Palestine Polytechnic University

College of Information Technology and

Computer Engineering.

Self-Parking Car.

Team:

Mohammad Abuarqoub.

&

Islam Shaheen.

Supervisor:

Dr. Alaa Halawani.

January 2021.

Acknowledgement

First and foremost, we would like to thank God for giving us the ability to do this work. Without his blessings, this achievement would not have been possible.

It has been a great opportunity to gain lots of experience in real time projects, followed by the knowledge of how to actually design and analyze real projects. For that we want to thank all the people who made it possible for students like us. We would like to express our deepest gratitude to our graduation project supervisor for his patience and guidance. Moreover, it is our duty to thank all the committee members for their generous discussions and encouragement. At last, we would like to thank all the people who helped, supported and encouraged us to successfully finish the graduation project.

Special thanks to our families and our friends for their unconditional love and support for us.

Abstract

Parking is usually a driver’s worst nightmare because, it not only requires the driver’s skills but also increases the possibility of other drivers bumping into their parked vehicle. Autonomous car parking was introduced to encounter the parking problems. Self-parking Car technology is a car that guides itself without human conduction.

In this project, we developed the car's brain by Arduino microcontroller, so that it helps the driver to detect the right place to stop the car inside a narrow area between other cars without a collision, through sensors to detect the appropriate space , which is the ultrasonic and also to measure the distance needed to advance and return, and also the encoder was used to measure the number of turns The engine that works to drive and move the wheels, this motor engine controlled by Arduino by using driver to link between Arduino and DC motor.

Abbreviations:

DC	**Direct current**
TX	**Transmit**
RX	**Receive**
VCC	**Voltage, Common Collector**
GND	**Ground**
PC	**Personal Computer**
USB	**Universal Serial Bus**
IDE	**Integration Development Environment**
CPU	**Central Processing Unit**
EEPROM	**Electrically Erasable Programmable Read-Only Memory**
ADC	**Analog to Digital Converter**
SRAM	**Synchronous Access Memory**
PWM	**Pulse With Modulation**

Table of Contents:

List of Figures:

List of Tables:

Chapter 1:Introduction

1.1 Overview of the project:

Microcontrollers are used to develop intelligent systems that can be found in different applications in our daily lives. In this project, we aim to design Self-Parking Car using Arduino.

Self-car driving we are developed autonomous cars using the ultrasonic sensor. In this car, we used three ultrasonic sensors to find the obstacles sides. If anyone side obstacles detected means the car will move in another direction.

1.2 Motivation:

The main motivation for building this project is how self-parking works by building a system that stops the car between two cars on its own, without hitting any surrounding car, and this project reduces many of the problems that people suffer from.

1.3 Goals and Objectives:

The goal of this project is to build a Self-Parking Car systems that achieves the following objectives:

1: Exploiting narrow spaces.

2: Helping drivers with special needs.

3: Helping people who do not have high driving skills.

4. Reducing traffic accidents.

1.4 Short description of the project:

A system that works to stop a car automatically between two cars parked properly without colliding with them, by using the appropriate parts for the self-parking car.

1.5 Problem Statement:

Most of the time people suffer from the problem of parking the car in the appropriate places due to high traffic demand ,Following the rapid increase of car ownership, many cities are lacking car parking areas. So this project was chosen, which aims to stop a car between two cars independently.

The advantages of this project are:

1: No need to worry about the cars from accidents.

2: Reducing the worry of people who do not have high skills in driving.

3: Helping people with special needs.

1.6 System Requirements:

The system must achieve the following functionality:

1. The system must be user-friendly to be used by a beginner and advanced users.
2. Design a suitable car that achieves the goal of the project.
3. Checking all project parts .
4. High accuracy in the driver.
5. High accuracy in the sensors .

1.7 Expected Results:

We expect to accomplish the following at the end of the project:

1. A simple, low-cost system used to Self-Parking Car.
2. Stops the car properly.
3. The system reducing the number of collisions between cars.
4. Sensor data should be highly accurate.
5. Install the appropriate code on the board.

1.8 Project Gantt chart:

Task	Duration (week)													
	First semester							Second semester						
	٢	٤	٦	٨	١٠	١٢	١٤	٢	٤	٦	٨	١٠	١٢	١٤
Planning														
Project requirements														
Analyzing and design														
Project development														
Project and testing and maintenance														
Documentation														

Table 1: ***Project Gantt chart***

1.9 Report Outline:

The rest of this report is organized as follows: Chapter 2 introduces some literature review including available self-parking car and related project and talks briefly about the theoretical background of the project, hardware and software components. Chapter 3 discusses the conceptual design of the system, block diagrams, flowchart, and detailed hardware connection.

Chapter 2:Literature Review and Theoretical

2.1 Background:

This chapter provides a short review of some related systems and introduces a theoretical background of the project, including a description of the hardware and software components used in the system.

2.2 Literature Review:

Based on our research, we came across very limited products that are related to the current project. Following are example project:

2.2.1: Smart Car Parking System using Arduino UNO:

At the point when IoT is increased with sensors and actuators, the innovation turns into an occurrence of the more broad class of digital physical frameworks, which likewise incorporates advances. For Example, keen networks, virtual power plants, brilliant homes, astute transportation and shrewd urban communities. Among the difficulties that confront in everyday life one of most unavoidable test is parking the car wherever people go. As our need expands our setting out increments however because of extreme increment in utilization of vehicles and increment in populace this project confront the intense assignment of parking car especially amid busiest hours of the day. Amid pinnacle hours the majority of the saved parking zone gets full and this leaves the client to scan for their parking among other parking area which makes more movement and abandons them with no sign on accessibility of parking spot. To defeat this issue there is certainly a requirement for composed parking in business condition. To outline such parking there need to assess reservation of parking space with ideal parking spot which relies upon cost and time. However this project compose the time driven grouping strategy which takes care of the issue of parking utilizing opening assignment technique[2].

2.2.2: Self-Parking System Based in a Fuzzy Logic Approach:

This paper describes a control system which automatically parks a scaled automobile inside a rectangular reduced space given certain conditions and making decisions based in fuzzy logic. The control is developed by the processing of entry variable data from simulated sensors of a specific scenario, and the run of three models in cascade to achieve a decision-action method. Finally, this paper shows a description of the way this project was achieved, and concludes with the acceptable results that a fuzzy control can provide in a management of this kind of mechanism by the use of taking decision models[3].

2.3 THEORETICAL BACKGROUND

2.3.1 General Description:

Traffic crisis are one of the most common problems and lack of knowledge of determining adequate parking space. There are several rules to observe:

1. Choose the right car skeleton to place all parts on.
2. Ultrasound: It must determine the correct and appropriate distance ,operate with frequencies from 20 kHz up to several gigahertz.
3. A DC motor (Direct Current motor) : normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate. If you switch the leads, the motor will rotate in the opposite direction.
4. L298N Motor Driver : one of the easiest and inexpensive way to control DC motors . It can control both speed and spinning direction of two DC motors.
5. The optical encoder : High accuracy in determining the measuring rotational motion.

2.4 Hardware System Components:

This section describes all hardware used in our project. It presents a figure for each one with short description about its work principle and why it is used in the system.

1. Microcontroller:

During my search, we`ve encountered microcontroller alternatives options.

