POLITECNICO DI TORINO
Repository ISTITUZIONALE

Open-Source Based and ETSI Compliant Cooperative, Connected, and Automated Mini-Cars

Original
Open-Source Based and ETSI Compliant Cooperative, Connected, and Automated Mini-Cars / Farina, L., Gavioli, F., Iandolo, S., Moretti, F., Perrone, G., Piccoli, M., Raviglione, F., Rapelli, M., Solida, A., Burgio, P., Grazia, C.A., Bazzi, A.. - (In corso di stampa). (2026 IEEE 103rd Vehicular Technology Conference (VTC2026-Spring) Nice (FRA) June 9–12, 2026).

Availability:
This version is available at: 11583/3016330 since: 2026-10-05T12:34:53Z

Publisher:
IEEE

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

20 November 2026

Open-Source Based and ETSI Compliant Cooperative, Connected, and Automated Mini-Cars

Lorenzo Farina*†, Federico Gavioli‡, Salvatore Iandolo‡, Francesco Moretti‡, Giuseppe Perrone¶, Matteo Piccoli*, Francesco Raviglione§, Marco Rapelli¶, Antonio Solida‡, Paolo Burgio‡, Carlo Augusto Grazia‡, and Alessandro Bazzi*†

*DEI, Università di Bologna, 40136 Bologna, Italy
†National Laboratory of Wireless Communications (WiLab), CNIT, 40136 Bologna, Italy
‡DIEF, University of Modena and Reggio Emilia, 41125 Modena, Italy
§Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, 10129 Torino, Italy
¶Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy

arXiv:2603.06343v1 [cs.RO] 6 Mar 2026

***Abstract*—The automotive sector is following a revolutionary path from vehicles controlled by humans to vehicles that will be fully automated, fully connected, and ultimately fully cooperative. Along this road, new cooperative algorithms and protocols will be designed and field tested, which represents a great challenge in terms of costs. In this context, in particular, moving from simulations to practical experiments requires huge investments that are not always affordable and may become a barrier in some cases. To solve this issue and provide the community with an intermediate step, we here propose the use of 1:10 scaled cooperative, autonomous, and connected mini-cars. The mini-car is equipped with a Jetson Orin board running the open Robot Operating System 2 (ROS2), sensors for autonomous operations, and a Raspberry Pi board for connectivity mounting the open source Open Stack for Car (OScar). A key aspect of the proposal is the use of OScar, which implements a full ETSI cooperative-intelligent transport systems (C-ITS) compliant stack. The feasibility and potential of the proposed platform is here demonstrated through the implementation of a case study where the Day-1 intersection collision warning (ICW) application is implemented and validated.**

***Index Terms*—Vehicle-to-everything; Cooperative-Intelligent Transport System; Low-cost hardware; Open-source software.**

I. Introduction

Robotaxis are on the road in some cities, more and more automation features are being mandated in new vehicles, and, regarding connectivity, direct vehicle-to-everything (V2X) connectivity is being added to the already consolidated availability of cellular communications.[1] Yet, most of the vehicles are still unable to share information, and the next cornerstone will be achieved once vehicles can exchange their local views and cooperate to execute joint maneuvers.

Along this path, new cooperative algorithms and protocols need to be designed and validated. Although the use of simulators is an important first step, it needs to be followed by field tests that require huge investments and may not always be affordable. Hence, an intermediate step between simulations and field tests is required. To this aim, we propose an original platform built on a 1:10 scaled car, equipped with automated functions and standard compliant communication protocols.

The idea of a scaled platform is not new, and rather a large number of proposals have been made recently as surveyed for example in [1], [2]. More specifically, in [1], more than twenty different proposals are considered and compared, which all have strengths and weaknesses, but all miss the implementation of a full standard-compliant communication protocol stack. Among them, for instance, a very low-cost solution is proposed in [3], consisting of 1:24 scaled mini-cars, each controlled by a Raspberry Pi Zero and connected via Wi-Fi; the limited size and cost do not allow hosting sensors like LiDARs or to install sophisticated software for control and communication. Additionally, a well-known platform is the slightly more expensive Duckietown proposed in [4], which uses a Raspberry Pi 2 platform with a camera and Wi-Fi. As in the previous case, the limited cost and small size restrict the sensors and software that can be deployed. Among all, the setup proposed in [5], which was designed for the study of edge computing solutions, is the one closest to ours. The authors use indeed the same 1:10 scaled mini-cars equipped with real sensors (a LiDAR and a camera). In contrast to our proposal, they do not specifically address communication protocols and exploit standard Wi-Fi.

