build a robot which can recognize a lot of words. Our basic idea is to develop some sort of menu driven control for our robot, where the menu is going to be voice driven. What we are aiming at is to control the robot using following voice commands. The robot can do these basic tasks:- 1. move forward 2. move back 3. turn right 4. turn left 5. load 6. release 7. stop (stops doing the current job). The robot is controlled through Bluetooth sensor.

The Arduino voice-controlled robot car is interfaced with a Bluetooth module HC-05 or HC-06. We can give specific voice commands to the robot through an Android app installed on the phone. At the receiving side, a Bluetooth transceiver module receives the commands and forwards them to the Arduino and thus the robotic car is controlled.

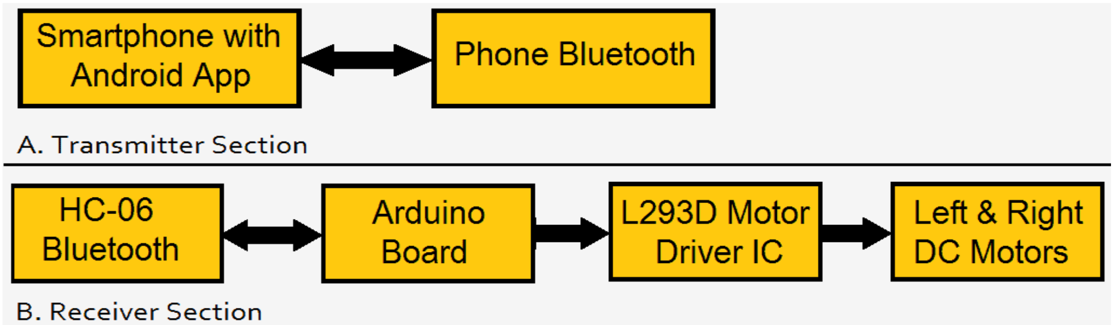

Figure 1.1 Voice Control Flow Diagram

Input (user Input)	Output (Robot movement)
Forward	Moves Forward
Back	Moves back
Right	Turns right
Left	Turns Left

Table 1.1 Input vs Output table

In this project, the robot basically works on human speech command. The Voice Control Robot is controlled by using voice command which is directly given by user to the robot. We can say, this is a wireless robot. The android application is installed in smartphone which works as a transmitter. The commands are given by this android application. The android application AMR Voice is use to recognize the Arduino using a Bluetooth link. The Bluetooth module (HC-05) which is connected to the Arduino.

1.3.2 Auto barrier detection using IR sensor:

Varieties of sensors are available which can be used for the detection of obstacles some of the very popular sensors are: Infrared sensors (IR), Ultrasonic sensors, Cameras, which can be used as a part of Computer Vision, Sonar. It can measure the distance in its field of view of about thousands to hundreds of points.

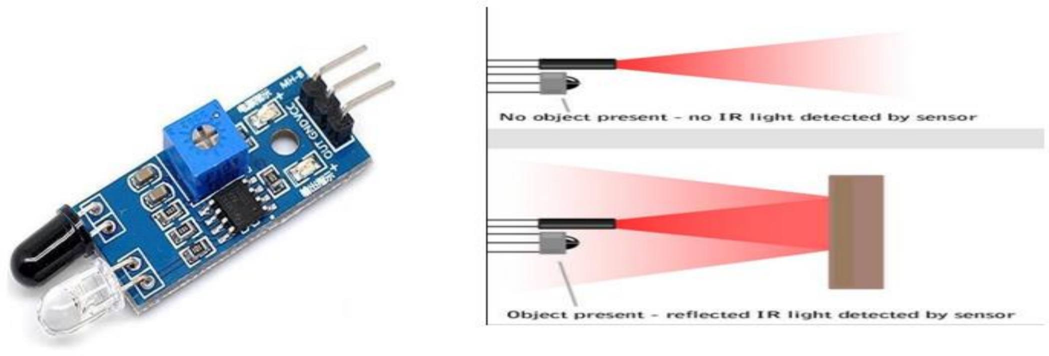

Figure 1.2 IR sensor and Its working Principal

In the design of robot, we are using ultrasonic sensors for obstacle detection and avoidance. It can be used to improve the overall vision system of mobile robot . IR sensors are widely used for measuring distances, so they can be used in robotics for obstacles avoidance. IR sensors are also faster in response time than ultrasonic sensors [9]. In addition, the power consumption of IR sensor is lower than ultrasonic sensors

[10]. Active Infrared (IR) sensors can be an emitter and detector, which operate at the same wavelength. It is also known as photoelectric sensor working with reflective surfaces [11]. IR sensor can be categorized as retro-reflective sensors and diffuse reflection sensors. Retro-reflective sensors are proper for harsh environment conditions and have much larger detection range than the diffuse reflective sensor [11]. IR sensors use a specific light sensor that can detect a selective light wavelength in the IR spectrum. When an object is close to the sensor, the light from the LED bounces off the object and into the light sensor as shown in Figure 1.

1.3.3 Manual Mode through Android App:

The third system that made the entire purposed system is the manual mode. In manual mode the robot can be controlled through the android application installed in the phone. The android application is built through the use frame work known as MIT app Inventor. It uses the concept of block programming.

The detailed discussion about the purposed system is discussed in system design section with its hardware.

1.4 MOTIVATION:

The main purpose of this project is to build a cost-effective “Voice Control Real Time Barrier Detection” system, which can detect the barrier and avoid it automatically or manually through the use of voice command or android application.

1.4 OBJECTIVE:

The first step is to connect the robot to the mobile application for it to function remotely. To obtain the accurate results, sensors of the device plays a very vital role.

- The obstacle avoidance robot is able to move around in an unknown environment without colliding with surrounding objects.
- The robot would have the capacity to detect obstacles in its path based on a predetermined threshold distance.
- After obstacle detection, the robot would change its course to a relatively open path by making autonomous decision.
- It would require no external control during its operation.
- It can measure the distance between itself and the surrounding objects in real-time.
- It would be able to operate effectively in unknown environment.
- It can be controlled either using a mobile application or by voice control.
- They can be used for household work like automatic vacuum cleaning.
- They can also be used in dangerous environments, where human penetration could be fatal

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

2.1 LITERATURE SURVEY

2.1.2 Artificial Intelligence

Artificial intelligence is the branch of computer science that develops machines and software with human-like intelligence. It is the intelligence exhibited by software or machines. The central goals of artificial intelligence research include knowledge, reasoning, learning, planning, perception, the ability to manipulate and move objects and natural language processing [12]. The field was founded on the claim that a central property of humans is intelligence, and that it can be sufficiently well described to the extent that a machine can simulate it.

2.1.3 Robotics and Robots

It is a branch of technology and deals with designing, construction, operation, and application of robots. It also deals with the computer systems for their sensory, control, information processing and feedback. These technologies deal with automated machines that can replace humans in manufacturing processes or dangerous environments. These robots resemble humans in behavior, appearance, and/or cognition. Robotics requires a working knowledge of mechanics, electronics, and software . Robots are machines and are of a wide range [13]. The common feature of robots is their capability to move. They perform physical tasks. Robots have many different forms [14]. They range from industrial robots, whose appearance is dictated by the function they are to perform. Or they can be humanoid robots, which mimic the human movement and our form. Robots can be grouped generally as: Manipulator robots (for e.g., industrial robots), Mobile robots (for e.g., autonomous vehicles), ØSelf-reconfigurable robots, the robots that can conform themselves to the task at hand. Robots may act according to their own decision-making ability, provided by artificial intelligence or may be controlled directly by a human, such as remotely-controlled bomb disposal robots and robotic arms; or. However, the majority of robots fall in between these extremes, being controlled by pre-programmed computers [15].

2.1.4 Robot Working:

Human beings on a basic level are made of five major components:

- A muscle system that can move the body structure
- A body structure itself
- A power source that can activate the muscles and sensors
- A sensory system which can receive information about the body and the surrounding environment
- A brain system which can process sensory information and tell the muscles what to do.

Robots are made up of the same components as above. A typical autonomous robot has a sensor system, a movable physical structure, a power supply and a computer brain that controls all of these elements. Basically, robots are man-made versions of the animal life. They are machines that can replicate human and animal behavior [16].

2.1.5 Robot Learning:

Robot learning is an intersecting research field between robotics and machine learning. It studies techniques that allow robots to acquire skills and adapt to its environment by learning various algorithms. Learning can take place either by self-exploration or through guidance (from a human teacher), like in robot learning that learns by imitation.

2.1.6 Autonomous Robot:

Autonomous robots are independent of any controller and can act on their own. The robot is programmed to respond in a particular way to an outside stimulus [17]. The bump-and-go robot is a good example. This robot uses bumper sensors to detect obstacle. When the robot is turned on, it moves in a straight direction and when it hits an obstacle, the crash triggers its bumper sensor. The robot gives a programming instruction that asks the robot to back up, turn to the right direction and move forward. This is its response to every bump. In this way, the robot can change direction every time, it encounters an obstacle.

A more elaborate version of the same idea is used by more advanced robots. Roboticists create new sensor systems and algorithms to make robots more perceptive and smarter. Today, robots are able to effectively navigate a variety of environments. Obstacle avoidance can be implemented as a reactive control law whereas path planning involves the precomputation of an obstacle-free path which a controller will then guide a robot along [18].

Some mobile robots also use various ultrasound sensors to see obstacles or infrared. These sensors work in a similar fashion to animal echolocation. The robot sends out a beam of infrared light or a sound signal. It then detects the reflection of the signal. The robot locates this distance to the obstacles depending on how long it takes the signal to bounce back.

Some advanced robots also use stereo vision. Two cameras provide robots with depth perception. Image recognition software then gives them the ability to locate, classify various objects. Robots also use smell and sound sensors to gain knowledge about its surroundings.

More advanced robots are able to analyze unfamiliar environments and adapt to them. They even work on areas with rough terrain. This kind of robots can associate particular terrain patterns with particular actions.

For example, a rover robot constructs a land map with the help of its visual sensors. If the map depicts a bumpy terrain pattern, the robot decides to travel some other way. Such kinds of system are very useful for exploratory robots and can be used to operate on other planets. Figure 2.1 shows the bot developed by NASA called Urbie. It is designed for various military purposes and is able to move through stairs and other such paths.

