(IJACSA) International Journal of Advanced Computer Science and Applications,
Vol. 16, No. 2, 2025

Developing an IoT Testing Framework for Autonomous Ground Vehicles

Murat Tashkyn, Amanzhol Temirbolat, Nurlybek Kenes, Amandyk Kartbayev
School of Information Technology and Engineering, Kazakh-British Technical University, Almaty, Kazakhstan

***Abstract*—Autonomous ground vehicles play a crucial role in the Internet of Things, offering transformative potential for applications such as urban transportation and delivery services. These vehicles can operate autonomously in uncertain environments, making reliable testing essential. This study develops and analyzes a testing framework for autonomous ground vehicles, focusing on their motion control systems and electronic modules. The research reviews testing methods for printed circuit boards (PCBs), highlighting the need for JTAG testing implementation for vehicle modules. Functional testing was conducted on key components such as cameras, LiDARs, and wireless interfaces under various conditions. Results show that JTAG testing successfully detects faults with precise localization, while functional tests confirm stable component performance. Environmental tests revealed that most components perform reliably within optimal conditions, with failures occurring at temperatures beyond ±70°C and humidity levels exceeding 90% RH. The developed testing system enhances the reliability of autonomous delivery vehicles.**

Keywords—Autonomous vehicle; testing system; IoT; functional testing; electronic modules; delivery automation

I. INTRODUCTION

Intelligent autonomous vehicles are particularly relevant today. They can operate in uncertain environments and are of significant interest for a wide range of practical applications, including food delivery. This study focuses on the development and analysis of a testing system for unmanned delivery vehicles. Before such vehicles can be deployed in real-world operations, extensive testing is required to ensure their stability, reliability, and performance, as well as prior research on testing systems for unmanned delivery vehicles.

However, several challenges complicate the testing process for these vehicles.

- Testing in real-world conditions: It is essential to test delivery vehicle modules under conditions that closely mimic real environments. This includes verifying components under the influence of multiple adverse factors, such as humidity.
- Testing of PCB modules: Testing individual modules separately cannot guarantee the proper functioning of the entire delivery vehicle after assembly. Testing of all printed circuit boards (PCBs) as a unified system is necessary, since the vehicles are cyber-physical systems that perceive, process, and physically respond to information from the real world.
- Automation of testing: Manual testing introduces the risk of human error, which, even in a single component, can render the entire vehicle non-operational.
- Regulatory gaps: Unmanned delivery vehicles lack specific regulations and testing requirements, making their verification process unclear and inconsistent.

A plan of the testing system is developed, representing a set of methodologies for assessing the operational functionality and reliability of an autonomous delivery vehicle. They consist of several key subsystems, including a power supply, lighting system, motor, control system, communication, and sensing system. Thus, the delivery vehicles under investigation are complex systems comprising numerous modules. Each module requires thorough testing, as its functionality directly impacts the performance of the entire system. This would require complete disassembly and retesting of the vehicle.

The control system manages the operation of all components and coordinates their interactions. Drives are used to control movement and can be electric, hydraulic, or pneumatic, allowing the vehicle to maneuver in space, including on-the-spot rotations and 360-degree turns, which are particularly crucial in confined places. The control system also processes data from the perception system and makes decisions based on that information, without which its operation would be impossible.

At the core of the structural diagram is a computational device that governs the robot's actions through control algorithms, as shown in Fig. 1. This device processes video streams received from onboard cameras. Wheel controllers receive speed requirements for each wheel from the platform controller and manage the motor to maintain the specified speed under varying driving conditions. The peripheral controller regulates the operation of the lid motor, the locking mechanism, and the onboard lighting system.

The platform controller ensures power delivery to the platform, regulates current in each power branch, and switches to a backup power source when necessary. The power supply provides electricity to all the vehicle's systems. It can include various energy sources such as batteries, generators, or other power sources, which gives the potential for scaling the system to other types of autonomous transport or applications. Also EMC testing is a critical stage, ensuring that the system can operate with influence from external radiation sources.

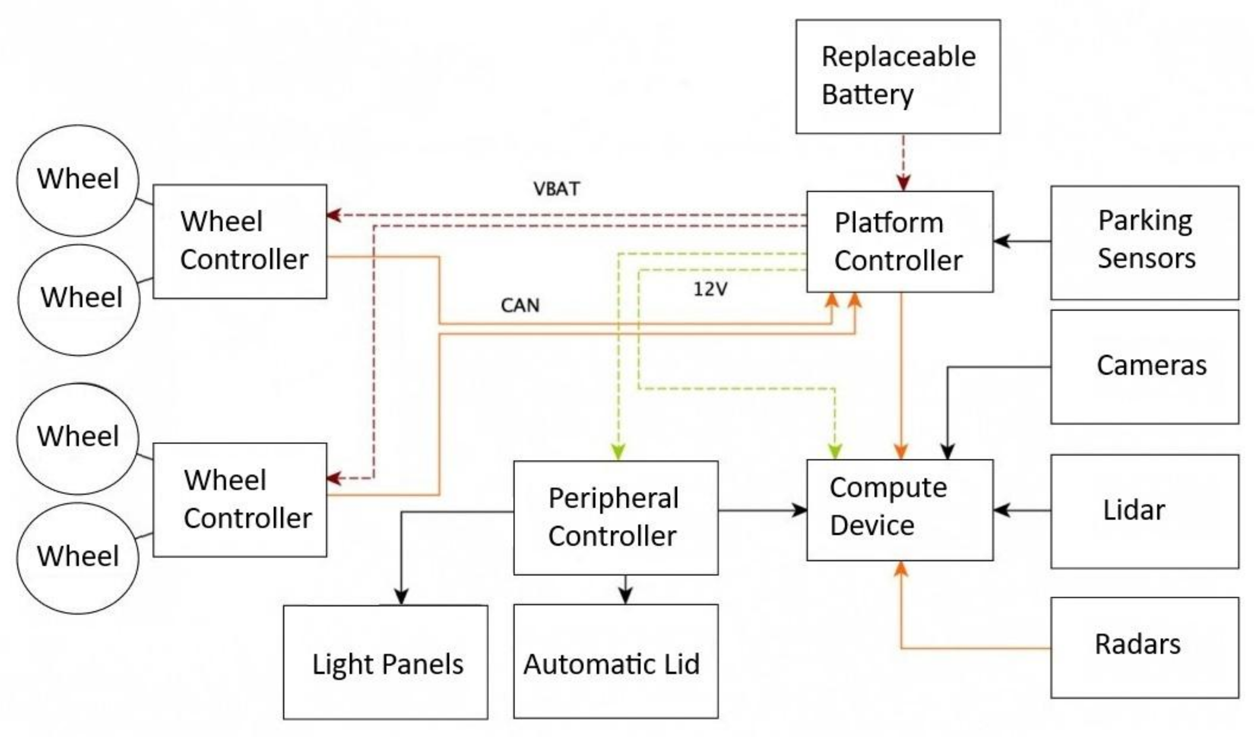


Fig. 1. The structural diagram of the vehicle under test.

