Afro-Asian Journal of Scientific Research (AAJSR)
المجلة الأفروآسيوية للبحث العلمي (AAJSR)
E-ISSN: 2959-6505
Volume 2, Issue 4, October - December 2024
Page No: 210-220
Website: https://aajsr.com/index.php/aajsr/index
SJIFactor 2024: 5.028 ISI 2024: 0.580 معامل التأثير العربي (AIF) 2023: 0.51

Design and Construction of a Self-Driving Wheeled Vehicle Using Ultrasonic Sensors and an Arduino UNO Board

Ahmed Saed Ishleebtah[1*], Nouri Ali Almabrouk Alhashi[2]
[1]Physics Department, Faculty of Education, Bani Waleed University, Bani Waleed, Libya
[2]Higher Institute of Medical Sciences and Technologies, Bani Walid, Libya

تصميم وبناء مركبة عجليه ذاتية التحكم باستخدام حساسات الموجات فوق الصوتية وبطاقة اردوينو

أحمد سعد محمد اشليبطة[1]، نوري علي المبروك الحاشي[2]
[1]قسم الفيزياء، كلية التربية، جامعة بني وليد، ليبيا
[2]المعهد العالي للتقنيات الطبية، بني وليد، ليبيا

*Corresponding author: ahmedeshlibta@bwu.edu.ly

Received: August 22, 2024	Accepted: October 28, 2024	Published: November 24, 2024

Abstract:
In this paper, a collision-avoidance robot was designed and built to move along a specific path, which consists of various directions. The robot is programmed to avoid collisions within the path and maintain a safe distance from the walls. To solve this problem, we used ultrasonic sensors to detect any obstacles and also to determine the distance between the robot and the walls.

Keywords: Arduino, robot, Design, wheeled vehicle, sensors.

الملخص
في هده الورقة تم تصميم وبناء روبوت تفادي الاصطدام وجعله يتحرك في مسار معين وهذا المسار عبارة عن طريق له اتجاهات مختلفة، بحيث أن يتفادى الروبوت الاصطدام داخل المسار وجعل مسافة أمان بينه وبين الجدران، ولحل هذه المشكلة قمنا باستخدام حساسات الموجات فوق الصوتية للكشف عن أي عائق وأيضا لتحديد المسافة بين الروبوت والجدران.

الكلمات المفتاحية: روبوت، اوردوينو، حساس، مركبة عجلية، تصميم.

Introduction
In recent years, there has been a focus on using Arduino boards to create robots in many Arab and international universities. In Yemen, in 2016, a project was developed for a robot that operates via voice, changing its direction and stopping when it approaches an object. One of the goals of this project is to develop it for use by amputees. In 2014, the Advanced Technology Applications Center in Algeria developed a mobile robot equipped with a robotic arm to transport objects from one place to another [1-4]. In this research, we designed and built a self-driving wheeled vehicle (collision avoidance robot) using ultrasonic sensors and an Arduino UNO board to perform the required task.

Arduino UNO board
A small electronic circuit used for programming the microcontroller from Atmel, ATmega328. This circuit provides ports for connecting electronic components directly to the microcontroller through 6 analog ports and 14 (input/output) digital ports. Of these ports, 6 can be used as PWM outputs, also known as pulse-width modulation (Pulse Width Modulation). The circuit also includes a Crystal Oscillator with a

frequency of 16MHz, in addition to a USB port for communication with the computer. There is a separate power input, as well as an ICSP Header, which is an additional method for programming the microcontroller while it is still connected to the board, unlike the USB. You can consider this Arduino board as a mini development and programming board, ready for direct use. It almost contains everything you need to work on it, whether through the USB port or an external power source like a battery [5-7].

Figure (1): The Arduino UNO Board.

General specifications of the Arduino UNO board.

Table (1): General Specifications of the Arduino UNO Board.

The controller	ATmega328
Operating voltage	V5
Input voltage (preferred use)	V12 – 7
Input voltage limits	V20 - 6
Digital inputs and outputs	14
Analog inputs	6
The value of the DC current on each input/output	mA40
The value of the DC current for the 3.3 V	mA50
Flash Memory	Flash Memory 32Kb (0.5 KB is used to load the application)
Processor speed	16MHz

The microcontroller and memory

Microcontrollers are similar to small computer units, and the ATmega328 microcontroller has a processor speed of 16 MHz and a total memory of 32 Kilo Bytes.

