MAK HILL Publications Journal of Engineer and Applied Sciences

Research Article
doi: 10.59218/makjeas.2022.82.91

OPEN ACCESS

Key Words
Robotics, autonomous, logistics, telerobots, inventory bots, intelligent robotics

Corresponding Author
Ammar Arif Ansari
School of Engineering and Physical Sciences, Heriot-Watt University, Dubai, United Arab Emirates

Received: 12 November 2022
Accepted: 2 December 2022
Published: 29 December 2022

Citation: Ammar Arif Ansari, Shayma Akbar, Joshua Hedwic Goldwin, Hamza Karim Shahryar, Milen Jacob Alappattu and Talha Ubaid, 2022. Design and Implementation of a Semi-Autonomous Telerobotic Warehouse Management Robot for Logistic Applications. J. Eng. Applied Sci., 17: 82-91, doi: 10.59218/makjeas.2022.82.91

Design and Implementation of a Semi-Autonomous Telerobotic Warehouse Management Robot for Logistic Applications

Ammar Arif Ansari, Shayma Akbar, Joshua Hedwic Goldwin, Hamza Karim Shahryar, Milen Jacob Alappattu and Talha Ubaid

School of Engineering and Physical Sciences, Heriot-Watt University, Dubai, United Arab Emirates

ABSTRACT
The purpose of this study is to describe, in detail, the process of planning, executing and working on an autonomous inventory bot. This study provides a brief overview of the project, the scope of the robot, the building and the economic analysis of the system. The working, construction, digital design and modelling of the robot are explained in detail. The specifications of this project include but not limited to, use of Arduino Mega Microcontroller as the main controller, the navigation of shelves independently, led indicators notifying the status of the robot, autonomous rerouting and use of audio commands to alert a person obstructing the robot's path while minimizing the dimensions of the robot to as minimal as 300×300×500 mm.

INTRODUCTION

To cater to the demand of fresh and innovative inventory robots, our team has created our version of the inventory robot. Implementation of the project, a robot named John Smith, is a machine that embodies a blend of modern design and a comprehensive system[1]. Besides the given specifications, it has the additional feature of adaptability to different rack heights and the ability to access higher and lower racks using a scissor lift. We have limited our robots' dimensions to 300 (l)×300(w)×500 (h)[2].

Features of our implementation include:

- Scanning of items on shelves at two different heights using the scissor lift (which can be expanded to give access to the robot at multiple shelves)
- **Obstacle detection:** Using ultrasonic sensor, sound alerts to request people to move out of the robot's path and alternate routing when the obstacle has not been moved from the path[3]
- Line following system used to traverse the entire supermarket on a given fixed path

The scope of this project does not include a sophisticated software implementation for comparing and analysing large datasets of inventory for items scanned by the robot.

MATERIALS AND METHODS

With online shopping booming, brick-and-mortar stores are racing to catch up to the competition. A robotic friend cruising around supermarkets is soon to become the new norm. These robots can have multiple purposes like sanitation, hazard identification, inventory restocking and scanning and identifying misplaced or mislabelled items.

Concept development and evaluation: The team had reviewed numerous design ideas and concepts to represent our interpretation of the robot. Several body designs and features were discussed in the early weeks of this project that have resulted in our version of the best robot[4].

Early versions of JOHN consisted of a body with three wheels (two motored and one caster wheel), a neck (to contain all wiring) and a head (for scanning).

Our first design used a rotating camera as its head which scans inventory and detects obstacles with an LCD screen that gives live status of the bot. The Arduino would be placed inside the main body with it covered from the top and accessible from below. The second consisted of a fixed camera head and an open-faced body for ease in accessibility to the wiring[5].

Mechanical design: Our later designs wanted to introduce more accessibility and mobility into robot which led us to use the scissor lift. We also revised the shape of the base as a more classic box to maximise usable area. With this design update, we had finalized on using a barcode scanner for scanning items and using the ultrasonic sensors for obstacle detection. The three-wheel combination was retained in the newer designs. We played around with the number of scissor links to use in the robot to give it maximum range while still being in the prescribed limit. The two rear wheels use a 3-6V DC Gear Motor for power and the third front wheel is an undriven swivel caster wheel used mainly for stability and ease in steering the robot. This three-wheel design was chosen for its convenience in changing directions which can be done by altering the relative rate at which the two powered wheels are rotating. When two wheels rotate at the same rate and in the same direction, a robot will continue to move straight. These two motors are powered through a Motor Driver which gets it from a 6-7V DC Source (Fig. 1 and 2).

