Mechatronics System Design 1

Final Project

The Responsive Chair

Course offered by: Dr. Maher El Rafei

Team Members: Joya Maria Saade, Vanessa Hanna, Christopher Zein

Table of Contents

[Introduction: 5](#_Toc152775613)

[The Problem Statement 5](#_Toc152775614)

[The System’s Requirements 6](#_Toc152775615)

[Function diagram 7](#_Toc152775616)

[Error and trial 12](#_Toc152775617)

[Teamwork and Tasks Details 13](#_Toc152775618)

[Mechanical implementation 13](#_Toc152775619)

[Chassis Model 13](#_Toc152775620)

[Motors and Drivers 15](#_Toc152775621)

[Motors Details 16](#_Toc152775622)

[The electrical Components 18](#_Toc152775623)

[Pins and connections 22](#_Toc152775624)

[Actual implementation: 24](#_Toc152775625)

[Current Implementation Summary 28](#_Toc152775626)

[Old vs. New 29](#_Toc152775627)

[Conclusion 31](#_Toc152775628)

[References 32](#_Toc152775629)

Table of Figures

[Figure 1: Draft flow graph 11](https://lauedu74602-my.sharepoint.com/personal/vanessa_hanna01_lau_edu/Documents/Desktop/Fall%202023/Mechatronics%20Systen%20Design%201/Assigments/Final%20project/report.docx#_Toc152775815)

[Figure 2: Model Assembly 14](#_Toc152775816)

[Figure 3: Section view of the model 15](#_Toc152775817)

[Figure 4: Motors connections 17](#_Toc152775818)

[Figure 5: ESP8266 Connection diagram 20](#_Toc152775819)

[Figure 6: Pinning of the ESP8266 21](#_Toc152775820)

[Figure 7: Redesigned prototype 24](#_Toc152775821)

[Figure 8: Model exploded view 25](#_Toc152775822)

[Figure 9: Section view of the new model 26](#_Toc152775823)

[Figure 10: Flow graph of the new implementation 30](https://lauedu74602-my.sharepoint.com/personal/vanessa_hanna01_lau_edu/Documents/Desktop/Fall%202023/Mechatronics%20Systen%20Design%201/Assigments/Final%20project/report.docx#_Toc152775824)

[Figure 11: Hardware Connections 31](#_Toc152775825)

Table of Tables

[Table 1: Draft 1 of the detailed functions 8](#_Toc152775640)

[Table 2:Possible Solution 10](#_Toc152775641)

[Table 3: Chosen solutions 12](#_Toc152775642)

[Table 4: logic table of the motor’s direction bits 17](#_Toc152775643)

# Introduction:

The focus of this course was to fuse data coming from multiple sensors to predict movements, rotation, or behavior of a specific model. Through this project, all the acquired knowledge during this course was implemented and further developed to meet the prototype’s requirements.

## The Problem Statement

This project was inspired by the new Nissan invention in their meeting offices. They installed ‘parking’ chairs in their offices to move around and fit perfectly under the desk on a double clap of hands.



However, the parking chair is designed as one part and not a device that is added to the chair and can control it. Therefore, we modeled a device that can fit on every office chair and by installing it properly, it would function correctly. The implementation of this device requires a larger budget and deeper knowledge of image processing and control. While Nissan used computer vision through cameras mounted in their offices, we worked our way into multiple solutions for this model.

## The System’s Requirements

Before starting the design of the project, several requirements and restrictions were kept in mind. First, the system should be light, in terms of weight, design, and components to ensure a smooth workflow without overcomplicating ideas. The movement should be relatively fast and effortless to allow the chair to get quickly to its target while avoiding both moving and steady obstacles, thus the need to a precise navigation and obstacle detecting system. The core of our design is to initiate the movement of the chair remotely through a clap, to ensure a user friendly and innovative design. Moving to the primary restrictions on the system, we must note that the chair should remain stationary if someone is seated even if a clap is recorded and the motors must be lifted with a platform actuated by a stepper motor in order to protect them from damage when the chair is manually moved. Overall, the system is expected to showcase an intelligent and responsive design while abiding by the requirements and restrictions.

# Function diagram

After stating the basic requirements, the technical constraints and specifications, all the needed functions were summarized and stated in the first draft of the project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Function | Sensor | Mounting | Algorithm | Module | Pins |
| Geared Motors and Encoders |  | robot | Sending control signals to the motor driver module and receive encoder feedback | Motor driver with encoder input:  tb6612fng. | Connect the geared motors to the motor driver outputs. Connect the encoder outputs to the encoder input pins on the motor driver module. For motor driver: 7 input pins (AIN1, AIN2, BIN1, BIN2, PWMA, PWMB, STBY) and 6 output pins (OUT1A, OUT2A, OUT1B, OUT2B, GND) |
| Lock Mechanism | Light sensor (LDR) | Seat of the chair | Senses the presence of a person through variation of light intensity and sends analog signal to the controllers |  | Connected to the controller through Vcc of the Arduino, analog input and ground |
| Clap Detection | Sound sensor | On the robot/ on the bottom of the chair | Detects clap to activate/ deactivate the system |  | 5pins (Vcc, ground, analog input, digital input, digital input ground) |
| Obstacle Detection | Ultrasound sensor (HC-SR04) | On the robot | Obstacle detection through iterations of distance calculation |  | 4 pins: Vcc, GND, Trig, Echo (Digital pins) |
| Localization | BM7161-00-1 BLE beacon | The beacons will be mounted on three different locations in the room and the receiver on the robot | Radio communications and triangulation algorithms, RSSI computation and fingerprint algorithm to localize the robot | HM10/ HM11 BLE modules | 6 pins but 4 are used with the Arduino: Vcc, GND, TXD to a digital pin and RXD to another digital pin |
| Controllers | Arduino Mega 2560  Arduino Uno | Arduino Mega is connected to the robot and receives all the data while the Arduino uno only receives wireless data from the Arduino Mega | Using radio communication, the two Arduino controllers will communicate wirelessly and the serially connected Arduino to the laptop will transmit data to python for computations. | NRF24L01 | 8 pins: GND, Vcc, Ce and CSN are pwm pins, SCK, MOSI, MISO are digital pins |
| Path optimization |  | Python code | Path Planning algorithm |  |  |

