RESEARCH ARTICLE

ECEJOURNALS.IN

Advancing Autonomous Vehicle Technology: Embedded Systems Prototyping and Validation

Rigas Ramchurn*

Nigeria LNG Limited, Bonny Island, Nigeria

Keywords:
Medical Embedded Systems;
Embedded Network Protocols;
Signal Processing in Embedded Systems;
Hardware Acceleration;
System Integration

Corresponding Author Email:
emtukudoh@gmail.com

DOI: 10.31838/ESA/02.02.07

Received : 27.02.25
Revised : 18.03.25
Accepted : 25.05.25

Abstract

The arrival of autonomous vehicles (AVs) is changing the automotive industry at the speed of light. Technological innovation in these self driving marvels will change the way we look at transportation. The pivotal role of embedded systems that embed the AVs in their environment and let them perceive, decide in split second, and navigate the arena with minimal human intervention, are at the heart of this revolution. Rapid advancement of artificial intelligence, sensor technology, and embedded software development has been the journey towards fully autonomous vehicles. And as these technologies converge, they're building an amazingly rich ecosystem within each of those vehicles, able to process enormous amounts of data in real time, and to execute incredibly precise control commands. The drivers assistance features we are used to in our basic cars today have led us to cars that can operate completely in a very complex environment. The path to AV wide adoption isn't without hurdles, however. Embedded systems for the design and validation of autonomous vehicles have their own challenges and hurdles for engineers and developers. There are many challenges such as ensuring real time performance as well as achieving robust safety measures and integrating externally developed subsystems seamlessly. The industry has started to become more rigorous about the prototyping and validation of ideas, realizing that there's no point in having safe, reliable autonomous vehicles on the roads without doing as much.

How to cite this article: Ramchurn R (2025). Advancing Autonomous Vehicle Technology: Embedded Systems Prototyping and Validation. SCCTS Journal of Embedded Systems Design and Applictions, Vol. 2, No. 2, 2025, 56-64

Introduction

Development and testing of AV systems are necessarily multifaceted. The other issue to consider is that engineers must not only think in terms of the individual components, but the way they interplay with each other and make up the rest as a whole. It is this holistic view that is needed in order to develop AVs that can safely and efficiently operate in the highly unpredictable real world environments in which they will be operating. And as we continue to evolve in the autonomous vehicle technology world, we are going to learn what makes up the complex process, and how we are paving the way to the future of transportation [1-4]

Embedded Systems in Autonomous Vehicles: Understanding

Autonomous vehicle technology relies on embedded systems, those are pivotal link between hardware components and the intricate software that runs the AV function. Designed to carry out specialized tasks as part of what is larger part of the operation of the vehicle as a whole, each computer system in these specialized computer systems is an important component in the working as a whole of the autonomous system.[5-8]

Essentially, embedded systems in AVs add on top of sensor data and control algorithms while also coordinating two-way communication between the other subsystems. Realization of ability to process real time data, reliability as well as ability to operate in harsh environment under resource constraints characterize them. On the contrary, embedded systems deployed in AVs are specific toefficiency, low power dissipation and repercussion of environmental stimulus with minimal latency.

Embedded systems in autonomous vehicles are highly distributed due to architecture being distributed between several Electronic Control Units (ECUs) that work together to manage the multiple system features of the vehicle. Through different communication protocols, including CAN (Controller Area Network), FlexRay and Automotive Ethernet, these ECUs form a complex network facilitating high speed data exchange and coordinated decision making for the overall plant.[9-11]

Key components of AV embedded systems include:

1. **Sensor Interface Units:** These systems take in raw data from sensors such as cameras, LiDAR, radar, and ultrasonic sensors, filtering and distilling it into information useful in decision making algorithms.
2. **Central Processing Units:** These units often have high performance processors that are required to run complex algorithms for perception, path planning and decision making.
3. **Actuator Control Systems:** Responsible for translating high levels commands to specific control signals for steering, acceleration, and braking commands.
4. **Communication Modules:** They also control internal vehicle communications, and external vehicle-to-anything (V2X) connections.
5. **Safety-Critical Systems:** Embedded systems dedicated to fail safe operation and redundancy in case of component failures aimed to keep vehicle safety.

