

بسم الله الرحمن الرحيم

AN-NAJAH NATIONAL UNIVERSITY



Computer Engineering Department

Hardware Graduation Project

# Parking Robot

students

Bayan KHAROSHEH and Tasneem JAWABREH

[bayankharosheh@gmail.com](mailto:bayankharosheh@gmail.com)

[tasneemkjawabreh@gmail.com](mailto:tasneemkjawabreh@gmail.com)

Under the supervision of

Dr. Ashraf Armoush

A report submitted in partial fulfillment of the requirements for  
bachelor's degree in computer engineering in the Faculty of Engineering  
Information Technology - Hardware Project

MAY, 2022

## Acknowledgements

We would like to express our deep gratitude for Dr. Ashraf's serious efforts and valuable time. We were able to complete this job thanks to his great advice and input. Thanks should also go to AN-Najah University which gave us the opportunity to study. we are grateful to our special professors in the Computer Engineering Department, Faculty of Engineering who generously offered their knowledge and tracked our progress as university students. we would also like to extend our gratitude to our families who are always by our side and supported us. they gave us the strength to complete this project. Many thanks to our friends also who supported us at all times. Finally, we would want to express our thanks to everyone who assisted and encouraged us to work on this project.

Bayan Tasneem.

## **Disclaimer**

This report was written by students at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Constraints, Standards/ Codes and Earlier course work</b>	<b>7</b>
2.1	Constraints . . . . .	7
2.1.1	mechanical parts . . . . .	7
2.1.2	robot movement . . . . .	7
2.1.3	rotate the robot in a specific angle . . . . .	7
2.1.4	raspberry pi . . . . .	7
2.2	Earlier Courses . . . . .	8
2.2.1	Microcontroller . . . . .	8
2.2.2	Arduino course . . . . .	8
2.2.3	CPU lab . . . . .	8
<b>3</b>	<b>Literature Reviewn</b>	<b>9</b>
<b>4</b>	<b>Design and Implementation</b>	<b>10</b>
4.1	the garage design . . . . .	10
4.2	Hardware and Software Specifications . . . . .	12
4.2.1	Hardware Components . . . . .	12
4.2.2	Software Components . . . . .	17
4.3	Arduino Code . . . . .	18
4.4	Image processing Code . . . . .	21
4.5	Circuit Layout . . . . .	22
<b>5</b>	<b>Results and Analysis</b>	<b>23</b>
<b>6</b>	<b>Conclusions and Recommendation</b>	<b>27</b>

## List of Figures

1	Figure 4.1.1: Robot design . . . . .	10
2	Figure 4.1.2: Robot crane . . . . .	10
3	Figure 4.1.3: garage design . . . . .	11
4	Figure 4.1.4: garage wall . . . . .	11
5	Figure 4.2.1: Arduino Uno . . . . .	12
6	Figure 4.2.2: HC-05 Bluetooth Module . . . . .	13
7	Figure 4.2.3: RC522 RFID Module . . . . .	13
8	Figure 4.2.4: Buzzer . . . . .	14
9	Figure 4.2.5: Arduino Mega . . . . .	14
10	Figure 4.2.6: HC-05 Bluetooth Module . . . . .	15
11	Figure 4.2.7: L298N Dual H Bridge . . . . .	15
12	Figure 4.2.8: DC motor . . . . .	16
13	Figure 4.2.9: IR . . . . .	16
14	Figure 4.2.10: Raspberry Pi4 with Camera . . . . .	17
15	Figure 4.3.1: RFID Bluetooth code . . . . .	19
16	Figure 4.3.3: check empty parking space . . . . .	20
17	Figure 4.3.2: check parking number code . . . . .	21
18	Figure 4.5.1: Robot implementation . . . . .	22
19	Figure 4.5.2: scanner implementation . . . . .	23
20	Figure 5.1: the robot move to pring the car . . . . .	24
21	Figure 5.2: forklift lifts the car from the ground . . . . .	25
22	Figure 5.3: park the car . . . . .	26
23	Figure 5.4: park the car . . . . .	27

## Abstract

Many people have difficulties when they want to park their cars and it takes time. Our project aims to solve this problem by making a robot that works in the garage. It takes the car from the user at the entry of that garage and parks it at an empty parking space. When the user asked for his car, the parking robot will bring it back to him. The garage also has a scanner at the entry to make sure that no one can park his car without his parking card. Also, it warns them if there is no empty parking space. Our robot has a long and tiny arm to pick up the car and a camera which check if there is empty space. it will use image processing to select the empty space and move through it.