Hereafter, an original platform is proposed, which is based on 1:10 scaled autonomous mini-cars, equipped with an ETSI cooperative-intelligent transport systems (C-ITS) compliant on-board unit (OBU) realized using a Raspberry Pi 5 Model B device [6]. Remarkably, the proposed platform is, to the best of our knowledge, the first with a full standard-compliant protocol stack. The cost is around 3000 €, which is higher compared to other proposed solutions, but still sufficiently limited to make the testbed scalable.

To demonstrate the possible use of the platform, we have implemented a case study in which the Day-1 intersection collision warning (ICW) application is developed and tested using the proposed platform.

[1]In Europe, the number of vehicles equipped with ITS-G5 exceeded 2 million in 2025, while in the US the FCC has finalized at the end of 2024 new rules for the use of the ITS band by C-V2X.

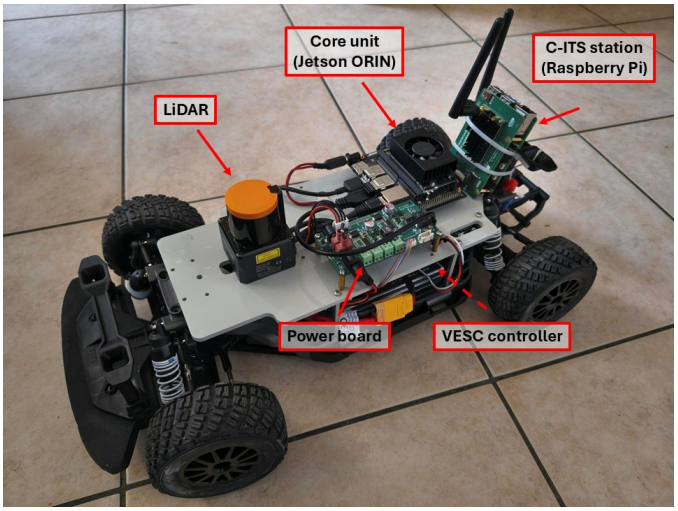


Fig. 1: Cooperative, connected and automated mini-car, equipped with the low-cost OBU.

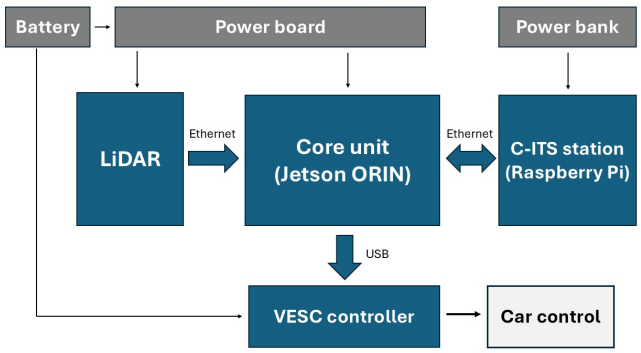


Fig. 2: Block scheme of the hardware components.

II. The Cooperative, Connected and Automated Mini-Car

In this section, some details about the platform are provided. A picture of the platform with the main components is shown in Fig. 1, a block scheme of the main parts and their connections is illustrated in Fig. 2, and a list of the hardware components with approximate costs is provided in Table I.

A. The F1tenth mini-cars

The mini-cars are based on the Roboracer platform [7], previously known as F1tenth. This platform was created to encourage research in the field of autonomous driving. Additionally, the web site provides a basic open-source software stack and a complete guide on how to build the mini-car. Over the years, it gathered an active community that contributed to the advancement, improvement and extension of the platform. The Roboracer platform is now a mature and modular solution that offers multiple alternatives for the software components to adapt to different use-cases beyond racing.