•Arduino Mega 2560:

The Arduino Mega 2560 is a microcontroller board , and the Mega is the addition to the Arduino family. It has 54 digital input/output pins , 16 analog inputs, 4 UARTs a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button .This board is physically larger than all the other boards ,and the MEGA uses a different processor allowing greater program size and more ,the Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip .The Arduino Mega works in the same way the Arduino Uno does but the difference is that it uses ATmega2560 microcontroller and has more number of pins. [5].

differences between Raspberry Pi and Arduino:

Raspberry Pi	**Arduino**
It is a mini computer with Raspbian OS .It can run multiple programs at a time.	Arduino is a microcontroller, which is a part of the computer. It runs only one program again and again.
It is difficult to power using a battery pack.	Arduino can be powered using a battery pack.
It requires complex tasks like installing libraries and software for interfacing sensors and other components.	It is very simple to interface sensors and other electronic components to Arduino.
It is expensive.	It is available for low cost.
Raspberry Pi can be easily connected to the internet using Ethernet port and USB Wi-Fi dongles.	Arduino requires external hardware to connect to the internet and this hardware is addressed properly using code.
Raspberry Pi did not have storage on board. It provides an SD card port.	Arduino can provide onboard storage.
Raspberry Pi has 4 USB ports to connect different devices.	Arduino has only one USB port to connect to the computer.
The processor used is from ARM family.	Processor used in Arduino is from AVR family Atmega328P.
This should be properly shutdown otherwise there is a risk of files corruption and software problems.	This is a just plug and play device. If power is connected it starts running the program and if disconnected it simply stops.
The Recommended programming language is python but C, C++, Python, ruby are pre-installed.	Arduino uses Arduino, C/C.++

Table ٢ :Raspberry Pi VS Arduino

Depending on the characteristics of each type, and keeping in mind the needs of the project, I discovered that Arduino MEGA as shown in Figure 1 is the best choice among the other options because there are features that meet the project requirements as we need them.

Figure 1: arduino mega

2. L298N Motor Driver:

The L298N is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms.

This L298N Motor Driver as shown in figure 2, Module is a high power motor driver module for driving DC Motors. This Module can control up to 4 DC motors, or 2 DC motors with directional and speed control[4].

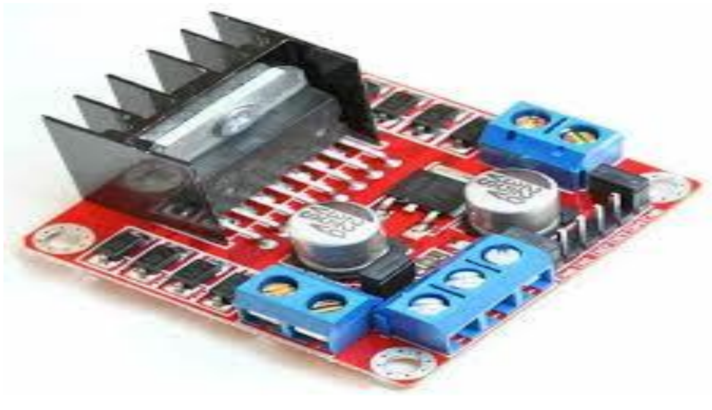

Figure٢ : L298N DRIVER

L298 Module Features & Specifications:

Driver Model:	L298N 2A
Driver Chip:	Double H Bridge L298N
Motor Supply Voltage (Maximum):	46V
Motor Supply Current (Maximum):	2A
Logic Voltage:	5V
Driver Voltage:	5-35V
Driver Current:	2A
Logical Current:	0-36mA
Maximum Power (W):	25W

Table ٣ :L298 Features & Specifications

3. HC-SR04 Ultrasonic Sensor:

HC-SR04 as shown in figure 3, is ultrasonic ranging sensor used to measure the distance to an object with high accuracy and stable readings. It consists of one ultrasonic transmitter, a receiver and control circuit. The transmitter will emit high frequency sound which bounce off any nearby solid object and the some of the sound will be reflected and detected by the receiver of the sensor. The emit signal and return signal will be proceeding by the control circuit in order to calculate the time different between them. Then using some simple formula, the distance between the sensor and reflected object will be calculated[6].

Figure 3: Ultrasonic

shows some specifications of HC-SR04:

Operating Voltage	5.0 (V) DC
Working Current	15(mA)
Ultrasonic Frequency	40 kHz
Max Range	4 (m)
Min Range	2 cm

Table ٤ : HC-SR04 Specifications

4. DC MOTOR :

A direct current or DC motor, converts electrical energy into mechanical energy. It is one of two basic types of motors: the other type is the alternating current or AC motor. Among DC motors, there are shunt-wound, series-wound, compound-wound and permanent magnet motors.

A DC motor consists of an stator, an armature, a rotor and a commutator with brushes. Opposite polarity between the two magnetic fields inside the motor cause it to turn. DC motors are the simplest type of motor and are used in household appliances, such as electric razors, and in electric windows in cars.

Figure 4: DC motor

5. encoder:

The optical encoder is a transducer commonly used for measuring rotational motion. It consists of a shaft connected to a circular disc, containing one or more tracks of alternating transparent and opaque areas. A light source and an optical sensor are mounted on opposite sides of each track. As the shaft rotates, the light sensor emits a series of pulses as the light source is interrupted by the pattern on the disc. This output signal can be directly compatible with digital circuitry. The number of output pulses per rotation of the disc is a known quantity, so the number of output pulses per second can be directly converted to the rotational speed (or rotations per second) of the shaft. Encoders are commonly used in motor speed control application[9].

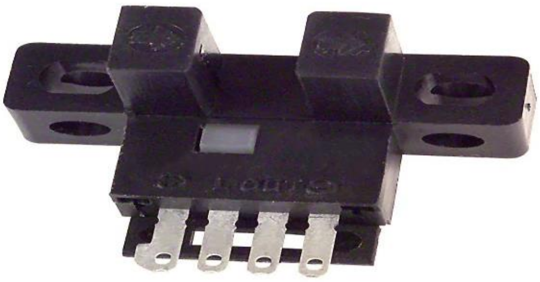

Figure٥ : Encoder

6. car robot chassis:

Another important component we had to choose was the chassis of the car, which is the most important component of the project through the figure showing several types of car bodies[8].

Feature/Item	 Approach 1 College lab & grad. student	 Approach 2 college lab	 Needed item In project ✓
Wheel	4	3	4
Feedback position	No	No	yes
price	45$	10$	60$

Table 5:car robot chassis

2.5 System Software Component:

This section will provide some information about the main programs and software technologies used in my project.

2.5.1 Arduino (IDE):

The Arduino Integrated Development Environment or Arduino Software (IDE) is an open source, contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It runs on various operating systems[1]..

2.6 Constraints:

A. The Arduino microcontroller does not support working multiprocessor or multithread, but we need to run multi sensor simultaneously and this case impossible
B. The car has no direction steering and we have rotated by moving one wheel
C. The vehicle structure was not balanced after installing the motors and controllers
D. The encoder does not measure correctly because it needs measuring at the same time of reading from the ultrasonic and the Arduino does not support multi-process
E. Because the battery power changes when the energy in it is consumed, it causes the movement to become confused or stopped .In the event that there is any defect in any part that leads to the system not working properly.

Chapter 3 :

System Design:

This chapter discusses the conceptual design of the system, it shows block diagram of system, flow chart, wiring diagram, design construes.

3.1 General Mechanism:

The system consists of the microcontroller ,encoder , driver and sensors that receive the configurations , and according to these configurations the Arduino Mega will trigger the sensors, and other elements to run the system as required.

The following steps demonstrate the mechanism of system work:

1) The microcontroller reads data from ultrasonic and encoder and makes decisions to control the system.
2) After the Microcontroller receives commands, it edits its default settings according to the new commands.
3) The Microcontroller send the data to driver and to wheel.