# 1 Introduction

Robots are used in multiple areas, especially where they can alleviate strenuous tasks or complete missions that are dangerous for a human to undertake. A parking robot is a robot designed to help people to park their cars. they just leave it in the garage entry and the robot does it for them. also, many Accidents occur often while parking or reversing cars. this robot works to lift the car and park it accurately without any human effort. in our project we built a model of the parking robot system. we built the robot and program it to take the car when someone orders him.

We executed this in two stages. Starting by designing the robot and the garage, collecting the hardware parts, and programming the entry of the garage. Then programmed the robot to move so it can get the car and move an angle to park it. In the second stage, we work on how the robot will decide where to park the car. we write image processing code so the robot can take an image and parking the car depending on what it sees in the image.

Our robot is relatively cheap, its work depends basically on the IR sensor. the robot moves on the main road and enters the garage and exits by following the IR sensor. also, it moves at a 90-degree angle when it parks the car and takes it back. we used Arduino to program the hardware part and we used raspberry pi and its camera to implement image processing code.

In this report, we will show our work. we will start with chapter 2 Constraints and Earlier course work. in this chapter, we talked about the issues and hurdles we ran into at the start of the project and throughout the process. We also talked about projects and courses that helped us enhance our skills and expertise, which showed up in the project's progress. The next chapter is Literature Review in chapter 3 we looked at several similar works as well as what makes ours unique. Then we narrow our focus to specific project components and discuss the extra functionality in our project. the fourth chapter is the Methodology chapter. here, we discuss the strategy of this project, from collecting the hardware parts and the robot body, choosing the design, connecting the circuit diagram, and wiring everything together, to writing the code and testing the final project. the chapter before last is Results and Analysis where we present and discuss our final findings. the last chapter is the Conclusions and Recommendation where we argued and discussed our findings.

## **2 Constraints, Standards/ Codes and Earlier course work**

### **2.1 Constraints**

#### **2.1.1 mechanical parts**

Our lack of experience in mechanical parts makes us face many problems, starting from finding the store or workshops that sell these parts, and we have some difficulty connecting these components.

#### **2.1.2 robot movement**

The accuracy of the electrical component that we find wasn't good enough to give us the actual position of the robot because the performance of each dc motor is different from the other and varies with time. so we use IR sensors to make sure that the robot will still move on a specific path.

#### **2.1.3 rotate the robot in a specific angle**

We face some problems when we try to rotate the robot in a specific angle first we try to make that by servo motor connected with the wheels to control the direction but the mechanical part of these wheels wasn't have good flexibility to be able to move by servo motor. After that we try to put one wheel in the front of the robot and the other 2 wheels in the rear of the robot, his design helps us to rotate the robot but it still not accurate to give us a specific angle so we use an IR sensor to make sure that the robot rotates 90 degrees.

#### **2.1.4 raspberry pi**

We face a lot of problems when we start dealing with raspberry pi, the first problem was that it turns off after a few minutes from running the code on it, to solve that problem we use a fan to let the raspberry pi cold to make it work for a long period of time, the second problem that the camera of raspberry pi is too slow so we make a delay on Arduino to make sure that the camera will take the right image ,the last problem was the connection between raspberry pi and Arduino because sometimes the raspberry pi don't read all signal that came from Arduino so we make the Arduino sending messages until the raspberry respond.

## **2.2 Earlier Courses**

### **2.2.1 Microcontroller**

We don't use PIC Microcontroller as we do in the Microcontroller course but that course give us the basic information about how to program hardware components and the lab for this course was very helpful because we deal with most of the hardware components that we use in our project.

