Nandor A. Kilyen, Rares F. Lemnariu, George D. Mois, Yaobin Chen, Brendan T. Morris, and Ionut Muntean

The IEEE ITSS and Bosch Future Mobility Challenge: A Hands-On Start to Autonomous Driving

Autonomous driving (AD) is a complex field and requires a wide range of competencies of scientists and engineers involved in the mobility domain. Finding the best-suited and most attractive field and developing the right competencies for it demands an early understanding of and interaction with the field. The Bosch Future Mobility Challenge (BFMC) is a competition created to provide students with a platform to interact with all AD domains and to develop and test their algorithms in an environment simulating a real city, using 1/10-scale model cars. This article presents an overview of the competition, its development, and its impact on the main stakeholders, i.e., the participants and the academic environment.

AD is on the way to becoming a reality, fueled by rapid advances in wireless communication, sensor technologies, and artificial intelligence. Self-driving vehicles will change current transportation systems, making them more efficient, safer, and greener by relying on electronics for the implementation of tasks, such as avoiding obstacles, crossing intersections, detecting traffic signs and crosswalks, and navigating through cities [1]. These intelligent functionalities can be grouped into four major classes, namely, sensors, perception and scene understanding, behavior and motion planning, and vehicle control and actuation [2]. These require a special set of competencies for scientists and engineers involved in their development. Self-driving cars are complex systems, requiring that the people working in the AD field on their design and development have competencies in different engineering disciplines, such as mechanical engineering, electrical engineering, and computer science.

Practice has shown that the structuring and decomposition of such system complexity and the management of the issues brought by real-life implementations are challenging tasks for students [3]. Therefore, training centers, universities, and the industry should adapt to these new requirements and prepare their students and associates entering the AD field. This requires investment in educational activities and the adaptation of curricula for raising interest in AD technologies. Currently, various companies, including Bosch and Microsoft, are investing in AD technologies education, reflecting the increased interest in this domain. However, there is a wide range of challenges associated with training in the AD field. One category consists of defining efficient ways of dealing with sensors, perception, localization and mapping, path-planning activities (both local and global), vehicle control, and communication [vehicle-to-vehicle (V2V) and vehicle-to-everything]. Others are related to the integration of hardware and software into the vehicle's self-driving algorithms, testing and evaluation strategies, and assessment of the limitations of AD systems [4]. Having this in mind, the Bosch Engineering Center, Cluj-Napoca, Romania, launched BFMC (https://boschfuturemobility.com/) in 2018 with the goal of providing a framework for students to develop competencies in the AD field.

BFMC is an international technical competition where teams of up to five students, mentored by a university faculty member or researcher, implement AD functionalities on 1/10-scale model cars in an environment simulating a real city. The car kits, built using open

Digital Object Identifier 10.1109/MITS.2021.3081939
Date of current version: 22 July 2021

source hardware and software modules, are provided by the organizers for each team admitted into the competition. They are affordable and scalable platforms based on the Raspberry Pi (RPi) camera for sensing, an RPi single-board computer (SBC) as the cognitive part for movement planning and mapping, and an STM32 Nucleo board as the controller for low-level functions, such as steering and moving forward or backwards. These components can be complemented by other electronic devices, such as ultrasonic sensors or more powerful data processing units, within certain limits, if a team desires.

After more than six months of development, during the competition finals, the autonomous vehicles are required to navigate in a city-like environment that includes several road signs, intelligent traffic lights, a complex digital map, crosswalks, moving pedestrians and vehicles, and a video-based localization system. The road elements are made up of a highway, city streets, a roundabout, parking places, and a ramp. The self-driving cars have to be programmed so that they complete the tasks given by the organizers within two challenges: a technical run and a speed run. During the technical challenge, the vehicles have to reach and pass designated obstacles on the map with as few mistakes as possible. During the speed run, the vehicles have to reach a point on the map and return to the starting point as fast as possible while obeying traffic rules. The challenge has evolved over time, starting from a local competition in 2018, where teams only from Cluj-Napoca participated, to an international one, which received submissions from more than 10 countries for the current 2021 edition. This year's event also marks the beginning of a collaboration with the IEEE Intelligent Transportation Systems Society (ITSS) that will further increase the relevance and quality of the competition within the academic and industrial communities.