Figure 2.1 Autonomous Urbie developed by NASA

An ant tries to get over an obstacle, it does not decide when it needs to get over an obstacle. It simply keeps trying different things until it gets over the obstacle. An alternative robot design takes a similar less structured approach, which can also be termed as a randomness approach.

When the robot gets stuck, it moves its appendages in every way until something works out. Force sensors work very closely with the actuators, instead of the computer directing everything based on a program.

2.2 TECHNICAL WORK/ PAPER REFERRED

[1]Fakhrul Razi Ahmad, Zakuan; et al "Performance Assessment of an Integrated Radar Architecture for Multi-Types Frontal Object Detection for Autonomous Vehicle". IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS).

Abstract: A hazard-free autonomous vehicle (AV) navigation requires the presence of a dependable perception module. For the frontal object and environment detection, the perception module needs to be able to provide a comprehensive detection region to allow blind spot monitoring. In this work, an integrated perception architecture for frontal object detection of an AV is introduced. The architecture is an integration of central, left and right sides radar for the frontal area. The radar placement of the design is expected to reduce the frontal blind spot region of the vehicle. The architecture's performance is analyzed by assessing each of the radar's performance. This includes their respective ability to measure multi-type of obstacles (pedestrian, black car and motorcycle) as well as evaluating the max range detection of the radars. The results show that each of the radar managed to detect multi-type obstacles as well as possessing a maximum detection range, which will allow for a reduced blind spot area and aids the

hazard mitigation actions. The findings of this preliminary study are important to develop a more comprehensive perception module of an autonomous vehicle.

[2]Obstacle Avoiding Robot By Faiza Tabassum, Susmita Lopa, Muhammad Masud Tarek & Dr. Bilkis Jamal Ferdosi

Abstract: Obstacle detection and avoidance can be considered as the central issue in designing mobile robots. This technology provides the robots with senses which it can use to traverse in unfamiliar environments without damaging itself. In this paper an Obstacle Avoiding Robot is designed which can detect obstacles in its path and maneuver around them without making any collision. It is a robot vehicle that works on Arduino Microcontroller and employs three ultrasonic distance sensors to detect obstacles. The Arduino board was selected as the microcontroller platform and its software counterpart, Arduino Software, was used to carry out the programming. The integration of three ultrasonic distance sensors provides higher accuracy in detecting surrounding obstacles. Being a fully autonomous robot, it successfully maneuvered in unknown environments without any collision. The hardware used in this project is widely available and inexpensive which makes the robot easily replicable.

[3]Rakesh Chandra Kumar, Md Saddam Khan, Dinesh Kumar, Rajesh Birua, "Obstacle Avoiding Robot- A Promising One"

Abstract: In today's world ROBOTICS is a fast growing and interesting field. ROBOT has sufficient intelligence to cover the maximum area of provided space. It has an infrared sensor which are 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. Autonomous Intelligent Robots are robots that can perform desired tasks in unstructured environments without continuous human guidance. The minimum number of gearmotor allows the walking robot to minimize the power consumption while construct a program that can produce coordination of multi-degree of freedom for the movement of the robot. It is found that two gearmotors are sufficient to produce the basic walking robot and one voltage regulators are needed to control the load where it is capable of supplying enough current to drive two gearmotors for each wheel.

[4] Obstacle-avoiding robot with IR and PIR motion sensors by R Ismail, Z Omar and S Suaibun.

Abstract: Obstacle avoiding robot was designed, constructed and programmed which may be potentially used for educational and research purposes. The developed robot will move in a particular direction once the infrared (IR) and the PIR passive infrared (PIR) sensors sense a signal while avoiding the obstacles in its path. The robot can also perform desired tasks in unstructured environments without continuous human guidance. The hardware was integrated in one application board as embedded system design. The software was developed using C++ and compiled by Arduino IDE 1.6.5. The main objective of this project is to provide simple guidelines to the polytechnic students and beginners who are interested in this type of research. It is hoped that this robot could benefit students who wish to carry out research on IR and PIR sensors.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

CHAPTER 3

3.1 SYSTEM REQUIREMENT SPECIFICATION

3.1.1 System Analysis:

The direct result of requirements analysis is Requirement's specification. Hardware requirements specifications list the necessary hardware for the proper functioning of the project [19]. Software requirements specifications is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of use cases that describe interactions the users will have the software. In software engineering, a functional requirement defines the function of a system and its components. A function is described as a set of inputs, the behavior, and outputs. A non-functional requirement that specifies the criteria that can be used to judge the operation of a system, rather than specific behavior [20].

3.1.2 Requirements

A Software Requirements means a requirement specification for a software system is a complete description of a behavior of the system to be developed. In addition to a description of a software functions, the SRS also contains non-functional requirements. Software requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification and validation of requirements for software.

Software Requirements

- **OPERATING SYSTEM**: Android 5.0 and upwards,
  Windows or Ubuntu
- **LIBRARY**: Arduino Library
- **LANGUAGES**: Arduino Programming language (C++)
- **Android Studio and Arduino IDE**

Hardware Requirements

- Arduino UNO microcontroller
- Infrared Proximity Sensor
- Senomotor
- Smart Phone
- Drums
- 12V DC Motors – 60rpm
- Voltage Regulator
- 1293d Motor Driver
- Bluetooth Sensor
- System: Pentium IV 2.4 GHz
- Hard Disk : 500 GB.
- RAM : 4 GB

Any desktop / Laptop system with above configuration or higher level

Functional Requirements

The functional Requirements Specification documents the operation and activities that a system must be able to perform.

Functional requirements include:

- Descriptions of how data is collected and stored.
- Descriptions of data cleaning and pre-processing methods
- Descriptions of work-flows performed by the system.
- Descriptions of outputs.
- How the system meets applicable regulatory requirements.

Non-Functional Requirements:

Non-functional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's quality attributes. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute them. Any attributes required by the customer are described by the specification. We must include only those requirements that are appropriate for our project.

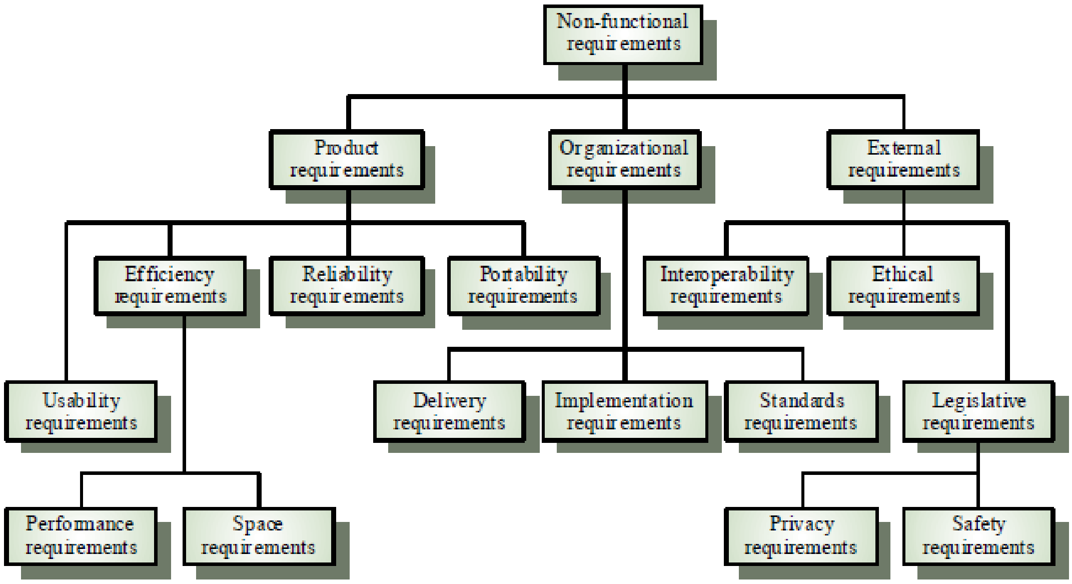

Figure 3.1 Non-Functional Requirement

Some Non-Functional Requirements are as follows:

- **Reliability:** The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a customer has revealed a couple of enhancements [22]. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.
- **Maintainability:** The system watching and upkeep should be fundamental and focus in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the employments are running without lapses.
- **Performance:** The framework will be utilized by numerous representatives all the while. Since the system will be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every

piece of its customers. For instance, if two test specialists are all the while attempting to report the vicinity of a bug, then there ought not to be any irregularity at the same time.

- **Portability:** The framework should to be effectively versatile to another framework. This is obliged when the web server, which s facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another framework.
- **Scalability:** The framework should be sufficiently adaptable to include new functionalities at a later stage. There should be a run of the mill channel, which can oblige the new functionalities.
- **Flexibility:** Flexibility is the capacity of a framework to adjust to changing situations and circumstances, and to adapt to changes to business approaches and rules. An adaptable framework is one that is anything but difficult to reconfigure or adjust because of diverse client and framework prerequisites. The deliberate division of concerns between the trough and motor parts helps adaptability as just a little bit of the framework is influenced when strategies or principles change.

3.2 ARDUINO OVERVIEW

Arduino is a popular programmable board used to create projects. It consists of a simple hardware platform as well as a free source code editor which has a “one click compiles or upload” feature. Hence it is designed in way that one can use it without necessarily being an expert programmer (Kushner 1987). Arduino offers an open-source electronic prototyping platform that is easy to use and flexible for both the software and hardware. Arduino is able to sense the environment through receiving input from several sensors.

It is also able to control its surrounding through controlling motors, lights and other actuators. The Arduino programming language that is based on the wiring and the Arduino development environment that is based on the processing are used to program the microcontroller found on the board (Banzi, 2005). Due to its open-source environment, one is able to easily write and upload codes to the I/O board. It is also worth to note that Arduino can be run on Linux, Mac OSX and Windows as its environment is written in Java

Figure 3.2 Arduino Uno logo

3.2.1 Arduino History

Arduino was released in 2005 by students from the Interaction Design Institute Ivrea (IDII) as a modest tool for Mac OSX and Windows. Since then, Arduino has been able to initiate an international-do-it yourself revolution at the electronics industry. The open-source microcontroller hardware has been designed in a way that it can easily interface with various sensors (registering user inputs) and driving the behaviors and responses of the external components such as speakers, motors, and LED (responding to the user inputs). The most important feature of Arduino is the ease of programmability hence users with little expertise are able to use it. This aspect has made Arduino one of the most popular tools of choice for designers and artists in creating interactive spaces and objects (Arduino Team).

