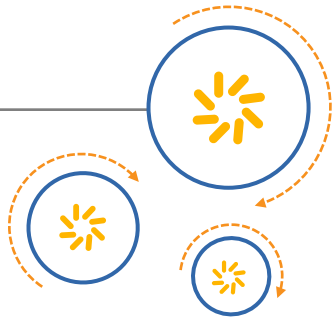
Qualcomm Technologies, Inc.



MSM8937.LA.2.0.1 Linux Android Release 00100 for Automotive

Release Notes for MSM8937.LA.2.0.1-00100-STD.PROD

80-PB677-1 A

April 7, 2017

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

Revision history

Revision	Date	Description
A	April 2017	Initial release of documentation for MSM8937.LA.2.0.1 Linux Android Release 00100 for Automotive (MSM8937.LA.2.0.1-00100-STD.PROD)

Contents

Tables

Figures

1 Getting started

1.1 Purpose

This document provides information for MSM8937.LA.2.0.1 a source code release of software.

Table 1-1 describes the software for this product line, divided into the release packages that must be downloaded separately and combined to have complete product line software set.

Table 1-1 Release packages

From chipcode.qti.qualcomm.com	From codeaurora.org
▪ Proprietary non-HLOS software ▫ Contains proprietary source and firmware images for all non apps processors ▫ An umbrella package built from a combined set of integrated individual component releases	
▪ Proprietary HLOS software ▫ Contains proprietary source and firmware images for the apps processor HLOS	▪ Open source HLOS software ▫ Contains open source for apps processor HLOS

The proprietary and open source HLOS packages are from separate sources and then combined according to the downloading instructions given in Sections 1.6 and 1.7.11. The unique build identification (build ID) code identifies each package with the following naming convention.

<PL Image>-<Version>-<Chipset>

- <PL_Image>–LA.Branch for Linux Android
- <Version> – Variable number of digits used to represent the build ID version
- <Chipset> – MSM8937

For example, LA.UM.5.6-01810-89xx.0-1

Table 1-2 Component build properties

Component build release	Source or binary only	Toolchain required for building source	Python version	Perl version	Cygwin	Supported build hosts
Android HLOS	Source	Android GNU toolchain Ubuntu 12.04 arm-linux-androideabi-gcc (GCC) 4.8	--	--	--	Linux only
MPSS	Source	Qualcomm Hexagon™ DSP 6.4.06	Python 2.7.5	Perl 5.14.2 (install XML::Parser module) sudo apt-get install aptitude sudo aptitude install libxml-simple-perl)	Windows builds only; needs tee.exe	Linux, Windows XP, and Windows 7
Boot loaders	Source	ARM Compiler Tools 5.01 update 3 (build 94)	Python 2.7.5	Perl 5.8.x Linux builds only	Windows builds only; needs tee.exe	Linux, Windows XP, and Windows 7
RPM	Source	ARM Compiler Tools 5.01 update 3 (build 94)	Python 2.7.5	Perl 5.6.1	Windows builds only; needs tee.exe	Linux, Windows XP and Windows 7 only
TZ	Source	Qualcomm Snapdragon™ processor LLVM ARM compiler 3.5.2.4 gcc-linaro-arm-linux-gnueabihf-4.9-2014.05_win32	Python 2.7.6	--	Windows builds only; needs tee.exe	Linux, Windows XP, and Windows 7 only
CNSS	Binary	-	-	-	-	-
ADSP	Source	Hexagon Tools Version: 5.1.05	Python 2.7.6	Perl 5.6.1	Windows builds only; needs tee.exe	Linux, Windows XP, and Windows 7 only
CPE	Binary	-	-	-	-	-
Video	Binary	-	-	-	-	-

1.2 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>.

Commands to be entered appear in a different font, for example, copy a:*.* b:.

Button and key names appear in bold font, for example, click Save or press Enter.

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

1.3 Release availability

This document and related product documents are available to individuals who work for our licensees. These documents are available on CreatePoint at https://createpoint.qti.qualcomm.com/ .

The electronic download of the software release code is made available only to select individuals chosen by the licensee. For copies of that code, contact the selected individuals within your own organization.

