**Getting Amstrong on a SD card**

Ubuntu distro: 14.04.5 Trusty

Angstrong distro: v2016.12 / Yocto 2.2

**Getting the Preloader and U-Boot images**

These are the first 2 images to be loaded during the boot sequence. For getting them, it is recommended to follow the tutorial in [4]. This is an extensive tutorial covering all the steps for obtaining a working Linux system. However, the last steps indicates how to obtain an almost handcrafted kernel and root filesystem, by using the Buildroot tool. In our case, we opted to build the Angstrom distribution, with an specific version made by Altera for their boards. For this purpose, the instructions for getting the kernel and root filesystem are explained below

**Getting kernel image and root filesystem**

The main instructions about getting the Angstrong required images were obtained at Rocketboards.org [1]. The files are stored on a github repository, and are cloned using the **repo** tool. Thus, prior to building the images, this tool has to be prepared, as indicated in the angstrom-manifest repository [2]:

|  |
| --- |
| *$ mkdir ~/bin*  *$ PATH=~/bin:$PATH*  *#Transfer repo tool data from server*  *$ curl http://commondatastorage.googleapis.com/git-repo-downloads/repo >~/bin/repo*  *$ chmod a+x ~/bin/repo* |

For building the source of the kernel and root file system, the **OpenEmbedded** framework [3] is used. For using it, is necessary to install the libraries listed in the *Getting started* section under Ubuntu/Debian subsection in their website:

|  |
| --- |
| *$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \*  *build-essential chrpath libsdl1.2-dev xterm* |

from [3] (http://www.openembedded.org/wiki/OE-Core\_Standalone\_Setup):

1) Clone the repositories for OE-Core (the core metadata) and BitBake (the build tool), checking out the latest stable branches of each one in turn:

git clone git://git.openembedded.org/openembedded-core oe-core

cd oe-core

git clone git://git.openembedded.org/bitbake bitbake

2) Check out the latest stable branches of both OE-Core and BitBake:

git checkout morty

cd bitbake

git checkout 1.32

cd ..

3) Set up the environment and build directory:

source ./oe-init-build-env [<build directory>]

The optional build directory may be specified, otherwise it is assumed you want to use the directory named "build".

4) First time configuration

The first time you run oe-init-build-env, it will setup the directory for you and create the configuration files conf/bblayers.conf and conf/local.conf. You should at least review the settings within the conf/local.conf file.

5) Build something: (not necessary)

bitbake <target>

A good simple place to start is bitbake core-image-minimal. This will do basic system sanity checks and build a small but practical image. If your system needs additional software installed, or other environment settings it will tell you what is needed.

If the previous step is done, an error may appear when building the project, due to incoherency with the bitbake configuration. If this happens, edit the *.repo/manifests/conf/bblayers.conf* file and check that the meta-altera path is correct.

gedit .repo/manifests/conf/bblayers.conf

the path to meta-altera (use Nautilus to search for "meta-altera")

- the path is currently : ~/bin/angstrom-manifest/sources/meta-altera

Once *repo* is downloaded and added to the system environment, and auxiliary libraries are installed; the Angstrom scripts can be cloned to a local folder and set up [1]:

|  |
| --- |
| *# Create a folder for the files and set it as the working directory*  *$ mkdir angstrom-manifest*  *$ cdangstrom-manifest*  *# provide git with email and name*  *git config --global user.email "you@example.com"*  *git config --global user.name "Your name"*  *#Download the repository metadata and checkout the desired version*  *$ repo init -u git://github.com/Angstrom-distribution/angstrom-manifest -b angstrom-v2016.12-yocto2.2*  *#Download the required layers*  *$ repo sync*  *# Set up the build environment to the cyclone5 machine*  *$ MACHINE=cyclone5 . ./setup-environment* |

It is recommended [1] to edit *.repo/manifest.xml* before sync and change the meta-altera entry to:

|  |
| --- |
| *<project name="kraj/meta-altera"path="sources/meta-altera"remote="github"revision="0c6e036fdfec58b69903a10e886a9dfae8fe4c9f"upstream="master"/>* |

gedit .repo/manifest.xml

Later on, open the conf/local.conf file and add the following lines:

|  |
| --- |
| UBOOT\_CONFIG = "de0-nano-soc"  KERNEL\_DEVICETREE = "socfpga\_cyclone5\_de0\_sockit.dtb"  IMAGE\_FSTYPES += "socfpga-sdimg"  PREFERRED\_PROVIDER\_virtual/kernel = "linux-altera"  PREFERRED\_VERSION\_linux-altera = "4.7%" |

Finally, build both the kernel and the root file system:

|  |
| --- |
| *$ bitbake virtual/kernel*  *$ bitbake console-image* |

The generated files will be at *<repository-directory>/deploy/glibc/images/*

**NOTE:** The generated *zImage* file is a symbolic file to another one. For copying the contents of the one it is pointing to, you can use the *cp* linux tool

**IMPORTANT:** The root filesystem had to be copied to an ext4 partition. Otherwise, the kernel is not able to mount it

**Setting up the root filesystem on the ext4 partition**

The root filesystem built following the previous steps was generated on different formats. Hence, it is possible to set it up on the SD on different ways. One of them is uncompressing the <Angstrom...tar.xz> file directly on the ext partition of the SD card:

|  |
| --- |
| *$ sudo tar -xf <compressed-file.tar.xz>-C /media/<name-of-mounted-SD-ext-partition>* |

[1] Angstrom On SoCFPGA, by Dalon WestergreenJan 2017 <https://rocketboards.org/foswiki/view/Documentation/AngstromOnSoCFPGA_1#How_to_Build_Angstrom_for_SoCFPGA>

[2] Angstrom-distribution / angstrom-manifest repository <https://github.com/Angstrom-distribution/angstrom-manifest>

[3] OpenEmbedded main page <http://www.openembedded.org/wiki/Main_Page>

[4] Embedded Linux Beginners Guide,

<https://rocketboards.org/foswiki/view/Documentation/EmbeddedLinuxBeginnerSGuide>