And due to the intricate interplay among these embedded systems, the platform is robust and responsive, with the ability to address the many challenges of autonomous driving. And, as we go through this article, we will see how these systems are created, tested, and validated so that these systems satisfy the very requirements that autonomous vehicle operation requires.[12-17]

The V-Model: AV Software Development: A Framework

Similar to the V-model, the development of embedded systems for autonomous vehicles is managed by the V model, which is a cornerstone upon which all things are built: the importance of verification and validation of an embedded system at every step of the development lifecycle. This is the type of 'V' model that will be used to plot out each phase of development and linked to its respective testing phase to produce a comprehensive and systematic approach to software construct and testing.

Table 1: Embedded Systems In Autonomous Vehicle Prototyping

System	Role
Sensors Integration	Sensors integration enables the autonomous vehicle to perceive its environment through LiDAR, cameras, and radar, providing real-time data.
Control Systems	Control systems process sensor data to make decisions for vehicle control, such as steering, acceleration, and braking, ensuring smooth operation.
Power Management	Power management systems ensure that embedded components have a stable power supply while optimizing energy consumption for longer driving ranges.
Communication Networks	Communication networks enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, allowing the vehicle to respond to external signals.
Safety Mechanisms	Safety mechanisms such as emergency braking, collision avoidance, and automatic alerting systems ensure the vehicle can respond appropriately to threats.
Navigation Systems	Navigation systems use GPS and real-time map data to guide the vehicle along its route, adjusting to dynamic traffic conditions and road obstacles.

In the V of the figure we find phases of development descending on the left hand side with high level requirements on the left and proceeding to the right down to the detailed design and implementation phases. The 'V' is but one representation of the integration and testing on the left in that they are the valid (or validated, as is more common) phases which are such that the right side of the 'V' represents the ascending phases of those phases. This symmetry guarantees that all stages in the development process have a testing counterpart, and thus, a well tested system at all times.[18-22]

The V-Model for AV software development typically includes the following stages:

Down the left side of the 'V' each phase makes outputs which become inputs to the next. Similarity is that as testing progresses deeper on the right side, each testing phase confirms its corresponding development phase, that the implementation conforms to required requirements and design criteria.

Proof of the V Model's strength is its early test planning and continuous validation across development. The value of this approach is most significant in the autonomous vehicle domain where safety and reliability are of utmost concern. Between integrating them early to facilitate development, developers will be able to pinpoint and address potential problems right in the very beginning of the development project, and reduce the likelihood of expensive intervention in the later stages and, instead, increase system maintainability.

In addition, the V-Model provides traceability from requirements into design elements and test cases. The traceability afforded by this is vital if AV is to be shown to comply with safety standards and regulations, which is a key component of the development process. It manages change efficiently by supporting such stuffs such as efficient change management and developers will assess impact of the modification on the whole system and update the corresponding test cases. The V-Model is a great way to provide a robust framework for the development of AV software; however, it's important to note that it can often be used in practice after iterating and adapting it to account for this extremely dynamic technological area of autonomous vehicles. Further sections in this model will show how this model is integrated with other development methodologies and testing strategies to form a complete approach to AV system development and validation (Figure 1).

