MULTI FUNCTION ROBOT CAR

A Main Project Report

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

S. SWATHI	**(19X41A0446)**
CH. RAGHUNADH REDDY	**(19X41A0409)**
N. JESWITHA	**(19X41A0434)**
V. SAI KUMAR	**(19X41A0452)**

Under the guidance of

Ms V. Sri Lakshmi

Assistant Professor

SRK INSTITUTE OF TECHNOLOGY

Enikepadu, Vijayawada-521108

Andhra Pradesh

April 2023

CERTIFICATE

This is to certify that the project report entitled "**MULTI FUNCTION ROBOT CAR"** is a bonafied record of work carried out by **S.SWATHI (19X41A0446), CH. RAGHUNADH REDDY (19X41A0409), N. JESWITHA (19X41A0434), V. SAI KUMAR (19X41A0452)** under my guidance and supervision in partial fulfilment of the requirements for award of the degree of Bachelor of Technology in **Electronics and Communication Engineering of Jawaharlal Nehru Technological University, Kakinada.**

Ms V. Sri Lakshmi	**Dr S. Sri Gowri**
(Project Guide)	(Head of the Department)

ACKNOWLEDGEMENT

We are glad to express our deep sense of gratitude to **Ms. V. Sri Lakshmi**, Assistant Professor, Electronics and Communication Engineering for her valuable guidance, motivation, constant inspiration and above all for her ever-co-operating attitude that enabled us in bringing this thesis in the present form.

We express our heartful gratitude to our professor and Head of the Department, Electronics and Communication Engineering **Dr. S. Sri Gowri**, for her great help and encouragement in doing our project successfully.

We also express our gratitude to our principal **Dr. M. Ekambaram Naidu**, for his encouragement and facilities during project. We thank one and all who have rendered help to us directly or indirectly in the completion of this work.

S. Swathi	**(19X41A0446)**
Ch. Raghunadh Reddy	**(19X41A0409)**
N. Jeswitha	**(19X41A0434)**
V. Sai Kumar	**(19X41A0452)**

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Robots have long borne the potential to bridge the gap between the cybernetic world and the physical world. Robots were originally developed for industrial use, such as in factories, but a new generation of robots has been designed to achieve symbiosis with humans. This Robot car can perform three operations. The four modes of operations are Bluetooth controlled Robot Car, Obstacle Avoiding Robot Car, and Human following Robot Car. Bluetooth is an open standard specification for a radio frequency (RF)-based, short-range connectivity technology that promises to change the face of computing and wireless communication. In the Bluetooth Controlled Car, it can go Front, Back, Left and Right. This Bluetooth Controlled Robot Car can Controlled by Android Application. The Arduino board is responsible for the motion of robot vehicle. The Obstacle Avoiding Robot Car Can find the Obstacles using Ultrasonic Sensor. The Human Follower Robot Car can Follow the Human using the Ultrasonic Sensor. The controlling device of the whole system is a Microcontroller. These Functions are programmed and controlled by Arduino Microcontroller.

CHAPTER 1
INTRODUCTION

1.1 Introduction to Robotics

A Robot is a machine that is usually designed to reduce the amount of human work where it is applicable. It is an analogy for any machine that is controlled by man varying from a simple toy to heavy machinery. Robots have even replaced humans in performing various tasks that they are unable to perform due to physical disability, size limitation or extreme environment . It is usually developed to reduce risk factor for human work and increase comfort of any worker. Besides this, it helps to save our time by increasing its efficiency manifolds. Our modern civilization is going towards autonomous work day by day for reducing unnecessary human efforts on easy matters and ensuring more comfort and safety for the concentration on other important works. We have designed a robot that has several tasks to perform besides following a line. We've had various encounters in our day to day life at the thought of which only one thing strikes our mind-what if this could be done autonomously? There are many tasks in a human's life that are quite time consuming with little output in return. The answer to this problem is automation. In our project we propose a robot design which is multi featured and on a relatively low cost. The line following mechanism can be applied to various places where one needs to monitor things.

Robotics is a branch of engineering and science that includes electronics engineering, mechanical engineering and computer science and so on. This branch deals with the design, construction, use to control robots, sensory feedback and information processing. These are some technologies which will replace humans and human activities in coming years. These robots are designed to be used for any purpose but these are using in sensitive environments like bomb detection, deactivation of various bombs etc. Robots can take any form but many of them have given the human appearance. The robots which have taken the form of human appearance may likely to have the walk like humans, speech, cognition and most importantly all the things a human can do. Most of the robots of today are inspired by nature and are known as bio-inspired robots. Robotics is that branch of engineering that deals with conception, design, operation, and manufacturing of robots. There was an author named Issac Asimov, he said that he was the first person to give robotics name in a short story composed in 1940's. In that story, Issac suggested three principles about how to guide these types of robotic machines. Later on, these three principles were given the name of Issac's three laws of Robotics. These three laws state that:

- Robots will never harm human beings.
- Robots will follow instructions given by humans with breaking law one.
- Robots will protect themselves without breaking other rules.

1.2 Characteristics

There are some characteristics of robots given below:

- **Appearance:** Robots have a physical body. They are held by the structure of their body and are moved by their mechanical parts. Without appearance, robots will be just a software program.
- **Brain:** Another name of brain in robots is On-board control unit. Using this robot receive information and sends commands as output. With this control unit robot knows what to do else it'll be just a remote-controlled machine.
- **Sensors:** The use of these sensors in robots is to gather info from the outside world and send it to Brain. Basically, these sensors have circuits in them that produces the voltage in them.
- **Actuators:** The robots move and the parts with the help of these robots move is called Actuators. Some examples of actuators are motors, pumps, and compressor etc. The brain tells these actuators when and how to respond or move.
- **Program:** Robots only works or responds to the instructions which are provided to them in the form of a program. These programs only tell the brain when to perform which operation like when to move, produce sounds etc. These programs only tell the robot how to use sensors data to make decisions.
- **Behaviour:** Robots behavior is decided by the program which has been built for it. Once the robot starts making the movement, one can easily tell which kind of program is being installed inside the robot.

1.3 Types of Robots

These are the some types of robots given below:

- **Articulated:** The feature of this robot is its rotary joints and range of these are from 2 to 10 or more joints. The arm is connected to the rotary joint and each joint is known as the axis which provides a range of movements.
- **Cartesian:** These are also known as gantry robots. These have three joints which use the Cartesian coordinate system i.e x, y, z. These robots are provided with attached wrists to provide rotatory motion.
- **Cylindrical:** These types of robots have at least one rotatory joints and one prismatic joint which are used to connect the links. The use of rotatory joints is to rotate along the axis and prismatic joint used to provide linear motion.
- **Polar:** These are also known as spherical robots. The arm is connected to base with a twisting joint and have a combination of 2 rotatory joint and one linear joint.
- **Scara:** These robots are mainly used in assembly applications. Its arm is in cylindrical in design. It has two parallel joints which are used to provide compliance in one selected plane.
- **Delta:** The structure of these robots are like spider-shaped. They are built by joint parallelograms that are connected to the common base. The parallelogram moves in a dome-shaped work area. These are mainly used in food and electrical industries.

Scope and limitations of robots: The advance version of machines are robots which are used to do advanced tasks and are programmed to make decisions on their own. When a robot is designed the most important thing to be kept in mind is that What the function is to be performed and what are the limitations of the robot. Each robot has a basic level of complexity and each of the levels has the scope which limits the functions that are to be performed. For general basic robots, their complexity is decided by the number of limbs, actuators and the sensors that are used while for advanced robots the complexity is decided by the number of microprocessors and microcontroller used. As increasing any component in the robot, it is increasing the scope of the robot and with every joint added, the degree of the robot is enhanced.

Advantages: The advantages of using robots are given below:

- They can get information that a human can't get.
- They can perform tasks without any mistakes and very efficiently and fast.
- Maximum robots are automatic, so they can perform different tasks without needing human interaction.
- Robots are used in different factories to produce items like plane, car parts etc.
- They can be used for mining purposes and can be sent to earth's madrid.

Disadvantages: The disadvantages of using robots are given below:

- They need the power supply to keep going. People working in factories may lose their jobs as robots can replace them.
- They need high maintenance to keep them working all day long. And the cost of maintaining the robots can be expensive.
- They can store huge amount of data but they are not as efficient as our human brains.
- As we know that robots work on the program that has been installed in them. So other than the program installed, robots can't do anything different.
- The most important disadvantage is that if the program of robots comes in wrong hands they can cause the huge amount of destruction.

Applications: Different types of robots can performs different types of tasks. For example, many of the robots are made for assembly work which means that they are not relevant for any other work and these types of robots are called Assembly Robots. Similarly, for seam welding many suppliers provide robots with their welding materials and these types of robots are known as Welding Robots. While on the other hand many robots are designed for heavy-duty work and are known as Heavy Duty Robots. There are some applications given below:

- Caterpillar plans which is aiming to develop remote-controlled machines and are expecting to develop heavy robots by 2021.
- A robot can also do Herding task.