3.2: Sub Circuits:

3.2.1: Block Diagram :

Figure 6 shows the general block diagram for the entire system, showing how the project works and where data is sent and received between pieces.

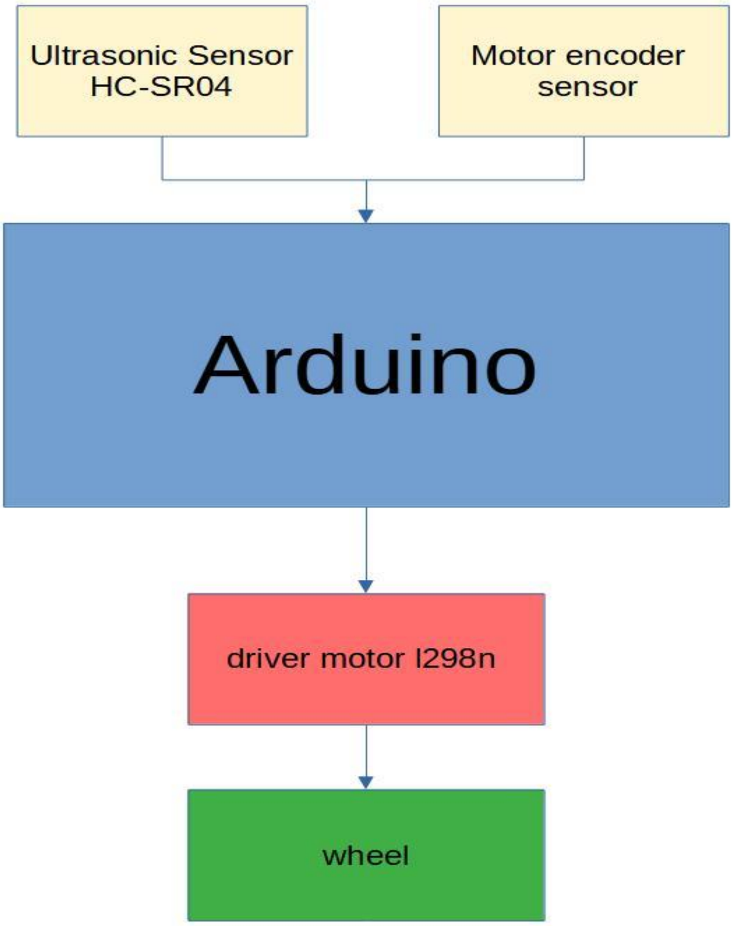

Figure٦ : block digram

3.2.2: Block Diagram :

The figure shows the correct parking mechanism:

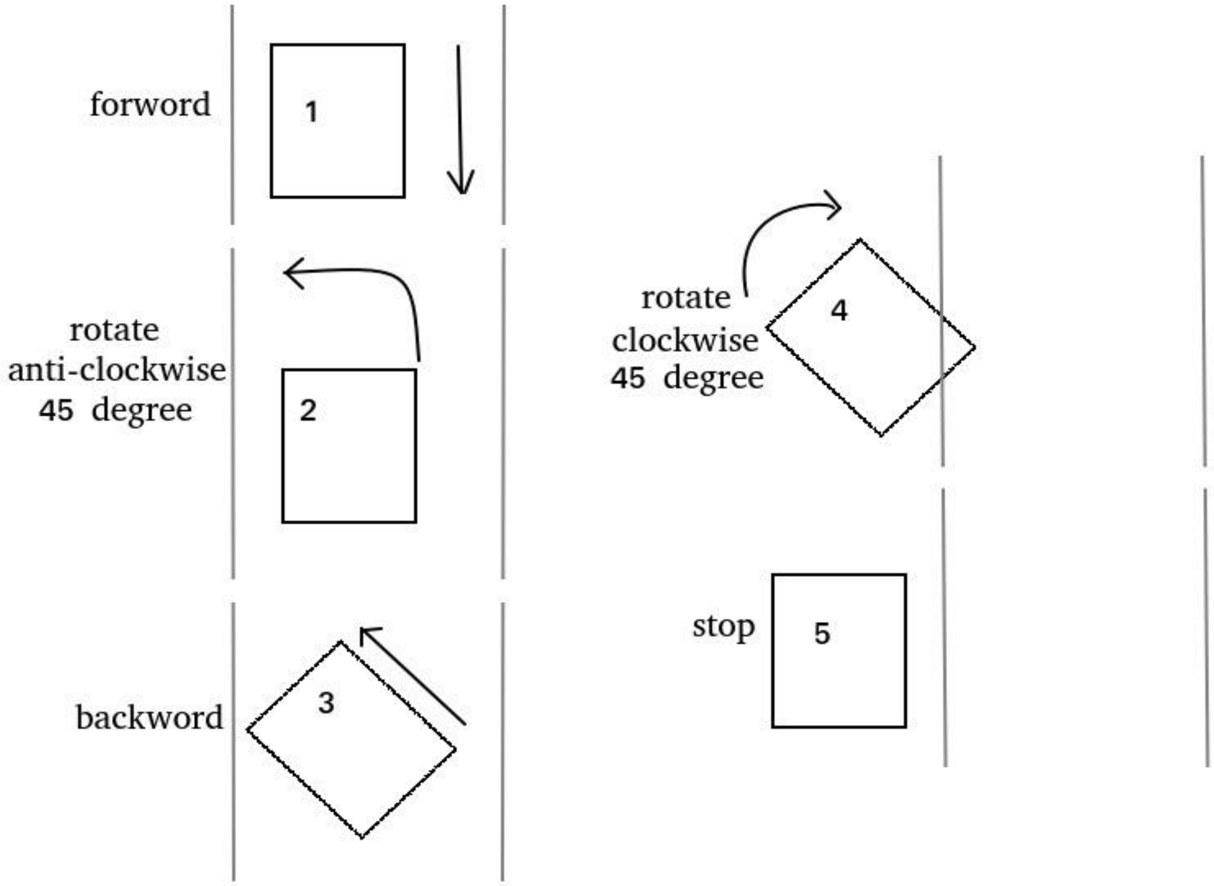

figure٧ :Mechanism to stop the car.

3.2.2: Flowcharts :

