Qualcomm Technologies, Inc.

Qualcomm® Snapdragon™ 410E (APQ8016E) r1034.2.1 Linux Embedded Software

Release Notes

LM80-P0337-5 Rev. C

April 10, 2018

Questions or comments: https://www.96boards.org/DragonBoard410c/forum

Qualcomm Snapdragon is a product of Qualcomm Technologies, Inc. and/or its subsidiaries. Other Qualcomm products referenced herein are products of Qualcomm Technologies, Inc. or its other subsidiaries.

Qualcomm, DragonBoard, and Snapdragon are trademarks of Qualcomm Incorporated, registered in the United States and other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

Use of this document is subject to the license set forth in Exhibit 1.

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

© 2018 Qualcomm Technologies, Inc. and/or its subsidiaries. All rights reserved.

Revision history

Revision	Date	Description
A	March 2018	Initial release
B	April 3, 2018	An update to *DragonBoard™ 410c Linux Embedded Release Notes* (LM80-P0337-4 Rev. A) document.
C	April 10, 2018	Fixed broken link in Section 2.2

Contents

Tables

1 Introduction

This document describes how to obtain, build, and load 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
- Supported features of the release and known issues
- Troubleshooting

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 DragonBoard410c

Refer to the following websites for additional information:

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

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

1.3 Terms and acronyms

Table 1-1 Acronyms

Term or acronym	Definition
eMMC	Embedded multimedia card
GCC	GNU Compiler Collection (GCC) tool
EABI	Embedded application binary interface
CDT	Configuration data table
GPT	Global Partition Table
HLOS	High-level operating system (generic term for WinMob, Linux and so on)
INITRD	Initial RAM disk

2 Build overview

2.1 Build Structure overview

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

A full build content contains the following components:

- gpt.bin – See Section 4.2 for more information
- sbc_1.0_8016.bin (CDT image)
- sbl1.mbn
- tz.mbn
- hyp.mbn
- rpm.mbn
- NON-HLOS.bin
- emmc_appsboot_signed.mbn (LK bootloader)
- boot.img
- rootfs.ext4

Some of the above images are available only in a binary format, which cannot be altered. Other images are built from open source projects available on public repositories.

The binary images are archived and hosted on Qualcomm® Developer Network website. The archive is referred to as a BSP Package.

The BSP Package contains the following:

- All proprietary images:
  - sbl1.mbn – used to boot from on-board flash memory
  - tz.mbn
  - hyp.mbn
  - rpm.mbn
  - NON-HLOS.bin
- CDT file
- Proprietary firmware for WLAN and video

2.2 Build Versions

Table 2-1 summarizes the different versions of the BSP and open source repository referred to in this document. The specified image composition is tested and its supported features and issues are documented in section 7**Error! Reference source not found.**.

NOTE: Other versions of combinations may not show the same functionality or issues.

Table 2-1 Download the source files

Date	Chipset	Manifest	Tag/Build ID	BSP Version
February 2018	APQ8016E	IMM.LE.1.0-33400-8x16.0.xml	IMM.LE.1.0-33400-8x16.0	1034.2.1

2.3 Project setup overview

A build is generally comprised out of four main components:

- GPT
- BSP package
- Apps bootloader (LK)
- Kernel + user space (Operating system)

The GPT is created using the db-boot-tools tool provided by Linaro. The BSP package can be downloaded from Qualcomm developer network website. The Apps bootloader and Operating System can be built together or separately, both processes are described below.

The Linux composition selected for this build is based on the YOCTO project, additional information is available at https://www.yoctoproject.org/.

The build environment used for the YOCTO project is BitBake. The BitBake build process automatically downloads required tools and generate the LK + Kernel + User Space images.

A high-level image creation process is as follows:

1. Download source files.
2. Set up the bitbake environment.
3. Run bitbake.
4. Sign the LK image.

Section 4.3 describes the process in details.

3 Linux build environment set up

3.1 Linux host

Compilation process for DragonBoard410c 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/.

3.2 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 the home directory, or, if the root or sudo access is available, install for all system users under a common location, such as /usr/local/bin or 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 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).

4 Build setup

4.1 BSP download

Board Support Packages are available at the following link:

https://developer.Qualcomm.com/hardware/DragonBoard-410c/software

For Yocto, download from the *Linux* Board Support Package section.

It is mandatory to match the Board Support Package release version to the HLOS release.

See section 2.2 for build version details.

The following images from the archive are used during the software flashing process:

```
/bootloaders-linux/hyp.mbn
/bootloaders-linux/NON_HLOS.bin
/bootloaders-linux/rpm.mbn
/bootloaders-linux/sbl1.mbn
/bootloaders-linux/tz.mbn
/cdt-linux/sbc_1.0_8016.bin
```

4.2 GPT creation

GPT maps the eMMC partitions to suit the build.

NOTE: Images download fails without the presence of a coherent GPT in the eMMC.

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 SD card 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
  ```

- The created file, gpt_both0.bin is used during the software flashing process

4.3 Building LK + HLOS

4.3.1 Download the source files

Refer to the Mass Market project on Code Aurora Forum for the source files:

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

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.1.0-33400-8x16.0.xml --repo-url=git://codeaurora.org/tools/repo.git
--repo-branch=caf-stable
repo sync
```

4.3.2 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_33400
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 "RDEPENDS_packagegroup-rpb_remove = \"docker\"" >> conf/local.conf
```

4.3.3 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
```

NOTE: In case, there is no need to build the full image (LK + Kernel + HLOS), the following commands can be used to build individual images:

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

Sign LK image

The generated LK image needs to be signed. The signLK tool is used for this purpose.