- Robots are increasingly been used more than humans in manufacturing while in auto-industry there are more than half of the labors are “Robots”.
- Many of the robots are used as Military Robots.
- Robots have been used in cleaning up of areas like toxic waste or industrial wastes etc.
- Agricultural robots.
- Household robots.
- Domestic robots.
- Nano robots.
- Swarm robots.

Advantages:

1. Increased Efficiency: Robots can work 24/7 without getting tired, leading to increased productivity and efficiency.
2. Improved Accuracy: Robots are capable of performing tasks with high precision and accuracy, reducing errors and improving quality.
3. Increased Safety: Robots can perform tasks that are dangerous for humans, improving overall safety in the workplace.
4. Reduced Labor Costs: The use of robots can lead to reduced labor costs, as robots can perform tasks more cheaply than human workers.

Disadvantages:

1. Initial Cost: Implementing and maintaining a robotics system can be expensive, especially for small and medium-sized businesses.
2. Job Losses: The increased use of robots may result in job losses for human workers, particularly in industries where manual labor is prevalent.
3. Limited Capabilities: Robots are still limited in their capabilities compared to human workers and may not be able to perform tasks requiring dexterity or creativity.
4. Maintenance Costs: Robots require regular maintenance and repair, which can be time-consuming and expensive.

1.4 System Overview

The project consists of three main parts- Bluetooth Controlled robot, Human follower robot and Obstacle avoider. The Bluetooth module helps us in forming an interaction between our robot and the android phone. Connections are made on the Arduino with the Bluetooth module and it is interfaced with the phone using an android application. Human following robot consists ultrasonic sensor which help to follow the object, this is similar to the obstacle avoiding robot only but opposite in the working. The obstacle avoider part of the robot is handled using Ultrasonic sensor.

CHAPTER 2
LITERATURE REVIEW

The main aim is to design the cost-efficient Bluetooth-controlled robot car for material handling which is eco-friendly. Many researchers had developed robotic systems to reduce human efforts and described their technologies. These robotic designs were controlled by software programs. On the other hand, the robotic system was developed using 8051 microcontroller, Bluetooth module. In this project we are designing communicating based robotic platform using Android and Bluetooth. This system is used to communicate the data wirelessly.

Obstacle avoidance robot using arduino has been designed and developed. In this we are creating autonomous robot which intelligently detects the obstacle in its path and navigates according to the actions that user set for it. So this system provides an alternate way to the existing system by replacing skilled labor with robotic machinery, which in turn can handle more patients in less cost.

Human following is a technique used by robot and autonomous vehicles to follow a human within a specific range. The human following robot is an automobile system that has ability to recognize obstacle, move and change the robot's position toward the subject in the best way to remain on its track. This project uses arduino, motors different types of sensors to achieve its goal.

CHAPTER 3

HISTORY OF ROBOTICS

3.1Defintions of Robotics

i. A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of task.
ii. A reprogrammable manipulator device.
iii. An automatic device that performs functions normally ascribed to humans or a machine in the form of a human.

3.2 What are Robots Made of?

Robots have 3 main components:

1) Brain - usually a computer

2) Actuators and mechanical parts - motors, pistons, grippers, wheels, gears

3) Sensors - vision, sound, temperature, motion, light, touch, etc.

With these three components, robots can interact and affect their environment to become useful.

What Do Robots Do?

Most robots today are used in factories to build products such as cars and electronics. Others are used to explore underwater and even on other planets.

3.3 Laws of Robotics

i. A robot must not injure a human being or, through inaction, allow a human being to Come to harm.
ii. A robot must always obey orders given to it by a human being, except where it Would conflict with the first law.

iii. A robot must protect its own existence, except where it would conflict with the first or second law. Later, this "Zeroth Law" is added.
iv. A robot must not injure a humanity or, through inaction, allow a humanity to come to harm.

3.4 Robot Sensing

The use of external sensing mechanisms allows a robot to interact with its environment in a flexible Manner. This is in contrast to pre-programmed operations in which a robot is “taught” to perform repetitive tasks via a set of pre-programmed functions.

The use of sensing technology to endow machines with a greater degree of intelligence in dealing with their environment is indeed an active topic of research and development in the robotics field.

The function of robot sensors may be divided into two principal categories:

1. Internal state.
2. External State.

Internal state sensors deal with the detection of variables such as arm joint position, which are used for robot control.

External state sensors, on the other hand, deal with the detection of variables such as range, proximity and touch.

External sensing is used for robot guidance as well as for object identification and handling. Although proximity, touch, vision is recognized as the most powerful of robot sensory capabilities, Robot vision may be defined as the process of extraction, characterizing, and interpreting information from images of a three-dimensional world.

The process, also commonly referred to as machine or computer vision, may be subdivided into six principal areas:

1. Sensing.
2. Pre-processing.
3. Segmentation.
4. Description.
5. Recognition.

6. Interpretation.

It is convenient to group these various areas of vision according to the sophistication involved in their implementation. We consider three levels of processing: low, medium and high level vision.

Here, we shall treat sensing and pre-processing as low-level vision functions. This will take us from the image formation process itself to compensations such as noise reduction, and finally to the extraction of primitive image features such as intensity discontinuities.

By 2050 robot "brains" based on computers that execute 100 trillion instructions per second will start rivalling human intelligence

In terms of our six subdivisions, we will treat segmentation, description, and recognition of individual objects as medium-level vision functions. High-level visions refer to processes that attempt to emulate cognition.

Artificial Intelligence

- **Sensing**

Learn about human and robotic vision systems, explore three modes of robotic sensing, and apply a robotic sensory mode to a specific task.

- **Thinking**

Human thinking (heuristic) and robotic thinking (algorithmic) will be explored as you gain an understanding of why a robot needs specific.

If we accept that computers will eventually be come powerful enough to simulate the mind, the question that naturally arises is: What processing rate will be necessary to yield performance on a parallel with the human brain?

By comparing how fast the neural circuits in the retina perform image-processing operations with how many instructions per second it takes a computer to accomplish similar work, believe it is possible to at least coarsely estimate the information-processing power of nervous tissue and by extrapolation, that of the entire human nervous system.

Motion detection in retina, if performed by efficient software, requires the execution of at least 100 computer instructions. Thus, to accomplish the retina's 10 million detections per second would require at least 1,000 MIPS.

The entire human brain is about 75,000 times heavier than the 0.02 gram of processing circuitry in the retina, which implies that it would take, in round numbers, 100 million MIPS (100 trillion instructions per second) to emulate the 1,500-gram human brain. Personal computers in 1999 beat certain insects but lose to the human retina and even to the 0.1-gram brain of a goldfish. A typical PC would have to be at least a million times more powerful to perform like a human brain.

3.5 Generations of Robots

3.5.1 A first-generation universal robots

Will handle only contingencies explicitly covered in their application programs. Unable to adapt to changing circumstances, they will often perform inefficiently or not at all. still, so much physical work awaits them in businesses, streets, fields and homes that robotics could begin to overtake pure information technology commercially.

3.5.2 A second generation of universal robot

With a mouse like 100,000 MIPS will adapt as the first generation does not and will even be trainable. Besides application programs, such robots would host a suite of software "conditioning modules" that would generate positive and negative reinforcement signals in predefined circumstances.

For example, doing jobs fast and keeping its batteries charged will be positive; hitting or breaking something will be negative. There will be other ways to accomplish each stage of an application program, from the minutely specific (grasp the handle underhand or overhand) to the broadly general (work indoors or outdoors). As jobs are repeated, alternatives that result in positive reinforcement will be favored, those with negative outcomes shunned. Slowly but surely, second-generation robots will work increasingly well.

3.5.3 A third generation of robots

To learn very quickly from mental rehearsals in simulations that model physical, cultural and psychological factors. Physical properties include shape, weight, strength, texture and appearance of things, and how to handle them. Cultural aspects include a thing's name, value, proper location and purpose.

Psychological factors applied to humans and robots alike include goals, beliefs, feelings and preferences. Developing the simulators will be a huge undertaking involving thousands of programmers and experience gathering robots.

The simulation would track external events and tune its models to keep them faithful to reality. It would let a robot learn a skill by imitation and afford a kind of consciousness. Asked why there are candles on the table, a third-generation robot might consult its simulation of house, owner and self to reply that it put them there because its owner likes candlelit dinners and it likes to please its owner. Further queries would elicit more details about a simple inner mental life concerned only with concrete situations and people in its work area.

3.5.4 Fourth-generation universal robots

With a humanlike 100 million MIPS will be able to abstract and generalize. They will result from melding powerful reasoning programs to third-generation machines. These reasoning programs will be the far more sophisticated descendants of today's theorem proves and expert systems, which mimic human reasoning to make medical diagnoses, schedule routes, make financial decisions, configure computer systems, and analyse seismic data to locate oil deposits and so on. Properly educated, the resulting robots will become quite formidable. In fact, they will outperform us in any conceivable area of endeavour, intellectual or physical Inevitably, such a development will lead to a fundamental restructuring of our society. Entire corporations will exist without any human employees or investors at all. Humans will play a pivotal role in formulating the intricate complex of laws that will govern corporate behaviour. Ultimately, though, it is likely that our descendants will cease to work in the sense that we do now. They will probably occupy their days with a variety of social, recreational, and artistic pursuits, not unlike today's comfortable retirees or the wealthy leisure classes.

