Software Design Document

Autonomous Driving Without Lane Marks

Humayra Rashid, Daniel Gomez, Henry Xiong, Rodrigo Valdez

May 10, 2025

Contents

Version History

Version	Date	Comment	Author(s)
0.1	05/07/2025	Initial planning and purpose section	Humayra Rashid
0.2	05/08/2025	Added architecture and UI description	Henry Xiong
0.3	05/09/2025	Finalized glossary	Humayra Rashid
0.4	05/09/2025	Finalized references	Humayra Rashid

1 Introduction

1.1 Purpose

This Software Design Document (SDD) outlines the architecture, behavior, and system design for an autonomous driving algorithm that enables navigation without lane markings. The goal is to aid researchers and engineers in replicating and evolving the prototype.

1.2 Intended Audience

- Developers – for implementation and system extension
- Project Managers – to track feature integration
- Researchers and Educators – for system reference
- Testers – to design unit and integration test strategies

1.3 System Overview

The system processes real-time camera and sensor input to make driving decisions in environments lacking road lane markings. It operates in both simulation and physical toy car settings, emphasizing modularity, sensor feedback, and AI-driven pathfinding.

2 System Architecture

2.1 Workflow and Component Breakdown

- **Camera Input:** Captures the environment in front of the vehicle.
- **Sensor Module (SM):** Reads proximity, angle, and orientation data.
- **Processing Module (PM):** Preprocesses and analyzes sensor data.
- **Response Module (RM):** Generates steering and throttle commands.
- **Application Layer (AM):** Bridges PM and RM to external interfaces.
- **Simulator (SIM):** Offers a risk-free, reproducible environment.
- **Prototype Platform:** Embedded hardware implementation on Raspberry Pi or Arduino.

2.2 Component Diagram

- Languages: C++, Python
- Tools: OpenCV, CARLA simulator, Raspberry Pi OS
- Hardware: Raspberry Pi, Pi Camera, L298N motor driver

3 User Interface

3.1 How to Use

- Clone the repository from GitHub.
- Run simulation scripts using Python.
- Deploy algorithm to Raspberry Pi and connect hardware components.
- Use CLI or debug output logs for validation and tuning.

3.2 Database Explanation

No persistent database is used. All computation occurs in real time. Logs can be optionally saved for analysis.

4 Glossary

- **PM** – Processing Module
- **RM** – Response Module
- **OCR** – Optical Character Recognition (not used but contextually related)
- **SIM** – Simulator for testing
- **Raspberry Pi** – Hardware computer used in prototype
- **OpenCV** – Open-source computer vision library
- **Motor Driver** – L298N board to control motors
- **GPIO** – General-Purpose Input/Output pins

5 References

- https://github.com/arielsl/FujitsuAutonomousDriving
- https://opencv.org/
- https://github.com/carla-simulator/carla
- https://projects.raspberrypi.org/
- https://www.arduino.cc/en/Guide/HomePage
- https://docs.python.org/3/