JOURNAL OF CRITICAL REVIEWS

ISSN-2394-5125 VOL 7, ISSUE 13, 2020

USE OF REAL-TIME OPERATING SYSTEMS IN EMBEDDED SYSTEMS IN THE CONTEXT OF AI AGENTS

Mr. Junaid Hussain Wani[1], Dr. Kaisar Hussain Shah[2]

[1,2]Computer Science, Anantnag Jammu & Kashmir
[1]*Email: wanijunaid94@gmail.com
[2]Email: kaisarhussainshah@gmail.com

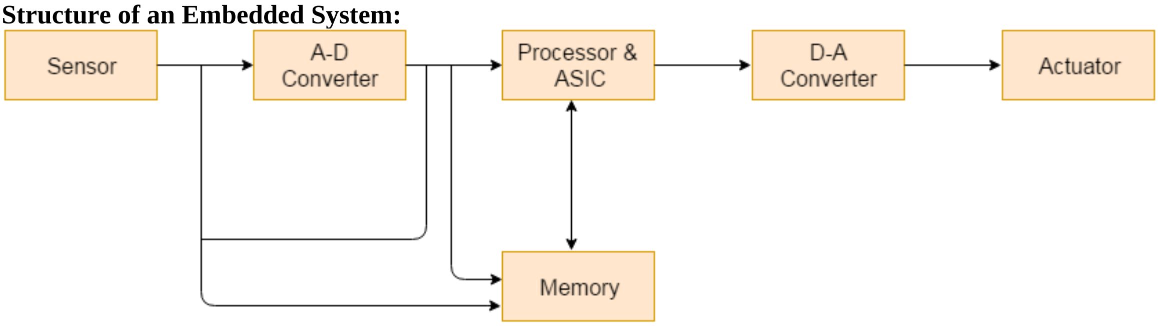
***Corresponding Author:** Mr. Junaid Hussain Wani
*Email: wanijunaid94@gmail.com

Abstract
This research paper explores the use of real-time operating systems (RTOS) in embedded systems, specifically in the context of AI agents. Embedded systems are pervasive in modern technology, and the integration of AI agents into these systems has become increasingly common. Real-time operating systems play a crucial role in ensuring the efficient and reliable operation of embedded systems, especially when they are tasked with running AI algorithms. This paper provides an overview of embedded systems, real-time operating systems, and AI agents, and then delves into the specific considerations and benefits of using RTOS in the context of AI agents within embedded systems.

1. Introduction

Embedded systems are specialized computing systems designed to perform dedicated functions within a larger mechanical or electrical system. These systems are found in a wide range of applications, including consumer electronics, automotive systems, industrial automation, medical devices, and more. With the advancement of artificial intelligence (AI) technologies, there is a growing trend towards integrating AI agents into embedded systems to enable intelligent decision-making and autonomous operation.

Basic Structure of an Embedded System:



Sensor: Sensor used for sensing the change in environment condition and it generate the electric signal on the basis of change in environment condition. Therefore it is also called as transducers for providing electric input signal on the basis of change in environment condition.

A-D Converter: An analog-to-digital converter is a device that converts analog electric input signal into its equivalent digital signal for further processing in an embedded system.

Processor & ASICs: Processor used for processing the signal and data to execute desired set of instructions with high-speed of operation.

Application specific integrated circuit (ASIC) is an integrated circuit designed to perform task specific operation inside an embedded system.

D-A Converter: A digital-to-analog converter is a device that converts digital electric input signal into its equivalent analog signal for further processing in an embedded system.

Actuators: Actuators is a comparator used for comparing the analog input signal level to desired output signal level for providing the error free output from the system.

