

Abstract

Embedded systems play an increasingly important role in our lives and can be found in wide variety of applications. A large part of the software for embedded systems is written in the rather unsafe languages C or C++ due to their excellent performance and low resource requirements. Rust is an alternative systems programming language which is equally efficient but provides additional safety.

Embedded concurrency frameworks provide generic functionality to manage concurrency, one of the key challenges of developing embedded software. In this thesis, three such frameworks based on Rust are presented and their performance is compared. Two of these frameworks are rather lightweight and primarily focus on efficient scheduling. They work closely with the hardware and therefore achieve good performance. The third framework, on the other hand, has different goals, namely execution of untrusted applications and abstraction between applications and the hardware, similar to desktop operating system. However, worse performance must be accepted in return.

Keywords: Embedded Systems, Microcontrollers, Rust, Concurrency Frameworks, Benchmarks