• Boot Loader: The program responsible for how the circuit understands the Arduino C language.

• SRAM: It is the memory used to temporarily store variables.

• Flash Disk: A storage space used to store the program we will write to operate the controller. At first glance, this number may seem very small, but in reality, it is sufficient to write many commands.

• EEPROM: The memory responsible for permanently recording some variables within the controller, retaining their values even after power is disconnected. We can consider it similar to the Hard Disk drive in a personal computer [8-10].

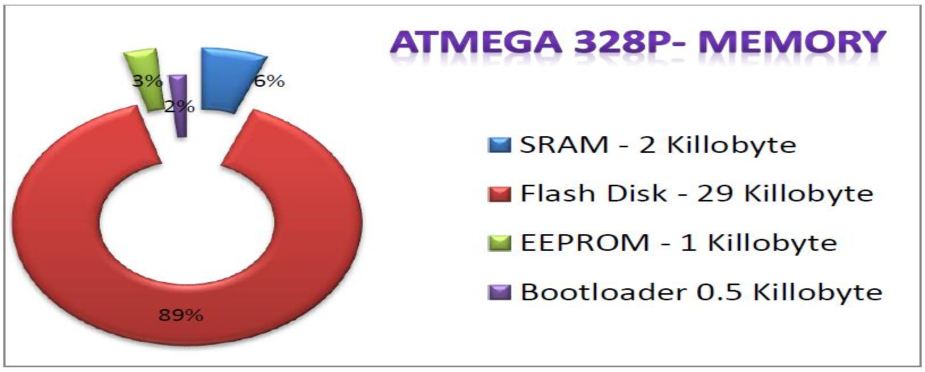


Figure (2): The memory distribution for the Arduino UNO board.

Robot Control and Operation Circuit

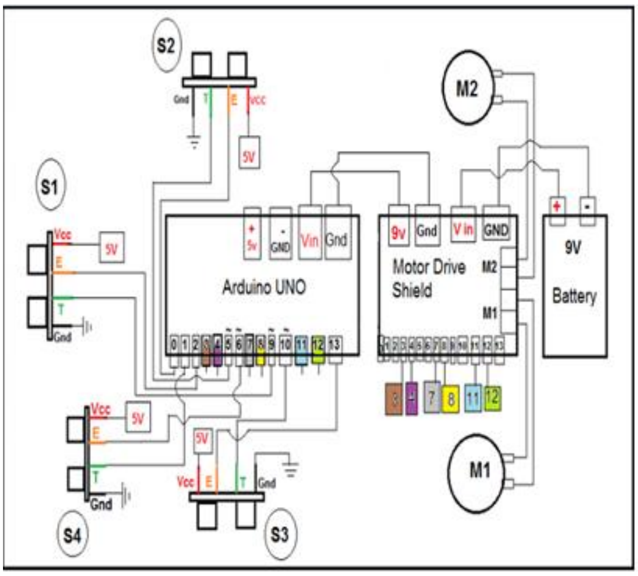
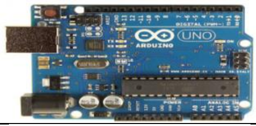
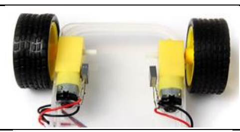


Figure (3): The robot's operating and control circuit.

The components used in the circuit

Table (2): The components used in the circuit.

Number	Type	The number	The shape
1	Arduino UNO board	1	
2	Motor driver cover L 293	1	
3	Ultrasonic sensor	4	
4	DC motor	2	
5	Connecting wires	26	
6	Robot structure	1	
7	The battery	1	
8	Balance wheel	1	

Specifications of the circuit components

Arduino UNO board

It is the card responsible for receiving software commands from the computer and writing them into the memory of the microcontroller embedded within it to perform the main task of the obstacle-avoiding robot, which has been studied and its specifications are in section 2.