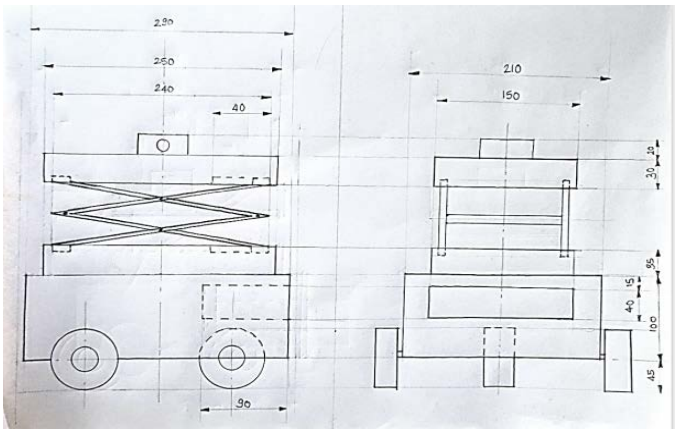

Fig. 1: Mechanical design

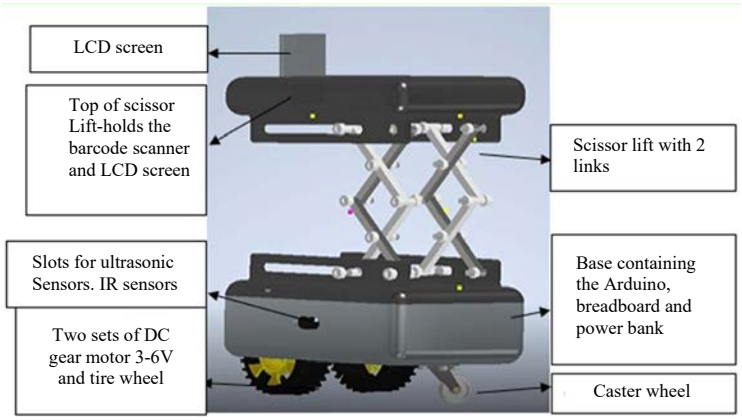

Fig. 2: CAD model rendering

The scissor lift is powered by a servo motor. Our design uses a two-link lift with two sets of scissors on each side of the robot, the two sets connected by two perpendicular rods: One at the base and one in the centre. The perpendicular rod at the base is connected to the servo motor via a sturdy motor. A 180° rotation of the servo moves the lift from its lowest to its highest position, the distance covered being ≈20 mm. The power for this servo is provided by a 5V DC Source. The Barcode Scanner is placed on top of the scissor lift and connected to the Arduino Mega in the base through an Arduino compatible USB Host Shield.

Sensor systems: One of the key features of any automated robot is to be able to receive information from different environments and situations. Sensors are the tools that would interpret this kind of information to either humans or machines. And our robot, John, is well equipped with diverse kinds of sensors for such unique functions. The movement and tasks of our robot is dependent on the functionality of the sensors. The Arduino Mega provides a great platform for the execution and working of these sensors. The sensors we used are as follows:

Ultrasonic sensor: Ultrasonic sensor is a device that can detect and measure the proximity of an object in terms of distance. This device uses ultrasonic sound waves which bounce back from the object to find out how far away the object is from the sensor. The ultrasonic sensor consists of a transducer, the part of the sensor which sends and receives the sound. The difference in time between emission and receiving of the ultrasound is what determines the distance between the object and the sensor. Hence, the ultrasound always bounces back from any object to the sensor.

Our robot is placed in an environment where human intervention is possible. Due to this, the robot needs the ability to detect the humans and take the necessary action rather than clashing with them. For this purpose, we are making use of the ultrasonic sensor which can detect if any object or human is in the way of the robot. Our specifications also include to maintain a physical distance of 20 cm from the shelves. The robot could autonomously detect the distance from the shelves using the ultrasonic sensor.

We are making use of 3 ultrasonic sensors. One of them is placed on front side of the robot, to detect any object or person, one placed on the left side of the robot, to measure the distance from the robot to the shelf and one placed on the right side (Fig. 3). The third sensor is used to check if the path is clear whilst the robot takes an alternate route.