Table 1: Draft 1 of the detailed functions

Moving on to a quick explanation of the above functions, one can start with the geared motors that drive the whole mechanism. The educated choice of using geared motors was based on the need of a high starting torque and low speed. The weight of the chair requires a relatively large torque for four small motors to move. The gear motors and the gearing mechanism used will be further explained in the motors’ section. To control the four motors, one can use the H-bridge module for more precise bidirectional control.

Accommodating for different test cases on the chair, the main constraint on the motors is the weight. Therefore, the motors should be completely off and at a certain elevation from the floor when an external weight is added on the chair such as a human sitting, or when the chair is moved manually to prevent the breakage of the shafts. The lock mechanism was developed to accommodate for these constraints.

Since the chair is sound activated, a sound detector must be mounted on the assembly.

The chair is moving in an unfamiliar and dynamic location. Therefore, obstacle detection and avoidance are needed for a smooth operation. However, the chair for now is still not static since the target location and its relative location are still unknown. Thus, a localization technique was suggested using low-energy Bluetooth signals. The method was highly studied and researched upon, and all the references will be mentioned in the appendix. Low energy beacons are located in multiple locations and the chair will detect the energy of the Bluetooth signal and compute it into relative coordinated that can approximate the relative location of the moving robot.

The choice of the controller was discussed tremendously, and the Arduino MEGA and UNO versions were agreed upon.

The logical sequence of the mechanism can be introduced briefly from the previous draft. The weight and movement sensor checks whether someone is sitting on the chair, or the chair is being move manually, if both conditions are false, and a clap is detected, the chair should start evaluating the signals strength coming from the BLE beacons and compute its relative location that is constantly updated to control the accuracy of the model. While moving around to reach its target location that was initialized relative to the room, the robot might encounter an obstacle, through the obstacle detection algorithm and sensors, and while fusing the data coming from the beacons, the robot should safely return under the desk.

The sequency will be explained in further detail later in the report. This was a small introduction to link the ideas together and start visualizing the product.

Multiple solutions can help in achieving the required functionalities from the robot. These solutions will be summarized in the below table.

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Solution 1 | Solution 2 | Solution 3 |
| Lifting Mechanism | Electrical magnets | Gear and axis | Bolt and stepper motor |
| Wheels | Mechanum wheels | Omni-wheels | Regular wheels |
| Motors and Encoders | DC motors with absolute encoders | DC motors with relative encoders |  |
| Plastic Body | Rover 5 | Predesigned chassis with built in wheels | 3D printed chassis |
| Lock Mechanism | Weight sensor | Thermal sensor | Photoelectric sensor |
| Clap Detection | Sound sensor |  |  |
| Obstacle Detection | Ultrasound sensor | Lidar | Camera (computer vision) |
| Localization | Camera (computer vision) | BLE beacons | Ultrasonic sensor, IMU |
| Controllers | Raspberry PI | One Arduino controller | Two Arduino controllers |
| Tracking | Mouse sensor | Camera (computer vision) |  |
| Path optimization | Path planning algorithm (A\*) | Dijkstra’s Algorithm |  |

Table 2:Possible Solution

A diagram of a diagram

Description automatically generatedThe logical sequence discussed earlier can be observed in the following flow graph in which the initial and theoretical choice of sensors was made to meet the requirements.

Figure 1: Draft flow graph

Due to the large computations and complexity of the expected algorithms, two Arduino boards were used: the Arduino Mega will be mounted on the assembly, and through Bluetooth, it sends the data received from the actuators and the sensors and sends it to the Arduino UNO that is connected serially to the PC. The data from the Arduino UNO is sent to a python file under JSON extension and the complex algorithms are done on python and the retrieved data that dictates the movement of the actuators is send back from the PC, serially to the UNO and then sent back through Bluetooth to the Arduino Mega that is controlling the actuators.

The sound sensor is used to capture the clapping sound through a tunable sensitivity that dictates the frequency of the clap. The light sensor detects if there is someone sitting, the ultrasound sensors are used for obstacle detection, the Bluetooth receiver is used to receive the data from the beacons and feed them to the Mega.