L293 Motor Driver Cover

The L293 motor driver shield is one of the most notable features of Arduino boards due to its ease of installation and programming. DC motors consume a very high current that the controller cannot supply directly because its current is very limited. Therefore, we resort to a circuit called a motor driver circuit, which provides sufficient current for the motor to operate based on commands from the controller. This achieves electrical isolation to protect the controller from damage and also protects the controller from reverse currents, as it contains a diode in its internal structure that works only in the forward bias and prevents any currents in the reverse bias [11-13]. This shield is supported by a software library that allows it to work with DC motors, servo motors, or stepper motors. In this research, we will focus on controlling DC motors only.

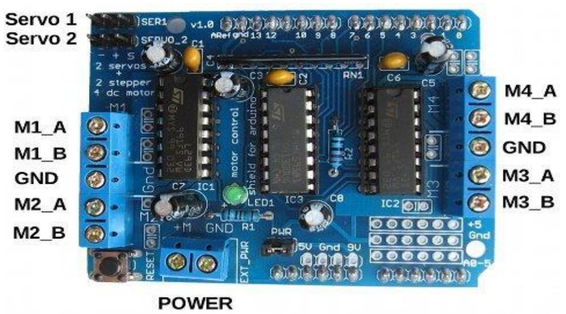


Figure (4): The engine cover drive outlets.

This cover can control the direction and speed of 4 DC motors by connecting the motors to the ports (M1, M2, M3, M4). Each port can only handle 600 mA, so when we want to connect motors to the robot with this cover, we only connect one motor to each port to avoid burning the cover. To control the motor speed, we use the programming code (motor.setSpeed). The speed can be adjusted between 0 and 255 in the programming code, as desired. The direction of the motor rotation can be controlled using the programming code (motor.run()), and one of the following commands should be placed between the parentheses:

(RELEASE) Stop

(FORWARD) Forward

(BACKWARD) Backward

Ultrasonic sensor

It is a sensor that uses the speed of sound to measure the distance between it and the target, where it has two ports on its front side, one for sending sound waves and the other for receiving the reflection of these waves after they collide with the object [14,15]. By recording the time taken for sending and receiving, the distance traveled can be determined using the following formula:

The distance between the sensor and the target = speed * (time / 2) ....The time for the round trip - The speed of waves in a vacuum is approximately 340 m/s.

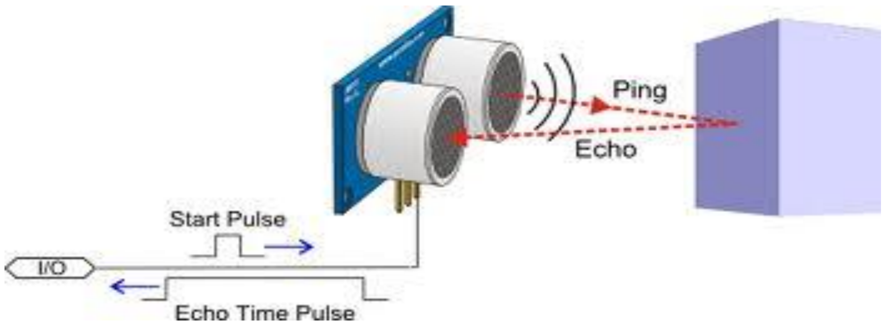


Figure (5): The principle of the ultrasonic sensor.

DC Motor
DC motor converts electrical energy into kinetic energy and operates only on direct current systems. DC motors rotate at a high number of revolutions per minute (rpm) relative to their size and can rotate clockwise or counterclockwise. According to Faraday's theory (if an electrical conductor is supplied with a current placed in a magnetic field, an electromotive force is generated for this conductor).
The field coils are supplied with direct current, creating a magnetic field. At the same time, the armature coils are supplied with direct current through carbon brushes, creating another magnetic field. The presence of these two magnetic fields generates a torque that causes the rotor to rotate [5].
Robot Structure
It is the one that brings together all the components of the circuit.
Balance Wheel
It provides balance at the rear of the robot and offers better rotation and path change for the robot.
Robot's Work Path
The robot's path or work area is a 60 cm wide path surrounded by walls on both sides, with different directions. In this research, we aim for the robot to move from the starting point to the end within the path and avoid colliding with the walls, maintaining a 20 cm safety distance between itself and the walls.