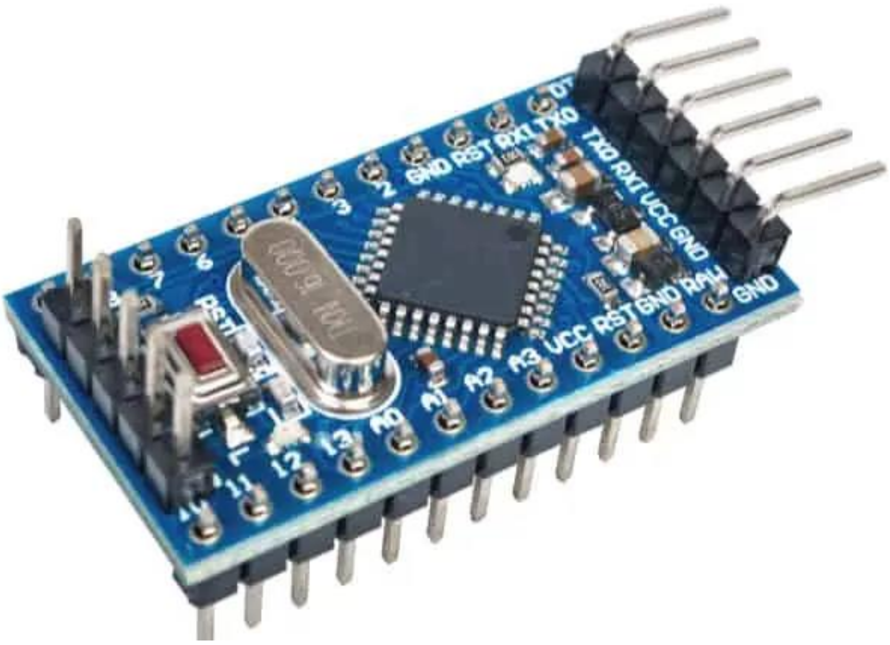
ISSN-2394-5125 VOL 7, ISSUE 13, 2020

What Is Embedded System Design (ESD)?

An embedded system is a self-contained, microprocessor-based computer system typically implemented as a component of a larger electrical or mechanical system. At the core of the embedded system is an integrated circuit that performs computational tasks. Also included are hardware and software, both of which are designed to execute a specific, dedicated function. Embedded systems can be highly complex or relatively simple, depending on the task for which they were designed. They may include a single microcontroller or a suite of processors with linked peripherals and networks. They may have no interface or highly intricate GUIs. Embedded systems programming instructions are stored in read-only memory or flash memory chips.

Why Is Embedded System Design Important?

Today, there are billions of embedded system devices used across many industries including medical and industrial equipment, transportation systems, and military equipment. Many consumer devices from digital watches to kitchen appliances and automobiles also feature them. Embedded systems are small, fast, powerful, and designed for very specific use cases. While general purpose systems can perform multiple functions, they can be too costly for many applications, and may also fail to measure up to embedded systems' reliability, low power consumption, minimal size, and other functional and performance features.



Embedded systems design is the process where hardware and firmware designers come together to build embedded systems from scratch. This involves PCB design, where the necessary components are connected to build functional circuits. To bring the electronics to life, the firmware is coded and then programmed into the microcontroller.
the most crucial and challenging part of embedded systems design is PCB design ,It's obvious that every stage of embedded systems design is important. However, if there's one area that I found particularly crucial to a successful outcome, it is the PCB design. You can fix bugs and other code issues with a firmware revision, but when PCBs are manufactured, any mistakes will result in substantially costly remedies and reworks. Therefore, you must carefully design and test when designing the PCB for an embedded system.

2. Real-Time Operating Systems (RTOS)

A real-time operating system is designed to manage tasks with specific timing constraints. Unlike general-purpose operating systems, RTOS is optimized for deterministic behavior and timely response to events. In the context of embedded systems, where precise timing and control are often critical, RTOS plays a vital role in ensuring that tasks are executed within specified time constraints.
RTOS is therefore an operating system that supports real-time applications by providing logically correct result within the deadline required.

