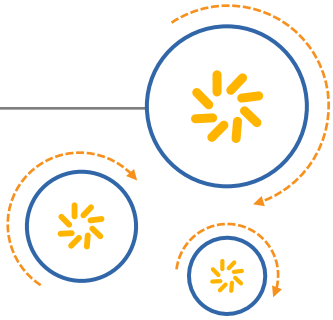
Qualcomm Technologies, Inc.



DragonBoard™ 410c Linux Embedded

Release Notes

LM80-P0337-4 Rev. A

November 9, 2017

For additional information or to submit technical questions, go to: https://www.96boards.org/DragonBoard410c/forum

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	November 2017	Initial release

Contents

1 Introduction 6
1.1 Introduction to www.codeaurora.org 6
1.2 Introduction to DragonBoard 410c 6
1.3 Documentation conventions 6
1.4 Terms and acronyms 6

2 Set up a Linux build environment 7
2.1 Required equipment and software 7
2.1.1 Linux host 7
2.1.2 Install Repo 7
2.1.3 Set up kernel build environment 8

3 Load images 9
3.1 Overview 9
3.1.1 Download the proprietary images 9
3.2 Boot from eMMC (normal boot) 10
3.2.1 Program images using Fastboot 10
3.3 Boot from SD 11

4 Set up a project 11
4.1 GPT. 11
4.1.1 GPT creation 11
4.2 CDT. 12
4.2.1 CDT flashing using Fastboot 12
4.3 Building HLOS 12
4.3.1 Yocto project reference manual 12
4.3.2 Download the source files 12
4.3.3 Set up a bitbake environment 12
4.3.4 Generate build images 13
4.4 Build the kernel 14
4.5 Build LK 15
4.5.1 Sign LK image 15

5 Supported features 15
5.1 GPS enablement 16
5.2 Real time clock usage 16
5.2.1 Connect coin-cell battery for RTC on DragonBoard 410c 16
5.2.2 RTC read and write from user space 17
5.3 Enablement and test examples 17
5.3.1 Enable USB camera 17
5.3.2 Enable CSI camera 17
5.3.3 RDI mode 1080p 18
5.3.4 RTSP 19
5.4 Swap partition 19

6 Troubleshooting 20
6.1 Cannot flash CDT using Fastboot 20
6.2 Device fails to boot 20

7 Known issues and limitations 21
7.1 Limitations.....21
7.2 Known issues.....21
7.3 Resolved issues.....21

Tables

Table 1-1 Acronyms and terms 6
Table 2-1 Required equipment and software 7
Table 4-1 Download the source files 12

1 Introduction

This document describes how to obtain, build, and program the software that is applicable to the Linux embedded software product “as-is” into a reference platform. This document describes the following:

- Set up a development environment and install the software
- Build the software and flash it onto the DragonBoard™ 410c platform

This document provides a description of chipset capabilities. Not all features are available, nor are all features supported in the software.

NOTE: Enabling some features may require additional licensing fees.

1.1 Introduction to www.codeaurora.org

Open source HLOS (high-level operating system) software for Qualcomm® Snapdragon™ chipsets is available on the Linux Foundation hosted site www.codeaurora.org.

1.2 Introduction to DragonBoard 410c

See the following websites for additional information:

https://developer.qualcomm.com/hardware/dragonboard-410c

https://www.96boards.org/DragonBoard410c/docs

1.3 Documentation conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `#include`.

Code variables appear in angle brackets, for example, `<number>`.

Shading indicates content that has been added or changed in this revision of the document.

1.4 Terms and acronyms

Table 1-1 Acronyms and terms

Acronym or Term	Definition
CDT	Configuration data table
EABI	Embedded application binary interface
eMMC	Embedded multimedia card
GCC	GNU Compiler Collection (GCC) tool
GPT	General-purpose timers

Acronym or Term	Definition
HLOS	High-level operating system (generic term for WinMob, Linux, and so on)
INITRD	Initial RAM disk
OE	OpenEmbedded
RPB	Reference plaftorm build

2 Set up a Linux build environment

2.1 Required equipment and software

Table 2-1 Required equipment and software

Serial number	Item description	Version	Source/vendor	Purpose
1	Power supply	x	x	DragonBoard 410c required power of 12 V, 2 A
2	Linux host	x	x	LE build host operating system
3	Repo	x	Google	Source management tool
4	GCC compiler	5.3 and above	Linaro	GCC compiler with ARM 64-bit support
5	ARM-EABI	N/A	Linaro	Compiler tool for little kernel (LK)
6	SKALES tool	N/A	Qualcomm Technologies, Inc. (QTI)	SKALES tool to pack images
7	INITRD image	N/A	Linaro	INITRD image
8	signLK	N/A	QTI	Sign LK images

2.1.1 Linux host

Compilation process for DragonBoard 410c LE requires Linux host machine. Recommended Linux machine for compiling the code is the latest release from Ubuntu.