### **2.2.2 Arduino course**

Arduino course help us a lot because most of our work was on Arduino, and we learn how to use a lot of hardware components that we used most of these parts on our project.

### **2.2.3 CPU lab**

That lab teaches us that we should test each component before working and when we have any problems with the work, and we learn how to connect and weld them together.

### 3 Literature Review

The process of parking the car in the parking space may seem an easy process, but some people face a lot of problems when doing this process, many people and companies have made various designs to solve this problem, one of the designs was based on the idea that the car stops parallel to the parked cars when it finds an empty car park. Then the wheels of the car deviate to be in the direction of the empty car park because of the ability for that robot to steer the wheels about  $95^\circ$  on one of the direction [1], and to solve a problem that some people not committing to park their cars in on one park space and park their car in a random way inside the garage the start-up, Stanley Robotics, introduce their parking robot which is used in airport parking to park the cars of the passenger's[2]. As we see there are a lot of related projects centered around the same principle and work to achieve the same result with different tools or different designs. our robot have achieved the same result of parking cars in the correct park space but its design is completely different from other related projects and cheaper because we depend on wheel speed to change the direction of the robot nested of mechanical parts that cost a lot of money.

## 4 Design and Implementation

When parking or reversing cars, collisions are common. so, in our project, we aim to organize parking cars using a parking robot. The car can be left at the garage door and then the robot can be ordered using a card. our robot work by carrying the car and putting it in the park in a tidy way. we built our robot in two stages starting with collecting the hardware parts and controlling them using Arduino then we use image processing to make the robot decide where to park.

### 4.1 the garage design

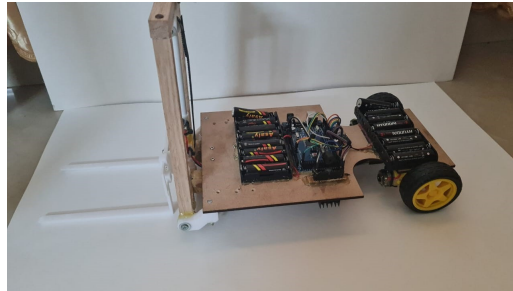


Figure 4.1.1: Robot design

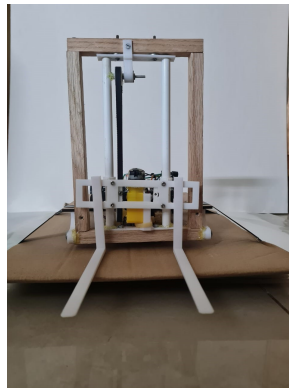


Figure 4.1.2: Robot crane

We designed our robot-like forklift. the body is from wood and PLA (3D Printer material) connected with the wheels and holds all the hardware parts as shown in Figure 4.1.2. The body connected with the crane which is part from PLA (3D Printer material) as shown in Figure 4.1.2. this crane lifts the car with its arms and the wheels move it to the particulate place.

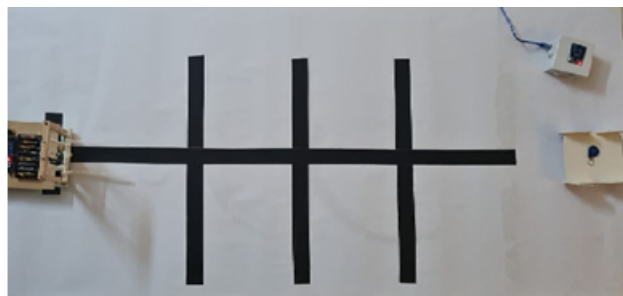


Figure 4.1.3: garage design



Figure 4.1.4: garage wall

Also, we designed our garages so the robot can move easily starting from the entry where the car wait to the robot's place. the main road crossed by a black line where the robot can follow it to reach the car and back to his place. there are other black lines crosse the car's place in each garage and in front of them so the robot can back. The robot follows lines and puts the car or takes it back. this is shown in Figure 4.1.3. also, every garage wall has a picture so the robot when seeing this picture will know if there's a car or not. this picture is shown in Figure 4.1.4. As the robot can see the shape, he could see the garage is empty. If the car is there it will cover the shape and the robot will know that the garage is full. Also, the picture includes another

shape this shape is different for every garage so the robot stores where he parked the car so he could back and get it.

