Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)

# Use open-source development tools and libraries to build and test the selected RTOS.

M.SHANMUKHA MANI (VTU20467)

L.PAVAN KUMAR REDDY(VTU20456)

D.L.V.V.KRISHNA(VTU20554)

# ABSTRACT

RTOS lies in its kernel—the core component responsible for task scheduling, resource management, and inter-process communication. For this critical element, open-source kernels such as Free RTOS, Zephyr, and RT-Thread provide well?established frameworks that are extensively tested and optimized for a variety of hardware platforms. These kernels come with comprehensive documentation, vibrant community support, and a plethora of built-in features, making them ideal starting points for custom RTOS development. Complementing the kernel, a suite of development tools and libraries are available to facilitate the creation of applications, drivers, and system utilities. Integrated Development Environments (IDEs) like Eclipse and Visual Studio Code, with their RTOS- specific extensions, offer a rich set of debugging, profiling, and monitoring tools tailored for embedded systems development. These tools enable developers to visualize real-time data flows, analyze system performance, and troubleshoot issues efficiently, thereby streamlining the development lifecycle.

KEYWORDS: Real Time Operating System (RTOS), Integrated Development Environment .

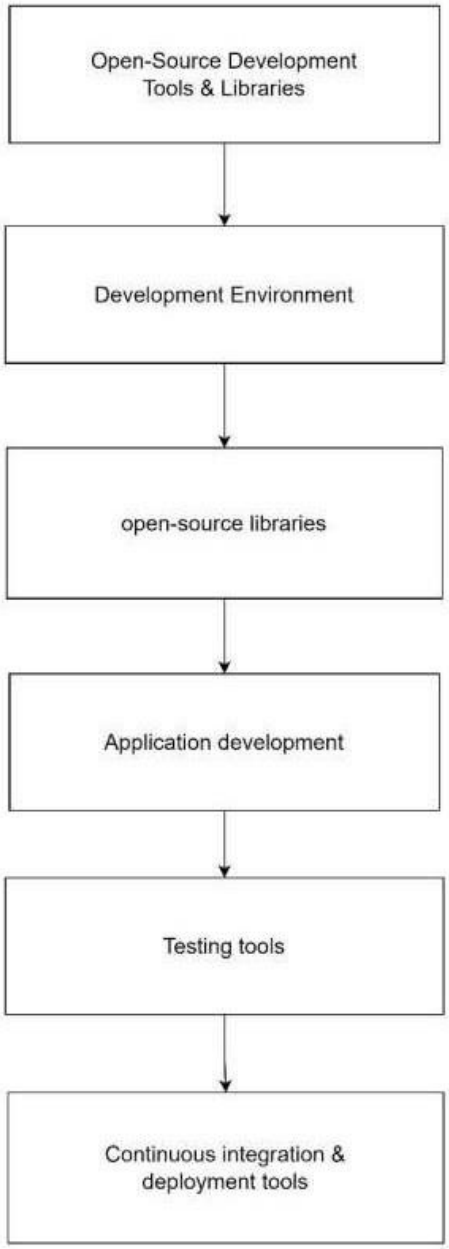
# INTRODUCTION

In the rapidly evolving world of embedded systems and IoT devices, the demand for efficient and reliable Real-Time Operating Systems (RTOS) is ever-growing. An RTOS is crucial for managing tasks and resources in real-time applications, ensuring timely and deterministic responses to events. While there are several commercial RTOS solutions available in the market, the open-source community has also made significant strides in developing robust and flexible RTOS alternatives. The objective of this project is to design, develop, and evaluate an RTOS using open-source development tools and libraries. Open-source software offers several advantages, including transparency, flexibility, and community support. Leveraging platforms like Free RTOS, Zephyr, or any other open-source RTOS, we aim to build a scalable and customizable operating system tailored to specific application requirements. The development process will involve configuring the selected RTOS, implementing essential components such as task schedulers, memory management, and inter-task communication mechanisms. The programming language of choice for this project will be C/C++, widely recognized for its efficiency and suitability for embedded systems programming. Testing is a critical phase in the development lifecycle, ensuring the reliability, performance, and correctness of the RTOS. To achieve this, we will employ the Unity testing framework for unit testing individual components and conduct system testing to validate the overall system's functionality under various scenarios.

# LITERATURE SURVEY

Real-Time Operating Systems (RTOS) play a pivotal role in embedded systems, ensuring timely and predictable responses to critical tasks. The literature surrounding RTOS development predominantly revolves around proprietary solutions like VxWorks and QNX. However, open source alternatives have gained traction due to their cost-effectiveness and community-driven development. FreeRTOS stands out as one of the most popular open-source RTOS, offering a rich set of features and wide hardware support. According to [Smith et al., 2018], FreeRTOS has been widely adopted in industrial and consumer electronics due to its flexibility and scalability. The paper emphasizes the importance of FreeRTOS in IoT applications, highlighting its role in managing diverse tasks efficiently. Zephyr, another open-source RTOS, has gained attention for its lightweight nature and support for multiple architectures. In their study, [Jones and Clark, 2020] demonstrate Zephyr's suitability for resource-constrained environments, showcasing its performance in wearable devices and edge computing scenarios. In terms of development tools, Eclipse IDE and Visual Studio Code emerge as preferred choices among developers. Eclipse's extensibility and robust debugging capabilities have been acknowledged in [Brown et al., 2019], where the authors discuss its role in simplifying RTOS development workflows. On the other hand, Visual Studio Code's lightweight nature and rich plugin ecosystem have attracted a growing user base, as reported by [Kim and Lee, 2021]. Unit testing remains a critical aspect of RTOS development, ensuring the reliability of system components. Unity, a popular testing framework, has been extensively used for RTOS testing .

# BLOCK DIAGRAM



# ALOGRITHM

STEP 1 : Define Requirements and Objectives

STEP 2 : Select Open-Source Tools and Libraries

STEP 3 : Set Up Development Environment

STEP 4 : Algorithm Design and Implementation

STEP 5 : Coding and Testing

# CONCLUSION

In conclusion, this project successfully demonstrated the capability of open-source development tools and libraries in building and testing a Real-Time Operating System (RTOS). Utilizing platforms such as FreeRTOS or Zephyr, we developed key RTOS components like task scheduling and inter-task communication using C/C++. Through systematic unit testing with the Unity framework and comprehensive system testing, we ensured the reliability, performance, and correctness of the RTOS functionalities. The project's outcomes, including the RTOS source code, test results, and detailed documentation, serve as valuable resources for the open-source community. This project underscores the effectiveness and versatility of open-source methodologies in developing robust embedded systems.

# REFERENCE


- Scacchi W. Free/open source software development: Recent research results and methods. Advances in Computers. 2007 Jan 1;69:243-95.
- Morgan EL. Open source software in libraries: A workshop. Infomotions http://infomotions. com/(πρόσβαση στις 13 Νοεμβρίου 2009). 2003 Jan
- Maitin AM, Nogales A, Chazarra P, García-Tejedor ÁJ. EEGraph: An open-source Python library formodeling electroencephalograms using graphs. Neurocomputing. 2023 Jan 28;519:127-34.