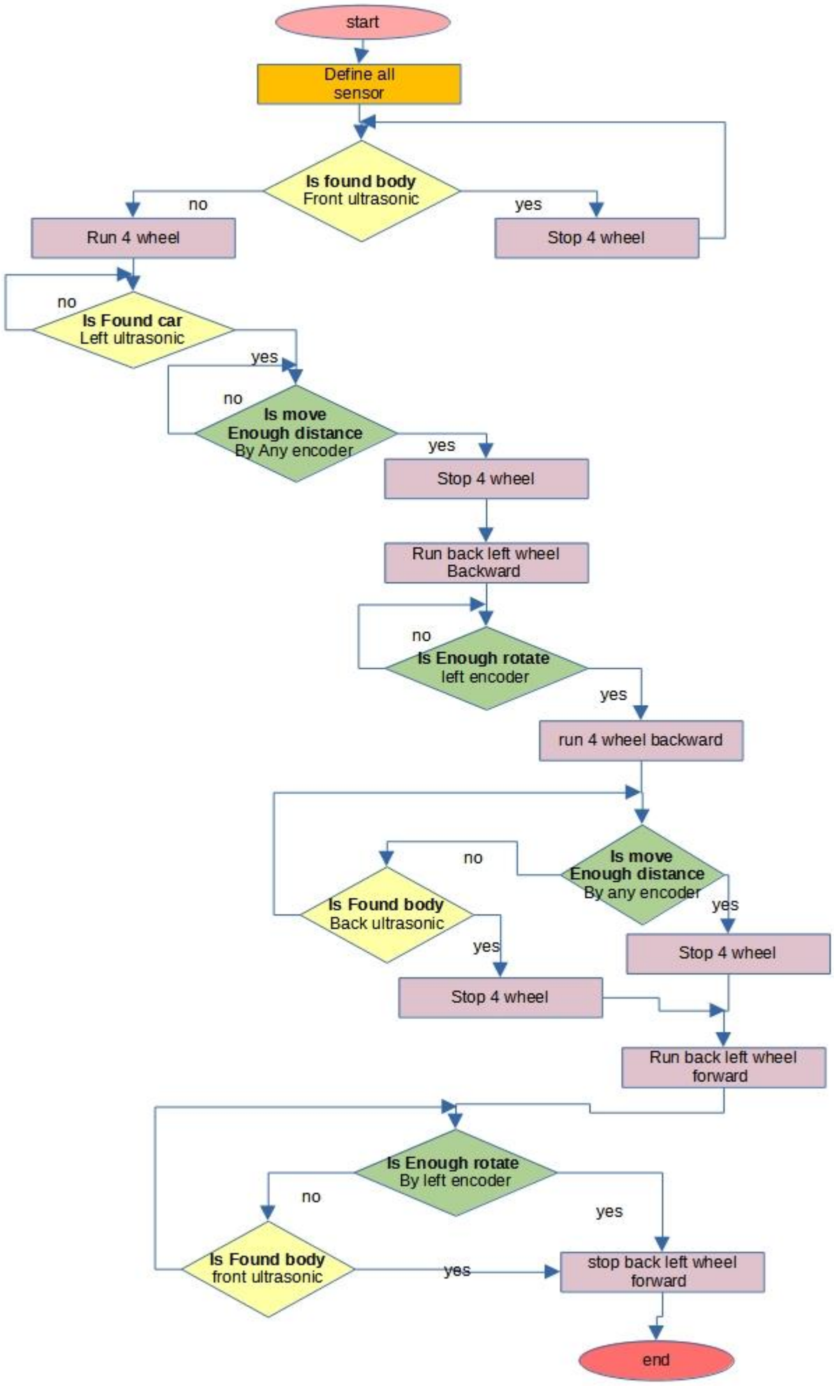

Figure٨ :Flowcharts

3.2.3 Wiring Diagrams:

In this section some component are shown with their connection with the Arduino Mega and some components the connection together the pins of each equipment will be connected physically as shown in the following figures.

