



**Bachelorarbeit
Studienarbeit
Forschungsarbeit
Masterarbeit**
zu vergeben

Rust-basiertes Konzept und Verifikation für den Datenaustausch in nebenläufigen und echtzeitfähigen Anwendungen

Hintergrund

In der modernen Produktion sind echtzeitfähige Steuerungen unerlässlich, um Maschinen und Anlagen effektiv betreiben zu können. Die Programmierung dieser Steuerungen erfordert jedoch eine besondere Herangehensweise, um die strikten zeitlichen Anforderungen zu erfüllen. Dazu müssen sowohl das Betriebssystem als auch das Steuerungsprogramm Echtzeitanforderungen genügen. Daher kommen echtzeitfähige Betriebssysteme und systemnahe Programmiersprachen wie C/C++ zum Einsatz. Rust ist eine noch recht junge Programmiersprache, die durch ihre Speichersicherheit gerade in kritischen Bereichen viele Vorteile bietet. Der Einsatz von Rust in echtzeitfähigen Steuerungen wird daher derzeit untersucht.

Problemstellung

Steuerungsprogramme von Maschinen und Anlagen verfügen über Schnittstellen, damit während des Betriebs Befehle, Daten und Konfigurationen mit der Umgebung ausgetauscht werden können. Zusätzlich werden nicht-echtzeitkritische Programmteile in separaten Umgebungen bzw. Tasks ausgeführt. Daher ist es notwendig, Daten zwischen Echtzeit- und Nicht-Echtzeit-Tasks auszutauschen. Hierfür sind je nach Art und Menge der Daten geeignete Datenstrukturen sinnvoll, so dass der Synchronisationsmechanismus den Echtzeitteil nicht negativ beeinflusst. Für Rust gibt es bisher keine solchen Datenstrukturen.

Aufgabe

In dieser Arbeit sollen daher anhand von typischen Anwendungsfällen notwendige Datenstrukturen identifiziert werden. Darauf aufbauend sollen Synchronisationsmechanismen entwickelt werden, welche die Echtzeit nicht negativ beeinflussen. Zur Validierung sollen formale Methoden eingesetzt werden.

Anforderung

- Interesse Rust
- Interesse an Echtzeitsystemen und Synchronisationsmechanismen von Datenstrukturen
- Freude am Programmieren

Kenntnisgewinn

- Rust
- Synchronisation in nebenläufigen Anwendungen
- Anwendung formaler Methoden