Competition

As we want the competitors to touch all of the AD aspects and use a cross-discipline architecture, each aspect requires a very different set of skills. We shaped the competition setup to cover all four AD-related aspects: sensing, perception and scene understanding, behavior and motion planning, and vehicle control and actuation. On the platform that we give to the students, we have already included a big part of the sensing and actuation modules through our basic software and car kit hardware. However, the participants have the freedom to change, add, or remove any part, adopting whatever solution they may find more fitting for the team activity (while still respecting the competition requirements).

A big problem the teams have to face on the sensing side (which we do not make available) that even we, as drivers, struggle to face, is associated with intense lighting changes. Adding noise-cancelling filters on their perception sensors is necessary to ensure that the perception layer will process the right information.

Since the camera is the main, most accessible, and most general sensor, many participants consider perception and scene understanding as the primary and most visible layer in AD, concentrating many resources on this. In our competition, they touch this topic in almost all of our challenges, i.e., when detecting the road marking, the traffic signs and traffic lights, traffic participants (such as static and moving cars or pedestrians), and other traffic elements (in our case roadblocks and highway separators).

Another very important layer, though not as visible at first glance, is behavior and motion planning. The capabilities of these algorithms are tested as students are planning their next move at an intersection based on the desired destination, as they are deciding when the car in front should be overtaken while respecting road marking restrictions, and as they are deciding whether staying in the lane or braking because of a pedestrian is more important. All of these topics have to be associated with a categorization of importance, fail-safe features, parallelization of the software, and a well-defined software architecture.

The last but not least layer, which is most visible to the outside world, is vehicle control and actuation. How well can the car stay on the road? Can it adapt its speed to the radius of the next curve? What about the car cruising uphill or downhill? Will the car be able to park within the limitations of this parking spot? What happens if the other car did not park within its limits? All of these are questions students have to answer through their chosen control method.

Requirements

As noted previously, each team has to complete two different types of runs on the track: a technical and a speed run. On both runs, all of the traffic rules apply (these are also included in the competition regulations).

To encourage the participation of both returning and new teams, a dynamic point system was adopted for the technical run with the purpose of giving the competition more modularity, multiple run possibilities, and freedom of choice. There is a series of small, specific scenarios on the competition track that are divided into two categories: mandatory and optional. Failure to complete all of the mandatory scenarios (reacting to a pedestrian on the crosswalk, overtaking static cars, parking in a designated spot, navigating at traffic lights intersection, and passing a ramp) results in no points on the specific run. Completing all of the optional obstacles (reacting to a pedestrian in an unsigned area, navigating on a highway and overtaking a moving vehicle, navigating in a roundabout, tailing a vehicle, turning right/left at an intersection based on a road block, and reaching the finish line) demonstrates enhanced team capabilities and results in additional points and an advance in the ranking.

What We Offer

- *A ready-to-use car kit for developing an autonomous solution:* Each team receives a 1/10-scale electric car kit for participating in the challenge. The car kit, composed of the hardware and software platforms, is handed over to the participating teams assembled, ready for the development and testing of autonomous algorithms.
- *A complex environment imitating real-life scenarios (the racetrack):* The challenge is organized on the custom-built racetrack, where the participants can demonstrate the performance of their approaches. The racetrack aims to reflect many real-life scenarios for autonomous vehicles, like parking spots, city streets, and highways.
- *A digital twin of the racetrack for testing:* This simulated environment is a replica of the elements proposed by the challenge's scenarios. It is designed to help the participants in developing, evaluating, and debugging their algorithms.

Hardware Platform

Our goal was to create a car kit that was ready to use for developing autonomous functionalities. We wanted to provide an easy-to-understand and flexible system that could be further modified. Each team receives a fully assembled kit, which is already flashed with the basic software version.

