**Overview of Real-time Operating Systems for embedded devices**

**VeselinStanchev[[1]](#footnote-2)**

**1Technical University of Sofia - Plovdiv Branch**

**Abstract:** Real-time OS are very useful for embedded devices and home automation. This paper describes well known real-time OS such as Zephyr OS, FreeRTOS and GNU/Linux. Their features will be compared. The paper performs the Overview task of PhD thesis.

**Keywords: risc-v assembly, avr assembly, operating system**

1. **Introduction**

Real-time operating systems can be used for embedded devices management or for civil or military aviation – for optical devices and weapon controls management. Real-time operating system manages devices with that has:

→ limited resources

→ limited time to complete task

→ sensors for communication

Target of the paper is to explore and describe features of well-known RTOS such as: FreeRTOS, Zephyr and Armbian GNU/Linux.

**2. Material and Methods**

Programming languages for real-time os development are C or Assembly language. Usage of assembly language gives more control of current task execution. For every RTOS will be analyzed:

→ Short description

→ Supported virtualizators

→ Supported Instruction set architectures

→ Possible assembly languages that can be used

→ Embedded device whitch is supported by the os

1) FreeRTOS Analysis

FreeRTOS is open-source real-time operating system that has portable libraries written in C programming language. Libraries provides support for popular communication protocols

such as: MQTT, TCP/IP, etc. FreeRTOS is available for different instruction set architectures such as:

→ x86

→ ARM

→ ARM-64

→ PIC

This OS can be used for different micro-controller boards like STM32-based,

PIC24-based and SiFive RISC-V – based boards. FreeRTOS can be used for IoT boards

witch supports connection to Amazon Web Services-based remote services.

There are several options for developing FreeRTOS applications:

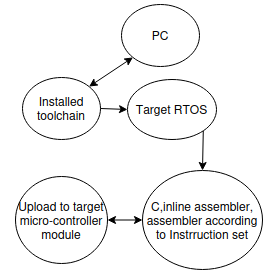
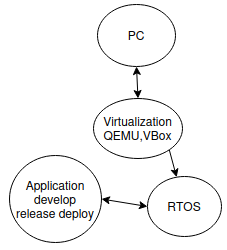
→ using default C-based SDK

→ using inline assembly language in C function

→ using standalone assembly language according to instruction set of target device

Options are shown on ***Figure1***

Applications can be developed on standard x86 computer using virtual environment using QEMU or Virtual-Box – based virtualization. It’s shown on ***Figure2***



***Figure1.*** RTOS development options ***Figure2.*** RTOS application development

**3. Results**

FreeRTOS, Zephyr and Armbian OS key features are analyzed and compared. Virtualizators for x86 simulation are shown. For every OS is shown supported device

**4. Conclusions and future work**

Analyzed operating systems are written in C and they support RISC-V architecture

because of C language abstraction. But when requirements for the RTOS are OS to be much faster and more useful for RISC-V projects then exists need to develop truly RISC-V assembly-

based OS. That is main target of my PhD thesis.

**References**:

1. A practical introduction to real-time systems for

undergraduate engineering

2. FreeRTOS Documentation, 2023

3. Zephyr Documentation,2023

4. Armbian Documentation,2023

Contact Information:

Veselin Stanchev,Eng. MSc.

Scientific field: Computer Architectures

E-mail: vrstanchev@gmail.com

ORCID iD: 0009-0009-2992-5326

1. PhD Theme: Methods and Tools to develop an assembly-based operating system for embedded devices [↑](#footnote-ref-2)