3.2.2 Arduino Development

While discussing the development of Arduino, it is worth introducing a brief history of microcontrollers. A revolutionary leap in the computing industry was seen in the 1960s following the development of solid-state computers (including the IBM 1401), that used transistors to process its operations and a magnetic core memory for its storage (instead of vacuum tubes), and these enabled an increase in the compactness of the computer hardware. In addition, Jack Kilby's invention of integrated circuits in 1959 enabled circuits and transistors to be fused into tiny chips of semiconducting materials (like silicon) as well as further miniaturization of the computer component. The other crucial development made in the same decade was the high-level computer programming languages, written in symbolic languages such as plain English, and this made computer codes somehow easy to learn and read than the earlier machine languages that consisted of letters and numbers only. This development enabled individuals with few years of expertise to carry out the basic operations on a computer.

FORTRAN (for the scientific calculators) and COBOL (for business application) were the two main languages that were introduced in that period.

The microprocessor was one of the greatest innovations in the history of the modern computer in the 1970's. Initially, the microprocessor miniaturized all the hardware components of CPU to fit into one, tiny, integrated circuit, popularly known as the microchip.

The microchip became the major driving component of the microcontrollers including Arduino which is made up of a microchip, input/output hardware and memory storage hardware for sensors. The microprocessor, due to the small form factor, was incorporated into a surfeit of electronic devices ranging from personal computers to calculators and are still used up to date. More programming languages were also developed in the 1970s and 80s including C, C++ and Java for applications in science and business. (Massimo, 2005)

3.2.3 Arduino Evolution

Having looked at the evolution of microcontrollers, there have been recent incarnations of the microcontrollers that have been designed in a way to fulfill the needs of hobbyists and casual users who happen to have a limited technical knowledge. In other words, the microcontrollers have moved from the more complex requirements in the scientific, business or commercial fields. Before the invention of Arduino, the PIC microcontroller board that was introduced by general instruments in 1985 was one of the most used tools for the electronic enthusiasts. The reasons as to why the PIC microcontroller board was preferred were the speed and ease of its programming through simple languages including PBASIC. An additional reason was that it was able to store programs on a flash memory chip that enabled the instructions on the board to be reprogrammed or erased at will with an infinite number of possibilities. It also supported output devices such as LEDs and motors as well as input sensors. There are other popular boards for the hobbyists including BASIC Stamp and wiring which are some of the microcontroller boards that were designed for tangible media explorations and electronic art. The two boards share the advantages of ease of rapid prototyping and simplicity of programming.

It was in 2005 when the Arduino team was formed in Italy and it consisted of Barragan Massimo, David Cuartielles, Gianluca Marino, Dave Mellis and Nicholas Zambetti. The main goal of this team was to develop an electronic prototyping platform that would simplify the wiring platform and make it accessible to the non-technical users especially in the creative fields. The Arduino, therefore, incorporated several characteristics including a programming environment that is based on the processing language that was conceived by Casey Reas and Ben Fry and other artists and designers. Arduino also incorporated the ability to program its board using a standard USB connection with a low price point (Wheat, 2001).

3.2.4 Arduino Past and Present

Within its first 2 years of existence, Arduino achieved rapid success where more than 50, 000 boards were sold. By 2009, Arduino had more than 13 different incarnations with each having a specialized application. For instance, Arduino Mini was a miniature to be used in small interactive objectives, Arduino BT was built with Bluetooth capabilities, and Arduino Lilypad for wearable technologies projects. Today, the Arduino microcontroller is a popular prototyping platform across the world and it is a good example of how software and hardware technologies that were originally created for business, military or scientific applications have been repurposed so as to serve the needs of people developing projects in new media and arts and design.

3.2.5 What are Arduino Boards?

Arduino board is an open-source platform used to make electronics projects. It consists of both a microcontroller and a part of the software or Integrated Development Environment (IDE) that runs on your PC, used to write & upload computer code to the physical board. The platform of an Arduino has become very famous with designers or students just starting out with electronics, and for an excellent cause.

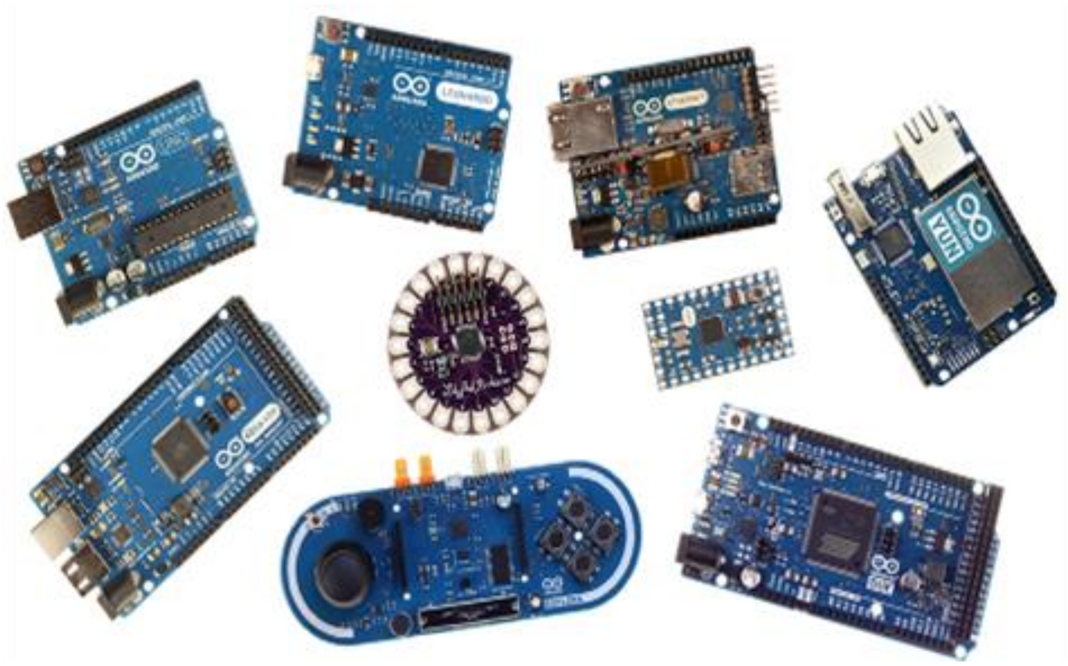

Figure 3.3: Types of Arduino Boards

3.2.6 Why Arduino Boards?

Arduino board has been used for making different engineering projects and different applications. The Arduino software is very simple to use for beginners, yet flexible adequate for advanced users. It runs windows, Linux and Mac.

Teachers and students in the schools utilize it to design low-cost scientific instruments to verify the principles of physics and chemistry. There are numerous other microcontroller platforms obtainable for physical computing. The Netmedia's BX-24, Parallax Basic Stamp, MIT's Handy board, Phidget and many others present related functionality.

Arduino also makes simpler the working process of microcontroller, but it gives some advantages over other systems for teachers, students and beginners.

- Inexpensive
- Cross-platform
- Simple, clear programming environment
- Open source and extensible software
- Open source and extensible hardware

3.2.7 Features of Arduino Boards.

Arduino Board	System Voltage	Processor	Memory	Digital I/O	Analogue I/O
Arduino Uno 	5V	16Mhz ATmega328	2KB SRAM, 32KB flash	14	6 inputs, 0 output
Arduino Due 	3.3V	84MHz AT91SAM3X 8E	96KB SRAM, 512KB flash	54	12 inputs, 2 outputs
Arduino Mega 	5V	16MHz Atmega2560	8KB SRAM, 256KB flash	54	16 inputs, 0 output
Arduino Leonardo 	5V	16MHz Atmega32u4	2.5KB SRAM, 32KB flash	20	12 inputs, 0 output
Arduino LilyPad 	2.7-5.5 V	Atmega 168v or 328v	1KB SRAM, 16KB Flash	14	6 inputs, 0 output
Arduino Nano 	5V	Atmel Atmega328	2KB SRAM, 32KB Flash	14	8 inputs, 0 output

Arduino FIO 	3.3V	Atmega328P	3.3KB SRAM, 32KB Flash	14	8 inputs, 0 output
Arduino Mini 	5v	Atmega328	2KB SRAM, 32KB Flash	14	8 inputs, 0 outputs
Arduino Pro Mini 	3.3V or 5V	Atmega328	2KB SRAM, 16KB Flash	14	6 inputs, 0 output

Table 3.1 Arduino Board Features

CHAPTER 4

SYSTEM DESIGN

CHAPTER 4

4.1 SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured.

Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. Until the 1990s, systems design had a crucial and respected role in the data processing industry. In the 1990s, standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design.

It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations. It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently. System Design focuses on how to accomplish the objective of the system.

The System Design Document is a required document for every project. It should include a high level description of why the System Design Document has been created, provide what the new system is intended for or is intended to replace and contain detailed descriptions of the architecture and system components of the system. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the developments of a system. The logical system design arrived at as a result of systems analysis is converted into physical design.

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. how to implement? It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate. In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.