The Roboracer autonomous platform is based on a commercially available 1/10 scaled radio-controlled car chassis. To enable the autonomous navigation of the vehicle, a deck was added to accommodate all the required hardware. This includes the core computing unit, sensors, power distribution and vehicle control boards. The vehicle is provided with a brush-less DC motor to control the speed and a servomotor for steering. These actuators are controlled with a Vedder Electronic Speed Controller (VESC) that abstracts the motor control, exposing a serial link to the core computing unit and providing feedback data. The VESC is also equipped with a BMI160 inertial sensor, whose measurements are available via the serial interface. Finally, the main sensor for the vehicle is a 2D LiDAR rangefinder, namely a Hokuyo UST-10LX. The core computing unit that runs the autonomous navigation tasks is an NVIDIA Jetson Orin, either NX or Nano Super, which is directly connected to all sensors and actuators. The NX and Nano Super host, respectively, an 8-core or a 6-core Arm Cortex-A78AE CPU, and an embedded GPU, which can be used to speed-up computationally-intensive tasks.

The software stack is built over Robot Operating System 2 (ROS2) [8], the de-facto standard framework for robotics applications. The framework provides two main abstractions: nodes, which represent software components, and topics, for inter-process communication, following a publish-subscribe paradigm. The main advantage of this approach is that the resulting software stack becomes modular, enabling the integration of ready-made software components to improve, adapt or extend the software stack.

The software components that make up the embedded software stack are the localization, the control, and the drivers for the sensors and the VESC. More specifically, the localization component is based on a particle filter [9], a widely used algorithm in robotics state estimation based on the Monte Carlo method. The control component is instead based on the pure pursuit algorithm [10], a geometric control algorithm used to follow a predetermined reference trajectory.

As detailed in Table I, the cost of the hardware components to realize a mini-car, before adding the C-ITS OBU, is on the order of 2500-3000 €, which is mainly due to the LiDAR (more than half of the budget with current prices), the chassis, and the Jetson ORIN. Regarding the LiDAR, we use a model that allows us to comply with the RoboRacer rules; however, cheaper alternatives are in principle possible. Concerning instead the Jetson Orin, both the NX or the Nano Super can be used, with the former allowing for improved performance but at a higher cost.

B. The C-ITS OBU hardware

To integrate C-ITS V2X capabilities into the RoboRacer autonomous platform, a compact but sufficiently powerful hardware unit and an open-source protocol stack need to be deployed. Concerning the hardware, the goal is the design of a compact low-cost solution, supporting *(i)* the IEEE 802.11p standard [11], *(ii)* the exchange of ETSI ITS-G5 messages, *(iii)* dual operation as both road side unit (RSU) and OBU, *(iv)* the usage of off-the-shelf components, and *(v)* the suitability for rapid deployment in experimental scenarios.

Our proposed solution, also detailed in [6], is based on a Raspberry Pi 5 Model B single-board computer, equipped with an Arm Cortex-A76 processor, 8 GB of LPDDR4 memory, and 128 GB of flash storage. The board is powered through a 5 V / 5 A USB-C interface and provides a PCIe 2.0 connection, enabling high-speed expansion through standardized Hardware Attached on Top (HAT) modules. This design choice enables the integration of additional peripherals, such as storage devices and wireless modules, while preserving a compact and portable form factor.

For vehicular communication, a PCIe HAT is used to host a MikroTik R11e-5HnD wireless card, based on the Qualcomm Atheros AR9580 chipset and equipped with a passive heat sink. Initially intended for IEEE 802.11a/n operation, the AR9580 chipset natively supports the 5.9 GHz frequency band used for ITS communications. Access to these frequencies is enabled at the software level through the Linux `ath9k` driver, which can be extended to support IEEE 802.11p operation.

Support for ITS-G5 communications in the 5.9 GHz band is activated through a set of modular and automated configuration scripts executed at runtime. These scripts apply custom patches to the `ath9k` driver to unlock IEEE 802.11p channels and update the Linux regulatory database (`regdb`) to include the appropriate frequency allocations [12].

A detailed list of the hardware components used in the proposed platform, together with their approximate costs, is reported in Table I. The overall cost of the setup is in the order of 150 €. This is significantly lower than the price of comparable commercial or proprietary V2X solutions, while still providing the flexibility and openness required for research, prototyping, and experimental validation.

