Periodica Polytechnica Transportation Engineering

Novel Traffic Light Control Concept for Cooperative, Connected and Automated Mobility (CCAM) Testing

Tamás Wágner[1*], Balázs Varga[1], Tamás Tettamanti[1,2], István Varga[1,2]

[1] Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[2] Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control, Kende u. 13–17., H-1111 Budapest, Hungary
[*] Corresponding author, e-mail: wagner.tamas@edu.bme.hu

Received: 10 October 2025, Accepted: 12 March 2026, Published online: 17 September 2026

Abstract
Automated vehicle testing and proving processes impose special requirements for traffic light control. Connected and Automated Vehicle (CAV) functions and related traffic control mechanisms must be tested and validated in a fully flexible and controlled test environment requiring specific technical features compared to the classical traffic management systems. In the paper, a novel concept for traffic light control is introduced together with its technical realization aspects on both intersection and central management levels, i.e., a networked control system has been designed. The control scheme realizes a fully adaptable automotive industry application to support not only the CAV but the Cooperative, Connected and Automated Mobility (CCAM) testing in general. The real-time performance has been demonstrated and evaluated by various test use cases, where the network latency as well as the computation performance have been investigated via real-world-like scenarios. Finally, as a proof of concept, the developed traffic light control system has been tested successfully as a part of a digital twin framework at the ZalaZONE Automotive Proving Ground.

Keywords
traffic light controller, traffic signal control, automated vehicle testing, automotive proving ground, CCAM

1 Introduction

Traffic lights (or traffic signals) are special control devices at road intersections or pedestrian crossings to realize safe and efficient traffic flow control (Papageorgiou, 2004). The world's first traffic light (with gas-lit signals) was designed by a railway engineer (J. P. Knight) and started operating in London in 1868. At the beginning of the 20th century the first electric traffic lights appeared in the USA (devices practically similar to today's traffic signal heads). Since then, this technology has spread out and developed intensively due to the persistent expansion of road transport.

To be precise, the notion of the Traffic Light Controller (TLC) is to be defined, i.e., traffic lights are just signalization devices, and the control of the signals is ensured by a logical hardware called TLC. The first TLCs were realized based on relays adopted from the railway signaling systems. Later, in the 1960s the electronic realization of TLCs' logic appeared. Then, mainly from the 1980s, the microcontroller-based TLC realization emerged.

Meanwhile, the classical role of the TLCs has not changed at all. Practically, the TLCs always intend to solve a multi-objective optimization problem, i.e., TLCs aim to fulfill the following goals at the intersection level or even at the network level (Qadri et al., 2020):

- reducing general travel costs (via the minimization of queue length, delay time, travel time, number of stops, fuel consumption);
- minimizing the accident risk;
- mitigating of pollutant and noise emission of road traffic;
- ensuring public transport priority;
- optimizing the road capacity utilization.

The evolution of traffic signaling has come a long way from "policeman switching" to today's network-wide adaptive traffic control. Still, the main structure of urban road traffic management remained the same, illustrated by Fig. 1.

An advanced traffic management system is typically represented by real-time applications, e.g., traffic-responsive traffic signaling or traffic-dependent route guidance, where traffic management logic can react to any

Cite this article as: Wágner, T., Varga, B., Tettamanti, T., Varga, I. (2026) "Novel Traffic Light Control Concept for Cooperative, Connected and Automated Mobility (CCAM) Testing", Periodica Polytechnica Transportation Engineering, https://doi.org/10.3311/PPtr.42458

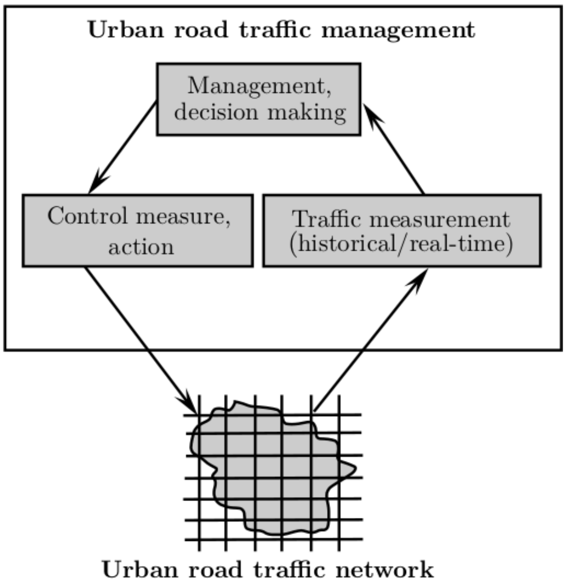

Fig. 1 General structure of the urban road traffic control

sensors or external interventions (Siemens AG, 2013; Tettamanti et al., 2014).

With the advent of Connected and Automated Vehicles (CAVs) the tasks of traditional TLCs extend (Horváth et al., 2019; Szűcs and Szűcs, 2024).

From now, traffic control devices must communicate or even cooperate with CAVs.

At the same time, traffic controllers must always ensure the traditional tasks for human drivers, vehicles on lower levels of automation, pedestrians, cyclists, and vulnerable road users (VRU), in general. While on freeways automated driving is practically solved (only policy and liability issues restrain the real-world application), fully autonomous intersections (when crossing traffic flows are controlled based on CAVs' communication without traffic lights (Carlino et al., 2013; Dresner and Stone, 2008; Hausknecht et al., 2011; Tettamanti et al., 2017)) will remain only a special case as it assumes no human drivers or VRUs on the road. This is simply due to the extremely complex urban context where at urban signalized intersections humans will always need zebra crossing and thus traffic light control (providing the green signal for safe crossing). The more, beside the pedestrians other human micro-mobility users, such as cyclists, e-scooters, will always be present in urban traffic.

Therefore, there is a real need for a flexible, real-world testing environment to push forward CCAM technology-based traffic control (Coppola et al., 2023; Li et al., 2023; Szalay, 2021). This article presents a significant extension of our preliminary research, which introduced the foundational concept of this flexible control system (Wágner et al., 2024).

According to the above, the main contributions of the research and development introduced in the paper are as follows:

1. A comprehensive method for a fully flexible, PLC (Programmable Logic Controller) based traffic management system is developed for future CCAM testing or homologation processes.
2. A robust communication technology applied in the manufacturing industry is adopted and successfully applied for traffic management applications.
3. The developed system is adaptable for any test tracks.

2 Requirements and standards for the Traffic Light Controller (TLC)

The majority of traffic management devices have the potential to endanger human life in the event of malfunction or failure and are, therefore, considered safety-critical devices. Consequently, the technical requirements of these systems are heavily regulated at both the international and national levels by laws and standards. For the novel type of TLC produced in our research, it is essential to comply with all legal and technical requirements. Therefore, it is necessary to investigate the regulations that apply to these devices and their viability for compliance.

