

# Zusammenfassung

Das *deep*-Projekt ermöglicht es schon jetzt, eine Java-Applikation direkt für PowerPC-Prozessoren zu entwickeln. Dadurch ist es möglich, eine echtzeitfähige Regelung oder Robotersteuerung in Java zu entwickeln.

Prozessoren auf der Basis der PowerPC-Architektur sind aber nicht mehr weit verbreitet und zu einem Nischenprodukt geworden. Aus diesem Grund wird *deep* zurzeit für die ARM-Architektur weiterentwickelt.

*deep* wird für den Unterricht der NTB verwendet und soll auch nach der Umstellung zur ARM-Architektur weiterhin für die Lehre verwendet werden können.

Um die Laufzeitumgebung von *deep* entwickeln zu können, wird ein Hardware-Debugger benötigt, der den Speicher und die Register von einem ARM-Prozessor lesen und schreiben kann.

Der *gdb*-Debugger ist ein weit verbreiteter Software-Debugger, der besonders für Sprachen wie C und C++ verwendet wird. Er bietet Features wie Sourcecode-Lookup, Single-Stepping und Breakpoints. Diese Features sind sehr nützlich und sollen auch für die Entwicklung von *deep*-Applikationen zur Verfügung stehen.

In dieser Arbeit wird ein ARM-Prozessor inklusive Experimentierboard gesucht, der günstig genug für den Unterricht ist und eine sehr gute Anbindung an ein FPGA hat, damit auch anspruchsvolle Robotik-Projekte damit realisiert werden zu können.

Für diese Hardware wird dann eine Toolchain entwickelt, mit der ein vom *deep* kompiliertes Programm in den Speicher des Prozessor geladen und gestartet werden kann.

Das bestehende Software-Interface für den Hardware-Debugger wird so angepasst, dass es mit der neuen Toolchain kompatibel ist.

Es wird gezeigt, dass der *gdb*-Debugger, inklusive der oben genannten Features, genutzt werden kann, um eine *deep*-Applikation zu debuggen, die auf dem ARM-Prozessor läuft. Dafür wird eine *deep*-Applikation mit STABS-Debuginformationen erweitert.