Create an installation CD and install it onto the computer by following the instructions at http://releases.ubuntu.com/.

2.1.2 Install Repo

The Repo tool is a source code configuration management tool used by the Android project. The Repo tool is a front-end to Git written in Python. It uses a manifest file to help download the code organized as a set of projects that are stored in different Git repositories.

To install Repo, perform the following steps:

1. Create a ~/bin directory in your home directory, or, if you have root or sudo access, install for all system users under a common location, such as /usr/local/bin or somewhere under /opt.

2. Download the Repo script.
   ```
   $ curl https://storage.googleapis.com/git-repo-downloads/repo >
   ~/bin/repo
   ```
3. Set the Repo script attributes to executable.
   ```
   $ chmod a+x ~/bin/repo
   ```
4. Include the installed directory location for Repo in your PATH.
   ```
   $ export PATH=~/bin:$PATH
   ```
5. Run Repo --help to verify the installation.
   ```
   $ repo –help
   ```
   The following message is displayed:

   usage: repo COMMAND [ARGS]

   Repo is not yet installed. Use “repo init” to install it here.

   The most commonly used Repo commands are as follows:

   - `init` → Install Repo in the current working directory
   - `help` → Display detailed help on a command

NOTE: To access to the online help, install Repo (repo init).

2.1.3 Set up kernel build environment

Set up the following tools to compile Linux kernel without rebuilding the complete HLOS.

2.1.3.1 Download SKALES

To pack images, download the SKALES tool from the following link:

- https://source.codeaurora.org/quic/imm/imm/skales/

2.1.3.2 Download GCC compiler

Download the GCC compiler with ARM 64-bit support from Linaro from the following link:

- https://releases.linaro.org/components/toolchain/binaries/latest/aarch64-linux-gnu/

2.1.3.3 Download INITRD image

Download the initial RAM disk (INITRD) image from the following location:

- http://builds.96boards.org/releases/dragonboard410c/linaro/debian/latest/

2.1.3.4 Download ARM-EABI compiler

The ARM-EABI compiler is used to compile the LK file. Use the following command to download the compiler:

```
git clone git://codeaurora.org/platform/prebuilts/gcc/linux-x86/arm/arm-
eabi-4.8.git -b LA.BR.1.1.3.c4-01000-8x16.0
```

2.1.3.5 Download signLK tool

The signLK tool is used to sign the LK files (emmc_appsbl.mbn) before flashing to the device.

NOTE: When the device rejects unsigned LK images, Fastboot is not available. In such scenarios, other alternative methods to update the device images are required.

Download the latest version of signlk from the following website:

Git clone https://git.linaro.org/landing-teams/working/qualcomm/signlk.git

3 Load images

3.1 Overview

A Linux build is references to a set of images that must be loaded to the device for proper functionality.

A full build content contains the following components:

- gpt.bin – See Section 4.1 for more information.
- CDT.bin – See Section 4.2 for more information.
- Sbl1.mbn
- tz.mbn
- hyp.mbn
- rpm.mbn
- NON-HLOS.bin
- emmc_appsboot.mbn (LK)
- boot.img
- rootfs.ext4

The following proprietary images are available only in the binary format and delivered in the BSP as described in the section that follows. The source code for other components is available for the user to download and self-compile.

See Section 4 for more information about self-building the images.

3.1.1 Download the proprietary images

Board support packages are available at the following link:

https://developer.qualcomm.com/hardware/dragonboard-410c/software

It is mandatory that you match the board support package release to the HLOS release.

For example, BSP file ‘linux-board-support-package-r1032.1’ matches HLOS release 1032.1.

3.1.1.1 BSP content

The BSP contains the following:

- All proprietary images:
  - sbl1.mbn
    - sbl1.sd.mbn – Use this file to boot from SD rescue card
  - tz.mbn
  - hyp.mbn
  - rpm.mbn
  - NON-HLOS.bin
- CDT file
- Proprietary firmware for WLAN and video

3.2 Boot from eMMC (normal boot)

3.2.1 Program images using Fastboot

1. To bring the device into the Fastboot mode, perform the following steps:
   a. Hold down the VOL-key.
   b. Connect the DC supply to the DragonBoard 410c.
   c. Press the power ON button.
   d. Plug the USB cable into the target.
2. Depending on your build environment, choose *one* of the following options:
   - Run the following command from the Windows command shell:
     `Fastboot devices`
   - Perform the following from the Linux environment:
     - Navigate to the following directory:
       `cd <$BUILDROOT>build\tmp-glibc\deploy\images\dragonboard-410c`
     - Run the following command:
       `sudo fastboot devices`