Automation of the testing process can eliminate human error, enhance transparency, ensure consistent and controlled testing conditions, and allow for efficient data collection and analysis. To address these challenges, new testing methods applicable to unmanned delivery vehicles were developed. These methods ensure reliable performance in real-world conditions while meeting the demands for evaluation.

The paper is organized as follows: the related works in Section II reviews existing literature on testing systems and highlights the challenges. The methodologies in Section III details the proposed testing framework. The results in Section IV presents findings from implementing the system, focusing on performance metrics. The discussion in Section V analyzes the implications of the results, addressing limitations, and future opportunities. Finally, the conclusion summarizes the study's contributions which is given in Section VI.

II. RELATED WORK

The research in the field of Internet of Things (IoT) and autonomous ground vehicles (AGVs) is rapidly evolving due to their significance in automation and smart transportation. IoT has been widely explored as a foundational technology for the communication and coordination of autonomous systems. Studies like Biswas and Wang emphasize the integration of IoT for data exchange between sensors, vehicles, and control systems, ensuring real-time decision-making and monitoring [1]. Similarly, Baliyan et al. have discussed the role of IoT in enhancing the efficiency and scalability of autonomous delivery vehicles [2]. The approach presented by Abdul Razak et al. [3] demonstrates how IoT-based monitoring can improve vehicle safety, which could be extended to autonomous ground vehicles for monitoring operator impairment in semi-autonomous modes due to alcohol consumption.

Testing methodologies for AGVs have been a significant focus in the literature. Son et al. presented a simulation-based testing framework to validate motion control systems in uncertain environments [4]. Their work highlighted the importance of virtual testing environments to mitigate risks during physical tests. Additionally, Brogle et al. proposed a hardware-in-the-loop (HIL) testing system for autonomous vehicles to evaluate hardware and software interactions under various operational conditions [5].

The implementation of JTAG (Joint Test Action Group) testing for PCB diagnostics has been widely adopted for automated electronic testing, as PCBs are integral to the operation of IoT-enabled systems. Techniques like boundary-scan testing (JTAG) have been explored in works such as Ling et al., which outlined the advantages of automating PCB testing processes [6]. Similarly, Yang et al. emphasized the need for adaptive testing systems to handle the growing complexity of electronic modules in autonomous systems [7].

Functional testing has been explored extensively in autonomous systems to validate their real-world applicability. Autonomous systems rely on accurate and robust sensors, including cameras, LiDARs, and wireless modules. Works by Jernigan et al. have focused on developing rigorous testing methodologies to ensure sensor reliability under varying environmental conditions [8]. Ma et al. further explored the resilience of wireless interfaces in harsh environments, which is critical for maintaining communication in delivery automation [9].

Resilience testing of AGVs and their components under extreme environmental conditions has been a critical focus area. Djoudi et al. proposed a comprehensive testing framework for unmanned delivery vehicles, emphasizing vibration and climate resistance testing [10]. Studies such as Zhang et al. have demonstrated methodologies to assess the durability of electronic modules in harsh climates, including extreme temperatures and vibrations [11]. Using microscopic traffic simulation in VISSIM and the Surrogate Safety Assessment Model (SSAM), Abuzwidah et al. [12] assessed CAV performance across 21 scenarios, highlighting substantial improvements in speed and reductions in accidents under different weather conditions. Their findings emphasize the necessity for CAVs to adapt dynamically to adverse weather for optimal safety. These analyses are crucial in identifying the limitations of functional testing, guiding the development of more effective testing protocols.

Automation in testing processes has become a cornerstone of quality assurance in AGVs. Research by Ostendorff et al. showcased improvements in boundary-scan testing techniques, enabling efficient fault detection in increasingly complex PCBs [13]. Research by Garikapati et al. demonstrated the application of automated AI testing frameworks for vehicle modules, significantly reducing manual effort and errors [14]. Similarly, Jeong et al. highlighted advancements in automated diagnostics for electronic modules, offering faster and more accurate fault detection [15].

Recent works, such as Sánchez-Martinez et al., have moved toward developing integrated testing systems combining hardware, software, and environmental validation [16]. Rahman and Thill reviewed the integration of autonomous vehicles within urban networks, focusing on the performance and the challenges of ensuring consistency [17]. Comparative studies, such as Kim and Kang, have evaluated testing methodologies for EVs, providing a framework to assess their applicability for specific use cases [18].

Advanced sensor systems such as LiDAR and cameras are pivotal in AGVs. Tang et al. examined the testing frameworks required for these sensors, focusing on accuracy, calibration, and environmental adaptability [19]. Studies like Giannaros et al. investigated the performance of wireless modules in urban and rural settings, ensuring seamless data exchange with control systems [20]. These studies emphasize the importance of testing frameworks in mitigating sensor-related failures in autonomous operations, and aligning testing protocols with specific application scenarios.

III. METHODOLOGY

The system is a comprehensive methodology for assessing the functional stability of autonomous delivery systems. Fig. 2 illustrates the testing system, which includes a series of specialized procedures aimed at identifying potential vulnerabilities and failures in the system's operation.

The testing process begins with JTAG testing of printed circuit boards, enabling the detection of possible defects in electrical circuits. Functional testing focuses on verifying the proper operation of critical components such as cameras, lidars, and wired and wireless connections.

This is followed by a series of tests for vibration and environmental resistance, designed to evaluate the system's ability to function under varying environmental conditions. Electromagnetic compatibility testing is a critical stage, ensuring that the system can operate without interference from external radiation sources. The delivery system testing framework represents a basic approach to evaluating the quality and reliability of autonomous delivery systems, ensuring their stable performance under diverse operational conditions.

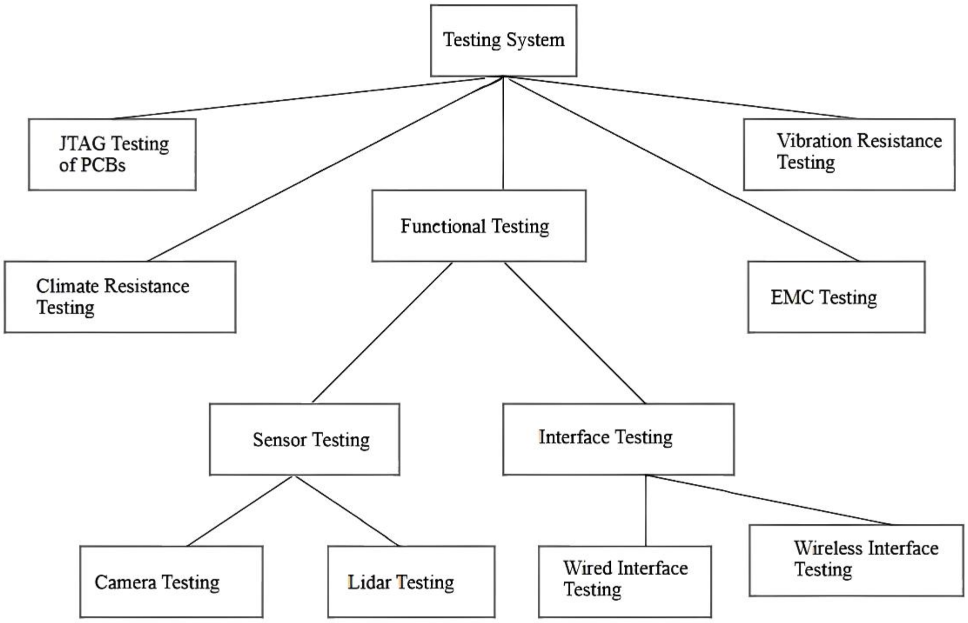


Fig. 2. The plan of the testing system.

