|  |  |  |
| --- | --- | --- |
| Group No. | Group Members (Regd. No.) | Project Title |
| *15* | *2041013129* | Web-Enabled Raspberry Pi Pico Car: Building a Remote-Controlled Robot with Web Server Integration |
| *2041018027* |

1. **Introduction:**

Obstacle avoidance Robot is designed to navigate the robot in unknown environment by avoiding collisions [1]. Obstacle avoiding robot senses obstacles in the path, avoid it and resumes its running. There are some very popular methods for robot navigation like wall following, edge detection, line following and many more. A more general and commonly employed method for obstacle avoidance is based on edge detection [2-3]. A disadvantage with obstacle avoidance based on edge detecting is the need of the robot to stop in front of an obstacle to provide a more accurate measurement. All mobile robots feature collision avoidance, ranging from primitive algorithms that detect an obstacle and stop the robot to avoid a collision, using some sophisticated algorithms that enable the robot to detour obstacles. The latter algorithms are more complex since they involve detection of an obstacle as well as quantitative measurements concerning the obstacle's dimensions. Once these have been determined, the obstacle avoidance algorithm needs to steer the robot around the obstacle and resume motion toward the original target. The steering algorithm ensures that the robot does not have to stop in front of an obstacle during its navigation. An ultrasonic sensor is used to detect any obstacle ahead of it and sends a command to the micro- controller. Obstacle avoidance in robots can bring more flexibility in manoeuvring in varying environments and would be much more efficient as continuous human monitoring is not required. This project developed an obstacle avoiding robot which can move without any collision by sensing obstacles on its course with the help of three ultrasonic distance sensors. Robots guided with this technology can be put into diversified uses, e.g., surveying landscapes, driverless vehicles, autonomous cleaning, automated lawn mower and supervising robot in industries.

1. **Problem identification and Problem Formulation:**

**Problem Identification:**

1. **Limited Control Range:** Traditional remote-controlled robots have a limited range based on the communication technology used. This can be a constraint, especially in scenarios where a wider range is desired.
2. **Dependency on Dedicated Remote-Control Devices:** Remote-controlled robots typically rely on specialized remote-control devices. This limitation can be addressed by leveraging web-enabled control, allowing users to use any device with a web browser.
3. **Lack of Accessibility:** Remote-controlled robots are often designed for specific remote controls, making them less accessible for users who do not own the dedicated remote. A web-enabled solution can enhance accessibility for a broader audience.
4. **Limited Interaction and Feedback:** Conventional remote-controlled robots may lack features for real-time interaction and feedback. Integrating a web server allows for two-way communication, enabling users to receive data from the robot and provide input dynamically.
5. **Integration Challenges:** Integrating web server functionality with a Raspberry Pi Pico-powered robot may pose challenges due to the need for coding skills in both hardware and web development.

**Problem Formulation:**

The problem can be formulated as follows:

**Objective:** Design and implement a web-enabled Raspberry Pi Pico car that allows remote control through a web interface, addressing the limitations of traditional remote-controlled robots.

1. **Objective of the Project:**
2. **Hardware Integration:**

* Assemble a robotic car using a Raspberry Pi Pico W as the main controller.
* Integrate motors, wheels, and sensors for movement and obstacle detection.
* Connect and configure any additional hardware components required for the project.

1. **Wireless Communication:**

* Implement wireless communication between the Raspberry Pi Pico W and a remote device (such as a computer or smartphone).
* Utilize Wi-Fi capabilities of the Raspberry Pi Pico W for communication.