   The list of registered devices is displayed.
3. After the device is detected, flash the binaries to the target. Run the following commands to flash all the LE application images:
   - Bootloaders <usually not needed>

```
Fastboot flash patition <path to gpt.bin>
Fastboot flash cdt <path to cdt.bin>
fastboot flash sbl1 <path to sbl1.mbn >
fastboot flash tz <path to tz.mbn >
fastboot flash rpm <path to rpm.mbn >
```

```
fastboot flash hyp <path to hyp.mbn >
fastboot flash modem <path NON-HLOS.bin>
```

- LE application images

```
fastboot flash aboot <path to emmc_appsboot.mbn >
fastboot flash boot <path to boot.img>
fastboot flash rootfs <path rootfs.ext>
```

4. Reboot the board. During power-up, the USER LED #4 glows green, which indicates that the bootup is completed.

3.3 Boot from SD

See the 96Boards.org and follow the latest release instructions to create a rescue SD card.

http://builds.96boards.org/releases/dragonboard410c/linaro/rescue/latest/

4 Set up a project

4.1 GPT

The GPT maps the eMMC partitions to suit the build.

NOTE: The device rejects to download images without the presence of the GPT in the eMMC.

4.1.1 GPT creation

Use the db-boot-tool to generate a GPT that is required to flash images to the eMMC.

- To create a GPT with CDT support, use the following commands:
  - `git clone https://git.linaro.org/landing-teams/working/qualcomm/db-boot-tools.git/`
  - `cd db-boot-tools`
- To create an empty image (sd.img) with the partition table from linux.txt, run the following command:
  - `sudo ./mksdcard -g -o sd.img -p dragonboard410c/linux/partitions.txt`
- To create a GPT backup, run the following command:
  - `sudo sgdisk -bgpt.bin sd.img`
- To convert GPT backup into the 'fastboot' format, run the following command:
  - `./mkgpt -i gpt.bin -o gpt_both0.bin`
- To flash the GPT to the DragonBoard 410c, run the following command:
  - `Fastboot flash partition gpt_both0.bin`

4.2 CDT

The configuration data table (CDT) is one of the most important data for functionality. The CDT is delivered within the BSP and used by the boot loaders to identify the hardware.

CAUTION: Flash the CDT to the board before booting the LE machine or else the device will fail to boot. See Section 6 for debugging assistance in this scenario.

4.2.1 CDT flashing using Fastboot

Flash the CDT (provided in the BSP) with the following command:

```
Fastboot flash cdt cdt.bin
```

If there is a failure to flash the file, see Section 6 for debugging assistance.

4.3 Building HLOS

4.3.1 Yocto project reference manual

The HLOS is built using the Yocto project. See the following for more information:
http://www.yoctoproject.org/docs/current/ref-manual/ref-manual.html

Additional information is available at https://www.yoctoproject.org/

4.3.2 Download the source files

Table 4-1 Download the source files

Date	Tag/Build ID	Chipset	Manifest
July 7, 2017	IMM.LE.1.0-28000-8x16.0	APQ8016, APQ8016E	IMM.LE.1.0-28000-8x16.0.xml

See the Code Aurora Forum to download the source files:
https://source.codeaurora.org/quic/imm/?h=IMM.LE.1.0-28000-8x16

https://source.codeaurora.org/quic/imm/manifest/tree/?h=IMM.LE.1.0-28000-8x16.0

To download the source files, run the following commands:

- `mkdir apps_proc`
- `cd apps_proc`
- `repo init -u git://codeaurora.org/quic/imm/manifest.git -b IMM.LE.1.0 -m IMM.LE.1.0-28000-8x16.0.xml --repo-url=git://codeaurora.org/tools/repo.git --repo-branch=caf-stable`
- `repo sync`

4.3.3 Set up a bitbake environment

To set up a build environment for the bitbake tool, run the following commands:

- `export OE_MACHINE="dragonboard-410c"`

- `export BUILD_NUMBER=IMM.LE.1.0-8x16_28000`
- `export IMAGES="rpb-desktop-image"`
- `QCOM_EULA=1 MACHINE=${OE_MACHINE} source ./setup-environment build`
- `echo "IMAGE_NAME_append = \"-${BUILD_NUMBER}\"" >> conf/local.conf`
- `echo "LICENSE_FLAGS_WHITELIST = \"commercial\"" >> conf/local.conf`
- `echo "ACCEPT_EULA_dragonboard-410c = \"1\"" >> conf/local.conf`
- `echo "MACHINE_ESSENTIAL_EXTRA_RRECOMMENDS += \" lk\"" >> conf/local.conf`
- ```
  echo "MIRRORS += \"\\
  http://libndp.org/files/ http://www.go-parts.com/mirrors-
  usa/blfs/conglomeration/libndp/ \\n \\
  http://libndp.org/files/
  http://ftp.osuosl.org/pub/blfs/conglomeration/libndp/ \\n \\
  http://libndp.org/files/
  http://ftp.cc.uoc.gr/mirrors/linux/lfs/LFS/svn/l/
  \\n \\
  http://libndp.org/files/ https://www.jabawok.net/gentoo/distfiles/ \\n
  \""
  >> conf/local.conf
  ```

