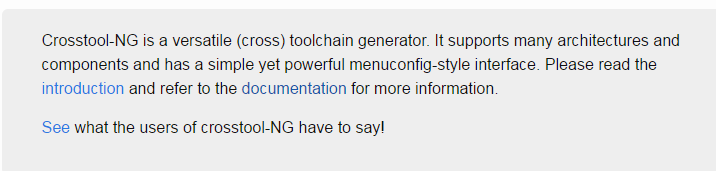
**Cross tool ng**



Download and extract.

Released 1.23.0

Start with host Linux ( Ubuntu ) requirements

<http://crosstool-ng.github.io/docs/os-setup/>

sudo apt-get install gcc gperf bison flex texinfo help2man make libncurses5-dev \

python-dev

The package contains autotools makefile so it is need to be configured

Generation of the ct-ng utility and miscellaneous build scripts.

./configure

make

sudo make install

This will cause the tool chain to be produce at the same directory.

\*Note – ct-ng should be in the path, if not add it or just type bin/ct-ng instead of ct-ng

Let’s move it to different directory, so we can do make clean without cleaning the ng cross tool it self

CT\_BUILDTREE= /home/elia/ct\_builds

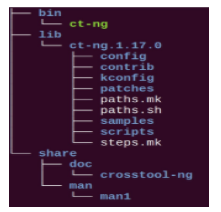
./configure --prefix=$CT\_BUILDTREE

make

sudo make install

At this point, if the toolchain build tree was generated outside the source tree, then the crosstool-NG sources are now no longer required for the rest of the toolchain build process (i.e. Phase 2).

In any case, after running make install, the hierarchy of the generated toolchain build tree should resemble:



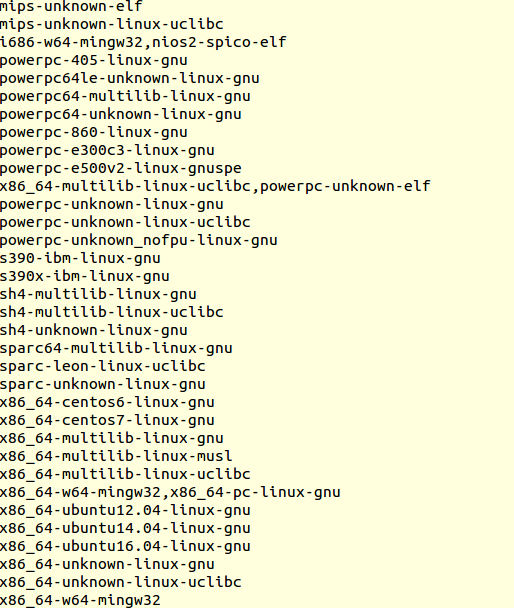
Took from:

<http://nairobi-embedded.org/040_using_crosstool-ng.html>

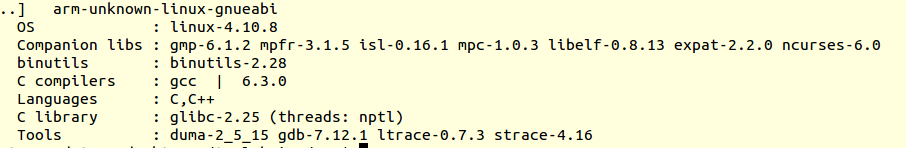
lets move now to the place where the ng-ct tool was installed and type:

bin/ct-ng list-samples

This will show us pre samples



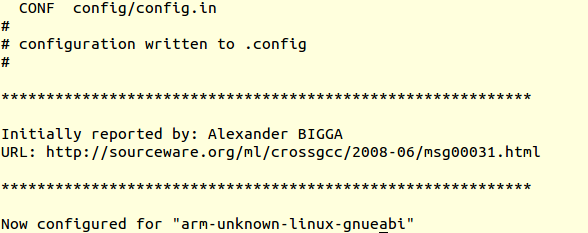
ct-ng show-arm-unknown-linux-gnueabi



We can see the compiler version, the languages supports , c library and bin utils.

To select the sample starting point:

ct-ng arm-unknown-linux-gnueabi

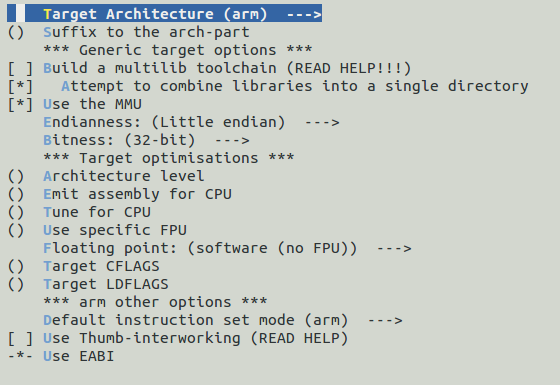


ct-ng menuconfig

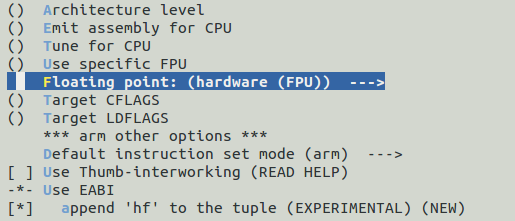
In case we want to compiler the default, we just need to do ct-ng build

Let’s go over some options, and change a bit

You can find the target processor in target options:



Let’s change the compiler to HF – hardware floating point, the one that we need to raspberry pi 3



Here is the **hf** that will append, lets see at the end.

For now, this is the only thing we want to change.

The **default** directory for the created tool chain is $HOME/x-tools, but can be configured under 'Paths and misc options':

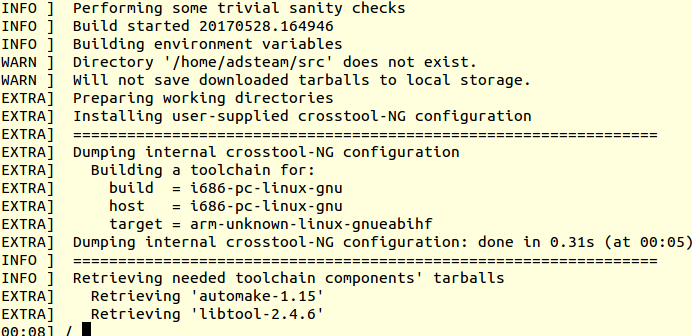
We can add environment var

export $CT\_TARGET to store the created toolchain

Invoking:

ct-ng build

And build started:



After an hour or so , build completed