1.4 Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at https://createpoint.qti.qualcomm.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

1.5 Download QTI proprietary software from ChipCode

QTI software can be downloaded from the ChipCode website. Designated points of contact in your organization can download the licensed software. The software is organized into distribution packages (distros) composed of subsystem image files. Each distro has a corresponding Git project. The Git tree includes revisions for previous builds that allow you to diff the changes between releases.

1. If you are new to ChipCode, review the following link for up-to-date documentation and a set of tutorial videos:

   https://chipcode.qti.qualcomm.com/projects/help/wiki

2. Create a top-level directory on the build PC and unzip each of the subsystem images to generate the following directory structure. In this example, <target_root> is the top-level directory.

   <target_root>

   /common/

   /cpe_proc/

   /boot_images/

/modem_proc/

/rpm_proc/

/trustzone_images/

/wcnss_proc/

/LINUX/

/adsp_proc/

/venus_proc

/addon <Applicable only if you have a WAPI license>

contents.xml

3. Ensure that the contents.xml file is located in the root folder as shown above.

1.6 Download open source HLOS software

The Linux Board Support Package (BSP) release is obtained in two parts, a proprietary release from ChipCode and an open source release from the Code Aurora Forum (CAF) site.

1. Open the contents.xml file that was downloaded from ChipCode (see Section 1.5) and search for the <name>apps</name> tag.
2. Below this tag, locate the <build_id> tag. This identifies the corresponding open source HLOS software build similar to LA.UM.5.6-01810-89xx.0-1.
3. Follow the instructions listed at https://www.codeaurora.org/xwiki/bin/QAEP/release (look under the Branch releases section) to find the APSS build ID in the released software.
4. In an empty directory, use the repo init command with the correct branch and manifest as indicated in the branch releases table:

   ```
   $ repo init -u git://codeaurora.org/platform/manifest.git -b release -m
   [manifest] –repo-url=git://codeaurora.org/tools/repo.git
   ```

5. Type the following repo sync command:

   ```
   $ repo sync
   ```

6. After the repo sync finishes, copy the vendor/qcom/proprietary directory tree from the proprietary HLOS release into the open source HLOS source tree contained in your workspace.

   ```
   $cp -r <LYA_build_location>/HY11-<build_id>/LINUX/android/*
   make –j4
   ```

Table 1-3 Android build information (CAF link for current release)

Date	Tag / Build ID	Chipset	Manifest	Android Version
April 05, 2017	LA.UM.5.6.c1-00300-89xx.0	msm8937_64 msm8937_32	LA.UM.5.6.c1-00300-89xx.0.xml	07.00.00

CAF:
https://source.codeaurora.org/quic/la/platform/manifest/tag/?h=LA.UM.5.6.c1-00300-89xx.0

Table 1-4 Metabuild components

Name	Build ID
META build	MSM8937.LA.2.0.1-00100-STD.PROD-2
ADSP	ADSP.8953.2.8.2-00053-00000-1
BOOT	BOOT.BF.3.3-00211-M8917LAAAANAZB-1
CNSS	CNSS.PR.4.0-00366-M8953BAAAANAZW-1
CPE	CPE.TSF.1.0-00035-W9335AAAAAAAZQ-1
Android	LA.UM.5.6.c1-00300-89xx.0-1
MPSS	MPSS.JO.2.0.c1-00179-8937_GENNS_PACK-1
RPM	RPM.BF.2.2-00213-M8937AAAAANAZR-1
TZ	TZ.BF.4.0.5-00033-M8937AAAAANAZT-1
Video	VIDEO.VE_ULT.3.1-00027-PROD-1
WLAN.ADDON_PR	WLAN.ADDON_PR.3.0-00004-M8953BAAAANAZW-1

For the build components and IDs, refer to the about.html file in the root of the repository on Qualcomm ChipCode.

1.7 Compile non-HLOS software

The following sections provide details to compile the non-HLOS software for MSM8937 device.

