Journal of Advanced Research in Operating System Development and Evolution
Volume 1, Issue 1 - 2025, Pg. No. 32-38
Peer Reviewed Journal

Review Article

The Role of Real-Time and Embedded Systems in Autonomous and Smart Technologies

Alka Gautam

Student, Indian Institute of Technology Bhilai, Chhattisgarh, India

INFO

E-mail Id:
alkagautam@gmail.com
Orcid Id:
https://orcid.org/0009-0004-5563-0664

How to cite this article:
Gautam A. The Role of Real-Time and Embedded Systems in Autonomous and Smart Technologies . *J Adv Res Oper Syst Dev Evol* 2025; 1(1): 32-38.

Date of Submission: 2025-01-18
Date of Acceptance: 2025-03-08

ABSTRACT

Real-time and embedded systems play a critical role in the operation and success of autonomous systems and smart technologies. These systems, often employed in sectors such as automotive, robotics, healthcare, and smart cities, are essential for ensuring responsiveness, reliability, and safety. This review explores the relationship between real-time and embedded systems, their significance in enabling autonomous and smart technologies, the challenges involved in their design, and the future trends driving innovation in these fields. We examine key architectures, operating systems, and the evolution of embedded hardware that have facilitated advancements in real-time processing and decision-making, which are crucial for the performance of modern autonomous systems.

Keywords: Real-Time Systems, Embedded Systems, Autonomous Systems, Smart Technologies, Robotics, Edge Computing, IoT, Safety, Embedded Hardware

Introduction

The integration of real-time and embedded systems within autonomous and smart technologies has led to the emergence of applications that demand the highest levels of precision, responsiveness, and reliability. These systems are crucial in ensuring that devices and machines respond accurately to environmental changes or user commands in real time. Real-time systems are designed to guarantee a timely response to specific events, while embedded systems are specialized for executing dedicated tasks within a broader system framework.

In the context of autonomous systems, such as self-driving cars, drones, and robotics, real-time and embedded systems are indispensable. These systems must continuously process input from sensors (e.g., cameras, radar, LiDAR) and immediately make decisions such as braking, steering, or avoiding obstacles, all while ensuring the safety and efficiency of operation. For example, in autonomous driving, a real-time operating system (RTOS) must prioritize tasks, such as sensor data processing and decision-making, to avoid delays that could lead to accidents. The processing must occur with absolute precision and under stringent time constraints, with no tolerance for delay or failure.[1]

Similarly, smart technologies—which include a wide range of devices and applications from smart homes and smart cities to healthcare devices—rely on real-time processing and embedded systems to function effectively. These systems enable continuous monitoring and automation. For example, in a smart home, embedded systems control lighting, heating, and security systems based on real-time sensor inputs, ensuring the environment adapts to user preferences or external conditions. In healthcare, embedded systems process data from medical devices in real time, providing immediate feedback that can be life-saving, such as adjusting insulin doses for diabetic patients or monitoring vital signs in intensive care units (ICUs).[2]

Real-Time Processing ensures that the devices involved in autonomous systems can respond within a strict, pre-defined time window. For example, self-driving cars must be capable of making decisions in fractions of a second. If a car's onboard system detects an obstacle ahead, the system must decide in real-time whether to apply the brakes, swerve, or take other actions to avoid a collision. Any delay could lead to dangerous outcomes, especially in environments where fast reactions are essential.

Embedded Systems are the hardware-software combinations that provide specialized functionalities. In autonomous systems, these systems are integrated into various components of the vehicle or robot. For example, the embedded systems in a robot enable it to process visual data from cameras or other sensors and make movement decisions accordingly. They are also designed to be energy-efficient, compact, and reliable, meeting the specific needs of the application they serve. A drone's embedded system controls the flight operations, adjusts the path based on real-time inputs, and ensures that the drone remains stable during flight.

In the broader scope of smart technologies, real-time embedded systems enable more seamless and intelligent user experiences. They work as the core element that processes sensory data and adjusts the systems in real time. For example, in smart grid technologies, embedded systems in electrical meters and sensors communicate data continuously to ensure efficient energy distribution and grid stability. Similarly, in industrial automation, real-time systems optimize processes by adjusting machinery operations based on feedback from sensors, improving productivity and safety.[3]

Ultimately, the interplay between real-time processing and embedded systems is essential in enabling the smooth and efficient functioning of autonomous and smart technologies. As these technologies continue to evolve and become more integrated into daily life, the demand for faster, more reliable, and more secure real-time systems will only increase. The ongoing advancements in embedded hardware, software optimization, and real-time operating systems will drive innovations that further enhance the capabilities and applications of these systems, making autonomous and smart technologies more capable and ubiquitous.