If there is any object or human (less than 30 cm away), blocking the path of the robot, the sensor in the front, which always remains active, will detect the obstacle and inform the Arduino that there is an object nearby. When this occurs, the robot comes to a complete stop and gives an audio warning to the person using a Piezo buzzer. If the obstacle does not move, the robot will take an alternate path, which involves going around the obstacle and re-joining the planned path (Fig. 3-5).

Infrared (IR) sensor: Infrared sensors or IR sensors is an electric component, used to detect several characteristics by emitting infrared radiation. This sensor is very much like that of a human vision to detect obstacles and paths.

For the movement of our robot, we have decided to use the method of line following. This method involves the robot following a path placed on the floor or an invisible path involving infrared lines. For this method, the robot needs to detect the line and move along the line and detect it simultaneously. This is possible using the IR sensors. The IR sensors can detect white or black due to their recognizable wavelengths.

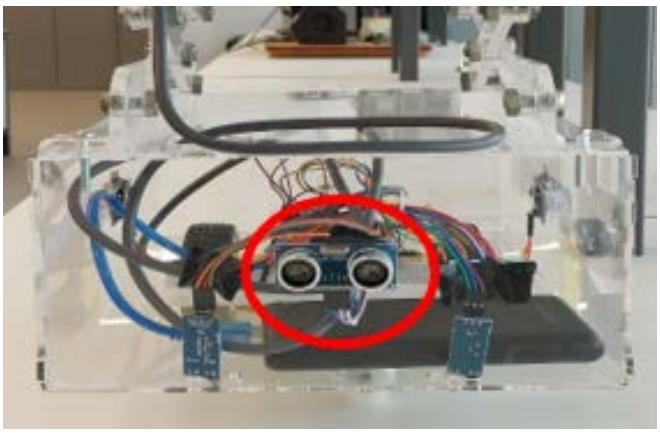

Fig. 3: HC-SR04 Sensor placement

Fig. 4: IR Sensor detecting stops

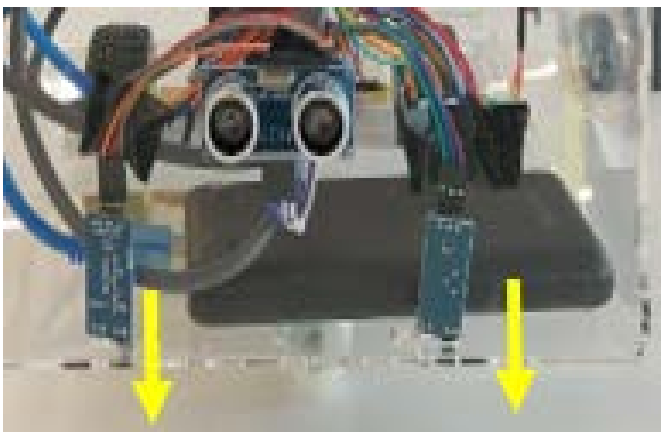

Fig. 5: IR Sensors detecting line

Our robot is equipped with 2 IR sensors which are used for following the line. These two sensors are placed at the bottom of the robot and as close to the ground so that the sensors could detect the highest wavelength possible without any errors. The two sensors: Left and right, will guide the robot along the line by detecting the color of the line and informing the Arduino to move in the direction of the detected color every 10 msec. We have equipped the robot with particularly 2 sensors because of the possible lateral movements, which would involve turning left or right. Whenever the left IR sensor detects a higher value, the robot turns towards the left and when the right IR sensor detects a higher value, the robot turns towards the right. However, if both the IR sensors detect a higher value, it would mean the robot would have to go straight.

Color sensor: A color sensor is a device that detects color, or its only purpose is to detect color. Similar to the working of the ultrasonic sensor, instead of sound waves, this sensor emits its own light and then obtains the reflected light from the object and determines its color. The color sensor module only has filters for the colors red, green and blue.

For the robot to know that it has reached a particular shelf, we planned to use the sensor to recognize distinct colors for different shelves, hence, it would be able to autonomously detect which shelf it has reached. This sensor was also used to trigger the scissors movement for the barcode scanner to scan the items on the shelf. We have planned to put different colors near the shelves corresponding to the category of items placed on them. However, only one color would be used in every shelf to trigger the up and down movement of the scissors. Every time the sensor detects the red color on the floor, the robot would stop and the scissors would start moving up and down, hence recognizing that there is a shelf nearby. Upon moving forward, it would detect another unique color, which would inform the robot at which shelf it is located.