4.2 SYSTEM ARCHITECTURE:

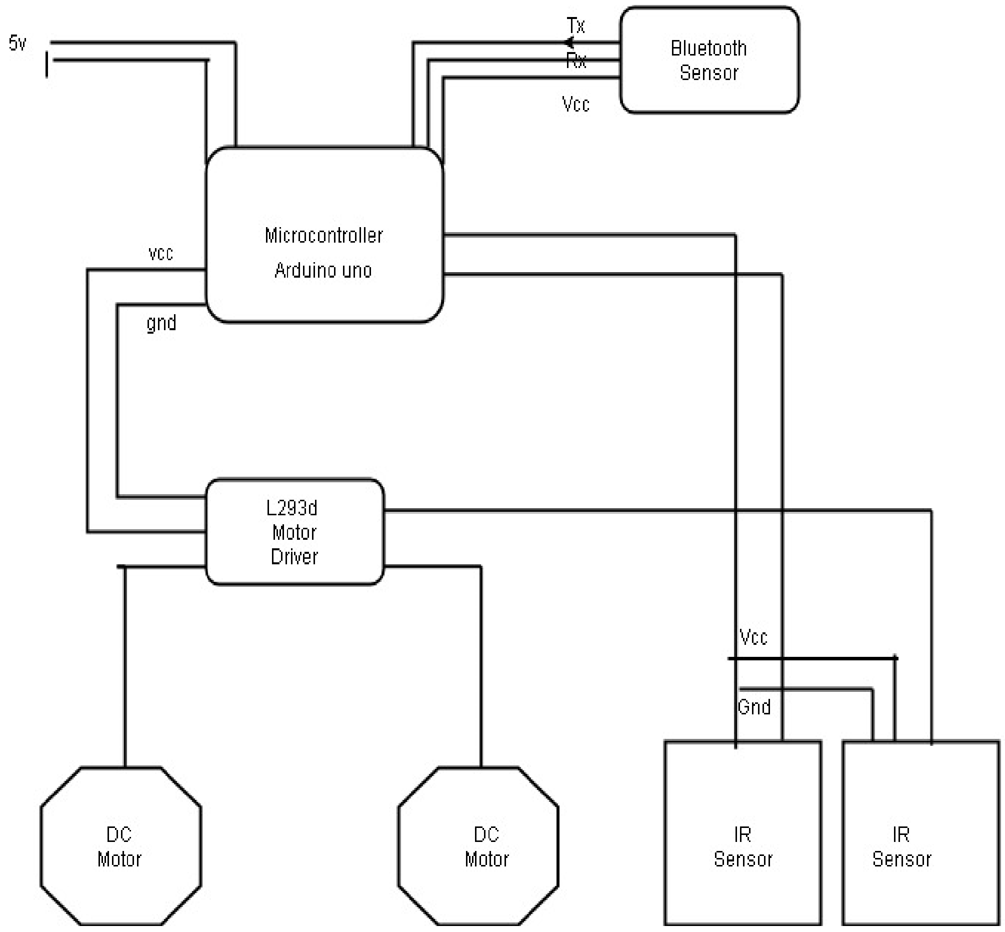

Figure 4.1:System Architecture

4.3 USE-CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Use-case diagram for owner:

• Initially, Owner starts the robot and then chooses between the automatic mode and manual mode.

• When anyone/anything comes in front of the IR Sensor, it senses and moves in another direction during avoidance mode.

• If in manual mode owner uses the voice control to make changes in the direction.

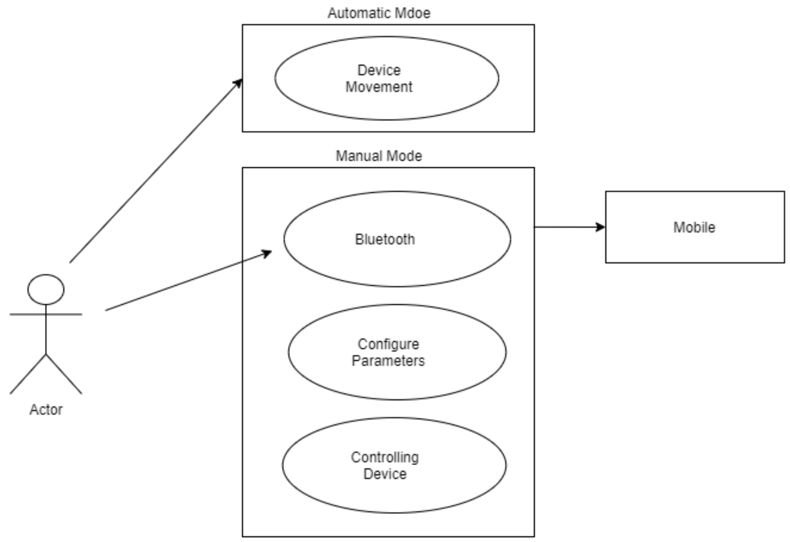

Figure 4.2: Use-case diagram

4.4 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

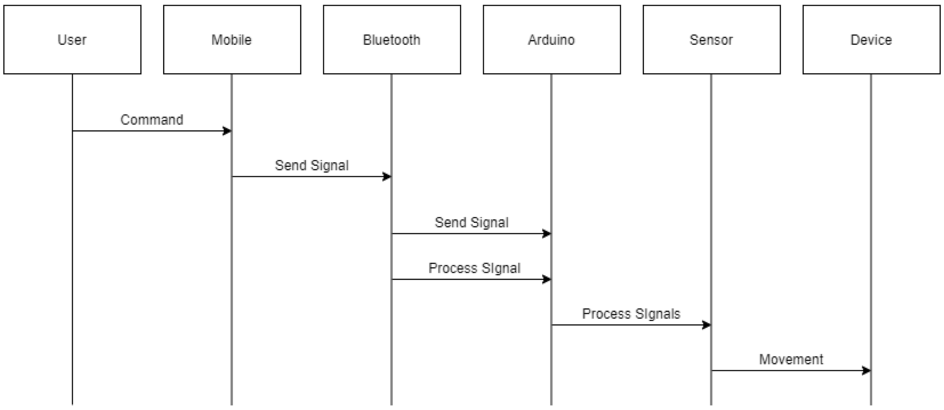

Figure 4.3: Sequence Diagram

4.5 ACTIVITY DIAGRAM

The Activity diagram shows the flow of activities or the actions that occur when user tries to interact with the system and tries to access the functionality provided by the system.

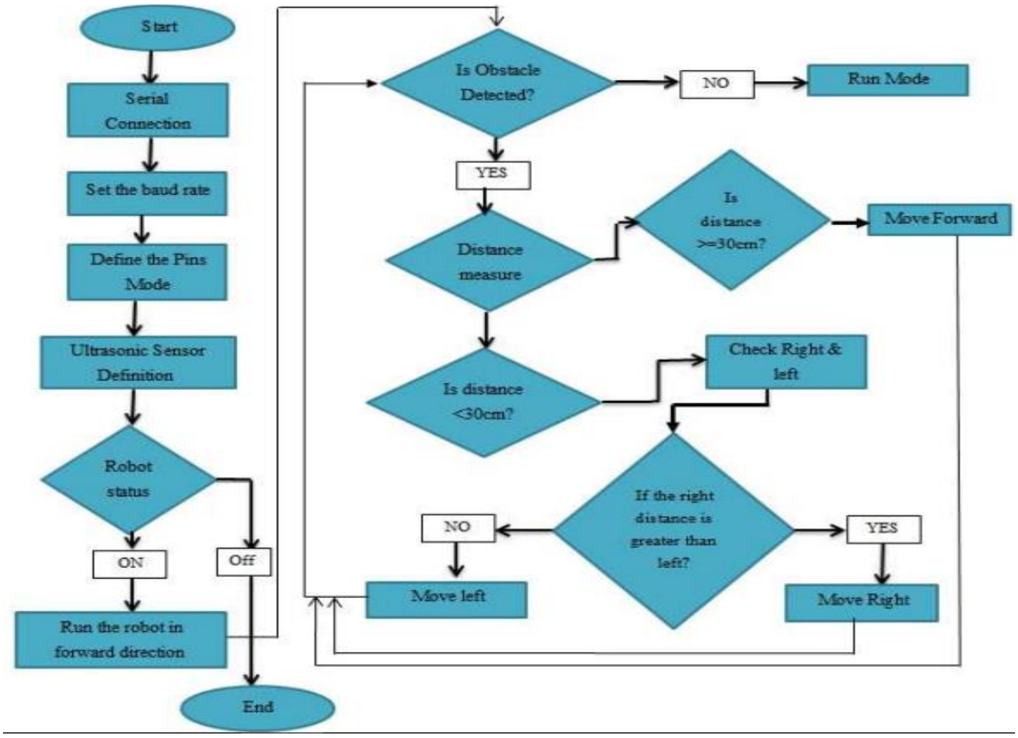

Figure 4.4: Activity Diagram

4.6 ARDUINO:

Arduino is an open source project that created microcontroller based kits interactive objects that can sense and control physical devices. Arduino is a prototype platform based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

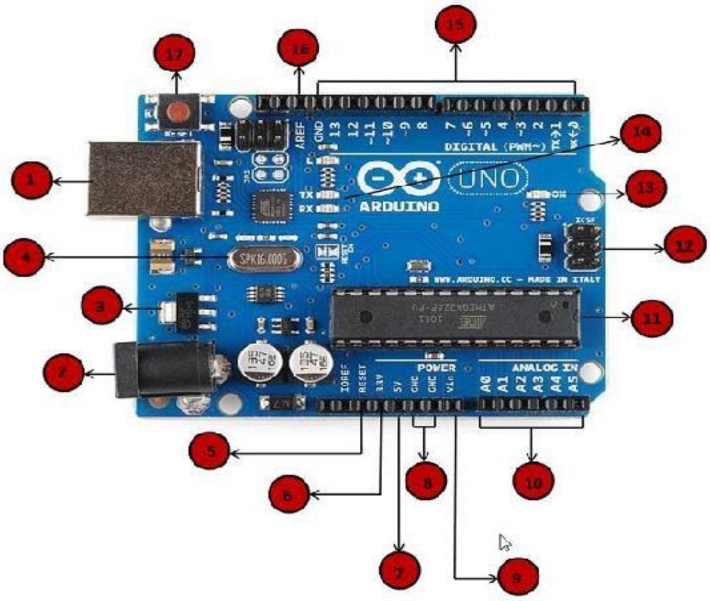

Figure 4.5 Arduino Uno

1- Power USB: Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection.

2- Power (Barrel Jack) Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack

3- Voltage Regulator: The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4- Crystal Oscillator: The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz

5, 17 - Arduino Reset:You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET.

6, 7, 8 ,9- Pins (3.3, 5, GND, Vin)

3.3V − Supply 3.3 output volt

5V − Supply 5 output volt

Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.

GND (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

Vin − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

10- Analog pins: The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11- Main-microcontroller: Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12- ICSP pin: Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13- Power LED indicator: This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14- TX and RX LEDs: On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led. The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15- Digital I/O: The Arduino UNO board has 14 digital I/O pins (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled ―~‖ can be used to generate PWM.

16- **AREF**: AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

4.3.1 Basic Functions of Arduino Technology

- Digital read pin reads the digital value of the given pin.
- Digital write pin is used to write the digital value of the given pin.
- Pin mode pin is used to set the pin to I/O mode.
- Analog read pin reads and returns the value.
- Analog write pin writes the value of the pin.
- Serial begins pin sets the beginning of serial communication by setting the rate of bit.

4.3.2 Advantages of Arduino Technology

- It is cheap.
- It comes with an open supply hardware feature that permits users to develop their own kit.
- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.

4.7 SENSORS

- **IR sensors:** An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measures only infrared radiation, rather than emitting it that is called as a passive IR sensor. Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These types of radiations are invisible to our eyes, that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, The resistances and these output voltages, change in proportion to the magnitude of the IR light received.