## Error and trial

At the start of the journey the following solutions were the best fit in terms of:

* Complexity
* Implementation time
* Availability
* Costs

The solutions available in this table were highly discussed with supervisors and Drs, yet some of them were changed later in the design due to uncontrolled factors.

|  |  |
| --- | --- |
| Function | Chosen Solution |
| Lifting Mechanism | Bolt and stepper |
| Wheels | Regular wheels |
| Motors and Encoders | DC motors with encoders |
| Plastic Body | 3D printed chassis |
| Lock Mechanism | Photoelectric sensor |
| Clap Detection | Sound sensor |
| Obstacle Detection | Ultrasound sensor |
| Localization | BLE beacons |
| Controllers | Two Arduino controllers |

Table 3: Chosen solutions

# Teamwork and Tasks Details

The tasks were split evenly and according to each member’s expertise. The mechatronics design model provided in Chapter 1 of the course was used as the main component of the teamwork. The mechanical, electrical, and coding were the main three categories. However, each member worked evenly on each of the three categories.

## Mechanical implementation

The mechanical category included the stepper motor that would lift the wheels and the motors, differential gearing used to move and rotate, and the choice of drivers, motors, and the design of the chassis. The stepper motor will be mounted on the inner ceiling of the chassis and connected to the movable platform through a designed screw that fits perfectly.

### Chassis Model

The dimensions of the chassis were taken accurately with a digital caliper. The chair was virtually designed for a more accurate model using SolidWorks. The model of the product accounted for the components that should fit inside, the movable chassis, and all the sensors that need open spaces to send signals. The product was perfectly shaped for a specific chair, yet it can be easily modified to fit any other chair.

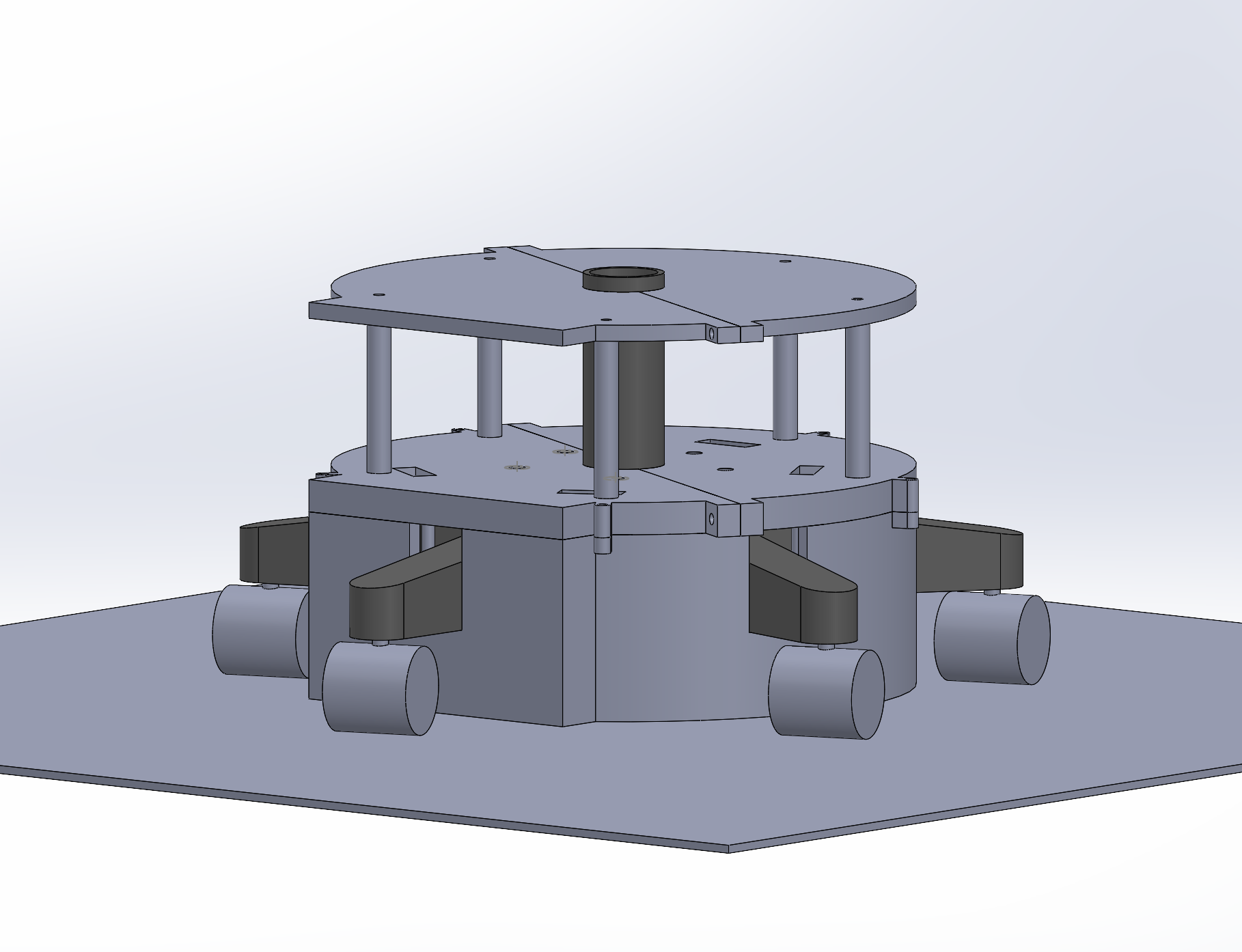


Figure 2: Model Assembly

From the figure above, the model was designed carefully and included the main features of the chair with the required dimensions. The unboxed space fits all the electronics that send signals and need open space to communicate. The closed bottom contains the motors, the stepper, the wheels and the drivers.