## 4.2 Hardware and Software Specifications

### 4.2.1 Hardware Components

#### Hardware component to order the robot:

For this part, we have used an Arduino Uno, HC-05 Bluetooth Module, RC522 RFID Module, and Buzzer.

- **Arduino Uno**



Figure 4.2.1: Arduino Uno

Arduino Uno is a microcontroller based on the ATmega328P; it is designed for experimentation. It has 14 digital input/output pins, a USB connection, a power jack, and a reset button. Here, we use it to make a system to order the robot. It is connected with the RFID module, the Bluetooth module, and the buzzer. It is shown in Figure 4.2.1.

- **HC-05 Bluetooth Module**

The HC-05 module is a Bluetooth module designed for wireless sequential communication configuration. It is able to utilize either the Master or Slave interface, making it a superior choice for connecting with a phone wirelessly. It has a working voltage of 4 to 6 volts and a 30-mA operational current. It has a range of 100 meters. Here, we use it to send a signal to the robot to tell it that someone orders it.

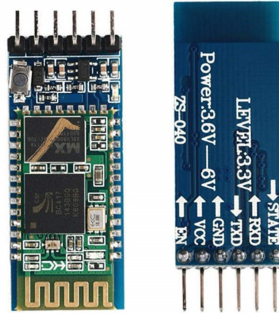


Figure 4.2.2: HC-05 Bluetooth Module

## ● RC522 RFID Module

The RC522 is a 13.56MHz RFID module based on the NXP semiconductors MFRC522 controller. The module supports I2C, SPI, and UART, and comes with an RFID card and a key fob by default. It's often used in attendance systems and other applications that identify people or objects. here, we use it order the robot for parking or to bring the car back.

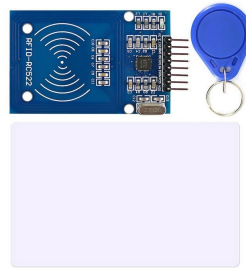


Figure 4.2.3: RC522 RFID Module

## ● Buzzer

A buzzer or beeper is an audio signaling device.it is powered through DC voltage and used in timers or alarm devices. here we connected it with

Arduino, so it works when the card passes through the RFID to confirm that the card is read, and the robot is coming.



Figure 4.2.4: Buzzer

## Hardware component for the robot:

For this part, we have used an Arduino Mega, HC-05 Bluetooth Module, 2 L298N Dual H Bridge, 3 DC Motor, 4 IR Sensor and Buzzer.

- **Arduino Mega**

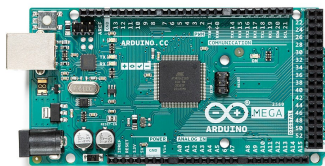


Figure 4.2.5: Arduino Mega

Arduino Mega is a microcontroller based on the ATmega328P; it is designed for experimentation. it has 54 digital input/output pins 16 analog inputs, 4 UARTs (hardware serial ports), a USB connection, a power jack, and a reset button. Here, we used it to program our robot. it connected with

the Bluetooth module, the buzzer, the 2 H-Bridge, and the IR sensors. Also, it is connected to the raspberry pi and both of them communicate through the serial connection.

- **HC-05 Bluetooth Module**

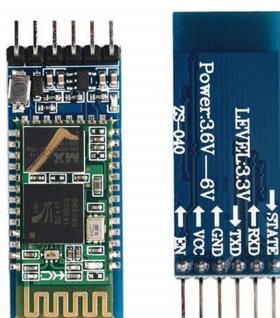


Figure 4.2.6: HC-05 Bluetooth Module

As was mentioned before this module is a Bluetooth module for wireless communication, and here we use it to get the sign from the first module in the entry of the garage. so, the robot when getting the sign move to the entry to bring the car or take it back if it already exists.

