

## Inhaltsverzeichnis

Inhaltsverzeichnis . . . . .	i
Abbildungsverzeichnis . . . . .	iii
1 Einleitung . . . . .	1
1.1 Über Yocto, OpenEmbedded, poky, meta-daten und Bitbake . . . . .	1
1.1.1 Die Yocto Community . . . . .	2
1.1.2 Die OpenEmbedded Community . . . . .	2
1.2 Docker . . . . .	2
2 Bitbake . . . . .	3
2.1 Bitbake Buildprozess . . . . .	3
2.2 Bitbake build tasks . . . . .	3
2.2.1 Fetch Task . . . . .	4
3 Setup Eclipse . . . . .	5
3.1 Installation, Einrichtung und Plugins . . . . .	5
3.2 Cross Compile und Remote Debugging Einstellungen . . . . .	5
Glossar . . . . .	8
Literatur . . . . .	9



## **Abbildungsverzeichnis**



## 1 Einleitung

### 1.1 Über Yocto, OpenEmbedded, poky, meta-daten und Bitbake

Das Yocto Project bezeichnet eine Community Gruppe welche sich zur Aufgabe gemacht hat, das Erstellen von Linux Distributionen für eingebettete Systeme zu vereinfachen.

Zusammen mit der OpenEmbedded Community pflegt das Yocto Project eine Software Build Umgebung mit dem Namen Bitbake, bestehend aus Python- und Shell-Skripten, welches das Erstellen von Linux Distributionen koordiniert. Parallel zu Bitbake entstehen und wachsen innerhalb der Yocto Community verschiedene Tools wie z.B. Wrapper, welche das Arbeiten mit Bitbake vereinfachen sollen, indem diese Tools beispielsweise einzelne Workflows abbilden.

Des Weiteren pflegen beiden Communities Dateien mit Metadaten (Metadatei), die in Form von Regeln beschreiben wie Software Pakete innerhalb für unterschiedliche Distribution und Hardware durch Bitbake gebaut werden müssen, erforderlich sind.

Diese Regeln werden Recipes genannt. Regeln bzw. diese Metadaten und somit auch das Build-System Bitbake arbeiten nach einem Schichten Modell. Dabei beschreiben Meta-Daten auf unterster Schicht allgemeine grundlegende Anleitungen zum Übersetzen der wichtigsten Funktionen eines Betriebssystems. Höhere Schichten erweitern (detaillieren) oder überschreiben diese grundlegenden Rezepte in immer höheren Schichten. So setzt auf Beschreibungen der Hardware (BSP, Board Support Packed Schichten) Schichten der Software- und Anwendungsschicht auf. Ein solches Schichtenmodell ermöglicht es, einzelne Schichten auszutauschen oder darunterliegende Einstellungen abzuändern. Ein vereinfachtes Modell ist abgebildet unter Gonzalez, 2018, S.26

### 1.1.1 Die Yocto Community

Die Yocto community stellt Meta-Daten / Recipes bereit, um ein minimalistisches Betriebssystem mit grundlegenden Linux Tools unter der Virtualisierungsumgebung QEMU für verschiedene Architekturen starten zu können.

Das Minimalisten Betriebssystem der Yocto Community wird „poky“ genannt.

Zudem Pflegt es zusammen mit der OpenEmbedded community das Buildsystem *bitbake*.

### 1.1.2 Die OpenEmbedded Community

Die OpenEmbedded Community stellt Meta-Daten Rezepte bereit, die auf dieses minimalistische Betriebssystem aufsetzen und genutzt werden können um ein Betriebssystem nach eigenen Wünschen gestalten (zusammen setzen) und für Hardware Plattformen konfigurieren zu können. Hierzu gehören sowohl Hardware als auch open Source Software Beschreibungen. Zusammen mit der Yocto Community pflegen und erweitern das Build Umgebung *Bitbake*, welches verschiedene Aufgaben in geregelter Reihenfolge im Multicore Betrieb durchführt. Weiteres in ??, Seite ??.

## 1.2 Docker

Bitbake benötigt, neben einem aktuellen Python (2) Interpreter, verschiedene Tools. So u.a. Git zum Herunterladen von Source Detain. Alle diese Abhängigkeiten wurden in einem Docker Container zusammengefasst. Bitbake kann unter unschädlichen Betriebssystemen und Plattformen auf gleiche Weise genutzt werden, ohne dass es und seine Abhängigkeiten neu konfiguriert werden müssen. Im Gegensatz zu anderen Virtualisierungstechniken hat Docker trotz Virtualisierungstechniken keinen großartigen Performance Verlust. Von einem Gebrauch von klassischen Virtualisierungstechniken ist aus Performance Gründen dringend abzuraten.

