Assignment: Real-time Hardware Project with RTOS

Spring 2024, Linnaeus University. 1DT302.

Fredrik Ahlgren

Objective

In this assignment, you will create a real-time hardware project that showcases the use of an RTOS. Your project must include a sensor, a button, an LED or buzzer, and real-time constraints that require the use of an RTOS. You will demonstrate the necessity of an RTOS by replicating the project using a simple MicroPython/CircuitPython program that does not fulfill the real-time requirements.

Learning resources

Either, choose FreeRTOS or Zephyr. You are also free to explore other options as this assignment is OS agnostic as long as you fulfil the RTOS requirements. When starting with real-time operating systems (RTOS) like FreeRTOS or Zephyr, the procedure is straightforward, even though the specific steps might vary between the two RTOSs.

UPDATE 240514: You are able to use the Python RTOS implementation https://github.com/Rybec/pyRTOS .

There is great learning material online to get started:

Introduction to RTOS Part 1 - What is a Real-Time Operating System (RTOS)? | Digi-Key Electronics

FreeRTOS on the Raspberry Pi Pico (RP2040) Part 1: VSCode Setup and Blinky Test!
[UPDATED] Let's run Zephyr RTOS on Raspberry Pi Pico. Ep.1

To begin, follow these general steps:

1. **Download the RTOS source code**: Download the source code for Zephyr or FreeRTOS. Both systems provide pre-configured demo applications for various platforms, which will help you get up and running quickly. Or

https://github.com/Rybec/pyRTOS . Note this implentation only works with CircuitPython (not MicroPython).

2. **Locate the relevant documentation**: Consult the Zephyr supported boards or FreeRTOS supported devices to find the specific documentation for your platform. This documentation will guide you in building, configuring, and setting up your hardware.

3. **Build the project**: Follow the instructions provided in the documentation to locate the appropriate project files in the RTOS directory structure. Open and build the demo project for your chosen platform.

4. **Run the demo application**: Read the documentation to set up the target hardware, download, and execute the demo application. The documentation will also provide details on the demo application's functionality, helping you confirm if it's executing correctly.

5. **Create your own project**: The easiest way to create your own RTOS project is to base it on the provided demo application. Gradually remove the demo functions and source files, replacing them with your own application code. Refer to troubleshooting resources, such as the FreeRTOS FAQ "My Application does not run, what could be wrong?" if you encounter any issues.

There are a vast amount of great Youtube guides and resources on the web. Search and you will get wiser, also don't forget to utilise any AI assistants in your journey.

Project Requirements

- Use RPi Pico 2040 as the microcontroller.
- Implement a real-time hardware project with at least one sensor, one button, and an LED or buzzer. **If you are using PyRTOS you need connectivity WiFi as well.**
- Add communication over Wi-Fi or other wireless communication (optional).
- Implement a test to prove the real-time requirements are fulfilled.
- Replicate the same project using a simple MicroPython/CircuitPython program that does not fulfill the real-time requirements.

Test Protocol

- Perform the test with the RTOS-based implementation and the MicroPython/CircuitPython implementation.
- Compare the results to demonstrate the necessity of using an RTOS to fulfill the real-time requirements of the project.

Report

- Describe the project, including the hardware components and their functions.
- Explain the real-time requirements and why an RTOS is necessary.
- Detail the implementation process for both the RTOS-based and MicroPython/CircuitPython versions of the project.
- Discuss the benefits and limitations of using an RTOS in your project.

Live Demonstration

Prepare a live demonstration of your real-time hardware project, showing the operation of the RTOS-based implementation and the MicroPython/CircuitPython implementation. Compare their performances and discuss the necessity of using an RTOS for your project.

Evaluation Criteria

- Project complexity and creativity.
- Fulfillment of real-time requirements.
- Quality of the test protocol and its ability to demonstrate the necessity of an RTOS.
- Thoroughness of the report, including clear explanations and supporting evidence.
- Effectiveness of the live demonstration in showcasing the project and its real-time requirements.

Notes

This assignment is designed for individual work. The total time required for completing this assignment is approximately 30 hours. Please submit your report and be prepared to present your live demonstration on the due date. Good luck, and have fun exploring the capabilities of RTOS in real-time hardware projects!