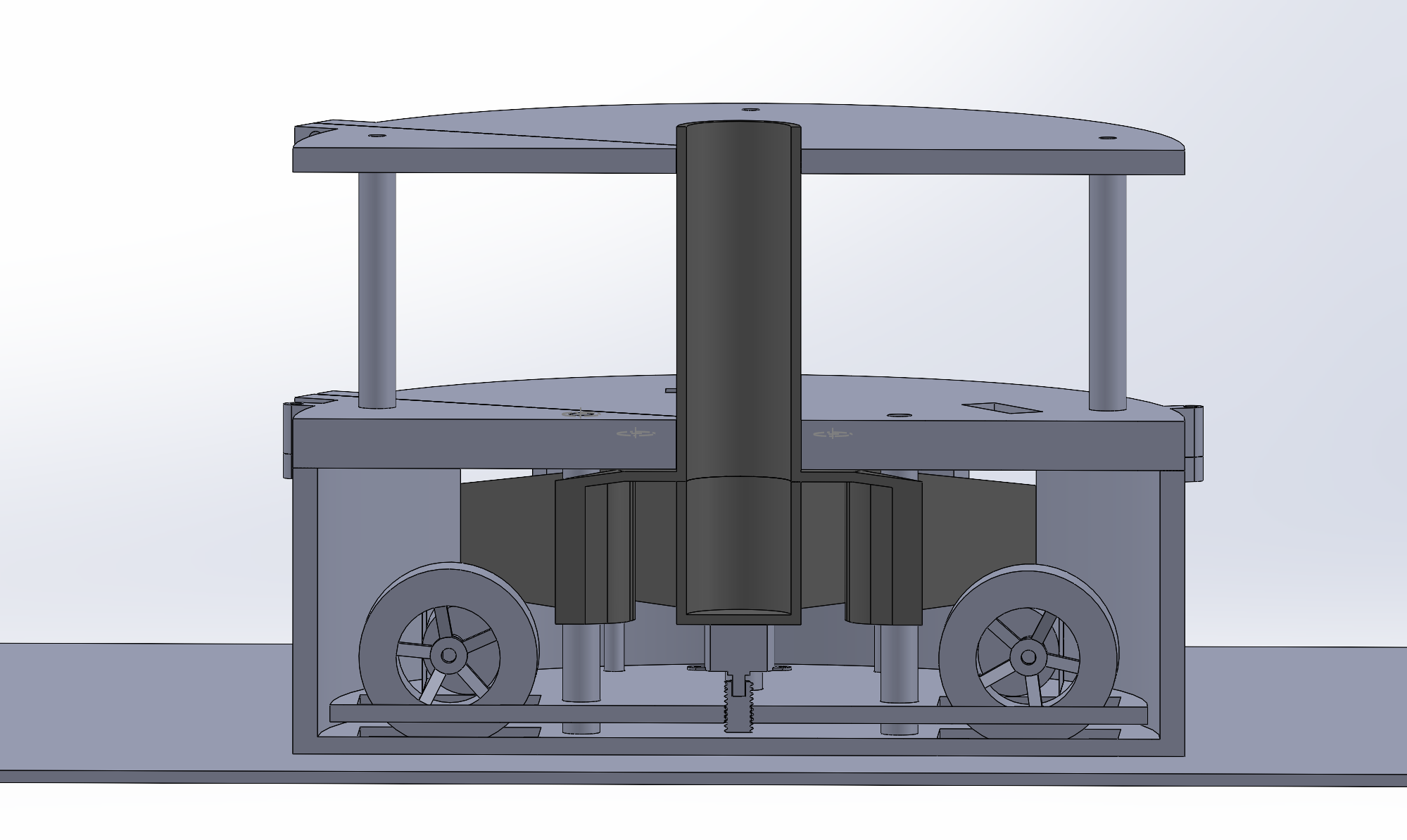




Figure 3: Section view of the model

The circled component is the screw that was designed to move the platform pointed at by the arrow. The screw is connected to the platform and to the stepper motor that is fixed to the bottom of the chair. The platform is prevented from rotating through holes that fit exactly on printed columns fixed to the bottom of the model.

### Motors and Drivers

For the geared motors, the torque was calculated based on the weight of the chair in hand and the geometry of the wheels that are holding the chair. It is important to note that the overall weight was not held by the motors and the wheels of the added component, yet it remained divided on the wheels of the chair itself. The motors and the wheels of the external device were only a movement assist to the assembly. However, a relatively high torque was needed. Since high speed was not needed for the project, geared motors were the perfect fit for a smooth operation. The chosen geared motors are four JGB37-520.

The datasheet of the motors can be found by clicking this link: <https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/108990006_Web.pdf>

### Motors Details

The chair weighed 4Kg. It had 5 legs with ends connected to rotating wheels. So, the torque applied can be approximated by the weight of the chair multiplied by the horizontal distance between the center of the chair and the center of the added wheel. Through detailed calculations the torque needed was supplied by the motors’ JGP37-520 family.

Four motors of this family were used to drive the robot. Each two motors lying on the same side will be connected to one side of one dual H-Bridge driver L298N for differential gearing. Differential gearing is used to move the component smoothly yet restrain the chair from rotating a lot since the front view of it should be facing the desk at the return location.

Each motor had 6 wires: Vcc supplied by 5V from the Arduino, M1, M2 for motor control connected to OUT1 and OUT2 on the motor driver, GND connected to the common GND, and two remaining wires for the encoders signal. The 12V of the driver is connected to Vin of the Arduino that is supplied by a rechargeable 12V DC battery, the 5V of the driver are connected to the 5V of the Arduino, the Enable A is connected to a PWM signal from the Arduino, IN1 and IN2 are connected to digital inputs of the Arduino to turn it off, on, clockwise and counterclockwise.