In the past decades, in order to facilitate the development of different safety-critical systems for the global market, various international organizations have been founded. In order to harmonize the various technical standards and specifications across the European continent, the European Organization for Public Standards (CEN: Comité Européen de Normalisation) was founded in 1962. CEN's mission is to create a single set of norms and standards in the EU (European Union) and EFTA (European Free Trade Association) countries, under the name of European Norms (EN). Member States commit themselves, by adopting various international agreements, to respect and integrate these standards into national regulations. In the following, we will primarily focus on the European standards, but it should not be forgotten that this general framework also applies to traffic management systems in the Americas and Asia.

The following standards pertain to the operation of traffic management systems:

- Traffic signal controllers - Functional safety requirements (CEN, 2015a)
- Traffic control equipment - Signal heads (CEN, 2015b)
- Road traffic signal systems (CEN, 2018)

The standards in question are of considerable length and complexity, and therefore, only a selection of them will be discussed in the following paragraphs. In particular, those elements that pertain to the most critical safety functions. In the event of a severe fault occurring during the operation of the traffic management system, the system must activate a fail-safe state that can guarantee safe operation.

The standards permit two operational modes: a flashing yellow signal or the deactivation of signals at the intersection.

Furthermore, the standards delineate what is meant by a "critical failure".

- Conflicting green signals: In this case, green signals are displayed at an intersection that allows conflicting traffic movements.
- Failure to display a red signal: At least one red signal is not displayed on the signal heads.
- Unintended signal: A signal is displayed that was not intentional.
- Failure to display the correct signal sequence: The sequence of signals displayed is incorrect, i.e. the sequence of signals does not comply with international standards.
- Failure to provide correct signal timing: An error occurs in the timing of the individual phases.
- Failure of safety timings: Failure of any safety time setting (intermediate green time or minimum green time).

One of the CCAM functions to be used in the future is the so-called SPaT (Signal Phase and Time) and MAP V2X (Vehicle-to-Everything) functions (Lin and Wei, 2023; Weingart, 2014). The implementation of these functions is essential in modern traffic management equipment, and therefore, the ISO/TS 19091:2017 standard (ISO, 2019), which defines the basic properties of V2X communication protocols, should be taken into account. In addition, if V2X communication systems also implement Dedicated Short-Range Communication (DSRC) technology, the (Federal Communications Commission, 2022) suggestion is also important in addition to the Message Set Dictionary for DSRC (J2735_202007) (SAE, 2020).

3 New challenges of TLCs in the light of CCAM testing needs

Traditionally, traffic light realizations have been predicated on the notion that traffic signals are exclusively perceived by human drivers. Consequently, all pertinent standards, such as the radiation angle of traffic lights (CEN, 2015a) and the positioning and quantity of signal heads at road crossings, are tailored to human perceptual capabilities.

Similarly, automotive testing has historically focused on in-vehicle functions and their safety.

However, with the advent of automated vehicles, there is a pressing need to fundamentally reassess and redefine the traditional approach to the production and operation of TLCs. This technological shift aims to facilitate cooperation between TLCs and vehicles, achievable through various communication modalities:

1. TLCs broadcast messages to road users (vehicles or even pedestrians) who can then utilize this information for their own operations.
2. Road vehicles relay information back to the TLCs e.g., bus priority.
3. In the future, communication shall be fully bidirectional/mutual between TLCs and road users.

Initially, vehicle sensors, primarily advanced camera-based image recognition algorithms, facilitated the perception of traffic signals. However, it is evident that image recognition of traffic signals cannot always be flawless or wholly reliable due to environmental factors (e.g., rain, snow, fog)(Chung et al., 2020). In scenarios where both machines – the traffic signal controller and the highly automated vehicle – can communicate and operate autonomously, reliance solely on perception-based communication is suboptimal. Therefore, the most effective solution involves the use of wireless communication methods, such as DSRC or Cellular V2X, to facilitate interaction between these entities.

In response, not only the concept of traffic control but also the associated testing methodologies and devices must undergo a fundamental transformation. The objective of the novel traffic light control system is twofold. On the one part, a system is to be developed that complies with all existing traffic light control standards while accommodating the requirements of highly automated mobility. On the other part, a fully flexible testing framework shall be deployed, offering a digital twin environment for the testing and proving processes of future CCAM developments.

4 Novel concept for a flexible traffic light control system

In this section, we outline a concept for a traffic light control system for testing mixed traffic scenarios with highly automated vehicles, practically CCAM application testing.

The system must accomplish two conflicting goals: it must be safe and reliable adhering to all the standards listed in Section 2 while providing an interface for flexible configuration for testing various traffic scenarios. This flexibility is a fundamental paradigm shift from traditional traffic control systems. Traditionally, traffic light programs are designed and supervised by traffic engineers. When it comes to testing CCAMs, specific scenarios (including custom signal programs) are conceived by automotive OEMs or suppliers with limited expertise in signal program design. Therefore, an intuitive and ergonomic tool has to be provided to the users that allows easy modifications to the traffic signal program while having all the necessary safeguards. However, the system has also been equipped with an option to switch to test mode, where these safeguards can be disabled.

4.1 Central software for traffic control

The central traffic management software is a crucial component of our system. It cannot function independently, without TLC units. Constant communication between the central software and the various system elements, as displayed in Fig. 2, is crucial for ensuring the proper functioning of the system.

The previously stated requirements emphasize the utmost importance of safety, reliability, and cybersecurity. Thus, during system development, safety and real-time performance are the main considerations. Due to these reasons, we employ a modern, safety-oriented programming language, Rust (Rust Foundation, 2024). This language's key advantage is its unique approach to memory safety, while its performance is close to that of the C language (Blandy et al., 2021; Bugden and Alahmar, 2022). Moreover, the Rust language has another remarkable feature, that it allows the use of libraries written in C, which can be a significant cost saver for companies (Rust-Lang, 2024).

For testing purpose, the traffic light control system can operate in various modes:

- In Static Mode, the traffic lights execute a pre-defined (default) phase plan. Only the system designers and maintainers are authorized to modify these phase plans.
- In Custom Mode, the traffic signal controller executes a user-generated phase plan. In this mode, the user has the option to disable the safety features that are implemented on PLCs. Section 4.3 contains a detailed description of these safety functions.
- In Interactive Mode, the user can control the signal lights in real-time through a command line or GUI in a dynamic fashion. Only two predefined operations are provided alongside to the real-time interaction: Fully deactivated traffic lights and error signaling traffic lights (blinking amber lights). A predefined phase plan is not provided.

The central software manages the aforementioned modes. The software establishes a connection between the user and the TLC (in this case a PLC) and activates the selected mode.

If a program in Static Mode is used, the user cannot intervene but only observe the events. In other words, communication in this case is one-way from the TLC/PLC to the user. During dynamic and interactive modes, the user has a high level of control over the system, with bidirectional communication. These two modes give the user so much freedom that scenarios can be performed that would otherwise not be possible with a standard TLC. For example, it is possible to violate the intergreen matrix or display conflicting green signals.