4.3.4 Generate build images

A build content is composed from the following images:

- LK: emmc_appsboot.mbn
- Kernel: boot-dragonboard-410c.img
- HLOS: rpb-desktop-image-dragonboard-410c.ext4.gz

When the process is completed, all the images are available in the following directory:

- build/tmp-glibc/deploy/images/dragonboard-410c

To create a build, run the following commands in apps_proc directory:

- `bitbake ${IMAGES}`
- `cd build/tmp-glibc/deploy/images/dragonboard-410c`
- `gunzip -fk rpb-desktop-image-dragonboard-410c.ext4.gz`
- `mv rpb-desktop-image-dragonboard-410c.ext4 rootfs.ext4`
- `mv boot-dragonboard-410c.img boot.img`

Use the simple open SignLK tool to sign LK for DragonBoard 410c. Download the latest version of SignLK tool from one of the following websites:

https://source.codeaurora.org/quic/imm/imm/sources/tree/signlk?h=IMM.LE.1.0

https://git.linaro.org/landing-teams/working/qualcomm/signlk.git

To sign emmc_appsboot.mbn, run the following commands in apps_proc/build folder:

- ```
  ./signlk -i=tmp-glibc/deploy/images/dragonboard-410c/emmc_appsboot.mbn -
  o=tmp-glibc/deploy/images/dragonboard-410c/
  ```

- `mv tmp-glibc/deploy/images/dragonboard-410c/emmc_appsboot.mbn tmp-glibc/deploy/images/dragonboard-410c/emmc_appsboot_unsigned.mbn`
- `mv tmp-glibc/deploy/images/dragonboard-410c/default/appsbl/emmc_appsboot.mbn tmp-glibc/deploy/images/dragonboard-410c/emmc_appsboot.mbn`
- `rm -Rf tmp-glibc/deploy/images/dragonboard-410c/default/`

NOTE: This environment allows to build each recipe alone.

- Kernel: `bitbake linux-linaro-qcomlt`
- LK: `bitbake lk`

4.4 Build the kernel

Linux kernel image can be compiled without rebuilding the complete HLOS part. To declare the global variables that are used for compilation process, run the following commands:

- `declare -x ARCH="arm64"`
- `declare -x CROSS_COMPILE="<Path to ARM 64bit gcc compiler>/aarch64-linux-gnu-"`

To generate the kernel image and modules, run the following commands:

- `cd apps_proc/sources/kernel`
- `declare -x ARCH="arm64"`
- `declare -x CROSS_COMPILE="<Path to ARM 64bit gcc compiler>/aarch64-linux-gnu-"`
- `make O=../build-${ARCH} defconfig distro.config`
- `make O=../build-${ARCH} -j4 Image dtbs KERNELRELEASE=4.9.29-linaro-lt-qcom`
- `make O=../build-${ARCH} -j4 modules KERNELRELEASE=4.9.29-linaro-lt-qcom`
- `make O=../build-${ARCH}  modules_install INSTALL_MOD_PATH=../build-${ARCH}-modules`

NOTE: The kernel modules (content of build-${ARCH}-modules) must be placed in the root file system under /lib/modules

- `<Path to Skales>/skales/dtbTool -o ../qfpr_dt.img -s 2048 ../build-arm64/arch/arm64/boot/dts/qcom/`
- `<Path to Skales>/skales/mkbootimg --kernel ../build-arm64/arch/arm64/boot/Image --ramdisk ../../initrd.img-4.9.29-linaro-lt-qcom --output ../qfpr-db410c.img --dt ../qfpr_dt.img --pagesize 2048 --base 0x80000000 --cmdline "root=/dev/disk/by-partlabel/rootfs rw rootwait console=ttyMSM0,115200n8 earlycon=msm_serial_dm,0x78b0000"`

4.5 Build LK

Before building the LK, see Section 2.1.3.4 for information on downloading the ARM-EABI compiler.

To generate the LK image, run the following commands:

- `cd apps_proc/sources/lk`
- `make -j4 msm8916 EMMC_BOOT=1 TOOLCHAIN_PREFIX=<path to arm-eabi tree>/bin/arm-eabi-`

NOTE: The LK images must be signed using the signLK tool to boot on the device. If an unsigned LK image was booted, see Section 6 for guidelines.

4.5.1 Sign LK image

Use the simple open SignLK tool to sign LK image for DragonBoard 410c.