The connection diagram of the motors and the driver used will be shown in detail in the next figure.

A diagram of a motor

Description automatically generated

Figure 4: Motors connections

The IN pins of the driver are connected to digital pins of the Arduino for directional control. The logic diagram of those bit is interpreted in the below table:

Table 4: logic table of the motor’s direction bits

|  |  |  |
| --- | --- | --- |
| Input 1 | Input 2 | Output |
| Low | Low | Motor Off |
| Low | High | Counterclockwise |
| High | Low | Clockwise |
| High | High | Motor Off |

The direction of the motors is specified by those bits while the speed of each motor is controlled by the PWM signal coming from the Arduino through the Enable bits of the driver. The speed is specified by the pulse width that can be easily controlled throughout the code.

The stepper motor has a minimal impact on the overall design. However, the importance of its function is maintaining the motors’ shaft intact while moving the chair manually. The stepper motor is connected through its driver to the Arduino. The stepper is used to lift the movable chassis and lift the motors and the wheels. A modeled screw was 3D printed and fixed on the stepper. The stepper turns the screw down to move the motors and wheels down and pin them to the floor to make sure that the wheels do not slip on the floor. Once the chair reaches the destination the stepper motor turns up and moves the wheels and the motors up. However, the motors are heavy and the stepper motor on its own will not be able to move it up. Therefore, the moveable chassis will be held to the top of the model through springs.

### The electrical Components

The main electrical view of the project relies on the sensors that drive the behavior of the project. The ultrasonic sensors are mounted at three locations on the robot to track the maximum obstacles around it. The ultrasonic sensors have 5 pins, Vcc, GND, Echo, Trig, and enable. The Vcc and GND are connected to the common 5V and the GND of the Arduino. The Echo is a signal sent from the Arduino. The Echo is connected to a digital pin. Once the digital pin sends a high signal, the ultrasound sensor sends an ultrasonic signal to measure the distance of the obstacle. The trigger pin is connected to another digital pin. Once the signal is reflected on the obstacle, the trigger pin receives a one. The time needed to receive the signal back can be multiplied by the sound speed to get the distance from the robot and the obstacle.

Another sensor used is the LDR. The LDR has three pins. One Vcc, GND, and an analog pin connected to the Arduino. This analog pin connected to the Arduino receives the resistance of the LDR. The resistance of the LDR is proportional to the light detected by the LDR. If someone is sitting on the chair, the LDR’s face is covered, and the resistance is calibrated through testing to check the threshold needed. The analog signal received from the LDR is compared to the threshold. The condition of the LDR has the highest priority. Once the resistance received passes the value, the chair can move, and all the other conditions will be checked.

Once the LDR condition is true, the sound sensor is turned on. The sound sensor is responsible for detecting the clapping sound. This sensor has 3 used pins. The Vcc, GND, and a digital pin. This digital pin is connected to an input pin to the Arduino. The sound sensor has a potentiometer that specifies the sensitivity of the sensor to the clapping sensor. Through testing, the resistance was specified, and the sound sensor detects the calp. Once it is detected, the digital pin receives a 1. If this condition is met, the ultrasound sensors are turned on, the Bluetooth sensor starts calculating the DB value of the low energy Bluetooth signal from the beacons in the room and fusing algorithm.

For localization, the first solution was Bluetooth. Low energy beacons are set in the room at different locations and an HM10 Bluetooth sensor connected to the Arduino. Each beacon has a specific MAC address that is sent in the signal received by the HM10. The sequence of bits received by the HM10 is parsed into different useful partitions. First, the MAC address of the beacon is extracted, then RSSI (signal strength) is parsed and converted into an integer value. This value is then used in a trilateration algorithm that converts the signal into coordinates relative to the starting location. Then, the robot moves towards this destination location and keeps computing the RSSI value throughout the movement. Three beacons were going to be used. Now, dealing with the HM10, the processor of this chap deals with AT commands. Each AT command is used to extract the needed value from the received signal. However, through trial and error, the HM10 bought in Lebanon are cloned chips and their firmware needed an update. After buying the first HM10, the chip was able to send information but was not able to receive. Through deep research, the same problem was highly encountered in the robotics world. One possible solution to change the HM10 chip to remove the possibility of buying a damaged chip. Another possible solution was updating the firmware of the chip. The firmware was updated twice for both chips. External wires were soldered for direct connection to the memory and controller bits of the HM10. The firmware update steps followed are found on this link: <https://circuitdigest.com/microcontroller-projects/how-to-flash-the-firmware-on-cloned-hm-10-ble-module-using-arduino-uno>

Even after firmware update, the two chips did not work. They remained unable to receive values from the beacons. After asking around, and checking previous projects based on the HM10, the HM10 bought was a clone and nothing could be done to retrieve the data from the beacons. This was an uncontrolled problem, and the only alternative was using a Wi-Fi sensor. The Wi-Fi sensor used was the ESP8622. This sensor is connected to the room’s Wi-Fi and can calculate the strength of the Wi-Fi signal that will be used for localization. The Wi-Fi sensor has 8 pins. It is a chip that operates on 3.3 V. The manipulation of this chip was hard yet attainable since it is also bought as a clone. Voltage division for all the pins was used to receive a 3.3V instead of 5V since the Arduino sends through its output and input pins 5V. The pins diagram of the chip is demonstrated in the next figure.

