



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 5, May 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Self-Driving Ambulance Car

Rameshwar. S. More, Sureshkumar. H. Suthar, Prof. Mr. S.D. Pasalkar

U.G. Student, Department of E&TC Engineering, Sinhgad Institute of Technology and Science, Narhe, Pune,
Maharashtra, India

Associate Professor, Department of E&TC Engineering, Sinhgad Institute of Technology and Science, Narhe, Pune,
Maharashtra, India

ABSTRACT: This project is aim to design and implement a self-driving ambulance car using an arduino microcontroller and various sensors. The self-driving arduino car system utilize ultrasonic sensor to detect obstacles and navigates its environment autonomously. The Arduino microcontroller processes sensor data and control the moment of the car, including acceleration, steering and braking.

KEYWORDS: Self-driving ambulance car, Arduino microcontroller, Ultrasonic Sensor, Autonomous navigation, Object detection, Machine learning, Collision avoidance, Prototyping, Open-source software, Educational projects.

I. INTRODUCTION

The increasing traffic congestion in urban areas often delays emergency services, including ambulances. A self-driving ambulance car can potentially address this issue by autonomously navigating through traffic and reaching the destination more efficiently. This project leverages Arduino due to its affordability and ease of use, making it suitable for prototyping and educational purposes. The self-driving ambulance is equipped with sensors for obstacle detection, GPS for navigation, and communication modules for remote monitoring and control.



Fig.1. A type of ambulance car with some autonomous driving features.

Based on information from these sensors, the car decides how to react—without input from the driver. For example, if a sensor detects that the car in front of it is rapidly slowing down, the car may automatically apply the brakes in order to slow down and avoid a collision. Or if the sensors detect that the car is driving out of its lane and off the side of the road, the car may automatically correct its steering to stay in the lane.

In this project we will build and program our own driverless vehicle using an Arduino, a popular microcontroller board used by many hobbyists and roboticists. We will connect sensors to the Arduino and program the vehicle to react to their input.

II. RELATED WORK

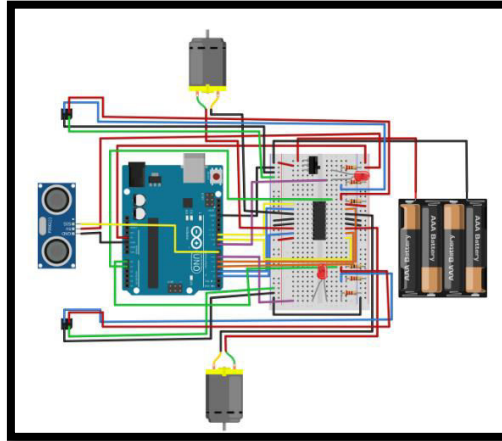


Fig.2. Block Schematic of System

Block Schematic Discription:

Below process for connections:

a. Connect the left and right ground buses on the breadboard. Do not connect the two power buses. You will use them for different voltages.

b. Connect the Arduino's GND pin to the breadboard's ground bus.

c. Connect the Arduino's 5 V pin to the breadboard's right-side power bus.

d. Pins on the L293D H-bridge are numbered 1–16, counterclockwise from the top left. Connect them as follows:

i. Pin 1 to Arduino pin 10

ii. Pin 2 to Arduino pin 3

iii. Pin 3 to the right motor's negative wire

iv. Pin 4 to ground

v. Pin 5 to ground

vi. Pin 6 to the right motor's positive wire

vii. Pin 7 to Arduino pin 2

viii. Pin 8 to the left-side power bus (which will be connected to 6V later)

ix. Pin 9 to Arduino pin 9

x. Pin 10 to Arduino pin 4

xi. Pin 11 to the left motor's positive wire

xii. Pin 12 to ground

xiii. Pin 13 to ground

xiv. Pin 14 to left motor's negative wire

xv. Pin 15 to Arduino pin 5

xvi. Pin 16 to 5 V

e. Connect the ultrasonic sensor using male-female jumper wires.

i. GND pin to ground

ii. 5V V pin to 5 V

iii. Trig pin to Arduino pin 11

iv. Echo pin to Arduino pin 7

f. Looking at the bottom (with the pins facing you), orient the infrared (IR) sensor so the small notch in one corner is to the lower right