C. OScar: An ETSI C-ITS compliant protocol stack implementation

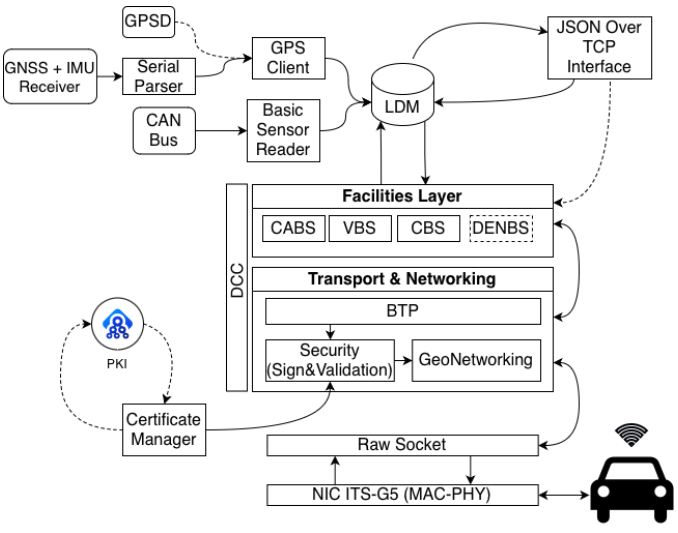


Fig. 3: Block diagram of the OScar stack and its main functional components

The software deployed on the mini-cars is based on a fully open-source implementation of the ETSI C-ITS networking stack. This implementation, named OScar [13], is released under the GPLv2 license and is regularly updated with new features, message versions, and security functionalities, in line with ETSI specifications [14]. The adoption of an open-source stack offers several advantages, including the ability to flexibly deploy the stack on embedded platforms such as Raspberry Pi boards without incurring licensing costs and being able to customize the framework in all its components for research purposes.

The internal architecture of OScar and its main functional components are illustrated in Fig. 3. It follows a modular, Linux-based software stack design that can be easily deployed on low-power embedded systems. The entire codebase can be compiled into a single executable of less than 3 MB, which can be configured either via command-line parameters or through a dedicated API that uses JSON-based data structures over TCP. This API enables, for example, the activation of Cooperative Awareness Message (CAM) dissemination or the configuration of ETSI Decentralized Congestion Control (DCC) mechanisms. Additional ETSI functionalities, such as DCC and secured communications, are also supported and can be enabled for extended experimental campaigns. Through the API, it is also possible to access the contents of the internal Local Dynamic Map (LDM), which maintains a dynamic representation of the road environment. The LDM is populated using information received from V2X messages, as well as data obtained from on-board sensors detecting other road users. In our setup, the LDM stores the most up-to-date information about other mini-cars based on data received through CAMs version 2 messages. This information can be retrieved by sending a JSON request that specifies a reference center position, which allows for the automatic computation of relative distances in addition to the kinematic state of surrounding mini-cars.

Beyond implementing the full set of ETSI C-ITS basic services for the transmission and reception of standardized messages, the framework also supports multiple positioning data sources. These comprise *(i)* the Linux `gpsd` daemon, which provides processed Position Velocity Time (PVT) data from a GNSS receiver; *(ii)* a virtual positioning provider that replays previously collected GNSS and Inertial Measurement Unit (IMU) traces using the JSON format adopted by the TRACEN-X recording and replaying platform [15]; and *(iii)* a highly efficient serial-parsing module that interfaces directly with GNSS receivers via a serial connection, decoding PVT data from NMEA sentences and, when available, IMU information from u-blox chipsets through UBX messages.

In our case, we leverage for positioning the cartesian coordinates expressed in the mini-car's local reference system. These are converted into NMEA sentences by a Python script running on the Jetson Orin and then forwarded to the OBU. On the OBU, a virtual device emulates a GNSS receiver and exposes the sentences to the `gpsd` daemon, thus leveraging the standard positioning pipeline described above.

TABLE I: List of hardware components and indicative prices.

