Linaro Releases

Name	Last modified	Size	
Parent Directory			
MD5SUMS.txt	13-Jan-2018 09:44	1.4K	open
boot-qcom-apq8064-cm-qs600-qcom-snapdragon-20160224-38.img.gz	27-Feb-2018 01:18	8.7M	open
boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img.gz	27-Feb-2018 01:18	8.7M	open
boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-38.img.gz	27-Feb-2018 01:18	8.7M	open
bootimg.cfg	13-Jan-2018 09:44	290	open
firmware-qcom-snapdragon-20160224-38.img.gz	13-Jan-2018 09:44	64.0K	open
initrd.img-4.4.0-linaro-lt-qcom	13-Jan-2018 09:44	3.1M	open
kernel.config	13-Jan-2018 09:44	113.5K	open
linaro-jessie-alip-qcom-snapdragon-20160224-38.img.gz	27-Feb-2018 01:18	633.1M	open
linaro-jessie-developer-qcom-snapdragon-20160224-38.img.gz	27-Feb-2018 01:19	363.8M	open
linux-firmware-image-4.4.0-linaro-lt-qcom_4.4.0-linaro-lt-qcom-2_armhf.deb	13-Jan-2018 09:44	51.4K	open
linux-headers-4.4.0-linaro-lt-qcom_4.4.0-linaro-lt-qcom-2_armhf.deb	13-Jan-2018 09:44	6.8M	open
linux-image-4.4.0-linaro-lt-qcom-dbg_4.4.0-linaro-lt-qcom-2_armhf.deb	27-Feb-2018 01:19	41.8M	open
linux-image-4.4.0-linaro-lt-qcom_4.4.0-linaro-lt-qcom-2_armhf.deb	13-Jan-2018 09:45	7.4M	open
linux-libc-dev_4.4.0-linaro-lt-qcom-2_armhf.deb	13-Jan-2018 09:45	782.2K	open
qcom-apq8064-cm-qs600.dtb	13-Jan-2018 09:45	31.1K	open
qcom-apq8064-eI_ERAGON600.dtb	13-Jan-2018 09:45	33.5K	open
qcom-apq8064-ifc6410.dtb	13-Jan-2018 09:45	33.5K	open
zImage	13-Jan-2018 09:45	5.6M	open

Release Notes

The Linaro Qualcomm Landing Team is pleased to announce the new release of the Linaro Linux release for Qualcomm™ Snapdragon® 600 processor. The Linaro Linux release 16.02 is an Debian-based Linaro Member Build that provides developers with a desktop like environment using Debian and the LXDE desktop, as well as a console-only image.

What's new in this release

- Upgrade to Debian 8.3.
- Upgrade to Linux kernel 4.4.
- Upgrade graphics components: Mesa 11.1.2, Xserver 1.17.3, Freedreno X11 video driver 1.4.0.
- Add support to auto resize the rootfs image during the first boot.
- Add support for HDMI audio

Features

The Linaro Linux version 16.02 for the Snapdragon 600 supports the following features:

- Provides a working Debian environment with access to Debian repositories (apt-get) and updates. It is based on Debian 8.0 (aka jessie).
- It is based on proprietary firmware from Inforce IFC6410 Android BSP Rel v1.5 for IFC6410 (or IFC6410PLUS) and eInfochips-ERAGON600-Ubuntu-Linux-BSP-Relv1.0 for ERAGON600.
- The following prebuilt images are released:
  - `boot` image that includes prebuilt kernel and initrd
  - `developer` image that includes Debian core packages as well as typical development packages
  - `alip` image that includes a desktop based on LXDE
- All images have a pre-configured user called `linaro`, and the password for this user is set to `linaro`
- The root file system can be installed on eMMC or any other external media such as USB, SD card, SATA.
- Support for the following Snapdragon 600 based boards:
  - Inforce IFC6410 Rev P1 and A1
  - Inforce IFC6410PLUS
  - Compulab Utilite2 (CM-QS600)
  - eInfochips ERAGON600 Development Board (based on ERAGON600 SOM)
- The following Snapdragon 600 features are supported:
  - Quad Core Krait 300 CPU (up to 1.7GHz)
  - Adreno 320 GPU, powered by `freedreno` Mesa/Gallium GPU driver
    - OpenGL 3.1, OpenGLES 3.0, GLX, EGL
    - xf86-video-freedreno driver v1.4.0, with XA support enabled
    - modesetting with GLAMOR support
  - Cpufreq, using ondemand governor by default
  - HDMI display and audio
  - UART, Ethernet, SD, eMMC, SATA
  - USB2.0 (Mouse, Keyboard, Storage)
  - Wifi using on-board QCA6234x
  - Bluetooth using on-board QCA6234x
  - CPU thermal sensors