A. *PCB Testing*

The testing of PCBs aims to select the most efficient method for further implementation, emphasizing the necessity of automating the testing process. Connection testing can detect missing pull-up resistors and signal "sticking" issues. This is achieved by setting specific values on the pins and comparing the read values against a predefined truth table.

Visual inspection and manual testing involve checking the quality of assembly, the presence and integrity of all components, and performing measurements using a multimeter. However, the increasing complexity of PCBs and the risk of human error reduce the efficiency of this method. An in-circuit tester (matrix testing) uses fixed sensor probes to check the integrity of soldered connections (Fig. 3 and Fig. 4). The disadvantages of this method include the high cost of the testing equipment, its large size, and the need to create a customized matrix contact field according to the PCB design.

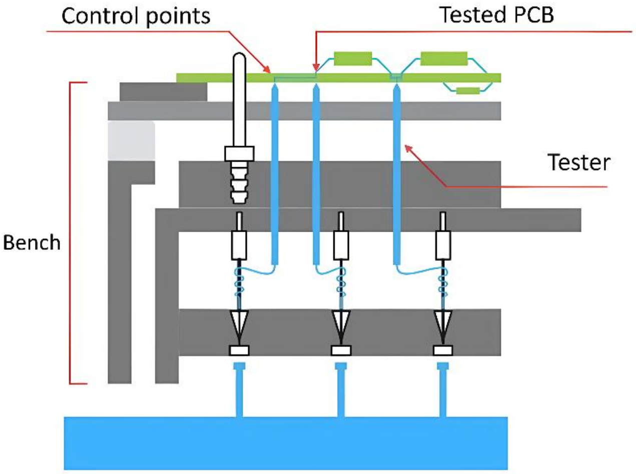


Fig. 3. Example of an in-circuit tester.

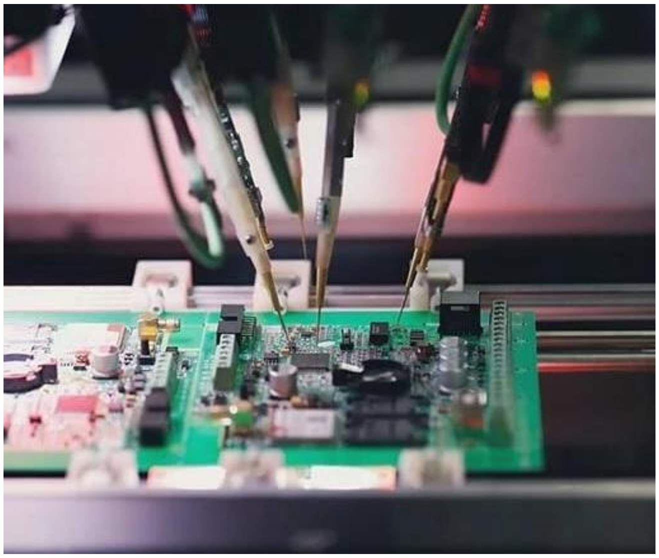


Fig. 4. Example of a flying grid tester.

JTAG testing, or connection testing, verifies whether the manufactured PCB matches the original design and identifies unintended circuit breaks or shorts [21]. For example, if the design specifies that certain chip pins must be connected somewhere on the board, the presence of the connection can be checked by applying values to one pin and reading them from others. Similarly, if the design specifies that certain pins should not be connected, JTAG testing can verify the absence of unexpected shorts by applying values to one pin and ensuring they do not influence others. The hardware of an autonomous delivery system can be tested in various ways, Table I provides more details of a comparison of PCB testing methods.

TABLE I. COMPARISON OF PCB TESTING METHODS

Method	*Error Probability*	*Performance*	*Retooling*	*Fixture Development*
Manual Testing	High	Low	Simple	Not Required
Matrix Tester	Low	High	Complex	Required
Flying grids	Low	High	Complex	Not Required
JTAG Testing	Low	High	Simple	Not Required

Thus, in the case of development and production involving numerous PCB designs manufactured in small batches, suitable testing options include purchasing a flying grids or utilizing JTAG testing. Both methods offer low error probability, high performance, and do not require the development of specialized fixtures. However, considering the cost and time required for reconfiguring the flying grids for each board, this work prioritizes JTAG testing.

B. *JTAG Testing*

To test a PCB using boundary scan, a BSDL (Boundary Scan Description Language) file must be downloaded from the chip manufacturer's website for each JTAG-supported chip. The supplementary text file describes the functions of the chip's pins.

The main advantage of boundary scan technology is the ability to set and read values at the pins without direct physical access. All signals between the device's core logic and its pins are intercepted by a serial scan path known as the Boundary Scan Register (BSR), which consists of a series of boundary scan cells. These cells are invisible during normal operation but can be used in test mode to set and/or read values from the device's pins or, in some cases, from the internal core logic. There are ten standard types of boundary scan cells, although manufacturers can define custom cell types to suit their hardware's functionality. The JTAG interface uses the following signal lines:

- TCK (Test Clock): To synchronize the internal operations of the state machine.
- TMS (Test Mode Select): Determines the next state of the state machine based on the rising edge of TCK.
- TDI (Test Data In): Represents data sent to the device's testing or programming logic. It is sampled on the rising edge of TCK when the state machine is in the correct state.

- TDO (Test Data Out): Represents data output from the device's testing or programming logic. It is valid on the falling edge of TCK when the state machine is in the correct state.
- TRST (Test Reset): An optional line that, if available, resets the TAP controller state machine.

JTAG is a synchronous interface, where signals are sampled on the rising edge of the clock (TCK) with the least significant bits first, and data output occurs on the falling edge. Boundary scan testing accelerates the preparation of tests for each project and eliminates the need for expensive test equipment. Furthermore, JTAG boundary scan can identify the precise location of faults, significantly simplifying diagnostics and repair.

With the increasing use of BGA (Ball Grid Array) packages, traditional PCB testing systems face limitations due to the inaccessibility of "internal" contacts. Boundary scan reduces test development costs by simplifying the management of chip pins for interaction with other board components. The standardized JTAG interface also allows individual tests to be created as library elements and reused across different projects, regardless of the JTAG-supported chips used. It is frequently used for programming chips on the board during production. When combined with boundary scan testing, this approach can save significant time and streamline the manufacturing process.

C. *Functional Testing*

To automate the testing process, it was necessary to choose a programming language and framework that would enable the development of a testing system with maximum simplicity and minimal programming expertise required for writing tests. Python was selected as the programming language due to its simplicity, flexibility, and widespread adoption.

In modern automated testing with Python, various testing frameworks are used. The most popular ones include:

- PyTest and PageObject;
- OpenHTF;
- Robot Framework.