Device	Details	Cost
F1tenth		
Chassis	Radio-controlled Traxxas mini-car	450 €
Core control unit	Jetson Orin Nano Super/NX	350/800 €
SD and SSD cards	Memories for the control unit	30 €
LiDAR	2D Hokuyo UST-10LX	1500 €
VESC	Car mobility controller with inertial sensor	250 €
Battery	Battery-pack	50 €
Power board	Custom PCB for power distribution	100 €
Small additional	ABS sheet and fasteners for assembling	20 €
C-ITS radio		
Main board	Rasberry Pi 5 Model B	80 €
Power supply	Cable-supply or power bank	20 €
Memory	MicroSD, at least 32 GB	10 €
miniPCIe module	Rasp5 HAT for miniPCIe	15 €
Fasteners	Spacers (4) and screws (4)	5 €
Wi-Fi mPCIe	R11E-5HND ath9k	25 €
Antenna conn.	From MMCX to SMA	5 €
Antenna	SMA antenna for 5.9 GHz	5 €

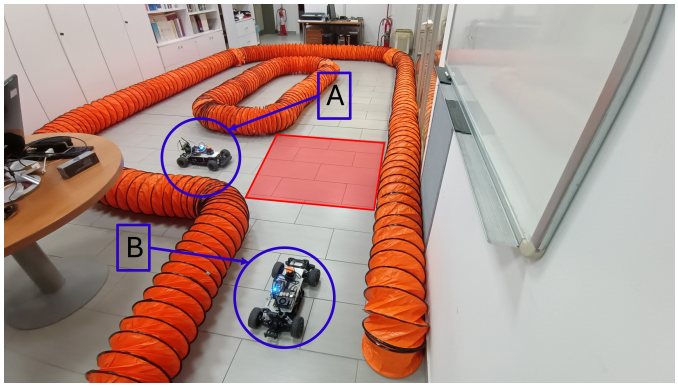


Fig. 4: Environment used to test the applications.

III. Example Application

The goal of the proposed platform is to develop and validate C-ITS applications. In this section, we show a case study in which we implemented a Day-1 application, namely the ICW, which exploits information obtained from CAMs to increase drivers' awareness and prevent collisions at intersections. The application was implemented using Python and runs in a target car; when the car is close to an intersection and another cooperative car is approaching the same intersection, the application generates a visual alert message to warn the driver.

To test the application, we set up the testing environment shown in Fig. 4. The orange flexible tubes are used to define an oval track for the mini-cars. The track is left open at one point to simulate an intersection area, highlighted in red. During the experiments, mini-car A continuously follows an oval trajectory inside the track, while mini-car B remains stationary. The autonomous guide of mini-car A is enabled by recording a map of the track using its LiDAR and by designing a reference trajectory within the mapped area. Fig. 5 shows the generated map, with red points representing the coordinates indicated in the CAMs transmitted by mini-car A while following the trajectory, as received by mini-car B. Although mini-car A moves at a fixed speed of 1 m/s, the coordinate conversion mechanism described in Section II-C applies a 1:10 scale (same as the mini-cars); consequently, OScar computes an effective speed higher than the real one and transmits CAMs at a frequency exceeding 1 Hz.

The ICW application is executed in a Python script on a PC connected to the OBU of mini-car B through a socket. The script continuously sends requests to OScar to retrieve the contents of its LDM. The received information is then processed to assess whether another vehicle (i.e., mini-car A) is approaching the intersection area. In such a case, a warning is generated and displayed on the screen. In a real-world implementation, this warning would be shown on an in-vehicle screen or used to produce a recognizable sound, to warn the human driver of the risk of a potential collision.

The experimental test used to validate the application is illustrated in Fig. 6, showing both external and on-board views.