Real-Time and Embedded Systems: An Overview

Real-Time Systems

Real-time systems are specialized computing systems designed to process input or events and provide responses within strict time limits. The importance of these systems lies in their ability to perform tasks on time, which is crucial in a wide range of mission-critical applications. Real-time systems are categorized into hard and soft real-time systems based on their response time requirements:

- **Hard Real-Time Systems:** These systems are designed with a stringent requirement: they must meet specific deadlines with zero tolerance for delays. Missing a deadline in a hard real-time system could result in catastrophic outcomes, such as system failures, data loss, or even loss of life in critical applications. For example:
- **Aerospace Systems:** In satellite control systems, the real-time processing of sensor data and control signals is critical to maintain operational safety. Any delay can compromise the mission.
- **Medical Devices:** In pacemakers or ventilators, any delay in processing vital data could lead to life-threatening consequences.
- **Autonomous Vehicles:** Real-time decision-making is vital for avoiding collisions and ensuring safety on the road.[4]
- **Soft Real-Time Systems:** While these systems still have time constraints, they are more flexible and can tolerate occasional delays without causing serious harm. Missing a deadline in a soft real-time system may cause performance degradation but does not lead to complete system failure. Examples include:
- **Multimedia Systems:** In video streaming or gaming applications, a slight delay might affect the user experience but does not have critical consequences.
- **Telecommunications Systems:** In some telecommunication applications, delays may affect call quality or video streaming but will not result in complete service disruption.
- **Non-Critical Industrial Control Systems:** For tasks such as automated assembly lines, missing some deadlines may reduce efficiency but not cause severe issues.

The success of real-time systems depends on both predictability and timeliness. Predictability refers to the system's ability to execute tasks within a defined time frame. Real-Time Operating Systems (RTOS) are crucial to managing the scheduling of real-time tasks. These specialized operating systems are designed to prioritize tasks, allocate resources effectively, and guarantee that time-critical tasks are executed without delay. Key features of RTOS include:

- **Task Scheduling:** RTOS manages the execution order of tasks to ensure that deadlines are met.
- **Interrupt Handling:** RTOS has advanced interrupt handling mechanisms to quickly react to real-time events.

- **Resource Allocation:** Ensures that system resources, such as CPU time, memory, and input/output devices, are allocated to high-priority tasks.

Embedded Systems

An embedded system refers to a dedicated computing system that is part of a larger system and is designed to perform specific functions within that system. Unlike general-purpose computers, embedded systems are highly optimized for performance, cost-effectiveness, and efficiency. They are often built with a focus on minimizing power consumption, as they are typically used in devices that require long operational lifetimes, especially in battery-powered environments.[5]

Key characteristics of embedded systems include:

- **Dedicated Functionality:** Embedded systems are specialized for a particular application. For example, an embedded system in a washing machine controls washing cycles, water levels, and spin speeds, while an embedded system in an automobile might control engine management, anti-lock braking systems (ABS), or infotainment systems. Unlike general-purpose computing systems, embedded systems are not designed to run multiple applications; they focus on executing a single task or a set of related tasks with efficiency.
- **Real-Time Processing:** Many embedded systems require real-time data processing, particularly in applications where timely responses are necessary to ensure the correct functioning of the overall system. Real-time embedded systems are used in applications such as:
- **Automotive Systems:** For controlling engine functions, airbags, and braking systems, embedded systems must respond quickly and reliably to sensor data.
- **Industrial Control Systems:** Embedded systems control machinery, monitor production lines, and ensure automated processes are running smoothly in real-time.
- **Medical Devices:** In devices like insulin pumps, pacemakers, and diagnostic equipment, embedded systems need to process sensor data and provide immediate feedback.
- **Low Power Consumption:** A key feature of embedded systems is their ability to operate on limited power sources. Battery-powered devices, such as smartphones, wearables, and IoT devices, require embedded systems to manage power efficiently. Power efficiency is especially important in devices that are expected to run for long periods without frequent battery replacement or recharging. Strategies to reduce power consumption in embedded systems include:
- **Sleep Modes:** Embedded systems can enter low-power states when idle, saving battery life.
- **Optimized Hardware:** Using low-power processors and components can significantly reduce energy usage.
- **Efficient Software Design:** Efficient algorithms and minimalistic software design help to reduce the power requirements of the embedded system.