Information about the IFC6410 and IFC6410PLUS boards

For more information about Linaro Linux release for Snapdragon 600 processor and the Inforce IFC6410 board, please check the Linaro wiki. This wiki page provides additional information on board setup and getting started with Debian on the IFC6410 (e.g. customize the kernel, rebuild the images, …).

The IFC6410 is no longer available for purchase and has been replaced by the Inforce IFC6410PLUS. For more information about this board, please check this website. The instructions in the release notes are the same for both the IFC6410 and the IFC6410PLUS.

Information about the Compulab Utilite2 Board

For more information about Compulab Utilite2 board, please check Compulab website.

Information about the eInfochips ERAGON600 Board

For more information about eInfochips ERAGON600 board, please check eInfochips website.

Download the release binaries

To install this release on a Snapdragon 600 based board, you need to download the following files:

```
firmware-qcom-snapdragon-20160224-38.img.gz
linaro-jessie-developer-qcom-snapdragon-20160224-38.img.gz
linaro-jessie-alip-qcom-snapdragon-20160224-38.img.gz
```

You also need to download the proper boot partition, based on the actual board you are using.

For the Inforce IFC6410 or IFC6410PLUS, please download:

```
boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-38.img.gz
```

For the Compulab Utilite2, please download:

```
boot-qcom-apq8064-cm-qs600-qcom-snapdragon-20160224-38.img.gz
```

For the eInfochips ERAGON600, please download:

```
boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img.gz
```

The build id is in the form: `TIMESTAMP-BUILD_NUMBER`.

All images downloaded from Linaro are compressed files and need to be uncompressed first:

```
gunzip *.img.gz
```

Upgrading an already installed version of Linaro Linux release for Snapdragon 600 processor

If you are already running on your board a previous version of the Linaro Linux release for Snapdragon 600 processor, you can upgrade your current installation by following the instructions in this section.

In order to upgrade an existing release, you must ensure that you are running a previous release that is based on Debian 8.0 (aka jessie), e.g. the same Debian version that this release supports. If you are unsure about which version of Debian you are running, you can check the corresponding Linaro Linux release notes, or you can run the following command on your board:

```
lsb_release -a
```

For now the Linaro Linux release does not support Debian distro version upgrade, when that happens you need to re-install the release completely, using the instructions in the rest of these release notes. If you are running the proper version of Debian, you can proceed with the upgrade.

To upgrade your current installation, three main components need to be updated:

- the boot partition (e.g. kernel)
- the root file system partition (e.g. user space)
- the firmware partition

In this release (e.g. Linaro 16.02), there are no firmware changes. If you have already installed the firmware partition on your board (for example if you have flashed the image into the cache partition) then you do not have anything else to do. You can ignore the section below called Managing the proprietary firmware. We will simply reuse the cache partition as it is.

To upgrade the root file system, on the board running the Linaro Linux release, please run the following commands:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

These commands will check for all possible updates, and install them all.

We now need to install the new kernel modules. You need to download on the board, the kernel debian package file from the release. This file is called:

```
linux-image-4.4.0-linaro-qcom_4.4.0-linaro-qcom-1_armhf.deb
```

On the board running the Linaro Linux release, you now need to install this debian package with:

```
sudo dpkg -i linux-image-4.4.0-linaro-qcom_4.4.0-linaro-qcom-1_armhf.deb
```