- `mv ./build-msm8916/emmc_appsboot.mbn ./build-msm8916/emmc_appsboot_unsigned.mbn`
- `../signlk/signlk.sh -i=./build-msm8916/emmc_appsboot_unsigned.mbn -o=./build-msm8916/emmc_appsboot.mbn -d`

5 Supported features

The following software features have been validated in this release on DragonBoard 410c:

- File system on flash memory
- HDMI display
- Wi-Fi STA (Open and WPA2 PSK) (via wpa_supplicant)
- Wi-Fi SoftAP (Open and WPA2 PSK) (via hostapd)
- Bluetooth (via hcitool and bluetoothctl)
- USB HID (mouse/keyboard)
- USB mass storage
- USB ethernet dongle
- USB camera
- SD card
- GPIOs (compliant to 96boards.org)
- SPI/I^2C (compliant to 96boards.org)
- Serial port (UART) (compliant to 96boards.org)
- Fastboot
- X Window server
- 3D graphics
- GPS (via gpsd and gpsmon)

- Video playback (via gstreamer, up to 1080p 30 fps, hardware decoder)
- Audio playback (software decoder)
- Power button (long press for 10 sec to shut down the device/2 sec to power up)
- Software deployment via fastboot
- CSI YUV camera (OV5640/5)
- RTC read and write
- Chromium browser
- Linaro OpenEmbedded (OE) reference platform build (RPB) for DragonBoard/corresponding morty release version
- Linux Kernel 4.9.29

5.1 GPS enablement

This build contains the initial release of GPS software. To start the GPS, run the following commands:

- `systemctl start gpsd.socket`
- `systemctl start gpsd`
- `systemctl start gnss-gpsd`
- `gpsdctl add /dev/ttyGPS0`
- `gpsmon`

To stop the GPS software, run the following commands:

- `close gpsmon`
- `gpsdctl remove /dev/ttyGPS0`
- `systemctl stop gnss-gpsd`

To restart the GPS software, run the following commands:

- `systemctl start gnss-gpsd`
- `gpsdctl add /dev/ttyGPS0`
- `gpsmon`

5.2 Real time clock usage

The power management chip (PM8916) contains inside hardware real time clock (RTC) that can be used to track current time even when the device is not connected to power.

5.2.1 Connect coin-cell battery for RTC on DragonBoard 410c

Coin-cell battery can be used to keep RTC up-to-date when device has no power attached.
The system was tested with ML-621S/ZTN. The user can use larger capacity coin-cell battery for longer keep alive periods, such as ML-2020/H1CN.

Rework instructions:

1. Remove metallic shield from APQ/PMIC area.
2. Remove C81.
3. Connect positive battery side to PM_VCOIN pad (U9.L11 or C81.1), and negative side to close GND (for example, C81.2).

5.2.2 RTC read and write from user space

To read and update RTC, you must have root access. To read and update RTC, run the following commands:

- Read RTC: `hwclock –r`
- Save current system time in RTC: `hwclock -w`

By default HLOS is limited to read-only access to hardware RTC. To overcome this limitation, when RTC is updated, a file with offset in seconds from current RTC is stored in file /etc/rtc_offset.

5.3 Enablement and test examples

5.3.1 Enable USB camera

- `gst-launch-1.0 -vv v4l2src device=/dev/video1 ! glimagesink`

5.3.2 Enable CSI camera

5.3.2.1 Inspect V4L2 configuration

- `media-ctl -d /dev/media0 -p`

5.3.2.2 Enable driver debug messages

- `echo 0x1e > /sys/class/video4linux/video0/dev_debug`

5.3.2.3 Switch between RDI and PIX

- `media-ctl -d /dev/media0 -l '"msm_ispif0":1->"msm_vfe0_rdi0":0[1]'`
- `media-ctl -d /dev/media0 -l '"msm_ispif0":1->"msm_vfe0_pix":0[0]'`
- `media-ctl -d /dev/media0 -l '"msm_ispif0":1->"msm_vfe0_rdi0":0[0]'`
- `media-ctl -d /dev/media0 -l '"msm_ispif0":1->"msm_vfe0_pix":0[1]'`

5.3.2.4 PIX mode 1280 × 960

5.3.2.4.1 Configure

- `media-ctl -d /dev/media1 -l '"msm_csiphy0":1->"msm_csid0":0[1],"msm_csid0":1->"msm_ispif0":0[1],"msm_ispif0":1->"msm_vfe0_pix":0[1]'`

- media-ctl -d /dev/media1 -V '"ov5645 1-0076":0[fmt:UYVY2X8/1280x960],"msm_csiphy0":0[fmt:UYVY2X8/1280x960],"msm_csid0":0[fmt:UYVY2X8/1280x960],"msm_ispif0":0[fmt:UYVY2X8/1280x960],"msm_vfe0_pix":0[fmt:UYVY2X8/1280x960]'

5.3.2.4.2 Preview

- export DISPLAY=:0
- GST_GL_PLATFORM=egl GST_GL_API=gles2 gst-launch-1.0 v4l2src device=/dev/video3 ! 'video/x-raw,format=NV12,width=1280,height=960,framerate=30/1' ! glimagesink sync=false