The central software communicates with all PLCs (for more details on PLCs, see Section 4.2). Additionally, the central software constantly monitors the state and output of the PLCs, enabling immediate detection of any malfunctioning devices. To ensure reliable communication between PLCs and the central software, a communication protocol called OPC Unified Architecture (OPC UA) is used. This platform-independent protocol can transmit large amounts of data securely and dependably (OPC Foundation, 2019).

To establish a flexible traffic management center (TMC), the requirements and functions have to be defined in detail. The control and monitoring of traffic signals must be possible through the central control software.

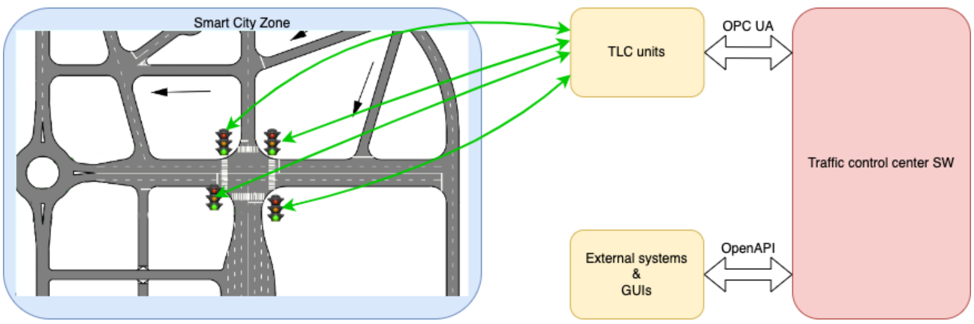

Fig. 2 Concept for the flexible traffic management center

Fig. 2 depicts the concept for the flexible TMC. The data on the objects in the managed area are transferred to the central control software via a data interface. This allows the control of individual lights as separate objects. The central control software manages the traffic signals via an open source Application Programming Interface (API) that must be accessible for external systems as well.

The flexible TMC includes two levels of control:

1. via an API: operation based on the commands from optional external software (e.g., MATLAB script, Python script);
2. via a GUI (graphical user interface): operation based on the commands given on a GUI.

The TMC enables the creation of an unlimited number of digital twins for each intersection, a capability implemented through a unique virtual PLC program. These virtual PLCs primarily aim to enable users to test various CCAM functions without requiring a physical PLC, thus facilitating simulations. Moreover, access to these virtual PLCs is designed to mirror that of physical PLCs, ensuring a seamless transition for users between virtual and physical environments.

4.2 Edge computing using PLCs

The PLCs utilized in our system are supplied by WAGO. This system leverages various distinctive features of WAGO PLCs, enabling standardized yet flexible operation. In developing the PLC programs, we place a strong emphasis on rigorous testing processes, especially considering that these programs are developed from the ground up. Our aim is to optimize memory usage and performance for each program, with all tests conducted in accordance with a well-defined strategy to ensure the system's safe and error-free functionality.

Despite the reliability and broad range of network interfaces offered by WAGO PLCs, it is crucial to design each program to function effectively in a fail-safe mode. We pay particular attention to scenarios involving communication failures between the TMC and the PLCs, ensuring that the system maintains operational capability in offline mode.

Moreover, an additional layer of operational safety is achieved by the fact that all instructions are processed directly on the PLCs. The TMC's role is limited to transmitting information from users to the relevant PLCs, where actual instruction evaluation occurs. This architecture ensures that the timing processes in the system are guaranteed to run accurately.

4.3 Safety functions and error handling

Traffic lights are safety critical devices that, if they malfunction, can endanger lives. Therefore, it is important to ensure that all safety requirements are met when designing their control systems, and to use multiple redundancy systems where necessary. For these reasons, the decision has been made to implement TLCs using PLCs. PLCs are already extensively used in the manufacturing, automotive, and logistics industries. Their reliability and ease in developing PLC programs make them an ideal choice.

PLCs include built-in safety features, such as data integrity checkers, timer watchers, and input/output cable break detectors, which eliminate the need for separate checking of the PLC manufacturer's libraries and function blocks used in the program, and are capable of interrupting the program in the event of an error. However, PLC programmers are responsible for higher level error handling.

In the control program presented, we have implemented two additional Program Organization Units (POU) for error detection and handling.

The intergreen time matrix monitoring POU - This POU's task is to monitor if enough time has elapsed between two green phases at the intersection for proper clearance and ensure that conflicting green phases have not occurred simultaneously.

Cable status monitoring POU - This POU's purpose is to detect if any cable is broken at the outputs. If a malfunction is detected, it alerts the central software and deactivates the traffic lights.

Although the user has the freedom to set custom signal programs, even violating the intergreen matrix or flashing every light of a signal head, safety checks are always in place, notifying the user.

It is important to note that occasional failures are common in various systems. Occasional system failures can result from various causes, including an overloaded processor, memory overflow, or power supply failure. Although the probability of these failures is minimal, they must not be overlooked. System failures can result in POUs taking longer to run and, as a result, timings in the PLC program can be corrupted. The PLC program can quickly recognize errors and halt the traffic light control, switching to an error state. The PLC program is designed to automatically restart the light control within one minute in such cases. However, it is possible that the device has malfunctioned to such an extent that a technician will need to be called to repair it. Thus, the system is designed to restart itself only a limited number of times, instead of attempting endless restarts.

Although the system can only be accessed through its API, which includes an integrated data type and value validator, further data integrity, plausibility checks and diagnostic checks are conducted in the central software and on the hardware to ensure reliable operation.

4.4 Cyber security

Nowadays, ensuring cyber security is a top priority, especially in smart city systems like this one. It is crucial to strike a balance between the requisite level of security and the processing capability of various devices.

The system was designed with the goal of creating a completely internet-isolated environment. However, for simulation-based testing, the virtual PLCs have to access the central system. To resolve this contradiction, the physical components (real PLCs) can only be accessed within the local network, which is protected by a whitelisting firewall. To access the system remotely, users require a Virtual Private Network (VPN).

All users must identify themselves using a bearer token that is valid for a limited time and only for authorized junctions. If a user is authorized to a specific junction, they can access its virtual counterpart too, in order to prepare test cases prior to live testing. Tokens are distributed by the supervisory staff at the proving ground, ensuring that human checks are always in place. Unauthorized access is strictly prohibited. The system can only be accessed through the official API and GUI.

There is a rate limit implemented for API calls in order to protect the system from Denial-of-Service (DoS) attacks.

All communication within the network is encrypted between the PLCs and the central server using OPC UA's built-in encryption features. Additionally, communication with the PLC requires authentication, which is also an OPC UA feature and only accessible locally or via VPN.

