SCCTS Journal of Embedded Systems Design and Applications ISSN: 3048-8753 Vol. 2, No. 1, 2025 (pp. 36-42)
RESEARCH ARTICLE ECEJOURNALS.IN

Ultra Low Latency Communication in Wireless Sensor Networks: Optimized Embedded System Design

K. Booch[1], L.H. Wehrmeister[2], P. Parizi[3]*

[1-3]Department of Production and Management Engineering, Democritus University of Thrace, Xanthi, 67132, Greece

Keywords:
Hardware Descriptions for Embedded Systems;
Embedded System Modeling;
Embedded System Reliability;
Communication Protocols in Embedded Systems;
Cyber-Physical Systems

DOI: 10.31838/ESA/02.01.04

Received : 13.07.24
Revised : 24.10.24
Accepted : 18.12.24

Abstract

With the fast development of wireless technologies and embedded systems, dramatic applications of such technologies in industrial automation, healthcare monitoring, and smart infrastructure have been achieved. The critical need of highly reliable and fast communication between these sensors, actuators and controllers underlies these applications. From real time data transmission with your wireless sensor network to the edge of the envelope, this article explores cutting edge techniques and protocols for designing embedded systems to communicate with ultra low latency in wireless networks. We shall explore the challenges of millisecond level responsiveness engineers and researchers meet while developing embedded system. We will study the multiphased method to build communication systems that can satisfy stringent control specifications from hardware as well as software optimization. Welcome to our journey in the wireless communication world with low latency where every micro second matters and innovation is driving the future of the connected devices.

How to cite this article: Booch K, Wehrmeister LH, Parizi P (2025). Ultra Low Latency Communication in Wireless Sensor Networks: Optimized Embedded System Design. SCCTS Journal of Embedded Systems Design and Applictions, Vol. 2, No. 1, 2025, 36-42

Requirement of Low Latency in Wireless Sensor Networks

As with latency in computer networks and sensor networks, the term latency in the wireless sensor networks is an important criteria for determining overall performance and effectiveness of the whole system. For applications with strict needs for real time data processing and responsive actions to environmental changes, low latency communication is a necessity. For example, industrial control systems rely on fast information exchange between sensor and actuator to keep things stable and not fall down catastrophically. When considering the potential consequences of sluggish communication, the importance of minimizing delay only increases. For automotive systems a few milliseconds of delay can mean the difference between a successful collision avoiding maneuver and a severe crash. Likewise, in medical devices presence of timely transmitted vital signs is essential for patient safety, as well as effective treatment. When we begin to explore embedded systems, it is clear we can't just have ultra-low latency because it is convenient, it is often crucial to ensure our critical applications run reliably and functionally. The problem is how to design systems that provide this data constantly, no matter the wireless conditions or the resource constraints.[1-6]

Design Problems of Low Latency Embedded System

Many challenges exist in creating embedded systems capable of ultra low latency communication in both the hardware and software domains. The inherent unpredictability of wireless channels present one of the major impediments; there can be varying levels of delay and packet loss. But signal quality suffers due to interference, multipath fading, and other physical phenomena and it can take several transmissions to get the signal right. Another large obstacle is limited

computational resources in many embedded devices. Because embedded systems typically operate in constrained processing power, memory, and energy budgets, they are very different from conventional large scale powerful desktop computers or servers. Therefore, every part of the system including the choice of the microcontroller and the efficiency of the communication protocols need to be opted carefully. Complexity is also introduced from the trade off between performance and power consumption in the design process. More powerful and faster processors will certainly reduce processing latency, but at the cost of consuming even more energy, which may render them unsuitable for energy harvesting or battery powered devices. There's a careful balance to be made for responsiveness versus power efficiency, and frankly, it's not always easy.

Finally, a major problem is that real time systems require deterministic behavior. Operating systems and network stacks used in traditional way may cause unpredictable delays to make it very hard to ensure consistent low latency performance. Challenges in overcoming these challenges require the holistic approach that looks into all aspects of the embedded system design from the hardware architecture to the application level software.

Low Latency Communication Hardware Considerations

For ultra low latency communication design of embedded systems, there are certain hardware components to be chosen. The performance capabilities of the system is specifically determined by the choice of microcontroller or microprocessor. However, for applications with minimum delay, processors with high clock speed and with small instruction pipeline are often preferred. Nevertheless, it must be judged in light of power consumption implications, such as for battery powered devices. Another important piece is the radio frequency (RF) transceiver, which directly affects communication latency. Advanced modulation schemes and fast turnaround times between receive and transmit modes are common in the modern transceivers designed for low latency applications. Even more specialized transceivers include a hardware accelerator for packet processing in order to reduce the overall communication delay.