However, due to the technical incompatibility of the TCS3200 color sensor, we have used an extra IR sensor, placed on the left side of the robot. This IR sensor acts as a color sensor. Once it detects a black mark placed on the floor, it would inform the Arduino that it has reached a shelf and trigger the upward and downward movement of the scissors and start scanning the items on the shelf. Once it reaches the end of the shelf, it would detect another black mark which would stop the movement of the scissors and continue to follow the path. The different shades of the black mark is what would differentiate the shelves.

Software design and motor control: Any robot that would work autonomously or without human intervention needs an initial setup and a set of rules to follow so that it works according to the specifications desired. Coding the commands on a specific integrated development environment (IDE) that is compatible with the Arduino is one of the main necessities for our robot to function properly. Everything the robot does is mainly dependent on the code uploaded on the Arduino. Our code is made and divided into different methods or functions, for abstraction, which would reduce complexity and is understood easily by the users. The methods defined are then used accordingly in a loop. This loop would run for every ten milliseconds, executing different tasks according to the unique needs and scenarios.

The loop contains parameters that are to be measured during the loop interval. Some of them include the distance measured by the three ultrasonic sensors, namely the distance from the robot to any obstacle (distance Front), the distance from the robot to the shelves (distance Left) and the distance from the robot to any obstacle while the robot is taking an alternate path (distance Right). The method Usb.Task() is a function that activates the barcode scanner, which is connected to a USB host shield that is connected to the Arduino. The remaining three parameters are measured by the IR sensors. These values are read every 10 milliseconds to give an output of HIGH or LOW. Using these parameters, the robot does its basic function of following the path (Fig 6-9).

To execute different functions during different scenarios, we use the 'if' conditions that would define certain conditions and execute the appropriate methods if those conditions are satisfied.

For example, if both the IR sensors detect the black line and third IR sensor does not detect any black marking, this would mean that the robot need not scan any shelves and just move forward. Hence, the code is defined in such a way that when both the IR sensors detect the black line, it moves forward, or in other words, both the motors of the wheels move forward. While doing so, it also gives an LED indication to the users that the robot is following the line (Fig. 10 and Table 1).

Or if we take another example, if it involves other scenarios, such as when the third IR sensor detects a black marking, it needs to trigger the upward and downward movement of the scissors and give an indication that is scanning the items. Or if an obstacle which is 40 cm away comes in the way of the robot, the robot needs to give an audio warning to the user and then take an alternate route. These conditions are also defined using the 'if' conditions and executes different tasks.

```
void loop()
{
  dF = distanceFront();
  dL = distanceLeft();
  dR = distanceRight();
Serial.println("Distance from Objects on Path = ");
Serial.print(dF);
  Usb.Task();
  valueIR1 = digitalRead(IR1);
  valueIR2 = digitalRead(IR2);
  valueIR3 = digitalRead(IR3);
```

Fig. 6: Parameters measured in loop

```
if (valueIR1 == HIGH && valueIR2 == HIGH && valueIR3 == LOW)
{
  // Move both the Motors Forward when both the IR sensors are on Black Line
  Forward();
  blinkGreen();
  blinkYellow();
 // Serial.println("John is moving Forward");
  //Serial.println();
  lcd.setCursor(0,0);
  lcd.print("Moving");
  lcd.setCursor(0,1);
  lcd.print("Forward");
}
```

Fig. 7: Condition when IR sensors detects line

There are many methods and functions defined with certain tasks depending on the inputs of the sensors. The movement methods defined are:

The motors attached to the main circuit board "Arduino Mega" are not compatible to direct use, hence an interfacing integrated circuit (IC), namely, L298D Motor Driver, is used for giving the DC Motors capability to be controlled using digital signals from the main microcontroller. The PWM pins, aka, Pulse Width Modulation pins, modulates and manipulates the width of a digital square wave emitted to control the on/off state on an electronic component, hence making it possible for the components to be controlled digitally in multiple variants i.e., speed, brightness, torque etc (Fig. 11 and 12).