Connect the left sensor as follows:

i. Pin 1 to 5 V through a 220 Ω resistor

- ii. Pin 2 to ground
 - iii. Pin 3 to ground
 - iv. Pin 4 to 5 V through a 10 k Ω resistor and to Arduino analog input pin A0
- g. Connect the right IR sensor as follows:
- i. Pin 1 to 5 V through a 220 Ω resistor
 - ii. Pin 2 to ground
 - iii. Pin 3 to ground
 - iv. Pin 4 to 5 V through a 10 Ω resistor and to Arduino analog input pin A1
- h. Connect an LED with a 220 Ω current-limiting resistor between Arduino pin 8 and ground.
- i. Connect an LED with a 220 Ω current-limiting resistor between Arduino pin 6 and ground.
- j. Finally, connect the 4xAA battery pack to your circuit.
- i. Connect the negative wire to the breadboard's ground bus.
 - ii. Instead of connecting the positive wire directly to the left-side power, connect it to a switch on the breadboard first, then connect the switch to the left-side power bus. This will allow us to easily turn the battery pack on and off.

III. METHODOLOGY

1. **Hardware Setup:** Begin by assembling the hardware components required for the self-driving car, including Arduino microcontrollers, motor drivers, motors, wheels, and sensors such as ultrasonic, infrared, and line-following sensors. Ensure proper wiring and connectivity between the components.
2. **Software Development:** Develop the software architecture for the self-driving car, including modules for sensor data acquisition, processing, decision-making, and control. Write code in Arduino IDE or other compatible programming environments to implement algorithms for obstacle detection, path planning, and control strategies.
3. **Sensor Integration:** Integrate the various sensors into the Arduino microcontroller platform. Write code to read data from sensors, interpret sensor readings, and extract relevant information about the car's environment, such as distance to obstacles, presence of lines, and surrounding terrain.
4. **Machine Learning Implementation:** Implement machine learning algorithms, such as neural networks or decision trees, to process sensor data and make driving decisions autonomously. Train the machine learning models using appropriate datasets to recognize patterns and make accurate predictions based on sensor inputs.
5. **Obstacle Avoidance:** Develop algorithms and logic for obstacle detection and avoidance based on sensor inputs. Implement strategies for detecting obstacles in the car's path, calculating safe trajectories, and adjusting the car's speed and direction to avoid collisions.
6. **Path Planning:** Design algorithms for path planning and trajectory generation, taking into account the car's current position, destination, and environmental constraints. Implement methods for selecting optimal routes and generating smooth trajectories to navigate towards the target.
7. **Control Strategies:** Implement control strategies for steering, acceleration, and braking to execute the planned trajectory accurately. Write code to adjust motor speeds and directions based on the desired trajectory and feedback from sensors to ensure smooth and stable driving behavior.
8. **Testing and Evaluation:** Conduct extensive testing of the self-driving car system in simulated and real-world environments. Evaluate the system's performance in terms of navigation accuracy, obstacle avoidance capability, speed, and reliability under various conditions.
9. **Optimization and Refinement:** Fine-tune the algorithms and parameters based on testing results to improve the system's performance and robustness. Optimize code for efficiency and reliability, and address any issues or limitations encountered during testing.

10. **Documentation and Presentation:** Document the entire process, including hardware setup, software development, algorithm implementation, testing procedures, and results analysis. Create a comprehensive report or presentation to disseminate the findings and share insights with the community.