Applications of Real-Time and Embedded Systems

The widespread use of real-time and embedded systems spans a broad spectrum of industries, making them crucial for modern technological advancements. Key applications include:

- **Autonomous Vehicles:** Real-time systems and embedded sensors are integral to self-driving cars. These systems continuously process data from cameras, LiDAR, radar, and GPS to make real-time decisions for navigation, object detection, and collision avoidance.
- **Healthcare:** In medical devices such as pacemakers, infusion pumps, and diagnostic equipment, embedded systems must provide real-time responses to ensure patient safety and effective treatment.
- **Consumer Electronics:** Devices like smart televisions, washing machines, and thermostats rely on embedded systems to manage operations efficiently. These systems provide real-time data processing and user interaction capabilities.
- **Industrial Automation:** Embedded systems control manufacturing robots, automated assembly lines, and monitoring systems. These systems must meet real-time requirements to ensure that production processes are efficient and safe.
- **IoT Devices:** Smart home devices, wearable fitness trackers, and environmental sensors are powered by embedded systems. These systems process data from sensors, provide real-time updates, and allow for automation and remote control.

In conclusion, real-time and embedded systems are foundational to the functioning of many of today's most advanced technologies. They enable rapid decision-making, efficient data processing, and reliable performance, all of which are essential for the success of autonomous systems, smart devices, and various other applications. As technology continues to evolve, these systems will play an even more critical role in driving innovation and enabling smarter, safer, and more efficient products and services.[6]

Autonomous Systems and Real-Time Processing

Autonomous Vehicles

Autonomous vehicles (AVs) rely heavily on real-time processing and embedded systems to operate safely and efficiently in dynamic and unpredictable environments. These vehicles must continuously collect and analyze data from various sensors to understand their surroundings and make real-time driving decisions. The integration of sensors such as cameras, LiDAR (Light Detection and Ranging), and radar allows the vehicle to perceive its environment in high detail, while the real-time system processes this data to drive actions such as braking, steering, and acceleration. The key components of real-time embedded systems in autonomous vehicles include:

- **Sensor Fusion:** Autonomous vehicles rely on the fusion of data from different sensors to create a unified understanding of the environment. For example, LiDAR provides depth perception, cameras offer visual data, and radar detects moving objects. The real-time system must efficiently process this sensor data to create an accurate model of the surroundings and allow the vehicle to make quick decisions.
- **Decision-Making Algorithms:** Once the real-time system has gathered and processed sensor data, it must make decisions based on algorithms that may include machine learning models, control algorithms, and path planning systems. For instance, machine learning models are used to predict the movements of pedestrians, cyclists, and other vehicles, allowing the autonomous car to adjust its speed or direction. This requires high-performance embedded systems to process large amounts of data quickly and ensure that decisions are made within milliseconds to guarantee the safety of the passengers and others on the road.
- **Safety and Redundancy:** Real-time embedded systems also support safety mechanisms, including redundancy, to ensure that critical systems—such as emergency braking or lane-keeping assistance—function reliably. If one system fails, another takes over to prevent accidents, ensuring that the AV meets the hard real-time deadlines critical for safety.[7]

Drones and Robotics

Real-time embedded systems are also central to the functioning of drones and robotic systems, where they provide the processing power needed to handle complex, real-time tasks such as navigation, object detection, and control. The ability of drones and robots to respond to dynamic environments requires high levels of precision and responsiveness, which is provided by real-time operating systems and embedded hardware.

- **Navigation and Control:** Drones and robots use a combination of sensors—such as gyroscopes, accelerometers, cameras, and GPS—to navigate their environments. Real-time embedded systems process the data from these sensors and control actuators (motors, servos, etc.) to adjust their position, orientation, or movement accordingly. For example, in a drone, real-time systems manage the flight path by adjusting for wind conditions, obstacles, and changes in altitude.
- **Real-Time Decision Making:** Drones and robots must make decisions in real-time, often autonomously. For instance, a drone flying through a forest must detect and avoid obstacles, adjust its flight path, and monitor its battery level. Each of these tasks requires real-time data processing to ensure smooth operation. Similarly, industrial robots working on assembly lines must be able to identify and manipulate objects, which requires real-time data processing to detect objects and control the robot's movement.
- **Communication and Coordination:** In swarm robotics, where multiple robots collaborate to perform tasks, real-time embedded systems enable communication and coordination between robots. Each robot must process local data and share it with other robots in real time to ensure synchronized actions. This is crucial in applications like warehouse automation, where robots must work in unison to efficiently transport goods.