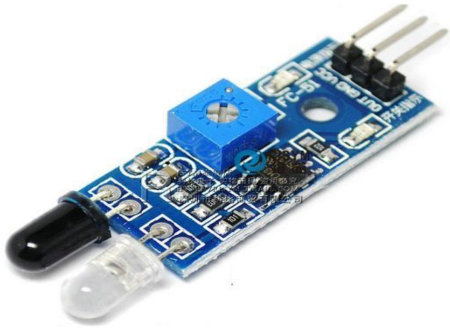

Figure 4.6 IR Sensor

Optical fire detectors as an example are the most commonly used ones in industrial applications since several decades and in private homes since some years. Nevertheless, severe drawbacks (sensitivity to phenomena like nuisance aerosols usually leading to false alarms) existed since optical smoke detectors are on the market. Recently several methods (multi-angle, multi-wavelengths) have been reported and implemented showing how to improve the optical scattering sensors for more robust operation.

CHAPTER 5

SYSTEM IMPLEMENTATION

CHAPTER 5

5.1 SYSTEM IMPLEMENTATION:

An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming languages.

To implement a system successfully, a large number of inter-related tasks need to be carried out in an appropriate sequence. Utilizing a well-proven implementation methodology and enlisting professional advice can help but often it is the number of tasks, poor planning and inadequate resourcing that causes problems with an implementation project, rather than any of the tasks being particularly difficult. Similarly with the cultural issues it is often the lack of adequate consultation and two-way communication that inhibits achievement of the desired results.

Developing “Voice Controlled Robot with Real Time Barrier Detection” was a major challenge for us. Various hardware and software used for the process to make an effective robot. The microcontroller used for this project is Arduino Uno. This microcontroller will help to coordinate all the activities of the robot. The microcontroller receives measured values from the two sensors in the form of analog and digitized them. It receives the signal to rotate the motors for movement of robot. The IR sensor will receive signal in binary for which will be read, converted and put into working by the microcontroller. The microcontroller will have predefined functions in the form of words which will be read by the voice recognition and convert it to meaningful signal and send it to the microcontroller.

5.2 HARDWARE IMPLEMENTATION:

- **Jump wire:** A **jump wire** (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering

Figure 5.1:Jump Wire Types

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

- **PC with Arduino Controller IDE:** Arduino Controller IDE helps to implement the software realization into the hardware and makes to hardware to follow the written instruction. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

- **ARM Microcontroller:** ARM, previously Advanced RISC Machine, originally Acorn RISC Machine, is a family of reduced instruction set computing (RISC) architectures for computer processors, configured for various environments. British company ARM Holdings develops the architecture and licenses it to other companies, who design their own products that implement one of those architectures—including systems-on-chips (SoC) and systems-on-modules (SoM) that incorporate memory, interfaces, radios, etc. It also designs cores that implement this instruction set and licenses these designs to a number of companies that incorporate those core designs into their own products.

- **USB:** USB was designed to standardize the connection of computer peripherals (including keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters) to personal computers, both to communicate and to supply electric power. It has largely replaced interfaces such as serial ports and parallel ports, and has become commonplace on a wide range of devices. USB connectors have replaced other types for battery chargers.

Figure 5.2:UART Dart Cable

- **IR sensor:** An object can be detected with an infrared system consisting of an infrared transmitter and a receiver. More in detail an IR transmitter, also known as IR LED, sends an infrared signal with a certain frequency compatible with an IR receiver which has the task to detect it. There are different kind of IR sensors for different type of application. IR technology is used, for example, in proximity sensors to detect a near object, in contrast sensors to find a path or in counting sensors to count objects.

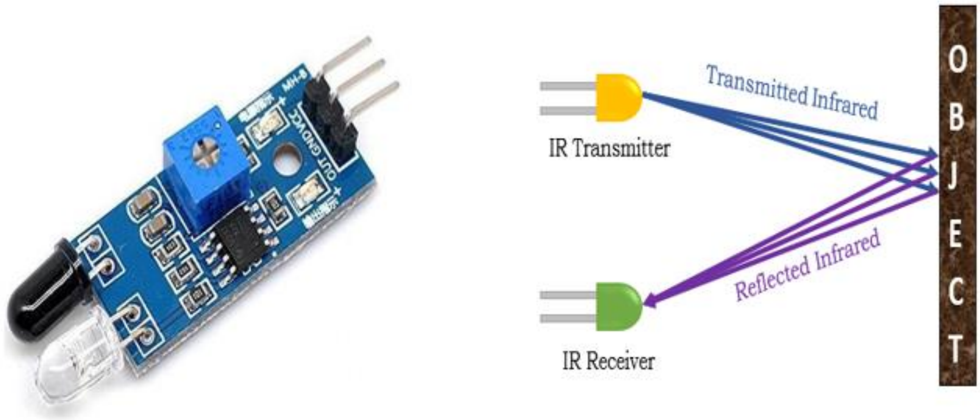

Figure 5.3: IR Sensor

- **Breadboard:** A breadboard, or protoboard, is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used when slicing bread. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

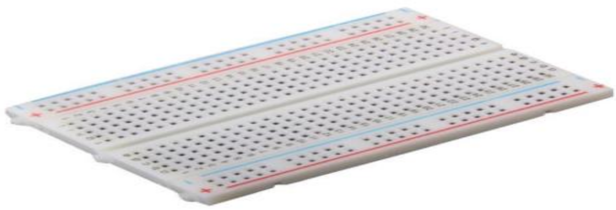

Figure 5.4: Bread Board

- **Chassis:** A load bearing framework for car was used to hold all the components together. load-bearing framework of an artificial object, which structurally supports the object in its construction and function. An example of a chassis is a vehicle frame, the underpart of a motor vehicle, on which the body is mounted; if the running gear such as wheels and transmission, and sometimes even the driver's seat, are included, then the assembly is described as a rolling chassis.

Figure 5.5: Chassis

- **L293d motor driver**: The L293D is a popular 16-Pin **Motor Driver IC**. As the name suggests it is mainly used to drive motors. A single **L293D IC** is capable of running two DC motors at the same time; also, the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Micron rollers like Arduino, PIC, ARM etc. this IC will be the right choice for you.

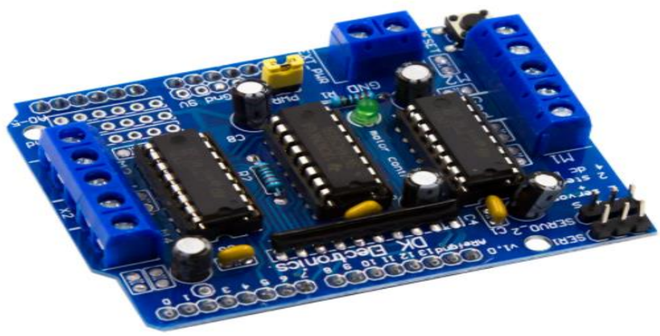

Figure 5.6: l293d motor driver

- **Bluetooth Sensor:** HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration. It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications. It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

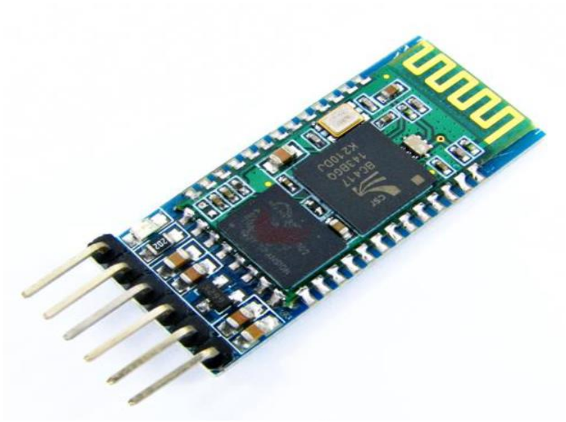

Figure 5.7: Bluetooth Sensor HC-05 Module

5.3 SOFTWARE IMPLEMENTATION

5.3.1 Arduino Programming:

Arduino programs are written in the Arduino Integrated Development Environment (IDE). Arduino IDE is a special software running on your system that allows you to write sketches (synonym for program in Arduino language) for different Arduino boards. The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language. After the sketch is written in the Arduino IDE, it should be uploaded on the Arduino board for execution.

The first step in programming the Arduino board is downloading and installing the Arduino IDE. The open-source Arduino IDE runs on Windows, Mac OS X, and Linux.

5.3.2 Block Based Coding:

Block-based coding utilizes a drag-and-drop learning environment, where programmers use coding instruction “blocks” to construct animated stories and games. It’s an entry-level activity, where kids can gain a foundation in computational thinking through visuals as opposed to coding that is based in text. When we mention “blocks,” it’s a way to describe the “chunks” or “pieces” of instructions a user is putting together in order to tell their creation what to do. For example, Scratch programming is block-based, and is one of the most popular options when it comes to agile software development.

5.3.3 MIT APP INVENTOR:

MIT App Inventor framework was used for block programming. MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create application software(apps) for two operating systems (OS): Android, and iOS, which, as of 8 July 2019, is in final beta testing. It is free and open-source software released under dual licensing: a Creative Commons Attribution ShareAlike 3.0 Unported license, and an Apache License 2.0 for the source code.

It uses a graphical user interface (GUI) very similar to the programming languages Scratch (programming language) and the StarLogo, which allows users to drag and drop visual objects to create an application that can run on android devices, while a App-Inventor Companion (The program that allows the app to run and debug on) that works on iOS running devices are still under development. In creating App Inventor, Google drew upon significant prior research in educational computing, and work done within Google on online development environments.