To prepare for possible cyber-attacks, the system generates a detailed log of all login data, allowing for analysis after an attack to identify and address system vulnerabilities.

4.5 Traffic-actuated signals

The proposed system is capable of traffic-actuated signal control. Although the traffic signal heads and sensors (e.g., inductive loop detector) are both connected directly to the PLC, the feedback loop is not realized locally. That is a reasonable choice, as vehicle testing scenarios may require custom traffic actuation and modifying the PLC program on the fly is not a good practice and might compromise safety. Conversely, users can switch to interactive control mode and periodically read the data of the sensor through the API. Then, users can freely tailor any sophisticated control algorithm that fits their research or testing needs. The traffic signal calculated by the controller can be fed back to the system via the API. Granting users authority over the control logic not only provides them with increased flexibility but also guarantees the server's integrity and shifts the burden of computationally complex algorithms to them. The only downside is that it may lead to a longer feedback loop, causing additional delays.

5 Evaluation and testing

At the final stage of the development process, we conducted a series of tests to verify the performance of the system. The primary focus of these tests was to evaluate the system's reliability, as well as its ability to handle heavy workloads and sustain real-time performance.

5.1 Network latency

In this test, the primary aim was to analyze the network latency in communications between PLCs and client applications, a critical factor for future V2X implementations. Since PLCs lack built-in V2X-compatible hardware, external network interfaces are necessary to enable these functions. Consequently, it is essential to verify that network hardware components and drivers in PLCs minimize latency. This is vital not only for efficient V2X communication but also for the real-time control of traffic lights from the traffic control center. The test involved using UaExpert software developed by Unified Automation GmbH (Unified Automation GmbH, 2023), conducted on a fully isolated wired Ethernet network with only the PLC and client computer connected. During this test, we monitored 13 OPC UA nodes.

The test results in Table 1 demonstrate the system's capability to maintain exceptionally low latency levels, even as the number of nodes increases. These results are notably promising, considering that they are significantly lower than the 100 ms latency threshold (Petrov et al., 2021) required for Vehicle-to-Infrastructure (V2I) communication.

Table 1 Network latency results using OPC UA communication protocol on Wago PLC

Number of monitored nodes	Average time per call [ms]
1	5.7687
5	13.6086
10	14.5056
13	14.8261

5.2 Network traffic load of communication

To determine the appropriate network design to ensure the reliable performance of the system, it was necessary to measure the amount of data being received and transmitted between network nodes. Our analysis relied on Wireshark (Wireshark Foundation, online), a software tool utilized to continuously monitor the amount and type of data passing through the client's network interfaces. The test configuration, where a PLC only communicated with one client, simulated the actual operating environment with 4 monitored traffic signal heads (12 OPC UA nodes). Based on the complexity of the intersection, it is quite easy to provide an estimate of the expected traffic throughout the system and, based on that, determine which network devices should be used.

The data presented in Table 2 indicate that the network traffic generated by the PLCs is notably low, suggesting that the system operates efficiently under conditions of low network load. Consequently, based on these findings, it is advisable to prioritize network devices that emphasize low latency over those focusing on high bandwidth.

Table 2 Average OPC UA communication network load

	Average network load
Without encryption or authentication	768 bytes/s
With encryption or authentication	954 bytes/s

5.3 CPU Load

For PLCs, maintaining a low load on the central processing unit is particularly important to prevent system overload. Should the load become excessively high, this could jeopardize the PLC's ability to execute programs within the designated cycle times, potentially resulting in hazardous system anomalies. Moreover, ensuring a low CPU load is essential to provide adequate computing power for the implementation of future extensions and additional functionalities.

Table 3 indicates that the PLC system's computational capacity is not fully utilized, suggesting that there is no risk of overloading the PLC, i.e., a considerable amount of unused computing capacity remains for the integration of additional functions. Nevertheless, the data reveal that OPC UA's authentication and encryption processes incur a marginally higher average CPU load, albeit this increase is negligible.

Table 3 Average and maximum CPU load of the PLC

	Average CPU load	Max CPU load
Without encryption or authentication	29%	30.01%
With encryption or authentication	29%	30.36%

5.4 API response time

During testing, we conducted a latency measurement test across the system. The test scenario consisted of communicating with the system via the API described in Section 4.1, see Fig. 3. The API call was initiated from a microcomputer equipped with both a network interface and a General-Purpose Input/Output (GPIO). We made a total of 100 API calls to turn on and off traffic light bulbs. We measured the time elapsed between the API call and the PLC output status change.

Table 4 displays the measurement results, while Fig. 4 shows the corresponding histogram. The results indicate that the time elapsed between the API call and the change at the PLC output is negligible, ensuring real-time operation (Tahir and Katz, 2021).

6 Proof of concept - traffic control center for ZalaZONE Automotive Proving Ground

The system previously discussed is specifically tailored to address the unique needs and requirements of the new generation of automotive proving grounds. All functionalities have been conscientiously designed and developed with these criteria in mind. An exemplary case is ZalaZONE in Hungary, which primarily aims to assist automotive companies in developing safe and modern CCAM functions. The ZalaZONE test track serves as an excellent demonstration and provides a robust proof of concept for our system(Szalay et al., 2019).

6.1 PLC-based traffic light controller

The innovative approach adopted in the design of ZalaZONE has involved the use of separate PLCs, shown in Fig. 5, to control traffic lights at each intersection within the Smart City zone, a departure from traditional TLCs. This novel method has enabled the realization of control systems that were previously only theoretical. The inherent flexibility in PLC programming has imposed minimal constraints on engineers, facilitating the creation of systems that are highly safe, user-friendly, and innovative. Such developments hold the potential to significantly expedite the development and market introduction of CCAM vehicles, while also acting as a versatile tool for automotive test engineers. A notable advantage is the ability of testers to deactivate the TLC's fault management mechanisms at the push of a button, thereby generating specific test scenarios unachievable with conventional

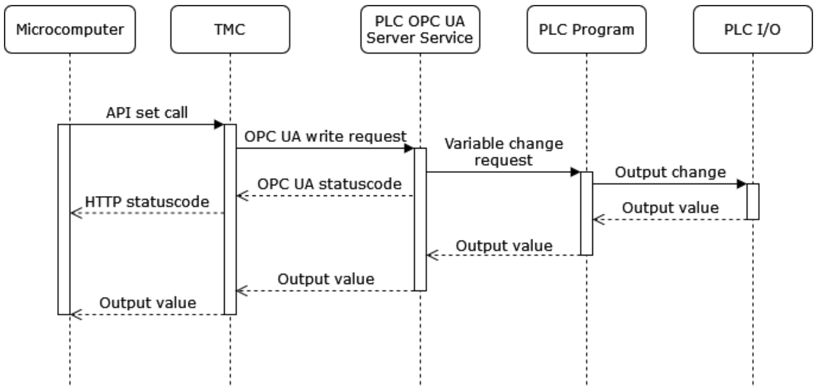

Fig. 3 API call trace between the client and PLC

Table 4 Results of the API latency test

Parameter	Value
The mean of API calls duration	14.8500 ms
The median of API calls duration	14.2051 ms
The mode of API calls duration	3 ms