A diagram of a circuit board

Description automatically generated

Figure 5: ESP8266 Connection diagram

Figure 3 shows the connections of the ESP8266 with the Arduino UNO. The transmitter pin TX of the Arduino is connected to the receiving pin of the ESP8266 through a voltage division. The RX pin of the Arduino is connected to the TX pin of the ESP8266. And the enable pin is connected to the 3.3 V through the 10Kohm resistor.

A diagram of a circuit board

Description automatically generated

Figure 6: Pinning of the ESP8266

The Wi-fi module also communicates using AT commands. The Virtuino library was used to communicate with this module since it is a clone.

The Tx and Rx communication is set through the SoftwareSerial library in C++.

SoftwareSerial espSerial =  SoftwareSerial(2,3); 2 is the Rx pin and 3 is the Tx pin

The methos used for Wi-Fi connection are explained through the comments of the code and will be explained in the presentation. All the AT commands can be found on this page: <https://electronics-fun.com/esp8266-at-commands/>

The two Arduino boards should communicate through a radio channel created by a NRF modules. One should be connected to the Arduino Mega and the other to the Arduino Uno. The code was successfully implemented. However, the amount of data that needed to be sent was too high and at a very high rate. Therefore, this option was removed for simplicity and to minimize error.

The code of the NRF24 is available as Node A and Node B in the files.

However, two Arduinos were still used due to the high computations and calculations done. The Arduino Uno was used for the Wi-fi tracking since the sensor was not compatible with the Arduino Mega. The Wi-fi strength was found on the Arduino Uno and sent through the SPA and SCL pins of both. The SPA pin of the Uno (A4) was connected to the Arduino Mega and the SCL pin of the Uno (A5) was connected to the SCL pin of the Mega. This value was transferred as a byte to the Mega and used to track the movement of the robot throughout the room. The Mega was connected to all the other sensors: LDR, three ultrasound sensors, Sound sensor and the motors.

### Pins and connections

The pins used are summarized in the below table:

|  |  |  |
| --- | --- | --- |
| Component Name | Component pin | Arduino Pin |
| LDR | OUT | A0 Mega |
| Sound Sensor | OUT | DI 46 Mega |
| Ultrasound sensor 1 | Trig | DO 52 Mega |
| Echo | DI 53 Mega |
| Ultrasound sensor 2 | Trig | DO 50 Mega |
| Echo | DI 51 Mega |
| Ultrasound sensor 1 | Trig | DO 48 Mega |
| Echo | DI 49 Mega |
| L298N driver | ENA | PWM 9 |
| ENB | PWM 10 |
| IN1 | DO 8 |
| IN2 | 7 |
| IN3 | 6 |
| IN4 | 5 |
| 12V | Vin |
| Motors (Right) | Vcc | 5V |
| GND | GND |
| M1 | OUT1 L298N |
| M2 | OUT2 L298N |
| Motors (Left) | Vcc | 5V |
| GND | GND |
| M1 | OUT3 L298N |
| M2 | OUT4 L298N |
| ESP8266 Wi-fi | Tx | PWM 2 Uno |
| Rx | PWM 3 Uno |
| Vcc | 3.3V |
| GND | GND |
| EN | 3.3V |

## Actual implementation:

Once the model was done, we went for 3D printing. The model was too large for all the 3D printers available, and the cost was too high. Therefore, we had to accommodate and convert the model into a prototype. The prototype represents the chair through its external wheels connected to the small model. The model was redesigned from scratch to fit the machines’ requirements.

A grey object with a wire

Description automatically generated with medium confidence

Figure 7: Redesigned prototype

A grey object with wheels

Description automatically generated with medium confidence

Figure 8: Model exploded view

All the previous characteristics of the model are still met yet it is more compact and does not fit on a chair. Even the movable platform is still there with the screw and the stepper motor.

