1 Introduction

Since the previous report, the NacionalnaKlasa team has continued rapid development and system integration. The work included debugging the existing codebase, resolving Angular version issues, and integrating new components for communication with the Nucleo board and servo control in autonomous mode. Inter-process communication (IPC) on the Raspberry Pi and with the Nucleo was successfully tested. Lane and stop-line detection were implemented and integrated with good performance, while basic vehicle control was developed with planned future improvements. An initial dataset was created from onboard sensor and camera data and used to train a machine learning model for traffic sign detection, running in a separate thread. During this period, academic obligations required a temporary redistribution of tasks to ensure continued progress on the project.

2 Planned activities

Task assignments evolved naturally based on individual interests and strengths, becoming more clearly defined over time while remaining flexible as the project progressed.

- Konstantin Malešević & Filip Goldberger

Konstantin, the team leader, and Filip have contributed to debugging and refactoring the codebase, integrating new components, testing inter-process communication (IPC), and implementing communication with the Nucleo board for autonomous servo control. In the next period, their focus will shift to further refactoring and stabilizing server modules, improving system communication and connectivity, and optimizing vehicle control logic to enhance driving stability and autonomy. They will also develop a state-based control system with predefined parameters for different driving scenarios, as well as integrate a navigation module for route planning and waypoint following, aligned with the state-based control system.

- Minja Drakul & Iva Mančev

Minja and Iva have been focused on traffic sign detection and dataset-related tasks, and in the next period, they will concentrate on expanding and refining the datasets, improving preprocessing and organization, and enhancing the training, evaluation, and testing of traffic sign detection models. They will also work on integrating and validating detection modules to ensure reliable and accurate model performance

3 Status of planned activities

- Development and integration of the lane detection algorithm, along with implementation of basic vehicle control DONE
- Real-time monitoring of lane detection, traffic sign detection and vehicle control DONE
- Debugging and analysis of the provided codebase, including resolving Angular version issues DONE
- Integration of new system components, enabling communication with the Nucleo board, servo control in autonomous mode, and inter-process communication (IPC) within the system DONE
- Implementation of a Flask-based debugging server and dashboard optimization to reduce system load DONE
- Development of stop-line detection DONE
- Traffic sign detection (dedicated component, separate thread) *in progress, improvements planned*

- Development of a state-based control system for different driving scenarios *in progress*

4 General status of the project

- At the beginning of this phase, we focused on debugging the provided codebase, where a critical issue related to Angular version compatibility was identified and successfully resolved. This issue required a significant investment of time and temporarily blocked further development until the debugging process was completed.
- For communication with the Nucleo board, we utilized the existing serial communication module. We developed a test component for autonomous mode, which currently receives lane information from the line detection module and forwards calculated steering angles to the Nucleo via the existing module. In the future, this test component is planned to evolve into a full state-based autonomous control system, handling multiple driving scenarios and controlling the vehicle accordingly. This setup allowed us to validate inter-process communication (IPC) between our autonomous mode component and the Nucleo board.
- Lane detection was developed and our algorithm was integrated into the system, where it demonstrated reliable performance. Based on the lane detection output, basic vehicle control logic was implemented, enabling autonomous steering in its initial form, with planned improvements in future iterations. Work on stop-line detection has been successfully completed, and the system is now fully functional and ready for use.
- Traffic sign detection was implemented as a separate component, with the detection module fully integrated into the system. A new dataset was created based on data collected from the vehicle, and a custom YOLO model was built and used for traffic sign detection in the car. We observed that the model shows promising performance; however, in the future, we plan to invest more time in this task with the goal of improving its accuracy and robustness. Additionally, a Flask-based debugging server was implemented to facilitate system monitoring and testing.
- To reduce system load and improve performance, the dashboard was redesigned to prioritize streaming from the low-resolution (lo-res) camera instead of the main camera. Since the lo-res stream has a lower resolution, it requires fewer resources while still enabling real-time monitoring of lane detection, traffic sign detection, and vehicle control.

5 Upcoming activities

In the next period, we plan to work on:

- Further refactoring and stabilization of server modules, including improvements in system communication and connectivity, alongside ongoing optimization of vehicle control logic to enhance driving stability and autonomy.
- Continued development, training, and integration of the traffic sign detection algorithm using a newly created dataset, which will be further extended and improved in future reports.
- Development of a state-based control system with predefined parameters for different driving scenarios (e.g. intersections, highways), including speed, steering gain (α), and control parameter (k) for each state.
- Development and integration of a navigation module for route planning and waypoint following, aligned with the state-based control system.