The PyTest framework and the PageObject pattern allow separation of test logic from implementation, simplifying test management. However, they require significant effort during the initial development stage due to the need for low-level descriptions of testing processes and conditions for passing tests. While PyTest is excellent for integration and system testing, it may be excessive and less convenient for specific hardware board testing compared to Robot Framework. The OpenHTF library, developed by Google, was also considered. It includes a built-in graphical interface but has significant drawbacks, such as a lack of documentation, necessitating source code study, and the absence of academic work utilizing the framework.

Robot Framework offers several key advantages over other testing frameworks [22]. Its syntax is highly human-readable, making it easier to write and maintain tests, which is particularly beneficial in robotic development involving multi-disciplinary teams of hardware and software engineers. It includes built-in mechanisms for parallel test execution, detailed reporting, and logging, facilitating test analysis and statistical data collection. The framework also provides flexible mechanisms for Python integration, enabling the use of Python libraries during testing. Additionally, it allows the addition of new libraries and plugins for managing specific hardware and protocols. Thus, it is the most suitable choice for this task, offering a balance between test readability and maintainability.

To implement the delivery module testing system, readily available components were chosen, as they allow for quick hypothesis testing and reduce the development cost of the testing system. The Raspberry Pi Model 3 B+ was selected as the testing board due to its features, including an expanded 40-pin GPIO (General Purpose Input/Output) connector ideal for device testing (see Fig. 5). Additionally, the Raspberry Pi includes a CSI camera port, USB 2.0 ports, and a Micro SD slot for operating system booting and data storage. All tests conducted with the system's hardware are performed by the test bench operator.

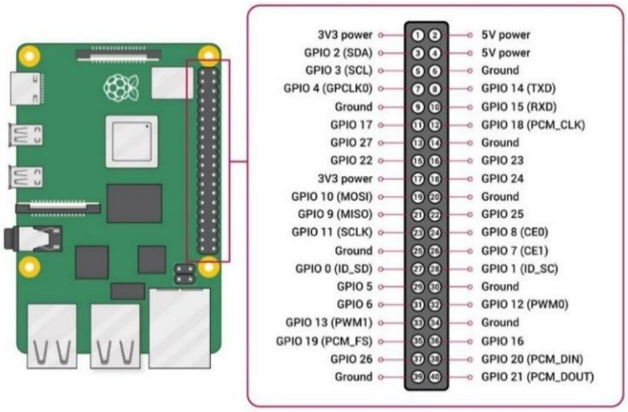


Fig. 5. External GPIO pins on rPi.

D. *Testing Sensors*

The Raspberry Pi 3 Model B+ Camera Module with a 5 MP resolution was chosen as the test camera because it connects via the CSI (Camera Serial Interface), similar to the cameras used in delivery vehicles, as shown in Fig. 6. This camera is classified as a MIPI (Mobile Industry Processor Interface) camera and connects directly to the VideoCore video chip through a CSI-2 (Camera Serial Interface-2) port, which helps conserve the Raspberry Pi's system resources, leaving USB ports available for other peripherals.

To test the camera’s functionality, it must be confirmed that the camera can capture an image directly from the Raspberry Pi board. This involves configuring the camera in the system and installing Python libraries such as picamera for camera access and pillow for image processing. A test case was then created with the following logic: an image is captured using the camera, saved to the operating system as a file, then loaded and validated (ensuring the file is a valid image). Based on this validation, a report is generated indicating whether the test passed (Pass) or failed (Fail).

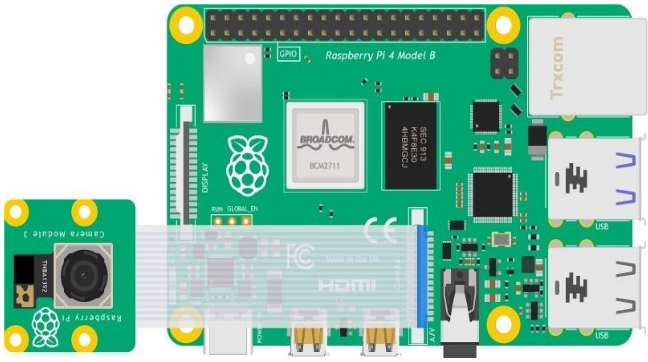


Fig. 6. The raspberry Pi 3 Model B+ camera module.

LiDARs are used for scanning and mapping the environment, aiding autonomous delivery vehicles in navigating safely. However, the diversity of LiDAR types and their operation in varying conditions complicates the development of universal testing procedures. Delivery vehicles typically use mechanical LiDARs. These feature a laser emitter (usually at a 905 nm wavelength) and a photodetector mounted on a rotating platform. This high pulse density enables the LiDAR to generate a visual 3D map of the surrounding area using a cloud of reflected points. The denser the laser pulses, the more detailed the point cloud.

Accelerometers in delivery vehicles measure acceleration along the X, Y, and Z axes. The accelerometer connects to the control electronics through the following pins:

- Power (V): Connected to the microcontroller's operating voltage.
- Ground (G): Connected to the microcontroller's ground.
- Data Signal (D): The data pin for the I²C bus, connected to the microcontroller's SDA pin.
- Clock Signal (C): The clock pin for the I²C bus, connected to the microcontroller's SCL pin.

To read values accurately from the accelerometer, the data for each axis is stored in high and low bytes. These must be combined into a single 16-bit value by performing a bit shift operation and adding the low byte to the shifted high byte.

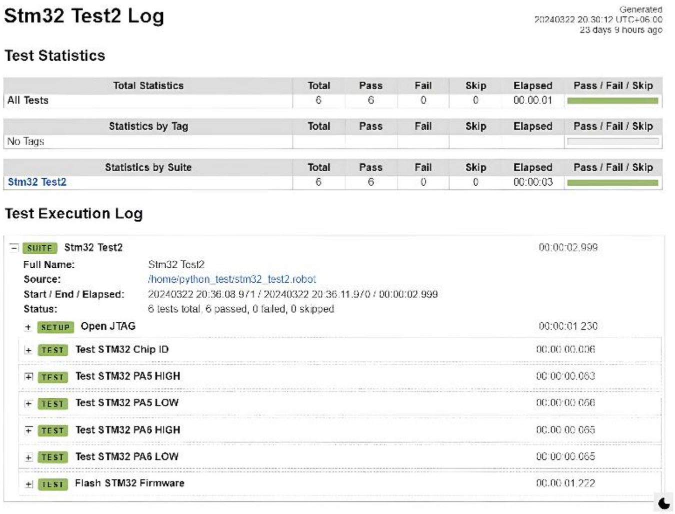
In this section, functional testing was conducted to verify the operation of key sensors in the autonomous delivery vehicle. For instance, for testing the Wi-Fi wireless interface, a test was implemented to connect to a network using the Linux Network Manager, the most popular system for managing network connections on Linux systems. The next step involves testing the developed automated test cases.

IV. RESULTS

A. *Testing Boards*

Before starting the test, ensure that the STM32F401RE board is connected to the Raspberry Pi 3. Begin by reading and verifying the board's ID. Next, test the two microcontroller pins, PA5 and PA6. Set PA5 on the STM32 to 1 (HIGH) and read the state of the corresponding pin on the Raspberry Pi. Verify that PA5 is set to 1, then similarly set and verify 0 (LOW). Repeat the same process for PA6. Finally, upload the firmware to the board and confirm that it has been successfully written, as shown in Fig. 7.