The platform is built using one STM32 Nucleo F401RE microcontroller and one RPi 4B SBC as the main processing components. The sensors premounted on the car kit include a Raspberry camera v2, an AMT103 encoder, and a BNO055 smart sensor. The actuators are a brushed dc motor and a servomotor, where the servomotor controls the car's steering system and the dc-brushed motor manages its linear movement. The dc motor is driven by a VNH5019 motor driver, which receives commands from the microcontroller, while the AMT103 encoder is used to provide a feedback loop and determine the dc motor's rotation speed. Therefore, the AMT103 encoder has a mechanical connection with the motor. The BNO055 is a smart sensor with accelerometer, gyroscope, magnetometer, and orientation software that can provide data related to the car's movement and heading. We designed two printed circuit boards: a carrier board for BNO055 and a power distribution board. The former provides the communication and power connections for the BNO055 integrated circuit while the latter takes care of supplying a proper voltage for each electrical component of the car. The connections among the hardware elements are presented in Figure 1.

Software Platform

We offer open source tools, examples, and guides for the participants to help in the starting steps. These resources help them control the car and implement the interaction with the sensors and actuators. The given software tools have a layered architecture, as displayed in Figure 2. The lower layer runs on the microcontroller and has the role of controlling the dc motor and the servomotor. This layer is named the *output layer*. The higher layers like sign detection, path planning, and decision making run on the SBC. The microcontroller accepts external commands through asynchronous serial communication (UART), which connects it to the SBC.

The microcontroller also grants direct interfaces to the connected sensors or drivers, such as the encoder and the motor driver. We integrated a control-loop mechanism to stabilize the rotation speed of the motor, where a proportional-integral-derivative controller drives the dc motor based on the encoder's measurements. In addition, a linear converter is implemented to control the steering system through the servomotor. In this way, higher-level tools manage the car's linear speed or front-wheel rotation by sending a simple message to the microcontroller based on the decision-making modules.

The higher-level tools are also open source software components running on the RPi. This solution enables the participants to easily customize their projects. In Figure 2, the preintegrated tools are depicted with orange containers on the input layer. These software components provide interfaces to the car's sensors, the infrastructure, and other cars. Each tool has a different purpose, but they can also be integrated into a single project, maintaining the flexibility and modularity of the project. The following software components are also integrated: a car tracker, camera handler, traffic light tracker, GPS tracker, and BNO055 listener. The purpose of the car tracker is to acquire the position and orientation of other cars on the track by interacting with the localization system. The GPS tracker aims to obtain car coordinates based on the indoor localization system, in a manner similar to the one provided by a real-life GPS. The camera-handler component grants access to images from the RPi camera. The BNO055 listener is an interface to acquire measurements from the smart sensor, like acceleration, rotation, or orientation, while the light tracker communicates with the traffic lights to determine their current state.

Racetrack Environment

To create a smart city-like infrastructure, we make a series of open source tools available for the students, which include their respective application programming interfaces and guides for how to use and test them. The tools touch the topics of secure communication, server validation, package description, and information speed and accuracy. These systems consist of a camera-based localization system (which simulates real-life GPS), V2V communication and vehicle-to-infrastructure communication, smart traffic lights, and a traffic monitor.

The localization system is composed of a series of RPi computers connected to a server via the Robot Operating System platform [5], [6]. With the help of cameras, the RPis detect the ArUco (minimal library for Augmented Reality applications) markers (which are placed on top of the cars) to generate coordinates and rotation in the frame view and communicate them to the server. The server knows the position and rotation of each camera, transposes the coordinates of the quick response on the camera frame to the track coordinates, and shares them with the respective clients (competitors' cars). The information is used for planning the car path based on its objectives.

For the case of the V2V communication, the moving obstacles (identical to the competition vehicles) broadcast their own ID, coordinates, and rotation on the network via User Datagram Protocol (UDP) messages (for increased communication speed). This information is used by the competitors to validate information like the presence in the intersection of other cars, the overtaking of cars, or the speed of the cars on the track.

For the case of the V2V communication, there are two options. On the smart traffic lights side, a similar approach to the V2V communication is used: the traffic lights stream their IDs and state on the network via UDP messages. On the competitors' cars side, the traffic monitor is used so that they send the coordinates of each encountered obstacle to the server. The data in this case are also used for the evaluation of the team run.