- **L298N Dual H Bridge**

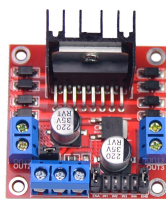


Figure 4.2.7: L298N Dual H Bridge

An H-Bridge is a circuit controlled by pulse width modulation that can drive a current in either polarity (PWM). It is used to regulate the speed and direction of motors. We used it to control the robot's movement by controlling the dc motors which attached to the wheels. also, we used another one for the Forklift hand so it can lift the car.

- **DC motor**

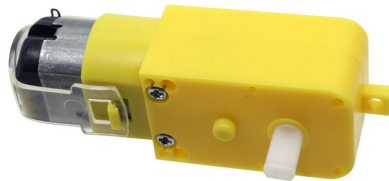


Figure 4.2.8: DC motor

A DC motor is a type of electric motor that converts electrical energy into mechanical energy using direct current. It works with 12 volts. We attached 2 of them with 2 wheels of the robot. And connecting them with the H-Bridge.

- **IR Sensor**



Figure 4.2.9: IR

Infrared (IR) sensors are electrical devices that detect and quantify infrared radiation in their surroundings. When infrared photons strike a white surface, they are reflected back and captured by photodiodes, causing voltage changes. However, when IR light strikes a black surface, the light is absorbed by the black surface and no rays are reflected back, leaving the photo diode without light or rays. here we used this sensor to detect the black line so the robot will follow it.

## • Raspberry Pi4 with Camera

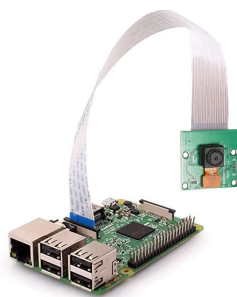


Figure 4.2.10: Raspberry Pi4 with Camera

The Raspberry Pi is a low-cost, credit-card-sized computer that plugs into a computer monitor or TV. it can connect to a camera. It is a capable little device that supports programs in languages like Scratch and Python. Also, it uses to program robots. as we need a camera to let the robot see the garage, we used it. we downloaded its operating system. then we downloaded visual studio code to program in python and the needed libraries for image processing. also, we connected it with Arduino Mega. they communicate through the serial connection, so the raspberry pi sends data for what the camera sees, and the Arduino moves the robot.

### 4.2.2 Software Components

To make our robot work first we connected the robot component and its body. Also, we connected hardware components to order the robot in the entry of the garage. Then we programmed both using Arduino ide. Last

thing we used raspberry pi and python to make the robot see if the garage is empty or not.

## Arduino IDE

The Arduino Integrated Development Environment (IDE) is a piece of software that helps with code editing, compilation, and debugging. The Arduino IDE runs on the Java Platform. It's for writing and uploading code to Arduino-compatible boards. We write the code using it then

## Raspberry Pi4 Operating System and Visual Studio code

Raspberry Pi OS (Raspbian) is a Debian-based operating system for Raspberry Pi. as we need a camera to let the robot see the garage, we downloaded the OS and used it to enable the camera. also, this OS supports programming languages like Scratch and Python. so, we downloaded visual studio code to program our robot using python and the needed libraries for image processing.

### 4.3 Arduino Code

We use Arduino IDE to write two codes:

Scanner code:

First, we include the library for RC522 RFID which is called `<MFRC522.h>` to read the card number and then send the value of the card using Bluetooth which uses serial communication, we include `<SoftwareSerial.h>` library to use software serial nested of using TX, RX pins on the Arduino. For each card number, we send a character nested of sending the whole number.

```

String card = "";
for (byte i = 0; i < mfrc522.uid.size; i++)
{
  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
  Serial.print(mfrc522.uid.uidByte[i], HEX);
  card.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
  card.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();
Serial.print("****");
card.toUpperCase();
digitalWrite(BUZZER, HIGH );
delay(50);
digitalWrite(BUZZER, LOW );
delay(1000);

if (card.substring(1) == "8A C0 B4 A3")
{
  mySerial.write('A');
}

