1 Yocto Image

1.1 Einfhrung

Kevin und Stephan haben whrend diesem Projekt das Thema Yocto Image bearbeitet. Dabei sollten wir mit Hilfe von Yocto ein Image erstellen, welches eine Linux Distribution und die, von vorherigen Gruppen erstellte, Anwendungssoftware des FHF-Trains. Allerdings hat die Linux Distribution wegen des Linux Kerns nicht funktioniert, weshalb wir uns dafr entschieden haben, nur ein RootFile-System mit der Anwendungssoftware zu erstellen, welche dann auf den FHF-Train bertragen wird.

1.2 Definition Yocto

Das Yocto Projekt ist ein Open Source Werkzeug, welches fr Entwickler von embedded Linux Systeme geschaffen wurde. Das Ziel von Yocto ist es, durch Einfgen und Ausschliessen gewhlter Komponenten ein Linux System zu erstellen, welches von dem Nutzer auf das Endgert angepasst wird. Um das alles zu ermglichen, benutzt Yocto hauptschlich BitBake und Open Embedded. Dafr haben wir Poky verwendet, welches uns ermglicht den Kern, den bootloader und ein RootFile-System zu erstellen.

1.3 Definition BitBake

BitBake ist ein Tool, welches in Metadaten gegebene Aufgaben ausfhrt. Diese Metadaten werden in BitBake-spezifischen Dateien gespeichert. Dazu gehren Rezepte (.bb), Konfigurationsdateien (.conf), Klassen (.bbclass) und Append Dateien (.bbappend). Append Dateien erweitern Rezepte, die den gleichen Namen besitzen. Diese Dateien werden in 'layer.conf'-Dateien aufgelistet und werde durch eine 'bblayers.conf' an BitBake referenziert.

Es gibt verschiedene BitBake Befehle, die einem erlauben, aus den oben genannten Dateien ein Image zu erstellen. Dazu gehren:

- core-image-minimal: Das kleinste mgliche Image, welches dem Zielsystem erlaubt die Kommandozeile zu booten.
- core-image-base: Ein Image, welches core-image-minimal durch Hardwareunterstzung wie beispielsweise W-LAN und Bluetooth erweitert.
- core-image-full-cmdline: Dieses Image erweitert minimal durch Kommandozeilen-Befehle.

1.4 OpenEmbedded

Open Embedded ist eine atomatisiertes build Framework und eine Cross-Compiling Umgebung. Diese Umgebung ermglicht es Mageschneiderte Linux Images fr eingebettete Systeme zu erstellen.
muss noch weiter erlutert werden

1.5 Hob

Hob ist ein BitBake User Interface, welches das Erstellen der Images vereinfacht. ber Hob kann man einfach das Zielsystem und den BitBake-Befehl auswhlen. Wenn man eine eigene Layer-Datei erstellt hat, kann man diese auch der Liste der aktuellen Layers hinzufgen. Des Weiteren bietet Hob eine komplette Liste von Paketen, die in den Layers aufgefhrt sind und lsst den User auswhlen, welche er installieren mchte. Nach den Einstellung startet Hob den BitBake-Befehl und erstellt eine Image Datei.

2 Ablauf Yocto

2.1 Erstellung einer virtuellen Maschine

Um Yocto benutzen zu knnen mussten wir zur aller erst eine virtuelle Linux Maschine erstellen Yocto nur auf einem Linux basierten Betriebssystem funktioniert. Die folgenden Betriebssysteme wurden von den Entwicklern getestet und als stabil bezeichnet:

- Ubuntu 12.04 (LTS)
- Ubuntu 13.10
- Ubuntu 14.04 (LTS)
- Fedora release 19 (Schrdinger's Cat)
- Fedora release 20 (Heisenbug)
- CentOS release 6.4
- CentOS release 6.5
- Debian GNU/Linux 7.x (Wheezy)
- OpenSUSE 12.2

Da wir mit Ubuntu bis jetzt am meisten Erfahrung hatten, haben wir uns recht schnell dafr entschieden Ubuntu 14.04 (LTS) zu benutzen.

2.2 Yocto Installation

Sobald wir mit der Erstellung einer virtuellen Maschine fertig waren mussten wir uns um die Installation vom Yocto Projekt kmmern. Um Yocto (bzw. Poky) zu installieren mussten wir zuerst die folgenden Pakete mittels der Kommandozeile apt-get install installieren:

- gawk
- wget
- git-core
- diffstat
- unzip
- texinfo
- gcc-multilib
- build-essential
- chrpath
- socat
- cpio
- python
- python3
- python3-pip
- python3-pexpect
- xz-utils
- debianutils
- iputils-ping
- libsdl1.2-dev
- xterm