The UrJTAG utility is used for JTAG testing, allowing direct interaction with boards through the GPIO pins of the Raspberry Pi. Custom keywords were defined to structure the tests and avoid code duplication, such as opening and closing the JTAG interface and retrieving the state of a GPIO pin. The interaction with API libraries is considered but it also allows for precise fault localization, simplifying diagnostics and repair; however, its reliance on boundary scan capabilities limits its applicability to boards designed with JTAG support, potentially excluding legacy systems or simpler PCBs without such interfaces.



Stm32 Test2 Log

Generated
20240322 20:30:12 UTC+06:00
23 days 9 hours ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00.00.01	🟩

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Stm32 Test2	6	6	0	0	00:00:03	🟩

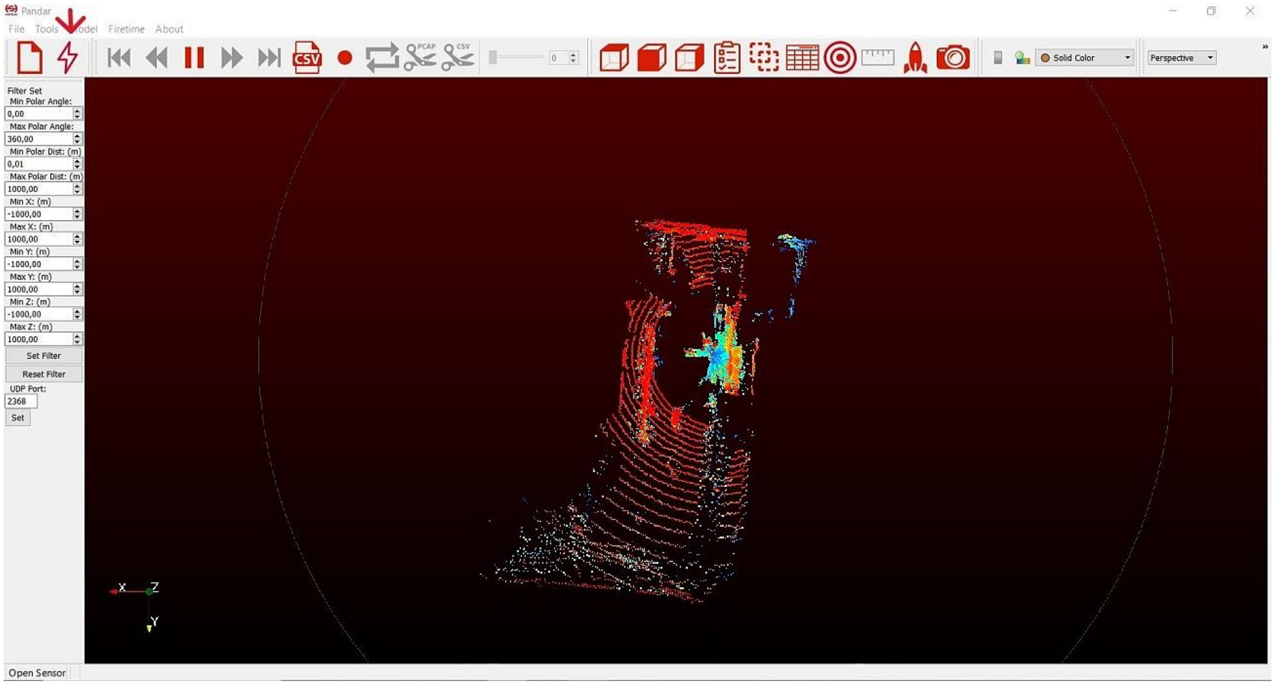
Test Execution Log

```
− SUITE Stm32 Test2                                              00:00:02.999
    Full Name:            Stm32 Test2
    Source:               /home/python_test/stm32_test2.robot
    Start / End / Elapsed: 20240322 20:36:08.971 / 20240322 20:36:11.970 / 00:00:02.999
    Status:               6 tests total, 6 passed, 0 failed, 0 skipped
  + SETUP Open JTAG                                              00:00:01.230
  + TEST Test STM32 Chip ID                                      00:00:00.006
  + TEST Test STM32 PA5 HIGH                                     00:00:00.063
  + TEST Test STM32 PA5 LOW                                      00:00:00.066
  + TEST Test STM32 PA6 HIGH                                     00:00:00.065
  + TEST Test STM32 PA6 LOW                                      00:00:00.065
  + TEST Flash STM32 Firmware                                    00:00:01.222
```

Fig. 7. JTAG testing log file.

Robot Framework tests are written in files with the .robot extension. These files use a BDD-like syntax, and the test file is named stm32_test2.robot. Test cases and keywords from the stm32_test.robot file interact with libraries by using methods belonging to those libraries.

- The jtag.py file is a library designed for working with the JTAG interface.
- The init method initializes the process for JTAG operations and sets non-blocking reading mode.
- The _set_nonblock method configures the stdout read descriptor to non-blocking mode for asynchronous interaction.
- The send method sends commands to the JTAG process, ensuring data transmission and calling flush for immediate delivery.
- The recv method reads data from the JTAG process, handles potential errors, and returns a tuple containing the execution status and response text.
- The bsdl, set_extest, and set_signal methods are used for configuring the BSDL, switching to EXTEST mode, and setting a signal on a pin, respectively.



Filter Set
Min Polar Angle: 0,00
Max Polar Angle: 360,00
Min Polar Dist: (m) 0,01
Max Polar Dist: (m) 1000,00
Min X: (m) -1000,00
Max X: (m) 1000,00
Min Y: (m) -1000,00
Max Y: (m) 1000,00
Min Z: (m) -1000,00
Max Z: (m) 1000,00
UDP Port: 2368
Solid Color
Perspective

Fig. 8. LiDAR point cloud.

The detect_id method checks whether a device with a specified identifier (ID) is connected. It sends a detect command to the JTAG process and analyzes the response to determine if the specified ID is listed among the detected devices. The quit method ensures the proper termination of the JTAG process. It sends the quit command to the process and waits for it to exit. Python-based frameworks like Robot Framework has proven effective for evaluating key components such as cameras, LiDARs, and accelerometers; however, while it supports modular testing, it requires careful configuration and additional effort for integrating new device-specific libraries.

B. Testing Functional Parts

The LiDAR under test should be inspected for external mechanical damage and the condition of its lens. Prepare the necessary equipment:

- LiDAR;
- Laptop or PC with Ethernet, equipped with software for viewing the point cloud (e.g., PandarView);
- HESAI Interface Box;
- HESAI power supply;
- Ethernet patch cord.

Open PandarView and click "Receive Data from Ethernet." A point cloud should appears as in Fig. 8. This testing methodology was developed to verify the LiDAR's functionality and ensure the proper operation of its mechanical components, including the laser receiver and emitter.