For environment control, monitoring, and evaluation, we developed a competition application named *Masterix*. The application, together with other automatic systems on the track, such as a moving pedestrian and a stopwatch, ensures that no human interaction is necessary during the runs. Through this application, the entire map is seen from a bird's-eye view. The cars on the track are displayed with the coordinates taken from the localization system, and the states of the traffic lights are shown with the states they stream on the network. The time of the car on the map, activated/deactivated by the remote stopwatch, is also displayed. The position of the pedestrians and the traffic signs can be set and changed using the same application. It can also control the cycle times of the traffic lights and activate the movement of the

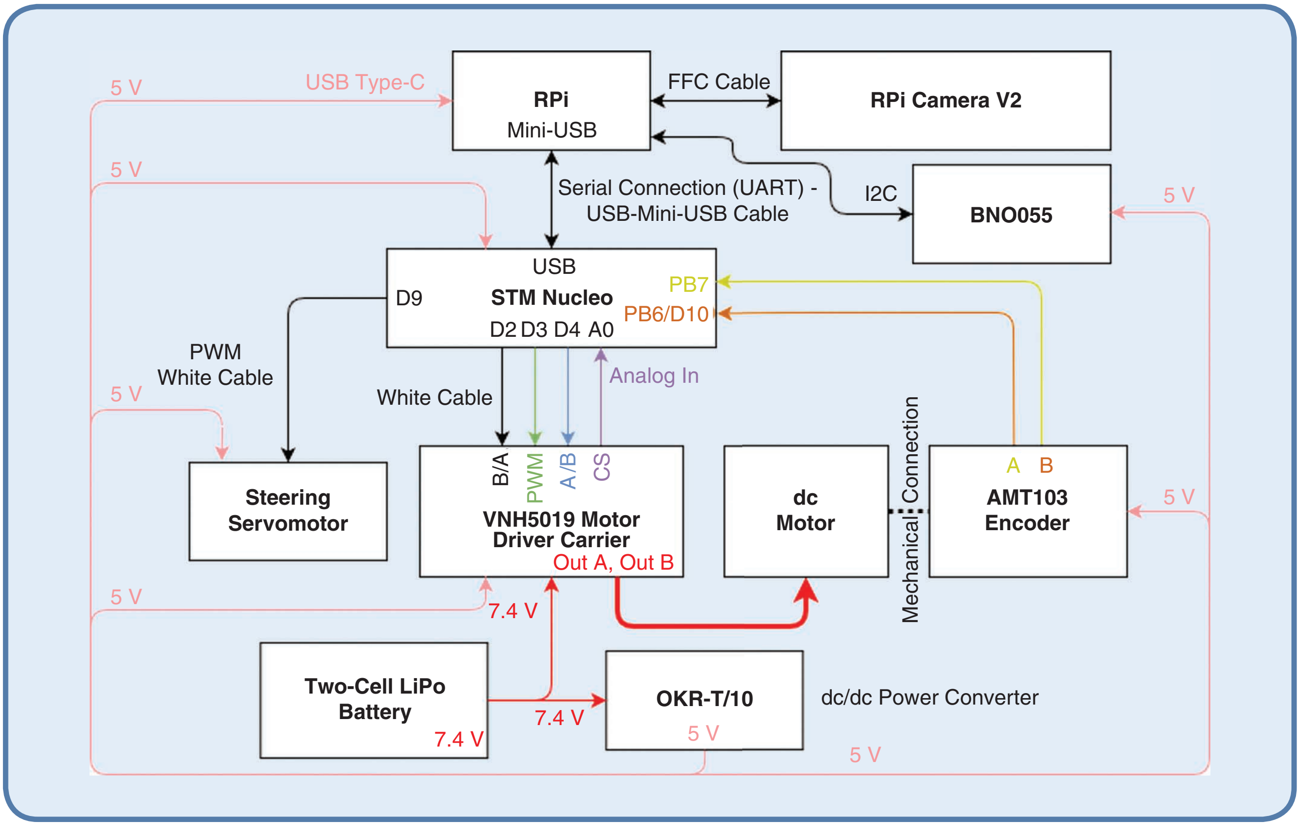


FIG 1 The car kit hardware block diagram. A/B: A or B; B/A: B or A; CS: chip select; D: digital; FFC: flexible flat cable; PBx: Port B pin x of microcontroller; PWM: pulsewidth modulation.