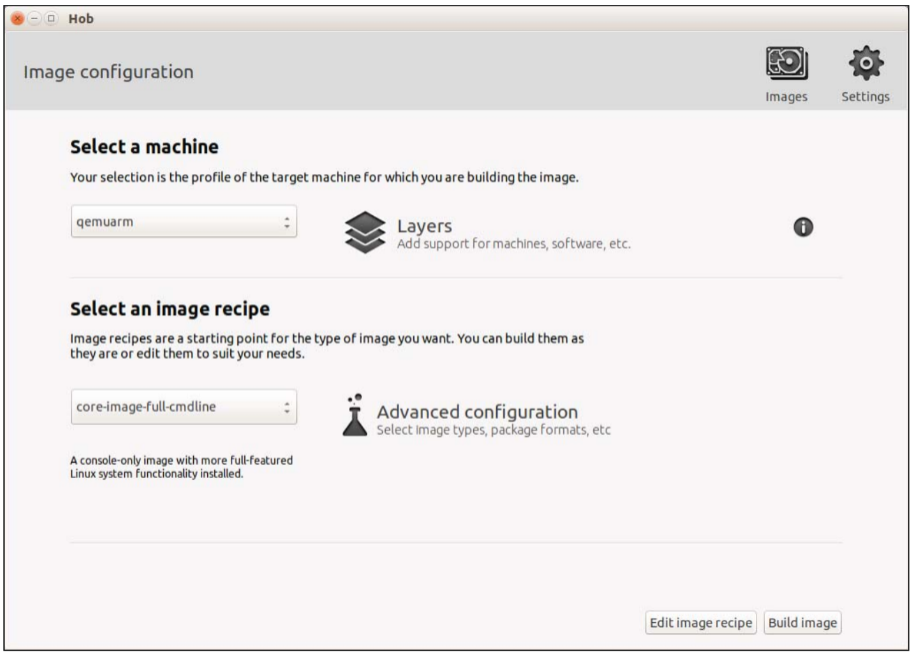
Danach haben wir im Homeverzeichnis einen Yocto Ordner angelegt indem wir mittels der Kommandozeile git clone git://git.yoctoproject.org/poky –branch daisy den source Code vom Yocto Projekt heruntergeladen haben.

2.3 Erstellung eines einfachen Images

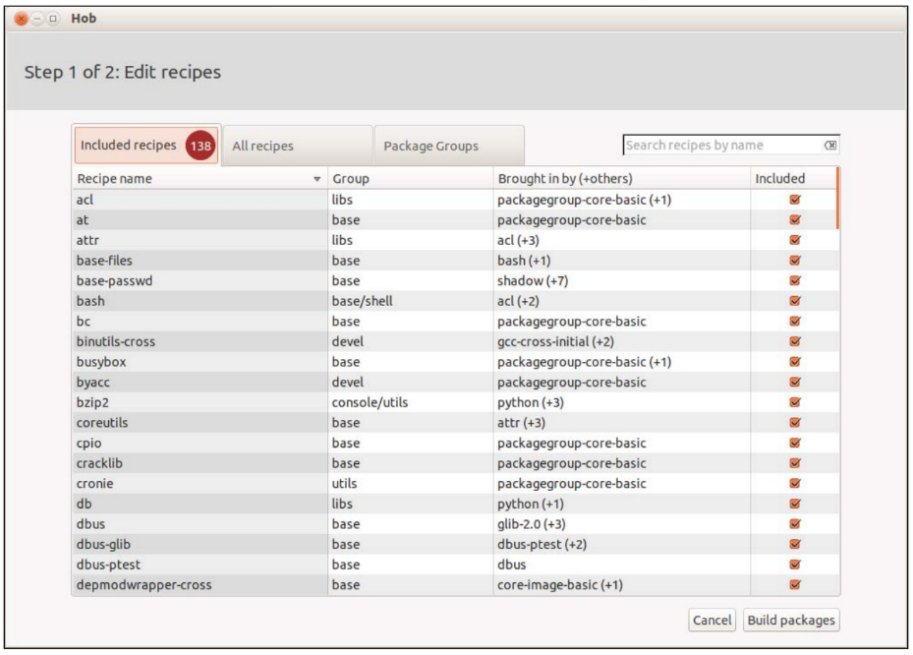
Mit Hilfe der Kommandozeile 'source poky/oe-init-build-env' wird es schlielich mglich Poky zu benutzen. Poky benutzt 'BitBake' um aus 'Rezpeten', Images zu machen. Bitbake ist im Grunde genommen ein Make Werkzeug das andere Make Werkzeuge (wie die, die wir ganz am Anfang installiert haben) benutzt um das Image zu bauen. Wir haben uns entschieden zuerst ein einfaches Image zu bauen mit Hilfe vom core-image-minimal Rezept. Dieses baut ein einfaches und leeres Root Filesystem, das eigentlich zum Testen und entwickeln von Kernels und Bootloaders gedacht ist. Um dies zu bewerkstelligen haben wir die Kommandozeile 'bitbake core-image-minimal' benutzt. Da wir den BitBake allerdings auf einer virtuellen Maschine und nicht auf einem nativen Linux Betriebssystem haben laufen lassen hat er um die 12 Stunden gedauert (auf einem nativen Betriebssystem soll der BitBake zwischen 3 und 4 Stunden dauern). Als das Image gebaut war konnten wir es mit Hilfe von der Kommandozeile 'runqemu qemux86' emulieren.