Simulation: AV System Development is a Cornerstone. Embedded systems for autonomous vehicles rely heavily on simulation to develop and validate the resulting embedded systems. Simulation provides a safe, cost effective, and highly scalable testing and tuning environment of AV technologies. Simulation creates virtual images of real world scenarios that allows developers to determine the

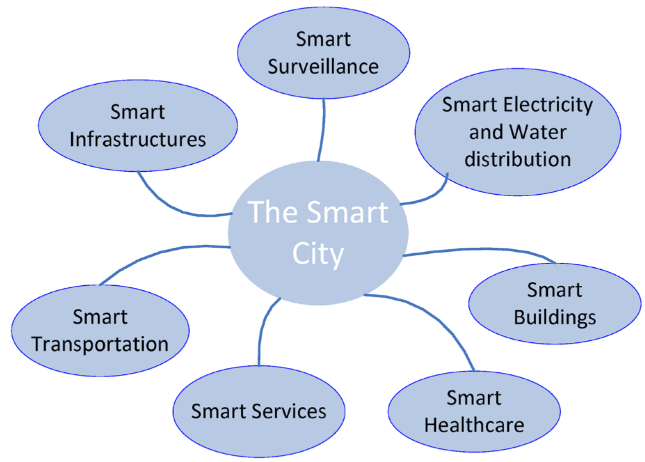

Fig 1: V-Model for AV software development typically includes the following stages

performance of the system, identify possible issues, and optimize algorithms without having to exhaustively test the product on the real world. Simulation is not only important in AV development, it is essential. They can help to create disparate and frightening situations that are hard to replicate in actual world test. Indeed, it is this capability that is important in making sure that autonomous vehicles can operate in many different situations the vehicle may encounter on the road, from everyday driving situations to edge cases that most of the vehicle's capabilities.

Key aspects of simulation in AV development include:

1. **Sensor Simulation:** Testing perception algorithms with inputs from various sensors which include cameras, LiDAR, and the radar.
2. **Environmental Simulation:** Making these virtual environments represent real life conditions, including weather, traffic, and road infrastructure.
3. **Vehicle Dynamics Simulation:** To test and evaluate vehicle performance, we model the vehicle's physical behavior.
4. **Traffic Simulation:** Realistic AVs Generate Traffic Scenarios To Test That The AVs Can Interact With Other Vehicles And Pedestrians.
5. **Hardware-in-the-Loop (HIL) Simulation:** Provide interface to simulate environments to interact with integrated hardware components to test the interaction of software and hardware systems.

The HIL testing process typically involves the following steps:

1. **Scenario Definition:** That means, creating a range of test scenarios to accommodate for different

driving conditions, traffic situation, as well as potential edge cases.

2. **Test Case Development:** What specific test cases you are able to design to test different aspects of the system's good performance and functioning.
3. **Simulation Execution:** Real time test of the test cases with the physical ECUs driving the physical vehicle by simulating physical inputs to the ECUs as they would in the real vehicle.
4. **Data Collection and Analysis:** Collecting the performance data and analyze how the system respond to any issues or areas for change to improve.
5. **Iteration and Refinement:** Basing design and analysis on insights from testing in order to refine the algorithms, tune the parameters, and otherwise optimize the system performance.

This highlights one of the main benefits in HIL testing: being able to detect integration issues and system level problems that would not show up in software only simulation or in component testing. Developers can test with actual hardware, spotting timing issues, communication delays, real world constraints that affect system performance. This testing also provides a way to verify the robustness and fault tolerance of AV systems. Simulation of unexpected situations like failure modes and edge cases help developers to evaluate how the system behaves when subjected to various situations, so that safety critical functions maintain functionality even in an adverse situation.

HIL testing also provides automotive developers with a way to validate OTA update processes; ensuring that software updates can be safely deployed to a vehicle's systems without interfering or breaking functionality and causing additional issues. It's particularly important for autonomous vehicles, which rely on frequent software updates to get better performance and new features.