System responsivity is strongly tied to memory architecture. Carefully designed memory hierarchies enable the rollling off power peaks and performance lags, and fast, low latency memory types like SRAM can get us the critical data very quickly. In some cases a more traditional way, is to off load certain tasks to dedicated hardware accelerators or co processors leaving main processor busy with time critical tasks. For example, wireless link quality and reliability is highly affected by antenna design and selection. In some cases, MIMO (Multiple Input Multiple output) Systems or directional antennas might be capable of helping to improve signal strength, reduce interference, thus reducing the transmit retransmission rates and reducing the consecutive delay.

Minimizing Delay: Software Optimization Techniques.

Hardware provides the foundation but software optimisation is equally important to ultra low latency communication. Real time operating systems (RTOS) is a frequently used at the operating system level , providing such deterministic behavior and fine timing control. The features like priority based scheduling and interrupt handling on these specialized operating systems are necessary to present consistent low latency performance. The other part of software optimization is efficient memory management. Techniques like memory pooling, using pre allocated blocks of memory rather than dynamic allocation can help reduce uncertainty from unpredictable delays like memory fragment and allocation overhead. Just as processing time and memory access patterns can be minimized by careful data structure design and algorithm selection, the current situation can be improved.

Instead, small protocols have been developed to be used in the networking stack for low latency communication. One of the most common ways to achieve these is specialized protocols which often eliminate most handshakes, reduce header overhead and introduce efficient error correction mechanisms. More advanced systems may even use them to decouple the communication scheme from the traditional layers, managing custom, application specific communication schemes directly on top of the physical layer (Table 1).

Moreover, the processing time can be further reduced with code optimization techniques like loop unrolling, function inlining, and use of SIMD (Single Instruction, Multiple Data) instructions. Furthermore, by taking care of interrupt handling and task prioritization, carefully, time critical operations can happen at once, without delay on less crucial tasks.

Table 1: Design Elements for Low Latency Wireless Communication

Element	Contribution
Low-Latency Protocols	Low-latency protocols minimize the time taken for data to travel across the network, reducing delays in sensor data transmission.
Optimized Routing	Optimized routing techniques ensure that data packets follow the fastest, least-congested path, minimizing transmission delays in the network.
Data Compression	Data compression reduces the size of transmitted data, decreasing the time required to send data across the network and lowering latency.
Energy-Efficient Hardware	Energy-efficient hardware minimizes power consumption, allowing sensor nodes to operate for longer periods, crucial for real-time communication.
Parallel Processing	Parallel processing distributes tasks across multiple cores or processors, speeding up data handling and reducing latency in embedded systems.
High-Speed Transceivers	High-speed transceivers support faster data transmission, reducing the overall communication time between sensors and network gateways.

Wireless Communication Protocols for Low Latency Applications

It is important to be able to select the right wireless communication protocol in order for an embedded system to achieve ultra low latency. Wi-Fi and the likes are used extensively in the traditional sense, but they might not be suitable for those application which require millisecond-level responsiveness. Specifically, specialized protocols are often used which are specifically designed to support low latency, deterministic communication.

Time-Sensitive Networking (TSN) is one protocol that does this: It extends Ethernet standards for enabling deterministic, low latency communication over wired and wireless networks. Features like time synchronization, traffic scheduling and frame preemption are incorporated by TSN to ensure timely delivery of critical data with strict bound.

One other very promising approach uses Industrial Internet of Things (IIoT) protocols such as OPC UA (Open Platform Communications Unified Architecture) over TSN. This combination provides the opportunity for standardized vendor independent communication with real time capabilities designed, for a broad application of industrial control. In order to achieve ultra-low latency wireless communication, the protocols based on the IEEE802.15.4e standard (e.g. WirelessHART, ISA100.11a), provide deterministic performance through time-slotted channel hopping in combination with mesh networking. Such protocols are intended to work reliably in harsh industrial environments and achieve cycle times of the order of milliseconds.[5-9]

Minimizing Latency Network Topology and Routing Strategies

Latency Minimizing Network Topology and Routing Strategies

Minimizing end to end latency in wireless sensor networks heavily relies on the design of network topology and routing strategies. Star topologies offer the simplest, and possibly simplest way of performing single hop communication, but are not immediately suitable for large scale deployments, or for environments with a great deal of obstacles. However, in terms of robustness and flexibility, mesh networks offer multi hop communication. Nevertheless, the very routing algorithm must be cleverly designed so that packets make a nontrivial route through the network. Techniques like geographic routing that take advantage of information showing node location to make forwarding decisions can reduce the number of hops and total latency.