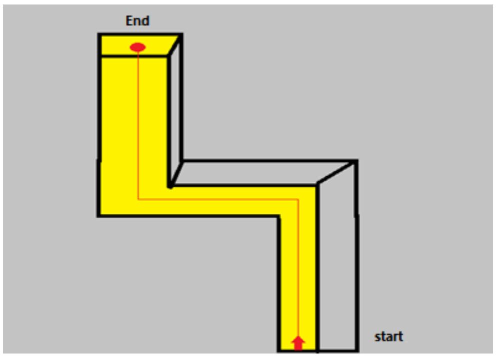


Figure (6): The robot's operational path**.**

Symbols of the tools used in the circuit.

Table (3): The symbols of the tools used in the circuit.

The Name	Code
The sensor at the front of the robot	S1
The sensor on the right side of the robot	S2
The sensor on the left of the robot	S3
The sensor at the top of the robot	S4
The motor on the left of the robot	M1
The motor on the right of the robot	M2

The terminals connected to the control panel
All terminals are connected to the digital ports of the control panel:
1(.Pin 0) The input port connected to the receiver in sensor S2
2.(Pin 1) The input port connected to the transmitter in sensor S4
3.(2 Pin) The input port connected to the receiver in sensor S1
4.(Pin 3) Output port connected to the motor control panel
5.(Pin 4) Output port connected to the motor control panel
6. Output port connected to the motor control panel(Pin 5)Output port connected to the transmitter in sensor S2
7.(Pin 6) Output port connected to the receiver in sensor S4
8.(Pin 7) Output port connected to the motor control panel
9.(Pin 8) Output port connected to the motor control panel
10.(Pin 9) Output port connected to the transmitter in sensor S1
11.(Pin 10) Output port connected to the transmitter in sensor S3
12.(Pin 11) Output port connected to the motor control panel
13.(Pin 12) Output port connected to the motor control panel
14.(Pin 13) The input port associated with the transmitter in sensor S3

Principle of Operation

When we power the motor control board, the controller immediately follows the programming steps. The controller activates the transmitter in the sensors (S1, S2, S3, S4) via the digital pins Pin[9,5,10,6] on the controller, which have been programmed to be outputs. According to the order of activation of each transmitter in the program, the controller triggers the transmitter with a pulse of length (us 10) to send 8 pulses of length (kh) 40) into the void. When the wave hits the wall, it reflects back as an echo to the receiver in the sensor and then to the controller via the digital pins Pin[2,0,13,1], which have been programmed to be inputs. The echo pulse signal is converted into a distance in (Cm) using the Pulse In() function. From here, we obtain the distances between the robot and the walls within the robot's working area, compare the distance values for each sensor, and execute the algorithm conditions as illustrated in the figure (7).