The HIL testing methodologies for autonomous vehicle technology keep moving forward as the technology itself develops. Under modern HIL systems we often see high fidelity graphics and physics engines that enhance the realism of the simulation, and advanced data analytics capabilities that make it easy for developers to identify and diagnose problems. HIL testing has a lot of advantages, but one must be aware of all the pitfalls. In other words, HIL simulations do not necessarily capture the entire dynamics of real world vehicle behavior and environmental interactions. HIL testing is then usually combined with other validation methods, like software in the loop (SIL) testing, full vehicle road testing, etc. to form a comprehensive autonoma vehicle systems validation strategy.[23-27]]

Vehicle-in-the-Loop Testing: In the Creation of Real-World Conditions: The Validating AV Systems

Vehicle in the Loop (VIL) testing is the next step in the validation of autonomous vehicle systems, linking simulation based testing with on the road trials. The technique involves running a real vehicle in a simulated environment to evaluate the performance of complete AV systems within controlled but realistic environments. Validating the complex interact between software, hardware and the physical dynamics of the vehicle in scenarios that closely mimic real world driving scenarios is VIL testing (Figure 2).

The primary objective of VIL testing is to examine how the entire system performs under the physical reality of vehicle motion, sensor limits and environmental terms. However, this methodology offers significant insight into the behavior of the system which may not be fully realized in software simulation or stationary hardware in the loop tests. Additionally, VIL testing is essential to test the AV's ability to handle edge cases and unknown scenarios. Introducing virtual elements into the real world allows testers to create scenarios that stress the limits of these structures – an important first step towards identifying possible failure modes and increasing robustness as a whole.

The data obtained with VIL tests is valuable for optimizing machine learning models and decision making algorithms. This rich, multi modal data captured on real sensors interacting with physical as well as virtual environments provides a real world dataset for training and validating AI systems bridging the gap between simulated and real system performance.

A key challenge of VIL is that the testing methodologies are being pushed to advance in terms of advanced technologies such as integrating 5G communication for real time data exchange, high fidelity environmental models, and AI driven simulation generation. This has permitted more extensive and efficient testing processes to occur and to speed up the development and validation of autonomous vehicle systems. While VIL testing offers significant advantages, it's important to note that it complements rather than replaces other testing methodologies. A

