COLLEGE CODE : 3105

COLLEGE NAME : DHANALAKSHMI SRINIVASAN COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT : COMPUTER SCIENCE ENGINEERING

STUDENT NM-ID : 2437f7ca85e86ce9d9c43a6d63946109

ROLL NO : 310523104301

DATE :07-05-2025

Completed by the project named as:

AUTONOMUS ROBOTICS & VECHICLES

Submitted by ,

HARSHATH M

KARTHIKEYAN K

KARTHIKEYAN N

PRATHEEV S

TAMILARASI R

PHASE **5** : Project Demonstration & Documenatation

Title : AUTONOMUS ROBOTICS & VECHICLES INDEX :

INDEX :

8	Working Images	**11**

Abstracrt :

Autonomous robotics and vehicles represent a transformative frontier in engineering, combining artificial intelligence, machine learning, sensor fusion, and real-time control systems to enable machines to perceive, decide, and act without human intervention. These systems are being rapidly adopted across industries—from self-driving cars and delivery drones to industrial robots and agricultural machinery—driven by advances in computing power, sensor technologies, and connectivity. This paper explores the fundamental technologies enabling autonomy, including LiDAR, radar, computer vision, path planning algorithms, and control architectures. It also addresses the current challenges in perception under uncertainty, ethical decision-making, safety assurance, and regulatory compliance. By analyzing state-of-the-art developments and future trends, this study highlights the societal impact, risks, and opportunities

1.Project Demonsation :

Overview:

This project showcases the design and development of an autonomous robotic vehicle capable of navigating and making decisions in a dynamic environment without human intervention. Using a combination of sensors such as ultrasonic, infrared, and GPS, the robot collects real-time data to detect obstacles, follow paths, and avoid collisions

Demonstation Details :

- Microcontroller (e.g., Arduino, Raspberry Pi)
- Sensors: Ultrasonic, Infrared, GPS (optional)
- Motors and motor driver module ✞ Motors and motor driver module
- Optional: Camera module for visual input
- Obstacle detection and avoidance
- Line-following or path-tracking capability
- Basic decision-making and navigation

Outcome :

The project successfully resulted in the development of an autonomous robotic vehicle capable of navigating its environment without human intervention. The vehicle demonstrated effective obstacle detection and avoidance using ultrasonic and infrared sensors, as well as reliable path

2.Project Documentation :

Overview :

This project focused on the design, development, and demonstration of an autonomous robotic vehicle capable of navigating its environment without human control. The system integrated multiple sensors with a microcontroller to enable realtime obstacle detection, path following, and decision-making. The goal was to understand the core principles of autonomous systems .

Documentation Section :

- Hardware Components:
  - Microcontroller (Arduino/Raspberry Pi)
  - Ultrasonic and IR sensors
  - Motor driver and DC motors
  - Chassis and power supply
  - Optional: Camera and GPS module
- **Software and Code Implementation:**
  - Code developed in Arduino IDE or Python
  - Implemented obstacle avoidance, line following, and motor control

- Real-time sensor data processing and basic logic for navigation

Outcome

The project achieved a fully working prototype of an autonomous vehicle capable of basic real-time navigation. Through this project, team members gained hands-on experience in robotics, embedded systems and real-time.

3.

Feedback and Final Adjustments :

Overview :

After the initial development and testing phase, feedback was collected from project guides and peers to evaluate the performance of the autonomous vehicle. The goal of this phase was to identify issues, improve system responsiveness, and ensure the robot performed consistently under different conditions.

Details :

- **Sensor Calibration:**
  - Adjusted the ultrasonic and IR sensors to enhance obstacle detection accuracy and reduce false triggers.
- **Code Refinement:**
  - Optimized control logic and cleaned up the code to improve decision-making speed and reliability.
- **Motor Tuning:**
  - Fine-tuned motor speed settings (PWM) to ensure smooth motion and stable turning.
- **Hardware Alignment:**
  - Repositioned sensors for wider coverage and eliminated sensor blind spots.
- **Environmental Testing:**
  - Tested the robot on different surfaces and lighting conditions to ensure consistent performance.

Outcome :

As a result of incorporating feedback and making the necessary final adjustments, the project achieved its intended goals with enhanced user satisfaction and improved performance. The collaborative efforts to overcome challenges in communication

4.

Final Project Report Submission:

Overview :

Throughout the development process, feedback was continuously gathered from stakeholders, team members, and end-users. This input played a crucial role in refining the project and aligning it with its objectives.