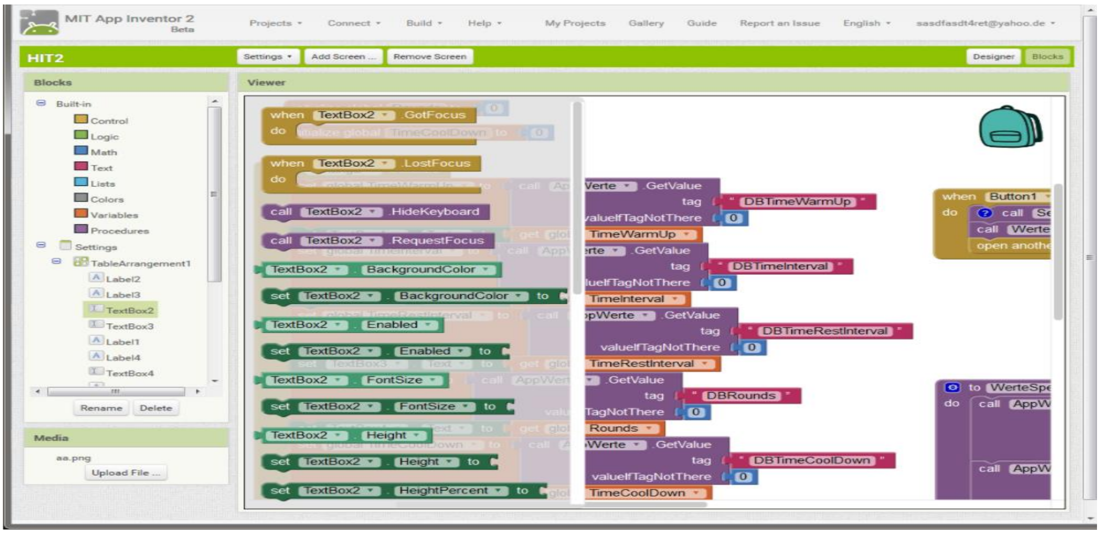

Figure 5.8: MIT App Inventor Framework

5.4 CODING AND SNAPSHOTS

5.4.1 Arduino Code

```
String voice;

int

M11 = 13, //Connect Motor 1 Blue wire To Pin #13

M12 = 12, //Connect Motor 1 Violet wire To Pin #12

M21 = 11, //Connect Motor 2 Blue wire To Pin #11

M22 = 10; //Connect Motor 2 Violet wire To Pin #10


//--------------------------Call A Function-------------------------------//

void forward() {

    digitalWrite(M11, HIGH);

    digitalWrite(M12, LOW);

    digitalWrite(M21, HIGH);

    digitalWrite(M22, LOW);

    }


void backward(){

    digitalWrite(M11, LOW);

    digitalWrite(M12, HIGH);

    digitalWrite(M21, LOW);

    digitalWrite(M22, HIGH);


}

void left(){

    digitalWrite(M11, LOW);
```

```
    digitalWrite(M12, LOW);
    digitalWrite(M21, HIGH);
    digitalWrite(M22, LOW);
    }
void right(){
    digitalWrite(M11, HIGH);
    digitalWrite(M12, LOW);
    digitalWrite(M21, LOW);
    digitalWrite(M22, LOW);
    }
void stay(){
    digitalWrite(M11, LOW);
    digitalWrite(M12, LOW);
    digitalWrite(M21, LOW);
    digitalWrite(M22, LOW);
    }

//-----------------------------------------------------------------------//
 void setup() {
  Serial.begin(9600);
  pinMode(M11, OUTPUT);
  pinMode(M12, OUTPUT);
  pinMode(M21, OUTPUT);
  pinMode(M22, OUTPUT);

}
//-----------------------------------------------------------------------//
```

```
 void loop() {
  while (Serial.available()){  //Check if there is an available byte to read
  delay(10); //Delay added to make thing stable
  char c = Serial.read(); //Conduct a serial read
  if (c == '#') {break;} //Exit the loop when the # is detected after the word
  voice += c; //Shorthand for voice = voice + c
  }
  if (voice.length() > 0) {
    Serial.println(voice);
//----------------------------------------------------------------------//
  //----------Control Multiple Pins/ LEDs----------//
       if(voice == "*forward") {forward();}  //Move Robot Forward (Call Function)
  else if(voice == "*backward"){backward();} //Move Robot Backward (Call Function)
  else if(voice == "*left"){left();} //Move Robot Left (Call Function)
  else if(voice == "*right"){right();} //Move Robot right (Call Function)
  else if(voice == "*stop"){stay();} //Stop Robot (Call Function)



//----------------------------------------------------------------------//
voice="";}} //Reset the variable after initiating
```

Obstacle avoider

```
void setup()
{
pinMode( 8 , INPUT);
pinMode( 9 , INPUT);
pinMode( 10 , OUTPUT);
```

```
pinMode( 11 , OUTPUT);

pinMode( 12 , OUTPUT);

pinMode( 13 , OUTPUT);

}


void loop()

{

if ((digitalRead( 9)==1) && (digitalRead( 8)==0))

{

digitalWrite( 13 , LOW );

digitalWrite( 12 , LOW );

digitalWrite( 11 , HIGH );

digitalWrite( 10 , LOW );

}


else if ((digitalRead( 8)==1) && (digitalRead( 9)==0))

{

digitalWrite( 11 , LOW );

digitalWrite( 10 , LOW );

digitalWrite( 13 , HIGH );

digitalWrite( 12 , LOW );

}


else if ((digitalRead( 8)==1) && (digitalRead( 9)==1))

{

digitalWrite( 13 , LOW );

digitalWrite( 12 , LOW );
```

```
digitalWrite( 11 , LOW );

digitalWrite( 10 , LOW );

}

else if ((digitalRead( 8)==0) && (digitalRead( 9)==0))

{

digitalWrite( 13 , HIGH );

digitalWrite( 12 , LOW );

digitalWrite( 11 , HIGH );

digitalWrite( 10 , LOW );

}

}
```

5.4.2 Block Code

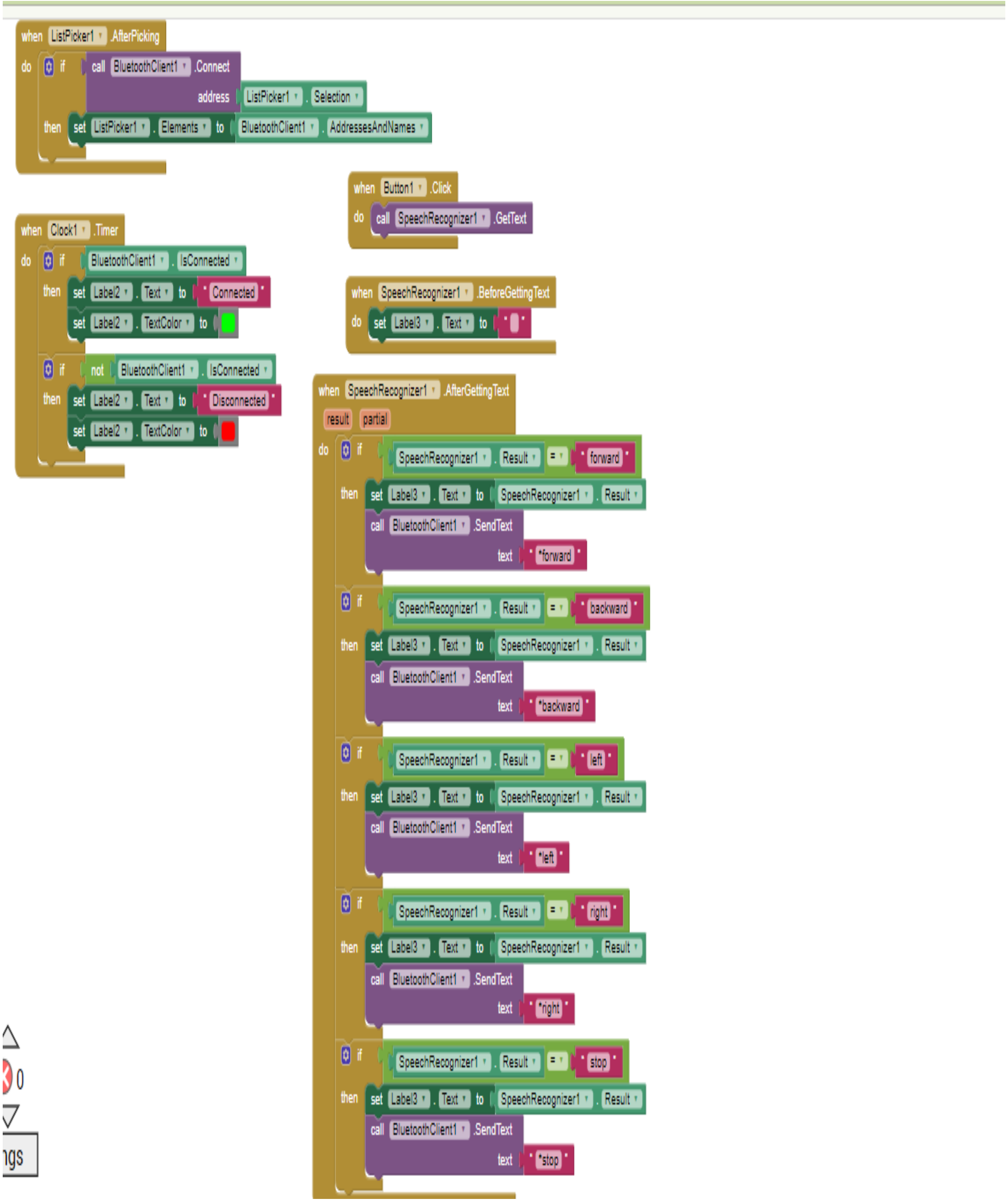

Figure 5.9: MIT App Inventor Block Code

CHAPTER 6

SYSTEM TESTING

CHAPTER 6

6.1 SYSTEM TESTING:

In this chapter, an overview of testing is provided to verify the correctness and the functionality of the system. Software testing is the process of analyzing a software item to detect the differences between the existing and required conditions and to evaluate the features of software item. Software testing is an activity that should be done throughout the development process. Software testing is a task intended to detect defects in software by contrasting a computer program's expected results with its actual results for given set of inputs.

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

6.2 TESTING PRINCIPLE

Before applying methods to design effective test cases, a software engineer must understand the basic principle that guides software testing. All the tests should be traceable to customer requirements.

6.3 TESTING METHODS

There are different methods that can be used for software testing. They are,

6.3.1 Black-Box Testing

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

6.3.2 White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

6.4 LEVELS OF TESTING

There are different levels during the process of testing. Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

6.4.1 Functional Testing:

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. There are five steps that are involved while testing an application for functionality.

6.4.2 Non-functional Testing:

This section is based upon testing an application from its non-functional attributes. Non-functional testing involves testing software from the requirements which are non-functional in nature but important such as performance, security, user interface, etc. Testing can be done in different levels of SDLC. Few of them are

6.4.3 Unit Testing:

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually. The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality. Test cases and results are shown in the Tables.