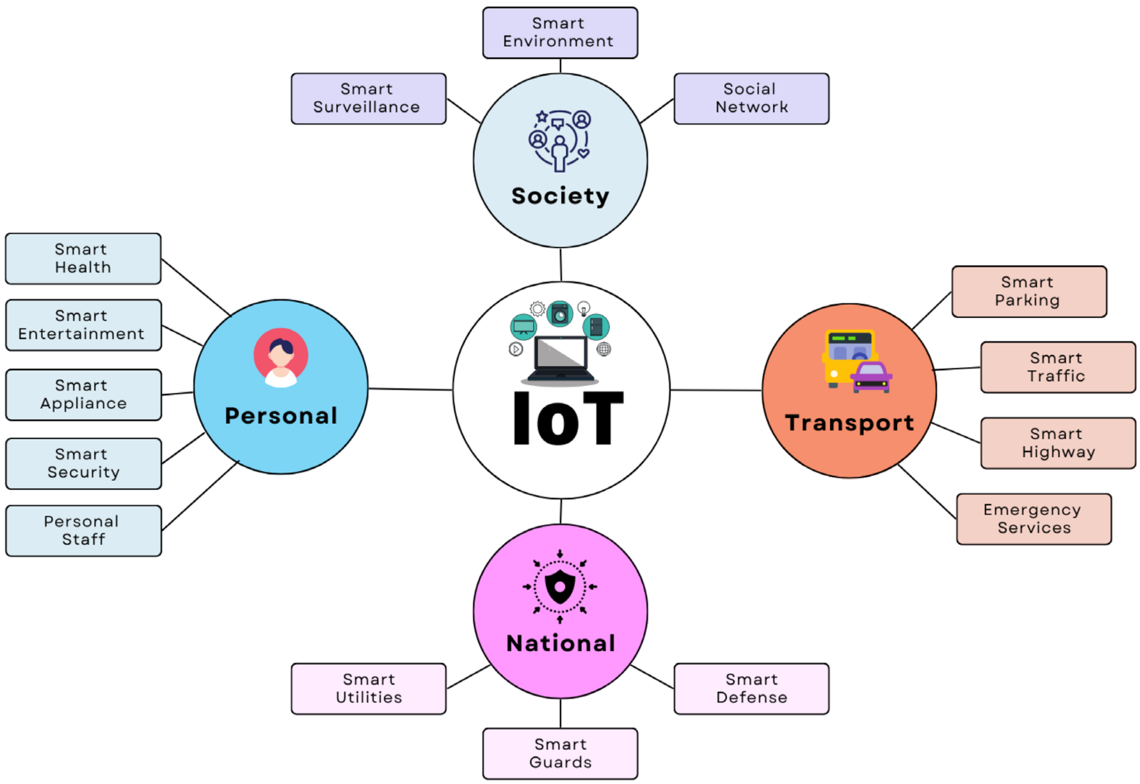

Fig. 2: Vehicle-in-the-Loop Testing: In the Creation of Real-World Conditions

comprehensive validation strategy for autonomous vehicles typically includes a combination of simulation, HIL testing, VIL testing, and extensive real-world trials to ensure the safety and reliability of the final system.

Addressing Cybersecurity Challenges in AV Embedded Systems

As autonomous vehicles become increasingly connected and reliant on complex software systems, cybersecurity has emerged as a critical concern in the development and validation of AV embedded systems. The potential consequences of a successful cyber attack on an autonomous vehicle could be severe, ranging from data breaches to compromised vehicle control, making robust cybersecurity measures an essential component of AV system design and validation.

The cybersecurity challenges faced by AV embedded systems are multifaceted, stemming from the vehicle's extensive connectivity, the complexity of its software architecture, and the critical nature of its operations. Key areas of concern include:

1. **Data Protection:** Safeguarding sensitive information such as location data, user profiles, and system configurations from unauthorized access or theft.
2. **Communication Security:** Ensuring the integrity and confidentiality of data exchanged between the vehicle and external systems, including V2X (Vehicle-to-Everything) communications.
3. **Software Integrity:** Protecting the AV's software systems from tampering, unauthorized modifications, or the introduction of malicious code.
4. Access Control: Implementing robust authentication and authorization mechanisms to prevent unauthorized access to vehicle systems and functions.
5. **Intrusion Detection and Prevention:** Developing systems capable of identifying and mitigating potential cyber threats in real-time.

Addressing these challenges requires a comprehensive approach that integrates security considerations throughout the entire development lifecycle of AV embedded systems. Some key strategies and best practices include:

1. Security by Design: Incorporating security features and considerations from the earliest stages of system architecture and design, rather than treating security as an afterthought.

2. Secure Boot and Firmware Updates: Implementing secure boot mechanisms to ensure the integrity of the system at startup, and developing secure processes for over-the-air (OTA) updates.
3. Encryption and Key Management: Utilizing strong encryption algorithms and robust key management systems to protect sensitive data and communications.
4. Network Segmentation: Designing the vehicle's internal network architecture to isolate critical systems and limit the potential spread of security breaches.
5. Continuous Monitoring and Updating: Implementing systems for ongoing monitoring of vehicle security status and rapid deployment of security patches and updates.
6. Penetration Testing and Vulnerability Assessment: Regularly conducting thorough security audits and penetration tests to identify and address potential vulnerabilities.
7. Compliance with Industry Standards: Adhering to established automotive cybersecurity standards and best practices, such as ISO/SAE 21434 for automotive cybersecurity engineering.

The validation of cybersecurity measures in AV embedded systems presents unique challenges due to the dynamic nature of cyber threats and the complexity of AV architectures. Traditional testing methodologies must be augmented with specialized security testing techniques, including:

1. Fuzz Testing: Inputting invalid, unexpected, or random data to various system interfaces to uncover potential vulnerabilities.
2. Threat Modeling: Systematically identifying potential security threats and attack vectors to guide testing and mitigation efforts.
3. Security-Focused Simulation: Developing simulation scenarios that specifically test the system's response to various cyber attack scenarios.
4. Hardware Security Testing: Evaluating the physical security of embedded systems, including resistance to side-channel attacks and hardware tampering.
5. Red Team Exercises: Conducting simulated cyber attacks by security experts to identify weaknesses in the system's defenses.