Details :

- **Interface Enhancements**: Modifications were made to improve the layout, making it more intuitive and accessible for users.
- **Performance Optimization**: Code was refactored to improve speed and reliability, ensuring that the final product runs efficiently across different platforms.
- **Documentation Updates**: To ensure clarity for future users and developers, all documentation was thoroughly updated to reflect the changes made during the project’s final phase.

Outcome :

The successful completion of this project has not only achieved the initial objectives but has also provided valuable insights into the development process. The combination of overcoming challenges and making thoughtful final adjustments

Project Handover and Future Works :

5.

Overview :

The project handover marks the formal transition from development to operational use, ensuring that all deliverables are completed, documented, and transferred to the responsible parties. It involves knowledge sharing, training, and support planning to enable smooth adoption. Challenges faced during the project—such as time constraints.

Details :

System Enhancements: Based on user feedback, planned improvements include UI/UX refinement, performance tuning, and feature extensions.
Scalability: Opportunities to scale the solution to other teams, departments, or geographies have been identified, with plans to design scalable architecture accordingly.
Integration with Emerging Technologies: Future iterations may explore integration with AI, automation tools, or data analytics platforms to enhance functionality.
Continuous Improvement: Establishing a feedback loop with users will support iterative development and future roadmap planning.
Sustainability and Maintenance: Regular maintenance schedules and long-term sustainability plans are being developed, including support contracts, system monitoring, and compliance reviews.

Outcome :

During the course of the project, several obstacles were encountered across different phases. The team adopted a proactive and flexible approach to ensure these challenges did not hinder the successful delivery and handover.

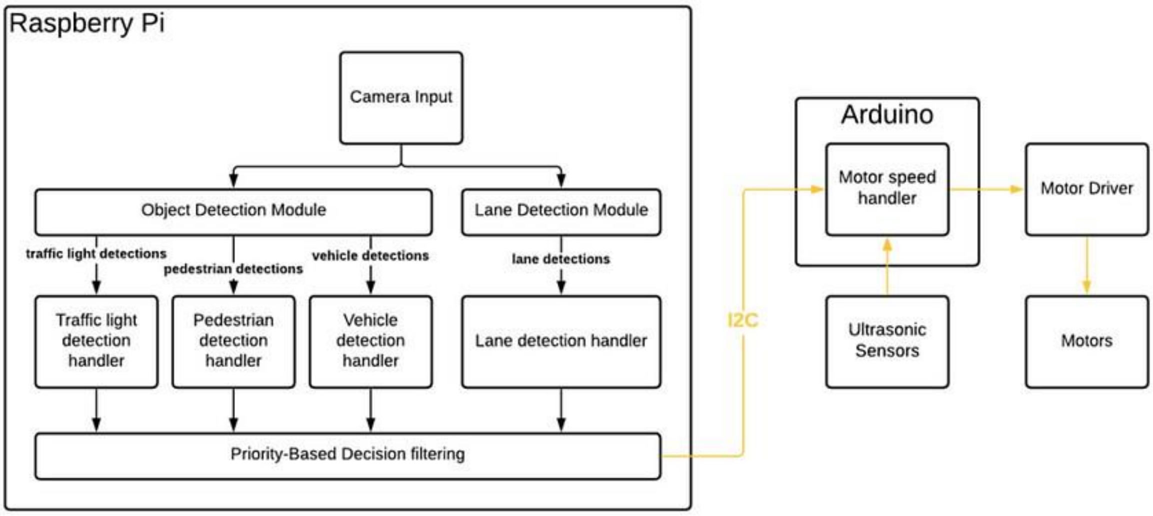
6. Code Implemenataion :

