Excel@FIT 2026
http://excel.fit.vutbr.cz

Autonomous vehicle on RC model car chassis for NXP CUP

Ján Skovajsa*

Abstract
The aim of this project is to design and program an autonomous racing car model for the international NXP Cup [1, 2] competition. The problem is solved using a distributed Dual-MCU architecture with two NXP FRDM-KL27Z [3] boards and a TSL1401 [4] linear camera FIG. 2. The vehicle features custom cast silicone tires mounted on 3D-printed rims to prevent under-steering during fast cornering Tire Diagram. The car processes camera data at a 125 Hz frame rate using an asynchronous DMA pipeline and navigates using adaptive Dual-PID [5] control logic FIG. 3. The model successfully completed the track challenges and achieved 3rd place [1] in the regional NXP Cup. This work demonstrates the effectiveness of hardware-software co-design, custom mechanical enhancements, and efficient real-time control algorithms in embedded systems.

*xskovaj00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

[Motivation] The NXP Cup [1, 2] is an international competition that challenges students to build and program autonomous vehicles, demanding practical skills in embedded systems, motor control, and computer vision.

[Problem definition] The objective is to design a fully autonomous vehicle controlled by NXP microcontrollers that can navigate a track with black borders as fast as possible, and accurately stop in front of an obstacle for the Mouser Lap of Honor [2]. The vehicle must use a camera as its primary navigation sensor, alongside MCUs or MPUs manufactured exclusively by NXP [2].

[Existing solutions] Successful previous teams [6] often replaced standard kits with professional RC chassis, utilized advanced cameras for faster frame rates, and implemented advanced algorithms for line detection and car control.

[Our solution] I utilized a modified DFRobot chassis with custom cast silicone tires Tire Diagram. The electronics feature a Dual-MCU architecture using two FRDM-KL27Z [3] boards communicating via a 500 kHz SPI bus FIG. 2. A TSL1401 linear camera [4] acts as the primary sensor for track navigation. It also incorporates a frame-to-frame line tracking algorithm, ensuring the system does not confuse the left and right edges and reliably tracks the correct line even when other distracting lines are near the track.

[Contributions] Key achievements include a highly optimized 125 Hz line-detection pipeline FIG. 3 using asynchronous data collection via ADC and DMA, a dual-PID system, and variable speed around the track based on its shape, all of which contributed to achieving a 3rd place finish in the competition.

2. Hardware and Architecture

The vehicle's hardware is built upon a distributed dual-MCU architecture FIG. 2 to ensure real-time responsiveness without bottlenecking the main processor. An additional reason for this approach was that a single KL27Z MCU [7] lacks sufficient hardware timers to independently control all the necessary motors and sensors.

The Master MCU (FRDM-KL27Z) handles the TSL1401 linear camera, executes the line detection algorithm, and drives the front-axle steering servo alongside the rear-wheel BLDC motors. It also contains the dual-PID and adaptive speed control logic for the car. The Slave MCU (also FRDM-KL27Z) is strictly dedicated to managing the HC-SR04 [8] ultrasonic sensor for obstacle detection, as well as the six TCRT5000 IR sensors located underneath the car for recognizing the finish line. To combat under-steering at high speeds, I developed custom cast silicone tires Tire Diagram.

3. Processing and Control Logic

The software utilizes an asynchronous ADC and DMA pipeline [7] to achieve a camera frame rate of 125 Hz, capturing 128 pixels per frame in 12-bit samples FIG. 3.

The line-detection algorithm processes this raw data by first flattening pixel sensitivity using a stored calibration profile and applying an 8px median filter to remove single-pixel spikes FIG. 4. An adaptive threshold then measures average brightness and auto-adapts to each frame to mark the dark pixels representing the track's edges.

To optimize the detection process, the algorithm employs frame-to-frame line tracking. This means it does not blindly scan the entire sensor array; instead, it searches for possible edges only around the line's position from the previous frame. Within this localized search window, the system uses cost-scored candidates to precisely identify the left and right edge positions. It scores every dark candidate using the following cost function:

$$\text{price} = I(i) + \lambda \cdot |i - \hat{i}|$$

where:

- $I(i)$ is the preprocessed pixel intensity (after applying calibration and a median filter),
- $\hat{i}$ is the previous-frame edge position,
- λ is a constant (0.2) applying a distance penalty.

After evaluating these scores and checking the minimum track width, it calculates the track center.

The vehicle's steering is governed by a Dual-PID [5] logic system FIG. 5. A Normal PID provides stable tracking used primarily in curves, while the system switches to a Fast PID (with softer gains for smoother steering) when the vehicle is centered on a straightaway at high speeds.

4. Conclusions

The implemented autonomous vehicle successfully met the requirements of the NXP Cup rules [2]. A major factor in this success was the development of a highly robust, state-aware line tracking algorithm. By anchoring edge searches to previous frame positions and employing dynamic thresholding with cost-penalty scoring, the system reliably filters out false lines and track anomalies.

To support this high-speed computational pipeline, a distributed Dual-MCU architecture FIG. 2 was implemented. By eliminating timing conflicts, this approach guaranteed a stable 125 Hz camera loop and allowed for the seamless integration of obstacle detection, finish-line sensors, and precise motor control. Furthermore, custom mechanical upgrades, such as the cast silicone tires, effectively resolved the physical under-steering limitations of the stock chassis. This integration of advanced software algorithms, dual-processor electronics, and mechanical engineering culminated in a 3rd place finish at the regional NXP Cup competition.

Acknowledgements

I would like to thank my supervisor, Ing. Jakub Liška, for his guidance and expert advice throughout the development of this project.

References

[1] NXP Cup. [online], 2025. https://nxpcup.nxp.com/.

[2] The NXP Cup Rules 2025/2026. [online], 7 2025. https://nxpcup.nxp.com/docs/en/supporting-information/NXP-Cup-2026-Rules.pdf.

[3] Freescale Semiconductor, Inc., Austin, Texas, USA. *FRDM-KL27Z User's Guide*, 2015.

[4] Texas Advanced Optoelectronic Solutions, Inc., Plano, Texas, USA. *TSL1401R-LF 128 x 1 LINEAR SENSOR ARRAY WITH HOLD*, 5 2006. https://www.farnell.com/datasheets/315815.pdf.

[5] Karl J. Åström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America, Research Triangle Park, NC, USA, 2 edition, 1995.

[6] Tim-Louis Lewejohann. Supersmart autonomous car (nxp cup 2020). [online], 2020. https://www.electromaker.io/project/view/supersmart-autonomous-car-nxp-cup-2020.

[7] Freescale Semiconductor, Inc., Austin, Texas, USA. *Kinetis KL27 Microcontroller*, 2015. https://www.nxp.com/docs/en/data-sheet/KL27P64M48SF2.pdf.

[8] DigiKey Electronics, Thief River Falls, Minnesota, USA. *HC-SR04 Ultrasonic Ranging Module*, 2025. https://www.digikey.com/htmldatasheets/production/1979760/0/0/1/HC-SR04.pdf.