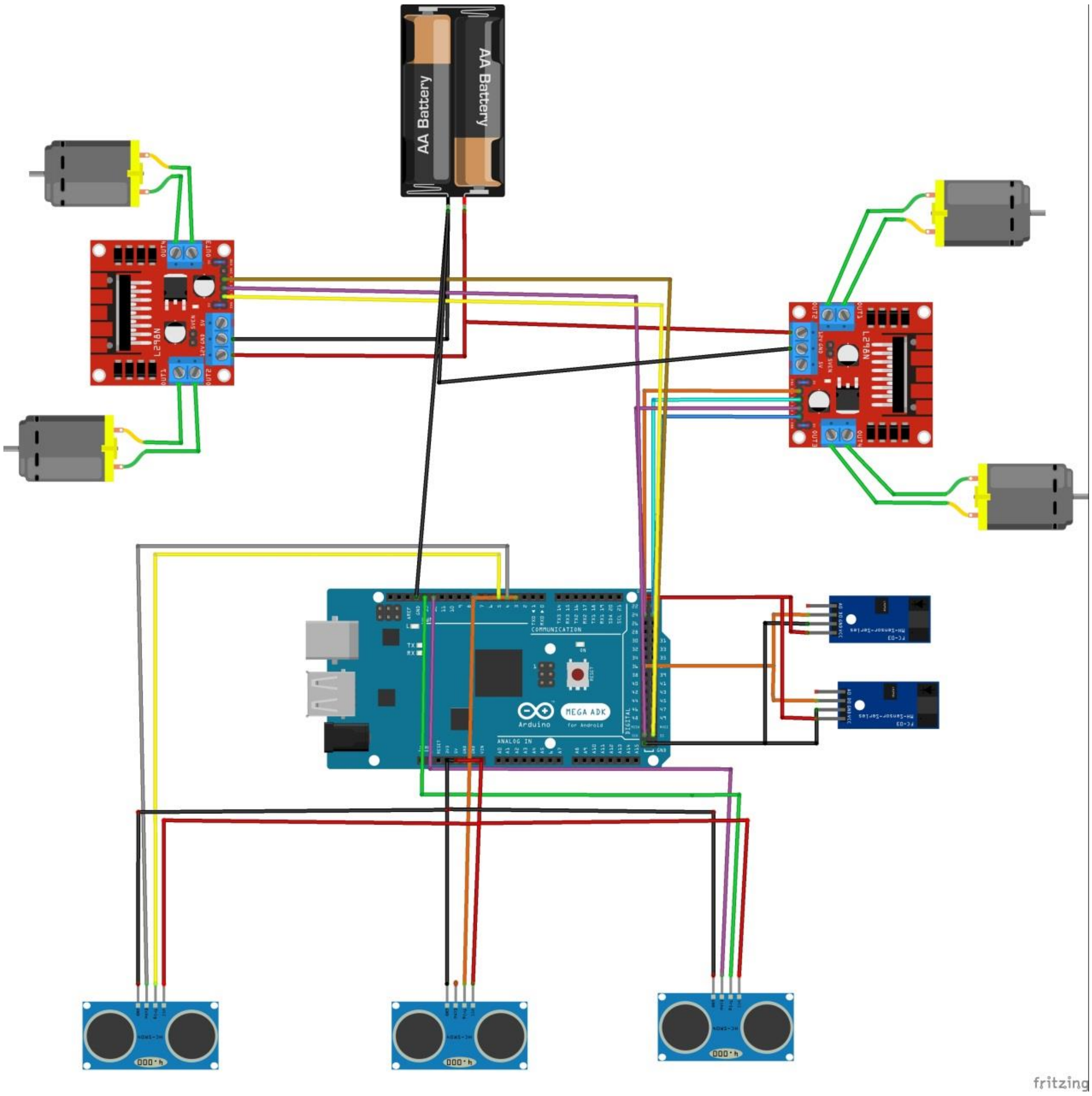

Figure٩ :Wiring Diagrams

3.2.3.1 Wiring Diagrams of ultrasonic:

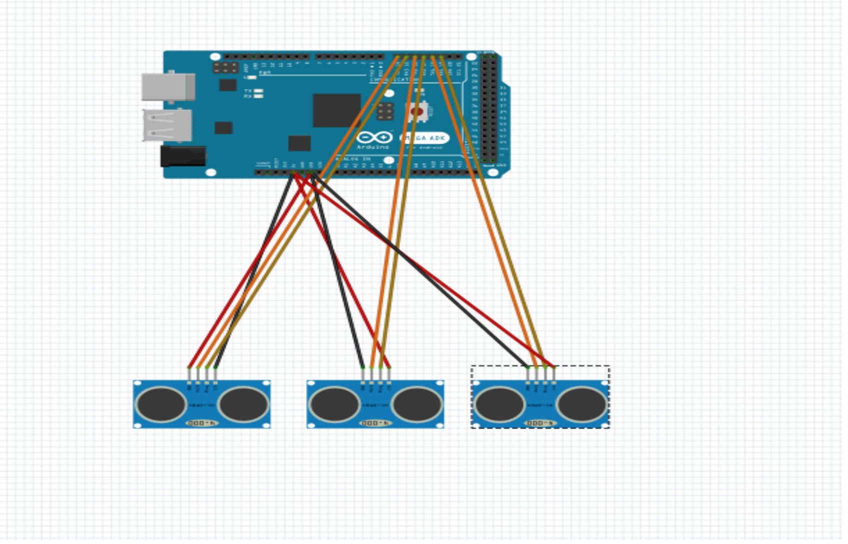

Figure 10: ultrasonic wiring

3.2.3.2 Wiring Diagrams of encoder :

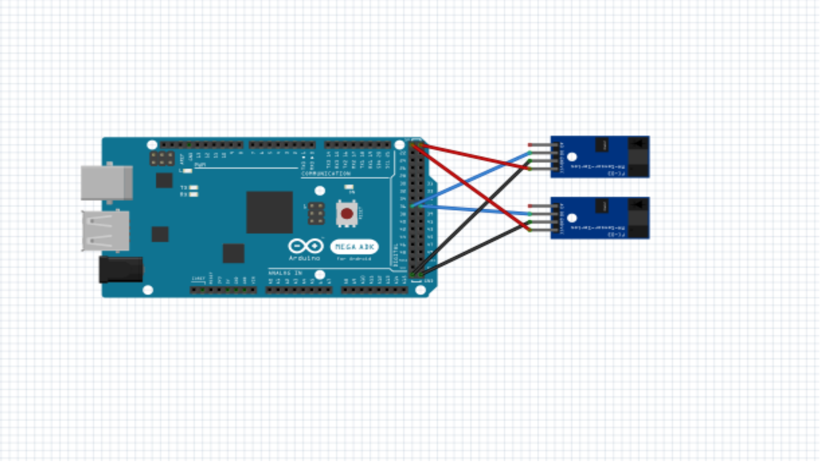

Figure 11 :encoder wiring

3.2.3.3 Wiring Diagrams of driver and DC MOTOR :

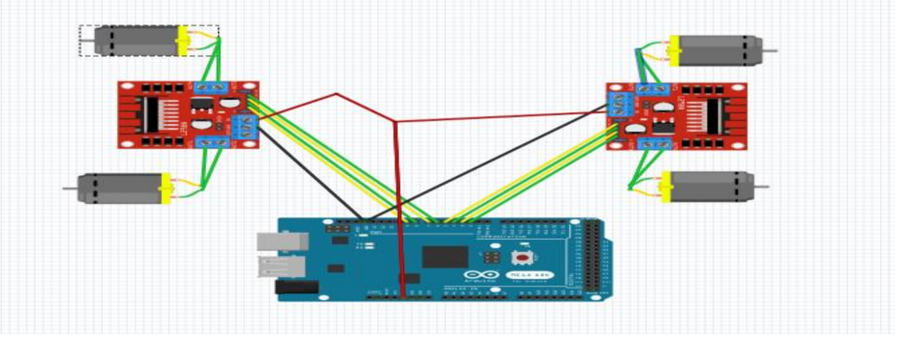

Figure 12:Wiring driver and DC motor wiring

3.2.3.4 Wiring Diagrams of driver and DC motor and battery :

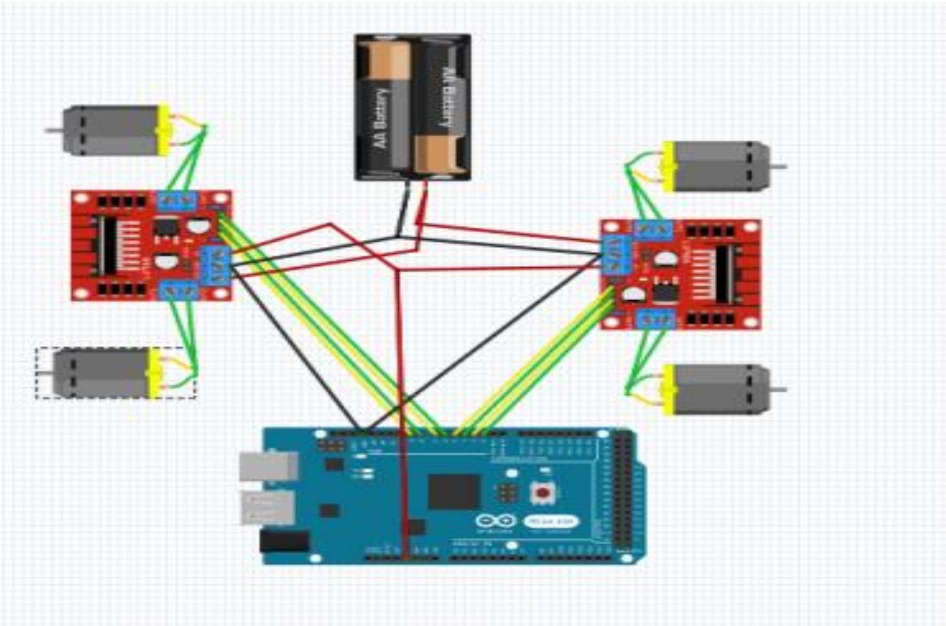

Figure 13: driver and DC MOTOR &battery wiring:

3.2.4 : System Schematic diagram:

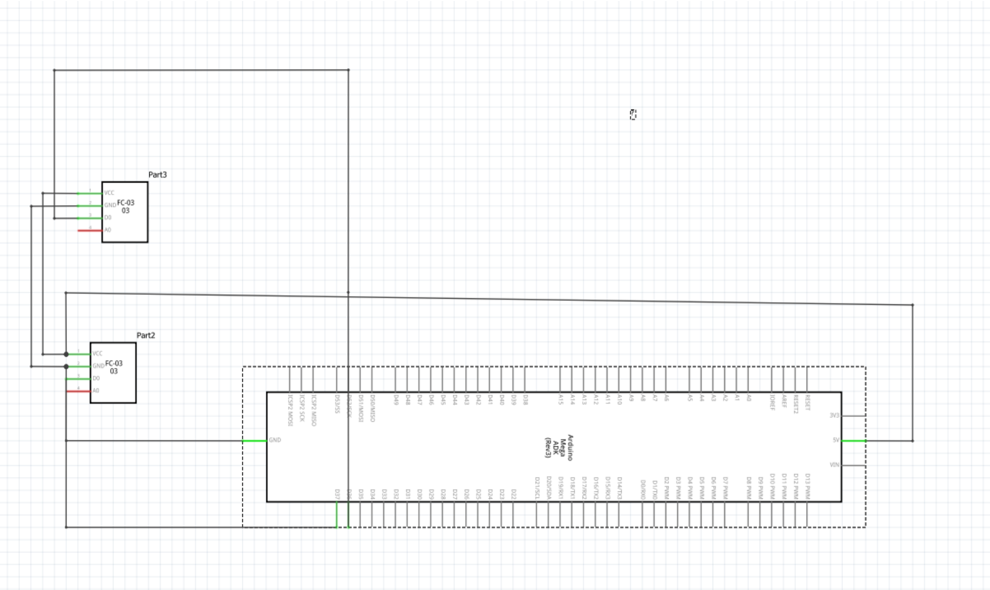

Figure١٤ :Schematic diagram of encoder.