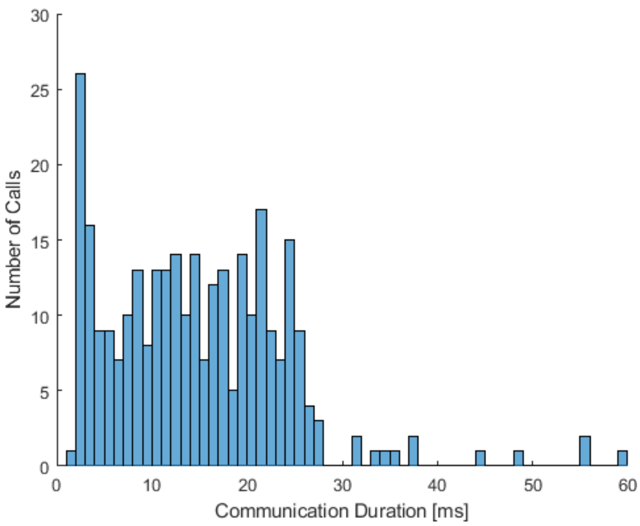

Fig. 4 Histogram of API calls latency

Fig. 5 PLC for traffic signal control (WAGO PFC200)

systems, like displaying both red and green signals simultaneously on a traffic light. Additionally, the system offers testers considerable freedom in using their programs, thereby substantially reducing their workload. While these functionalities may appear minor, they are crucial in ensuring that the system fulfills all requirements and substantially simplifies the research and development process. Building upon this innovative approach in hardware design, the next crucial phase is the development of software. This step involves tailoring the central control software to the specific needs of the PLC-based system, ensuring that each individual light is not only freely programmable but also integrates seamlessly with the advanced control systems developed for the ZalaZONE proving ground.

6.2 Software development

Following the precise specification of system requirements and functionalities, the focus shifts to software development. On the one hand, the central control software must be developed subject to the previously specified requirements, i.e., every individual light is freely programmable. This also means that all safety functions that are commonly used in road traffic management can be switched off (e.g., intergreen matrix, green conflict detection). On the other hand, the programming of traffic signal controllers at the intersections also has to be established. To fit into the initial concept, freely programmable Linux-based PLCs were procured to be deployed on the ZalaZONE proving ground. The PLCs account for the safe operation of the signalized intersection as well as constituting a link between traffic signals and the control functions of the TMC.

Laboratories of the university are creating a prototype intersection to be controlled by one of these PLCs.

As a basic function, the PLC software must constantly monitor that the traffic signals work according to the signal plan, even if the plan intentionally contains errors (e.g., green conflicts). Another basic function is to monitor the signal heads and their circuits to ensure they are not faulty.

6.3 Development of the digital twin

A parallel priority area of the signal control project aims to extend testing options with the application of traffic simulation tools. A Python-based simulation environment was established providing the framework for running different traffic scenarios on the model of the real test field and enabling the user to alter the settings while the simulation is running (Fig. 6). Two different traffic simulators have been deployed for realistic traffic simulation within the digital twin framework: a commonly used industrial tool (PTV Vissim) (Tettamanti and Horváth, 2023) and the open-source SUMO software (Lopez et al., 2018) frequently applied for research projects. The extension of the control framework is essential for the establishment of a digital twin with the predefined functions of the system. The digital twins will enable the traffic signal control system to be modeled in the traffic simulations in real-time at ZalaZONE's Smart City and University Test Track. In this way, mixed reality and Vehicle-in-the-Loop tests will also be possible (Horváth et al., 2019). A practical example for this is to virtually generate different traffic situations around a test vehicle using validated traffic simulators.

The development and testing of the integration between digital twins and PLCs were undertaken concurrently with the development of PLC-based lighting control programs. Throughout this phase, a distinct set of communication protocols, differing from those in contemporary systems, was employed. Leveraging the digital twins previously established, university and laboratory personnel commenced exploring implementation possibilities of various V2X communication protocols via PLCs.

In the course of testing, they successfully integrated the PLC into the OMNet++ (Varga, 2010) and SUMO simulation environments and managed to generate SPaT/MAP messages. Fig. 7 illustrates the base architecture of the system. In this environment, SPaT/MAP messages were emulated

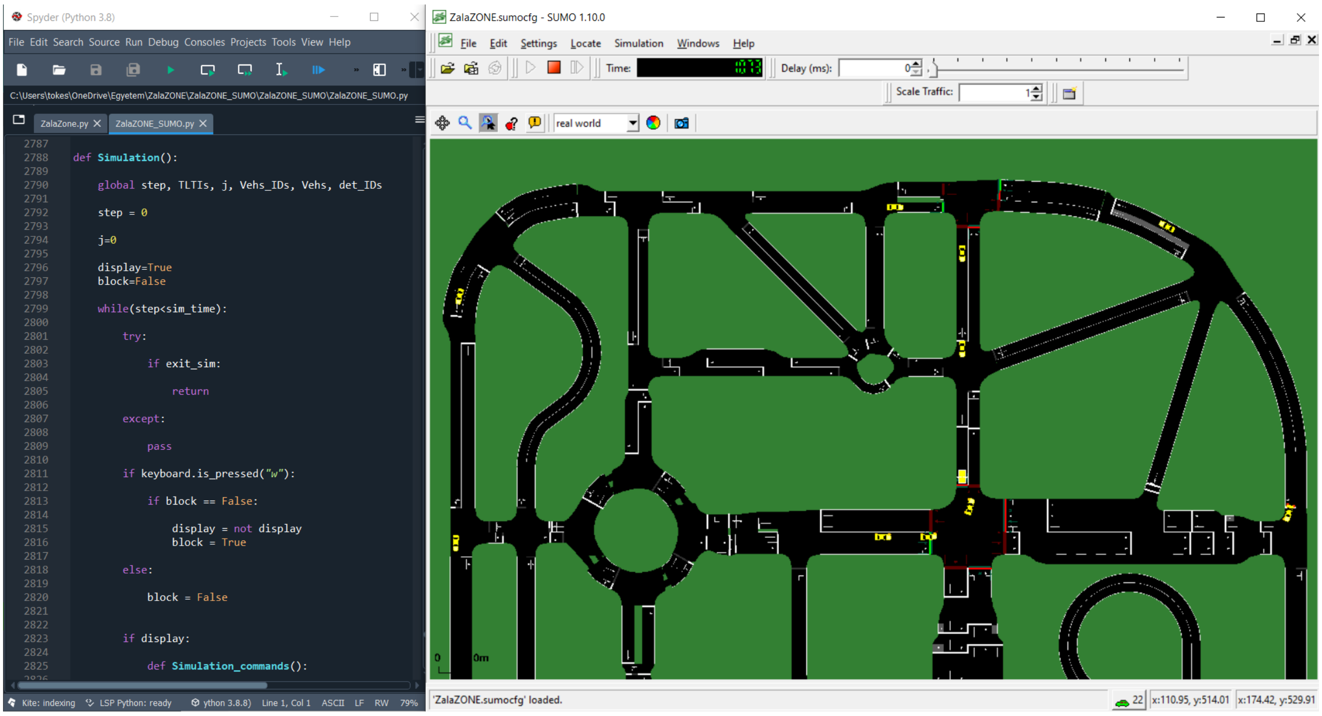

Fig. 6 SUMO traffic simulation controlled by Python (Python Software Foundation, online) script applied in the digital twin