We ensured that the LiDAR is not vibrating. Strong vibrations and low-frequency mechanical oscillation sounds can indicate issues with the bearings. Testing under poor visibility conditions should include assessments in fog, rain, and snowfall. A fog machine can be used to simulate these weather conditions. The greater the number of laser pulses, the denser the LiDAR point cloud. Based on various point clouds, the autonomous delivery vehicle's computational system constructs objects that form a three-dimensional representation of the surrounding environment.

Using a vibration test bench is a more effective method for assessing vibration resistance compared to driving over various road surfaces, as it allows testing under controlled, consistent conditions. Therefore, the parameters of the vibration test bench must be calculated to ensure optimal testing. Testing for mechanical factors, particularly vibration, requires preliminary calculations to select the most suitable equipment in terms of technical specifications and cost-effectiveness.

The vibration test bench is the core component and actuator of the vibration system, reproducing a specific type of vibration and transmitting it to the test object. Tests are typically conducted on a vibration stand equipped with one or more shakers, which register the sample's response to a predefined vibrational load. The market for vibration test benches is extensive and includes a wide variety of models. As a result, decisions often lean towards purchasing the most powerful vibration test bench available. For this purpose, the Tira vibration test bench, as shown in Fig. 9, with a thrust force of up to 32k, is selected as it meets the necessary requirements.

Fig. 9. Vibration test bench.

Thus, the developed system has successfully passed all tests and is suitable for testing real devices. However, for comprehensive testing of an autonomous delivery vehicle, it is insufficient to assess only its electronic modules using JTAG testing and functional tests. It is essential to conduct vibration resistance testing to evaluate the durability of electronic, electrical, and mechanical modules under vibrational stress. This is crucial as delivery vehicles may encounter potholes, gravel, and cobblestones, which impose vibrational and impact loads that could lead to the failure of individual modules or the entire vehicle [23]. The results of the developed automated tests are presented in Table II.

TABLE II. TEST CASES OF THE DEVELOPED SYSTEM

Test Name	*Steps*	*Expected Result*	*Test Passed*
All pins connected	Connect the test board to the test bench. Run. Wait for the test to complete.	Pass	Yes
Camera test (connected)	Connect the camera to Raspberry Pi. Run the test.	Pass	Yes
One pin disconnected	Disconnect PA5, run test, check result.	Fail: PA5 HIGH $0 \neq 1$	Yes
Camera test (disconnected)	Disconnect camera, run test, check result.	Fail: Photo not captured	Yes
Accelerometer test	Connect accelerometer, run test, check result.	Pass	Yes
WiFi test (disconnected)	Disable network device, run test, check result.	Fail	Yes
Vibration test	Place device on Tira, run Tira, visual check.	No cracks	Yes
EMC test	Execute EMC protocol, check connetion	Device stays connected	Yes

Environmental conditions are typically described using statistical variables such as temperature, humidity, air quality, and so on. These factors are critical for the functionality and lifespan of delivery vehicles, and manufacturers must ensure that their modules can operate within specified conditions while maintaining their stated performance characteristics.

Most climate tests are conducted in a climate chamber, as shown in Table III, which can simulate changes in temperature, humidity, dew, and frost. Additional conditions, such as dust and solar radiation exposure, are tested using dedicated chambers. Commonly measured parameters for determining the electrical safety of delivery vehicle modules include current, voltage, leakage path length between conductors, and the energy of emitted waves. It is crucial to note that electrostatic discharge (ESD) is unacceptable for electronic modules when they are not yet enclosed in a protective casing.

TABLE III. THE RESULTS FOR CLIMATE CHAMBER TESTS

Test Name	*Testing Methodology*	*Result*
Temperature Testing	Climate chamber, temperature range: -40°C to +85°C; condensation cycles and frost formation.	Most components operate within -20°C to +60°C; failures occur at ±70°C; frost impacts 30% of components
Humidity Testing	Controlled humidity chamber, 10%–95% RH	Corrosion risk increases by 40% at 85% RH; short circuits above 90% RH; risk varies by material
Dust Exposure Testing	Dust chambers with varying particle size (ISO 12103-1), Arizona test dust	Dust penetration in enclosures above 5-micron particles; 20% degradation in 3 weeks
Solar Radiation Testing	UV and infrared radiation exposure tests	UV exposure causes 15% material degradation over 1000 hours; discoloration starts at 500 hours
Electrical Safety Testing	High-voltage insulation resistance testing	Leakage current below 1mA at 1000V; insulation resistance >10MΩ meets IEC standards
Electrostatic Discharge (ESD) Testing	ESD simulator testing at 2kV–15kV discharge levels	Devices withstand up to 10kV discharge; failures start above 12kV

Once the delivery vehicle is fully assembled, it must withstand electrostatic discharge, as all electronic components are securely shielded by the casing. However, the reliance on specialized equipment introduces significant costs and operational constraints, especially for small-scale manufacturers; additionally, vibration testing under controlled conditions may not fully replicate the complexities of real-world terrain.

V. DISCUSSION

The developed testing system demonstrates significant advancements in ensuring the reliability and functional stability of autonomous delivery systems. By combining JTAG testing, functional testing, and environmental assessments, the system provides a comprehensive framework for identifying vulnerabilities and verifying the performance of key modules under diverse conditions. However, while the system achieves its primary objectives, there are opportunities for enhancement and certain limitations to address.

One notable strength of the system lies in its use of JTAG testing, which offers a highly efficient and cost-effective method for verifying PCB integrity. Unlike manual or in-circuit testing methods, JTAG testing eliminates the need for expensive test fixtures and reduces human error. It also allows for precise fault localization, simplifying diagnostics and repair. Although, its reliance on boundary scan capabilities restricts its applicability to boards designed with JTAG support, potentially excluding legacy systems or simpler PCBs without such interfaces [24].

Functional testing, implemented through Python-based frameworks like Robot Framework, has proven effective for evaluating key components such as cameras, LiDARs, and accelerometers. The framework's human-readable syntax and Python integration streamline test development and maintenance, making it accessible to multidisciplinary teams. While Robot Framework supports modular testing, it requires careful configuration and additional effort for integrating new device-specific libraries, which may pose challenges for teams with limited resources or expertise.

Environmental testing introduces another layer of robustness by simulating real-world conditions in controlled environments [25]. The use of climate chambers for temperature, humidity, and frost testing, alongside vibration benches for mechanical stress evaluation, provides valuable insights into the durability of delivery systems. Nevertheless, the reliance on specialized equipment, such as the Tira vibration test bench, introduces significant costs and operational constraints, especially for small-scale manufacturers. Additionally, while vibration testing under controlled conditions is highly effective, it may not fully replicate the complexities of real-world terrain [26].

The inclusion of sensor-specific tests, such as those for LiDARs, camera, and accelerometers, highlights our focus on the core functionality of autonomous vehicles. Nevertheless, the diversity of sensor technologies and operational environments poses a challenge to the development of universal testing procedures [27]. For example, mechanical LiDARs with rotating components require different calibration and durability tests compared to solid-state LiDARs. Similarly, environmental factors like fog or heavy rain can disproportionately affect sensor performance, requiring further refinements to testing methodologies. The system also incorporates EMC testing to ensure devices can operate without interference [28]. This is particularly important for delivery vehicles, which rely on seamless communication between components. While the system's EMC tests have been effective, integrating real-time data collection and analysis during such tests could provide additional insights and improve overall reliability.

