**Prepare buildroot image with Qt5, examples, and external tool chain**

This document can be found here:

<https://github.com/EliArad/MyDocuments/blob/master/Linux/Buildroot/Prepare_buildroot_image_with_Qt5_examples_and_external_toolchain.odt> or docx

In this document I will show how to setup buildroot with external compiler that was built for the raspberry pi 3 with crosstool-ng.

Add QT plus examples and run it on my raspberry pi 3.

I will use Linux frame buffer first, with the 64 bit version of the raspberry pi 3 def config in buildroot.

Buildroot has three options to use toolchain

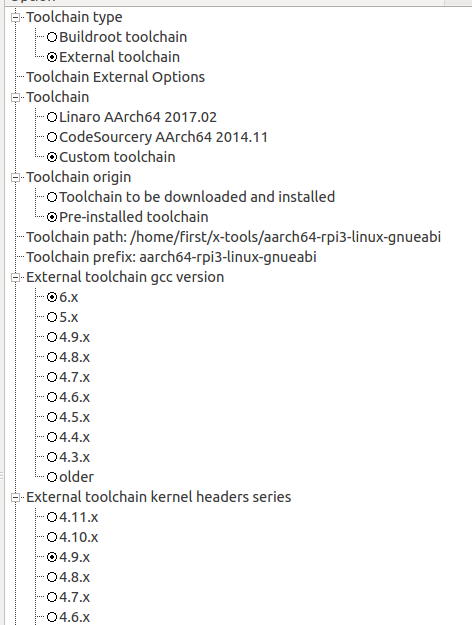
1. internal – download and compile

2. external – download and install – no compile

3. External – custom – use pre toolchain

The first options can take an hour or more depend on your computer just to prepare the tool chain

We want to avoid this option



The external download and install is a very good options, but here we want to use a toolchain that we configured using crosstool-ng.

crosstool-ng has already a raspberry pi 3 configuration.

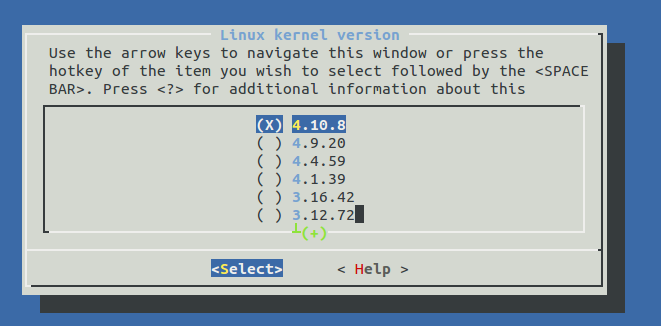
**ct-ng aarch64-rpi3-linux-gnueabi**

The only thing I changed is the kernel version, from 4.10 to 4.9

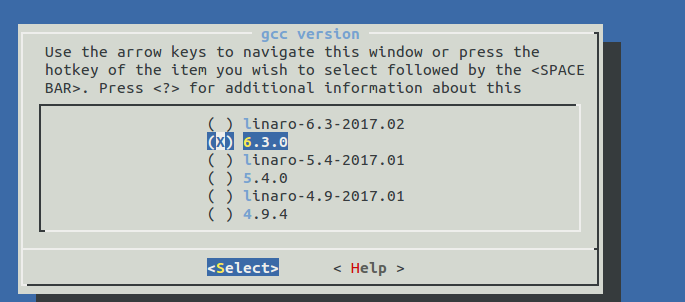
Due the fact that raspberry pi 3 configuration form buildroot uses kernel 4.9

When I selected 4.9 as it was I got kernel panic – init not syncing. ( don’t know If its actual problem)

But it is better to match the kernel headers from the toolchain and the buildroot.



You should match the gcc version



the toolchain path is a full path to the toolchain without the bin directory

the prefix is the name of the compiler without the – or -g++

There are tools that prefix is with the – and here it is without.

**QT5 buildroot setup**

[**http://doc.qt.io/qt-5/embedded-linux.html**](http://doc.qt.io/qt-5/embedded-linux.html)

There are several plug-ins that can be used with QT

\* Linux frame buffer

\* EGLFS

\* Direct FB

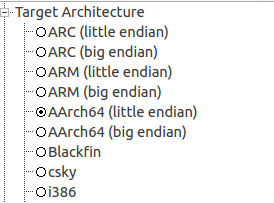
\* Wayland

In this trial I will use Linux fb, qt of course , qt examples.

I am using the 64 bit version in this build

I started from

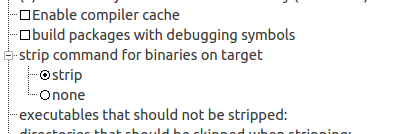
make raspberrypi3\_64\_defconfig



I changed the download directory and savedefconfig

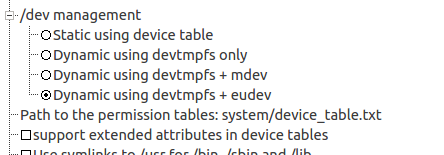


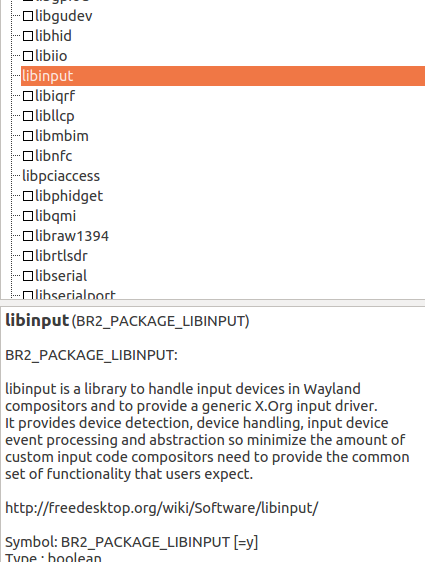
I did not enabled compile cache, if things does not work, try make clean and clean again.



Im am using the external pre build compiler from croos tool ng

In system configuration I needed to enable eudev in order to get libinput support

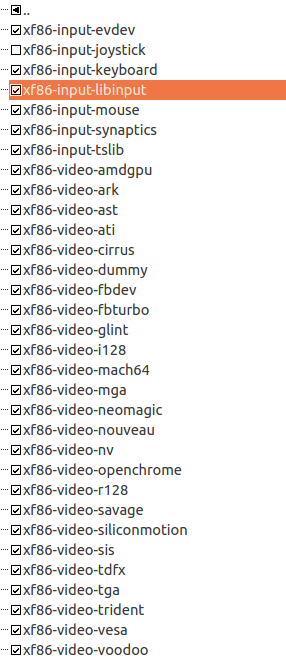




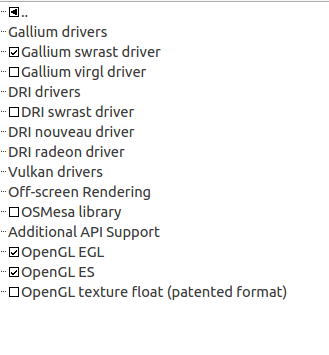
I enabled all the drivers in x.org

actually we need xf86-input-libinput and the video fbdev

but who care for now.



Enable mesa software drivers:



Just a note, the open-GL VC4 driver from Broadcom will not appear in the 64 bit version

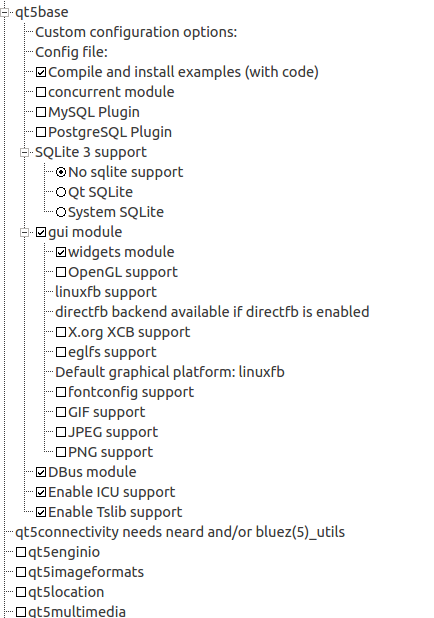
\* I don't know yet , need to check if it just a build root setting or the driver does not supports 64 bit

For now we want to work with the Linux frame buffer and not open gl.

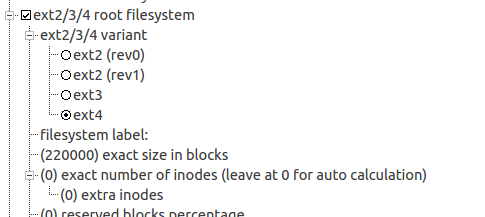
Qt5:

enable Linux frame buffer support

also add examples , it will helps us to quickly check if QT works or not.

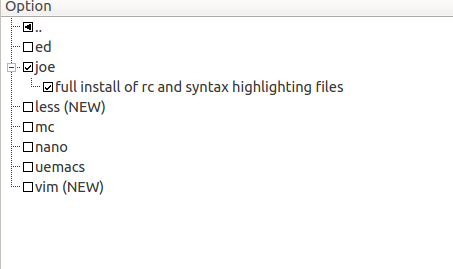


We will need to increase the default block size for the root file system:



You can determine the actual size with block to giga and giga to block ( search in Google)

always include Joe editor – my favorite.