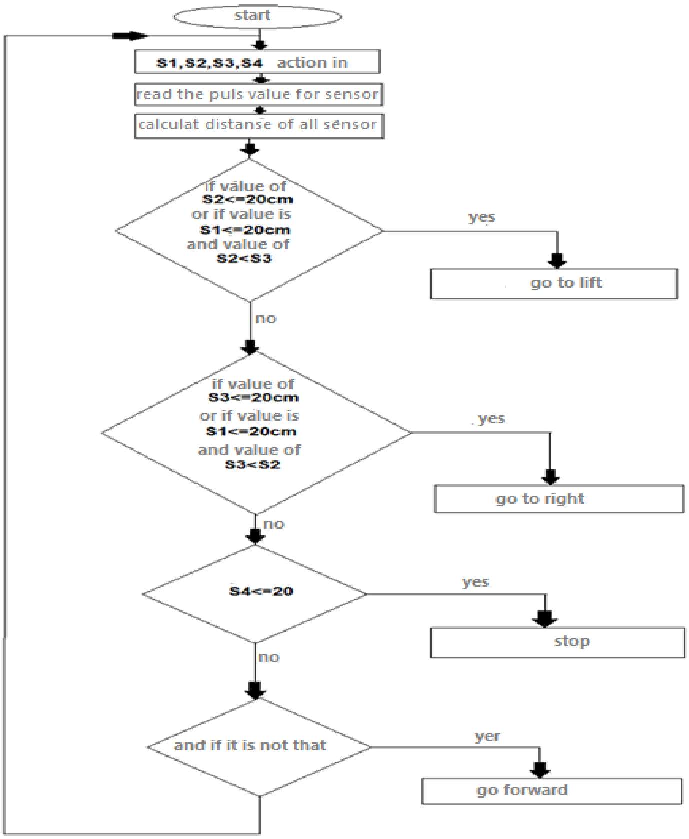


Figure (7): The driver algorithm.

The principle of the robot's operation within its designated path relies on comparing distance values and then executing the conditions written in the program. The program's conditions are as follows:

The first condition

If the value of S2 is less than or equal to 20 cm, or if the value of S1 is less than or equal to 20 cm and the value of S2 is less than S3, the controller executes the first condition, which is to execute the FORWARD command on motor M2 and the BACKWARD command on motor M1, at a speed of (80). The first condition and execution command are as follows:

```
if((s2 <=20)||(s1 <=20)&&(s2 <s3))
{

  left();
}
```

```
void left()
{
motor1.run(BACKWARD);
motor1.setSpeed(80);
motor2.run(FORWARD);
motor2.setSpeed(80);
}
```

The second condition

If the value of S3 is less than or equal to 20 cm, or if the value of S1 is less than or equal to 20 cm and the value of S3 is less than or equal to S2, the controller executes the first condition, which is to execute the BACKWARD command on motor M2 and the FORWARD command on motor M1 at a speed of 80. The second condition and execution command are as follows:

```
else if((s3 <=20)||(s1 <=20)&&(s3 <s2))
{
  rite();
}
void rite()
{
motor1.run(FORWARD);
motor1.setSpeed(80);
motor2.run(BACKWARD);
motor2.setSpeed(80);
}
```

The third condition.

If the value of S4 is less than or equal to 20 cm, the controller executes the third condition, which is to execute the RELEASE command on motors M1 and M2 at speed (0). The third condition and the execution command are as follows:

```
else if (s4 <=20)
{
stop1();
}
void stop1()
{
motor1.run(RELEASE);
motor1.setSpeed(0);
motor2.run(RELEASE);
motor2.setSpeed(0);
  }
```

The fourth condition.

If all the previous conditions are not met, the controller executes the fourth condition, which is to execute the FORWARD command on motors M1 and M2 at a speed of (80). The fourth condition and the execution command are as follows:

```
else
{
forward();
}
```

```
void forward()
{
motor1.run(FORWARD);
motor1.setSpeed(80);
motor2.run(FORWARD);
motor2.setSpeed(80);
}
```

The final form of the robot

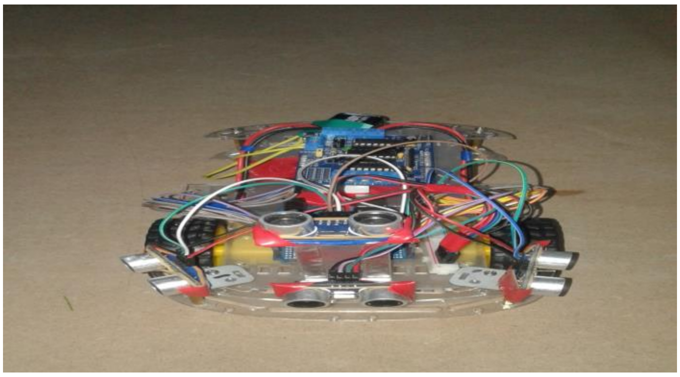


Figure (8): The final design of the collision avoidance robot.

Discussion of the results.

In this paper, the robot was tested on its designated path, as shown in the figure (9).