Functions defined for motor control:

- **Forward():** This method sets the two gear motors (the wheels) to high, which makes both wheels move forward and hence, the whole robot would move forward
- **Left():** This method sets the left motor to high, which would only make the left wheel move,

```
if (valueIR3 == HIGH)
  {
    // Stop both Motors when IR sensors are on Black Line
    Stop();
    delay(1000);
    moveScissor();
    blinkRed();
    Serial.println("John is Scanning Items from the Upper Shelf");
    Serial.println();
    lcd.setCursor(0,0);
    lcd.print("Scanning Items");
    lcd.setCursor(0,1);
    lcd.print("from Shelves");
    delay(2500);
    Forward();
  }
```

Fig. 8: Conditions as IR sensor detects shelf mark

```
if (dF < 30)
{
  // Robot follows alternate path
  Stop();
  buzz();
  alternatePath();
  Forward();
  blinkRed();
  blinkGreen();
  blinkYellow();
  lcd.setCursor(0,0);
  lcd.print("Taking");
  lcd.setCursor(0,1);
  lcd.print("Alternate Path");
}
```

Fig. 9: Condition as HC-SR04 sensors detects Obstacles

which turns the robot towards the left. Another method, *altLeft*, would set the left wheel to move backward and the right wheel to move forward. This would also result in the robot turning towards the left but smoother since both motors are involved. This method is used only for alternate path

- **Right():** Here, the right motor is set to high, which would only make the right wheel move, which turns the robot towards the right. Similarly, *altRight*, would set the right wheel to move backward and the left wheel to move forward which would result in a smoother right turn. This method is also used for alternate path