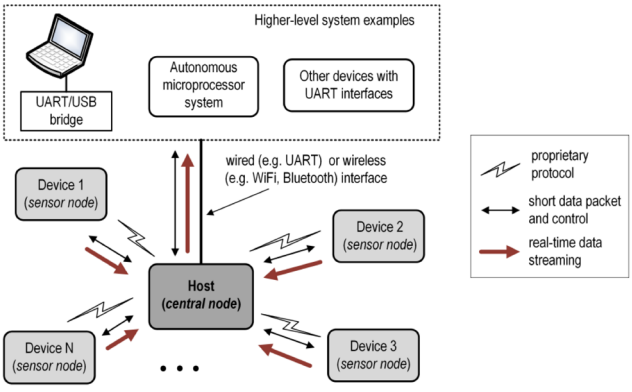

Fig. 1: Minimizing Latency Network Topology and Routing Strategies

Sometimes hybrid topologies, which include a mix of star and mesh networks, may provide the best of both worlds, offering low latency and high network resilience together. For example, a cluster tree topology can provide low latency communication between local clusters with scalable multi hop communication between clusters. There are advanced techniques of routing to increase reliability and lower latency, as data is sent over multiple paths at the same time. This approach can be used to minimize adverse consequences of link failures and congestion, so critical data gets its destination as timely as possible.

Reliability and Error Correction Mechanisms

In such environments where wireless confers the risk of interference and packet loss, constructing application effective error correction and reliability mechanisms over low-latency communication in effect is a necessity. Although reliable automatic repeat request (ARQ) schemes are available, they are often very slow with intervening retransmissions. In most cases more sophisticated approaches are required for ultra low latency applications.

FEC techniques can reduce the need to retransmit by providing the receiver to correct errors without needing add additional data. But FEC is expensive and there is an overhead of FEC that must be weighed against possible latency savings. Good compromise between reliability and efficiency can be obtained by adaptive FEC schemes that adjust the level of error correction according to channel conditions. In hybrid ARQ (HARQ) protocols, these different techniques combine the recoveries of ARQ and FEC by allowing the receiver to store and combine data from several transmissions. The approach discussed here substantially decreases the number of retransmissions, ultimately improving the overall latency in challenging wireless environments.

If the applications can tolerate some data loss (which in some cases they can) techniques like Unequal Error Protection (UEP) can be used. However, UEP focusses on protecting critical data based on which, there can be transmission of less important information at reduced or no error correction in exchange of considerable latency, sometimes sacrificing data.

Distributed Embedded Systems: Time Synchronization

Accurate time synchronization is an essential part of many low latency wireless sensor networks, in distributed control systems where multiple nodes need to coordinate the timing of their actions. Syncing in tight limits across a network of embedded devices is difficult because of clock drift, propagation delays, and resource constraints. Time synchronization in networks already have some protocols, e.g. the Precision Time Protocol (PTP) and the Network Time Protocol (NTP), which are the base for it. Yet, these protocols may require adaptation or optimization for deployment in constrained resource embedded devices and wireless environment.

Hardware assisted timestamping is one possible way to increase synchronization accuracy. Flattening the stack by timestamping packets at the MAC or PHY layer minimizes the effect of the software stack delays and opens the door for more precise synchronization. In certain scenarios the common time reference is provided from an external time source, for example a GPS receiver. This approach provides high accuracy, but the cost, power consumption or environmental constraints of such an approach may not be appropriate for all applications.

Low Latency Devices Power Management Strategies

The need to simultaneously balance ultra low latency communication with high power efficiency in embedded system design poses a major challenge to the embedded system design of battery powered or energy harvesting devices. For devices that need to last, effective power management strategies are needed to make them responsive. One common method of saving energy is duty cycling in which active and sleep states are alternated between a number of devices. But typical duty cycling approaches introduce a lot of latency since devices have to awake before they can communicate. This issue is overcome by advanced wake-up radio technologies utilizing ultra low power receivers to quickly turn on the main radio when communications are required. At times of low activity, dynamic voltage and frequency scaling (DVFS) techniques allow active processors to reduce their operating parameters in order to conserve power and save energy. However, the processor must be able to scale quickly to full performance when low-latency operation is necessary.