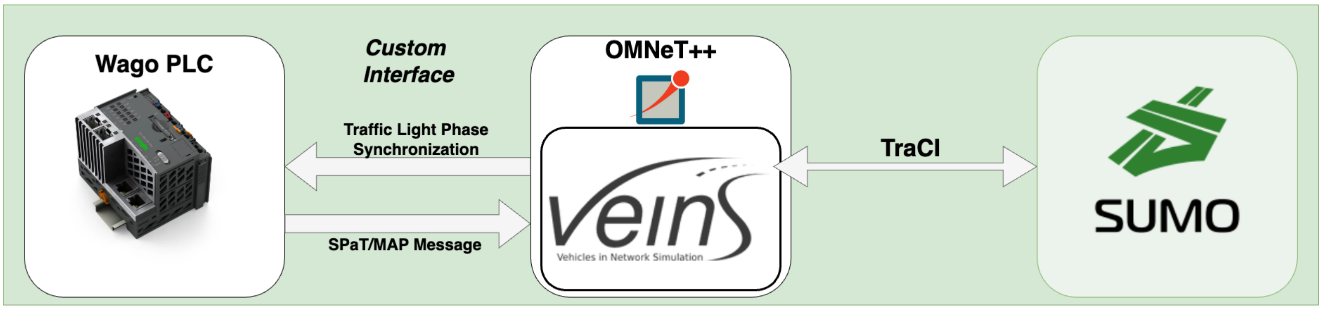

Fig. 7 SPaT/MAP integration on the PLC

and sent to the OMNeT++ simulation using a Python program based on information from the PLC program variables via a Wi-Fi access point (Wágner et al., 2023).

6.4 Advanced application: Case study on reinforcement learning-based traffic control

To further illustrate the flexibility and advanced capabilities of the presented traffic control architecture, we developed and deployed an artificial intelligence (AI)-based adaptive control system on the PLC-based framework. This case study demonstrates how the framework can be extended beyond predefined or manual test cases to accommodate intelligent control algorithms.

At the core of this application is a reinforcement learning (RL)-based agent, designed to optimize traffic light timing in real time. The goal of the agent was to minimize traffic congestion and reduce vehicle emissions by dynamically adjusting green signal durations according to incoming traffic.

The RL agent, based on a Double Deep Q-Network (DDQN) architecture (van Hasselt et al., 2015), was trained in a high-precision microscopic traffic simulation environment (SUMO). To ensure the realism of the training process, the simulation model was meticulously designed based on a T-junction, see Fig.8. The agent's inputs (observations) were designed for practical accessibility in real-world systems, consisting of vehicle counts and occupancy data measured by virtual inductive loop detectors placed on the approaches to the intersection. The reward function was designed to penalize excessive queuing and elevated CO_2 emissions, thereby encouraging both traffic efficiency and environmental sustainability.

One of the key innovations of this study was the successful deployment of the trained AI model directly on the WAGO PLC hardware, which forms the backbone of the ZalaZONE traffic control system. This was achieved by leveraging the unique hybrid architecture of the system:

- AI inference in a Docker container: The trained neural network ran in a lightweight Docker container on the PLC's embedded Linux operating system. A custom-developed, low-power application written in Rust was responsible for managing the inference process.
- Integration with deterministic safety logic: The AI agent communicates with the primary safety-critical control program running in the CODESYS (3S-Smart Software Solutions GmbH, 2018) environment via an internal TCP MODBUS interface on the localhost. In this setup, the AI agent provides a recommendation for an optimal signal phase plan based on the current traffic status and sends over the green phase lengths to the PLC. However, the final decision remains with the deterministic PLC program, which compares the proposal with predefined safety constraints, such as the intermediate time matrix and conflict monitoring rules described in Section 4.3.

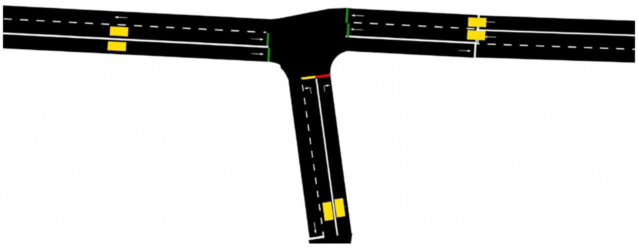

Fig. 8 T-junction in the simulation environment

This hybrid approach ensures that the system's operation fully complies with all safety requirements, even when using a non-deterministic AI for optimization purposes, and that a fail-safe state is triggered if an unsafe command is proposed.

To strictly guarantee the deterministic operation of the safety-critical PLC program, resource division is enforced at the operating system level. The PLC runtime executes with real-time priority (SCHED_FIFO), ensuring that it preempts the lower-priority AI container (SCHED_OTHER) immediately whenever control tasks require CPU resources (The Linux Fundation, online). Furthermore, the system stability is ultimately safeguarded by the hardware watchdog, which forces the outputs to a safe state in the event of any system anomaly, ensuring that the added software complexity does not compromise the safety-critical operation.

We compared the performance of the RL-based controller with a traditional fixed-time signaling scheme under different traffic loads, see Fig. 9. The results indicated that the AI controller significantly outperformed the baseline model, particularly under high traffic loads. It achieved a measurable reduction in total CO_2 emissions, confirming that the adaptive strategy results in smoother traffic flow and fewer unnecessary stops.

We also compared the CPU usage of the system during running the AI control algorithm. System performance when the without running the AI can be seen in Fig. 10 and in Table 5. In Fig. 11 and in Table 6, the system performance results can be seen during interference with the AI. The CPU usage in both cases were similar, however, the deviation of the CPU usage was noticeable higher with the AI control.