Smart Technologies and Embedded Systems

The development of real-time embedded systems has been essential to the growth of smart technologies, which have become pervasive in modern life. These systems enable intelligent decision-making, continuous monitoring, and seamless integration of devices within larger ecosystems. From smart homes to industrial automation, embedded systems help manage and process data in real time, providing reliable and efficient operation. Below are key applications of embedded systems in the smart technology landscape:

Internet of Things (IoT)

The Internet of Things (IoT) refers to the network of interconnected devices that communicate and exchange data over the internet. IoT devices range from wearable health monitors and smart thermostats to smart appliances and industrial sensors. Real-time embedded systems are the backbone of IoT, enabling these devices to collect, process, and respond to data in real-time. Key functionalities include:

- **Data Collection and Processing:** Many IoT devices are designed to collect sensor data continuously—whether it is temperature, motion, humidity, or health metrics. Real-time embedded systems ensure that data is processed locally (at the edge of the network), allowing for rapid decision-making and reducing the need to send data to cloud servers. This improves performance and reduces latency.
- **Edge Computing:** Real-time embedded systems are crucial for edge computing, where data processing occurs close to the data source, such as in a smart thermostat or health monitoring device. By processing data locally, edge computing minimizes bandwidth use and reduces the time it takes to make decisions. For example, an IoT-enabled smart city might use edge computing to monitor traffic, detect accidents, and dynamically control traffic lights in real-time, minimizing congestion and improving traffic flow.
- **Low Power Operation:** Many IoT devices, such as wearable fitness trackers or environmental sensors, must operate on limited battery power. Real-time embedded systems are optimized for low-power consumption, allowing these devices to operate for extended periods without frequent recharging or battery changes.[8]

Smart Cities and Industrial Automation

The concept of smart cities encompasses the use of connected devices and systems to improve urban living. Embedded systems play a critical role in managing infrastructure, transportation, energy, and environmental monitoring. Real-time processing ensures that smart city systems respond quickly and efficiently to changing conditions. Examples include:

- **Smart Traffic Management:** Real-time embedded systems are used in traffic management to monitor traffic flow, adjust signal timings, and respond to accidents or traffic congestion. Smart traffic lights dynamically change their cycle based on real-time data from sensors, ensuring smooth traffic flow and reducing wait times at intersections.
- **Smart Grids:** Embedded systems in smart grids enable real-time monitoring and control of energy distribution. These systems use real-time data from sensors to optimize energy use, detect outages, and balance load distribution across the grid. Real-time decision-making is crucial to ensure that energy is delivered efficiently and reliably.
- **Industrial Automation:** Embedded systems are also fundamental to the operation of industrial automation systems, which manage and optimize manufacturing processes. In industries such as automotive, electronics, and pharmaceuticals, real-time data collection and processing allow for adaptive control and quick responses to changing conditions on the production floor. Real-time systems help manage robots, sensors, and actuators to ensure that manufacturing lines run smoothly and that any faults or anomalies are addressed promptly.

The combination of real-time and embedded systems has paved the way for advancements in both autonomous systems and smart technologies. From self-driving cars and drones to smart cities and IoT devices, these systems ensure efficient, reliable, and responsive operations across a wide range of applications. As technology continues to evolve, the importance of real-time data processing, sensor integration, and embedded control will only increase, driving further innovation in autonomous and smart technologies. These systems will continue to shape the future of industries such as healthcare, automotive, industrial automation, and environmental monitoring, providing critical infrastructure and enabling safer, more efficient, and intelligent systems.