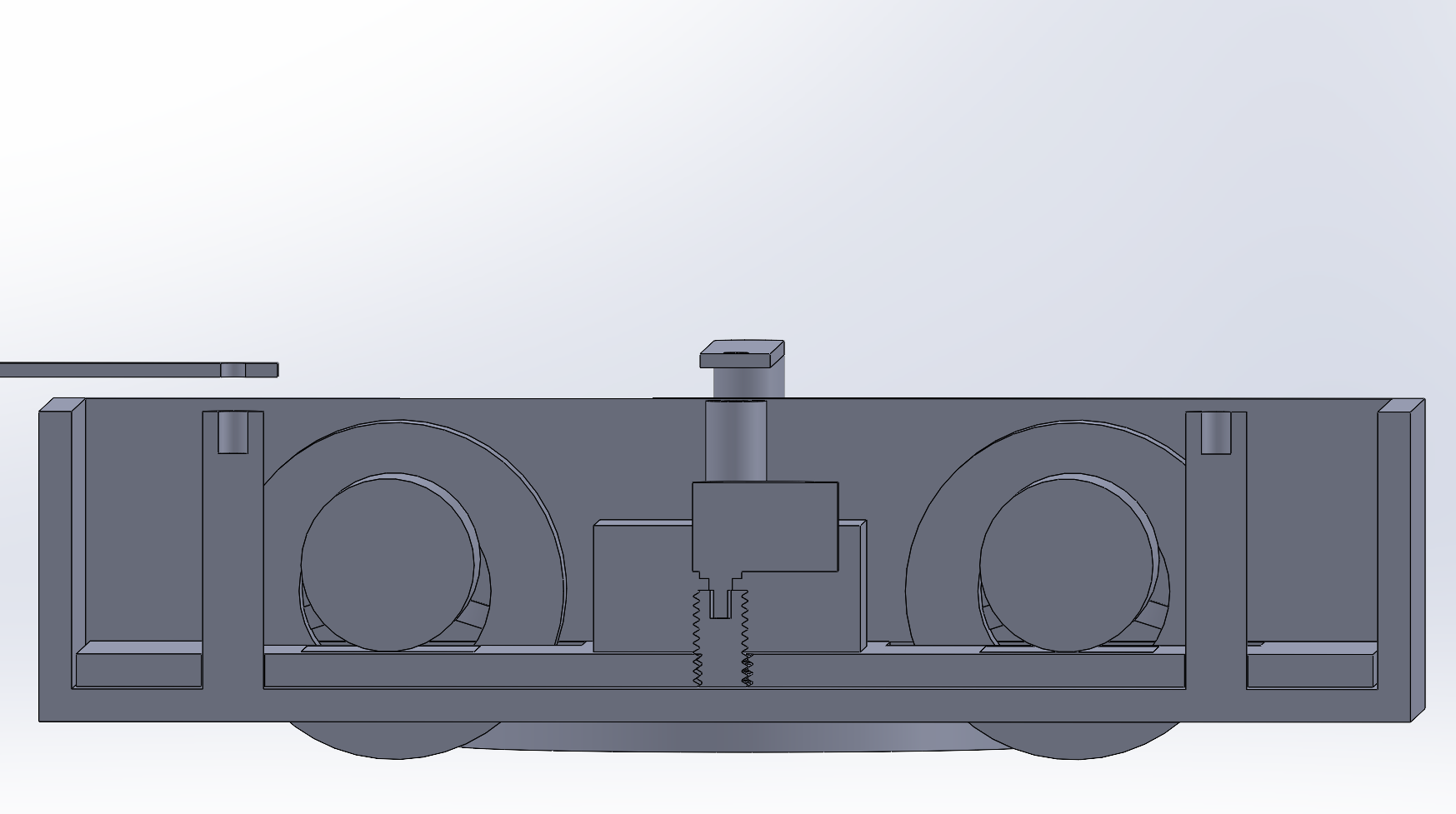


Figure 9: Section view of the new model

The model was to be printed at Dynamic IT Solutions. After one week, we received an inaccurate model with multiple problems. First, the movable platform’s wholes were tighter than the columns on which it should slide to lift the motors and the wheels so we had to carefully rub off the print to make it fit. Then, the screw was printed without the whole of the stepper’s shaft. This problem was not fixable, so we had to print a new screw. In addition, the teeth of the screw’s whole were not precise at all and very susceptible to slippage. Finally, the assembly was printed with a 25% filling, making it easier to break. The problems faced were not taken into consideration and the time was not in our favor. Therefore, we fixed the columns using resin glue, we enlarged the holes, and removed the stepper for safety issues. The stepper should be fixed on the bottom ceiling of the cover. Since it was printed with 25% fitting, the cover started to break due to the weight of the motors and the platform.

It should be noted that the code of the stepper motor was ready, and the connections were made. They were suppressed later after facing this issue. The stepper’s main role was to press down the platform to make sure that it did not lift while moving due to its weight. Instead of using a stepper motor, we fixed the platform using a U-shaped aluminum bearing to press down. It was fixed on the lower ceiling of the cover and tightly held down to the platform. The motors were fixed on the platform. All their connections were done internally with the driver and the wires were moved upwards through a whole.

Both Arduino boards were fixed on the cover’s top. All the sensors were connected through a large breadboard for a clearer view and an easier debugging scheme. A common ground was used for all components.

The needed calibrations were done based on the room we were working on. The strength of the Wi-Fi is first tested at the destination location. Then a threshold is set in the code for the chair to arrive correctly and accurately.

The motors were calibrated for a smooth rotation and the delay between the two sides was fixed to ensure that the robot would move correctly.

The final list of components used is:

* Wi-fi sensor ESP8266
* 4 DC geared motors JGP37-520
* LDR sensor
* Sound Sensor
* Three Ultrasound sensors
* Arduino Uno
* Arduino Mega
* L298N dual motor driver
* One breadboard
* 3D printed model.
* A rechargeable 12V battery
* Wires

# Current Implementation Summary

As a quick summary of the actual implementation of the robot, one can say that the target of the robot was achieved, and the team was able to accommodate different problems faced.

Starting with the logic sequence of the code, the initializing code can be considered the Wi-fi code computed by the Arduino Uno. After importing the needed libraries for the Wi-fi connection, and the needed variables, the main loop of the program fetches the Wi-fi strength of the network through the AT command: AT+CWJAP?

The response of this command is saved in ‘data’, a global variable used in the code. Since the Wi-fi strength starts with a -, the index of this character is used to parse the RSSI value and save it as an integer in RSSI. Then this value is sent by using the Wire library to the Arduino Mega through SDA pin.

The Arduino Mega, after all the needed imports and initializations, receives the RSSI value and saves it as a global variable. This value is constantly updated in the loop of the Arduino Mega. After entering this loop, the LDR condition is checked to make sure that no one is sitting on our virtual chair. Once the LDR output meets the required threshold calibrated throughout the process, the sound sensor is activated. If a clapping voice is detected, the system starts running. The methods that activate the ultrasound sensors and calculate the distances of obstacles are called. Then the main method that moves the motors accordingly is called.