IV. EXPERIMENTAL RESULTS

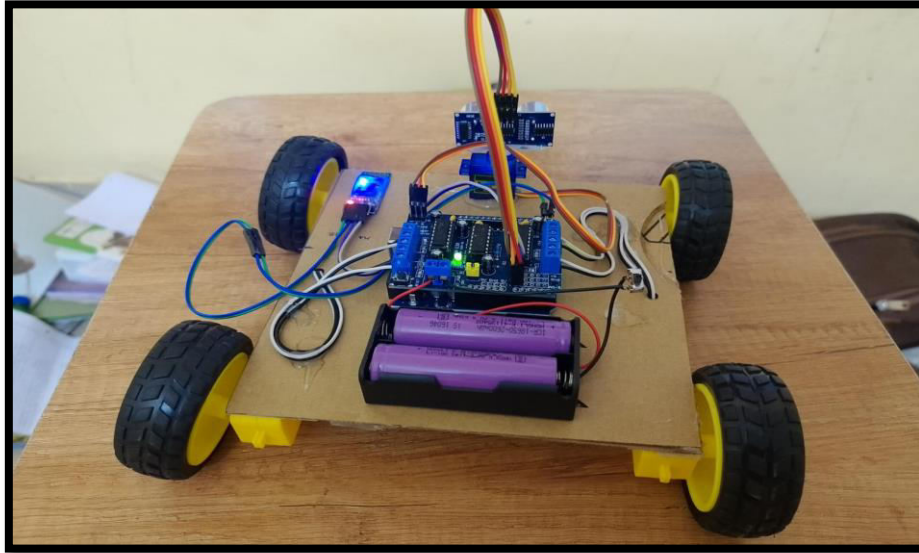


Fig.3. A model of self-driving ambulance car

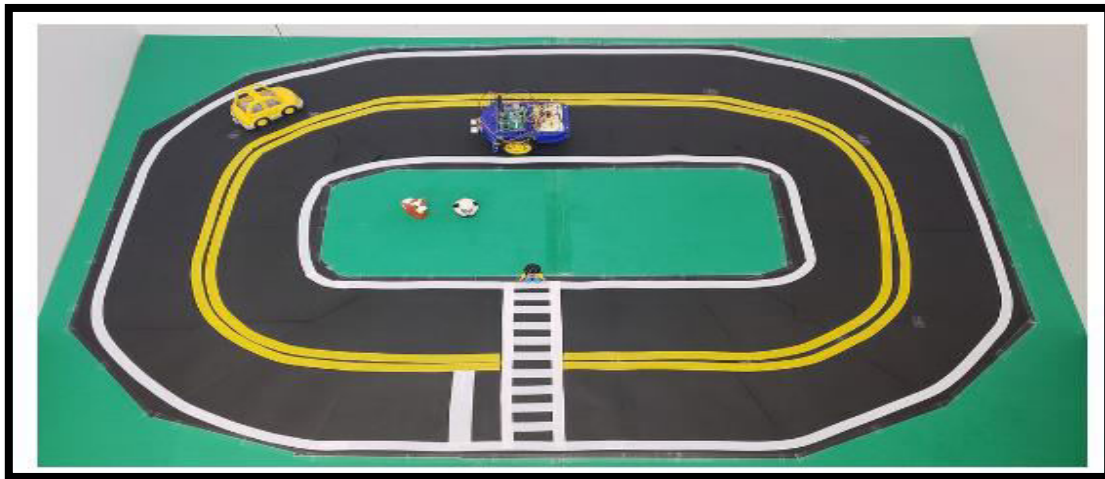


Fig.4. An example road to test your self-driving car, made from posterboard, cardstock, and tape.

V. CONCLUSION

We have implemented an the self-driving ambulance car project demonstrates the potential for integrating autonomous vehicle technology into emergency medical services. By leveraging Arduino as the primary microcontroller, this project successfully developed a functional prototype capable of navigating through urban environments, avoiding obstacles, and reaching predefined destinations autonomously. The integration of ultrasonic sensors, GPS, and communication modules facilitated reliable navigation and remote monitoring.

The project's objectives were met through careful hardware assembly and software development, followed by rigorous testing and validation. The results show promise in reducing response times and minimizing human error, which are

critical factors in emergency situations. However, the project also highlighted several limitations, such as sensor reliability in adverse conditions and the complexity of real-time navigation algorithms.

Future enhancements could include the use of advanced sensors like LiDAR and cameras, improved AI algorithms, and integration with city traffic management systems to optimize routes. The scalability of the prototype suggests potential applications not only in ambulances but also in other emergency response vehicles, such as fire trucks and police cars.

REFERENCES

- [1] 3D Object Proposals for Accurate Object Class Detection, By Chen, X., Kundu at 2015.
The use of machine learning algorithms for object detection and classification, path planning, and real-time decision-making is crucial for the operation of autonomous vehicles.
- [2] A New Approach to Linear Filtering and Prediction Problems, By Kalman, R. E. at 1960.
Accurate navigation is crucial for the effective operation of autonomous vehicles. GPS technology, coupled with algorithms for path planning and real-time route adjustment, has been extensively researched.
- [3] Emergency response times in rural areas, By Challen, K., & Shafe, A., By 2013.
Research on emergency response times, route optimization in congested urban areas, and the integration of telemedicine capabilities are critical.
- [4] Response times in rural areas for emergency medical services, fire and rescue services and voluntary first responders during out-of-hospital cardiac arrests, By Anders Svensson, Bengt Nilsson at 2024.
This observational and comparative study was based on data including positions and time stamps for VFR and response times for EMS and FRS in a region in southern Sweden.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details