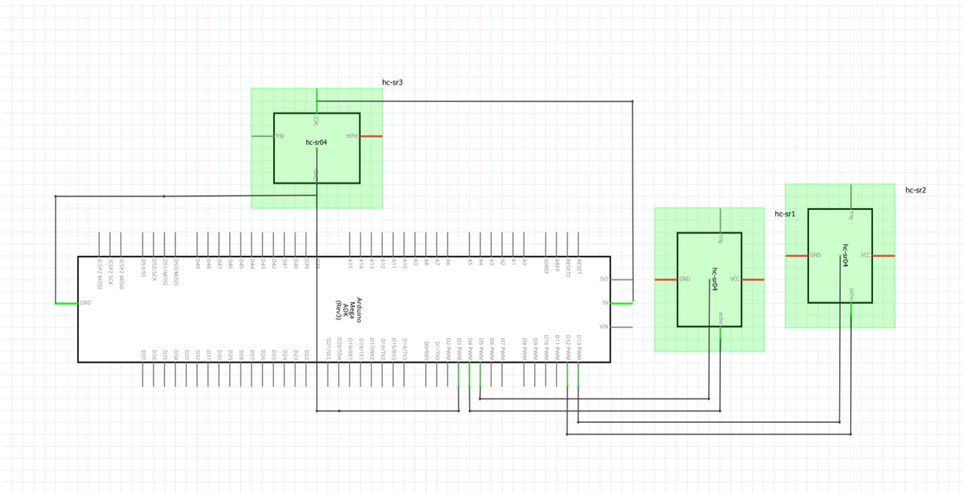

Figure١٥ :Schematic diagram of ultrasonic.

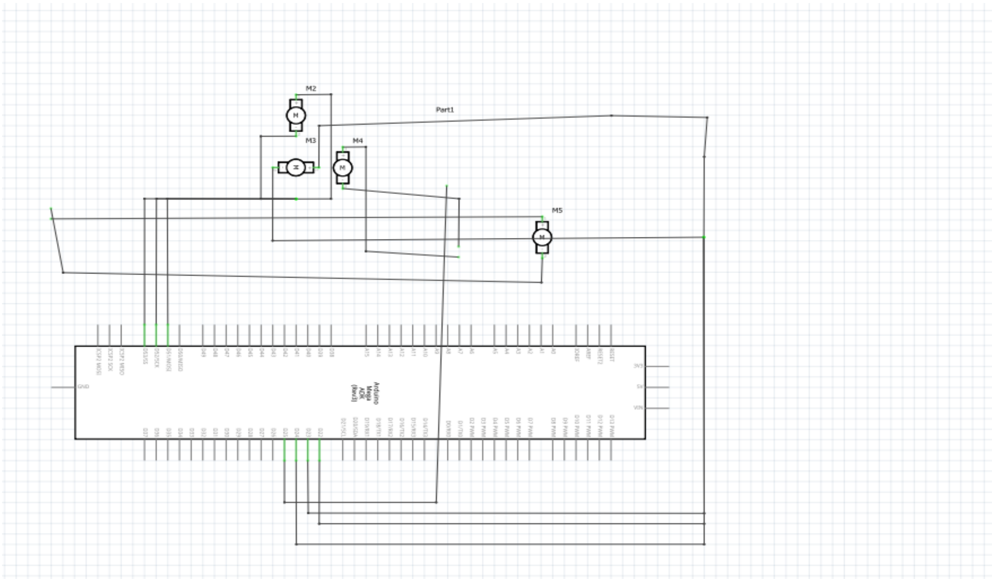

Figure١٦ :Schematic diagram of driver.

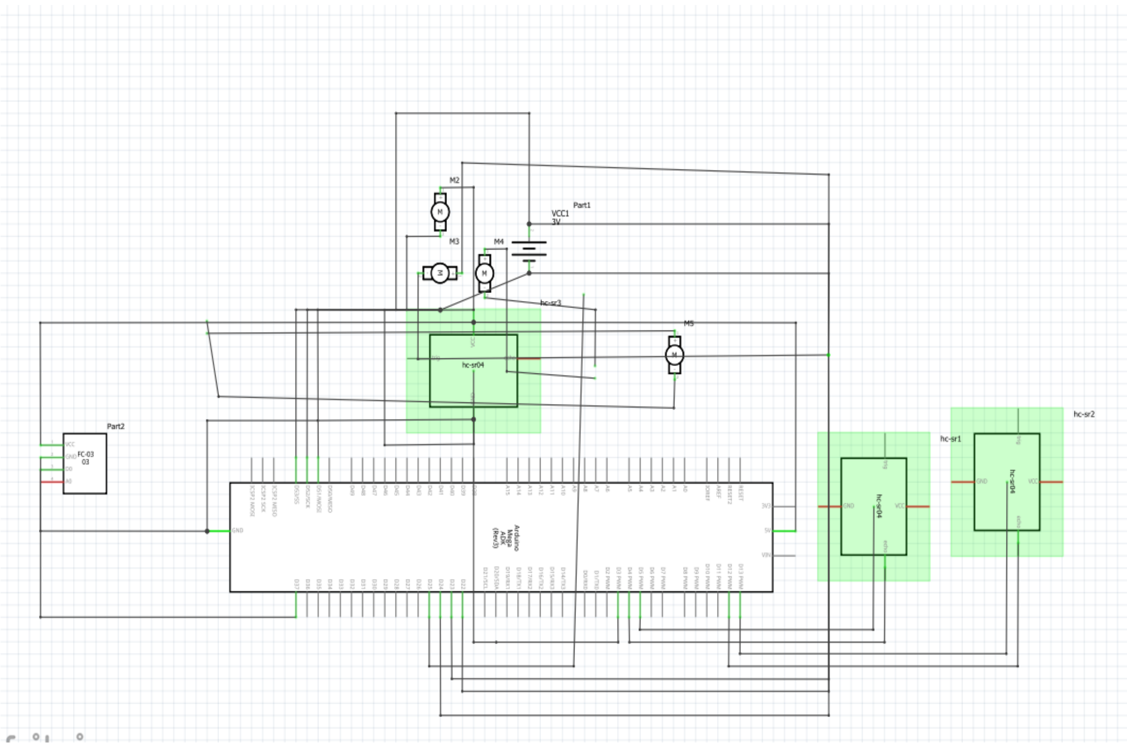

Figure١٧ :Schematic diagram of projects.

Chapter 4:
Software & Hardware Implementation

4.1 Overview

This chapter describes the implementation of the software and the hardware of this project, including the circuit connection, and programming of the microcontrollers.

4.2 Software Implementation tools

This section will provide some information about the main programs and software Technologies used in project.

4.2.1 Arduino IDE

We are using the Arduino IDE to program the Arduino microcontroller and write code for the sensor, and other associated components to Arduino .

4.3 Hardware Implementation

This section will provide some information about the hardware implementations of project.

