MINDBEND 2024

PATHFINDER

Team Name : Night Crawlers

Team Members: (i)Tiwari Om Sureshkumar

(ii) Dhanish Safaya

(iii) Kamlesh

(iv)Yogesh Suthar

Introduction :

In a world where technology meets innovation, the Pathfinder emerges as a beacon of exploration and discovery. Powered by Arduino, this remarkable robot represents the culmination of cutting-edge engineering and limitless curiosity. Designed to navigate complex terrains, overcome obstacles, and chart unknown territories, the Pathfinder embodies the spirit of adventure and scientific inquiry.

Equipped with state-of-the-art sensors, precise motor controls, and intelligent decision-making algorithms, the Pathfinder is more than just a machine; it's a trusted companion on your journey of exploration. Whether you're exploring the depths of a cavernous maze or mapping the uncharted expanses of distant planets, the Pathfinder is your steadfast guide, always ready to brave the unknown alongside you.

Specifications:

(i) Hardware Components:

(a)Arduino microcontroller board {Arduino Uno}.

(b) Motor driver module (here, L298N) to control the motors.

(c) HC –SR04 Ultrasonic sensors for measuring distances

(d) Two BO Motors of 150 rpm.

(e) Two Wheels and one Caster Wheel for locomotion.

(f) Chassis to hold the components together.

(g) Power source (here, two 18650 Li –ion batteries) for Arduino & Motors

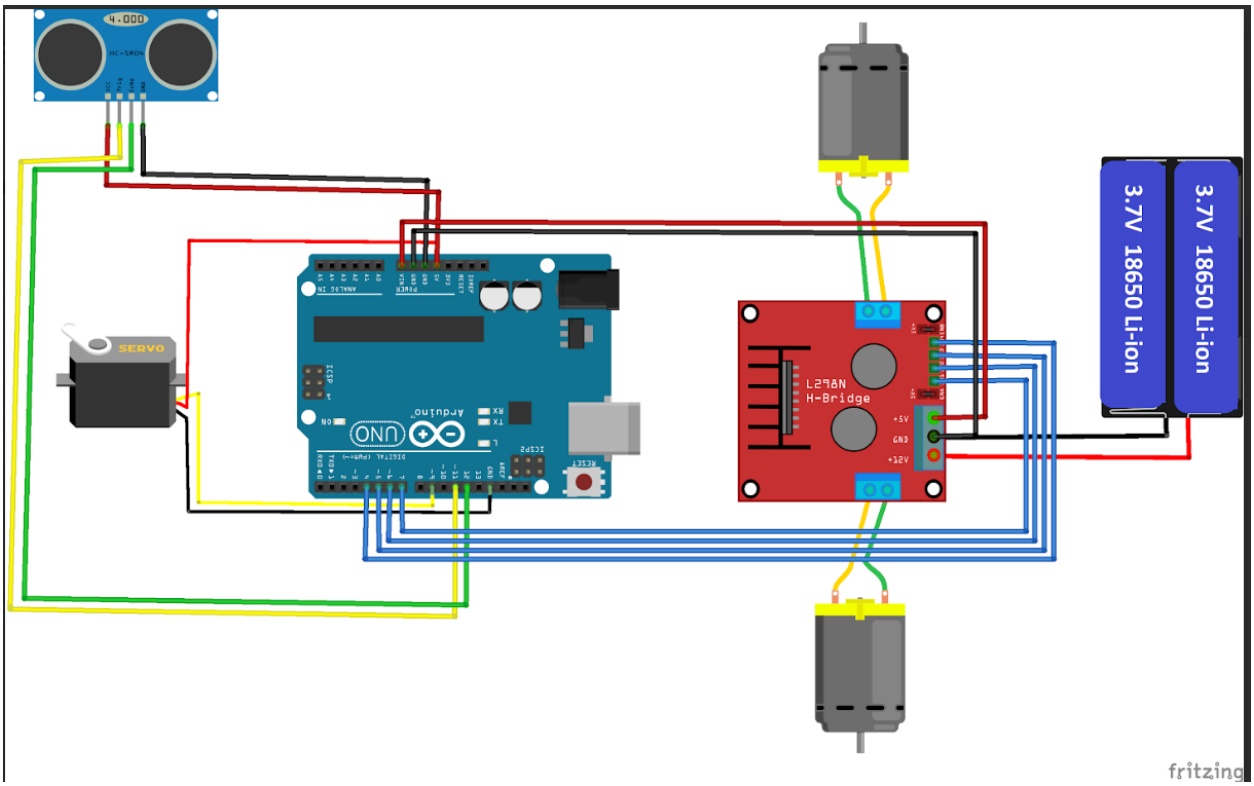
.

Components:

Sr. No	Name of the components	Quantity
01	Arduino Uno	01
02	BO Motors	02
03	Ultrasonic Sensor	01
04	L298N Motor Driver	01
05	Chassis	01
06	Li –ion batteries (3.7 Volts)	03
07	Wheels	02
08	Caster Wheel	01
09	Jumper Wires and Single Stranded Wires	NA
10	3mm Nut Bolts	07
11	Double Sided Tape	01
12	Insulation Tape	01
13	Servo motor	01
14	Screwdriver	01

Circuit Diagram :

The following Circuit Diagram represents the the manner and order of interconnections we used for the proper functioning of the Arduino Pathfinder Robot.



Algorithms :

This algorithm is designed to enable the pathfinder to work efficiently .The code implements a basic obstacle avoidance algorithm for the Pathfinder robot using ultrasonic sensors. Here's a breakdown of the algorithm:

1. Setup:

- The code initializes the pins for ultrasonic sensor connections, motor control pins, and servo motor control.

- The servo motor is set to the center position (90 degrees) to start with.

2. Loop:

- The main loop of the program continuously runs.

- It first checks the distance in front of the robot using the ping() function, which measures the distance using the ultrasonic sensor.

- If the distance in front of the robot is greater than a predefined threshold (maxLowDistance), the robot moves forward (front() function) and then waits for a short delay before checking again.

- If the distance in front of the robot is less than or equal to the threshold, indicating an obstacle, the robot stops ('Break()' function) and proceeds to check the distances to the right and left using the 'get_Distance()' function.

- Depending on which direction has a clearer path (greater distance), the robot turns right ('right()' function), left ('left()' function), or backs up ('back()' function) to avoid the obstacle.

3. Movement Functions:

- 'front()': Moves the robot forward by activating both left and right motors in the forward direction.

- 'back()': Moves the robot backward by activating both left and right motors in the reverse direction.

- 'left()': Turns the robot left by activating the left motor in reverse and the right motor forward.

- 'right()': Turns the robot right by activating the left motor forward and the right motor in reverse.

- 'Break()': Stops both left and right motors.

4. Servo Control:

- The ‘get_Distance()’ function rotates the ultrasonic sensor mounted on the servo motor to scan distances in multiple directions (front, right, and left) before returning it to the center position.

5. Ultrasonic Distance Measurement:

- The ‘ping()’ function sends a pulse through the trigPin to trigger the ultrasonic sensor.

- It then measures the time taken for the pulse to bounce back (echo) from an obstacle using the pulseIn() function.

- The distance to the obstacle is calculated based on the time taken for the pulse to travel and the speed of sound in air.

Overall, this algorithm allows the Pathfinder robot to autonomously navigate its environment by avoiding obstacles in its path using feedback from ultrasonic sensors.