5.3.2.4.3 Still capture

- gst-launch-1.0 v4l2src device=/dev/video3 num-buffers=1 ! videoparse format=nv21 width=1280 height=960 framerate=30/1 ! jpegenc ! filesink location=image_1280x960.jpg

  OR

- yavta -B capture-mplane -c10 -I -n 5 --requeue-last -f NV12 -s 1280x960 /dev/video3

5.3.2.5 Video recording

- gst-launch-1.0 -v -e v4l2src device=/dev/video3 num-buffers=300 ! video/x-raw,format=NV12,width=1280,height=960,framerate=30/1 ! v4l2video5h264enc extra-controls="controls,h264_profile=4,video_bitrate=15000000;" ! h264parse ! mp4mux ! filesink location=enc.h264.1280x960.mp4

5.3.3 RDI mode 1080p

- media-ctl -d /dev/media0 -l '"msm_csiphy0":1->"msm_csid0":0[1],"msm_csid0":1->"msm_ispif0":0[1],"msm_ispif0":1->"msm_vfe0_rdi0":0[1]'
- media-ctl -d /dev/media0 -V '"ov5645 1-0078":0[fmt:UYVY2X8/1920x1080],"msm_csiphy0":0[fmt:UYVY2X8/1920x1080],"msm_csid0":0[fmt:UYVY2X8/1920x1080],"msm_ispif0":0[fmt:UYVY2X8/1920x1080],"msm_vfe0_rdi0":0[fmt:UYVY2X8/1920x1080]'
- export GST_GL_XINITTHREADS=1
- export DISPLAY=:0
- GST_GL_PLATFORM=egl GST_GL_API=gles2 gst-launch-1.0 v4l2src device=/dev/video0 ! glimagesink sync=false

  OR

- yavta -B capture-mplane -c10 -I -n 5 --requeue-last -f UYVY -s 1920x1080 /dev/video0

  OR

- gst-launch-1.0 v4l2src device=/dev/video0 num-buffers=1 ! 'video/x-raw,format=UYVY,width=1920,height=1080,framerate=30/1' ! jpegenc ! filesink location=image01.jpg

  OR

- `gst-launch-1.0 filesrc location=image01.jpg ! jpegparse ! jpegdec ! imagefreeze ! glimagesink sync=false`

5.3.4 RTSP

5.3.4.1 Server

- `cvlc $1 --sout=#rtp{sdp=rtsp://:$port/$stream_name}`

5.3.4.2 Client

- `gst-launch-1.0 -vvv rtspsrc location=rtsp://$IP:5004/stream.sdp latency=10 ! rtph264depay ! h264parse ! v4l2video4dec ! glimagesink sync=false`

5.4 Swap partition

For some distinct memory intensive tasks, such as running Chromium browser for accessing rich web pages, a memory extension via swap partition can be helpful for faster operation.

To define and use a swap partition on an SD card, perform the following steps:

1. Insert a blank SD card into the SD card slot on the device (for example, DragonBoard 410c).
2. Partition and mount it.
3. Run the following commands:

NOTE: The red color text in the following code example displays user input.

```
sudo gdisk /dev/mmcblk1
Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N): y
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-123764702, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-123764702, default = 123764702) or {+-}size{KMGTP}: 7G
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 8200
Changed type of partition to 'Linux swap'
Command (? for help): n
Partition number (1-128, default 2):
First sector (34-123764702, default = 16779264) or {+-}size{KMGTP}:
Last sector (2048-123764702, default = 123764702) or {+-}size{KMGTP}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
Command (? for help): w
Final checks complete. About to write GPT data. THIS WILL OVERWRITE
EXISTING PARTITIONS!!
Do you want to proceed? (Y/N): y
```

4. Reboot the device (for example, DragonBoard 410c).
5. To define swap partition, run the following commands:
   ```
   sudo reboot now
   sudo mkfs -t ext4 /dev/mmcblk1p2
   sudo mkswap /dev/mmcblk1p1
   ```
6. To update `/etc/fstab`, run the following command:
   ```
   sudo sed -i '$a /dev/mmcblk1p1 none swap sw 0 0\n' /etc/fstab
   ```
7. Reboot the device. To verify that the swap was properly defined, run the following commands:
   ```
   sudo reboot
   free
   ```

6 Troubleshooting

6.1 Cannot flash CDT using Fastboot

Flash a proper GPT that includes a CDT partition, see Section 4.1 for more information.

6.2 Device fails to boot

If the device does not boot properly, check and find out the root cause of the failure.

