



Masterarbeit: Können asynchrone Features in modernem C++ die Kluft zwischen bare-metal und Embedded Linux Entwicklung reduzieren?



WORUM GEHT'S?

Heute gibt es unzählige Cross-Plattform-Frameworks, die es erlauben Code nur einmal zu schreiben und dann auf mehreren Plattformen ausrollen zu können. Bekanntere Vertreter sind Qt für den Windows, Linux und MacOS Bereich oder auch Flutter für die App- und Web-Entwicklung. Einige dieser Frameworks decken auch den Bereich „Embedded“ ab, dies setzt dann aber fast immer ein Embedded Linux voraus. Aber natürlich gibt es weiterhin viele Embedded Devices, die bare-metal, also ohne jegliches Betriebssystem oder mit einem RTOS programmiert werden.

Es gibt also eine Kluft zwischen bare-metal und RTOS-getriebenen Embedded Device und Devices mit Linux Betriebssystem. Natürlich kann eine gute Software Architektur helfen, aber ist in einem Entwicklungsprojekt einmal die Entscheidung für eine dieser Varianten gefallen, ist es nur sehr schwer möglich später zu wechseln.

Die zentrale Fragestellung dieser Abschlussarbeit ist, in wie weit diese Kluft zwischen den verschiedenen Ausführungsplattformen und damit die Portierbarkeit von Anwendungssoftware mithilfe von modernen Sprachfeatures verkleinert werden kann. Im Kontext der genannten drei Ausprägungen von Embedded Devices stellt C++ die vorherrschende Programmiersprache dar, welche auch den Ausgangspunkt der Untersuchungen darstellt: In dem neusten C++ Standard haben viele neue Features Einzug gehalten, welche es insbesondere ermöglichen asynchrones Verhalten direkt auf der Programmiersprachenebene auszudrücken (z.B. mittel Coroutines), ohne auf eine der zuvor genannten plattform-spezifischen Abstraktionen (z.B. Tasks, Threads, Framework-APIs) zurückgreifen zu müssen. Erste wissenschaftliche Untersuchungen haben gezeigt, dass diese modernen Sprachfeatures effektiv und effizient auf Embedded Devices eingesetzt werden können. Bislang offen ist hingegen die Frage, in wie weit die Portierbarkeit über die große Spannweite von bare-metal, RTOS bis hin zu Embedded Linux gegeben ist oder diese durch eine geschickte Ausnutzung von Sprachfeatures verbessert werden kann.

Im Kontext der Abschlussarbeit erfolgt die Untersuchung und prototypische Umsetzung einer Demo aus dem Smart Home Bereich mit nebenläufigen Coroutines in C++ sowie Tests auf den drei Typen von Embedded Devices. Die Demo kann eine Zutrittskontrolle, Heizungsregelung, Raumluftüberwachung oder eine Applikation realisieren, die dich interessiert. Hierzu sind zunächst bereits bestehende Ansätze, Bibliotheken etc. zu recherchieren. Die gezielte Analyse der bestehenden Lösungen zeigt Verbesserungen gerade in Hinblick auf die neuen Sprachfeatures in C++ auf. Daraus wird ein eigener Lösungsansatz entwickelt und anhand von Demonstratoren vorgestellt.

Hierzu wird folgendes Vorgehen vorgeschlagen:

- Einarbeitung in die Themenstellung
- Literaturrecherche zu Cross-Plattform-Frameworks
- Entwicklung einer Lösung mit C++ Coroutines
- Dokumentation der Ergebnisse

Die Betreuung der Aufgabenstellung findet in Zusammenarbeit mit Prof. Ulbrich, Lehrstuhl 12 der Fakultät Informatik, TU Dortmund statt.

BE PART OF SMART!

Thematischer Ansprechpartner:

Matthias Steinkamp
Mobil +49 162 2340192
matthias.steinkamp@smartmechatronics.de

Wir freuen uns, dich kennenzulernen!

Folge uns auf Social Media   



Weitere Jobs findest du bei unseren Online-Stellenanzeigen

WIR SIND SMART

Seit 2008 sind wir Entwicklungspartner für intelligente vernetzte Systeme. Unser Fokus liegt auf der Produktentwicklung, dem Produktmanagement und der technischen Beratung für unsere Kunden. Unser Ziel ist es, mit spannenden und innovativen Lösungen unsere Kunden und uns zu begeistern.

Smart Mechatronik wurde als „Top Arbeitgeber Mittelstand 2021“ ausgezeichnet.