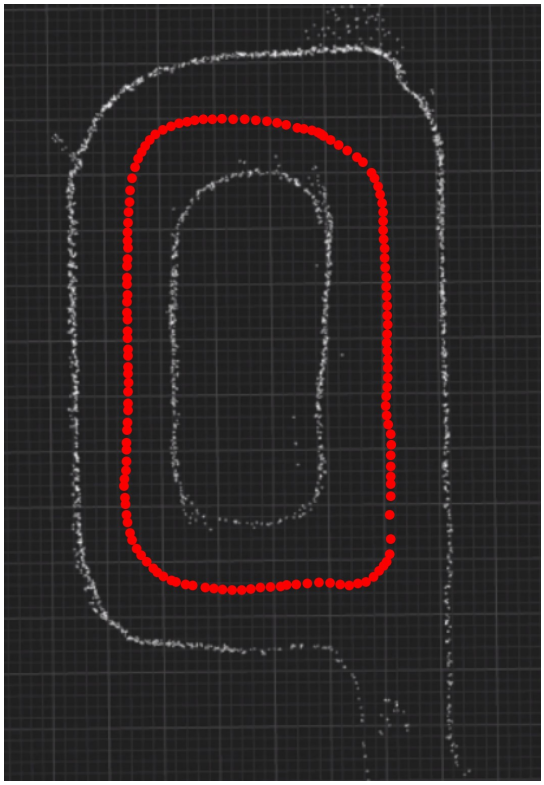
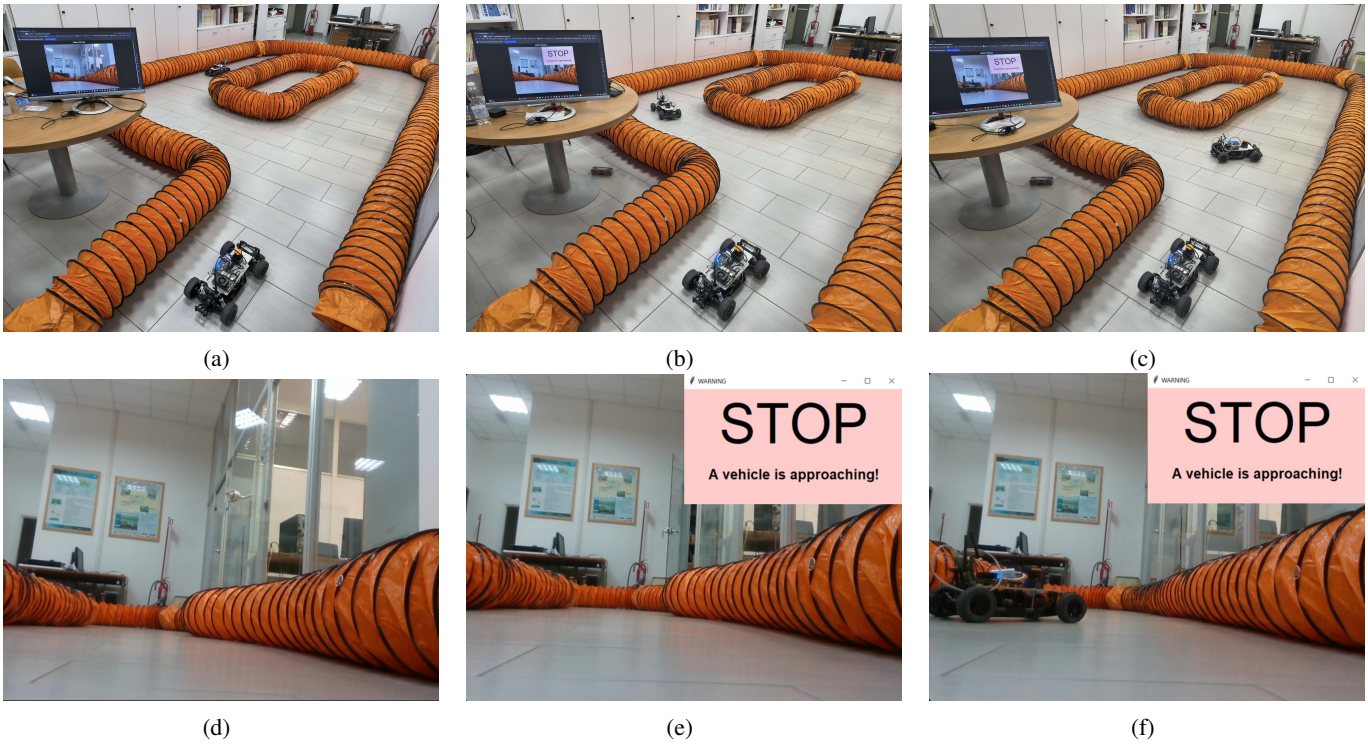


Fig. 5: Screenshot of the map with red points illustrating the position of mini-car A as indicated in its transmitted CAMs.

The on-board view is provided by a camera mounted on mini-car B, which transmits the acquired images to the PC. Fig. 6a describes the setup at the start of the test: mini-car A is still away from the intersection and not visible from mini-car B, as seen in Fig. 6d; thus, a warning for mini-car B is not yet triggered. In Fig. 6b instead, mini-car A approaches the intersection; Fig. 6e shows that mini-car B is still not visually aware of the presence of mini-car A, but the warning is triggered to inform the driver of the risk of collision. Then, mini-car A actually crosses the intersection in Figs. 6c and 6f.