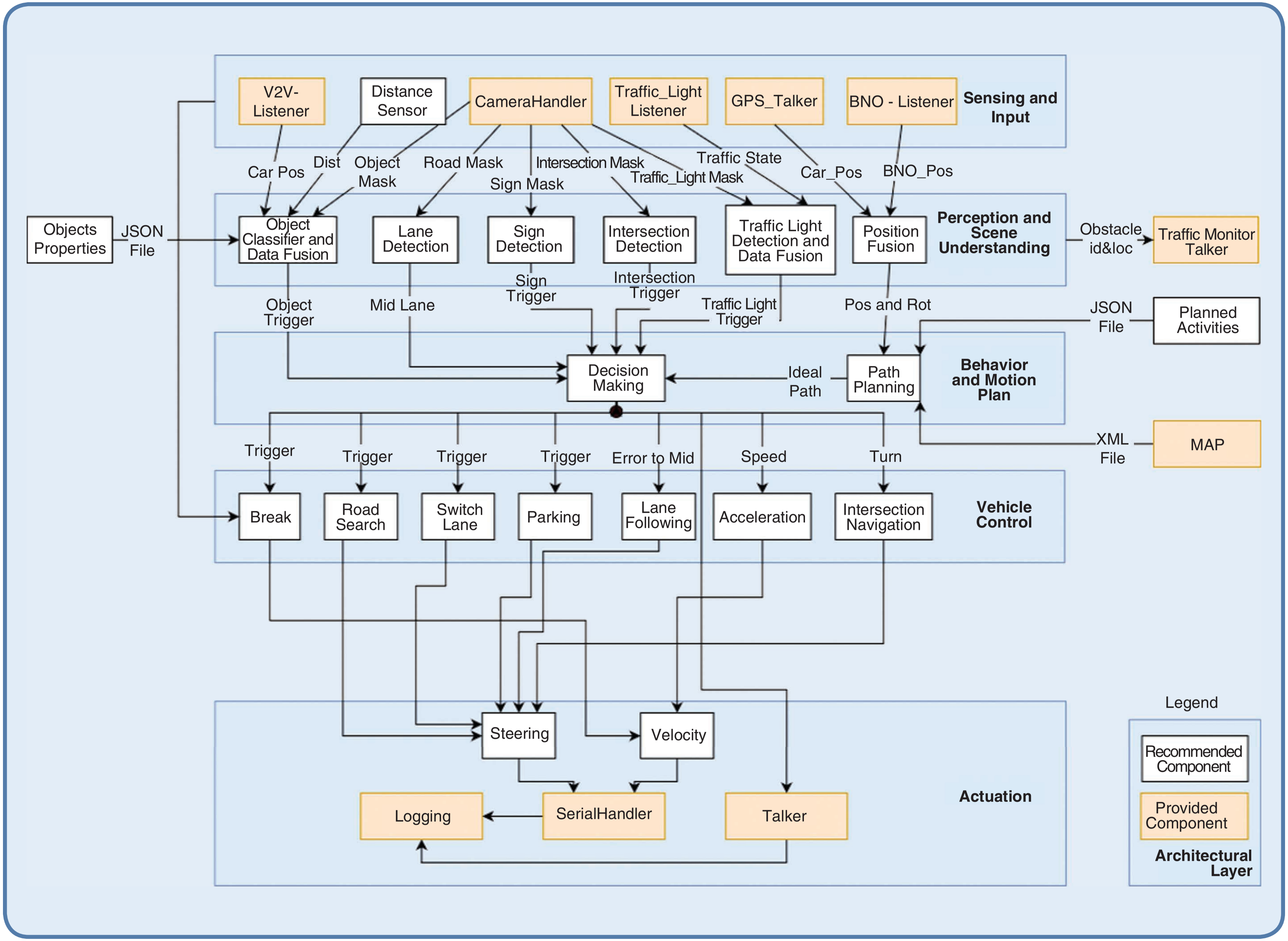


FIG 2 The suggested software architecture. dist: distance; id&loc: identity and location; JSON: JavaScript Object Notation; MAP: Manufacturing Automation Protocol; mid: middle; pos: position; rot: rotation.

pedestrian. The view from the application is also used to display the general overview to the audience for a better understanding of the team run.