The following are the most common problems that prevent the device from booting:

- The build images were not loaded successfully – See Section 3.2.1 for more information.
- If the CDT was not flashed to the device, perform the following steps:
  a. Connect the DragonBoard to a terminal using UART (external mezzanine is required) or ssh.
  b. Follow the boot log and search for the CDT configuration stated as follows:
     ```
     B -    291671 - CDT version:3,Platform ID:24,Major ID:1,Minor
     ID:0,Subtype:0
     ```
     If the platform ID is not 24, see Section 4.2.1 for guidelines to flash the CDT.
- DIP switch 1 is set to ‘ON’ – For proper boot, set the DIP switch 1 to 0.
- LK image is not signed – Perform the following steps:
  a. Connect the DragonBoard to a terminal using UART(external mezzanine is required) or ssh.
  b. Search for the following line:
     B - 417972 - Error code 3063 at boot_authenticator.c Line 407

The log displays that the LK image (emmc_appsboot.mbn file) was not signed before loading it to the device. See Section 4.5.1 for the steps and guidelines to sign the image.

CAUTION: If the LK image is not authenticated and loaded properly, Fastboot is not supported. See Section 3.3 for the steps on SD rescue.

7 Known issues and limitations

7.1 Limitations

- Internal GPS antenna requires clear open sky environment for optimal performance, so external antenna usage is recommended: https://developer.qualcomm.com/qfile/29467/lm80-p0436-42_add_ufl_ant_and_valid_gps_on_android_app_note.pdf
- When restarting the device by power cable reinsertion:
  - If micro-USB cable is attached, press the power-on button for at least 2 seconds.
  - If micro-USB cable is not connected – the device starts automatically.

7.2 Known issues

- Wi-Fi hidden SSID mode is not functional
- Wi-Fi and Bluetooth LE cannot be used at the same time.

7.3 Resolved issues

This section is not applicable to this release.

EXHIBIT 1

PLEASE READ THIS LICENSE AGREEMENT ("AGREEMENT") CAREFULLY. THIS AGREEMENT IS A BINDING LEGAL AGREEMENT ENTERED INTO BY AND BETWEEN YOU (OR IF YOU ARE ENTERING INTO THIS AGREEMENT ON BEHALF OF AN ENTITY, THEN THE ENTITY THAT YOU REPRESENT) AND Qualcomm Technologies, Inc. ("QTI" "WE" "OUR" OR "US"). THIS IS THE AGREEMENT THAT APPLIES TO YOUR USE OF THE DESIGNATED AND/OR ATTACHED DOCUMENTATION AND ANY UPDATES OR IMPROVEMENTS THEREOF (COLLECTIVELY, "MATERIALS"). BY USING, ACCESSING, DOWNLOADING OR COMPLETING THE INSTALLATION OF THE MATERIALS, YOU ARE ACCEPTING THIS AGREEMENT AND YOU AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THESE TERMS, QTI IS UNWILLING TO AND DOES NOT LICENSE THE MATERIALS TO YOU. IF YOU DO NOT AGREE TO THESE TERMS YOU MUST DISCONTINUE AND YOU MAY NOT USE THE MATERIALS OR RETAIN ANY COPIES OF THE MATERIALS. ANY USE OR POSSESSION OF THE MATERIALS BY YOU IS SUBJECT TO THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT.

1.1 **License.** Subject to the terms and conditions of this Agreement, including, without limitation, the restrictions, conditions, limitations and exclusions set forth in this Agreement, Qualcomm Technologies, Inc. ("QTI") hereby grants to you a nonexclusive, limited license under QTI's copyrights to use the attached Materials; and to reproduce and redistribute a reasonable number of copies of the Materials. You may not use Qualcomm Technologies or its affiliates or subsidiaries name, logo or trademarks; and copyright, trademark, patent and any other notices that appear on the Materials may not be removed or obscured. QTI shall be free to use suggestions, feedback or other information received from You, without obligation of any kind to You. QTI may immediately terminate this Agreement upon your breach. Upon termination of this Agreement, Sections 1.2-4 shall survive.

1.2 **Indemnification.** You agree to indemnify and hold harmless QTI and its officers, directors, employees and successors and assigns against any and all third party claims, demands, causes of action, losses, liabilities, damages, costs and expenses, incurred by QTI (including but not limited to costs of defense, investigation and reasonable attorney's fees) arising out of, resulting from or related to: (i) any breach of this Agreement by You; and (ii) your acts, omissions, products and services. If requested by QTI, You agree to defend QTI in connection with any third party claims, demands, or causes of action resulting from, arising out of or in connection with any of the foregoing.

1.3 **Ownership.** QTI (or its licensors) shall retain title and all ownership rights in and to the Materials and all copies thereof, and nothing herein shall be deemed to grant any right to You under any of QTI's or its affiliates' patents. You shall not subject the Materials to any third party license terms (e.g., open source license terms). You shall not use the Materials for the purpose of identifying or providing evidence to support any potential patent infringement claim against QTI, its affiliates, or any of QTI's or QTI's affiliates' suppliers and/or direct or indirect customers. QTI hereby reserves all rights not expressly granted herein.