As the field of autonomous vehicle technology continues to evolve, so too do the cybersecurity challenges and solutions. Emerging technologies such as artificial intelligence and machine learning are being leveraged to develop more sophisticated intrusion detection systems and adaptive security measures. Additionally, blockchain technology is being explored as a potential solution for secure data sharing and identity management in connected vehicle ecosystems.

Collaboration within the automotive industry and with cybersecurity experts is crucial for addressing the ever-evolving landscape of cyber threats. Initiatives such as the Automotive Information Sharing and Analysis Center (Auto-ISAC) facilitate the sharing of cybersecurity information and best practices among automotive manufacturers and suppliers, helping to elevate the overall security posture of the industry (Table 2).[28-30]

Although autonomous vehicle technology is maturing, the standards and interoperability of AVs

Table 2: Validation Techniques For Autonomous Vehicles

Technique	Procedure
Simulation-Based Testing	Simulation-based testing involves virtual environments to simulate various driving scenarios and test embedded system responses under controlled conditions.
Hardware-in-the-Loop	Hardware-in-the-loop testing integrates real hardware with software simulations to validate the performance of embedded systems in autonomous vehicles.
Field Testing	Field testing involves deploying the autonomous vehicle on public roads to collect real-world data and validate the behavior of embedded systems under actual conditions.
Real-Time Monitoring	Real-time monitoring involves tracking the performance of embedded systems during vehicle operation to ensure proper functioning and identify issues as they occur.
Fault Injection	Fault injection introduces controlled errors into the system to test how the vehicle's embedded systems handle failures and maintain safety.
Model Verification	Model verification ensures that the system models used in the vehicle‚Äôs embedded systems match their real-world counterparts and perform accurately.

is already being brought to the forefront. Regulatory frameworks will also need to evolve to match technological development while treading a fine line between innovation and safety as well as ethical issues.

With embedded system development and validation, the future of AV development promises to be exciting as well as frustrating, with a firm reliance on safety and reliability as well as continuous innovation and rigorous testing. With the unfolding of these trends, these trends will define not only what automotive technology looks within the autonomous vehicle, but also will define the larger transportation ecosystem and urban landscapes of the future.

Conclusion

There is no shortage of frontier in technology, autonomous vehicles' embedded systems development and validation being one such a frontier encompassing software engineering, artificial intelligence, and very strict safety protocols. From concept to ready to roll autonomous vehicles, as we've explored in this article, the journey is a complex synthesis of simulation, hardware in the loop verification, vehicle in the loop validation and extensive regulatory compliance. However, the V model defines a structured way to develop by being very serious about the verification and validation of the entire process. Simulation technologies are the safest and most scalable means to perform testing on complex scenarios, whereas hardware in the loop and vehicle in the loop methods are good at linking between virtual testing and real life performance. Cyber security threats can not be ignored and require proactive measures to tackle updated threats. But the future of autonomous vehicles isn't just about technological advancement, it's about making safer, more efficient, and more accessible transportation systems. By refining and validating these complex embedded systems over time we get closer to a future where autonomous vehicles permeate your life, how you commute and how you interact with the world around you. But the path to autonomous vehicle technology ahead of it is a challenge, and opportunity. If we keep ourselves committed to safety, pursue new validation methodologies, and collaborate with each other as we do with other industries, autonomous vehicles will become fully mature and a new generation of mobility, which is safer, more efficient and more accessible for everyone will come into reality.