Digital Twin

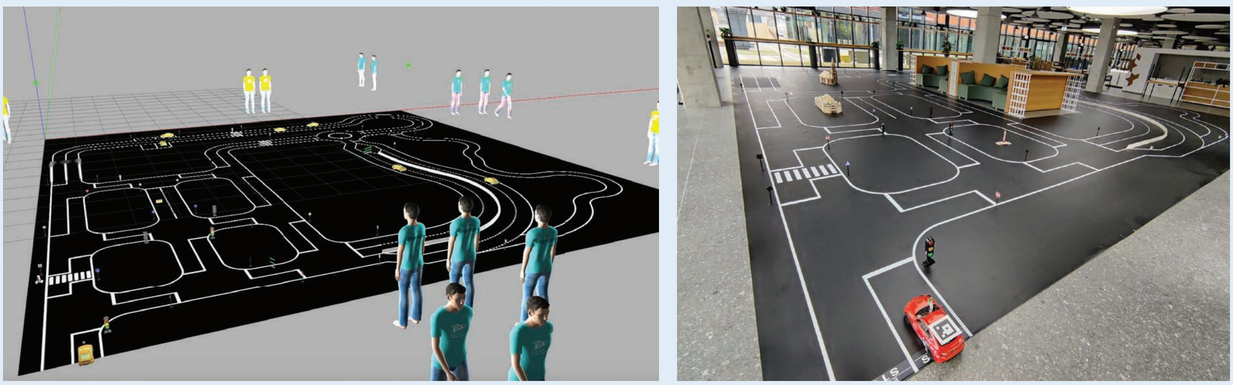
Many teams struggled on their first face-to-face encounter with our track due to the fact that the entire environment was not well understood and reproduced. Omitting some aspects, the challenges included the track dimensions, the differences in lighting, how all of the communication components could be used, how the entire communication works, and how to test their overall control algorithms. With the purpose of offering the students a replica of the entire competition environment and infrastructure, a Gazebo open source simulator, presented on the left of Figure 3, is provided. The simulator was developed fully in-house (3D models, textures, functionalities, interfaces, and plugins). The teams can use it to test the overall car performance without the need to build a physical competition replica. The simulator is also easily adjustable based on the needed testing/development case: each class of elements can be imported only when needed, the graphic user interface can be deactivated, and there is a recording mode for when the users want to register the car run.

Impact

After four editions, which grew in terms of the geographical distribution of the invited teams as well as the number of students and teams involved, the impact of the competition in the academic world has started to be more visible. The results are following our vision of contributing to the transformation of education toward the future of work in autonomous mobility. An international and diverse background among the participants is mandatory for ensuring an environment that fosters growth. To support this, our competition has grown every year—starting with a local 12-team competition in Cluj-Napoca in 2018 to a European-wide one in 2021 with 50 accepted teams from 10 countries (Figure 4).

For the participants themselves, the competition is a fast-lane development opportunity in terms of technical and nontechnical competencies. Working in teams for six months, the students are responsible for technical development; project management; interfacing with other teams, their university mentors, and organizers; and creating promotional materials and promoting their teams. The competition is a research and educational program in itself. Fun is another very important aspect that we emphasize. Things are don't stop here though: we encourage participating students to use their results as bachelor's or

FIG 4 The geographical distribution of the BFMC2021 teams.



