

# Rust als Systemprogrammiersprache

Ben Stuart, Hochschule RheinMain

---

Durch den zunehmenden Einsatz von eingebetteten System und IoT gewinnt die Notwendigkeit von sicherer und zuverlässiger Software an Bedeutung. Ein häufiges Problem solcher Geräte sind Fehler bei der Speicherverwaltung und Race Conditions, die oft unerkannt bleiben. Hinzu kommt, dass in solchen Geräten immer öfter Multicore-Architekturen zum Einsatz kommen, was eine weitere Herausforderung darstellt.

Um solchen Problemen entgegenzutreten, wird unter anderem der Einsatz von "sicheren" Programmiersprachen vorgeschlagen. Eine dieser Sprachen ist Rust. Rust wurde von Mozilla mit dem Ziel entwickelt, die angesprochenen Probleme mit Speicherverwaltungsfehlern sowie Data Races zu lösen.

Der Vortrag zeigt am Fallbeispiel der Middleware rsDDS, wie durch den Einsatz von Rust in verteilten eingebetteten Systemen die Softwarequalität verbessert werden kann. Unter anderem werden dabei Herausforderungen bei der Speicherverwaltung an den Schnittstellen, Interoperabilität mit existierenden C Anwendungen, gewonnene Vorteile und der Einfluss auf die Performance und Codegröße im Vergleich zu traditionell entwickelten Anwendungen untersucht.

rsDDS ist eine in Rust implementierte Middleware, die ein Subset des Data Distribution Service (DDS) der OMG realisiert. Wie ihr Vorgänger, die in C implementierte Middleware sDDS<sup>1</sup>, wurde rsDDS für den Einsatz in eingebetteten Systemen und für IoT prototypisch entwickelt. Hierbei wurde der modellgetriebene Ansatz des Vorgängers genutzt.

---