The final step until to complete the upgrade is to boot the new kernel. That requires to reboot the board into fastboot mode. You can use the standard instructions from next sections, which you have used when you installed the Linaro Linux release on your board. Of course you need to make sure to connect your board to the development host (USB for fastboot, UART for the serial console), and ensure that it is booted in fastboot mode, waiting for commands, then you can flash the images
on Inforce IFC6410 (or IFC6410PLUS) with:

```
sudo fastboot flash boot boot-qcom-apq8064-ifc6410-qcom-
snapdragon-20160224-38.img
```

on Compulab Utilite2 with:

```
sudo fastboot flash boot boot-qcom-apq8064-cm-qs600-qcom-
snapdragon-20160224-38.img
```

on eInfochips ERAGON600 with:

```
sudo fastboot flash boot boot-qcom-apq8064-eI_ERAGON600-qcom-
snapdragon-20160224-38.img
```

Your system is now up-to-date, and you are running Linaro 16.02 on your development board. If you have followed all the instructions above, and you are getting any error, please report them as bugs (see Support section below).

Managing the proprietary firmware

To properly use this release, a set of proprietary firmware blobs must be acquired separately from respective board vendors. For IFC6410 (or IFC6410PLUS) firmware, use Inforce Techweb website and for eInfochips ERAGON600 firmware, use eInfochips support centre as Linaro is not redistributing them in this release.
These firmware blobs are needed for audio, power management, video and graphics support. While it is possible to boot the release without the firmware installed, it is not recommended, as various errors can occur.

The Linaro release does not contain any proprietary software and thus provides an empty firmware image. The provided firmware image is only a placeholder for users to install the firmware blobs downloaded from respective board vendor website. The empty firmware image can be flashed as per the instructions in the next sections; of course some functionality will not work without the proprietary firmware blobs. To install the firmware and get all the features of the release, you should extract the firmware from the respective BSP release, install them in the firmware image, and flash the updated firmware image as per the instructions in the next sections.

For Inforce IFC6410 (or IFC6410PLUS)

For bootloader, kernel and firmware, this release is based on Inforce IFC6410 Android BSP Rel v1.5, which you need to download from Inforce Techweb. The ZIP file `Inforce-IFC6410_AndroidBSP_880160_Rel_Beta_V1.5.zip` includes all the proprietary firmware blobs required to use this Debian-based Linaro build.

To install the firmware blobs, first locate the file `proprietary.tar.gz` in the `source` folder after extracting the Inforce BSP release ZIP file. Then follow the next instructions on how to loop mount the empty firmware image included in the release and inject Inforce’s proprietary firmware blobs.

```
mkdir image
sudo mount -o loop firmware-qcom-snapdragon-20160224-38.img image
sudo tar xzf proprietary.tar.gz -C image/ --strip-components 8 proprietary/prebuilt/target/product/msm8960/system/etc/firmware/
sudo umount image
rmdir image
```

The firmware image `firmware-ifc640-20160224-38` has been updated, and now contains all the relevant proprietary firmware blobs.

For eInfochips ERAGON600

For bootloader, kernel and firmware, this release is based on eInfochips-ERAGON600-Ubuntu-Linux-BSP-Relv1.0, which you need to download from eInfochips support centre (Coming Soon). The ZIP file `firmware-eragon600-v1.0.img` includes all the proprietary firmware blobs required to use this Debian-based Linaro build.

To install the firmware blobs, first locate the file `firmware-eragon600-v1.0.img.gz` in the `binaries` folder after extracting the eInfochips ERAGON600 BSP release ZIP file. Then follow the next instructions to extract the firmware image.

```
gunzip  firmware-eragon600-v1.0.img.gz
```

The firmware image `firmware-eragon600-v1.0.img` contains all the relevant proprietary firmware blobs.

Installing the LXDE based image

Connect your board to the development host (USB for fastboot, UART for the serial console), and make sure that it is booted in fastboot mode, and waiting for commands, then you can flash the images:

for Inforce IFC6410 (or IFC6410PLUS)

```
sudo fastboot flash boot boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-38.img
sudo fastboot flash cache firmware-qcom-snapdragon-20160224-38.img
```

for Compulab Utilite2

```
sudo fastboot flash boot boot-qcom-apq8064-cm-qs600-qcom-snapdragon-20160224-38.img
sudo fastboot flash cache firmware-qcom-snapdragon-20160224-38.img
```

for eInfochips ERAGON600

```
sudo fastboot flash boot boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img
sudo fastboot flash cache firmware-eragon600-v1.0.img
```

then:

```
sudo fastboot flash -S 768M userdata linaro-jessie-alip-qcom-snapdragon-20160224-38.img
```

Note:

- fastboot should be able to split large files automatically, but it does not work. As such, it is mandatory to add `-S 768M`, when flashing files larger than 768MB as a workaround.
- the root file system must be flashed in the `userdata` partition, and the boot image must be flashed in the `boot` partition, otherwise the system will not boot.
- the firmware image can be flashed in any partition larger than 64MB, it is recommended to use `cache` partition, but not strictly required.