This method fuses the data coming from the ultrasound sensors with the Wi-Fi strength. The logic implemented in this method implies that if the robot approaches the target, meaning that the current different between the RSSI value and the target’s RSSI value is decreasing, the target is moving in the correct direction while checking for obstacles. If the robot is moving away from the target location, it rotates 90 degrees and rechecks. If keeps rotating until it reaches a point where it’s getting closer to the target location. Once this location is reached, the robot stops and breaks it’s motors and waits if it is moved again for a clapping sound to restart the cycle.

## Old vs. New

Although our will to implement the design on an actual chair, the size and weight of the chair were a principal constraint that required the use of larger motors and a bigger power source. Thus, the design was implemented on a 3D printed chassis to mimic a chair. We were also able to simplify the algorithms and connections without losing fundamental functionalities. Although the prototype is still able to get to its target, it now requires more time due to the absence of beacons and the use of a single Wi-Fi detection module. The prototype now must move for a short time, stop and check the Wi-Fi strength then decide the direction it should move towards.

One cannot deny that the initial solution was much more accurate and much more complex. The algorithms were all there. The components were bought, and the work was done. The only problem was the cloned HM10. Using Wi-fi is less accurate due to the constant fluctuations in its strength and the interference with other devices in the room.

For the 3D model, the first model was a complete product. The design was very accurate and esthetically pleasing. It accommodated all the restrictions that one might face.

The second model was just a prototype that did not fit on any chair. However, we managed to visualize the chair through the connected wheels on the side. The weight was not a problem anymore but the printing was a major issue.

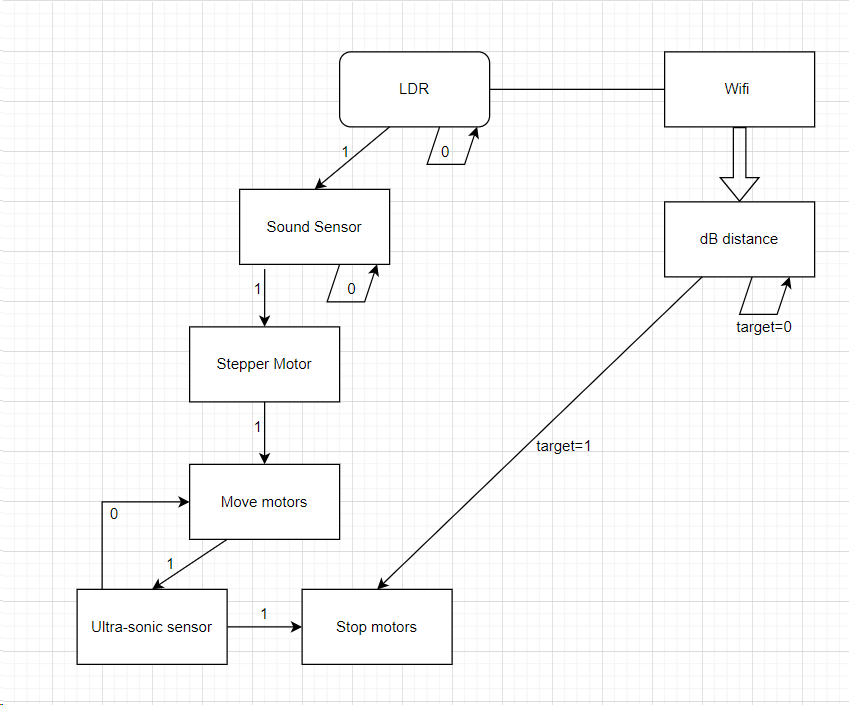


Figure 10: Flow graph of the new implementation

A circuit board with many wires

Description automatically generated

Figure 11: Hardware Connections

# Conclusion

In conclusion, the successful completion of our robotic projects marks a significant achievement in our robotic journey. Throughout the course of this project, we addressed various challenges, from the model’s design to the model’s printing, the modules’ cloning, and the complexity of the algorithm.

The project aimed to move a chair from one location to its set location under the desk. The main purpose of this project is to deploy the mechanism on many chairs such as in a library. The mechanism eases the movement of the chair in large buildings.

Through meticulous planning and execution, the team members gained the adaptability to uncontrolled problems.

The collaboration and teamwork among project members played a crucial role in the success of this project. Regular meetings, effective communication, and the division of tasks ensured a smooth workflow and a positive environment.

As we reflect on this project, we recognize the areas of improvement, mainly in achieving the first model that was designed and overcoming the discussed difficulties.

# References

B. (2021, December 29). *ESP8266 AT commands list and working explained - Electronics fun*. Electronics Fun. <https://electronics-fun.com/esp8266-at-commands/>

*How to Flash the Firmware on Clone HM-10 BLE Module using Arduino Uno*. (n.d.). <https://circuitdigest.com/microcontroller-projects/how-to-flash-the-firmware-on-cloned-hm-10-ble-module-using-arduino-uno>

*Arduino Uno + ESP8266-01*. (n.d.). <https://virtuino.com/index.php/virtuino-se/22-arduino-uno-esp8266-01>

*Interface L298N DC Motor Driver Module with Arduino*. (2022, October 16). Last Minute Engineers. <https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/>