2.4 Erstellung eines Root Filesystems mit Hob

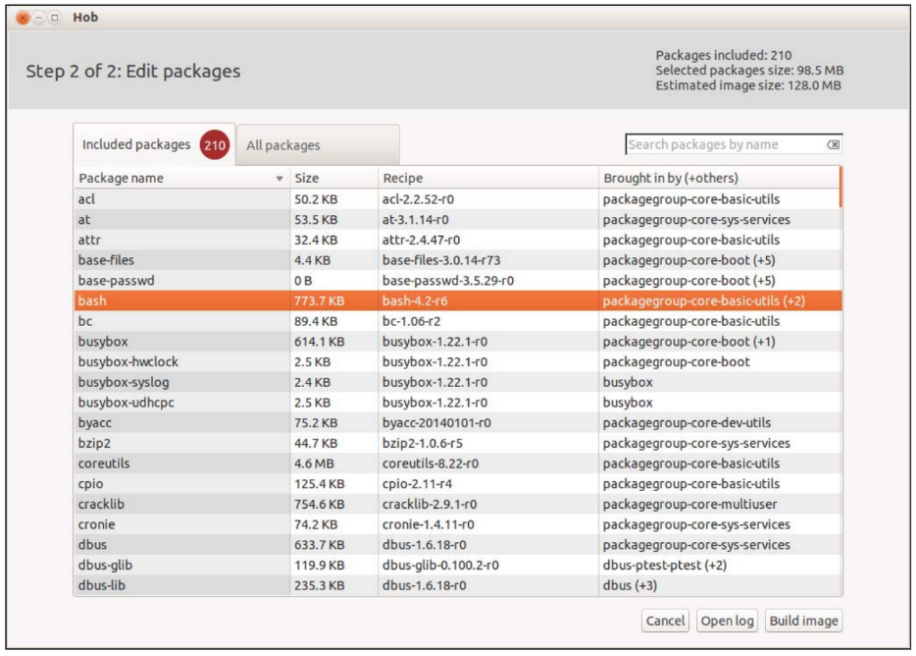
Poky bietet ein graphisches User Interface namens Hob an das den BitBake Prozess bersichtlicher fr den Benutzer macht:



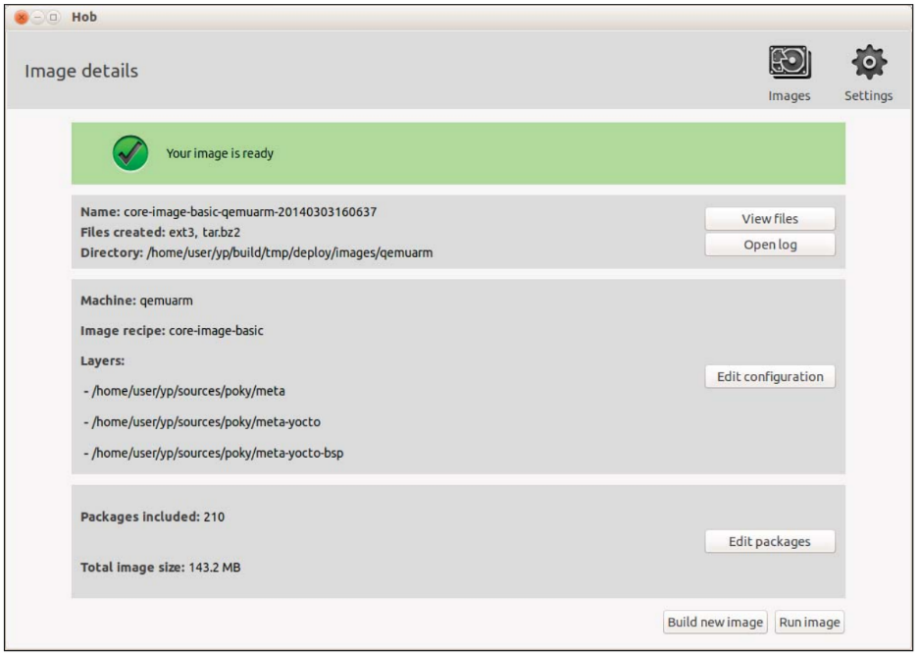
Mit Hilfe von Hob kann man dann ganz einfach eine ziel Maschine und ein Rezept aussuchen.



Danach kann man sogar whlen welche Pakete das Image braucht und welche nicht.



Nach dem auswhlen der Pakete kann der Build des Images beginnen.



Wenn der Build dann zu Ende ist kann das Image emuliert werden: