**Dragon Board bringup using Linux Kernel**

Download the Linaro GCC cross Compiler:

* <http://releases.linaro.org/components/toolchain/binaries/6.3-2017.02/aarch64-linux-gnu/gcc-linaro-6.3.1-2017.02-x86_64_aarch64-linux-gnu.tar.xz>
* Extract it.

###### Skales tool

Skales will be used later when creating the device tree.

* sudo apt-get install libfdt-dev
* git clone git://codeaurora.org/quic/kernel/skales
* export PATH=$PATH:/skales

#### Export path to cross compiler tool and confirm version

Exporting path will allow build system can find and use the right kernel .

#Create path

* export PATH=gcc-linaro-6.3.1-2017.02-x86\_64\_aarch64-linux-gnu/bin/:$PATH
* aarch64-linux-gnu-gcc --version

The program 'aarch64-linux-gnu-gcc' is currently not installed. You can install it by typing:

sudo apt-get install gcc-aarch64-linux-gnu

#Check version

* aarch64-linux-gnu-gcc –version

aarch64-linux-gnu-gcc (Linaro GCC 6.3-2017.02) 6.3.1 20170109

Copyright (C) 2016 Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There is NO

warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

#### *How to get and customize the kernel source code:*

* git clone -n <http://git.linaro.org/landing-teams/working/qualcomm/kernel.git>
* cd kernel
* git checkout -b kernel-17.09 debian-qcom-dragonboard410c-17.09

#### Set the right Kernel .config file

* This step creates the '.config' file
* The .config file is used by the build system when compiling the kernel
* Current Reference Platform config can be made by using distro.config

From with in kernel directory execute the following command:

* make ARCH=arm64 CROSS\_COMPILE=aarch64-linux-gnu- defconfig distro.config
* New .config file will be hidden but can be seen by executing ls -a from within kernel folder.
* To view all current configuration the .config file can be opened with a text editor such a vim .

#### Build Kernel Image

#Replace X from -jX with the number of cores on your host computer

* make ARCH=arm64 CROSS\_COMPILE=aarch64-linux-gnu- -jX

Compile the Kernel Modules

###make -jX modules KERNELRELEASE=4.9.56-linaro-lt-qcom ##

* make ARCH=arm64 CROSS\_COMPILE=aarch64-linux-gnu- modules\_install KERNELRELEASE=4.9.56-linaro-lt-qcom INSTALL\_MOD\_PATH=<Path in your system>/kernel/modules INSTALL\_MOD\_STRIP=1

#### Find kernel release string

* This was created during the kernel build.
* cat include/config/kernel.release

#Output

4.9.56

#### Generate modules.dep and map files

* Helps kernel find modules when system boots
* depmod -a -b <Path in your system>/kernel/modules 4.9.56

#### Create the device tree image

* The boot image consists of the table of device tree (dt.img), the kernel image (Image.gz) and an init ramdisk image.
* The dtbTool is a standalone application that will process the DTBs generated during the kernel build, to create the table of device tree image. This tool is included in the skales git tree.
* /skales/dtbTool -o dt.img -s 2048 <Path in your system>/kernel/arch/arm64/boot/dts/qcom.

Create the Ramdisk Image

* wget http://builds.96boards.org/releases/dragonboard410c/linaro/debian/17.09/initrd.img-4.9.56-linaro-lt-qcom

Create the Boot Image

* The tool *mkbootimg* is a standalone application that will process all files and create the boot image that can then be booted on the target board, or flash into the on-board eMMC.
* The boot image also contains the kernel bootargs, which can be changed as needed.
* ./skales/mkbootimg --kernel <Path in your system>/kernel/arch/arm64/boot/Image.gz --ramdisk initrd.img-4.9.56-linaro-lt-qcom --output boot-DB.img --dt dt.img --pagesize 2048 --base 0x80000000 --cmdline "root=/dev/disk/by-partlabel/rootfs rw rootwait console=ttyMSM0,115200n8"
* **sync**

**SD card Boot**

* Download prebuild Debian image from below link:
* <https://builds.96boards.org/releases/dragonboard410c/linaro/debian/latest/dragonboard410c_sdcard_install_debian-283.zip>
* Unzip it. Insert SD card into your host and format it, then type:
* sudo dd if=<PATH of your image file> of=<PATH of your SD card.Ex:/dev/sdb>
* **sync**
* Insert SD card into Dragon board and boot it.
* Once it was booted, power off dragon board and remove SD card and insert it into your Host.
* Now you will observe two mount points as **OS** and **rootfs** .
* *Just for observation*: Go to **OS** --> debian . There you will see *boot.img* .
* Replace that *boot.img* with your *boot-DB.img* .
* sudo cp <PATH of boot-DB.img> </media/USER\_NAME of ur system/OS/debian/boot.img>
* **sync**
* Remove SD card , put it into Dragon board and it will successfully boot up.