Flashing the LXDE image takes several minutes. Be patient. You should see the following fastboot traces on the PC while flashing the image:

```
sudo fastboot flash -S 768M userdata linaro-jessie-alip-qcom-snapdragon-20160224-38.img
sending sparse 'userdata' (785202 KB)...
OKAY [ 55.368s]
writing 'userdata'...
OKAY [121.134s]
sending sparse 'userdata' (418516 KB)...
OKAY [ 36.508s]
writing 'userdata'...
OKAY [110.017s]
finished. total time: 323.027s
```

And you should see the following traces on the serial console in the meantime:

```
[34320] fastboot: getvar:partition-type:userdata
[34510] fastboot: download:2fecc8b4
[89870] fastboot: flash:userdata
[210990] Wrote 524288 blocks, expected to write 524288 blocks
[211020] fastboot: download:198b50a4
[247510] fastboot: flash:userdata
[357520] Wrote 524288 blocks, expected to write 524288 blocks
```

Do not power off the PC, or the board while flashing.

Running the LXDE based image

The LXDE based image is expected to provide a desktop-like experience, as such it is recommended to use an HDMI monitor, as well as USB keyboard and mouse.

The default bootargs embedded in the released boot image will intentionally prevent the graphical login manager (lightdm) to start, since trying to start graphical environment without the proprietary firmware installed will lead to various boot errors. To prevent lightdm from starting, the keyword `systemd.unit=multi-user.target` was added into the bootargs. If you have properly installed and flashed into eMMC the proprietary firmware blobs and want to use the graphical interface, you can either start the login manager from the serial console, or alternatively you can change the bootargs.

To boot to the serial console and get a root prompt, you can simply run:

```
sudo fastboot continue
```

Note: The default bootargs enable the kernel messages to be displayed on the serial console.

Once you have a root prompt on the serial console, you have a functional Debian system. If you want to start the login manager to see the graphical interface on the HDMI monitor, from the serial console simply run

```
systemctl start ligthdm
```

The login manager window should appear within a few seconds, and you can log with the user `linaro`, using the password `linaro`. If you want to boot directly to the login manager, you can boot the image with different bootargs (e.g. remove the keyword `systemd.unit=multi-user.target`).

For Inforce IFC6410 (or IFC6410PLUS), you can run the following command:

```
sudo fastboot boot boot-qcom-apq8064-ifc6410-qcom-snapdragon-
20160224-38.img -c "console=ttyMSM0,115200n8 root=/dev/disk/by-
partlabel/userdata rootwait rw"
```

For eInfochips ERAGON600, you can run the following command:

```
sudo fastboot boot boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img -c "console=ttyMSM0,115200n8 root=/dev/disk/by-partlabel/userdata rootwait rw"
```

Finally, if you want to make the bootargs change persistent, you can modify the released boot image and reflash it into the `boot` partition:

for Infoce IFC6410 (or IFC6410PLUS)

```
abootimg -u boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-38.img -c "cmdline=console=ttyMSM0,115200n8 root=/dev/disk/by-partlabel/userdata rootwait rw"
sudo fastboot flash boot boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-38.img
```

for eInfochips ERAGON600

```
abootimg -u boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img -c "cmdline=console=ttyMSM0,115200n8 root=/dev/disk/by-partlabel/userdata rootwait rw"
sudo fastboot flash boot boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img
```

Note: `abootimg` is a tool to read/write/update Android boot partition, and is available in most standard Linux distribution; simply install the package using your Linux distribution methods. If you are using Debian or Ubuntu on your PC, simply run:

```
sudo apt-get install abootimg
```