References:

6. Paniego, J. M., Libutti, L., Puig, M. P., Chichizola, F., De Giusti, L., Naiouf, M., & De Giusti, A. (2019, October). Unified power modeling design for various raspberry pi generations analyzing different statistical methods. In Argentine Congress of Computer Science (pp. 53-65). Cham: Springer International Publishing.
7. Perleberg, M. R., Goebel, J. W., Melo, M. S., Afonso, V., Agostini, L. V., Zatt, B., & Porto, M. (2018, February). ASIC power-estimation accuracy evaluation: A case study using video-coding architectures. In 2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS) (pp. 1-4). IEEE.
8. Piscitelli, R., & Pimentel, A. D. (2011, May). A high-level power model for mpsoc on fpga. In 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (pp. 128-135). IEEE.
9. Reimer, A., Schulz, A., & Nebel, W. (2006, October). Modelling macromodules for high-level dynamic power estimation of FPGA-based digital designs. In Proceedings of the 2006 international symposium on low power electronics and design (pp. 151-154).
10. Brettel, M., Friederichsen, N., Keller, M., & Rosenberg, M. (2017). How virtualization, decentralization and network building change the manufacturing landscape: an industry 4.0 perspective. FormaMente, 12.
11. Bril, R. J. (2004). Real-time scheduling for media processing using conditionally guaranteed budgets.
12. Brooks, D., Tiwari, V., & Martonosi, M. (2000). Wattch: A framework for architectural-level power analysis and optimizations. ACM SIGARCH Computer Architecture News, 28(2), 83-94.
13. Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., ... & Azumi, T. (2018, April). Autoware on board: Enabling autonomous vehicles with embedded systems. In 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS) (pp. 287-296). IEEE.
14. Pratap, N. L., Vallabhuni, R. R., Babu, K. R., Sravani, K., Kumar, B. K., Srikanth, A., ... & Mohan, K. S. K. (2020). A Novel Method of Effective Sentiment Analysis System by Improved Relevance Vector Machine. Australian Patent AU, 2020104414, 31.
15. Malavolta, I., Lewis, G., Schmerl, B., Lago, P., & Garlan, D. (2020, June). How do you architect your robots? State of the practice and guidelines for ROS-based systems. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice (pp. 31-40).
16. Mishra, B., & Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey. Ieee Access, 8, 201071-201086.
17. Ahmed, E., Ahmed, A., Yaqoob, I., Shuja, J., Gani, A., Imran, M., & Shoaib, M. (2017). Bringing computation closer toward the user network: Is edge computing

the solution?. IEEE Communications Magazine, 55(11), 138-144.

