



Baseband Processing using Rust Programming Language

Student Assistant

(max. 19h/week, according to WissZeitVG)

Problem Statement

Traditionally, baseband processing software is implemented using the C or C++ programming languages. Manual low-overhead memory management and access and sometimes even inline assembly instructions maximize performance. However, the resulting code is of low portability and a great deal of safety-critical bugs stem from error-prone manual memory management. The Rust programming language [1] has been designed to overcome these known issues of C/C++. It promises reliability and performance with zero-overhead abstractions and extensive compile-time checks. The goal of this thesis is to evaluate the use of Rust for baseband processing and the development of a physical layer (PHY) library that can be used for flexible modem implementations. A particular focus is the utilization of vector processors [2].

The task may also be done in the scope of a thesis.

Tasks

- Learning about Rust programming language concepts
- Implementation of baseband processing kernels with idiomatic Rust
- Design of an application programming interface (API) for easy and efficient use
- Analysis and written documentation of the results in German or English

Expected Skills

- Knowledge of Rust and/or advanced programming concepts (or willingness to learn it)
- Experience in low-level programming
- Working with tools in a Linux command line environment

Contact Person

Viktor Razilov, viktor.razilov@tu-dresden.de

Please include a recent transcript of records and your CV when contacting.

Recommended References

[1] <https://www.rust-lang.org/>

[2] V. Razilov et al., "Communications Processing Using RISC-V Vector Extension," in Proceedings of International Wireless Communications and Mobile Computing Conference (IWCMC 2022), Dubrovnik, Croatia, (pp. 690-695), May 2022.