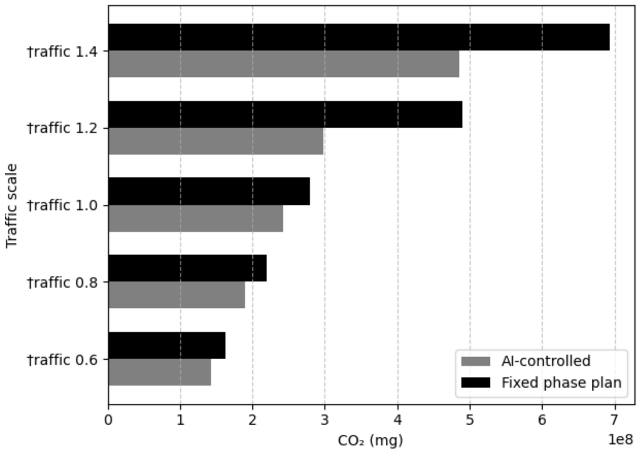

Fig. 9 CO_2 Emission comparison between the fixed phase plan and the adaptive AI-based traffic control

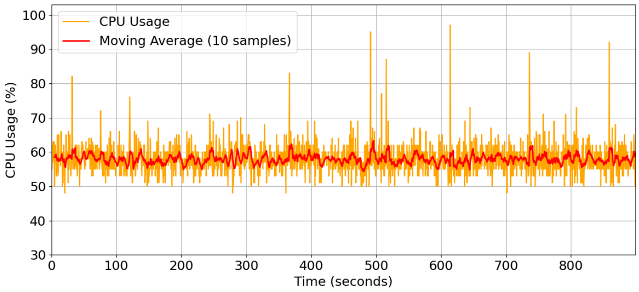

Fig. 10 CPU usage without AI in the first 15 minutes

Table 5 CPU measurement results without AI

Measurement	Value
Mean (%)	57.906071
Median (%)	58.000000
Standard deviation (%)	4.266656
Minimum (%)	48.000000
Maximum (%)	97.000000
25th Percentile (%)	55.000000
75th Percentile (%)	60.000000

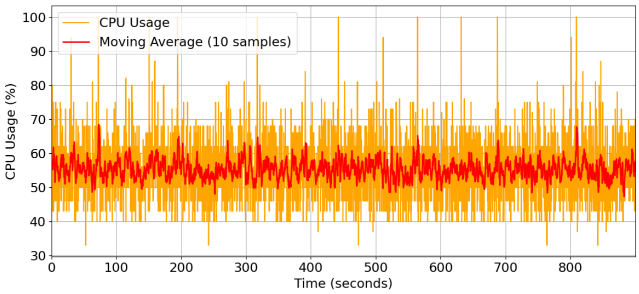

Fig. 11 CPU usage with AI in the first 15 minutes

Table 6 CPU measurement results with AI

Measurement	Value
Mean (%)	55.316068
Median (%)	56.000000
Standard deviation (%)	8.564599
Minimum (%)	33.000000
Maximum (%)	100.000000
25th Percentile (%)	50.000000
75th Percentile (%)	62.000000

7 Conclusion

In conclusion, the novel traffic light control system presented will revolutionize the testing possibilities for CAVs, offering unprecedented flexibility for test tracks. The system, which is based on the close cooperation of several different devices and subsystems, excels in testing real-time network communication, demonstrating its ability to work with a latency as low as 3 ms. In addition, the testing of PLCs that control the traffic light heads confirmed that modern technology is not only capable of handling these complex tasks but also provides extra computing power and thus the possibility to implement further features at a later stage.

The introduction of virtual PLCs and other support systems allows engineers to try out and refine their test cases before the real tests, increasing the efficiency and safety of the development process. These innovations not only make testing processes faster and safer, but also contribute significantly to the development of CAV technologies.

The system will soon be implemented on-site at the ZalaZone Automotive Proving Ground, with additional developments planned based on user feedback. This will mark a significant milestone in the system's development.

This research clearly demonstrates that modern technology and innovative approaches have the potential to transform vehicle technology testing, paving the way for safer and smarter transport systems of the future.

Acknowledgement

The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002);

Project no. 2024-2.1.1-EKÖP-00003 has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the EKÖP-24-3 funding scheme.

References

3S-Smart Software Solutions GmbH (2018) "CODESYS, (3.5.19.7)", [computer program] Available at: https://www.codesys.com [Accessed: 15 September 2025]

Blandy, J., Orendorff, J., Tindall, L. F. S. (2021) "Programming Rust: Fast, Safe Systems Development", O'Reilly. ISBN 9781492052586

Bugden, W., Alahmar, A. (2022) "Rust: The Programming Language for Safety and Performance", In: 2nd International Graduate Studies Congress (IGSCONG'22), Turkey. ISBN 978-605-73639-2-3 https://doi.org/10.48550/arXiv.2206.05503

Carlino, D., Boyles, S. D., Stone, P. (2013) "Auction-based autonomous intersection management", In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, Netherlands, pp. 529–534. ISBN 978-1-4799-2914-6 https://doi.org/10.1109/ITSC.2013.6728285

CEN (2015a) "EN 12675:2017 Traffic signal controllers - Functional safety requirements", European Committee for Standardization, Brussels, Belgium.

CEN (2015b) "EN 12368:2015 Traffic control equipment - Signal heads", European Committee for Standardization, Brussels, Belgium.

CEN (2018) "EN 50556:2018 Road traffic signal systems", European Committee for Standardization, Brussels, Belgium.

Chung, J. H., Kim, D. W., Kang, T. K., Lim, M. T. (2020) "Traffic Sign Recognition in Harsh Environment Using Attention Based Convolutional Pooling Neural Network", Neural Processing Letters, 51(3), pp. 2551–2573. https://doi.org/10.1007/s11063-020-10211-0

Coppola, A., D'Aniello, C., Pariota, L., Bifulco, G. N. (2023) "Assessing safety functionalities in the design and validation of driving automation", Transportation Research Part C: Emerging Technologies, 154, 104243. https://doi.org/10.1016/j.trc.2023.104243

Dresner, K., Stone, P. (2008) "A Multiagent Approach to Autonomous Intersection Management", Journal of Artificial Intelligence Research, 31(1), pp. 591–656. https://doi.org/10.1613/jair.2502

Federal Communications Commission (2022) "Dedicated Short Range Communications (DSRC) service" [online]. Available at: https://www.fcc.gov/wireless/bureau-divisions/mobility-division/dedicated-short-range-communications-dsrc-service [Accessed 09 September 2025]

van Hasselt, H., Guez, A., Silver, D. (2015) "Deep Reinforcement Learning with Double Q-Learning", Proceedings of the AAAI Conference on Artificial Intelligence, 30(1) pp. 2094–2100. https://doi.org/10.1609/aaai.v30i1.10295

Hausknecht, M., Au, T. C., Stone, P. (2011) "Autonomous Intersection Management: Multi-intersection optimization", In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, pp. 4581–4586. ISBN 978-1-61284-454-1 https://doi.org/10.1109/IROS.2011.6094668

Horváth, M. T., Lu, Q., Tettamanti, T., Török, Á., Szalay, Z. (2019) "Vehicle-In-The-Loop (VIL) and Scenario-In-The-Loop (SCIL) Automotive Simulation Concepts from the Perspectives of Traffic Simulation and Traffic Control", Transport and Telecommunication Journal, 20(2), pp. 153–161. https://doi.org/10.2478/ttj-2019-0014

ISO (2019) "ISO/TS 19091:2019 Intelligent transport systems – Cooperative ITS – Using V2I and I2V communications for applications related to signalized intersections", International Organization for Standardization, Vernier, Switzerland.