Despite these strengths, the system has limitations in scalability and regulatory compliance. The absence of standardized testing protocols for autonomous delivery vehicles means that manufacturers may face challenges in aligning their testing processes with emerging regulatory requirements. Furthermore, the system's reliance on high-performance test equipment, such as climate chambers and vibration benches, may not be feasible for all manufacturers, especially those operating in cost-sensitive markets.

Opportunities for improvement include the integration of machine learning algorithms to optimize test procedures and predict potential failures based on historical data [29]. Additionally, developing lightweight and portable testing solutions could reduce costs and improve accessibility for smaller manufacturers. The system could focus on expanding its adaptability and automation capabilities. Automated test execution and data analysis could significantly reduce the time and effort required for repetitive testing tasks, particularly for high-volume production scenarios. For example, integrating automated tools for capturing and analyzing LiDAR point clouds or accelerometer data could provide deeper insights into module performance under specific conditions.

Moreover, the testing framework could be extended to include real-time monitoring and diagnostics during operational testing. This critical opportunity lies in the development of modular testing architectures. Modular designs would also support scalability, allowing the testing framework to be adapted to different vehicle types or system configurations without significant reconfiguration. This would allow for dynamic adjustments in testing parameters, ensuring that modules are evaluated under a broader range of conditions, including unexpected environmental factors or system interactions [30]. Such an approach could also help detect intermittent faults that might not appear under standard test scenarios.

The incorporation of cloud-based testing and analytics could further enhance the system's capabilities [31]. A centralized platform for storing, analyzing, and sharing test results would enable manufacturers to benchmark performance across multiple production cycles or facilities. Additionally, cloud integration could facilitate collaborative development of standardized testing methodologies, allowing manufacturers to align their processes with industry best practices and emerging regulatory standards.

While the current framework emphasizes hardware testing, extending the scope to include software validation would provide a more holistic approach to system reliability. Autonomous delivery vehicles rely heavily on complex algorithms for navigation, object detection, and decision-making [32]. Testing these algorithms in simulated environments that mimic real-world scenarios could complement the hardware testing process, ensuring seamless integration and overall system robustness.

Finally, addressing regulatory alignment remains a critical area for improvement. By engaging with industry stakeholders and regulatory bodies, the system could be tailored to meet specific compliance requirements, paving the way for broader adoption in global markets [33]. Collaboration with regulatory bodies to define standardized testing frameworks could enhance the system's applicability and acceptance in the industry. Such efforts would also help establish the system as a benchmark for testing autonomous delivery vehicles, contributing to the standardization of quality assurance practices in this rapidly evolving field [34].

Overall, the developed system effectively addresses many challenges associated with testing autonomous delivery systems, further innovations in automation, modularity, and regulatory alignment will unlock new possibilities. By embracing these opportunities, the framework has the potential to become a solution for ensuring the safety, reliability, and functionality of autonomous delivery vehicles in diverse operational environments.

VI. CONCLUSION

The developed testing system provides a robust framework for assessing the functionality and reliability of autonomous delivery systems, combining JTAG testing, functional evaluations, and environmental assessments. It ensures comprehensive testing of critical components, including PCBs, sensors, and wireless interfaces, while maintaining cost-effectiveness through automation and streamlined processes.

However, the system has limitations. The reliance on specialized equipment like vibration benches and climate chambers can be cost-prohibitive for smaller manufacturers. Additionally, the absence of standardized protocols for autonomous delivery vehicles limits its regulatory alignment, and the diversity of sensor technologies complicates the development of universal testing methods.

Future work should focus on enhancing scalability and adaptability through modular and portable testing setups, integrating machine learning for predictive diagnostics, and expanding the framework to include software validation alongside hardware testing. Collaboration with regulatory bodies to establish standardized testing protocols and incorporating real-time data analytics will further strengthen the system's applicability and industry relevance.

REFERENCES

[1] A. Biswas and H.-C. Wang, "Autonomous Vehicles Enabled by the Integration of IoT, Edge Intelligence, 5G, and Blockchain," Sensors, vol. 23, no. 4, p. 1963, 2023. doi: 10.3390/s23041963.

[2] A. Baliyan, J. S. Dhatterwal, K. S. Kaswan, and V. Jain, "Role of AI and IoT Techniques in Autonomous Transport Vehicles," in AI Enabled IoT for Electrification and Connected Transportation, Transactions on Computer Systems and Networks. Singapore: Springer, 2022, pp. 1–16. doi: 10.1007/978-981-19-2184-1_1.

[3] S. F. A. Razak, S. Yogarayan, and A. Ullah, "Preventing Impaired Driving Using IoT on Steering Wheels Approach," HighTech and Innovation Journal, vol. 5, no. 2, 2024. doi: 10.28991/HIJ-2024-05-02-012.

[4] T. Duy Son, A. Bhave and H. Van der Auweraer, "Simulation-Based Testing Framework for Autonomous Driving Development," 2019 IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, 2019, pp. 576-583, doi: 10.1109/ICMECH.2019.8722847.

[5] C. Brogle, C. Zhang, K. L. Lim and T. Bräunl, "Hardware-in-the-Loop Autonomous Driving Simulation Without Real-Time Constraints," in IEEE Transactions on Intelligent Vehicles, vol. 4, no. 3, pp. 375-384, Sept. 2019, doi: 10.1109/TIV.2019.2919457.

[6] Q. Ling and N. A. M. Isa, "Printed Circuit Board Defect Detection Methods Based on Image Processing, Machine Learning and Deep Learning: A Survey," in IEEE Access, vol. 11, pp. 15921-15944, 2023, doi: 10.1109/ACCESS.2023.3245093.

[7] J. Yang, R. Bai, H. Ji, Y. Zhang, J. Hu, and S. Feng, "Adaptive Testing Environment Generation for Connected and Automated Vehicles with Dense Reinforcement Learning," arXiv, vol. 2402.19275, Feb. 2024. doi: 10.48550/arXiv.2402.19275.

[8] M. Jernigan, S. Alsweiss, J. Cathcart and R. Razdan, "Conceptual Sensors Testing Framework for Autonomous Vehicles," 2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan, 2018, pp. 1-4, doi: 10.1109/VNC.2018.8628370.

[9] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor and B. Vucetic, "High-Reliability and Low-Latency Wireless Communication for Internet of Things: Challenges, Fundamentals, and Enabling Technologies," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7946-7970, Oct. 2019, doi: 10.1109/JIOT.2019.2907245.

[10] A. Djoudi, L. Coquelin and R. Régnier, "A simulation-based framework for functional testing of automated driving controllers," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-6, doi: 10.1109/ITSC45102.2020.9294454.

[11] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 196, pp. 146–177, Feb. 2023. doi: 10.1016/j.isprsjprs.2022.12.021.

[12] M. Abuzwidah, A. Elawady, L. Wang, and W. Zeiada, "Assessing the Impact of Adverse Weather on Performance and Safety of Connected and Autonomous Vehicles," Civil Engineering Journal, vol. 10, no. 9, 2024. doi: 10.28991/CEJ-2024-010-09-019.