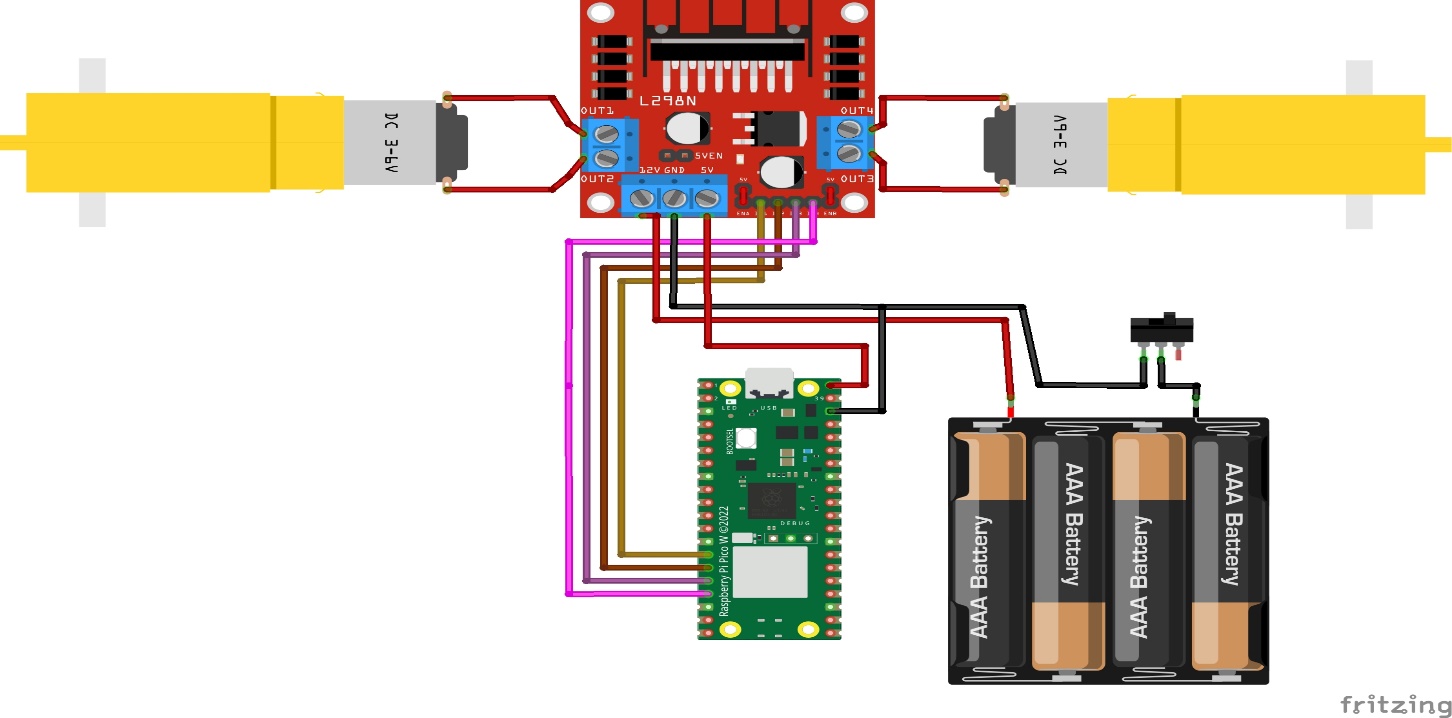
1. **Remote Control Functionality:**

* Implement a mechanism for remote control through the web interface.
* Enable users to control the movement of the robot (forward, backward, left, right) using buttons or other interactive elements on the web page.

1. **Testing and Optimization:**

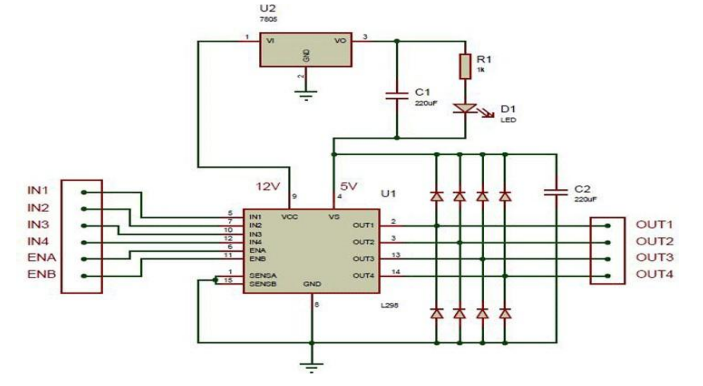
* Conduct thorough testing of the entire system, including the hardware and software components.
* Optimize the code and system performance for a smoother and more responsive user experience.

1. **Block Diagram of the Project:**

****

**Fig.1 Block Diagram of Web-Enabled Raspberry Pi Pico Car**

1. **Circuit Diagram of the project:**

****

**Fig.2 Circuit Diagram of Web-Enabled Raspberry Pi Pico Car**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Name of the Components | Specification | Quantity |
| 1 | **Raspberry Pi Pico W** | **125 MHz** | **1** |
| 2 | **Motor Driver L298N** | **-** | **1** |
| 3 | **DC Motor** | **150 RPM** | **2** |
| 4 | **Jumper Wires** | **-** | **15** |

1. **Components/Items Required:**

**Full Signature of Group members:**

**1. Sayon Mitra**

**2. Sidhartha Raj**

**Signature of Corresponding Faculty**

1. **References: -**
2. Safaric, R., Hedrih, I., Klobucar, R., & Sorgo, B. (2003, December). Remote controlled robot arm. In *IEEE International Conference on Industrial Technology, 2003* (Vol. 2, pp. 1202-1207). IEEE.
3. Sobh, T., Mihali, R., Ghimire, B., Vovk, K., Gosine, G., Batra, P., ... & Rosca, A. (2001). Case Studies in Web-Controlled Devices and Remote Manipulation. In *Proceedings of the International Workshop on “Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications,” IDAACS*.