Li, H., Makkapati, V. P., Wan, L., Tomasch, E., Hoschopf, H., Eichberger, A. (2023) "Validation of Automated Driving Function Based on the Apollo Platform: A Milestone for Simulation with Vehicle-in-the-Loop Testbed", Vehicles, 5(2), pp. 718–731. https://doi.org/10.3390/vehicles5020039

Lin, W., Wei, H. (2023) "Cyber-physical models for distributed CAV data intelligence in support of self-organized adaptive traffic signal coordination control", Expert Systems with Applications, 224, 120035. https://doi.org/10.1016/j.eswa.2023.120035

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wiessner, E. (2018) "Microscopic Traffic Simulation using SUMO", In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, pp. 2575–2582. ISBN 978-1-7281-0323-5 https://doi.org/10.1109/ITSC.2018.8569938

OPC Foundation (2019) "OPC United Architecture" [online]. Available at: https://opcfoundation.org/about/opc-technologies/opc-ua/ [Accessed 09 September 2024]

Papageorgiou, M. (2004) "Overview of Road Traffic Control Strategies", IFAC Proceedings Volumes, 37(19), pp. 29–40. https://doi.org/10.1016/S1474-6670(17)30657-2

Petrov, T., Sevcik, L., Pocta, P., Dado, M. (2021) "A Performance Benchmark for Dedicated Short-Range Communications and LTE-Based Cellular-V2X in the Context of Vehicle-to-Infrastructure Communication and Urban Scenarios", Sensors, 21(15), 5095. https://doi.org/10.3390/s21155095

Python Software Foundation "Python Language Reference, (Version 3.12)", [computer program] Available at: https://python.org

Qadri, S. S. S. M., Gökçe, M. A., Öner, E. (2020) "State-of-art review of traffic signal control methods: challenges and opportunities", European Transport Research Review, 12(1), 55. https://doi.org/10.1186/s12544-020-00439-1

Rust Foundation (2024) "Rust Programming Language" [online]. Available at: https://www.rust-lang.org/ [Accessed 09 September 2024]

Rust-Lang (2024) "Rust-Lang/Rust-Bindgen: Automatically generates rust FFI bindings to C (and some C++) libraries" [online], Available at: https://github.com/rust-lang/rust-bindgen [Accessed: 22 October 2024].

SAE (2020) "J2735SET_202007 V2X Communications Message Set Dictionary Set", SAE International, Warrendale, USA. https://doi.org/10.4271/J2735SET_202007

Siemens AG (2013) "Adaptive network control Sitraffic Motion MX" [pdf], Siemens, Munich, Germany. Available at: https://otlet.budapest.hu/api/media/get/8afec693-16df-11ec-bdf0-0242ac1a0004 [Accessed: 22 October 2024]

Szalay, Z. (2021) "Next Generation X-in-the-Loop Validation Methodology for Automated Vehicle Systems", IEEE Access, 9, pp. 35616–35632. https://doi.org/10.1109/ACCESS.2021.3061732

Szalay, Z., Hamar, Z., Nyerges, Á. (2019) "Novel design concept for an automotive proving ground supporting multilevel CAV development", International Journal of Vehicle Design, 80(1), pp. 1–22. https://doi.org/10.1504/IJVD.2019.105061

Szűcs, H., Szűcs, J. (2024) "The Environmental Sustainability Potential of Autonomous Vehicles: An Overview", Periodica Polytechnica Transportation Engineering, 52(3), pp. 246–256. https://doi.org/10.3311/PPtr.23933

Tahir, M. N., Katz, M. (2021) "Performance evaluation of IEEE 802.11p, LTE and 5G in connected vehicles for cooperative awareness", Engineering Reports, 4(4), e12476. https://doi.org/10.1002/eng2.12467

Tettamanti, T., Horváth, M. T. (2023) "A practical manual for Vissim-COM programming in Matlab and Python – 6th edition for Vissim version 2023", [pdf], Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics. Available at: https://www.researchgate.net/publication/377209782_A_practical_manual_for_Vissim-COM_programming_in_Matlab_and_Python_6_th_edition_for_Vissim_version_2023 [Accessed: 22 October 2024]

Tettamanti, T., Horváth, M. T., Varga, I. (2014) "Road Traffic Measurement and Related Data Fusion Methodology for Traffic Estimation", Transport and Telecommunication Journal, 15(4), pp. 269–279. https://doi.org/10.2478/ttj-2014-0023

Tettamanti, T., Mohammadi, A., Asadi, H., Varga, I. (2017) "A two-level urban traffic control for autonomous vehicles to improve network-wide performance", Transportation Research Procedia, 27, pp. 913–920. https://doi.org/10.1016/j.trpro.2017.12.160

The Linux Fundation "Scheduling - Policy and Priority" [online], Available at: https://wiki.linuxfoundation.org/realtime/documentation/technical_basics/sched_policy_prio/start?s[]=sched&s[]=fifo [Accessed: 15 December 2025]

Unified Automation GmbH (2023) "UaExpert — A Full-Featured OPC UA Client", [online] Available at: https://www.unified-automation.com/products/development-tools/uaexpert.html [Accessed: 15 September 2024].

Varga, A. (2010) "OMNeT++", In: Modeling and Tools for Network Simulation, Springer, pp. 35–59. ISBN 978-3-642-12330-6 https://doi.org/10.1007/978-3-642-12331-3_3

Wágner, T., Ormándi, T., Tettamanti, T., Varga, I. (2023) "SPaT/MAP V2X communication between traffic light and vehicles and a realization with digital twin", Computers and Electrical Engineering, 106, 108560. https://doi.org/10.1016/j.compeleceng.2022.108560

Wágner, T., Tettamanti, T., Varga, B., Varga, I. (2024) "PLC-Based Traffic Light Control for Flexible Testing of Automated Mobility", In: Proceedings of Ninth International Congress on Information and Communication Technology, London, United Kingdom, pp. 177–185. ISBN 978-981-97-4580-7 https://doi.org/10.1007/978-981-97-4581-4_13

Weingart, J. (2014) "Standardisation of SPaT and MAP", presented at Intertraffic Amsterdam 2014, Amsterdam, Netherlands, Apr. 15–18.

Wireshark Foundation (2024) "Wireshark Software, (Version 4.6.4)" [computer program]. Available at: https://www.wireshark.org/