```

Figure 4.3.1: RFID Bluetooth code

Robot code : For the second part we use Arduino mega, first we read the value of IR sensor and according to its value we made the robot move from the end of the garage to its entrance and made it stop on every parking space to check the parking space number and if its empty or not. To check if parking space is empty or not and to check its number we use raspberry pi camera that will receive 2 messages from Arduino when the robot reach every parking space ,the raspberry will send two signal using USB cable for serial communication .the first signal will tell the Arduino if the parking empty or not .

```

Serial.println("Waiting for RapsberryPI feedback");
while (true) {
  if (Serial.available() > 0) {
    rapsberryPI_Value = Serial.read(); // x or X to endecate an empty or not parking space

    Serial.println("What is this park ID? (0, 1, or 2)");
    while (Serial.available() == 0); // wait for RaspberryPi input
    park_id = Serial.parseInt(); // starting from 0 to 2

    if (rapsberryPI_Value == 'x') { // empty parking space
      Serial.println("Found empty parking space");
      found_parking_space = true;
      break;
    }
    else if (rapsberryPI_Value == 'X') {
      Serial.println("Moving to the next parking space");
      break;
    }
  }
}

```

Figure 4.3.3: check empty parking space

the second signal will tell the Arduino the number of the parking.

```

-----,-----,
while (true) {
  Serial.println("What is this park ID? (0, 1, or 2)");
  while (Serial.available() == 0); // wait for RaspberryPi input
  short park_id = Serial.parseInt(); // starting from 0 to 2
  if (car_park_id == park_id) {
    break;
  }
  else if (car_park_id < park_id) {
    moveForwardToParkSpace(0);
    delay(500);
  }
  else if (car_park_id > park_id) {
    moveBackwardToParkSpace(0);
    delay(500);
  }
}
}

```

Figure 4.3.2: check parking number code

## 4.4 Image processing Code

We write the image processing code on visual studio code in raspberry pi. this code opens the Raspberry Pi camera and takes pictures to let the robot decide where he can park the car. the detection of the image will be in two parts. When the robot stopped in front of one of the garages, this code will use the Open CV library to read the image. using `cv2.cvtColor()` we will change the picture colors' to HSV format so we can detect color. we will take the HSV image and determine the color range that we want the robot to see. this new image will show anything with this color and the other colors will convert to black using this function (`cv2.bitwise_and(img, img, mask=mask)`). after that, we used some filters to make the shapes clear for the robot in this color. using `cv2.findContours` we contoured all the shapes in the image. then detect which shape area is less than the specific area (this number depends on how much the camera is far from the wall). then in this shape, we counted its heads using `approxPolyDP()`. if we find the shape with 6 heads and the specific area, the robot will know the garage is empty. if not, then the car is set in front of the shape, and the garage is full. The second part, the first image taken before all previous changes, will enter other filters to detect the id of the garage. here we detected other shapes in

the same image so every garage has a different shape. the data stored from the detection of the two shapes will send in a serial connection to Aurdino.

## 4.5 Circuit Layout

to control the movement of the robot we use Arduino mega to program different parts first we use four Infrared (IR) sensors and connect them with pins 8,9,10,11 on the Arduino all these IRs used to read the black color to make sure that we are on the correct bath, we use two H-Bridge and connect them with a pulse with modulation pins on the Arduino, one of them is used for to control the direction and the speed of the two dc motors that make the movement of the robot, and the other H-Bridge used for the forklift dc motor.to make a connection between the Arduino Mega on the robot and Arduino Uno on the door we use Bluetooth (HC-05), we connect RX, TX from it with A8, A9 from Arduino mega respectively .to power the Arduino we use 12 volts and another 12 volts for the Bridge. and finally, we connect raspberry pi with Arduino using a USB cable.