Installing the Developer based image

Connect the board to the development host (USB for fastboot, UART for the serial console), and make sure that it is booted in fastboot mode, and waiting for commands, then you can flash the images:

for Inforce IFC6410 (or IFC6410PLUS)

```
sudo fastboot flash boot boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-38.img
sudo fastboot flash cache firmware-qcom-snapdragon-20160224-38.img
```

for Compulab Utilite2

```
sudo fastboot flash boot boot-qcom-apq8064-cm-qs600-qcom-snapdragon-20160224-38.img
sudo fastboot flash cache firmware-qcom-snapdragon-20160224-38.img
```

for eInfochips ERAGON600

```
sudo fastboot flash boot boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img
sudo fastboot flash cache firmware-eragon600-v1.0.img
```

then:

```
sudo fastboot flash -S 768M userdata linaro-jessie-developer-qcom-snapdragon-20160224-38.img
```

Note:

- the root file system must be flashed in the `userdata` partition, and the boot image must be flashed in the `boot` partition, otherwise the system will not boot.
- the firmware image can be flashed in any partition larger than 64MB, it is recommended to use `cache` partition, but not strictly required.

Running the Developer based image

To boot to the serial console and get a root prompt, you can simply run:

```
sudo fastboot continue
```

Note: The default bootargs enable the kernel messages to be displayed on the serial console.

Once you have a root prompt on the serial console, you have a functional Debian system.

Configuring and using Bluetooth and WLAN

Firmware are required for Bluetooth and WLAN to work properly. If you have followed the instructions so far, the firmware have been installed. For both WLAN and Bluetooth appropriate mainline drivers are used, respectively ath6kl and ar3k.

WLAN should be up and running out of the box, using the releases images. If you are building your own kernel, please be aware that the WLAN driver and some mac802.11 drivers are compiled as modules, and need to be deployed into the root file system. WLAN can be configured either on the command line, or using the graphical desktop applet once the desktop is started.

For Bluetooth, it is for now not enabled by default in the image, and you need to run the following command on the root prompt to properly start and configure the Bluetooth chip:

```
hciattach -s 115200 /dev/ttyMSM1 ath3k 3000000
```

And this command will return, if everything is working properly:

```
Device setup complete
```

Finally to make sure the Bluetooth chip is configured correctly, you can run the `hciconfig` tool, which should return something along these lines:

```
hciconfig
hci0:   Type: BR/EDR  Bus: UART
BD Address: 00:3C:7F:F0:F0:0A  ACL MTU: 1021:8  SCO MTU: 124:0
UP RUNNING PSCAN
RX bytes:558 acl:0 sco:0 events:27 errors:0
TX bytes:866 acl:0 sco:0 commands:27 errors:0
```

You can also scan for Bluetooth devices nearby, from the command line, e.g.:

```
hcitool scan
Scanning ...
        BC:5F:BC:44:FB:AC       Nexus 5
```

If you configure the Bluetooth chip before starting the graphical desktop, then you should be able to use the Desktop Bluetooth application/applet to connect to any device and use Bluetooth functions.

Note that if you are using the developer image you might need to install the Bluez applications and librairies:

```
sudo apt-get update
sudo apt-get install bluez
```

How to get and customize the kernel source code

The Linux kernel used in this release is available via tags in the Linaro Qualcomm Landing Team git repository:

```
git: http://git.linaro.org/landing-
teams/working/qualcomm/kernel.git
tag: debian-qcom-snapdragon-16.02
defconfig: qcom_defconfig kernel/configs/distro.config
```

The kernel compresssed image (`zImage`) is located in the `boot` image and partition and the kernel modules are installed in the root file system. It is possible for a user to rebuild the kernel and run a custom kernel image instead of the released kernel. You can build the kernel using any recent GCC release using the git tree, tag and defconfig mentioned above. This release only supports booting with device tree, as such both the device tree blobs need to be built as well. DTBs files need to be appended to the kernel image when the booltloader do not support device tree protocol, such as on the IFC6410 (or IFC6410PLUS) and the Utilite2.

Even though it is possible to build natively, on the target board, It is recommended to build the Linux kernel on a PC development host. In which case you need to install a cross compiler for the ARM architecture. It is recommended to download the Linaro GCC cross compiler.