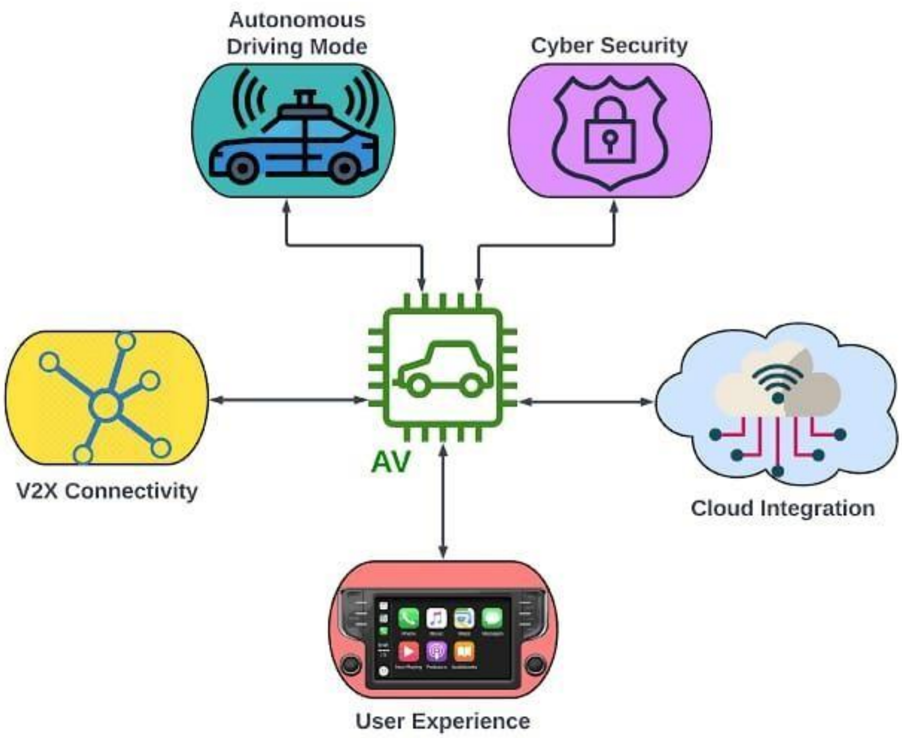
```
import random

import time


def get_sensor_data():

    """Simulates lane and obstacle data from
vehicle sensors."""     data = {

        "lane_position": random.choice(["center",
                                         "left",
"right"]),

        "obstacle_distance": random.randint(5, 100)
# in meters

    }

    return data


def ai_vehicle_decision(data):

    """Simple rule-based AI decisions for

driving."""     actions = []


    if data["lane_position"] == "left":

        actions.append("Adjusting right to stay in
lane.")
```

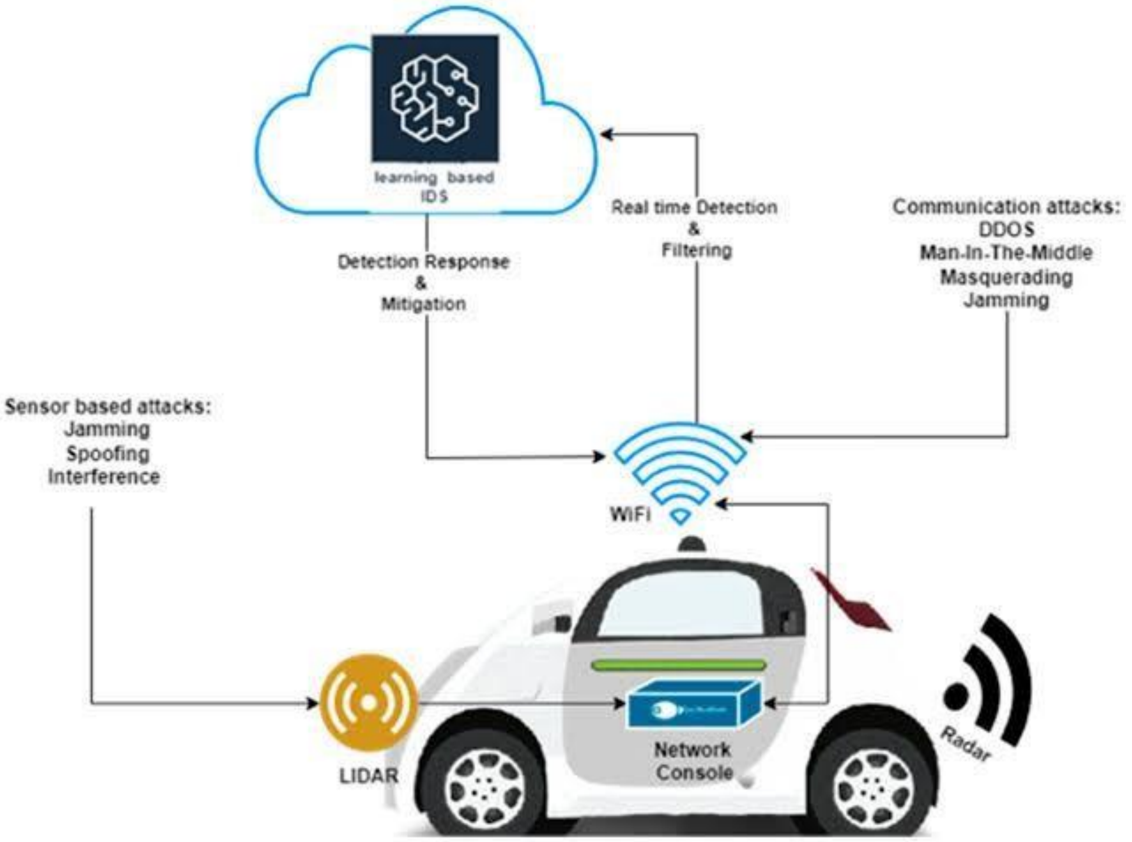
```
    elif data["lane_position"] == "right":
        actions.append("Adjusting left to stay in
lane.")     else:
        actions.append("Maintaining lane position.")

    if data["obstacle_distance"] < 20:
        actions.append("Obstacle detected! Slowing
down and preparing to stop.")     else:
        actions.append("Path is clear. Maintaining
speed.")

    return actions

def main():
    print("=== AI Autonomous Vehicle Simulator
===\n")
    for i in range(5):
        print(f"\n[Sensor Reading {i+1}] Gathering
vehicle data...")         time.sleep(1)         data =
get_sensor_data()
```