ISSN-2394-5125 VOL 7, ISSUE 13, 2020

RTOS CLASSFICATION

RTOS specifies a known maximum time for each of the operations that it performs. Based upon the degree of tolerance in meeting deadlines, RTOS are classified into following categories

·Hard real-time: Degree of tolerance for missed deadlines is negligible. A missed deadline can result in catastrophic failure of the system

·Firm real-time: Missing a deadly ne might result in an unacceptable quality reduction but may not lead to failure of the complete system

·Soft real-time: Deadlines may be missed occasionally, but system doesn't fail and also, system quality is acceptable.

RTOS must exhibit deterministic behavior, meaning that the time taken to complete an operation or respond to an event should be predictable and consistent.

RTOS plays a critical role in managing real-time constraints and ensuring reliable performance in embedded systems through task scheduling, interrupt handling, resource management, deterministic behavior, preemption support, timing services, low overhead, and configurability.

3. Embedded Systems in AI Agents

AI agents refer to software entities that can perceive their environment and take actions to achieve specific goals. When integrated into embedded systems, AI agents can enable capabilities such as machine learning, natural language processing, computer vision, and autonomous decision-making. The deployment of AI agents in embedded systems introduces new challenges related to computational resources, real-time responsiveness, and system reliability.

Embedded AI systems typically involve a combination of hardware and software components. The hardware, often a microcontroller or a specialized AI chip, is responsible for executing the AI algorithms. The software component includes the AI models and algorithms, which are typically trained on a more powerful system and then deployed to the embedded system.

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. In the context of artificial intelligence (AI), embedded systems play a crucial role in enabling AI applications to interact with the physical world, process data in real time, and make autonomous decisions. These systems are designed to perform specific tasks and are integrated into AI devices, such as autonomous vehicles, industrial robots, smart appliances, and IoT (Internet of Things) devices.

Applications of Embedded Systems in AI:

Autonomous Vehicles: Embedded systems play a central role in enabling the perception, decision-making, and control functions of autonomous vehicles through sensor integration, real-time processing, and actuator control.

Industrial Robotics: In manufacturing environments, embedded systems empower robotic arms and automation equipment with AI capabilities for tasks such as object recognition, manipulation, and quality control.

Smart Home Devices: Embedded systems integrated with AI enable smart home devices to understand user preferences, automate tasks based on environmental conditions, and provide personalized experiences.

Healthcare Devices: Medical devices benefit from embedded systems with AI capabilities for patient monitoring, diagnostic assistance, treatment optimization, and personalized care delivery.

ISSN-2394-5125 VOL 7, ISSUE 13, 2020

4. Integration of RTOS with AI Agents in Embedded Systems:

The integration of RTOS with AI agents in embedded systems presents several advantages. Firstly, RTOS provides deterministic scheduling and resource management, which is essential for meeting the timing requirements of AI algorithms running on embedded hardware. Additionally, RTOS facilitates efficient multitasking and prioritization of tasks, allowing for seamless integration of AI agent functionalities alongside other system-critical operations. It allows devices to make intelligent decisions locally, reducing the need for constant communication with a central server. This can improve performance, reduce latency, and enhance privacy. It also enables devices to adapt to their environment and learn from their experiences, improving their functionality over time.

When integrating AI agents into embedded systems, the compatibility and interaction between RTOS and AI components become essential to ensure efficient and reliable performance. This integration involves addressing challenges related to real-time constraints, resource management, and system responsiveness. Understanding the integration of RTOS with AI agents in embedded systems requires an exploration of the key considerations, benefits, challenges, and best practices associated with this convergence.

Integrating RTOS with AI agents presents several challenges. One of the key challenges is achieving a balance between the real-time requirements of the system and the computational demands of AI algorithms. AI agents often require significant computational resources, which can impact the real-time responsiveness of the system.

Another challenge lies in optimizing resource allocation within the embedded system. RTOS must efficiently manage resources to ensure that both the real-time tasks and AI algorithms have access to the necessary resources without causing conflicts or delays.