3.6 Robot Systems

Robots are comprised of several systems working together as a whole.

The type of job the robot does dictates what system elements it needs. The general categories of robot systems are:

1. Controller
2. Body
3. Mobility
4. Power
5. Sensors
6. Tools

3.6.1 Controller

The controller is the robot's brain and controls the robot's movements. It's usually a computer of some type which is used to store information about the robot and the work environment and to store and execute programs which operate the robot. The control system contains programs, data algorithms, logic analysis and various other processing activities which enable the robot to perform.

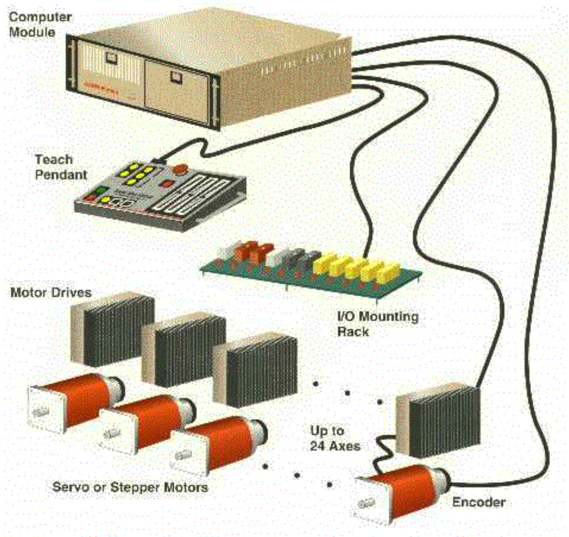

Figure 3.1: Controller

The picture above is an AARM Motion control system. AARM stands for Advanced Architecture Robot and Machine Motion and it's a commercial product from

American Robot for industrial machine motion control. Industrial controllers are either non-servos, point-to-point servos or continuous path servos.

A non-servo robot usually moves parts from one area to another and is called a "pick and place" robot. The non-servo robot motion is started by the controller and stopped by a mechanical stop switch. The stop switch sends a signal back to the controller which starts the next motion.

A point-to-point servo moves to exact points so only the stops in the path are programmed.A continuous path servo is appropriate when a robot must proceed on a specified path in a smooth, constant motion.

Mobile robots can operate by remote control or autonomously. A remote control robot receives instructions from a human operator. In a direct remote control situation, the robot relays information to the operator about the remote environment and the operator then sends the robot instructions based on theinformation received. This sequence can occur immediately (real-time) or with a time delay.

Autonomous robots are programmed to understand their environment and take independent action based on the knowledge they possess. Some autonomous robots are able to "learn" from their past encounters. This means they can identify a situation, process actions which have produced successful/unsuccessful results and modify their behaviour to optimize success. This activity takes place in the robot controller.

3.6.2 Body

The body of a robot is related to the job it must perform.Industrial robots often take the shape of a body less arm since its job requires it to remain stationary relative to its task. Space robots have many different body shapes such as a sphere, a platform with wheels or legs, or a balloon, depending on it's job.

The free-flying rover, Sprint Area is a sphere to minimize damage if it were to bump into the shuttle or an astronaut. Some planetary rovers have solar platforms driven by wheels to traverse terrestrial environments. Aerobat bodies are balloons that will float through the atmosphere of other worlds collecting data. When evaluating what body type is right for a robot, remember that form follows function.

Figure 3.2: Body

3.6.3 Mobility

How do robots move? It all depends on the job they have to do and the environment they operate in.

In the Water: Conventional unmanned, submersible robots are used in science and industry throughout the oceans of the world. You probably saw the Jason AUV at work when pictures of the Titanic discovery were broadcast.

To get around, automated underwater vehicles (AUV's) use propellers and rudders to control their direction of travel.

One area of research suggests that an underwater robot like RoboTuna could propel itself as a fish does using its natural adulatory motion. It's thought that robot that move like fish would be quieter, more manoeuvrable and more energy efficient.

On Land: Land based rovers can move around on legs, tracks or wheels. Dante II is a frame walking robot that is able to descend into volcano craters by repelling down the crater. Dante has eight legs; four legs on each of two frames. The frames are separated by a track along which the frames slide relative to each other. In most cases Dante II has at least one frame (four legs) touching the ground.An example of a track driven robot is Pioneer, a robot developed to clear rubble, make maps and acquire samples at the Chernobyl Nuclear Reactor site. Pioneer is track-driven like a small bulldozer which makes it suitable for driving over and through rubble. The wide track footprint gives good stability and platform capacity to deploy payloads. Many robots use wheels for locomotion.

In the Air/Space:

Robots that operate in the air use engines and thrusters to get around. One example is the Cassini, an orbiter on its way to Saturn. Movement and positioning is accomplished "reaction wheels." The thrusters and reaction wheels orient the spacecraft in three axes which are maintained with great precision.

Transformation 'matrix':

Robot arm kinematics deals with the analytical study of the geometry of motion of a robot arm with respect to the fixed reference coordinate system without regard to the forces/moments that cause the motion.

Thus, kinematics deals with the analytical description of the spatial displacement of the robot as a function of time, in particular the relations between the joint variable space and the position and orientation of the end-effector of a robot arm. There are two fundamental problems in robot armkinematics. The first problem is usually referred to as the direct (or forward) kinematics problem, Second problem is the inverse kinematics (or arm solution) problem. Since the independent variables in a robot arm are the joint variables, and a task is usually stated in terms of the reference coordinate frame, the inverse kinematics problem is used more frequently. Denavit and Hartenberg [1955] proposed a systematic and generalized approach of utilizing matrix algebra to describe and represent the spatial geometry of the links of a robot arm with respect to a fixed reference frame. This method uses a 4*4 homogeneous transformation matrix to describe the spatial relationship between two adjacent right mechanical links and reduces the direct kinematics problem to finding an equivalent 4*4 homogeneous transformation matrix that relates the spatial displacement of the hand coordinate frame to the reference coordinate frame.

3.6.4 Power

Power for industrial robots can be electric, pneumatic or hydraulic. Electric motors are efficient, require little maintenance, and aren't very noisy. Pneumatic robots use compressed air and come in a wide variety of sizes.

A pneumatic robot requires another source of energy such as electricity, propane or gasoline to provide the compressed air.

Hydraulic robots use oil under pressure and generally perform heavy duty jobs. This power type is noisy, large and heavier than the other power sources. A hydraulic robot also needs another source of energy to move the fluids through its components. Pneumatic and hydraulic robots require maintenance of the tubes, fittings and hoses that connect the components and distribute the energy. Two important sources of electric power for mobile robots are solar cells and batteries.

3.6.5 Sensors

Sensors are the perceptual system of a robot and measure physical quantities like contact, distance, light, sound, strain, rotation, magnetism, smell, temperature, inclination, pressure, or altitude.

Sensors provide the raw information or signals that must be processed through the robot's computer brain to provide meaningful information.

Robots are equipped with sensors so they can have an understanding of their surrounding environment and make changes in their behaviour based on the information they have gathered.

Sensors can permit a robot to have an adequate field of view, a range of detection and the ability to detect objects while operating in real or near real time within its power and size limits. Additionally, a robot might have an acoustic sensor to detect sound, motion or location, infrared sensors to detect heat sources, contact sensors, tactile sensors to give a sense of touch, or optical/vision sensors. For most any environmental situation, a robot can be equipped with an appropriate sensor. A robot can also monitor itself with sensors.

3.6.6 Tools

As working machines, robots have defined job duties and carry all the tools they need to accomplish their tasks on board their bodies. Many robots carry their tools at the end of a manipulator.

The manipulator contains a series of segments, jointed or sliding relative to one another for the purpose of moving objects. The manipulator includes the arm, wrist and end-effector. An end-effector is a tool or gripping mechanism attached to the end of a robot arm to accomplish some task.

It often encompasses a motor or a driven mechanical device. An end-effector can be a sensor, a gripping device, a paint gun, a drill, an arc welding device, etc. Tools are unique to the task the robot must perform.

CHAPTER 4

ARDUINO UNO

4.1 Arduino UNO Introduction

In order to use this board efficiently and without getting the board burnt / damaged we should be cautious in selecting a power source for our microcontroller board as for any electro mechanical device to work , selecting a power source is a crucial step. Arduino Mega 2560 can have power supply of two voltages i.e. 3.3 V and 5 V.3 voltage supply can be generated with the help of on board regulator and the maximum current is 50 mA that can pass through it. 5 volt supply can also be generated and regulated on board. DC supply can be given via power jack or by using USB cable port. Vin is an input power that can be used within the external power in range. GND is a ground terminal which can be connected to the ground of connection from which power is going to be supplied. In short there are 3 ways to power our board. We can use either USB cable to power our board and then transfer our code to board or we can power it up by Vin of board or via power jack. These last two methods to provide power to the board can be used once you built and compile code into board via USB cable [1].

Figure 4.1: Arduino Uno board

4.2 Pins of Arduino UNO Board & Uses

i. 15 pins out of this are useful for PWM (pulse width modulation)

ii. In analog side 16 input pins

iii. Ground pins : 5

iv. One pin for 3.3 volts

v. One reset button

vi. USART pins : 4 (These are hardware serial ports which produces maximum speed to set up communication)

vii. ISP programming pins :

viii. Crystal oscillator is added on the board having frequency of 16 MHz

ix. USB cable port (It is used to transfer and connect code from computer the board)

x. ICSP header (Used to program the board and to upload code from computer. Indeed a remarkable addition in Arduino MEGA 2560)

xi. Power Jack

xii. Resettable Poly fuse : (to provide extra layer of protection. It prevents USB port of the computer from overheating in case of high current flowing through microcontroller board)

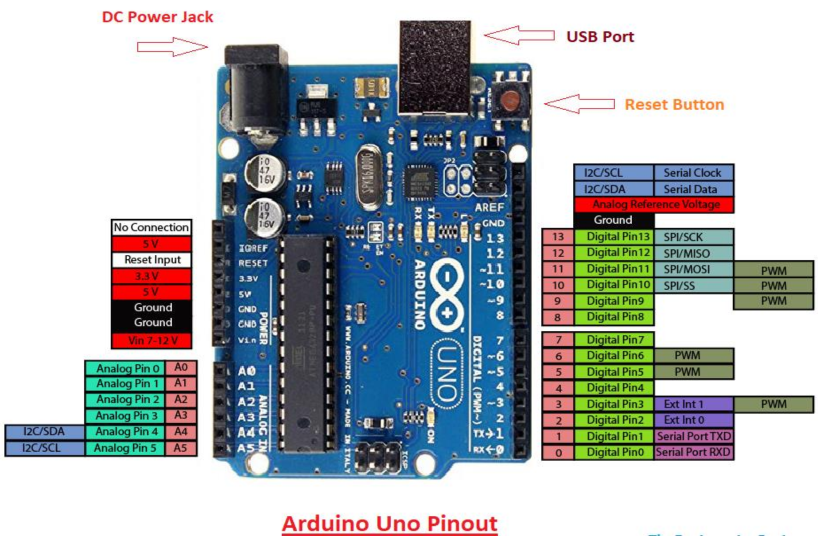

Figure 4.2: Arduino AT Mega 328 Pin diagram

4.3 Arduino UNO Pin Configurations

IOREF: This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then selects the proper power source.

PWM: PWM is provided by 3,5,6,9,10, 11pins. These pins are configured to provided 8-bit output PWM.

SPI: It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of the SPI library.

AREF: It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

TWI: It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.

Serial Communication: Serial communication is carried out through two pins called Pin 0 (Rx) and Pin 1 (Tx).

Rx: pin is used to receive data while **Tx** pin is used to transmit data.

External Interrupts: Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

4.4Applications of Arduino UNO

- 8 bit computer
- 3D printer
- To control and handle more than one motor
- Robot with many sensors
- CNC router
- Temperature detection and sensation
- Detection of water level

Table 4.1: Arduino Specifications

S.No.	Parameter	Description
1	Microcontroller	Atmega328
2	Crystal Oscillator	16MHz
3	Operating Voltage	5V
4	Input Voltage	5-12V
5	Digital I/O Pins	14 (D0 to D13)
6	Analog I/O Pins	6 (A0 to A5)
7	PWM Pins	6 (Pin # 3, 5, 6, 9, 10 and 11)
8	Power Pins	5V, 3.3V, Vin, GND
9	Communication	UART(1), SPI(1), I2C(1)
10	Flash Memory	32 KB (0.5KB is used by bootloader)
11	SRAM	2 KB
12	EEPROM	1 KB
13	ICSP Header	Yes
14	Power sources	DC Power Jack & USB Port

CHAPTER 5

HARDWARE IMPLEMENTATION

5.1 Block Diagram

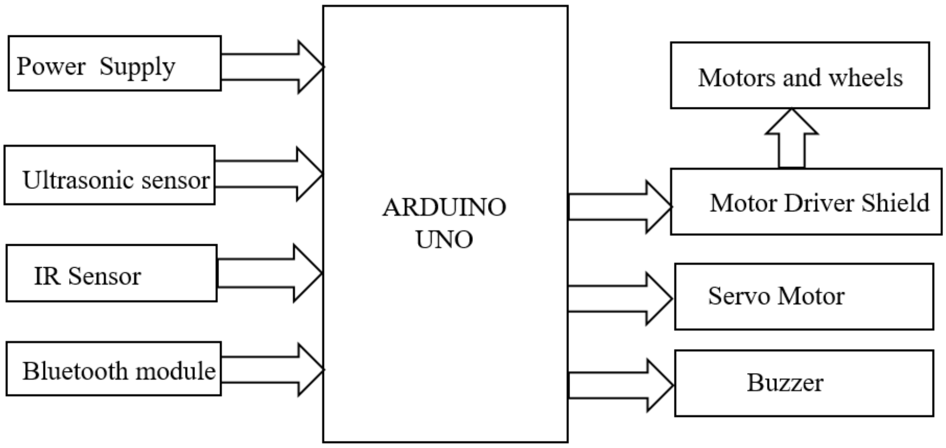

Figure 5.1: Block Diagram

5.2 Working Of Block Diagram

Arduino is the heart of this project where we can write programs and dump on it. The Power supply of 12v is connected to the Arduino Uno. The HC-SR04 of ultrasonic sensor is given into Arduino that is used to sense the object within the range of 150m. If the object is detected within the specific range it takes different direction. If not it moves forward until the obstacles are detected. This movement takes place until the power is ON. The IR sensor senses the obstacles in left and right directions . The movement takes place until it's range is covered.

Bluetooth module helps in the movement of the vehicle. This is done with the help of Android application that is worked on the basis MIT app inventor. This helps in communicating the vehicle. For switching the mode, the human follower an object is detected by either IR sensor or Ultrasonic sensor. The application of the vehicle takes places in communicating with the Android application. The servo motor helps in rotating in a particular angle that is specified. The motor driver shield is mainly used for the movement of the wheels that helps in rotation of the car or vehicle.

5.3 Arduino UNO Micro Controller

In order to use this board efficiently and without getting the board burnt / damaged we should be cautious in selecting a power source for our microcontroller board as for any electro mechanical device to work , selecting a power source is a crucial step.Arduino Mega 2560 can have power supply of two voltages i.e. 3.3 V and 5 V.3 voltage supply can be generated with the help of on board regulator and the maximum current is 50 mA that can pass through it.5 volt supply can also be generated and regulated on board. DC supply can be given via power jack or by using USB cable port.Vin is an input power that can be used within the external power in range.GND is a ground terminal which can be connected to the ground of connection from which power is going to be supplied.In short there are 3 ways to power our board. We can use either USB cable to power our board and then transfer our code to board or we can power it up by Vin of board or via power jack. These last two methods to provide power to the board can be used once you built and compile code into board via USB cable [1].

Figure 5.2: Arduino Uno Micro controller

5.4 HC05 Bluetooth Module

HC05 Bluetooth Module A Bluetooth module is a module designed for transparent wireless serial connection setup. HC-05 Bluetooth module is used in a robot controlled via Bluetooth to establish communication between mobile phone and the microcontroller. It operates at a voltage of a convenient 1.8 volts. HC-05 has a range of 10m. It has four pins: VCC (Power 3.3 – 6V), GND, TXD, RXD.

- It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
- It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.
- It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network (PAN). It uses frequency-hopping spread spectrum (FHSS) radio technology to send data over air.
- It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

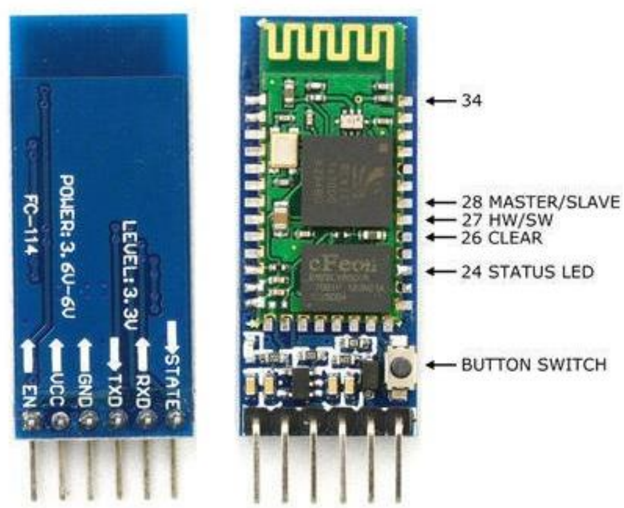

Figure 5.3: HC05 Bluetooth Module

5.5 L293D Motor Driver

L293D is a typical motor driver or motor driver IC which allows DC motor to drive on either direction. L293D shield is a driver board based on L293 IC, which can drive 4 DC motore and 2 Stepper or Servo motors at the same time. Each channel of this modulw has the maximum current of 1.2A and does not work if the above voltage is more than 25V or less than 4.5V [3].

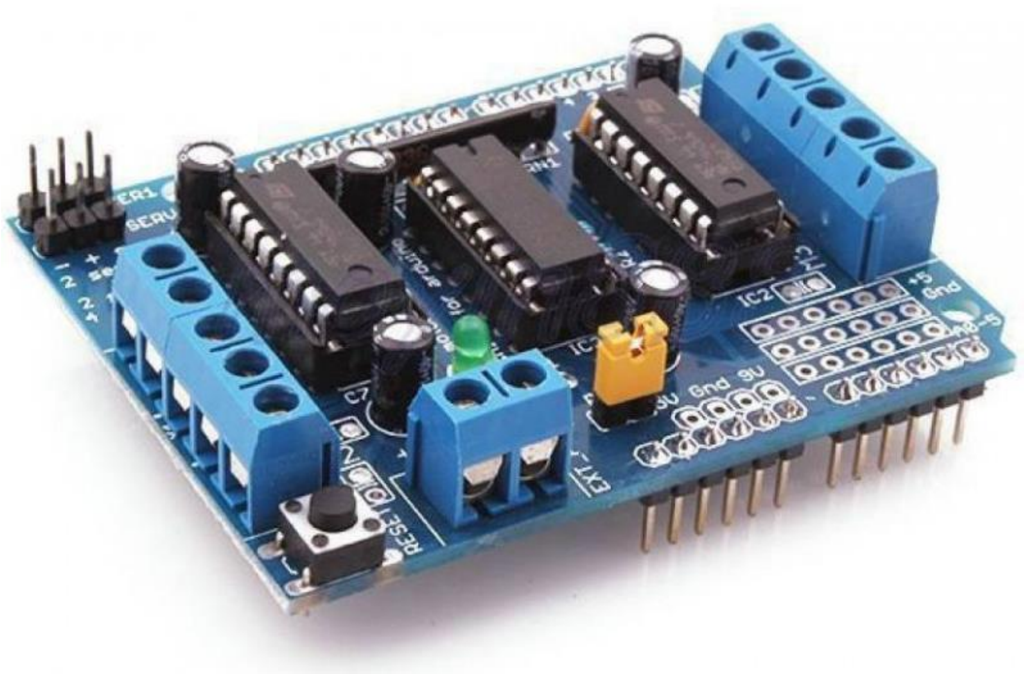

Figure 5.4: L293D Motor Driver Shield

5.6 TT Gear Motors

These durable (but affordable!) plastic gearbox motors (also known as 'TT' motors) are an easy, low-cost way to get your projects moving. This is a TT DC Gearbox Motor with a gear ratio of 1:48, and it comes with 2 x 200mm wires with breadboard-friendly 0.1" male connectors. Perfect for plugging into a breadboard or terminal blocks. You can power these motors with 3VDC up to 6VDC, they'll of course go a little faster at the higher voltages. We grabbed one motor and found these stats when running it from a bench-top supply

- At 3VDC we measured 150mA @ 120 RPM no-load, and 1.1 Amps when stalled
- At 4.5VDC we measured 155mA @ 185 RPM no-load, and 1.2 Amps when stalled
- At 6VDC we measured 160mA @ 250 RPM no-load, and 1.5 Amps when stalled

Figure 5.5: TT Gear Motors and Wheels

5.7 Bluetooth android application

Smartphones are more affordable and efficient hand held devices which can be used to support collaborative activities in a community. Bluetooth is mainly used for data exchange; add new features to smartphones viewed. This application is compatible for android devices. The app framework provides access to the Bluetooth functionality through Bluetooth APIs. These APIs let apps connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features. To test Bluetooth connection functionality, you can press the button on the android application.

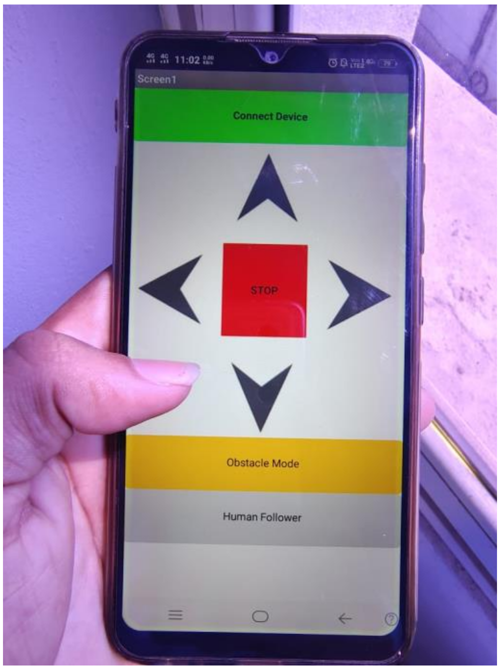

Figure 5.6: Bluetooth Control Application Interface

Table 5.1: Bluetooth Controls

MOVEMENT	LEFT MOTORS	RIGHT MOTORS
FORWARD	CLOCKWISE	CLOCKWISE
LEFT	ANTI CLOCKWISE	CLOCKWISE
RIGHT	CLOCKWISE	ANTI CLOCKWISE
BACKWARD	ANTI CLOCKWISE	ANTI CLOCKWISE

5.8 Infrared Sensors

IR sensors work by using specific light sensors to detect a particular light wavelength in the infrared spectrum. The main reason of selecting IR LEDs over other Opto-electronic sensors is due to the colour combination of the path which is black and white. Since, black path acts as a perfectly black body surface which absorbs the IR radiation emitted by IR transmitter so the IR receiver cannot receive the radiation from the transmitter but in case of white path, the opposite phenomenon takes place i.e. the IR receiver receives the radiation from the IR transmitter and hence it detects the path [2].

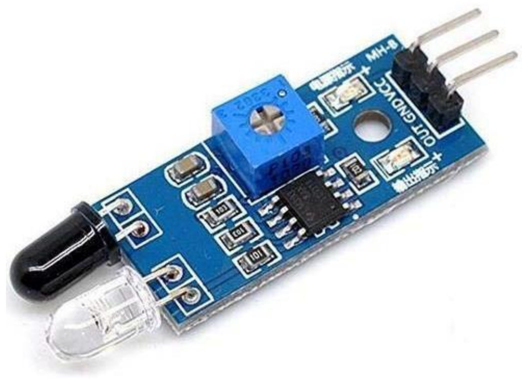

Figure 5.7: Infrared Sensor

Infrared Transmitter: It is a light emitting diode which emits infrared radiations. Hence they are called IR LED's. The radiation emitted is invisible to human eye. A simple infrared transmitter can be constructed using an infrared LED, a current limiting resistor and power supply.

Infrared Receiver: Photodiodes are usually used as IR receivers. Different types of IR receivers' exist based on the wavelength, voltage, etc. The wavelength of the receiver circuit should match that of the transmitter.

5.9 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target) [4].

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is $D = \frac{1}{2} T \times C$ (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025

seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be:

$$\mathbf{D = 0.5 \times 0.025 \times 343}$$

Figure 5.8: Ultrasonic sensor

5.10 Servo Motor

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the DC servo motor working. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.

Figure 5.9: Servo Motor

5.11 Power Supply

Power supply is a broad term but this lesson is restricted to discussion of circuits that generate a fixed or controllable magnitude dc voltage from the available form of input voltage. Integrated circuit (IC) chips used in dc supply for their proper operation. In majority of the cases the required voltages are of magnitudes varying between -18 to +18 volts. The 9V battery is an extremely common battery that was first used in transistor radios. It features a rectangular prism shape that utilizes a pair of snap connectors which are located at the top of the battery. A wide array of both large and small battery manufacturers produce versions of the 9V battery. Possible chemistries of primary (non-rechargeable) 9V batteries include Alkaline, Carbon-Zinc (Heavy Duty), Lithium. Possible chemistries of secondary (rechargeable) 9V batteries include nickel-cadmium (NiCd), nickel-metal hydride (NiMH), and lithium ion. The performance and application of the battery can vary greatly between different chemistries, meaning that some chemistiries are better suited for some applications over others.

Figure 5.10: 12V 1.3ah Rechargeable Battery

5.12 Buzzer

Buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, train and confirmation of user input such as a mouse click or keystroke.

Figure 5.11: Buzzer

5.13 LED

A Light-emitting diode (LED) is a semiconductor device that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor.White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

Figure 5.12: Light Emittting Diode

5.14 Switch

Switches are used to turn ON/OFF devices and to connect different parts of a circuit. The slide-switch in Arduino moves the slider of the switch from the open position (ON) to the closed position (OFF). It allows the flow of current in the circuit without the need for splice wire.

Figure 5.13: Switch

CHAPTER 6

SOFTWARE IMPLEMENTATION

6.1 Arduino IDE

Arduino IDE which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware .The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save The Arduino integrated development environment(IDE) is a cross-platform application (for Windows, Mac OS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring.

Figure 6.1: Arduino IDE logo

The Arduino IDE supplies a software library from the Wiring project, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful message Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an “Arduino language.” However, the Arduino is, in fact, programmed in C++.