18. de Matos, E., Tiburski, R. T., Moratelli, C. R., Johann Filho, S., Amaral, L. A., Ramachandran, G., ... & Hessel, F. (2020). Context information sharing for the Internet of Things: A survey. Computer Networks, 166, 106988.
19. Alam, M., Rufino, J., Ferreira, J., Ahmed, S. H., Shah, N., & Chen, Y. (2018). Orchestration of microservices for iot using docker and edge computing. IEEE Communications Magazine, 56(9), 118-123.
20. Li, A., Wang, J., Baruah, S., Sinopoli, B., & Zhang, N. (2024, May). An empirical study of performance interference: Timing violation patterns and impacts. In 2024 IEEE 30th Real-Time and Embedded Technology and Applications Symposium (RTAS) (pp. 320-333). IEEE.
21. Liu, H., Wu, Y., Yu, Z., Vorobeychik, Y., & Zhang, N. (2023). Slowlidar: Increasing the latency of lidar-based detection using adversarial examples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5146-5155).
22. Nagaraju, V. S., Anusha, R., & Vallabhuni, R. R. (2020, December). A hybrid PAPR reduction technique in OFDM systems. In 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE) (pp. 364-367). IEEE.
23. Evans, I., Long, F., Otgonbaatar, U., Shrobe, H., Rinard, M., Okhravi, H., & Sidiroglou-Douskos, S. (2015, October). Control jujutsu: On the weaknesses of fine-grained control flow integrity. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 901-913).
24. Li, X., Gan, C., Gou, K., & Zhang, Y. (2019). A novel WDM-MAN enabling cross-regional reconfiguration and comprehensive protection based on tangent-ring. Optics Communications, 430, 416-427.
25. Marshall, G. J., Mahony, C. P., Rhodes, M. J., Daniewicz, S. R., Tsolas, N., & Thompson, S. M. (2019). Thermal management of vehicle cabins, external surfaces, and onboard electronics: An overview. Engineering, 5(5), 954-969.
26. Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. IEEE transactions on pattern analysis and machine intelligence, 24(4), 509-522.
27. Demiröz, B. E., Salah, A. A., Bastanlar, Y., & Akarun, L. (2019). Affordable person detection in omnidirectional cameras using radial integral channel features. Machine Vision and Applications, 30, 645-655.
28. Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.
29. Boukhayma, A., Bem, R. D., & Torr, P. H. (2019). 3d hand shape and pose from images in the wild. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10843-10852).
30. Babu, P. A., Sridhar, P., & Vallabhuni, R. R. (2022, February). Fake currency recognition system using edge detection. In 2022 Interdisciplinary Research in Technology and Management (IRTM) (pp. 1-5). IEEE.
31. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).
32. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.
33. Temel, D., Alshawi, T., Chen, M. H., & AlRegib, G. (2019). Challenging environments for traffic sign detection: Reliability assessment under inclement conditions. arXiv preprint arXiv:1902.06857.
34. Wali, S. B., Abdullah, M. A., Hannan, M. A., Hussain, A., Samad, S. A., Ker, P. J., & Mansor, M. B. (2019). Vision-based traffic sign detection and recognition systems: Current trends and challenges. Sensors, 19(9), 2093.
35. Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks, 32, 323-332.
36. Kavitha, M. (2020). A Ku band circular polarized compact antenna for satellite communications. National Journal of Antennas and Propagation, 2(2), 15–20.
37. Tirmare, A. H., Mali, P. S., Shirolkar, A. A., Shinde, G. R., Patil, V. D., & Tirmare, H. A. (2024). VLSI Architecture-Based Implementation of Motion Estimation Algorithm for Underwater Robot Vision System. Journal of VLSI Circuits and Systems, 6(2), 115–121. https://doi.org/10.31838/jvcs/06.02.13
38. Devi, G., Reena, P., Yuvarani, M., Kavitha, M., & Surendar, A. (2016). High-speed image searching for human gait feature selection. International Journal of Communication and Computer Technologies, 4(2), 88-95.
39. Prasath, C. A. (2023). The role of mobility models in MANET routing protocols efficiency. National Journal of RF Engineering and Wireless Communication, 1(1), 39-48. https://doi.org/10.31838/RFMW/01.01.05
40. Abdullah, D. (2024). Design and implementation of secure VLSI architectures for cryptographic applications. Journal of Integrated VLSI, Embedded and Computing Technologies, 1(1), 21-25. https://doi.org/10.31838/JIVCT/01.01.05
41. Abdullah, D. (2024). Strategies for low-power design in reconfigurable computing for IoT devices. SCCTS Transactions on Reconfigurable Computing, 1(1), 21-25. https://doi.org/10.31838/RCC/01.01.05

42. Rahim, R. (2024). Review of modern robotics: From industrial automation to service applications. Innovative Reviews in Engineering and Science, 1(1), 34-37. https://doi.org/10.31838/INES/01.01.08
43. Rahim, R. (2024). Scalable architectures for real-time data processing in IoT-enabled wireless sensor networks. Journal of Wireless Sensor Networks and IoT, 1(1), 44-49. https://doi.org/10.31838/WSNIOT/01.01.07
44. Cheng, L. W., & Wei, B. L. (2024). Transforming smart devices and networks using blockchain for IoT. Progress in Electronics and Communication Engineering, 2(1), 60–67. https://doi.org/10.31838/PECE/02.01.06