Challenges in Real-Time and Embedded Systems for Autonomous and Smart Technologies

Despite the tremendous potential of real-time and embedded systems in powering autonomous and smart technologies, there are several challenges that must be overcome to ensure their effective implementation and continued development. Below are some of the key challenges faced in real-time embedded systems for these applications:

Power Constraints

Many autonomous and smart technologies are deployed in battery-operated devices such as drones, IoT devices, and wearable health monitors, where power consumption is a critical factor. These systems often need to balance high computational demand and real-time processing requirements while maintaining low power usage. For example:

- **Low-Power Operation:** Drones must operate for extended periods while consuming minimal power. Real-time embedded systems need to process sensor data and make decisions in real time, but they must do so while conserving battery life.
- **Power Efficiency in IoT:** IoT devices, such as environmental sensors or health trackers, must operate continuously and transmit data periodically. The design of real-time embedded systems must ensure that they do not overconsume energy, particularly when they are powered by small batteries or energy-harvesting technologies.[9]

Solutions to address power constraints involve optimizing both the software and hardware. For example, low-power microcontrollers, energy-efficient sensors, and sleep modes that reduce power consumption during idle periods can help achieve the balance between real-time processing and power efficiency.

Safety and Security

Autonomous and smart technologies are often deployed in critical applications, such as self-driving cars, industrial robotics, and healthcare devices, where failure can lead to severe consequences. Ensuring the safety and security of real-time embedded systems is paramount:

- **Cybersecurity:** Autonomous systems can be vulnerable to cyberattacks, such as hacking, spoofing, or denial-of-service attacks. These threats can compromise the integrity of decisions made by the system and cause catastrophic failures. For instance, a self-driving car hacked to ignore traffic signs could result in a crash.
- **Safety:** Many autonomous systems, such as aerospace applications and medical devices, operate in environments where safety is critical. For example, embedded systems in drones must be designed to handle both expected and unexpected conditions, such as battery failure, sensor malfunctions, or sudden weather changes. Ensuring the system fails safely (i.e., fails in a way that minimizes harm) is a significant challenge.

To address these challenges, advanced security protocols, redundancy mechanisms, and fail-safe designs are needed. Real-time embedded systems must incorporate encryption, firewalls, and secure boot mechanisms to prevent unauthorized access and tampering. Additionally, real-time monitoring and alerting systems are critical to detect and address any system anomalies that might compromise safety or security.

Resource Limitations

Embedded systems are often constrained by their hardware resources, such as processing power, memory, and storage. This resource limitation can hinder the ability to implement complex algorithms, such as machine learning models, artificial intelligence (AI), or computer vision algorithms, which are increasingly integral to autonomous systems. Some of the key challenges include:

- **Processing Power:** Real-time decision-making requires high-performance computation. Embedded systems are typically built with low-power processors, which might not be able to handle complex tasks such as real-time image processing or running machine learning models.
- **Memory and Storage:** Autonomous systems, particularly those in vehicles or drones, need to process vast amounts of sensor data, but memory and storage are often limited. For example, an autonomous vehicle must store and process real-time camera data, LiDAR scans, and radar signals, but its embedded system may not have sufficient memory to store all the required data simultaneously.

To overcome these limitations, hardware acceleration (e.g., GPUs or FPGAs), edge computing, and cloud integration can be used. These technologies allow for offloading processing tasks to more powerful systems or distributing workloads across devices to achieve better performance without overloading individual embedded systems.[10]

Real-Time Communication

Reliable real-time communication is essential for autonomous systems and IoT devices, particularly those that require coordination and interaction with other devices in a network. Several communication challenges include:

- **Network Latency:** Autonomous vehicles, drones, and robots rely on constant communication with other devices or a central system for decision-making. Network delays can disrupt the timely exchange of data, which may lead to errors in decision-making. For instance, delays in traffic signal updates could cause an autonomous vehicle to miscalculate its position.
- **Bandwidth Limitations:** Real-time systems often need to transmit large amounts of data, especially in environments with dense sensor networks (e.g., smart cities or industrial IoT). Limited bandwidth can affect the speed and accuracy of data transmission, impacting real-time performance.
- **Communication Errors:** Errors in communication, such as dropped packets or corrupted data, can also disrupt real-time decision-making. In autonomous systems, such errors can lead to incorrect decisions, potentially causing safety issues.

Addressing communication challenges involves the implementation of low-latency communication protocols, 5G networks, mesh networks, and redundant communication channels to ensure that real-time data can be transmitted quickly and reliably.