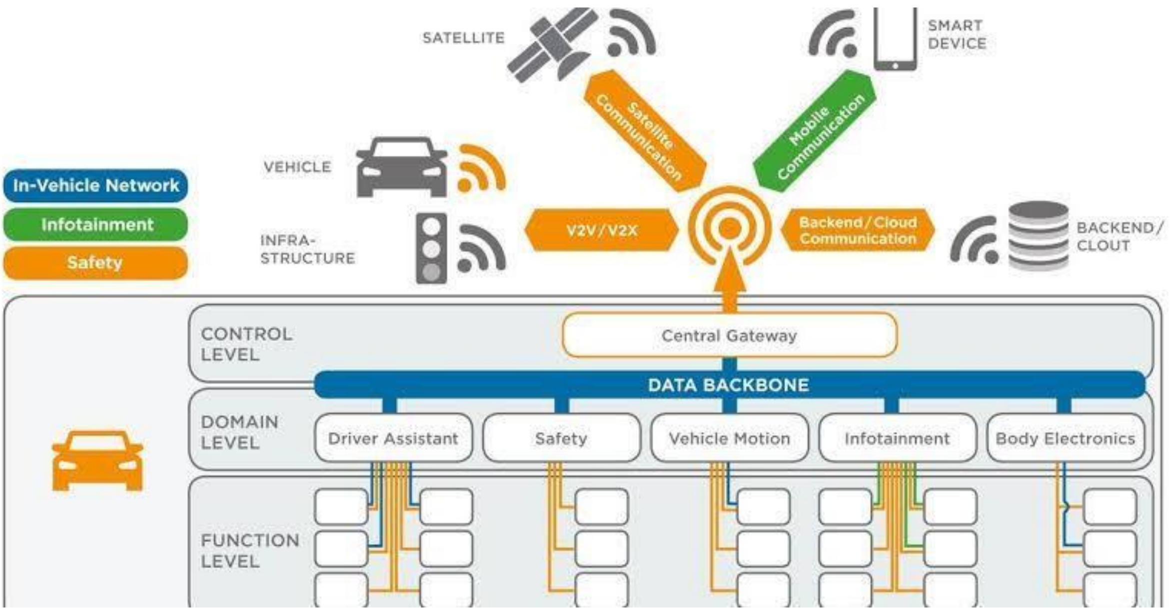
```
        print(f"Lane Position:
{data['lane_position']}")

        print(f"Obstacle Distance:
{data['obstacle_distance']} m")


        decisions =

ai_vehicle_decision(data)

print("\nAI Decisions:")          for

decision in decisions:

print(f"- {decision}")

time.sleep(2)


if __name__ == "__main__":

    main()
```

7.Working Images


Raspberry Pi
Camera Input
Object Detection Module
Lane Detection Module
traffic light detections
pedestrian detections
vehicle detections
lane detections
Traffic light detection handler
Pedestrian detection handler
Vehicle detection handler
Lane detection handler
Priority-Based Decision filtering
I2C
Arduino
Motor speed handler
Motor Driver
Ultrasonic Sensors
Motors


Autonomous Driving Mode
Cyber Security
V2X Connectivity
AV
Cloud Integration
User Experience


learning based IDS
Real time Detection & Filtering
Communication attacks:
DDOS
Man-In-The-Middle
Masquerading
Jamming
Detection Response & Mitigation
Sensor based attacks:
Jamming
Spoofing
Interference
WiFi
LIDAR
Network Console
Radar


SATELLITE
SMART DEVICE
Satellite Communication
Mobile Communication
VEHICLE
In-Vehicle Network
Infotainment
Safety
INFRA-STRUCTURE
V2V/V2X
Backend/Cloud Communication
BACKEND/CLOUT
CONTROL LEVEL
Central Gateway
DATA BACKBONE
DOMAIN LEVEL
Driver Assistant
Safety
Vehicle Motion
Infotainment
Body Electronics
FUNCTION LEVEL