The basic hardware that we are using in Arduino is Arduino UNO. The required Software that is installed in Arduino IDE is the version of Arduino IDE 1.6.8. Connect Arduino UNO to the system/pc which can be done using USB. Once the Arduino software was installed then select new file which an inbuilt two functions**.**

6.2 Pin Mapping of Atmega328P with Arduino

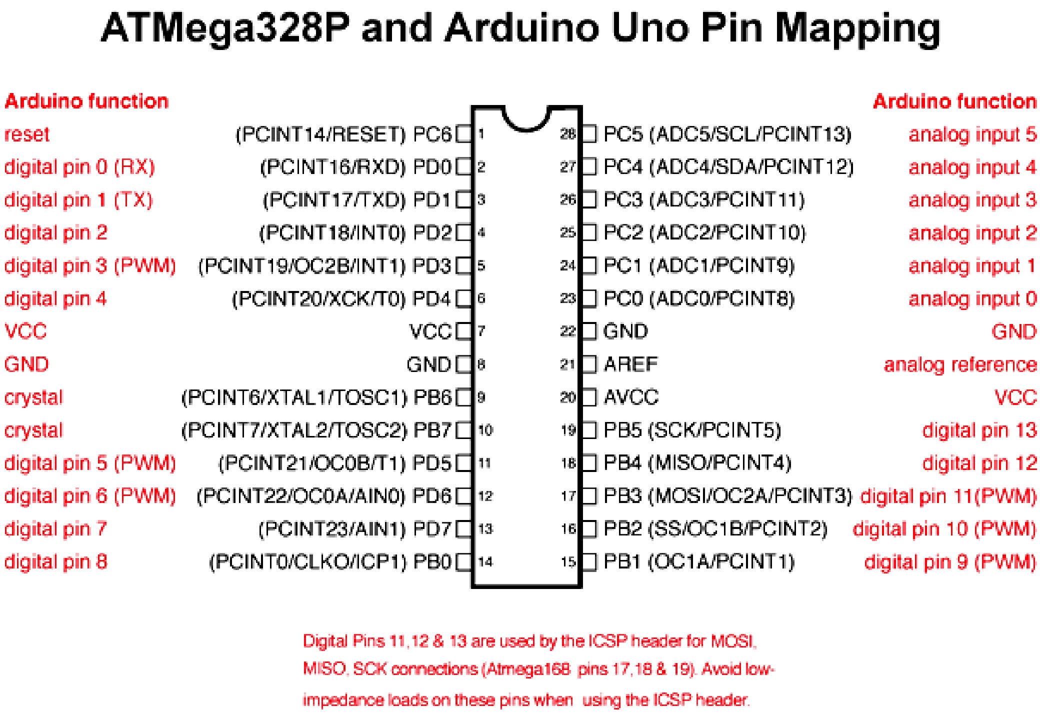

Figure 6.2: Atmega328P and Arduino UNO Pin Mapping

6.3 Features of Arduino Uno Board

The features of Arduino Uno ATmega328 includes the following.

• The operating voltage is 5V

• The recommended input voltage will range from 7v to 12V

• The input voltage ranges from 6v to 20V

• Digital input/output pins are 14

• Analog i/p pins are 6

• DC Current for each input/output pin is 40 mA

• DC Current for 3.3V Pin is 50 mA

• Flash Memory is 32 KB

• SRAM is 2 KB

• EEPROM is 1 KB

• CLK Speed is 16 MHz

6.4 Applications

The applications of Arduino Uno include the following.

- Arduino Uno is used in Do-it-Yourself projects prototyping.
- In developing projects based on code-based control.
- Development of Automation System.
- Designing of basic circuit designs.
- Automated gardening systems.
- Measuring and monitoring systems.
- Educational purposes for teaching electronics and programming.

6.5 Android Application Development

6.5.1 Introduction

The smartphone is an information nexus in today's digital age, with access to a nearly infinite supply of content on the web, coupled with rich sensors and personal data. However, people have difficulty harnessing the full power of these ubiquitous devices for themselves and their communities. Most smartphone users consume technology without being able to produce it, even though local problems can often be solved with mobile devices. MIT App Inventor is designed to democratize this technology and is used as a tool for learning computational thinking in a variety of educational contexts, teaching people to build apps to solve problems in their communities.

MIT App Inventor is an online development platform that anyone can leverage to solve real-world problems. It provides a web-based "What you see is what you get" (WYSIWYG) editor for building mobile phone applications targeting the Android and iOS operating systems. It uses a block-based programming language built on Google Blockly (Fraser,2013) and inspired by languages such as StarLogo TNG (Begel & Klopfer,2007) and Scratch ,Maloney, Resnick, Rusk, Silverman, & Eastmond,2010) empowering anyone to build a mobile phone app to meet a need. To date, 6.8 million people in over 190 countries have used App Inventor to build over 24 million apps. We offer the interface in more than a dozen languages. People around the world use App Inventor to provide mobile solutions to real problems in their families, communities, and the world. The platform has also been adapted to serve requirements of more specific populations, such as building apps for emergency/first responders and robotics.

Discussing the pedagogical value of MIT App Inventor and its use as a tool to teach and encourage people of all ages to think and act computationally. We also describe three applications developed by students in different parts of the world to solve real issues in their communities. We conclude by discussing the limitations and benefits of tools such as App Inventor and proposing new directions for research.

Figure 6.3: MIT App Inventor Logo

6.5.2 MIT APP INVENTOR OVRVIEW

The MIT App Inventor user interface includes two main editors: the design editor and the blocks editor. The design editor, or designer is a drag and drop interface to lay out the elements of the application's user interface (UI). The blocks editor is an environment in which app inventors can visually lay out the logic of their apps using color-coded blocks that snap together like puzzle pieces to describe the program. To aid in development and testing, App Inventor provides a mobile app called the App Inventor Companion (or just "the Companion") that developers can use to test and adjust the behavior of their apps in real time. In this way, anyone can quickly build a mobile app and immediately begin to iterate and test. App Inventor's design editor. App inventors drag components out from the palette (far left) to the viewer (center left) to add them to the app. Inventors can change the properties of the components (far right).

6.5.3 MIT APP INVENTOR DESIGN GOALS

In the design of MIT App Inventor, introducing mobile app development in educational contexts was a central goal. Prior to its release, most development environments for mobile applications were clunky, only accessible with expertise in systems level or embedded programming, or both. Even with Google's Android operating system and the Java programming language, designing the user interface was a complex task. Further, use of the platform required familiarity with Java syntax and semantics, and the ability to debug Java compilation errors (e.g., misspelled variables or misplaced semicolons) for success. These challenges presented barriers to entry for individuals not versed in computer science, App Inventor's target demographic. We briefly highlight and discuss design goals for the App Inventor project, specifically, the use of components to abstract some of the complexity of platform behavior, and the use of blocks to eliminate complexity of the underlying programming language. These goals can be further explained as aligning the visual language to the mental models of young developers and enabling exploration through fast, iterative design.

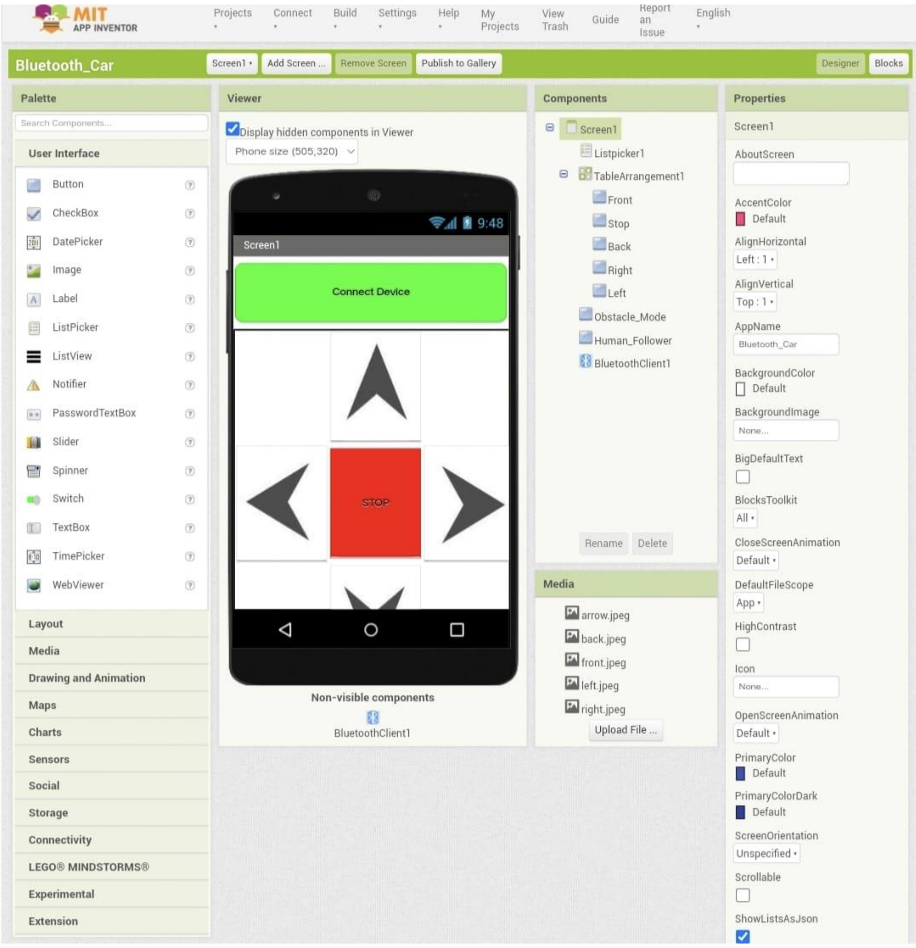

Figure 6.4: MIT App Inventor User Interface

6.5.4 MIT APP INVENTOR IN EDUCATION

The primary aim of MIT App Inventor is providing anyone with an interest in building apps to solve problems with the tools necessary to do so. Instructional materials developed by the team are primarily oriented toward teachers and students at the middle- and high-school levels, but app inventors come in all ages from around the world. In this section, we describe a few of the key components of the MIT App Inventor educational strategy, including massively online open courses (MOOCs) focused on MIT App Inventor, the Master Trainer (MT) program, the extensions functionality of App Inventor that allows incorporation of new material for education, and research projects that have leveraged App Inventor as a platform for enabling domain-specific computing.

6.5.5 BENEFITS OF VISUAL PROGRAMMING USING MOBILE

Users of the App Inventor platform benefit from being able to repurpose the computational thinking skills they learn to interface with physical space in the external world. The visual programming Application Inventor and the abstraction and the compartmentalization of concepts into components and blocks allow the app inventor to focus more on decomposing their problems into solvable elements. The facility of running apps on mobile devices allows the students to experience their own apps as part of an ecosystem they interact with daily, and with which they are intimately familiar. Since this encapsulation reduces the time it takes to build an app, even a straightforward prototype, app inventors can quickly grasp and iterate without paying a significant cost in terms of a compile-load-run cycle that is typical with mobile app development.

6.5.6 CONCLUSION

The MIT App Inventor project continues to push the boundaries of education within the context of mobile app development. Its abstraction of hardware capabilities and the reduction of complex logic into compact representations allow users to quickly and iteratively develop projects that address real-world problems. We discussed how App Inventor's curriculum development incorporates elements of computational thinking and encourages computational action with real-world effects. We also presented a number of projects that effectively accomplish this mission. We continue to grow the platform to democratize access to newer technologies, preparing future generations for a world in which computational thinking is a central part of problem-solving.

Chapter 7

Result and Conclusion

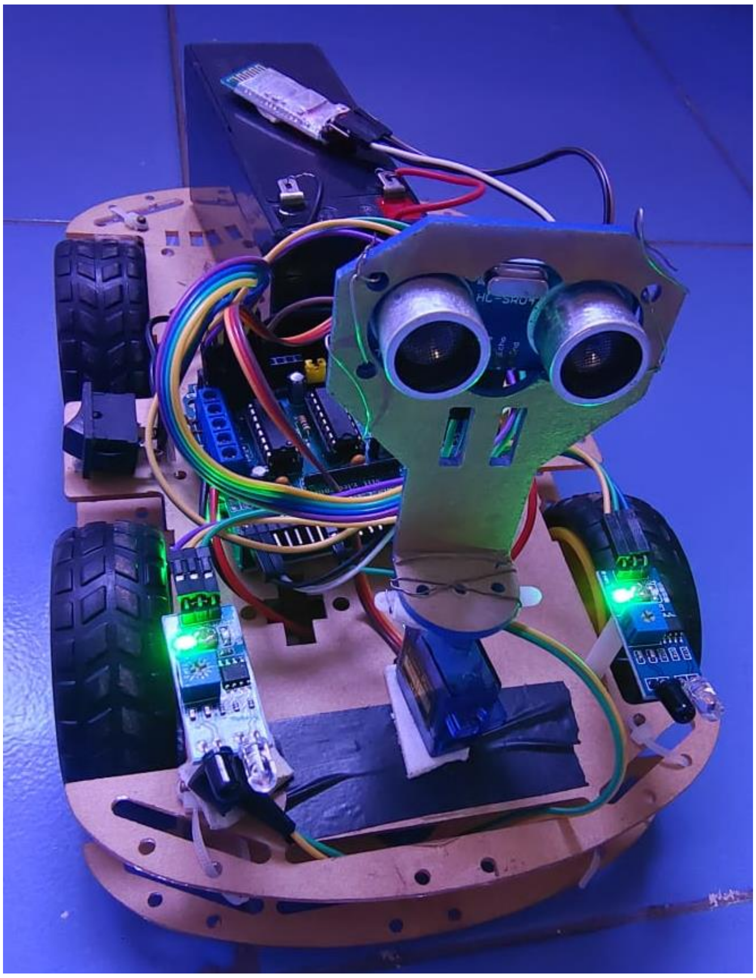

Figure 7.1: Robot Car Output

Working of the Project

In this project, we have developed a specific app named as Bluetooth Car. The app contains three functions. They are Bluetooth Controller, Obstacle avoider and Human/Object Follower. By using these commands we control the Car.

The Working of this car is explained as

1st Function: The main purpose of this function is controlled using Bluetooth. That means the car is commanded using specific functions like Front, Back, Left, Right and Stop. Here the car is controlled with the access of Bluetooth Controller.

2nd Function: The main purpose of this function is obstacle avoider. To avoid obstacles we opt this function. It detects the Obstacle around 35cm, if the obstacle is not being detected it goes forward i.e, in the direction of front.

In the case of obstacle detection it checks for the left and right with the help of ultrasonic sensor interfaced through Servo Motor. It moves in the direction where the obstacle is not detected. The detection of obstacle is identified through buzzer (2 seconds).

3rd Function: The purpose of this function is to follow the movement of a Human or an object. The car detects the range of Human or Object within greater than 10cm and less than 30cm and follows the movement. It consistes of two IR sensors i.e, in the direction of left and right. The purpose of this left IR and right IR Sensors is to sense whether the person is present or not. If the person is detected in a paticular direction the car turns to that particular direction.

Applications

- Industrial automation
- They can be used for household work like automatic vacuum cleaning.
- Low range Mobile Surveillance Devices
- Assistive devices (like wheelchairs)
- Home automation
- The robot is small in size so can be used for spying.
- With few additions and modifications, this robot can be used in the borders for detecting and disposing hidden land mines.
- It can be used in various industries where human intervention is not desired.
- It can be used to develop robot with military applications.
- It provides more application based on Android operating system.
- With tremendous smart phone in markets, it is bound to have many more applications in near future.
- Obstacle avoiding robots can be used in almost all mobile robot navigation systems.
- Robots can perform limitless including assisting in carrying loads for people working in hospitals, libraries, airports etc

Appendix

Code

```
#include<NewPing.h>
#include<Servo.h>
#include<AFMotor.h>
#define RIGHT A3
#define LEFT A2
#define TRIGGER_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 100
#define MAX_SPEED 100
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
AF_DCMotor Motor1(1,MOTOR12_1KHZ);
AF_DCMotor Motor2(2,MOTOR12_1KHZ);
AF_DCMotor Motor3(3,MOTOR34_1KHZ);
AF_DCMotor Motor4(4,MOTOR34_1KHZ);
Servo myservo;
boolean goesForward=false;
int pos =0;
int distance2=100;
int speedSet=0;
char command;
int buzzer=A4;

void setup()
{
  pinMode(buzzer,OUTPUT);
  Serial.begin(9600);  //Set the baud rate to your Bluetooth module.
  myservo.attach(10);
  myservo.attach(10);
```

```
  myservo.write(100);
  delay(2000);
  distance2 = readPing();
  delay(100);
  distance2 = readPing();
  delay(100);
  distance2 = readPing();
  delay(100);
  distance2 = readPing();
  delay(100);
{
for(pos = 90; pos <= 180; pos += 1){
  myservo.write(pos);
  delay(15);
} for(pos = 180; pos >= 0; pos-= 1) {
  myservo.write(pos);
  delay(15);
}for(pos = 0; pos<=90; pos += 1) {
  myservo.write(pos);
  delay(15);
}
}
pinMode(RIGHT, INPUT);
pinMode(LEFT, INPUT);

}

void loop(){
  if(Serial.available() > 0){
    command = Serial.read();
    Stop(); //initialize with motors stoped
    //Change pin mode only if new command is different from previous.
```

```
    //Serial.println(command);
    switch(command){
    case 'F':
      forward();
      break;
    case 'B':
       back();
      break;
    case 'L':
      left();
      break;
    case 'R':
      right();
      break;
    case 'S':
      Stop();
      break;
    case 'H':
     human();
      break;
    case 'O':
      obstacle();
      break;
    }
  }
}

void forward()
{
  Motor1.setSpeed(100); //Define maximum velocity
  Motor1.run(FORWARD); //rotate the motor clockwise
  Motor2.setSpeed(100); //Define maximum velocity
```

```
  Motor2.run(FORWARD); //rotate the motor clockwise
  Motor3.setSpeed(100);//Define maximum velocity
  Motor3.run(FORWARD); //rotate the motor clockwise
  Motor4.setSpeed(100);//Define maximum velocity
  Motor4.run(FORWARD); //rotate the motor clockwise
  Serial.println("BT Forward");
}

void back()
{
  Motor1.setSpeed(100); //Define maximum velocity
  Motor1.run(BACKWARD); //rotate the motor anti-clockwise
  Motor2.setSpeed(100); //Define maximum velocity
  Motor2.run(BACKWARD); //rotate the motor anti-clockwise
  Motor3.setSpeed(100); //Define maximum velocity
  Motor3.run(BACKWARD); //rotate the motor anti-clockwise
  Motor4.setSpeed(100); //Define maximum velocity
  Motor4.run(BACKWARD); //rotate the motor anti-clockwise
  Serial.println("BT Backward");
}

void left()
{
  Motor1.setSpeed(100); //Define maximum velocity
  Motor1.run(BACKWARD); //rotate the motor anti-clockwise
  Motor2.setSpeed(100); //Define maximum velocity
  Motor2.run(BACKWARD); //rotate the motor anti-clockwise
  Motor3.setSpeed(100); //Define maximum velocity
  Motor3.run(FORWARD);  //rotate the motor clockwise
  Motor4.setSpeed(100); //Define maximum velocity
  Motor4.run(FORWARD);  //rotate the motor clockwise
  Serial.println("BT Left");
```

```
}

void right()
{
  Motor1.setSpeed(100); //Define maximum velocity
  Motor1.run(FORWARD); //rotate the motor clockwise
  Motor2.setSpeed(100); //Define maximum velocity
  Motor2.run(FORWARD); //rotate the motor clockwise
  Motor3.setSpeed(100); //Define maximum velocity
  Motor3.run(BACKWARD); //rotate the motor anti-clockwise
  Motor4.setSpeed(100); //Define maximum velocity
  Motor4.run(BACKWARD); //rotate the motor anti-clockwise
  Serial.println("BT Right");
}

void Stop()
{
  Motor1.setSpeed(0); //Define minimum velocity
  Motor1.run(RELEASE); //stop the motor when release the button
  Motor2.setSpeed(0); //Define minimum velocity
  Motor2.run(RELEASE); //rotate the motor clockwise
  Motor3.setSpeed(0); //Define minimum velocity
  Motor3.run(RELEASE); //stop the motor when release the button
  Motor4.setSpeed(0); //Define minimum velocity
  Motor4.run(RELEASE); //stop the motor when release the button
  Serial.println("Stopped");
}
void human(){
  while(1){
  delay(5);
 unsigned int distance = sonar.ping_cm();
Serial.println("Human Follower Mode");
```

```
Serial.print("distance");
Serial.println(distance);

int Right_Value = digitalRead(RIGHT);
int Left_Value = digitalRead(LEFT);

Serial.print("RIGHT");
Serial.println(Right_Value);
Serial.print("LEFT");
Serial.println(Left_Value);
Serial.print("before if");
if((Right_Value==1) && distance>=10 && distance<=30 &&(Left_Value==1)){
  Serial.print("in if block");
  Motor1.setSpeed(100);
  Motor1.run(FORWARD);
  Motor2.setSpeed(100);
  Motor2.run(FORWARD);
  Motor3.setSpeed(100);
  Motor3.run(FORWARD);
  Motor4.setSpeed(100);
  Motor4.run(FORWARD);
}
else if((Right_Value==0) && (Left_Value==1)) {
  Serial.print("right block");
  Motor1.setSpeed(100);
  Motor1.run(FORWARD);
  Motor2.setSpeed(100);
  Motor2.run(FORWARD);
  Motor3.setSpeed(100);
  Motor3.run(BACKWARD);
  Motor4.setSpeed(100);
  Motor4.run(BACKWARD);
```

```
}
else if((Right_Value==1)&&(Left_Value==0)) {
  Serial.print("left block");
  Motor1.setSpeed(100);
  Motor1.run(BACKWARD);
  Motor2.setSpeed(100);
  Motor2.run(BACKWARD);
  Motor3.setSpeed(100);
  Motor3.run(FORWARD);
  Motor4.setSpeed(100);
  Motor4.run(FORWARD);
}
else if((Right_Value==1)&&(Left_Value==1)) {
  Serial.print("stop block");
  Motor1.setSpeed(0);
  Motor1.run(RELEASE);
  Motor2.setSpeed(0);
  Motor2.run(RELEASE);
  Motor3.setSpeed(0);
  Motor3.run(RELEASE);
  Motor4.setSpeed(0);
  Motor4.run(RELEASE);
}
else if(distance > 1 && distance < 10) {
  Serial.print("less distance block");
  Motor1.setSpeed(0);
  Motor1.run(RELEASE);
  Motor2.setSpeed(0);
  Motor2.run(RELEASE);
  Motor3.setSpeed(0);
  Motor3.run(RELEASE);
  Motor4.setSpeed(0);
```

```
 Motor4.run(RELEASE);
 }

 command = Serial.read();
 if(command=='S'){
   break;
 }
 }
}
void obstacle(){
  int distanceR = 0;
int distanceL =  0;
delay(40);
Serial.println("Obstacle Mode");
while(1){
if(distance2<=35)
{
 moveStop();
 delay(100);
 {
 digitalWrite(buzzer,HIGH);
 delay(500);
 digitalWrite(buzzer,LOW);
 }

 moveBackward();
 delay(300);

 moveStop();
 delay(200);
 distanceR = lookRight();
 delay(200);
```

```
 distanceL = lookLeft();
 delay(200);

 if(distanceR>=distanceL)
 {
   turnRight();
   moveStop();
 }else
 {
   turnLeft();
   moveStop();
 }
}else
{
 moveForward();
}
distance2 = readPing();
command = Serial.read();
 if(command=='S'){
   break;
 }
}
}

int lookRight()
{
    myservo.write(30);
    delay(1000);
    int distance = readPing();
    delay(100);
    myservo.write(100);
    return distance;
```

```
}

int lookLeft()
{
    myservo.write(180);
    delay(1000);
    int distance = readPing();
    delay(100);
    myservo.write(100);
    return distance;
    delay(100);
}

int readPing() {
  delay(70);
  int cm = sonar.ping_cm();
  if(cm==0)
  {
    cm = 250;
  }
  return cm;
}

void moveStop() {
  Motor1.run(RELEASE);
  Motor2.run(RELEASE);
  Motor3.run(RELEASE);
  Motor4.run(RELEASE);

  }

void moveForward() {
```

```
 if(!goesForward)
  {
    goesForward=true;
    Motor1.run(FORWARD);
    Motor2.run(FORWARD);
    Motor3.run(FORWARD);
    Motor4.run(FORWARD);
   for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the
speed up to avoid loading down the batteries too quickly
   {
    Motor1.setSpeed(speedSet);
    Motor2.setSpeed(speedSet);
    Motor3.setSpeed(speedSet);
    Motor4.setSpeed(speedSet);
    delay(5);
   }
  }
}


void moveBackward() {
    goesForward=false;
    Motor1.run(BACKWARD);
    Motor2.run(BACKWARD);
    Motor3.run(BACKWARD);
    Motor4.run(BACKWARD);
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the
speed up to avoid loading down the batteries too quickly
  {
    Motor1.setSpeed(speedSet);
    Motor2.setSpeed(speedSet);
```

```
    Motor3.setSpeed(speedSet);
    Motor4.setSpeed(speedSet);
    delay(5);
  }
}


void turnRight() {
  Motor1.run(FORWARD);
  Motor2.run(FORWARD);
  Motor3.run(BACKWARD);
  Motor4.run(BACKWARD);
  delay(500);
  Motor1.run(FORWARD);
  Motor2.run(FORWARD);
  Motor3.run(FORWARD);
  Motor4.run(FORWARD);
}

void turnLeft() {
  Motor1.run(BACKWARD);
  Motor2.run(BACKWARD);
  Motor3.run(FORWARD);
  Motor4.run(FORWARD);
  delay(500);
  Motor1.run(FORWARD);
  Motor2.run(FORWARD);
  Motor3.run(FORWARD);
  Motor4.run(FORWARD);
}
```

References

[1] Arduino Bluetooth Robot Car : 18 Steps (with Pictures) - Instructables

[2] Available: https://wiki.eprolabs.com/index.php?title=IR_Obstacle_ Sensor

[3] Available: https://www.rakeshmondal.info/L293DMotor-Driver

[4] Available:https://maker.pro/arduino/projects/diy-arduino-human-following-robot

[5] International Journal of Electronic and Electrical Engineering. ISSN 0974-2174, Volume 7, Number 5 (2014) “Android Phone Controlled Robot Using Bluetooth” by Arpit Sharma, Reetesh Verma, Saurabh Gupta and Sukhdeep Kaur Bhatia, Dept. of Instrumentation & Control Engg, J.S.S. Academy of Technical Education Noida,India.

[6] Proceedings of the Intl. Conf. on Innovative trends in Electronics Communication and Applications ICIECA2013, December 2013 “Secured Robotic Line Follower Vehicle” by Er. Faruk Bin Poyen, Er. Santu Guin, Er. Dibyendu Banerjee, Deep Mukherjee, Department of AElE, UIT The University Of Burdwan, Burdwan, India.