Civil Real-Time operating systems and assembly languages for their programming

Veselin Stanchev [1 a)]

[1]Technical University of Sofia, Plovdiv Branch Tsanko Doustabanov 25,4000
Departament of Information and Communication Technologies
[a)] Corresponding author: vrstanchev@tu-plovdiv.bg

Abstract. Real-Time operating systems are usage in many different domains. They can be used in military industries. There are OS for robotics ROS, for drones, for missile system control and for aircraft control system. Free assembly languages will becomes popular because of open-source instruction set architecture – RISC-V. RiSC-V architecture and assembly language will becomes dominate architecture for developing of RTOS in this field. This paper describes the most known open-source real-time operating systems and assembly languages for them.

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. There are several types of instruction-set architectures. x86, RISC, CISC and RISC-V instruction-set architectures are the most known. There are different free assembly languages for different devices. For x86 based devices Netwide assembler or Yarn assembler can be used. For devices based on RISC based devices there are GNU assembler, AVR assembler. For RISC-V based devices can be used RISC-V assembler.

OPEN-SOURCE REAL-TIME OPERATING SYSTEMS

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 tool-chains
- Supported Instruction set architectures
- Possible assembly languages that can be used
- Embedded device witch is supported by the os

ZEPHYR ANALYSIS

Zephyr is real-time operating system written in C programming language. Zephyr is available for different instruction set architectures such as:

- ARM-64
- x86
- MIPS

This OS can be used for different micro-controller boards like NEORV32,KB-2040. There are several options for developing Zephyr applications:

- using default C-based SDK
- using standalone assembly language according to instruction set of target device. Zephyr can be used through QEMU virtualizator. ARM C Tool-chain is supported. Cmake can be used for automation of build process of the project. [3]**.**

ARMBIAN GNU/LINUX ANALYSIS

Armbian is real-time Debian – ported OS for micro-controller modules; SD card can be used for storage. Armbian is available for different instruction set architectures such as:

- x86_64
- RISC

This OS can be used for different micro-controller boards like OrangePi, BananaPi. [4]**.**

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 **FIGURE 2**.

GNU Make, Cmake can be used for automation of build process of the project. Compiled once the project can be deployed to multiple compatible boards using pipelines based on chosen build tool. Build automatization is shown on **FIGURE 3.**

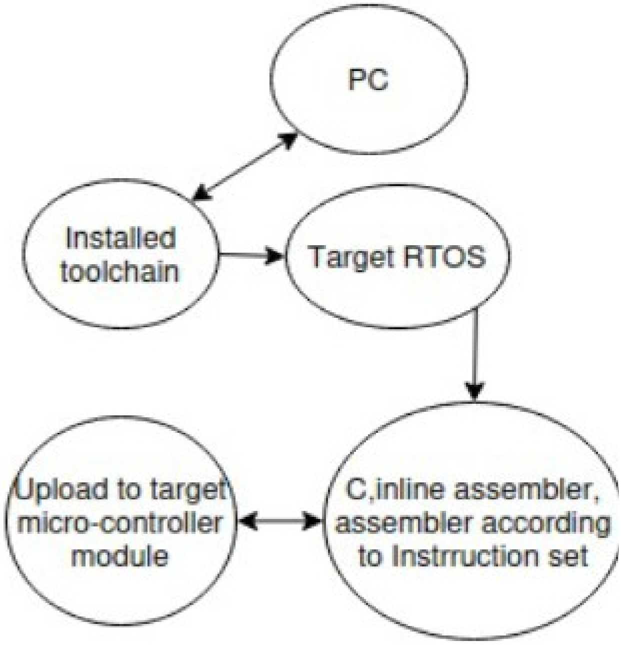
PC
Installed toolchain
Target RTOS
Upload to target micro-controller module
C,inline assembler, assembler according to Instrruction set

FIGURE 1. RTOS development options

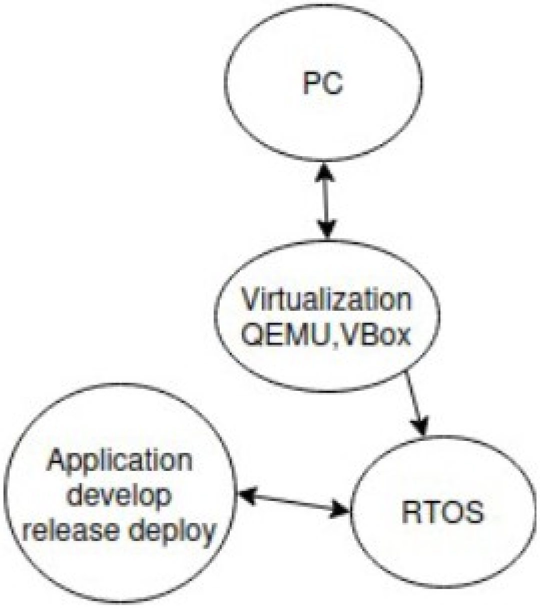

FIGURE 2. RTOS application development

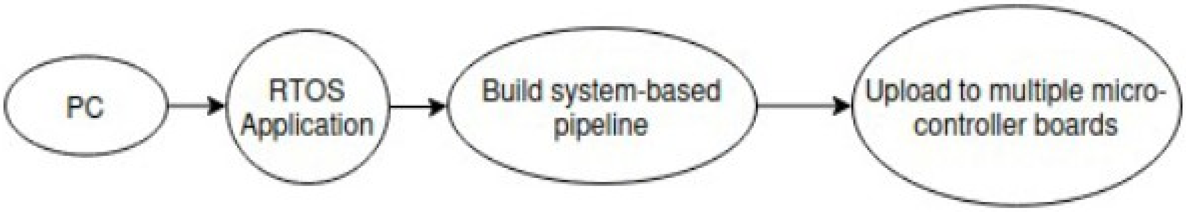
PC
RTOS Application
Build system-based pipeline
Upload to multiple micro-controller boards