To build the Linux kernel, you can use the following instructions:

```
git clone -n http://git.linaro.org/landing-
teams/working/qualcomm/kernel.git
cd kernel
git checkout -b kernel-16.02 debian-qcom-snapdragon-16.02
export ARCH=arm
export CROSS_COMPILE=<path to your GCC cross compiler>/arm-
linux-gnueabihf-
make qcom_defconfig distro.config
make -j4 zImage dtbs
```

The DTB file needs to be appended to the `zImage`, as such, you can run the following command to prepare an image

for the IFC6410 (or IFC6410PLUS):

```
cat arch/arm/boot/zImage arch/arm/boot/dts/qcom-apq8064-
ifc6410.dtb  > zImage-dtb
```

for Compulab Utilite2:

```
cat arch/arm/boot/zImage arch/arm/boot/dts/qcom-apq8064-cm-
qs600.dtb  > zImage-dtb
```

for eInfochips ERAGON600:

```
cat arch/arm/boot/zImage arch/arm/boot/dts/qcom-apq8064-
eI_ERAGON600.dtb  > zImage-dtb
```

To boot a custom kernel image, you can run the following `fastboot` command:

```
sudo fastboot boot -c "console=ttyMSM0,115200,n8
root=/dev/disk/by-partlabel/userdata rootwait rw
systemd.unit=multi-user.target" -b 0x80200000 zImage-dtb
```

If you have followed these instructions, the root file system was flashed into the `userdata` partition. The `-c` arguments represent the `bootargs` that will be passed to the kernel during the boot.

If you want to permanently use a custom kernel image, you can update the boot image and reflash it into the `boot` partition, for example,

for IFC6410 (or IFC6410PLUS):

```
abootimg -u boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-
38.img -k zImage
sudo fastboot flash boot boot-qcom-apq8064-ifc6410-qcom-
snapdragon-20160224-38.img
```

for ERAGON600:

```
abootimg -u boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img -k zImage
sudo fastboot flash boot boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-20160224-38.img
```

How to get and customize Debian packages source code

This release is based on Debian 8.0 (aka jessie), and it is not possible to use a different Debian release (e.g. it is not possible to downgrade to an older Debian release, nor is it possible to use a newer release, such as the one being currently developed).

Since all packages installed in Linaro Debian-based images are maintained either in Debian archives or in Linaro repositories, it is possible for users to update their environment with commands such as:

```
sudo apt-get update
sudo apt-get upgrade
```

All user space software is packaged using Ubuntu/Debian packaging process. As such you can find extensive information about using, patching and building packages in the Ubuntu packaging guide or The Debian New Maintainers Guide. If you quickly want to rebuild any package, you can run the following commands to fetch the package source code and install all build dependencies:

```
sudo apt-get update
sudo apt-get build-dep <pkg>
apt-get source <pkg>
```

Then you can rebuild the package locally with:

```
cd <pkg-version>
dpkg-buildpackage -b -us -uc
```

Notes:

- you can drop patches in debian/patches/ and update debian/patches/series before building with dpkg-buildpackage to customize the source code.
- all associated .deb files will be located in the root folder, and can be installed in the system with dpkg -i .deb.
- all these commands should be executed on the target directly, not on the development host. It is generally enough to build packages natively, on the target platform. For some packages, it is possible to cross compile Ubuntu/Debian packages however this goes beyond the scope of this wiki page.

How to install the root filesystem on external storage

The root file system can be installed on SD card, external USB drive or external (m)SATA drive instead of being flashed to the eMMC. The method is to extract the content of the root file system image from the release, and copy it onto a proper partition on an SD card or USB drive, then boot the target platform with the proper `root=` argument in the bootargs. The root file system image file is using an Android sparse image format to optimize the file size. To convert a sparse image into a mountable raw image file, and mount it on your host, run:

```
simg2img linaro-jessie-alip-qcom-snapdragon-20160224-38.img linaro-jessie-alip-qcom-snapdragon-20160224-38.img.raw
mkdir rootfs
sudo mount -o loop linaro-jessie-alip-qcom-snapdragon-20160224-38.img.raw rootfs
```

Note that `simg2img` is available in most standard Linux distribution; simply install the package using your Linux distribution methods. If you are using Ubuntu or Debian on your PC, simply run `sudo apt-get install android-tools-fsutils`. Assuming the SD card or USB drive that should contain the root file system is properly formatted as ext4 partition and mounted on your host as `/mnt/disk`, you can copy the content of the image with:

```
cd rootfs
sudo cp -a * /mnt/disk
cd ..
sudo umount rootfs
```

It is very important to run the copy with admin privileges and with `-a` in order to preserve files ownership and permission.