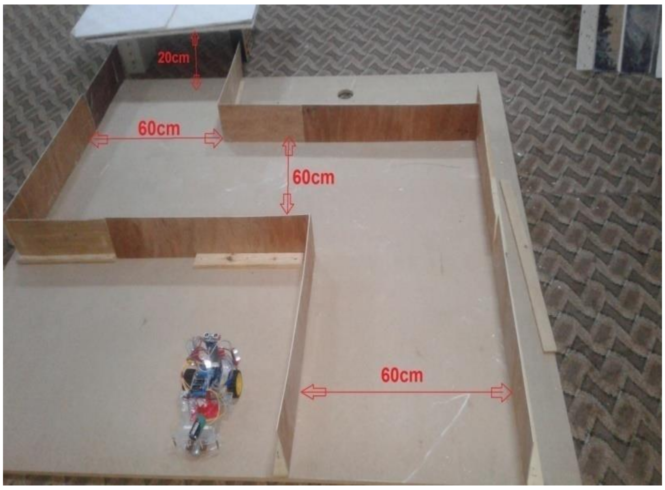


Figure (9): The robot's operating path.

The robot was tested in its designated work area, and the results were as follows: -

When the robot was started from the starting point, it moved forward approximately 90cm, meaning the controller executed the fourth condition in the program, which was explained earlier. Figure (10) shows the robot at the starting point and when it moved forward.

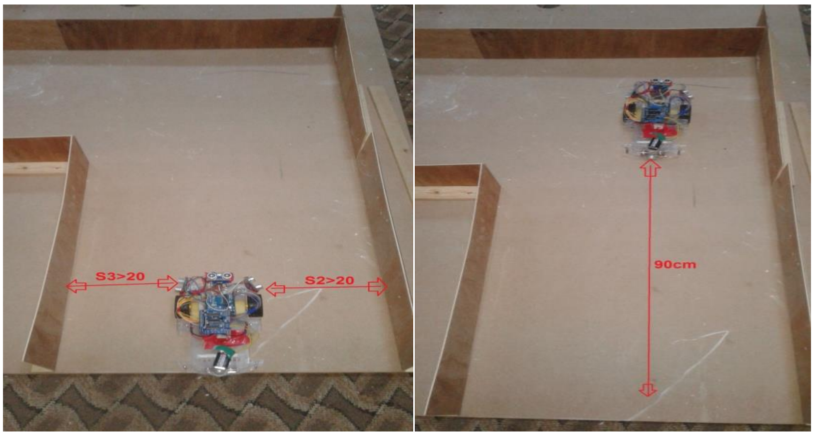


Figure (10): The robot when it was at the starting point and when it moved forward.

While the robot was moving and approaching the wall, it changed its direction to the left, meaning that the controller executed the first condition in the program which was explained earlier. Figure (11) shows the robot with the wall in front of it, and when it changed its direction to the left.

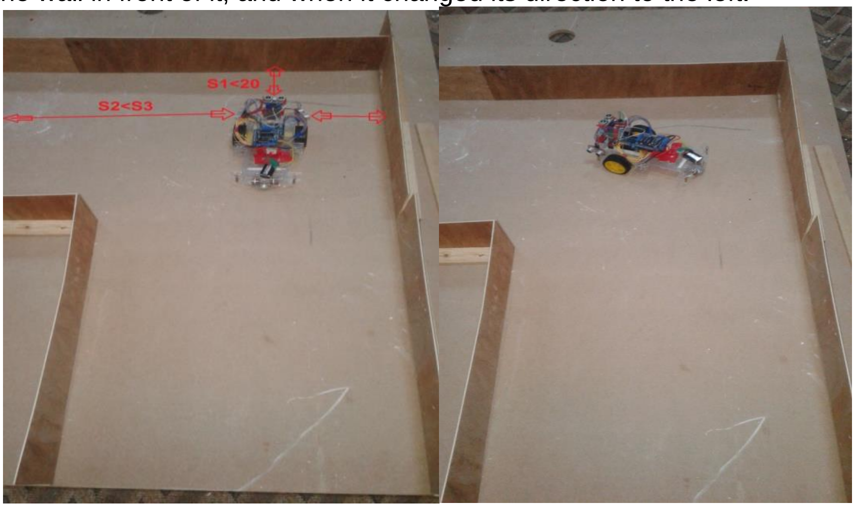


Figure (11): The robot with a wall in front of it, and when it changed its direction of movement to the left.