1.4 **WARRANTY DISCLAIMER.** YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT THE USE OF THE MATERIALS IS AT YOUR SOLE RISK. THE MATERIALS AND TECHNICAL SUPPORT, IF ANY, ARE PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED. QTI ITS LICENSORS AND AFFILIATES MAKE NO WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE MATERIALS OR ANY OTHER INFORMATION OR DOCUMENTATION PROVIDED UNDER THIS AGREEMENT, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR AGAINST INFRINGEMENT, OR ANY EXPRESS OR IMPLIED WARRANTY ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. NOTHING CONTAINED IN THIS AGREEMENT SHALL BE CONSTRUED AS (I) A WARRANTY OR REPRESENTATION BY QTI, ITS LICENSORS OR AFFILIATES AS TO THE VALIDITY OR SCOPE OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT OR (II) A WARRANTY OR REPRESENTATION BY QTI THAT ANY MANUFACTURE OR USE WILL BE FREE FROM INFRINGEMENT OF PATENTS, COPYRIGHTS OR OTHER INTELLECTUAL PROPERTY RIGHTS OF OTHERS, AND IT SHALL BE THE SOLE RESPONSIBILITY OF YOU TO MAKE SUCH DETERMINATION AS IS NECESSARY WITH RESPECT TO THE ACQUISITION OF LICENSES UNDER PATENTS AND OTHER INTELLECTUAL PROPERTY OF THIRD PARTIES.

1.5 **LIMITATION OF LIABILITY.** IN NO EVENT SHALL QTI, QTI'S AFFILIATES OR ITS LICENSORS BE LIABLE TO YOU FOR ANY INCIDENTAL, CONSEQUENTIAL OR SPECIAL DAMAGES, INCLUDING BUT NOT LIMITED TO ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL DAMAGES, ARISING OUT OF THE USE OR INABILITY TO USE, OR THE DELIVERY OR FAILURE TO DELIVER, ANY OF THE MATERIALS, OR ANY BREACH OF ANY OBLIGATION UNDER THIS AGREEMENT, EVEN IF QTI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE FOREGOING LIMITATION OF LIABILITY SHALL REMAIN IN FULL FORCE AND EFFECT REGARDLESS OF WHETHER YOUR REMEDIES HEREUNDER ARE DETERMINED TO HAVE FAILED OF THEIR ESSENTIAL PURPOSE. THE ENTIRE LIABILITY OF QTI, QTI's AFFILIATES AND ITS LICENSORS, AND THE SOLE AND EXCLUSIVE REMEDY OF YOU, FOR ANY CLAIM OR CAUSE OF ACTION ARISING HEREUNDER (WHETHER IN CONTRACT, TORT, OR OTHERWISE) SHALL NOT EXCEED US$10.

2. **COMPLIANCE WITH LAWS; APPLICABLE LAW.**

Any litigation or other dispute resolution between You and Us arising out of or relating to this Agreement, or Your relationship with Us will take place in the Southern District of California, and You and QTI hereby consent to the personal jurisdiction of and exclusive venue in the state and federal courts within that District with respect any such litigation or dispute resolution. This Agreement will be governed by and construed in accordance with the laws of the United States and the State of California, except that body of California law concerning conflicts of law. This Agreement shall not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded.

3. **CONTRACTING PARTIES.** If the Materials are downloaded on any computer owned by a corporation or other legal entity, then this Agreement is formed by and between QTI and such entity. The individual accepting the terms of this Agreement represents and warrants to QTI that they have the authority to bind such entity to the terms and conditions of this Agreement.

4. **MISCELLANEOUS PROVISIONS.** This Agreement, together with all exhibits attached hereto, which are incorporated herein by this reference, constitutes the entire agreement between QTI and You and supersedes all prior negotiations, representations and agreements between the parties with respect to the subject matter hereof. No addition or modification of this Agreement shall be effective unless made in writing and signed by the respective representatives of QTI and You. The restrictions, limitations, exclusions and conditions set forth in this Agreement shall apply even if QTI or any of its affiliates becomes aware of or fails to act in a manner to address any violation or failure to comply therewith. You hereby acknowledge and agree that the restrictions, limitations, conditions and exclusions imposed in this Agreement on the rights granted in this Agreement are not a derogation of the benefits of such rights. You further acknowledges that, in the absence of such restrictions, limitations, conditions and exclusions, QTI would not have entered into this Agreement with You. Each party shall be responsible for and shall bear its own expenses in connection with this Agreement. If any of the provisions of this Agreement are determined to be invalid, illegal, or otherwise unenforceable, the remaining provisions shall remain in full force and effect. This Agreement is entered into solely in the English language, and if for any reason any other language version is prepared by any party, it shall be solely for convenience and the English version shall govern and control all aspects. If You are located in the province of Quebec, Canada, the following applies: The Parties hereby confirm they have requested this Agreement and all related documents be prepared in English.