Test Case : -	UTC-1
Name of Test: -	Voice Recognition Test
Items being tested: -	Cross platform for voice testing and motor movement
Sample Input: -	Voice command Right
Expected output: -	The Robot should turn and move in right direction
Actual output: -	Robot turn Right Successfully
Remarks: -	Pass.

Table 6.1: User Test Case 1

Test Case : -	UTC-2
Name of Test: -	Voice Recognition Test
Items being tested: -	Cross platform for voice testing and motor movement
Sample Input: -	Voice command left
Expected output: -	The Robot should turn and move in left direction
Actual output: -	Robot turned Right Successfully
Remarks: -	Pass.

Table 6.2: User Test Case 2

Test Case : -	UTC-3
Name of Test: -	Voice Recognition Test
Items being tested: -	Cross platform for voice testing and motor movement
Sample Input: -	Voice command Forward
Expected output: -	The Robot should move in forward direction
Actual output: -	Robot moved forward Successfully
Remarks: -	Pass.

Table 6.3: User Test Case 3

Test Case : -	UTC-4
Name of Test: -	Voice Recognition Test
Items being tested: -	Cross platform for voice testing and motor movement
Sample Input: -	Voice command Backward
Expected output: -	The Robot should move in Backward direction
Actual output: -	Robot moved Backward Successfully
Remarks: -	Pass.

Table 6.4: User Test Case 4

Test Case: -	UTC-5
Name of Test: -	Auto Barrier Detection
Items being tested: -	IR Sensor integration with Arduino Uno

Sample Input: -	Barrier Place at the front of the robot
Expected output: -	The Robot should detect the Barrier in Real time and avoid the barrier and move toward non-barrier path.
Actual output: -	Robot successfully detected and avoided the barrier moving toward non-barrier path.
Remarks: -	Test Passed

Table 6.5: User Test Case 5

6.4.4 Integration Testing

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

1.Bottom-up Integration

This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

2.Top-down Integration

In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations. Table 6.3.2 shows the test cases for integration testing and their results

Test Case : -	ITC-1

Name of Test: -	Barrier Detection and Voice Recognition Command
Items being tested: -	Barrier Detection and voice recognition
Sample Input: -	Barrier Infront of the robot
Expected output: -	Should avoid barrier automatically or with voice command
Actual output: -	Functioned Properly
Remarks: -	Test Passed

Table 6.6: Integration Testing 1

6.4.5 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is important because of the following reasons:

System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.

The application is tested thoroughly to verify that it meets the functional and technical specifications.
The application is tested in an environment that is very close to the production environment where the application will be deployed.

System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

Test Case: -	STC-1
Name of Test: -	System Testing
Items being tested: -	IR Sensors, Voice Recognition System and Android App
Sample Input: -	Barrier plus voice command and manual command from android app
Expected output: -	The robot should detect and avoid barrier either automatically or manually through voice command or through android command
Actual output: -	Functioned Properly
Remarks: -	Test Passed

Table 6.7: System Testing 1

RESULTS

RESULTS

The figure shows the interfacing of components which are used in this system. Here we have a complex overall hardware implementation such that a possible barrier is continuously monitored by IR sensor. If the sensor figures out possible barrier, it will try to turn to optimize path with less or no barrier. The integration also involves the voice command integration with our mobile phone.

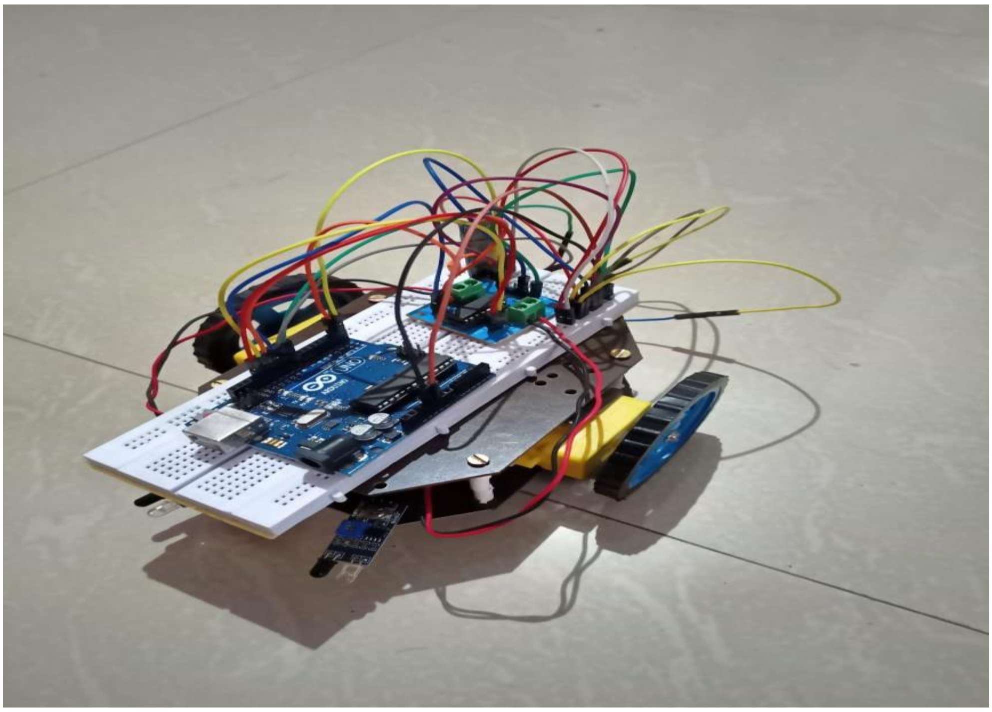

Figure 7.1 Overall Robot

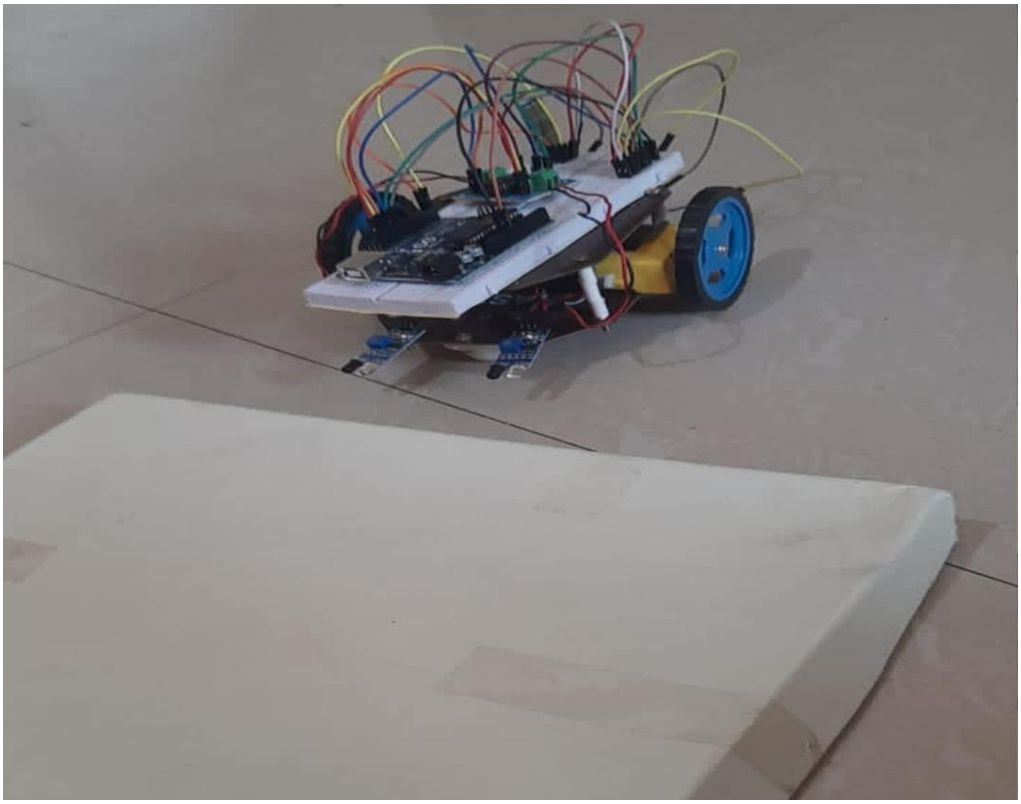

Figure 7.2 In front of obstacle

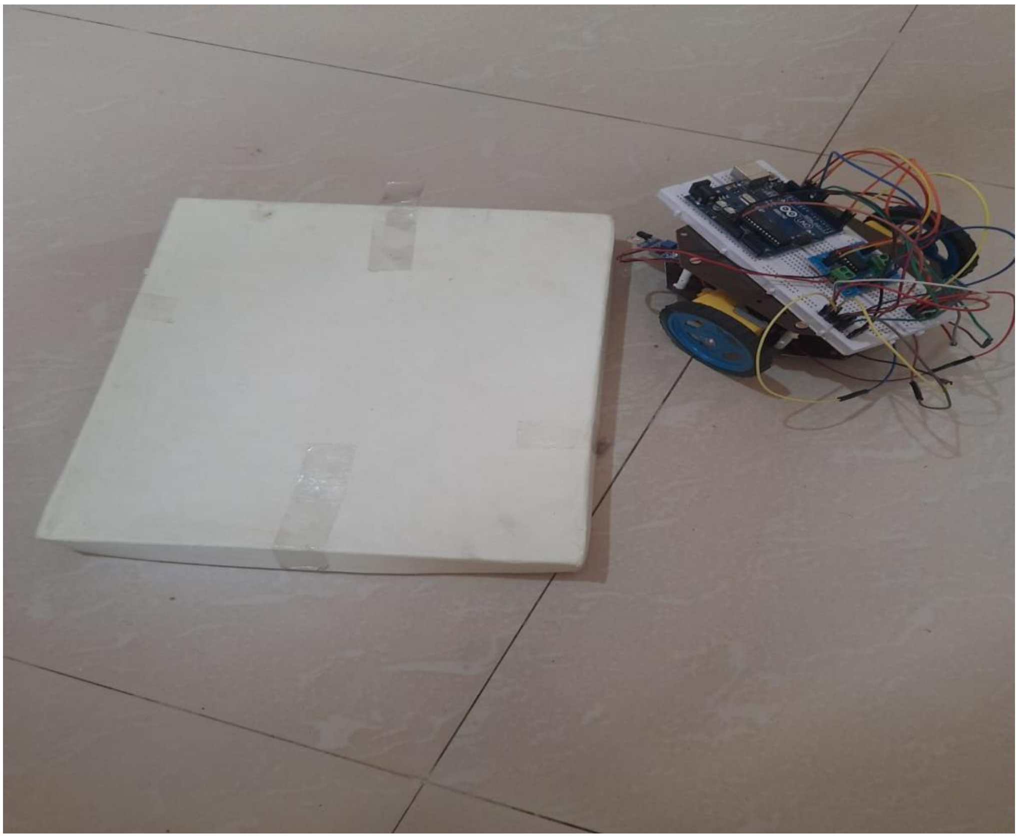

Figure 7.3 Avoiding obstacle

If barrier is detected by the IR sensor:

- "1" will be returned by the IR sensor to the Arduino.
- "1" is interpreted as high in the program.
- This high will be read by the respective if clause.
- If clause contains the condition for all the scenarios.
- First Scenario: If left is high and right is low, it will move to right direction.
- Second Scenario: If left is low and right is right, it will move to left direction.
- Third Scenario: If left is high and right is high, it will stop.
- Forth Scenario: If left is low and right is low, it will keep on moving in the forward direction.