The robot moved forward approximately 15cm and approached another wall on the left side. The robot then redirected itself, moved away from the wall, and continued its path. This means that the controller executed the second condition, which was explained earlier. Figure (12) shows the robot approaching the wall on the left side and changing its direction of movement.

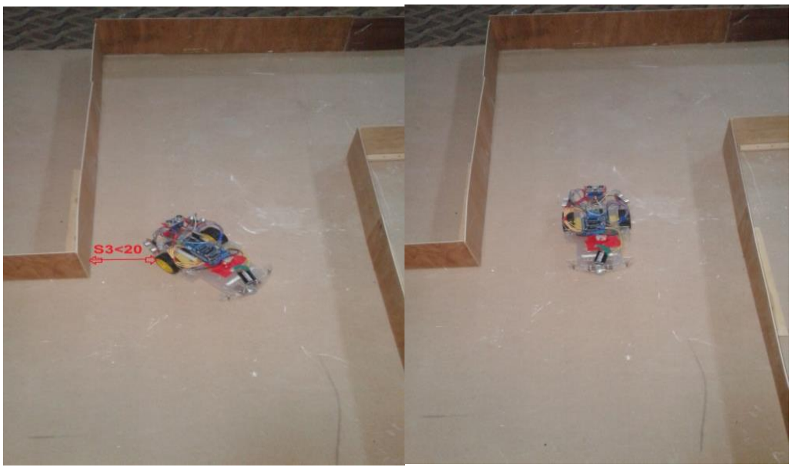


Figure (12): The robot approaching the wall from the left side, and when it changed the direction of its movement.

While the robot was following its path, it approached another wall in front of it. The robot then changed its direction to the right, meaning the controller executed the second condition. Figure (13) shows the robot in front of the wall, and when it changed its direction to the right.

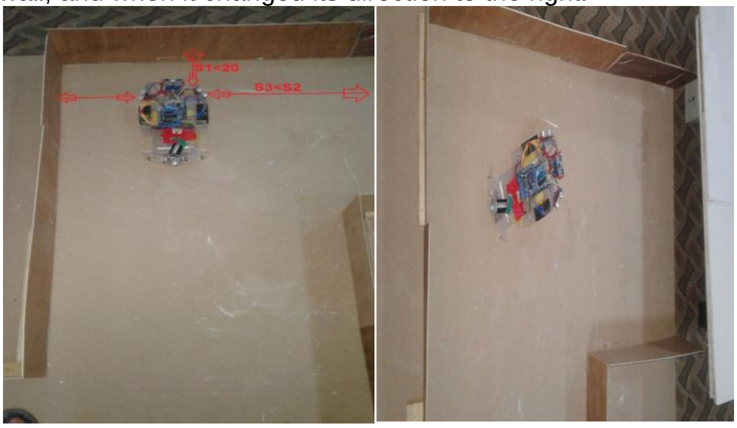


Figure (13): The robot with a wall in front of it, and when it changed its direction of movement to the right.

When the robot approached the endpoint, it veered to the left and got closer to the wall, then changed its direction to the right, meaning the controller executed the second condition. However, while changing its direction to the right, the robot moved towards the wall. Figure (14) shows the robot approaching the wall from the left side and then from the right side.

Figure (14): The robot approaching the wall from the left side and then from the right side.

The robot changed its direction to the left, meaning the controller executed the first condition. Then, the robot moved forward, meaning the controller executed the fourth condition. While the robot was moving forward, it stopped working, meaning the controller executed the third condition, which was explained earlier. Thus, the robot reached the endpoint, and Figure (15) illustrates the robot's movement towards the endpoint and when it stopped working.

Figure (15): The robot moving towards the endpoint and stopping.

Conclusion

In this paper, the following has been designed and implemented:

1. Design and implementation of a collision avoidance robot control system using an Arduino UNO board. Design and implementation of a collision avoidance robot control system using an Arduino UNO board.
2. Building an algorithm to program the microcontroller (Arduino UNO) to control the robot within the work area. Building an algorithm to program the microcontroller (Arduino UNO) to control the robot within the workspace.
3. Achieving the best results. Getting the best results.

