



## 19579 - PR: Programming Wireless Sensor Networks

Das Jam Build-System und Interrupt-Handling auf dem ARM

Oliver Hahm, Kaspar Schleiser

Department of Mathematics and Computer Science

Institute of Computer Science

November 3rd, 2010

- Open-Source Build-System als Ersatz für Make
- Implementierungen für alle gängigen Betriebssysteme
- Einfache und übersichtliche Syntax
- Bringt bereits viele wichtige Regelsätze mit (so genannte Jamrules)
- Aufgeteilt in folgende Dateien
  - `Jamfile`
  - `Jamrules`

- Leerzeichen dienen zur Trennung von Zeichenketten (auch vor Doppelpunkt oder Semikolon!)
- Wichtige Befehle und Variablen:
  - *SubInclude* - zum Einbinden von Unterverzeichnissen
  - *SubDir* - zum Setzen des aktuellen Unterverzeichnisses
  - *CCFLAGS* - Parameter für den C-Compiler
  - *HDRS* - Pfade zu den Header-Dateien
  - *TOP* - Wurzelverzeichnis

- Einführung von Modulen
- Jeder Modulordner und die übergeordneten Ordner enthalten ein Jamfile
- In der *Jamrules* werden globale Variablen gesetzt (können in Unterordnern überschrieben oder erweitert werden)
- *Jamrules.common* definiert und überschreibt Regeln
- Weitere Befehle und Variablen:
  - *Module* - Definiert ein Modul und seine Abhängigkeiten, z. B.:  
`Module temperature : main.c temp.c : swtimer sht11 ;`
  - *UseModule* - Gibt die Module an, die verwendet werden sollen