4.3.1: Arduino configuration:

The figure 14&15 shows how the parts are attached to the car:

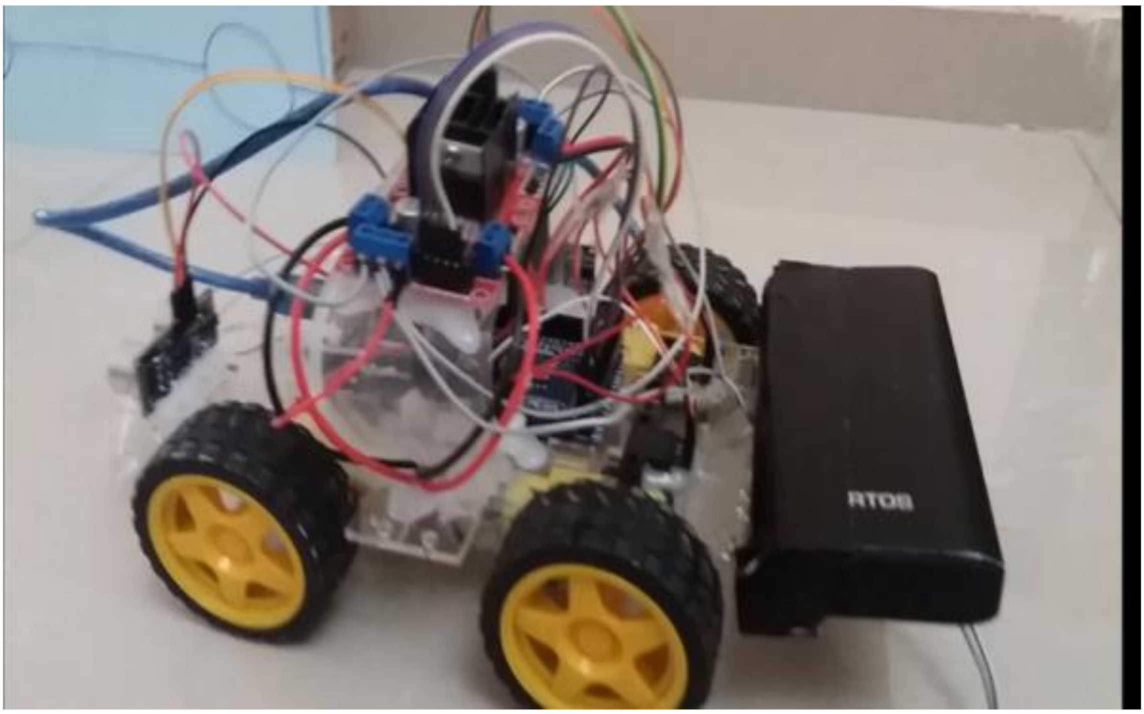

Figure 18:Arduino configuration

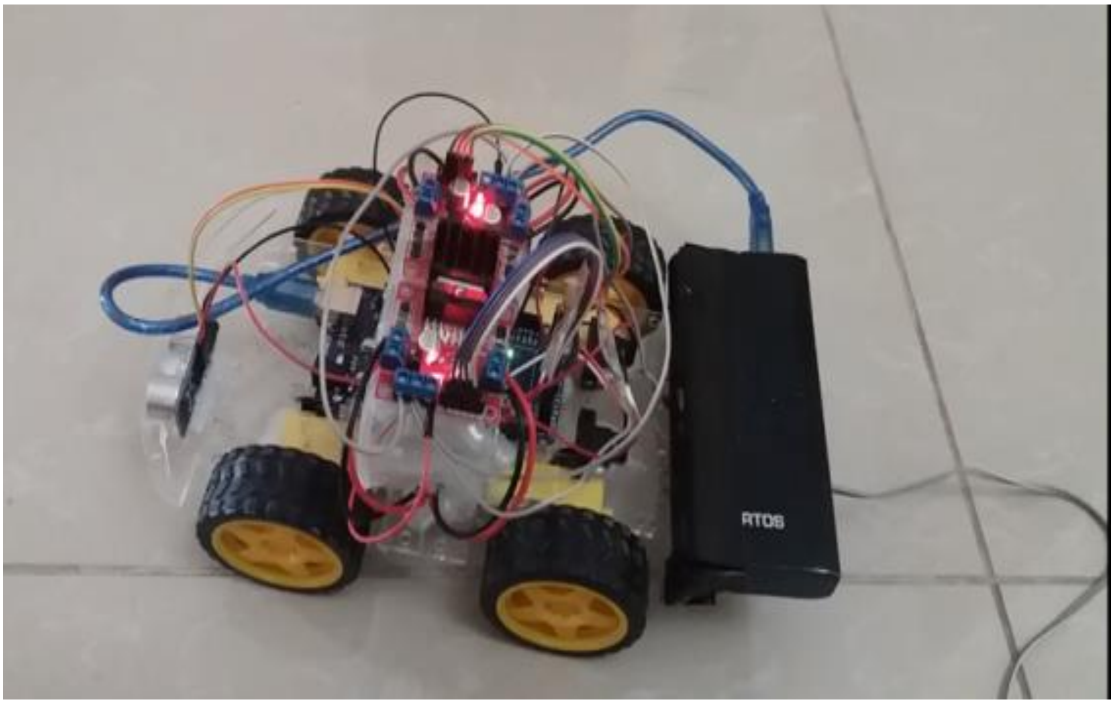

Figure 19:car configuration

4.4 Implementation Results

At the end of the implementation process, the self-parking system was built for the car by programming the system correctly and linking the parts together to perform the tasks required of the self-parking car, as the figure shows how the car stops in the empty place and not the place where another vehicle is.