Reference

[1] P. Wang, T. Ouyang, S. Zhao, X. Wang, Z. Ni, and Y. Fan, "Intelligent vehicle formation system based on information interaction," *World Electric Veh. J.*, vol. 15, no. 6, p. 252, 2024.

[2] N. Das, A. Bhuyan, P. P. Sonowal, A. Lama, and N. Boruah, "Design and Implementation of Arduino based low cost Autonomous Car," in *2024 6th International Conference on Energy, Power and Environment (ICEPE)*, 2024, vol. 83, pp. 1–6.

[3] S. N. T. Yasin, M. A. M. Tahir, and I. Hussaini, "Autonomous navigation: An Arduino-based smart car obstacle avoidance system," *Politeknik & Kolej Komuniti Journal of Engineering and Technology*, vol. 9, no. 1, pp. 53–62, 2024.

[4] A. A. Ahmed, A. Alsharif, T. Triwiyanto, M. Khaleel, C. W. Tan, and R. Ayop, "Using of neural network-based controller to obtain the effect of hub motors weight on electric vehicle ride comfort," in *2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 2022.

[5] B. Poornima and P. L. S. Kumari, "A dual-step approach for implementing smart AVS in cars," *Eng. Technol. Appl. Sci. Res.*, vol. 14, no. 4, pp. 15771–15778, 2024.

[6] S. N. Reddy, A. Samuel, G. S. Teja, and K. Bindu, "Voice-driven intelligence on wheels: A speech-based smart vehicle powered by IoT," in *2024 International Conference on Inventive Computation Technologies (ICICT)*, 2024.

[7] M. Abduljawwad, M. Khaleel, T. S. Ogedengbe, and S. Abraheem, "Sensors for daily utilization," *Int. J. Electr. Eng. and Sustain.*, pp. 106–119, 2023.

[8] M. A. Islam, M. F. Abdullah Sayeedi, J. F. Deepti, S. R. Bappy, S. S. Islam, and F. Hafiz, "MosquitoMiner: A light weight rover for detecting and eliminating mosquito breeding sites," in *2024 IEEE Region 10 Symposium (TENSYMP)*, 2024, pp. 1–6.

[9] A. Shaik, S. Devunuri, M. Venkata Ramarao, and B. Harish Babu, "Development of low-cost forward collision warning and avoidance system for electric microcar," in *2024 International Conference on Computational Intelligence for Green and Sustainable Technologies (ICCIGST)*, 2024, pp. 1–6.

[10] M. Belrzaeg, A. A. Ahmed, M. M. Khaleel, A. Alsharif, M. M. Rahmah, and A. S. D. Alarga, "Suspension system control process for buses with in-wheel motors," in *ICCEIS 2022*, 2023, vol. 29, p. 4.

[11] A. Ghayth, M. Şimşir, M. Khaleel, A. Ahmed, and A. Alsharif, "An investigation of Inverse-Automatic Mechanical Transmission of EV using gear downshift approach," *Int. J. Electr. Eng. and Sustain.*, pp. 1–9, 2023.

[12] A. S. D. Alarga, H. S. Hawedi, I. Imbayah, A. A. Ahmed, A. Alsharif, and M. M. Khaleel, "A simple Design of Automatic Bag Valve Mask Ventilator using arduino," in *2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 2023.

[13] M. Belrzaeg, A. A. Ahmed, A. Q. Almabrouk, M. M. Khaleel, A. A. Ahmed, and M. Almukhtar, "Vehicle dynamics and tire models: An overview," *World J. Adv. Res. Rev.*, vol. 12, no. 1, pp. 331–348, 2021.

[14] A. Eltaleb *et al.*, "Smart home sensor systems: Advancements and applications," *International Journal of Electrical Engineering and Sustainability (IJEES)*, vol. 1, no. 3, pp. 120–228, 2023.

[15] I. Imbayah, A. Alsharif, M. Khaleel, A. A. Ahmed, and H. J. El-Khozondare, "Studying the Possibility of Smart Farms based on solar System Using (IoT) Technology in Libya," *Sebha University Conference Proceedings*, vol. 3, no. 2, pp. 97–102, 2024.