In the *apps_proc/build/tmp-glibc/deploy/images/DragonBoard-410c* directory run the following commands to sign the LK:

```
git clone https://git.linaro.org/landing-teams/working/Qualcomm/signlk.git
./signlk/signlk.sh -i=emmc_appsboot.mbn
```

The following images from the directory are used during the software flashing process:

```
emmc_appsboot_signed.mbn
boot.img
rootfs.ext4
```

5 Load images

5.1 Overview

Ensure that all images are available for flashing, this includes all files specified in section 2.1

The platform can boot from the eMMC or an SD card. Each method follows a different process.

5.2 Boot from eMMC (standard boot)

The following process describes the process to flash the images using the fastboot method.

The fastboot tool is not installed by default, thus, it is required to install it on the host before executing the following:

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. 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
     ```
   - Run the following command from the Windows command shell:
     ```
     sudo fastboot devices
     ```

   The list of registered devices appears.
3. After the device is detected, flash the binaries to the target. Run the following commands to flash all the LE application images (if running from a Linux host, add `sudo` in the beginning of the command):
   - GPT
     ```
     fastboot flash partition <path to gpt_both0.bin>
     ```
   - BSP
     ```
     fastboot flash cdt <path to sbc_1.0_8016.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 >
     ```
   - LE application images
     ```
     fastboot flash aboot <path to emmc_appsboot_signed.mbn >
     ```

```
fastboot flash boot <path to boot.img>
fastboot flash rootfs <path rootfs.ext4>
```

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

5.3 Boot from SD

Refer to 96Boards.org and follow the latest release instructions to create a rescue SD card.
http://builds.96boards.org/releases/DragonBoard410c/linaro/rescue/latest/

6 FW/BSP update from Qualcomm Developer Network

The Linux build includes a set of binary firmware images to operate different integrated controllers or processors.

The linux-fw project has a snapshot of Qualcomm's firmware files, however the snapshot may not be up-to-date.

It is recommended to pick the latest firmware from Qualcomm's developer network website and update the build with them.

This section describes how to update the firmware files on the DragonBoard 410c device.

Prerequisites:

- Install mtools:
  ```
  sudo apt-get install mtools
  ```
- Download required BSP from Qualcomm Developer Network:

https://developer.qualcomm.com/download/db410c/linux-board-support-package-r1034.2.1.zip

The instructions are as follows:

1. Unzip the BSP to a new folder *linux-board-support-package-r1034.2.1*:
   ```
   unzip linux-board-support-package-r1034.2.1.zip
   ```
2. Extract the firmware and create lib/firmware directory layout:
   ```
   mkdir -p bsp/lib/firmware/qcom/venus-1.8/

   cp -r linux-board-support-package-r1034.2.1/proprietary-linux/*
   bsp/lib/firmware/

   MTOOLS_SKIP_CHECK=1 mcopy -i linux-board-support-package-
   r1034.2.1/bootloaders-linux/NON-HLOS.bin ::image/modem.* ::image/mba.mbn
   ::image/wcnss.* bsp/lib/firmware/

   cp bsp/lib/firmware/venus.* bsp/lib/firmware/qcom/venus-1.8/
   ```
3. Copy and overwrite the content of /lib/firmware on DragonBoard (for example, by using sdcard, scp, rsync, or any other method).
4. Synchronize the filesystem on DragonBoard:
   ```
   $ sync
   ```
5. Reboot the DragonBoard 410c device.

7 Supported features

The following software features are 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/I2C (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
- OpenEmbedded/Yocto Morty
- Linux Kernel 4.9.29
- Diag interface over USB, UART, or Ethernet-over-USB

- Wi-Fi Factory Test Mode (over Diag) for WLAN precertification testing
- Bluetooth Factory Test Mode (over Diag) for Bluetooth precertification testing
- Wayland/Weston
- Boot from SD card

7.1 GPS enablement

This build contains basic operation of GPS receiver functionality.

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
```

7.2 Real time clock usage

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

7.2.1 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 is 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 are as follows:

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).

7.2.2 RTC read and write from user space

To read and update RTC, the root access is required. 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.

7.3 Enablement and test examples

7.3.1 Enable USB camera

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

7.3.2 Enable CSI camera

7.3.2.1 Inspect V4L2 configuration

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

7.3.2.2 Enable driver debug messages

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

7.3.2.3 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]'
```

7.3.2.4 PIX mode 1280 x 960

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_cs
id0":0[fmt:UYVY2X8/1280x960],"MSM_ispif0":0[fmt:UYVY2X8/1280x960],"MSM_vfe0
_pix":0[fmt:UYVY2X8/1280x960]'
```

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
```

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
```

7.3.2.5 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
```

7.3.3 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
```

7.3.4 RTSP

Server

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

Client

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

7.4 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 is properly defined, run the following commands:

```
sudo reboot
free
```

8 Troubleshooting

8.1 Cannot flash CDT using Fastboot

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

8.2 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 common problems that prevent the device from booting:

- The build images are not loaded successfully. See Section 5.2**Error! Reference source not found.** for more information.
- If the CDT is 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 5.2 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:
  c. Connect the DragonBoard to a terminal using UART(external mezzanine is required) or ssh.
  d. 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) is not signed before loading it to the device. See Section 0 for the steps and guidelines to sign the image.

NOTE: If the LK image is not authenticated and loaded properly, Fastboot is not supported. See Section 5.3 for the steps on SD rescue.

9 Known issues and limitations

9.1 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 sec.
  - If micro-USB cable is not connected, the device starts automatically.

9.2 Known issues

Modem stability Wi-Fi hidden SSID mode is not functional.

9.3 Resolved issues

Bluetooth and Wi-Fi STA coexistence is improved in 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.