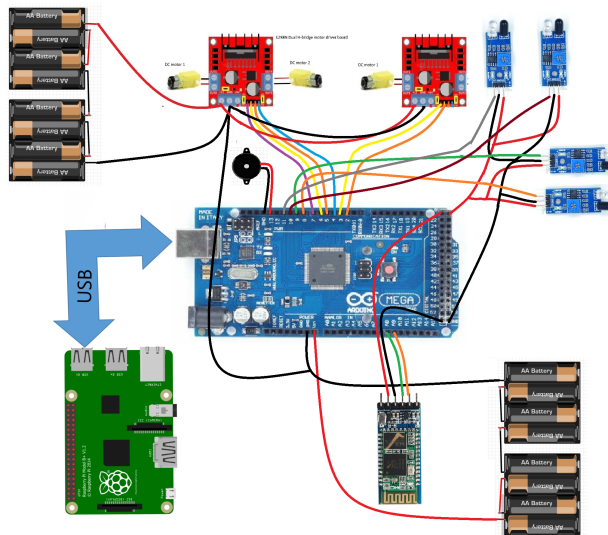


Figure 4.5.1: Robot implementation

To read the user card we use RC522 RFID and send the value of the card by Bluetooth (HC-05) which is connected to pins 8, 7.

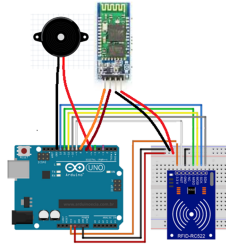


Figure 4.5.2: scanner implementation

## 5 Results and Analysis

We compiled the source code for every step and make sure that everything is working well before we move to the next step, that help us a lot to determine where the errors come from.

In the final stage, our project fulfilled the requirements related to it like reading the user card by the scanner (RFID), the signal was sent to the robot via Bluetooth to take the car, and the user car was parked in an empty parking space finally, the robot brings the user car when he scans his card for the second time.

Sometimes we got some problems with our robot: the forklift sometimes failed to lift up the car properly and the raspberry pi doesn't give us a very accurate result either.

step1:the robot go to the entry of the garag to take the car



Figure 5.1: the robot move to bring the car

step2:the robot carries the car

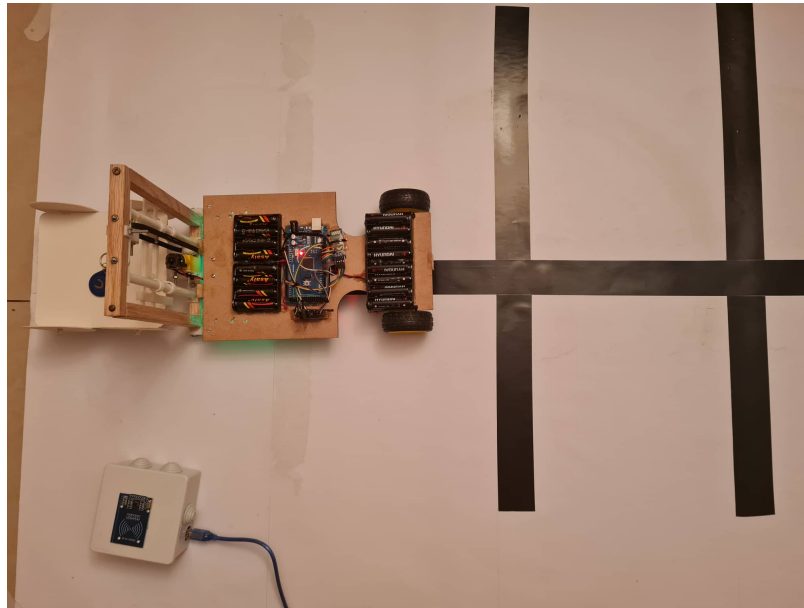


Figure 5.2: forklift lifts the car from the ground

step3:after reach to the first parking space the robot check if there is a car in the parking space if not the robot move as shown to directing to the car to an empty parking space.