In some cases, such heterogeneous multicore architectures may be used, that is, the use of high performance cores for time critical tasks and energy efficient cores for background processing. It provides

Table 2: Techniques for Optimizing Latency in Embedded Systems

Technique	Goal
Task Prioritization	Task prioritization ensures that time-sensitive tasks are given higher precedence, reducing delays in critical data processing.
Edge Computing	Edge computing brings data processing closer to the sensor nodes, reducing the need for data transmission to remote servers and minimizing latency.
Time Division Multiplexing	Time division multiplexing splits available bandwidth into time slots, allowing multiple devices to share the channel without causing interference or delays.
Sleep Mode Management	Sleep mode management ensures that sensor nodes remain in low power states during idle periods, reducing power consumption without affecting communication latency.
Real-Time Data Processing	Real-time data processing allows for instant analysis and action based on the incoming sensor data, ensuring timely responses and decisions in dynamic environments.
Caching Techniques	Caching techniques store frequently accessed data locally, reducing the need for repeated network requests and decreasing latency.

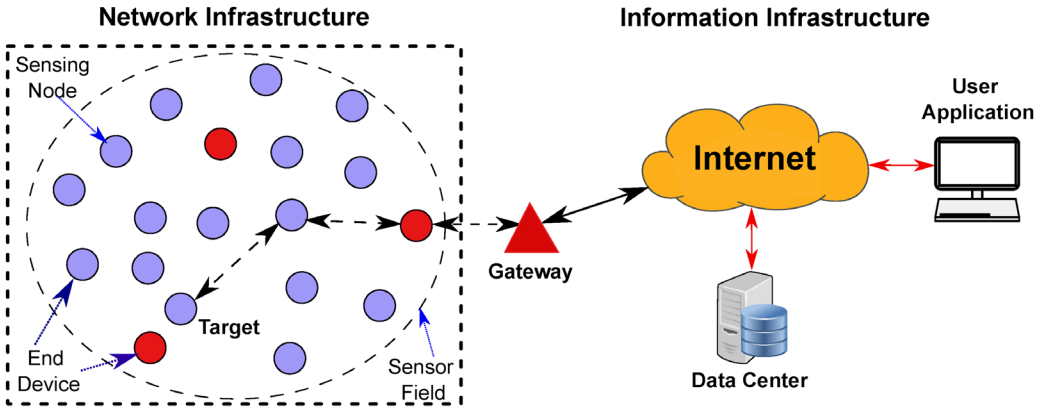

Fig. 2: Low Latency Devices Power Management Strategies

a way for the system to trade performance for power consumption depending on which application demands it (Figure 2).[10-14]

Low Latency Wireless Communication: Security Considerations

Due to additional processing overhead and delays from traditional security mechanisms, the security of low latency wireless communication is very challenging. Yet the criticality of many real time control applications dictate that mechanisms for securing these applications against unauthorized access and data tampering must be robust. Security measures can be made without negatively impacting latency on lightly encrypted devices through the use of specially designed lightweight encryption algorithms. These algorithms sacrifice some security strength in the interest of better performance, and are thus well suited for application with threat models that accept such compromises.

In addition, physical layer security techniques such as frequency hopping and spread spectrum modulation can provide a last line of defense against eavesdropping and jamming attack while incurring only minor latency. Security of the mobile system is improved via these approaches, which exploit the properties of the wireless channel. Hardware acclerated crypto modules can offload cryptographic operation from main the processor to help reduce overall system latency for environments that require greater security from cryptographic operation. Modern microcontrollers and SoCs have built in dedicated security coprocessors to carry out this role.[15-21]

Validation and Testing of Low Latency Embedded Systems

As a result, the development process for ultra low latency embedded systems for wireless sensor networks requires first thorough testing and validation. Because of the stringent timing requirements and chance of a system being safety critical in many applications, it is

imperative that the system reliably meets its latency and reliability guarantees within a large performance space. Hardware in the loop simulation is a great way to test embedded systems in real world like scenarios without the need of a full physical setup. HIL can run simulation of environmental conditions of the system and network topologies to reduce surprises as well as potential issues early in the system development.

For measuring the latency of individual system components and end-to end communication paths, specialized test equipment, such as high precision oscilloscopes and logic analyzers are required. Identification of highway bottlenecks and verification that timing requirements will be met at each stage of the communication process is an invaluable application of these tools. Channel emulators can be used to emulate different RF conditions as well as interference scenarios for wireless systems. With this, developers can simulate how the system behaves under stress conditions that may be too hard or impossible to re-create in the 'real world'.

