

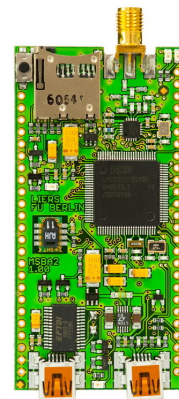
Bachelor/Master Thesis

Embedded Programming Using Modern Languages

Embedded devices usually employ constraint CPUs, are equipped with limited RAM and are often battery powered. Hence, embedded software needs to be highly optimized in regards to its resource consumption in order to provide reasonable responsiveness and battery life. This led to C and assembly being the primary – if not the only supported – programming languages, as they allow an experienced programmer to generate highly optimized code. However, C and assembly are difficult to master and subject to often security relevant and hard to debug memory bugs and data races. Many modern programming languages in contrast provide memory safety and thread-safety in addition to being easier to learn and more productive in development. In recent years the performance of the code generated with modern programming languages as well as the computation power of embedded devices has increased. Thus, an exploration on whether modern programming languages have become viable in embedded programming is worthwhile.

Goals

- Choose one modern programming language that is interesting for embedded programming, e. g.
 - Go [1] (already assigned to a student)
 - Rust [2, 3]
 - ...
- Discuss the maturity of the programming environment and how embedded programming could profit from using it instead of C and assembly
- Evaluate the programming language in regard to its performance and productivity
- Compare the performance and productivity to eLua and MicroPython



MSB-A2 Sensor
Node

References

- [1] Yanni Coroneos. G.E.R.T. <https://github.com/ycoroneos/G.E.R.T>
- [2] Jorge Aparicio. Embedded in Rust. <http://blog.japarc.io/quickstart/>
- [3] Vladimir Pouzanov et al. Zinc <https://zinc.rs/>

Project type Bachelor/Master Thesis
Duration 1 Semester

Contact Marian Buschsieweke
E-Mail marian.buschsieweke@ovgu.de
Room G29-314