1.7.1 Set build Windows environment

Before issuing the non-HLOS build commands, certain command environment settings are set to ensure the correct executable path and toolchain configuration. The specific environment settings vary based on your host software installation, but it is similar to myenviron_amss.cmd script (for Windows), which sets the path to point to the ARM toolchain lib, include, bin, and license file configuration.

```
#
# myenviron_amss
#
SET ARMLMD_LICENSE_FILE=< mylicense_file>@< mylicense_server>
set ARM_COMPILER_PATH=C:\apps\ARMCT5.01\94\bin64
set PYTHON_PATH=C:\Python26
set PYTHONPATH=C:\Python26
set MAKE_PATH=C:\apps\ARMCT5.01\94\bin64
set GNUPATH=C:\cygwin\bin
set CRMPERL=C:\Perl64\bin
set PERLPATH=C:\Perl64\bin
set ARMHOME=C:\Apps\ARMCT5.01\94
set ARMINC=C:\Apps\ARMCT5.01\94\include
set ARMLIB=C:\Apps\ARMCT5.01\94\lib
set ARMBIN=C:\Apps\ARMCT5.01\94\bin
set ARMPATH=C:\Apps\ARMCT5.01\94\bin
set ARMINCLUDE=C:\Apps\ARMCT5.01\94\include
```

```
set ARMTOOLS=ARMCT5.01
set
PATH=.;C:\Python26;C:\Apps\ARMCT5.01\94\bin;C:\apps\ARMCT5.01\94\bin64;C:\c
ygwin\bin;%PATH%
set HEXAGON_ROOT=C:\Qualcomm\HEXAGON_Tools
set HEXAGON_RTOS_RELEASE=5.0.07
set HEXAGON_Q6VERSION=v4
set HEXAGON_IMAGE_ENTRY=0x08400000
```

1.7.2 Build Boot loaders

Compiler version: ARMCT501B94. For correct Tool versions, refer to the Table 1-2 and set the export paths appropriately.

1. For Linux, verify that the following paths are set. Refer to setenv.sh in your boot build:

   ```
   <target_root>\boot_images\build\ms\setenv.sh
   ```

   ```
   export ARMLMD_LICENSE_FILE= <LICENSE FILE INFO>
   export ARM_COMPILER_PATH=/< Path to compiler>/arm/RVDS/5.01bld94/bin64
   export PYTHON_PATH=/<Path to python>/python/2.7.5/bin
   export MAKE_PATH=/<Path to make>/gnu/make/3.81/bin
   export ARMTOOLS=ARMCT5.01
   export ARMROOT=/<Path to compiler>/arm/RVDS/5.01bld94
   export ARMLIB=$ARMROOT/lib
   export ARMINCLUDE=$ARMROOT/include
   export ARMINC=$ARMINCLUDE
   export ARMBIN=$ARMROOT/bin64
   export PATH=$MAKE_PATH:$PYTHON_PATH:$ARM_COMPILER_PATH:$PATH
   export ARMHOME=$ARMROOT
   export armlmd_license
   ```

2. Navigate to the following directory:

   ```
   cd <target_root>/boot_images/build/ms
   ```

   Where <target_root> is the top-level directory created in Section 1.5.

3. Depending on the build environment or release use one of the command options described in Table 1-5.

Table 1-5 Build boot loaders

Build environment	Build command

Linux	Build images –
	`./build.sh TARGET_FAMILY=8937 --prod`
	Clean the Build –
	`./build.sh TARGET_FAMILY=8937 --prod -c`

Windows	Build images –
	`build.cmd TARGET_FAMILY=8937 --prod`
	Clean the Build –
	`build.cmd TARGET_FAMILY=8937 --prod -c`

1.7.3 Build TZ images

Compiler – LLVM3521LGCC

1. Navigate to the following directory:
   `cd <target_root>/trustzone_images/build/ms`
2. Run the commands described in Table 1-6 to build all images.

Table 1-6 TZ build commands

Build environment	To compile MSM8937

Linux	Build images –
	`build.sh CHIPSET=msm8937 devcfg sampleapp`
	
	Clean the build –
	`build.sh CHIPSET=msm8937 devcfg sampleapp -c`

Windows	Build images –
	`build.cmd CHIPSET=msm8937 devcfg sampleapp`
	
	Clean the build –
	`build.cmd CHIPSET=msm8937 devcfg sampleapp -c`

1.7.4 Build RPM

Use the following commands to build RPM (<target_root> is the top-level directory). Ensure that the tool versions are as specified in Table 1-7. The Compiler version is ARMCT501B94.