(a)

(b)

FIG 3 The (a) simulator and (b) real-track view.

master's degree projects and even as a start for their Ph.D. degree theses. Many have presented their BFMC results as bachelor's or master's theses, and a few have written scientific articles presented at student conferences or have become mentors, as alumni, while working toward Ph.D. degrees. A selected few were invited by their university mentors to remain in their universities after graduation to teach programming courses.

Such a competition can create deep changes in universities when academic mentors believe in it and see it as a chance to steer their research and academic activities toward challenging, future-oriented AD topics. From translating the competition requirements into the structure of new courses to adopting AD as a new research direction, the challenge is laying a strong foundation in the academic world.

The necessary step for increasing the recognition and impact of the competition is the partnership with the IEEE ITSS, established in 2021 for an indefinite period, where the two partners have assumed the responsibility to develop new benefits for the participants and promote the event together. Among these benefits are providing workshops and additional educational components as well as access to publishing the participants' work in IEEE conference proceedings or journals.

Our students make this competition a success. Therefore, we briefly list a few technical approaches of the participant teams—approaches that we have enjoyed during the three previous editions. Each team chose to focus on the direction its members found most important and best-suited for their ideas and components while opting for basic solutions in other areas. The approaches were very diverse, and some were similar to solutions found in real-life AD vehicles: parallel hardware processing on field-programmable gate array for image recognition, parallel processing on CPU, parallel processing on the graphic board (CUDA), heat maps for road markings, machine learning for traffic elements, complex and adaptive packages for better improvements over time, and a focus on redundancy and fail-safe features.

BFMC started as a framework for students to develop future competencies for AD and as a vision to impact the academic world at a much deeper level. After four editions, we can say that the results confirm we are on the right track, and we are glad to be able to continue this success story.

About the Authors

Nandor A. Kilyen, Rares F. Lemnariu, George D. Moisare and Ionut Muntean are all with the Innovation Group, Bosch Engineering Center Cluj and the Technical University of Cluj-Napoca, Romania.

Yaobin Chen is with the Purdue School of Engineering and Technology, Indianapolis, IN 46202 USA, He is also ITSS VP for Technical Activities.

Brendan Morris is with the University of Nevada, USA. He is also ITSS VP for Conferences.

References

[1] A. M. de Souza, H. F. Oliveira, Z. Zhao, T. Braun, L. Villas, and A. A. F. Loureiro, "Enhancing sensing and decision-making of automated driving systems with multi-access edge computing and machine learning," *IEEE Intell. Transp. Syst. Mag.*, early access, Mar. 2020. doi: 10.1109/MITS.2019.2953513.

[2] Ö. Ş. Taş, F. Kuhnt, J. M. Zöllner, and C. Stiller, "Functional system architectures towards fully automated driving," in *Proc. 2016 IEEE Intell. Veh. Symp. (IV)*, Gothenburg, Sweden, pp. 304–309. doi: 10.1109/IVS.2016.7535402.

[3] N. Ueter, K.-H. Chen, and J.-J. Chen, "Project-based CPS education: A case study of an autonomous driving student project," *IEEE Des. Test.*, vol. 37, no. 6, pp. 39–46, Dec. 2020. doi: 10.1109/MDAT.2020.3012085.

[4] A. Carballo, D. Wong, Y. Ninomiya, S. Kato, and K. Takeda, "Training engineers in autonomous driving technologies using autoware," in *Proc. 2019 IEEE Intell. Transport. Syst. Conf. (ITSC)*, Auckland, New Zealand, pp. 3347–3354. doi: 10.1109/ITSC.2019.8917152.

[5] A. Mancini, E. Frontoni, and P. Zingaretti, "Embedded multisensor system for safe point-to-point navigation of impaired users," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3543–3555, Dec. 2015. doi: 10.1109/TITS.2015.2489261.

[6] S. Guirguis, M. Gergis, C. M. Elias, O. M. Shehata, and S. Abdennadher, "ROS-based model predictive trajectory tracking control architecture using LiDAR-based mapping and hybrid A* planning," in *Proc. 2019 IEEE Intell. Transport. Syst. Conf. (ITSC)*, Auckland, New Zealand, pp. 2750–2756. doi: 10.1109/ITSC.2019.8916783.

ITS