

# Analysis and evaluation of the LLVM IR in the context of high performance automotive embedded controllers

Thema:

Analysis and evaluation of the LLVM IR in the context of high performance automotive embedded controllers

Art:

BA

Betreuer:

Christian Wolff, Stefan Voget (Conti)

Student:

Philipp Weber

Professor:

Christian Wolff

Status:

abgeschlossen

Stichworte:

LLVM, Intermediate Representation, compiler, multi language development, C++, Ada, Rust, Haskell, Python

angelegt:

2017-01-25

Antrittsvortrag:

2017-02-13

## Hintergrund

Im Bereich der automotive embedded Entwicklung wird aufgrund der hohen Sicherheitsstandards, seit langem auf „C“ als standardisierte Programmiersprache gesetzt. Durch diese Einschränkung auf nur eine Sprache geht aber die Möglichkeit verloren, die Vorteile andere Programmiersprachen und deren eventuell für bestimmte Aufgaben besser geeignete Tools, zu verwenden. Auch wird die Zusammenarbeit von Teams und Abteilungen erschwert. Da die Sicherheitsstandards aber weiterhin bestehen, werden bei der Continental AG im Moment, mehrere Lösungsansätze für dieses Problem getestet. Einer davon ist die Benutzung des LLVM compiler frameworks, um für, egal in welcher Sprache verfassten Source Code, durch die LLVM Intermediate representation (LLVM IR) eine einheitliche Plattform, auf die sich die Sicherheitsstandards anwenden lassen zu schaffen und trotzdem die Möglichkeit für verschiedenen Ausgangssprachen zu bieten.

## Zielsetzung der Arbeit

Im Rahmen dieser Arbeit soll die LLVM-Technologie hinsichtlich der Anwendbarkeit für automotive embedded controller evaluiert werden. Zu diesem Zweck soll in einem ersten Schritt die Darstellung von relevanten Programmiersprachen in LLVM IR auf Unterschiede, Gemeinsamkeiten, Informationsverlust, etc. untersucht werden. Mit den dadurch erworbenen Kenntnissen sollen dann

spezifische Testalgorithmen in den entsprechenden Sprachen als Anwendungsbeispiele verfasst und auf ihre Ausführbarkeit in einem embedded System getestet werden. Im Rahmen dieser Test kann dann ein Fazit über die Anwendbarkeit von LLVM gezogen werden und somit der Grundstein für eine eventuelle zukünftige Vertiefung dieses Lösungsansatzes gelegt werden.

## Konkrete Aufgaben

- Erstellen einer Liste der Darstellung von Sprachelementen relevanter Programmiersprachen (Rust, Ada, C++, Haskell, Python) in LLVM IR
- Analyse der genannten Liste
- Entwicklung eines Testszenarios in Form von Algorithmen in den jeweiligen Sprachen
- Ausführen der Algorithmen in einem embedded system
- Evaluation der Eignung von LLVM für die Nutzung als einheitliche Zwischenplattform für multi language development

## Erwartete Vorkenntnisse

- Praktische Kenntnisse in der Programmierung
- Basiswissen über Compiler

## Weiterführende Quellen

Nach Absprache mit dem Betreuer

From:

<https://wiki.mi.ur.de/> - **MI Wiki**

Permanent link:

[https://wiki.mi.ur.de/arbeiten/llvm\\_ir\\_evaluation?rev=1507562321](https://wiki.mi.ur.de/arbeiten/llvm_ir_evaluation?rev=1507562321)

Last update: **09.10.2017 15:18**