## 2 Bitbake

### 2.1 Bitbake Buildprozess

Die Buildumgebung *Bitbake*, im Kern bestehend aus Python- und Shellscripen die mit unterschiedlichen Softwareentwicklungstools wie GIT, Make oder Autotools interagieren, führt in einer geregelten Reihenfolge verschiedene Aufgaben aus. Die Abbildung auf Gonzalez, 2018, S. 20 zeigt abstract den Buildprocess:

- Reihenfolge der angepassten config files gepasst werden.
- Parsen der config files.
- Buildschritte (Siehe 2.2 Seite 3)
- Packetieren in unterschiedliche Pakettypen
- Generieren und bereitstellen von **SDK**!s
- Aufräum- und Nacharbeiten

### 2.2 Bitbake build tasks

Jedes *recipe* erbt eine Reihe von standard Build-Tasks. Hierzu gehören u.a.

- Herunterladen (fetchen) und Sammeln von Source Dateien (z.B. Git Repositorien, ftp Servern oder lokalen Dateien.)
- Übersetzen, konfigurieren, patchen, installieren, verifizieren usw. von Paketen auf mehreren Prozessor Kernen
- Bereitstellen von Entwicklung und Verwaltungstools

Eine Auflistung der Standard-Task kann, mitsamt kurzer Beschreibung, entnommen werden Gonzalez, 2018, S. 171-172

### 2.2.1 Fetch Task

Eine der Hauptaufgaben von Bitbake besteht dadrinne, die jeweiligen Softwarepakete in Ihren benötigten Versionen und Revisionsständen zusammen aus unterschiedlichen Quellen zusammen zu sammeln und dem Buildsystem zur Verfügung zu stellen. Solche Quellen können sein:



## 3 Setup Eclipse

### 3.1 Installation, Einrichtung und Plugins

Zur Entwicklung von C/C++ kann Eclipse CDT verwendet werden.

Zudem sind zum Cross compilieren und Remote Debuggen sowie zum remote Deployen nachfolgende Plugins bzw zusätzliche Eclipse-Softwaremodule nötig: (*Help -> Install new Software* )

- C/C++ Remote (Over TCF/TE) Run/Debug Launcher.
- Remote System Explorer User Actions
- TM Terminal via Remote System Explorer
- TCF Target Explorer

Des weiteren stellt die Yocto download Seite [downloads.yoctoproject.org](http://downloads.yoctoproject.org) ein SDK Plugin bereit, welches das Konfigurieren von Remote Einstellungen für jeweilige Entwicklungsprojekt vereinfacht und zusammenfasst. Zum installieren muss beispielsweise zu den Installations-Quellen in eclipse <http://downloads.yoctoproject.org/releases/eclipse-plugin/2.6.1/oxygen/> hinzu gefügt werden.

### 3.2 Cross Compile und Remote Debugging Einstellungen

TODO



## Glossar

B | D | M | R | W | Y

### B

**Bitbake** Bitbake ist ein Framework ähnlich wie GNU Make, bestehend aus Python Sripten welche das Erstellen von Linux Distributionen mittels Metadaten koordiniert. . 1

### D

**Docker** Docker ist eine Software zur virtualisierung von einzelnen Anwendungen. 2

### M

**Metadaten** Metadaten oder Metainformationen sind Informationen über andere Daten. 1

**Metadaten** Metadaten oder Metainformationen sind Informationen über andere Daten. 7

**Metadatendatei** Eine Datei, welche Metadaten (Informationen zu anderen Daten) über andere Daten enthält.. 1

### R

**Recipies** Rezepte oder Anleitungen. 1

### W

**Workflow** Ein Workflow ist ein Arbeitsablauf und beschreibt Schritte in ihrer Reihenfolge die nötig sind, um eine Aufgabe, Arbeitspaket oder etwa eine Anweisung zu erfüllen. Dabei sind die Arbeitsschritte häufig wiederkehrend in anderen Arbeitsabläufen. . 1

**Y**

**Yocto Project** Eine Community Gruppe welche eine Software Buildumgebung pflegt und weiterentwickelt, sowie Metadaten für diese Buildumgebung zur verfügung stellt um ein minimalistisches Linux system mit grundlegenden Tools unter der virtualisierungsumgebung QEMU zu starten.. 1

## Literatur

### Books

Gonzalez, Alex (2018). *Embedded Linux Development Using Yocto Project Cookbook* -. 2nd Revised edition. Birmingham: Packt Publishing. ISBN: 978-1-788-39921-0.