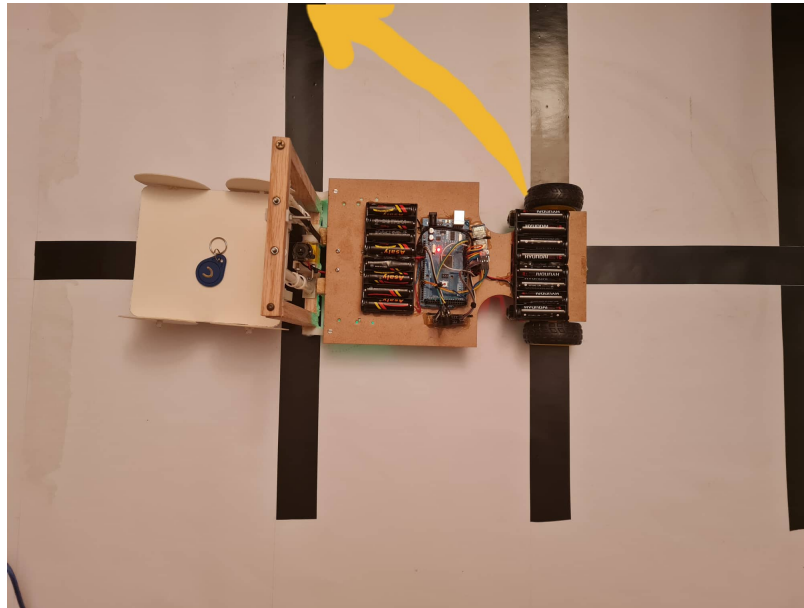


Figure 5.3: park the car

the same steps done on reverse order to bring the user car back to him.

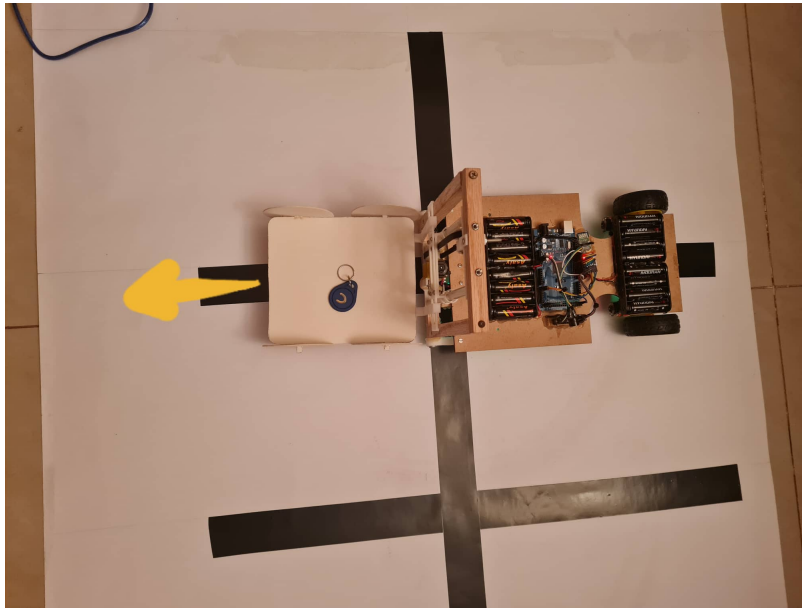


Figure 5.4: park the car

## 6 Conclusions and Recommendation

in our project, we try to make a robot that can be used to park cars in the garage by using a card with a different number for each user and applying the users to retrieve their car by scanning their card at the entry of the garage. We can improve our project by using a mobile application nested of a card to make it easier for the user and minimize the size of the robot to reduce the robot bath and park more cars in the garage. we can also contact between two Arduino by wifi nested of Bluetooth to increase the range of connection.

## Bibliography

- [1] Alexander Schaub, Juan Carlos Ramirez de la Cruz, Darius Burschka, Autonomous Parking using a Highly Maneuverable Robotic Vehicle, 47,3, 2640-2645.
- [2] LYON HAS THE WORLD'S FIRST ROBOTIC PARKING LOT. (2015). Retrieved May 15, 2022, from <https://1cars.org/16090-in-lyon-the-first-parking-lot-with-robots-was-opened.html>