NOTE: Linux builds may not work for early ES releases.

1. Open a command prompt and change to the following directory:
   `cd <target_root>\rpm_proc\build`
2. Use the appropriate command from the below Table based on the build environment.

Table 1-7 RPM build commands

Build environment	Build command

Linux	Build images –
	`./build_8937.sh`
	Clean the Build –
	`./build_89537.sh –c`

Windows	Build images –
	`build_8937.bat`
	Clean the Build –
	`build_8937.bat -c`

1.7.5 Build ADSP

ADSP images are being released as binaries. Open DSP OEMs can have their own compilation. Below build commands are the used by Qualcomm. Compiler version: Hexagon version 5.1.05

1. Open a command prompt and change to the following directory:

```
cd <target_root>\adsp_proc
```

2. Use the appropriate command from the below Table based on the build environment.

Table 1-8 ADSP build commands

Build environment	Build command

Linux	Build images
	`python ./build/build.py –c msm8937 –o all`
	Clean the build
	`python ./build/build.py –c msm8937 –o clean`

Windows	Build images
	`python build/build.py –c msm8937 –o all`
	Clean the Build –
	`python build/build.py –c msm8937 –o clean`

1.7.6 Build MPSS image

Use the compiler as – Hexagon™ version – 6.4.06 and above. For correct tool versions, refer to Table 1-2 and set the export paths appropriately.

1. For Linux, verify that the following paths are set by referring to setenv.sh in the build.

```
<target root>\modem_proc\build\ms\setenv.sh
     export ARMLMD_LICENSE_FILE=<LICENSE FILE INFO>
ARM_COMPILER_PATH=/pkg/qct/software/arm/RVDS/2.2BLD593/RVCT/Programs/2.2
/593/linux-pentium
     PYTHON_PATH=/pkg/qct/software/python/2.7.5/bin
MAKE_PATH=/pkg/gnu/make/3.81/bin
export ARMTOOLS=RVCT221
export ARMROOT=/pkg/qct/software/arm/RVDS/2.2BLD593
export ARMLIB=$ARMROOT/RVCT/Data/2.2/349/lib
export ARMINCLUDE=$ARMROOT/RVCT/Data/2.2/349/include/unix
export ARMINC=$ARMINCLUDE
export ARMCONF=$ARMROOT/RVCT/Programs/2.2/593/linux-pentium
export ARMDLL=$ARMROOT/RVCT/Programs/2.2/593/linux-pentium
export ARMBIN=$ARMROOT/RVCT/Programs/2.2/593/linux-pentium
export PATH=$MAKE_PATH:$PYTHON_PATH:$ARM_COMPILER_PATH:$PATH
export ARMHOME=$ARMROOT
export HEXAGON_ROOT=/pkg/qct/software/hexagon/releases/tools
```

2. Navigate to the following directory:

```
Cd <target_root>/modem_proc/build/ms
```

3. Perl: install XML::Parser module

   sudo apt-get install aptitude

   sudo aptitude install libxml-simple-perl

4. Depending on your build environment, use one of the following commands described in Table 1-9.

Table 1-9 MPSS build commands

Build Variant	Build environment	Build Commands	Comments
Kitchen sink (ALL RATs)	Linux	▪ Build images – `./build.sh 8937.genns.prod -k` ▪ Clean the Build – `./build.sh 8937.genns.prod -c`	Main flavor including all RATs (W+T+G+C+L).
	Windows	▪ Build images – `build.cmd 8937.genns.prod -k` ▪ Clean the Build – `build.cmd 8937.genns.prod -c`	

1.7.7 Build WCNSS image

NOTE: WCNSS image is released as a Binary and no build compilation is needed.

1.7.8 Codec Processing Engine (CPE) image

The CPE was introduced from WCD9330 (TomTom) Audio Codec to support Ultra-low power SVA Snapdragon Voice Activation (SVA) solution. The CPE is a non-HLOS image and is stored in the apps processor file system.

The CPE image is delivered as a binary and no build instructions or build loading is required as it is down loaded to WCD93xx CPE on-chip memory by codec driver using the SLIMBUS interface.

1.7.9 Video image

NOTE: The VIDEO image is released as a binary and no build compilation is required.