Exit x config and save.

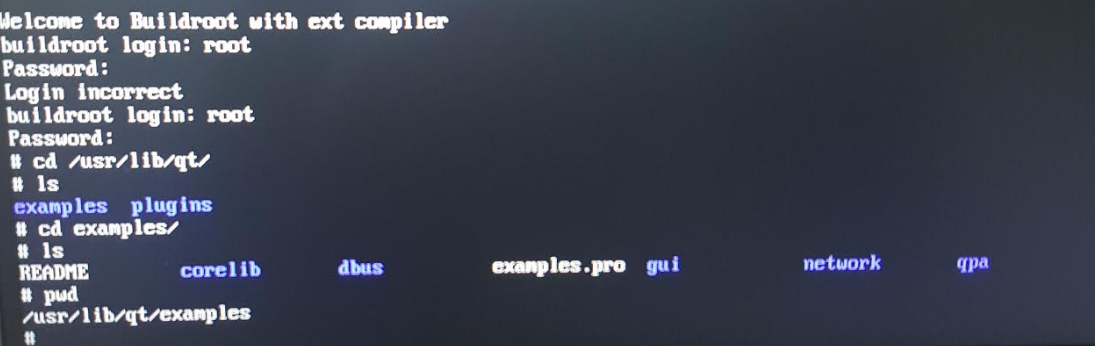
Run

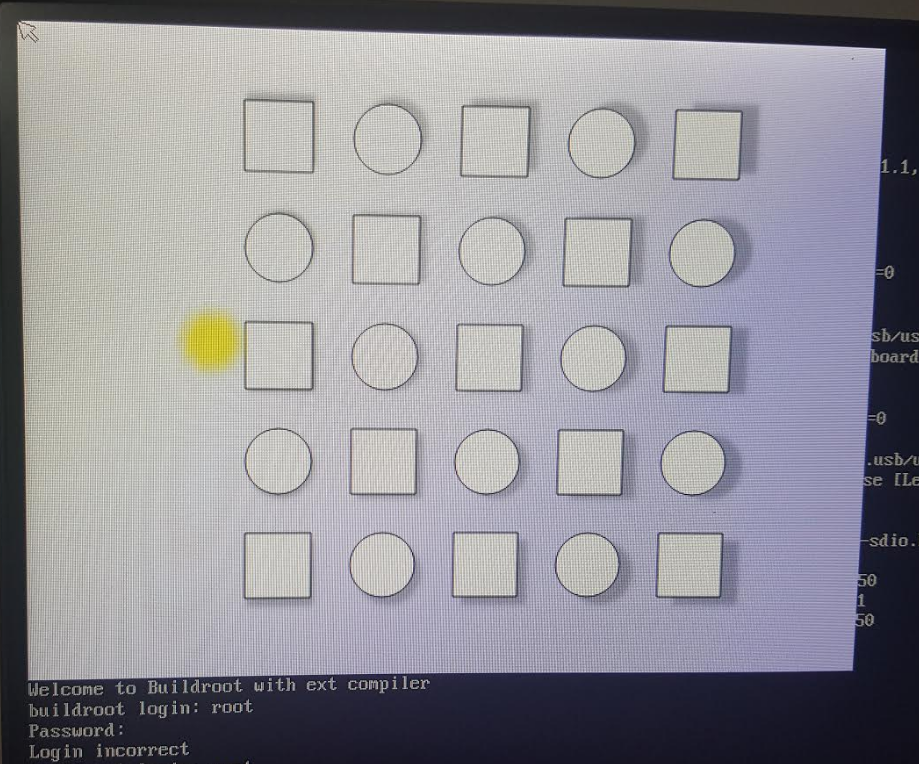
make

The QT5 compilation takes lots of time.

Boot the raspberry pi 3 , and enter the password if you did

You should see the example directory in usr/lib/qt/examples





Notes:

1. I am flashing the sdcard.img with Etcher



Etcher already make partitions according to the image , format the sd card and make sure you dont select your main /dev/sda hard drive.

2. the example that was compiled are located in the target in

/usr/lib/qt/examples

3. in the target , the plugins are installed at:

/usr/lib/qt/plugins/platforms/

4. Always important to add dropbear SSH server

Once added, it will run automatically

5. Next I will build a 32 bit and try the open GL VC4 driver.

6. Hope it helped, it took me some time to gather all this information.