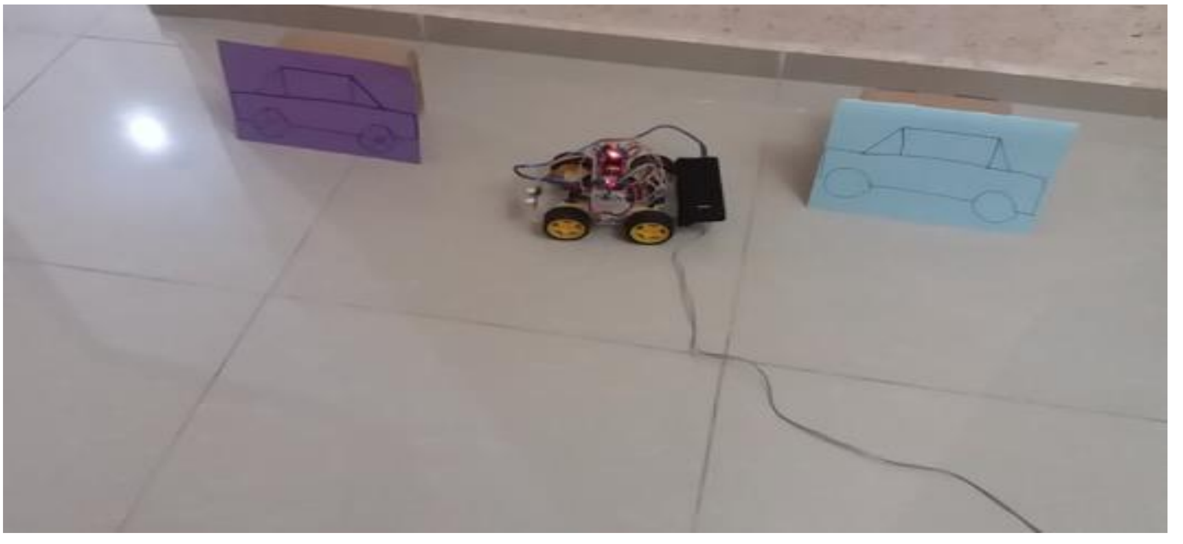

Figure 20:car results

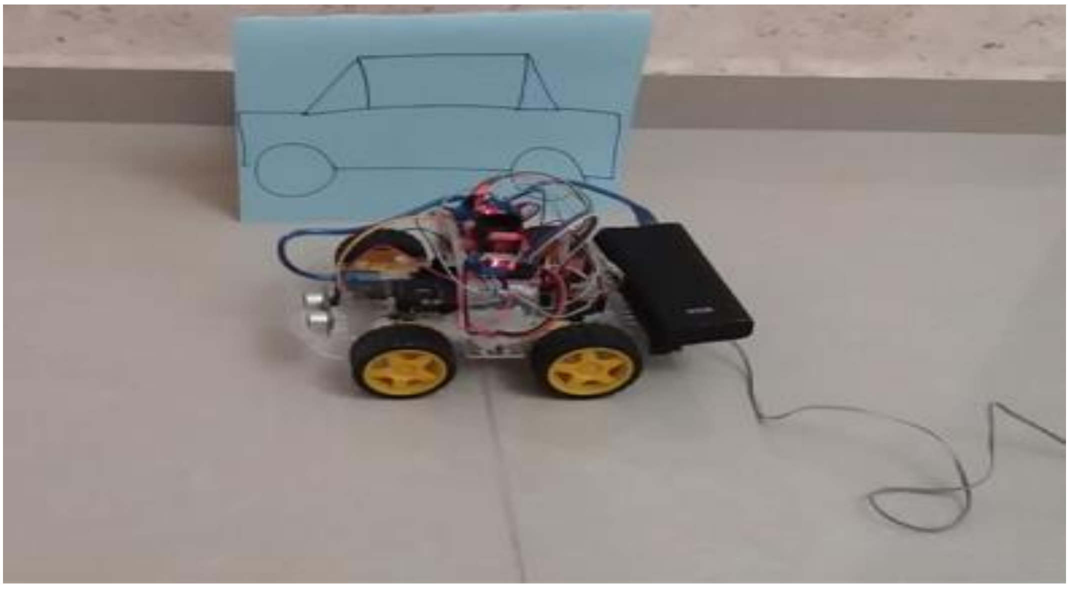

Figure 21:car results

figure 22:car results

4.5 Implementation Issues

During the implementation of the project, we faced many obstacles and we had to take Several issues to achieve the most appropriate design for the system and access to the best features Related to the objectives of the project. These issues and results are summarized as follows:

The problems:

1) Ultrasound: determine the distance?
2) The connection between the Arduino device and other parts?
3) Code programming?

The solution :

1. Ultrasound: determine the correct distance:
2. The connection between the android device and other parts.
3. Programming the parts appropriately.

Chapter 5
Validation & Testing

5.1 Overview

In this chapter we will discuss the testing of all component of the system and the results obtained. We test all the parts to ensure that all of the functions work perfectly and without errors.

5.2 Software Testing:

5.2.1 Code in Arduino IDE :

In the figure 23&24: shows parts of the code that have been worked on to link all parts of the project to each other and program them to perform the required function in an elaborate and effective manner[10],

To check and test the validity functions:

1. Determine the distance.
2. Run driver .
3. Run encoder .
4. Run DC motor .

5.3 Hardware Testing

In this section we will discuss the testing of components.

5.3.2 Testing move of the car :

At the end of the implementation process, the self-parking system was built for the car by programming the system correctly and linking the parts together to perform the tasks required of the self-parking car, as the figure shows how the car stops in the empty place and not the place where another vehicle is.

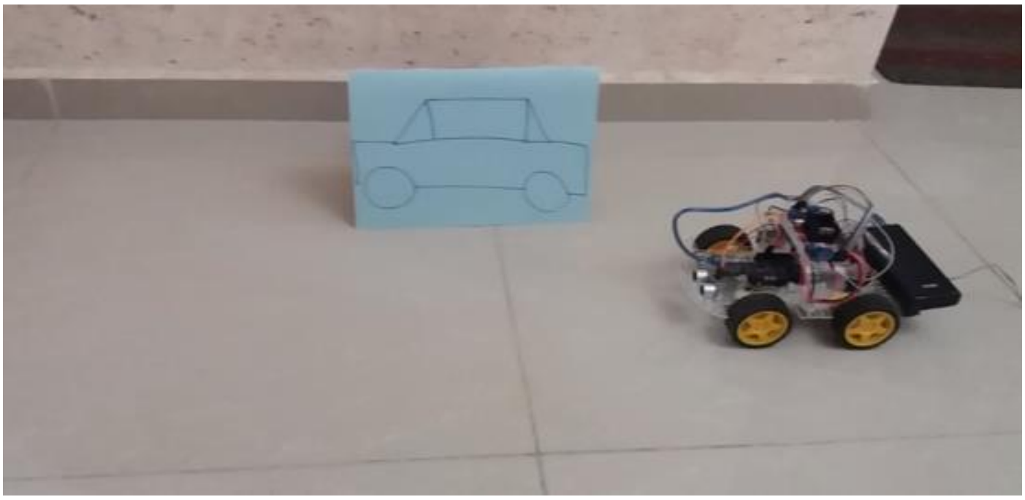

Figure 23:Testing move of the car

5.4 System Test

After ensuring that all the parts are working properly, we started assembling and integrating the parts with each other to make the system ready to operate.

5.4.1 Arduino mega, ultrasonic sensor , DC motor, Driver integration test

After making sure that each piece was done separately and examined well, all the project parts were connected to each other and making sure of their work, and then the code was loaded on the Arduino Mega and the car's working mechanism was tested accurately.

system was completed and tested successfully and it works as required without any major problems.

Chapter 6
Conclusion & Future work

6.1 Overview

In this chapter, we will conclude the challenges, the final result of the system worked on, and the future work of the project.

6.2 Final Result

The system was able to stop a car between two cars without colliding with either of them through the use of sensors that work to determine the appropriate distance to stop this car correctly, as the distance values are sent to the Arduino mega and it works on processing these values and analyzing them in order to give the vehicle orders to move correctly.

The main system components namely: Arduino Mega, Ultrasound, driver , Encoder ,are integrated together to form an integrated system that reads and processes sensor values to do the required functions.

6.3 Conclusion

The project was ultimately successful, we were able to build the car, interface between sensors and motor drivers, and in the end, program self-parking car behavior. However, there were several challenges that arose during the process. The largest problem that came up during the project was the chassis car itself. It is a common problem in robotics for the many project ,and the program of code .

While the previous two challenges did inhibit our experimental results to a certain extent, we were ultimately successful and being able to see self-parking car . With so many factors that could contribute to incorrect behavior , and finally we were satisfied with the end result of this project.

6.4 Futures Works

In the future, the system can be developed by adding more efficient and effective parts to perform new functions such as determining the number of cars and determining the number of suitable places to park the car, and the functionalities of the system are open for further extent.

References:

[1] Arduino IDE, Retrieved September 20 2020,from URL: https://www.arduino.org/

[2] Smart Car Parking System using Arduino UNO, Retrieved September 3 2020, from URL: https://www.researchgate.net/publication/318486476_Smart_Car_Parking_System_using_Arduino_UNO

[3] Self-Parking System Based in a Fuzzy Logic Approach, Retrieved September 11 2020, from URL: https://ieeexplore.ieee.org/abstract/document/5163901

[4] Interface L298N, Retrieved October 17 2020,from URL: https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

[5] Getting Started with Arduino MEGA2560, Retrieved October 1 2020,from URL: https://www.arduino.cc/en/Guide/ArduinoMega2560

[6] Ultrasonic Sensor, Retrieved October 7 2020,from URL: https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm#:~:text=Advertisements,or%201%E2%80%9D%20to%2013%20feet.

[7] Fritzing , Retrieved August 30 2020,from URL: https://fritzing.org/

[8] 4 wheel drive, Retrieved August 27 2020,from URL: https://roboindia.com/store/smart-car-chassis-4wd

[9] Encoder-disk, Retrieved August 30 2020,from URL: https://create.arduino.cc/projecthub/mitov/measure-motor-speed-rpm-with-optocoupler-and-encoder-disk-c3a0e4

[10] Run a sketch in Arduino IDE , Retrieved September 30 2020,from URL: https://jayconsystems.com/blog/run-a-sketch-in-arduino-ide