Why to use Real Time Operating Systems In Embedded Systems:

Real-time operating systems (RTOS) are essential in AI embedded systems for several reasons. Firstly, RTOS provides deterministic behavior, ensuring that critical tasks are executed within a specific time frame. This is crucial in AI applications where real-time processing and response are required, such as autonomous vehicles, industrial automation, and robotics. Secondly, RTOS offers efficient task scheduling and management, allowing for prioritization of tasks based on their importance and deadlines. In AI embedded systems, this capability is vital for handling multiple concurrent tasks, including sensor data processing, decision-making algorithms, and control loops. Thirdly, RTOS provides a high level of reliability and fault tolerance, which is essential in safety-critical AI applications. The ability to recover from system failures or unexpected events is crucial in maintaining the integrity and performance of AI embedded systems. Additionally, RTOS often comes with built-in support for real-time communication protocols, which is beneficial for AI applications that require seamless interaction with external devices or networks. Moreover, RTOS can offer low latency and minimal interrupt latency, enabling AI embedded systems to respond swiftly to input stimuli and produce timely outputs.

Key Considerations for Integration

The integration of RTOS with AI agents in embedded systems necessitates careful consideration of several key factors. Firstly, the real-time nature of embedded systems demands that the RTOS provides deterministic behavior and precise timing control to support AI algorithms' execution within specified deadlines. Additionally, the resource constraints typical of embedded platforms require efficient utilization of processing power, memory, and other system resources to accommodate both the RTOS and AI functionalities. Furthermore, seamless communication and synchronization between the RTOS and AI agents are critical for ensuring coherent operation without compromising real-time performance.

Benefits and Challenges

The utilization of RTOS in conjunction with AI agents offers numerous benefits such as improved system responsiveness, predictable task execution times, and enhanced reliability. However, challenges may arise in terms of complexity in system design and potential overhead associated with managing real-time constraints alongside AI computations.

Real-time operating systems (RTOS) and artificial intelligence (AI) are two advanced technologies that have significantly impacted various industries. When combined, they can offer a wide range of benefits that can improve efficiency, flexibility, and performance.

Enhanced Efficiency

One of the key benefits of combining RTOS with AI is improved efficiency. AI algorithms can be integrated into the RTOS to automate various tasks, allowing for faster decision-making and execution. This can lead to reduced processing times and improved resource allocation, resulting in a more efficient system.

Increased Flexibility

RTOS and AI together offer increased flexibility in system design and application development. AI algorithms can adapt to changing conditions and requirements, providing a more responsive and dynamic system.

Improved Performance

The combination of RTOS and AI can lead to better overall system performance. AI algorithms can be used to optimize the allocation of system resources, prioritize tasks, and identify potential bottlenecks. This can help to ensure that the system runs efficiently and effectively, even under heavy workloads.

Case Studies and Applications

This section provides case studies and examples of real-world applications where RTOS is effectively used in conjunction with AI agents within embedded systems. Examples may include autonomous vehicles, smart appliances, industrial automation platforms, and IoT devices leveraging AI capabilities through RTOS-based embedded solutions.

Future Directions

The future directions for the use of RTOS with AI agents in embedded systems encompass enhanced real-time performance, integration of edge computing capabilities, security and safety considerations, resource management and optimization, interoperability with AI frameworks, and energy-efficient operation. These advancements will shape the next generation of intelligent embedded systems capable of supporting sophisticated AI applications while meeting stringent real-time requirements.

Conclusion

In conclusion, the integration of real-time operating systems with AI agents within embedded systems presents a compelling opportunity to enable intelligent functionalities while ensuring deterministic behavior and timely responses. By carefully considering the specific requirements and challenges associated with deploying AI agents on embedded hardware, developers can leverage the benefits of RTOS to create robust and efficient intelligent embedded systems.

Reference

1. IBM Research Blog. (2018). Embedded AI: The next wave of artificial intelligence.
2. https://ieeexplore.ieee.org/Xplore/home.jsp
3. https://www.engineersgarage.com/rtos-real-time-operating-system/
4. https://dl.acm.org/
5. https://link.springer.com/