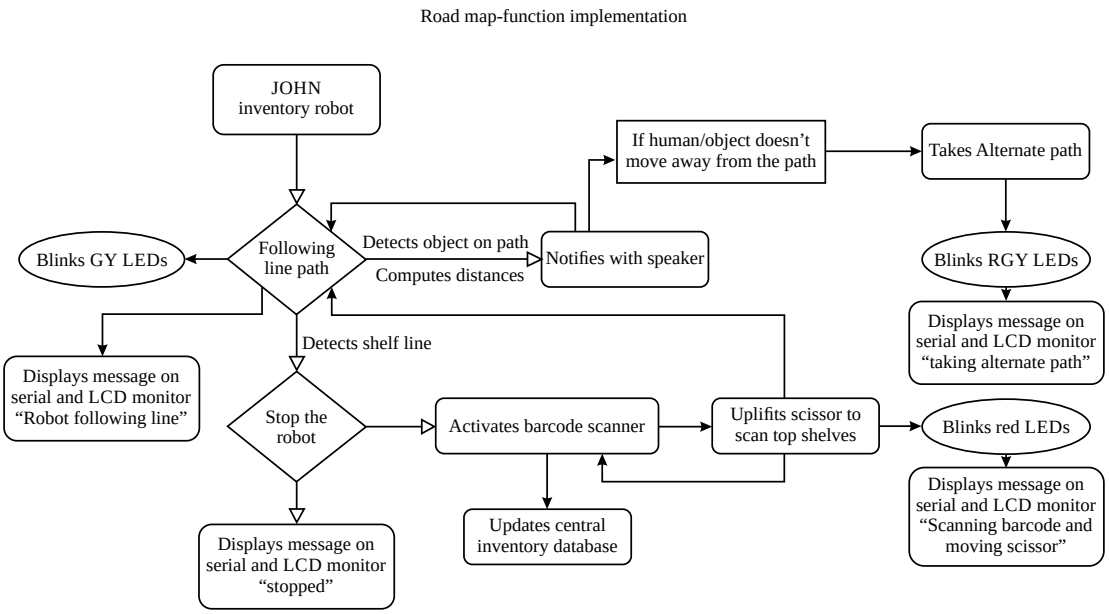

Fig. 10: Algorithm implementation roadmap

Table 1: Methods and Functions Used in the Algorithm

Function	Description	Declared when
alternatePath()	This method is defined to take a pre-planned alternate route It is mainly coded with the movement methods such as the altleft() and altright() methods to go around the obstacle	If(dF<30)
moveScissor	This method activates and triggers the scissors connected to the servo motor. It is encoded with a loop to move the servo from 0 to 270 degrees which would cause the scissors to go up and vice versa	If(valueIR3 == HIGH)
blinkRed()	This would activate and pulse the Red LED bulb	If(dF<30), f(valueIR3 == HIGH)
blinkGreen()	This would activate and pulse the Green LED bulb	Forward(), Right(), Left(), If(dF<30), If(valueIR3 == HIGH)
blinkYellow()	This would activate and pulse the Yellow LED bulb	Forward(), Right(), Left(), If(dF<30), If(valueIR3 == HIGH)
distanceFront()	This method is used to activate the inputs of the ultrasonic sensor placed on the front and return the distance from the obstacle to the robot	Used in the Loop
distanceLeft()	This method is used to measure the distance from the shelf to the robot	Used in the Loop
distanceRight()	This method is used to measure the distance of any obstacle from the robot while taking the alternate route	Used in the Loop
buzz()	This method is defined to activate the Piezo buzzer by looping it to send high and low signals to the buzzer	If(dF<30)

- **Stop():** This method sets both the motors to low, which would make both the wheels to stop moving. The robot would not move when this method is declared. Other methods used in the code as show in Fig. 10

RESULTS AND DISCUSSIONS

We made our initial simulations and connections using the website www.tinkercad'.com, a popular website to create circuits involving the Arduino. During the initial stages of our design, we started working with the Arduino UNO, which was available on the online tinker cad. The basic structure of our wiring and components was designed in Fig. 11.

However, it was clear that we required more pins than the Arduino UNO provided. This gave us the chance to use the Arduino MEGA. Since the Arduino MEGA wasn't available on the online simulation, computer simulations were not easily available. Hence, we evaluated each and every component on the real MEGA instead of simulation.

Testing and calibration:

DC motor wheels: After receiving the Arduino kits, we conducted an initial test by using a simple Arduino code for the DC motors to run. The motors at first were running at terribly slow speed and this was an issue as these two motors would be supplying the wheels with power that needs to be enough for the whole body to move smoothly. As the body was partially moving in the absence of the scissor lift, we decided to try out different methods to increase the speed of the motor such as changing the motor driver, using a different power bank supply and using a pair of different motors. Finally, we decided to use the power supply