Figure 7.4 Application interface

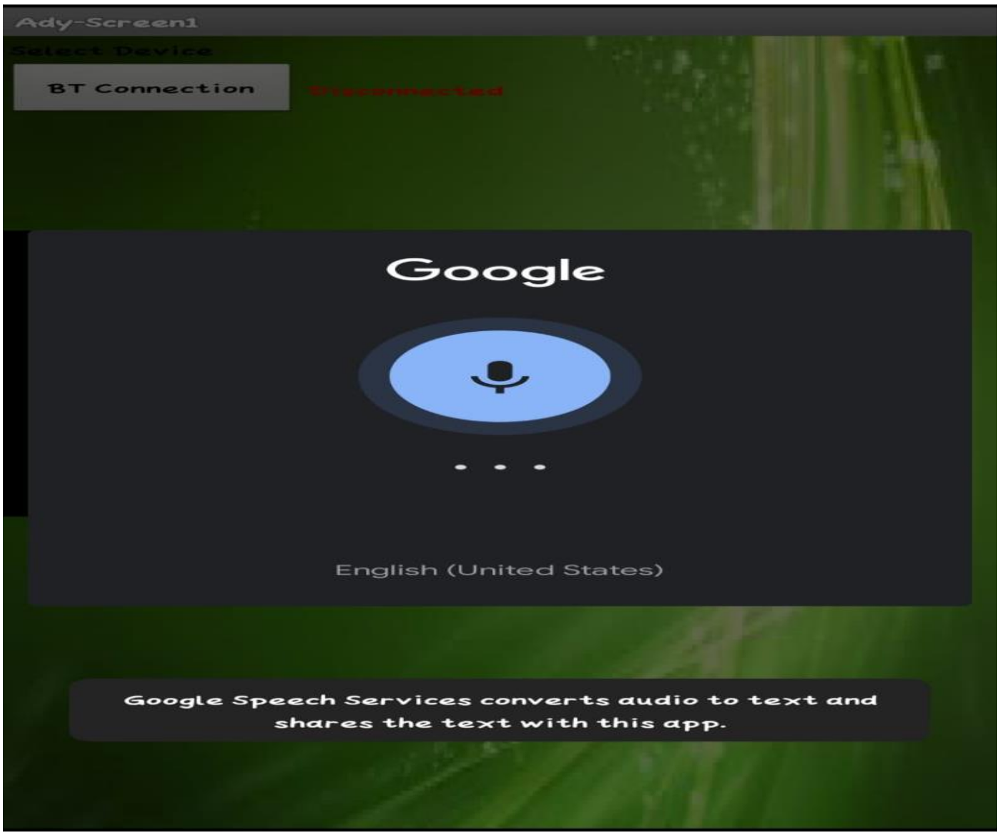

Figure 7.5 Voice interface

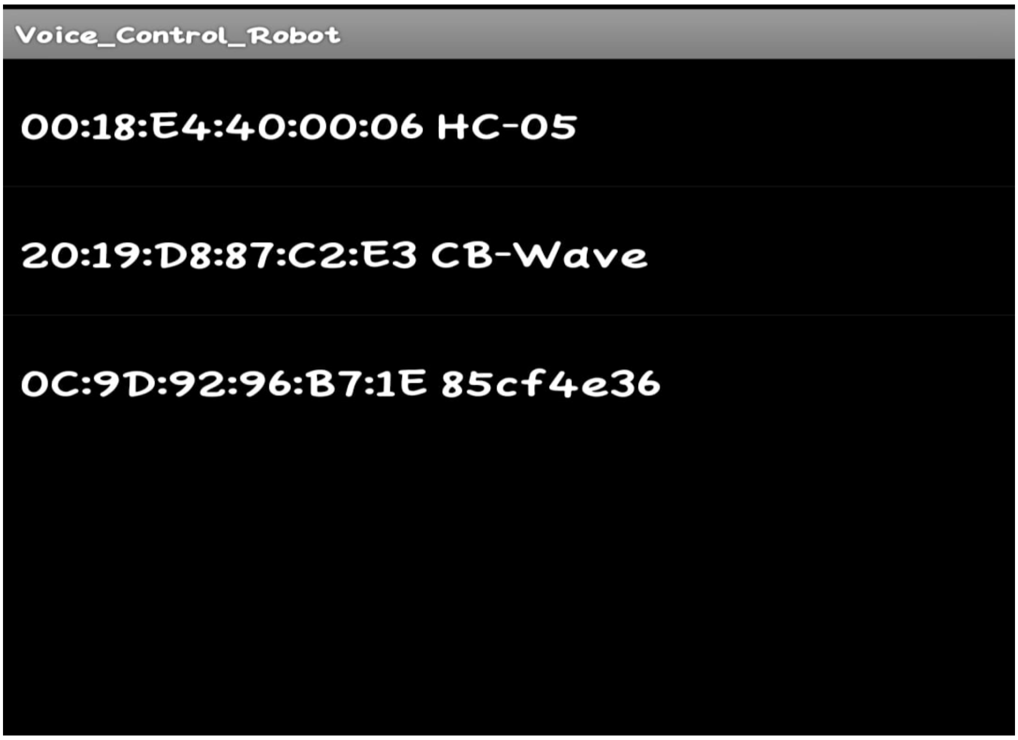

Figure 7.6 Bluetooth interfac

CONCLUSION

The system can be used in the car, buses and other vehicles to make it more sophisticated automated system, which will drastically reduce the hazards of accidents and also make people life easier in many ways. The system reduces accidents as it can detect any obstacle in which to apply the necessary action to prevent the accidents from happening. It makes the vehicle automated so people can work or do any other activity while the car drives them to their destination safely on their own. Voice automated system will also help in different ways like for example; if a person is stuck in rain, they can just call their car to the desired location so as they won't get wet. This is one of the simplest examples. In this way the system can change people's life for the better. The best implementation for this project will be to integrate it into a wheelchair for differently abled people. The person on the wheelchair can control their movements by either voice control or pressing buttons on the app. If they cannot do that then we have the automatic obstacle avoidance system installed. The chair will move automatically by avoiding obstacles.

Future Enhancement

- Camera can be integrated in the system and the tight spots can be seen hence it will be more effective in disaster rescue operations.
- By upgrading the wheels to track wheels provides better stability, control and response.
- AI and machine learning approaches can be integrated with the system for making it more smart.
- Through distributive learning and path routing algorithm, this robot can be implemented into field action for sending in aid or getting information out the affected area.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. J. Alok Kumar Gupta, "IOT based Electrical Device Surveillance and Control System," *IEEE*, p. 1, 2019.

[2] T. Kramp, "Introduction to the Internet of Things," in *Enabling Things To Talk*, Springer, 2013, pp. 1-10.

[3] S. Ranger, "zdnet.com," zdnet, 3 Feb 2020. [Online]. Available: https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/. [Accessed 20 July 2021].

[4] G. D. Maayan, "The IoT Rundown For 2020: Stats, Risks, and Solutions," p. 1, 2020.

[5] K. Ikeuchi, Computer Vision A Reference Guide, Springer, 2016.

[6] T. D. Olin KE, "Autonomous cross-country navigation: an integrated perception and planning system," *IEEE Expert,* pp. 16-30, 1991.

[7] M. M. K. L. Schroeder P., "National Survey on Distracted Driving Attitudes and Behaviors," *Google Scholar,* no. 2012, 2012.

[8] "With 53 road crashes per hour, road accidents in India continue to be highest in world," 16 Feb 2021.

[9] M. T, "International Journal of Mechanical, Aerospace, Industrial, Mechatronic and," no. 3, pp. 267-272, 2009.

[10] B. S. B. V. a. J. R. G. Hubballi P, "Indian J. Sci. Res," no. 12, pp. 155-8.

[11] S. B, "A Guide to IR/PIR Sensor Set-up and Testing," A Guide to IR/PIR Sensor Set-up and Testing.

[12] Z. Saleh, "Artificial Intelligence Definition, Ethics and Standards," *Research Gate*, 2019.

[13] G. V. Natalia Shmatko, "Bringing the skill gap in Robotics: Global and National Environment," 2020.

[14] BBVA'S OpenMind Project, "A Decade of Transformation in Robotics," in *Towards a New Enlightenment? A Transcendent Decade*, BBVA, 2020.

[15] F. M. Mordechal Ben-Air, Robotics and Their Applications, Springer, 2017.

[16] T. Harris, "How Robots Work," How Robots Work, 16 April 2002. [Online].

[17] M. FélixIngrand, "Deliberation for autonomous robots: A survey," *Science Direct*, pp. 10-44, 2017.

[18] F. V. C. L.-A. Francisco Rubio, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems (IJARS),* 2019.

[19] C. E. A. Project, "System Requirements for STEWARDS," 2019.

[20] wikipedia, "Functional Requirement," wikipedia, 10 june 2021. [Online]. Available: https://en.wikipedia.org/wiki/Functional_requirement. [Accessed 15 July 2021].

[21] altexsoft, "alexsoft software and engineering," alexsoft, May 2018. [Online]. [Accessed July 2021].

[22] S. V. Babitha. R, "FTFT : Fault Tolerance with Fragmentation Technique for Cloud Storage," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* p. 6, 2019.