Future Trends

As autonomous systems and smart technologies continue to evolve, real-time embedded systems will play an even more critical role in ensuring reliable and efficient performance. The following future trends are expected to shape the field:

Machine Learning and AI in Real-Time Systems

Machine learning and artificial intelligence (AI) are expected to be increasingly integrated into real-time embedded systems. These algorithms will enable autonomous decision-making in more complex, dynamic environments. Some notable developments include:

- **AI-Driven Decision Making:** Autonomous vehicles and drones will use AI to improve their decision-making capabilities, such as detecting anomalies in sensor data or predicting human behavior in real time.
- **Hardware Acceleration for AI:** The integration of specialized hardware accelerators (e.g., neural processing units (NPUs), FPGAs) will enable real-time embedded systems to run complex machine learning models efficiently without sacrificing performance.[11]

5G and Edge Computing

The deployment of 5G networks and the growth of edge computing will significantly enhance real-time embedded systems. 5G offers low-latency and high-bandwidth communication, which is essential for the real-time transfer of data. Edge computing, where data is processed closer to the source, will reduce reliance on cloud servers and improve response times in time-sensitive applications such as autonomous vehicles and smart cities.

- **5G Networks:** The low-latency and high-throughput capabilities of 5G will allow autonomous systems to make faster, more accurate decisions. For instance, autonomous vehicles could rely on real-time communication with other vehicles and infrastructure to improve traffic flow and safety.
- **Edge Computing:** Edge computing will enable devices to process data locally, reducing reliance on cloud-based systems and minimizing network latency. This is especially beneficial for time-sensitive applications such as industrial automation, smart homes, and healthcare monitoring.

Quantum Computing

While still in the early stages, quantum computing holds the potential to revolutionize real-time processing in embedded systems. Quantum computers can process massive amounts of data and perform complex calculations at unprecedented speeds, which could enhance the capabilities of real-time embedded systems in applications like AI, cryptography, and big data analysis.

- **Quantum-Resistant Algorithms:** As quantum computing advances, embedded systems will need to incorporate quantum-resistant encryption to protect sensitive data from being compromised by quantum-enabled attacks.[12]

Conclusion

Real-time and embedded systems are the cornerstone of autonomous systems and smart technologies. They provide the foundation for critical functions like decision-making, control, and data processing in environments that demand high reliability and precision. Despite significant advancements, challenges such as power efficiency, security, resource constraints, and real-time communication remain prominent.

As technology progresses, real-time embedded systems will continue to evolve, supported by innovations such as AI integration, 5G networks, edge computing, and quantum computing. These advancements will shape the future of industries such as automotive, healthcare, robotics, and IoT, driving further improvements in autonomous technologies and smart systems across the globe.

References

1. Hsieh H, Huang H, Lee H. Real-time systems and applications. IEEE Trans Ind Electron. 2019; 66(4):345-360.
2. Thomas J, Ford G. Embedded systems in smart technologies. J Comput Sci Eng. 2021; 32(2):98-107.
3. Williams R. Real-Time Operating Systems: Design and Implementation. 3rd ed. Boston: Pearson; 2020.
4. Xu Y, Li Z, Chen P. Security challenges and solutions in autonomous systems. Comput Sci Rev. 2020; 34(1):142-155.
5. Yang T, Yang Y, Zhang S. Machine learning in real-time embedded systems. J Embedded Comput. 2020; 42(3):123-134.
6. Agha M, Naqvi R, Batool R, et al. Real-time operating systems for embedded systems: A review of features and challenges. J Real-Time Systems. 2021;57(1):1-22.
7. Li X, Zhang Y, Wang L, et al. Autonomous driving: Technologies, challenges, and future directions. IEEE Trans Autom Sci Eng. 2020;17(4):1341-1358.
8. Malek S, Batool S, Abbas H. IoT-based smart city applications: Integration, design, and future perspectives. Future Gener Comput Syst. 2022;108:141-158.
9. Patel S, Kim J, Kumar D, et al. Embedded systems in healthcare: Review of technologies, applications, and future trends. IEEE Access. 2020;8:53369-53385.
10. Moore E, Zhang T, Sakamoto M. Challenges in real-time data communication for autonomous systems. Int J Comput Commun Eng. 2020;9(2):45-52.
11. Sharma V, Bhatti N, Singh S. Power-efficient embedded systems for Internet of Things applications: A survey. J Low Power Electron. 2021;17(3):200-210.
12. Kumar A, Singh G, Sharma P, et al. Security challenges in autonomous vehicles and smart systems. J Cybersecurity. 2021;7(4):1-11.