Conclusion

Wireless sensor network is a frontier of real time control and monitoring, and designing embedded systems for ultra low latency communication in it is a frontier of real time system design. By selecting hardware, running software, designing a protocol and designing a system architecture, engineers can produce robust, efficient systems that can meet the most exacting latency requirements. Looking toward the future, emerging technologies like 5G and beyond, edge computing, and advanced AI driven optimization of low latency wireless communication will potentially bring us closer to the limits of what's currently possible. With these advancements we can now apply these across new use cases, such as autonomous vehicles, industrial automation, and immersive augmented reality. The continued evolution of embedded system design that aims to achieve ever lower latencies while maintaining security, reliability, and energy efficiency is a continuing challenge. From the physical layer to the application level, success in this domain requires a holistic approach to factor in every aspect of the system in this article, we've looked at how successful deployment of systems managing blocks that include elements of both robotics and computing does. With challenges that we embrace and technologies we pioneer, we can build wireless sensor networks that can react to our world with unimaginable speed and precision, unbounded in how we will ultimately experience and manage our environment.

References:

1. Tan, L., Yu, K., Bashir, A. K., Cheng, X., Ming, F., Zhao, L., & Zhou, X. (2023). Toward real-time and efficient cardiovascular monitoring for COVID-19 patients by 5G-enabled wearable medical devices: a deep learning approach. Neural Computing and Applications, 1-14.
2. Taylor, C. R., Venkatasubramanian, K., & Shue, C. A. (2014, April). Understanding the security of interoperable medical devices using attack graphs. In Proceedings of the 3rd international conference on High confidence networked systems (pp. 31-40).
3. Durrani, Y. A., Abril, A., & Riesgo, T. (2007, May). Efficient power macromodeling technique for IP-based digital system. In 2007 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1145-1148). IEEE.
4. El-dib, D., El-dib, D. A., Alkabani, Y., El-moursy, M., & Shawkey, H. H. (2016). Automated FPGA Power Characterization Methodology. International Journal of Electrical and Computer Sciences IJECS-IJENS, 16(2).
5. Richa, M., Prévotet, J. C., Dardaillon, M., Mroué, M., & Samhat, A. E. (2023). High-level power estimation techniques in embedded systems hardware: an overview. The Journal of Supercomputing, 79(4), 3771-3790.
6. Rajeev Ratna Vallabhuni, P Sampath, Pedakolmi Venkateswarlu, S. Sree Hari Raju, M. Suresh Kumar, Chandan kumar Sonkar, Vasamsetti Srinivas, V. Ramya, Dhinakaran A, Bishnu Kant Shukla, Mimi Mariam Cherian, J Logeshwaran, "BLOCK CHAIN BASED CENTRALIZED CLOUD APPLICATION IN SMART CITIES," The Patent Office Journal No. 05/2023, India. Application No. No. 202341003902 A.
7. Fang, J., Liu, S., & Zhang, X. (2017, October). Research on cache partitioning and adaptive replacement policy for CPU-GPU heterogeneous processors. In 2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES) (pp. 19-22). IEEE.
8. Fang, J., & Pu, J. (2010, May). Dynamic fair cache partitioning for chip multiprocessor. In 2010 Third International Joint Conference on Computational Science and Optimization (Vol. 2, pp. 283-287). IEEE.
9. Fitzgerald, B., López, S., & Sahuquillo, J. (2013, June). Drowsy cache partitioning for reduced static and dynamic energy in the cache hierarchy. In 2013 International Green Computing Conference Proceedings (pp. 1-6). IEEE.
10. Yourdon, E. (1989). Modern structured analysis. Yourdon press.
11. Wagner, F., & Wolstenholme, P. (2003, April). Modeling and building reliable, re-useable software. In 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2003. Proceedings. (pp. 277-286). IEEE.