FIGURE 3. RTOS application pipeline automatization

FREE ASSEMBLY LANGUAGES FOR PROGRAMMING OF REAL-TIME OPERATING SYSTEMS

There are different free assembly languages for embedded devices such as: AVR Assembler, PIC Assembler,RISC-V Assembler. For each assembler will be analyzed:

- Types of available instructions
- Supported micro-controllers
- Compilation process of target assembler

AVR ASSEMBLER

AVR assembler is free and open-source assembly language. It supports several types of instructions such as:

- arithmetical-logic operations
- control flow branch instructions
- bite control instructions
- byte control instructions

This assembler is used for range of RISC-based device families such as:

- Raspberry Pi-based
- Atmega and Attiny-based boards

RISC-based single board computers. Compilation steps for AVR assembler are shown on **FIGURE 4**.

PIC ASSEMBLER

PIC assembler is free and open-source assembly language. It supports several types of instructions such as:

- control flow branch instructions
- bite control instructions

This assembler is used for range of PIC-based device families such as:

- PIC10-based
- PIC20-based
- PIC-based single board computers.
  Compilation steps for PIC assembler are shown on **FIGURE 5**. RTOS Development scheme is shown on **FIGURE 6.**

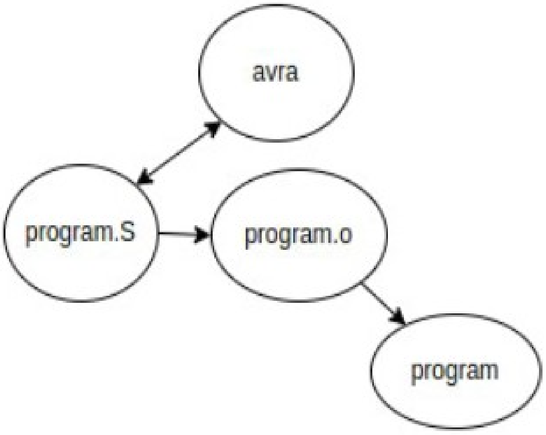

FIGURE 4. AVR assembler compilation

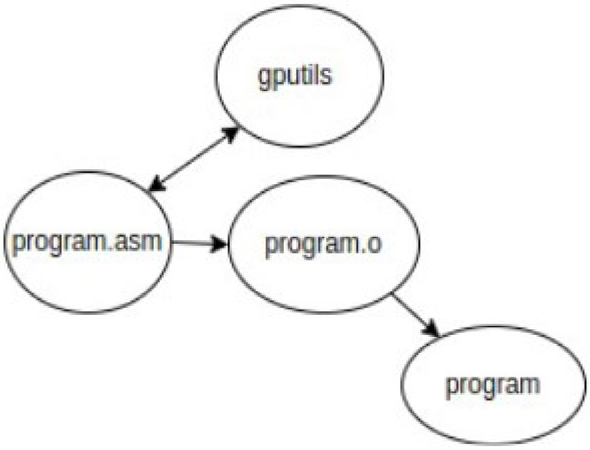

FIGURE 5. PIC assembler compilation

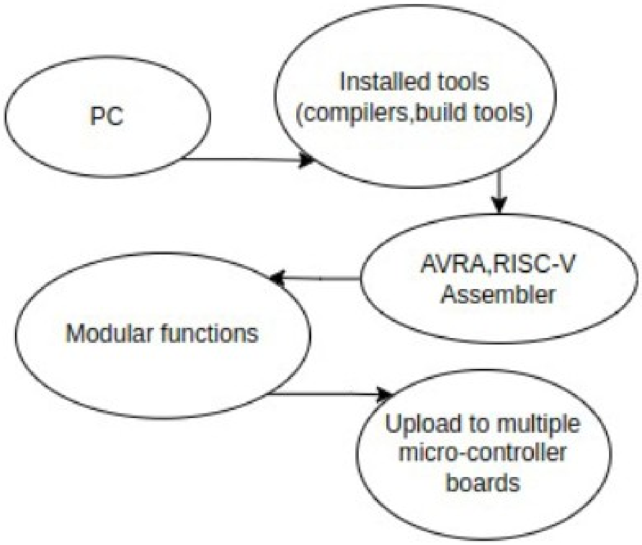
PC
Installed tools (compilers,build tools)
AVRA,RISC-V Assembler
Modular functions
Upload to multiple micro-controller boards

FIGURE 6. RTOS Development scheme

CONCLUSION

Real-time operating systems are used in multiple fields. Features of freeRTOS , Zephyr and others are compared and different ways for their programming are analyzed. Different instruction-sets and assembly languages are described. Target of the paper is complete.

REFERENCES

1. A practical introduction to real-time systems for undergraduate engineering
2. FreeRTOS Documentation, 2024
3. Zephyr Documentation,2024
4. Armbian Documentation,2024
5. QEMU Documentation,2024
6. GNU Binutils Reference Manual, 2024
7. Atmel, ATTiny85 Datasheet,2018
8. Microchip, PIC10F320 Datasheet,2024
9. Linux Manual Pages
10. Design micro-controller module with target to use AVR/PIC assembler (Stanchev, Master Degree Thesis), 2024