[13] S. Ostendorff, J. Sachsse, H. Wuttke, and J. Meza Escobar, "Adaptive Test System to Improve PCB Testing in the Automotive Industry," SAE Int. J. Passeng. Cars – Electron. Electr. Syst., vol. 6, no. 1, pp. 294–300, 2013. doi: 10.4271/2013-01-1230.

[14] D. Garikapati and S. S. Shetiya, "Autonomous Vehicles: Evolution of Artificial Intelligence and the Current Industry Landscape," Big Data Cogn. Comput., vol. 8, no. 4, p. 42, 2024. doi: 10.3390/bdcc8040042.

[15] Y. Jeong, S. Son, E. Jeong, and B. Lee, "An Integrated Self-Diagnosis System for an Autonomous Vehicle Based on an IoT Gateway and Deep Learning," Appl. Sci., vol. 8, no. 7, p. 1164, 2018. doi: 10.3390/app8071164.

[16] R. Sánchez-Martinez, J. E. Sierra-García, and M. Santos, "Performance and Extreme Conditions Analysis Based on Iterative Modelling Algorithm for Multi-Trailer AGVs," Mathematics, vol. 10, no. 24, p. 4783, 2022. doi: 10.3390/math10244783.

[17] M. M. Rahman and J.-C. Thill, "Impacts of connected and autonomous vehicles on urban transportation and environment: A comprehensive review," Sustainable Cities and Society, vol. 96, p. 104649, Sep. 2023. doi: 10.1016/j.scs.2023.104649.

[18] B. Kim and E. Kang, "Toward Large-Scale Test for Certifying Autonomous Driving Software in Collaborative Virtual Environment," in IEEE Access, vol. 11, pp. 72641-72654, 2023, doi: 10.1109/ACCESS.2023.3295500.

[19] S. Tang, Z. Zhang, Y. Zhang, J. Zhou, Y. Guo, S. Liu, S. Guo, Y.-F. Li, L. Ma, Y. Xue, and Y. Liu, "A Survey on Automated Driving System Testing: Landscapes and Trends," ACM Trans. Softw. Eng. Methodol., vol. 32, no. 5, Art. 124, p. 62, Sep. 2023. doi: 10.1145/3579642.

[20] A. Giannaros, A. Karras, L. Theodorakopoulos, C. Karras, P. Kranias, N. Schizas, G. Kalogeratos, and D. Tsolis, "Autonomous Vehicles: Sophisticated Attacks, Safety Issues, Challenges, Open Topics, Blockchain, and Future Directions," J. Cybersecur. Priv., vol. 3, no. 3, pp. 493–543, 2023. doi: 10.3390/jcp3030025.

[21] F. Zhang, S. D. Paul, P. Slpsk, A. R. Trivedi and S. Bhunia, "On Database-Free Authentication of Microelectronic Components," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 1, pp. 149-161, Jan. 2021, doi: 10.1109/TVLSI.2020.3039723.

[22] L. Jian-Ping, L. Juan-Juan and W. Dong-Long, "Application Analysis of Automated Testing Framework Based on Robot," 2012 Third International Conference on Networking and Distributed Computing, Hangzhou, China, 2012, pp. 194-197, doi: 10.1109/ICNDC.2012.53.

[23] J. Hu, T. Xu and R. Zhang, "Testing and Evaluation of Autonomous Vehicles Based on Safety of the Intended Functionality," 2021 6th International Conference on Transportation Information and Safety (ICTIS), Wuhan, China, 2021, pp. 1083-1086, doi: 10.1109/ICTIS54573.2021.9798586.

[24] E. Tramacere, S. Luciani, S. Feraco, A. Bonfitto, and N. Amati, "Processor-in-the-Loop Architecture Design and Experimental Validation for an Autonomous Racing Vehicle," Applied Sciences, vol. 11, no. 16, p. 7225, 2021. doi: 10.3390/app11167225.

[25] N. Assymkhan and A. Kartbayev, “Advanced IoT-Enabled Indoor Thermal Comfort Prediction Using SVM and Random Forest Models” International Journal of Advanced Computer Science and Applications (IJACSA), 15(8), 2024. doi:10.14569/IJACSA.2024.01508102.

[26] Z. Tahir and R. Alexander, "Coverage based testing for V&V and Safety Assurance of Self-driving Autonomous Vehicles: A Systematic Literature Review," 2020 IEEE International Conference On Artificial

Intelligence Testing (AITest), Oxford, UK, 2020, pp. 23-30, doi: 10.1109/AITEST49225.2020.00011.

[27] M. M. Long, T. T. Diep, S. H. Needs, M. J. Ross, and A. D. Edwards, "PiRamid: A compact Raspberry Pi imaging box to automate small-scale time-lapse digital analysis, suitable for laboratory and field use," HardwareX, vol. 12, Oct. 2022. doi: 10.1016/j.ohx.2022.e00377.

[28] F. Nussipova, S. Rysbekov, Z. Abdiakhmetova, and A. Kartbayev, "Optimizing loss functions for improved energy demand prediction in smart power grids," International Journal of Electrical and Computer Engineering (IJECE), vol. 14, no. 3, pp. 3415–3426, 2024. do: 10.11591/ijece.v14i3.pp3415-3426.

[29] D. Bruggner, A. Hegde, F. S. Acerbo, D. Gulati and T. D. Son, "Model in the Loop Testing and Validation of Embedded Autonomous Driving Algorithms," 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 2021, pp. 136-141, doi: 10.1109/IV48863.2021.957553.

[30] R. R. Arany, H. Van der Auweraer and T. D. Son, "Learning Control for Autonomous Driving on Slippery Snowy Road Conditions," 2020 IEEE Conference on Control Technology and Applications (CCTA), Montreal, Canada, 2020, pp. 312-317, doi: 10.1109/CCTA41146.2020.9206260.

[31] L. Shuguang, L. Zhonglin, W. Wenbo, Z. Yang, H. Jierui and C. Hong, "Vehicle-in-the-Loop Intelligent Connected Vehicle Simulation System Based on Vehicle-Road-Cloud Collaboration," 2022 IEEE 25th Intern. Conf. on Intelligent Transportation Systems (ITSC), Macau, China, 2022, pp. 1711-1716, doi: 10.1109/ITSC55140.2022.9922190.

[32] M. Ng, D. Jagetiya, X. Gao, H. Shi, J. Gao and J. Liu, "Real-Time Detection of Objects on Roads for Autonomous Vehicles Using Deep Learning," 2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, 2022, pp. 73-80, doi: 10.1109/BigDataService55688.2022.00019.

[33] D. Schepis, S. Purchase, D. Olaru, B. Smith, and N. Ellis, "How governments influence autonomous vehicle (AV) innovation," Transportation Research Part A: Policy and Practice, vol. 178, p. 103874, Dec. 2023. doi: 10.1016/j.tra.2023.103874.

[34] T. Sever and G. Contissa, "Automated driving regulations – where are we now?," Transportation Research Interdisciplinary Perspectives, vol. 24, p. 101033, Mar. 2024. doi: 10.1016/j.trip.2024.101033.