12. G. Sirisha, Rajeev Ratna Vallabhuni, Valli Madhavi Koti, Nitin S Goje, Debasis Das, Prakhar Yadav, K. Vetrivel Kumar, Devasis Pradhan, V. Kannan, J Logeshwaran, "AI/ML SUPPORT FOR ULTRA-LOW LATENCY APPLICATIONS AT THE EDGE OF THE NETWORK," The Patent Office Journal No. 02/2023, India. Application No. No. 202341001497 A.
13. Wehrmeister, M. A., Becker, L. B., Wagner, F. R., & Pereira, C. E. (2005, May). An object-oriented platform-based design process for embedded real-time systems. In Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05) (pp. 125-128). IEEE.
14. Lindberg, T., Meinel, C., & Wagner, R. (2010). Design thinking: A fruitful concept for IT development?. In Design thinking: Understand–improve–apply (pp. 3-18). Berlin, Heidelberg: Springer Berlin Heidelberg.
15. Shao, Y. S., Xi, S. L., Srinivasan, V., Wei, G. Y., & Brooks, D. (2016, October). Co-designing accelerators and SoC interfaces using gem5-Aladdin. In 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO) (pp. 1-12). IEEE.
16. Xiao, Y., Nazarian, S., & Bogdan, P. (2019). Self-optimizing and self-programming computing systems: A combined compiler, complex networks, and machine learning approach. IEEE transactions on very large scale integration (VLSI) systems, 27(6), 1416-1427.
17. P. Balaji, Muhammadu Sathik Raja M., K. Kalamani, Rajeev Ratna Vallabhuni, Jobin Varghese, R. Suganthi, Srinivasa Narayana, K. Ramalashmi, Diana Earshia V, Thoram Saran Kumar, J Logeshwaran, V. Kannan, "ANALYSIS AND DETECTION OF DEPRESSION SEVERITY SCORES BASED ON EEG SIGNAL USING MACHINE LEARNING APPROACH," The Patent Office Journal No. 02/2023, India. Application No. No. 202341001221 A.
18. Bolchini, C., Cherubin, S., Durelli, G. C., Libutti, S., Miele, A., & Santambrogio, M. D. (2018). A runtime controller for OpenCL applications on heterogeneous system architectures. ACM SIGBED Review, 15(1), 29-35.
19. Sadagopan, V. K., Rajendran, U., & Francis, A. J. (2011, July). Anti theft control system design using embedded system. In Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety (pp. 1-5). IEEE.
20. Salewski, F., & Kowalewski, S. (2008). Hardware/software design considerations for automotive embedded systems. IEEE Transactions on Industrial Informatics, 4(3), 156-163.
21. Simonot-Lion, F. (2009). Guest editorial special section on in-vehicle embedded systems. IEEE Transactions on Industrial Informatics, 5(4), 372-374.
22. Balvad, K., Lalit, D., Re-Ann, C., Halimpusan, T., & Beyes, J. O. (2022). Unveiling 4G LTE: Revolutionizing connectivity in the digital age. National Journal of Antennas and Propagation, 4(1), 41–46.
23. Muyanja, A., Nabende, P., Okunzi, J., & Kagarura, M. (2023). Flip-Flop Realization: Conventional Memory Elements Design with Transistor Nodes. Journal of VLSI Circuits and Systems, 5(1), 20–27. https://doi.org/10.31838/jvcs/05.01.03
24. R.T., Yamine, & N.S., Nithya. (2015). Survey on high utility itemset mining. International Journal of Communication and Computer Technologies, 3(2), 63-66. https://doi.org/10.31838/IJCCTS/03.02.03
25. Veerappan, S. (2023). Designing voltage-controlled oscillators for optimal frequency synthesis. National Journal of RF Engineering and Wireless Communication, 1(1), 49-56. https://doi.org/10.31838/RFMW/01.01.06
26. Arvinth, N. (2024). Integration of neuromorphic computing in embedded systems: Opportunities and challenges. Journal of Integrated VLSI, Embedded and Computing Technologies, 1(1), 26-30. https://doi.org/10.31838/JIVCT/01.01.06
27. Surendar, A. (2024). Survey and future directions on fault tolerance mechanisms in reconfigurable computing. SCCTS Transactions on Reconfigurable Computing, 1(1), 26-30. https://doi.org/10.31838/RCC/01.01.06
28. Borhan, M. N. (2025). Exploring smart technologies towards applications across industries. Innovative Reviews in Engineering and Science, 2(2), 9-16. https://doi.org/10.31838/INES/02.02.02
29. Uvarajan, K. P. (2024). Integration of blockchain technology with wireless sensor networks for enhanced IoT security. Journal of Wireless Sensor Networks and IoT, 1(1), 23-30. https://doi.org/10.31838/WSNIOT/01.01.04
30. Uvarajan, K. P. (2024). Integration of artificial intelligence in electronics: Enhancing smart devices and systems. Progress in Electronics and Communication Engineering, 1(1), 7–12. https://doi.org/10.31838/PECE/01.01.02