Deriving Run-time Monitors from Deductive Verification in RTOS-based Applications

Background

Embedded systems are widely deployed in safety-critical domains, including avionics, industrial automation, automotive electronics, and critical infrastructure. These systems must comply with strict timing and safety constraints. As a result, **real-time operating systems (RTOSes)**—such as **FreeRTOS**—are typically employed together with worst-case timing analysis and real-time scheduling algorithms to ensure that critical operations complete within predetermined deadlines.

Recent research has proposed low-overhead runtime instrumentation for RTOSes to detect anomalies induced by security threats or soft errors [1]. In parallel, a deductive verification approach for proving safety properties of cooperative FreeRTOS applications has been developed [2]. However, the instrumentation process is not yet fully automated and lacks a formal foundation; Deductive verification ensures correctness only in ideal and static scenarios; No mechanisms exist to react to runtime deviations caused by unexpected disturbances or hardware faults.

This intersection provides a unique research opportunity:

- Deductive verification provides a **formal specification** of expected behavior.
- Runtime monitoring can detect **deviations** from that behavior.
- Combining both enables **adaptive**, **efficient**, and **fault-tolerant** execution for RTOS-based systems while avoiding over-provisioning.

Project Description

This thesis investigates how formally verified properties can be translated into **run-time monitors** for FreeRTOS applications, supported by lightweight instrumentation mechanisms.

Core Objectives

- Extract formally verified properties from an existing deductive verification framework for cooperative FreeRTOS applications.
- Translate these properties into **runtime monitors**, serving as guidance to:
    - detect anomalies,
    - identify potential property violations,

    - trigger adaptations or early reactions.
- Integrate lightweight instrumentation mechanisms to observe deviations.
- Conduct a **comprehensive evaluation** on representative FreeRTOS workloads.

Possible Extensions (depending on progress)

- Adaptive scheduling policies guided by runtime feedback.
- Compiler-assisted extraction of metrics to support monitor generation.
- Generalization to other RTOSes (e.g., Zephyr, RTEMS).

Student Profile

This project is suitable for students interested in:

- Dependable and real-time embedded systems
- Formal verification and deductive reasoning
- Systems programming (C/C++, embedded toolchains)

Experience with FreeRTOS, formal methods, or compiler/toolchain development is beneficial.

Supervision

This project will be co-supervised by:

- **Dr.-Ing. Kuan-Hsun Chen** (k.h.chen@utwente.nl)
- **Prof. Dr. Paula Herber** (paula.herber@uni-muenster.de)

You will work at the intersection of dependable embedded systems and formal verification—an excellent preparation for research or industry careers in safety-critical and robust computing.

References

1. Lightweight Instrumentation for Accurate Performance Monitoring in RTOSes
2. Deductive Verification of Cooperative RTOS Applications