The SD card or USB drive is now ready to be plugged and used on the target. For USB drive it is recommended to use externally powered USB drive or through an externally powered USB HUB. When booting with the root file system on USB drive or SD card, the `boot` partition from the eMMC is still being used for the kernel and initrd. As such you need to update the `boot` image with the new bootargs, and reflash it. If using a USB or (m)SATA drive or SD card with a single partition, the device to boot from will be either `/dev/sda1` for USB drive or `/dev/mmcblk1p1` for SD card. For example, for USB drive, run:

for Inforce IFC6410 (or IFC6410PLUS)

```
abootimg -u boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-38.img -c "cmdline=console=ttyMSM0,115200n8 root=/dev/sda1 rootwait rw systemd.unit=multi-user.target"
sudo fastboot flash boot boot-qcom-apq8064-ifc6410-qcom-snapdragon-20160224-38.img
```

for eInfochips ERAGON600

```
abootimg -u boot-qcom-apq8064-eI_ERAGON600-qcom-snapdragon-
20160224-38.img -c "cmdline=console=ttyMSM0,115200n8
root=/dev/sda1 rootwait rw systemd.unit=multi-user.target"
sudo fastboot flash boot boot-qcom-apq8064-eI_ERAGON600-qcom-
snapdragon-20160224-38.img
```

Using X11 with modesetting video driver and GLAMOR

While not enabled by default, it is possible to use X11/GLAMOR. In order to start X server with the modesetting driver, with GLAMOR support, your first need to make sure that X server is not running:

```
sudo systemctl stop lightdm
```

Then you can run the following commands to switch from freedreno video driver, to modesetting, and restart the LXDE desktop:

```
sudo sed -i 's/freedreno/modesetting/'
/usr/share/X11/xorg.conf.d/42-freedreno.conf
sudo systemctl start lightdm
```

To switch back to freedreno video driver, please run:

```
sudo systemctl stop lightdm
sudo sed -i 's/modesetting/freedreno/'
/usr/share/X11/xorg.conf.d/42-freedreno.conf
sudo systemctl start lightdm
```

Known issues and limitations

- WLAN might not work properly. There is a race condition during the boot, and wlan might not work properly, there are 2 workarounds for now:
  - at boot, before starting the desktop interface, run: `modprobe -r ath6kl_sdio`, then `modprobe ath6kl_sdio`. That needs to be done at each boot
  - for a more permanent workaround, run the following commands:

```
sudo mkdir /mnt
sudo mount /dev/disk/by-partlabel/userdata /mnt
sudo cp -a /lib/firmware/* /mnt/lib/firmware/
sudo umount /mnt
reboot
```

- Some software codecs might not be installed by default; you might install additional codecs such as `gstreamer1.0-plugins-ugly`.
- Once you have flashed the boot partition with a valid boot image, the board will automatically boot that image when powered on. To avoid automatically booting you can jumper the board as explained in this page for IFC6410 (or IFC6410PLUS). To boot the ERAGON600 board in fastboot mode, please refer `section 7.4.2 Fastboot mode` on `Software Reference Manual` at eInfochips support centre. If you cannot use this method and still needs to reboot into fastboot mode, you can erase the boot partition on a running target using the following command `sudo cat /dev/zero > /dev/disk/by-partlabel/boot`

Feedback and Support

For general question or support request, please use:

- Inforce Techweb or Inforce Forums
- Compulab Developer Resources
- Linaro IRC: #linaro on irc.freenode.net
- Linaro mailing list

For any bug related to the Linaro Member Build, please submit issues to the Linaro bug tracking system. To submit a bug, select “File a Bug”, then Product: “Qualcomm LT”, or simply follow this link.

Bugs will be reviewed and prioritized by the team. For any bug report it is recommended to provide as much information as possible, and at the very list please include the name of the release you are using, the output of `dpkg -l` to list all packages installed, as well, as the boot log (output of `dmesg`).

Qualcomm Snapdragon is product of Qualcomm Technologies, Inc.

Running linaro-license-protection 7f962b7.