(a) (b) (c)

(d) (e) (f)

Fig. 6: External and on-board views of three experimental phases. (a-c) External camera snapshots of the three phases: mini-car A away from the intersection area, approaching the intersection area and crossing the intersection area, respectively. (d-f) Corresponding on-board (vehicle perspective) views for the same phases.

IV. Conclusion

In this work, a new platform for the field trial of ETSI C-ITS applications is proposed, where a 1:10 mini-car is equipped with sensors and a standard compliant OBU using hardware with limited cost and open-source software. The applicability of the platform is demonstrated with a case study implementing a Day-1 ICW application. The plan for future work is to apply the platform to Day-2 collective perception applications and Day-3+ maneuver coordination applications, with particular reference to intersections.

Acknowledgment

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - RESTART), project MoVeOver, and national research centre on mobility (CN00000023 - MOST).

References

[1] A. Mokhtarian, J. Xu, P. Scheffe, M. Kloock, S. Schfer, H. Bang, V.-A. Le, S. Ulhas, J. Betz, S. Wilson, S. Berman, L. Paull, A. Prorok, and B. Alrifaee, "A survey on small-scale testbeds for connected and automated vehicles and robot swarms: A guide for creating a new testbed [survey]," *IEEE Robotics & Automation Magazine*, 2025.

[2] S. Schfer and B. Alrifaee, "From small-scale to full-scale: Assessing the potential for transferability of experimental results in small-scale CAV testbeds," in *2024 IEEE ICVES*, 2024.

[3] N. Hyldmar, Y. He, and A. Prorok, "A fleet of miniature cars for experiments in cooperative driving," in *IEEE ICRA 2019*.

[4] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, D. Hoehener, S.-Y. Liu, M. Novitzky, I. F. Okuyama, J. Pazis, G. Rosman, V. Varricchio, H.-C. Wang, D. Yershov, H. Zhao, M. Benjamin, C. Carr, M. Zuber, S. Karaman, E. Frazzoli, D. Del Vecchio, D. Rus, J. How, J. Leonard, and A. Censi, "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in *IEEE ICRA 2017*.

[5] A. Elmoghazy, K. Elgazzar, and S. Alwidian, "A real-world testbed for V2X in autonomous vehicles: From simulation to actual road testing," in *IEEE ICFEC 2024*.

[6] L. Farina, M. Piccoli, S. Iandolo, A. Solida, C. A. Grazia, F. Raviglione, C. Casetti, and A. Bazzi, "Low cost c-its stations using raspberry pi and the open source software oscar," in *IEEE VTC2025-Spring*.

[7] M. O'Kelly, V. Sukhil, H. Abbas, J. Harkins, C. Kao, Y. V. Pant, R. Mangharam, D. Agarwal, M. Behl, P. Burgio, and M. Bertogna, "F1/10: An open-source autonomous cyber-physical platform," 2019.

[8] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, 2022.

[9] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, *Particle Filters for Mobile Robot Localization*. Springer New York, 2001, pp. 401–428.

[10] R. S. Wallace, A. Stentz, C. E. Thorpe, H. P. Moravec, W. Whittaker, and T. Kanade, "First results in robot road-following," in *IJCAI 1985*.

[11] C. A. Grazia, "On the performance of IEEE 802.11 p Outside the Context of a BSS Networks," in *IEEE PIMRC 2018*.

[12] S. Iandolo, C. A. Grazia, M. Klapez, and M. Casoni, "Odu: Open dsrc unit toward effective dissemination of v2x applications," in *IEEE VTC2025-Spring)*.

[13] M. Rapelli, F. Raviglione, and C. Casetti, "OScar: An ETSI -Compliant C- ITS Stack for Field- Testing with Embedded Hardware Devices," in *MedComNet 2024*.

[14] ETSI, "ETSI TS 102 731 V1.1.1 (2010-09): Intelligent Transport Systems (ITS); Security; Security Services and Architecture," Tech. Rep.

[15] D. Gasco, C. M. Risma Carletti, F. Raviglione, M. Rapelli, and C. Casetti, "TRACEN-X: Telemetry Replay and Analysis of CAN Bus and External Navigation Data," in *IEEE VTC2025-Fall*.