Java on the bare ARM metal

Titel of the master thesis:	Java on the bare ARM metal
Advisor:	Prof. Dr. Urs Graf
Expert:	Bernhard Sprenger
Industrial partner:	Open Source Projekt: *deep*
Author:	Marcel Gehrig
Submission date:	10. August 2018

Lead:

In this paper a toolchain is developed, which can be used to load and run a *deep* application on a development board with an ARM processor.

The toolchain will be extended, so it can be used with the *gdb* debugger. Advanced *gdb* features like single stepping through the Java source code on a host-PC can be used to debug the program running on the ARM processor.

Abstract:

The compiler from the *deep*-project can be used to compile Java code directly for an embedded PowerPC processor. It is possible to develop a simple motor controller or a complex robot controller with real-time capability in plain Java.

PowerPC processors are no longer in widespread use. They are now used only for some special applications. For this reason, *deep* is being developed for the use with ARM processors.

deep is currently used for education at the NTB and it should remain usable after the switch to the ARM micro-architecture.

For the development of the *deep* runtime environment, a hardware debugger is needed to read and write memory and processor registers.

gdb is a software debugger, which is popular for programming languages like C and C++. It has features like source code lookup, single stepping and breakpoints. These are really helpful features and they should also be provided for the development of *deep* applications.

In this paper an ARM processor and a matching development board will be evaluated. The development board should be cheap enough for NTB lessons. The processor has to have very good connectivity for an FPGA so that demanding robotic projects can be implemented with the same processor.

A toolchain is then designed, which can be used, to write a *deep* application to the memory of the processor.

The existing software interface for the hardware debugger will be adapted, so it can be used with the new toolchain.

A *deep* application will be supplemented with STABS debug information. The supplemented application can than be compiled and executed on the ARM processor. *gdb* can then be used on a host computer to debug the *deep* application on the ARM processor using features like source code lookup and single stepping.