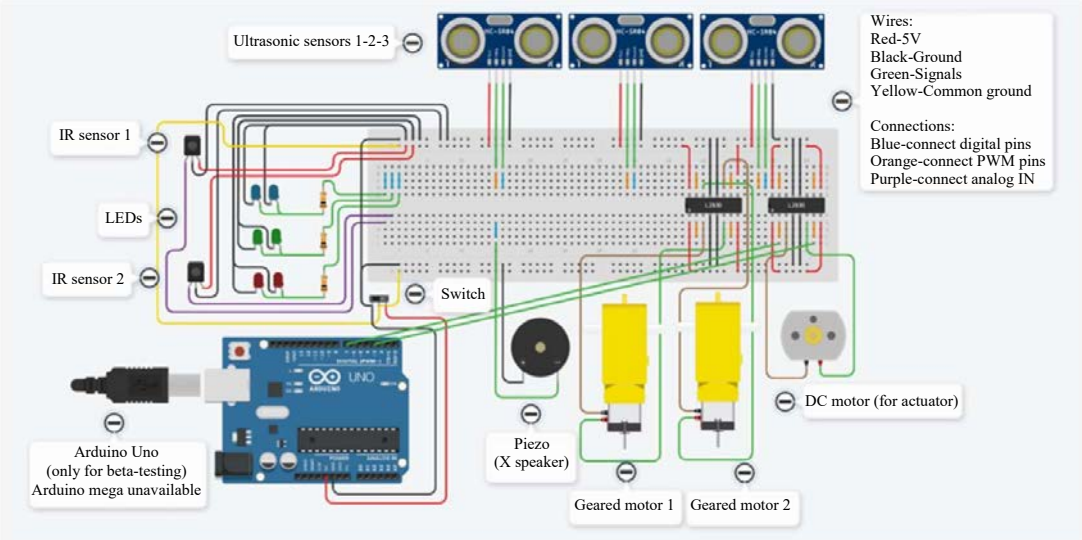

Fig. 11: Design and wiring simulation of robot

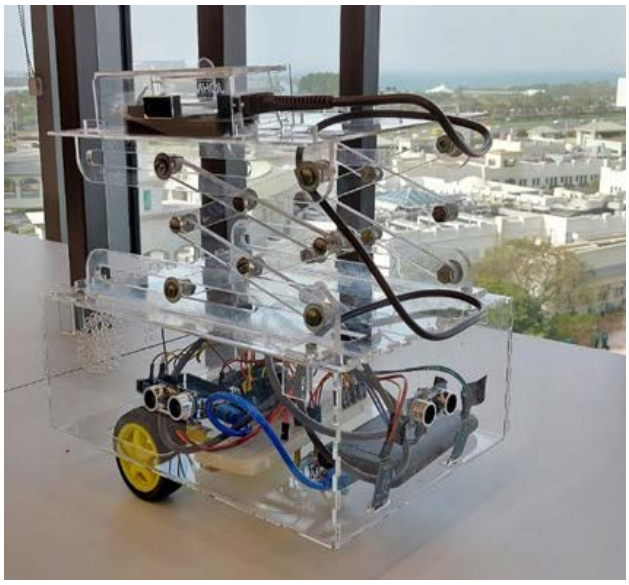

Fig. 12: Final look of the robot

available in the lab and see if the problem was with the motor or the supply (power bank) we were using. After conducting the test, we had found out that the motor was running fast enough when given enough voltage from the voltage supply therefore, we decided to use a DC generator.

Scissor lift: The scissor lift was a crucial part of our design, we faced a lot of issues at first regarding the scissor lift however found solutions where needed. Our first issue we faced was to make it move up and down, we initially used an actuator that was powered using a 5V motor and had a 3D printed cover however when the scissor lift was laser cut, the actuator was not powerful enough to make it move up and down when the top was added. We, therefore, decided to use an actuator that was powered by a motor with greater voltage, greater torque and it did the job. The other issue we faced with the scissor lift was while going up and down the base was not aligned and the structure was not stable enough. To find a solution to this problem we had first tried to tighten and loosen various bolts however, it did not resolve the problem. A metal rod that was connected to both

sides from the bottom and was free to move back and forth and this helped us stabilize the lifting process.

Laser cutting: Laser cutting was a crucial part of our project which had initially started laser cutting, the process was going smooth until we started to assemble the parts already laser cut, the interlocks between the base and sides were not fitting. At first, we suspected our dimensions were inaccurate however after checking the dimensions on our file and comparing them to the parts already laser cut, it was not the issue. We then suspected that the software had a calibration issue when connected to the laser printer and after making changes of a few mm on our dimensions we successfully printed the parts needed with a perfect fit.

Assembly: After laser cutting all the parts needed we started to assemble the robot during this process we faced multiple difficulties and had to test out different ways to tackle the problem some of the difficulties we had faced were drilling through some parts as acrylic could be sensitive therefore it needed to be handled with care, we had first conducted a test on a small portion of acrylic and moved our way into the parts needed to be drilled. Moreover, folding acrylic using heat had to be assessed first by our team members before doing it and we had learned a few tips while doing this.

CONCLUSION

The goal of the automation system is to do and assess all required activities more effectively, reliably and precisely than a human worker could. The most essential factor, of course, is to make the system more secure. Our self-contained sensor-based inventory checking system was created to give our clients with continuous, inventory monitoring and reporting to a central database to manage and maintain the inventory status of the products. Automation is useful for speeding up the flow of information and allowing for the monitoring and control of operations more effectively and efficiently.

REFERENCES

1. Hussain, A.M.A. and H.M.D. Habbi, 2022. Design and performance analysis of a 3-phase induction motor for solar photovoltaic fed pumping system. J. Eng. Applied. Sci., 15: 773-782.
2. Trujillo, J.L.A., L.B. Vazquez and R.R. Serrezuela, 2022. Mathematical model of the march of a bipedal robot applying quaternions oriented to the development of anthropomorphic robots. J. Eng. Applied Sci., 15: 88-93.
3. Jancy, S. and C. Jayakumar, 2022. Sequence statistical code based data compression model using genetic algorithm for wireless sensor networks. J. Eng. Applied. Sci., 14: 831-836.
4. Kumar, M., T.D. Singh, 2022. Design and development of bluetooth base home automation system using FPGA. J. Eng. Applied Sci., 16: 370-376.
5. Ghani, M.A., M.B. Jali and N.A. Zulkifli, 2022. Confusion in design and facilities of layout plan: GMP r equirements. J. Eng. Applied. Sci., 14: 3277-3282.