1.7.10 Update Non-HLOS.bin

This release uses componentized META:

- Files such as partition.xml and pil-splitter.py have been moved to the common/config folder.
- All the T32 files, build loading files, and debug scripts have been moved to the common/core folder.
- All sectools related files have been moved to the common/sectools folder.

- META Scripts reside under common/build folder.

The below table describes the supported build flavors.

The contents.xml files (apart from the default contents.xml) are now placed in the common/build folder. Copy the required contents file to the root folder, and rename it as contents.xml.

Meta contents.xml	Compiled images for the prerequisite	META compilation procedure	Comments
contents.xml (Included in build by default for 64-bit)	All the images are compiled for MSM8937 as mentioned in respective tables. APSS must be compiled for 64 bit. (Default - 64 bit APSS)	1. CD <target_root>\common\build 2. Run python build.py	Generates NON-HLOS.bin for MSM8937 ▪ 64-bit APPS. ▪ Modem RATs (W+T+G+C+L)
contents_msm8937_32bit.xml (common/build folder) Note: Copy the above file to the root folder, and rename it as contents.xml.	All the images are compiled for MSM8937 as mentioned in respective tables. APSS must be compiled for 32 bit.	1. CD <target_root>\common\build 2. Run python build.py	Generates NON-HLOS.bin for MSM8937 ▪ 32-bit APPS. ▪ Modem RATs (W+T+G+C+L)

NOTE: Customers who are not entitled to get WAPI DISTRO must comment the following section in contents.xml to compile the META.

```
  <build>
      <name>wapi</name>
      <role>wapi</role>
      <chipset>msm8937</chipset>
      <build_id>WLAN.ADDON_PR.3.0-xxxxx-M8953BAAAANAZW-1</build_id>
      <windows_root_path
cmm_root_path_var="WAPI_BUILDROOT">..\WLAN.ADDON_PR.3.0</windows_root_path>
      <linux_root_path
cmm_root_path_var="WAPI_BUILDROOT">../WLAN.ADDON_PR.3.0</linux_root_path>
      <image_dir>addon</image_dir>
      <release_path>HY11_CompileTest</release_path>
      <buildfile_path>cd/</buildfile_path>
      <build_command>cd ./addon/build_wapi; source
./build.sh</build_command>
    </build>
```

If MPSS / WCNSS / ADSP images are recompiled, use the following commands to update the NON-HLOS.bin file with the new images (<target_root> is the top-level directory).

1. Navigate to the following directory:

   ```
   cd <target_root>/common/build
   ```

2. Enter the command:

   ```
   python build.py
   ```

In Linux, you are advised to copy Linux/android code downloaded from codeaurora.org to the LINUX/android directory downloaded from Qualcomm ChipCode for compilation.

1.7.11 Build and load HLOS

HLOS build needs software code from CAF and HY11 (proprietary part)

1. In a BASH shell, navigate to the Android source tree base directory.
   `cd /LINUX/android`
2. Type the following command to configure the build environment shell settings.
   `source build/envsetup.sh`

NOTE: You must use the source command so the environment settings are defined in the current shell.

3. Type the lunch command to select the build configuration, or enter with no parameters to see an interactive menu for making selections.
   `lunch msm8937_64-userdebug (64-bit kernelspace and 64-bit user space)`
   `lunch msm8937_32-userdebug (32-bit kernelspace and 32-bit user space)`

4. Run `make` to start the build. To run parallel builds for faster build times on a multicore build machine, run the following command:

   `make –j` (total number of processor cores).

   For build machine having 4 processor cores the command looks as follows:

   `make –j4`

NOTE: Find the number of processor cores in the build server that is used for optimized build processing

2 Supported ASICs

2.1 Supported ASICs

This software can be used with the MSM ASICs and revisions, with the indicated release quality. ASIC revisions available at the time of this release are assumed to be supported, unless otherwise indicated.

ASIC hardware	ASIC hardware rev	P code	RR code	Note
MSM8937	-	-	-	

ES = Engineering Sample; FC = Feature Complete; CS = Commercial Sample

Platform information

The following platforms are supported in this release:

- MSM8937 with PM8937, PMI8937/PMI8952 + WTR2965 + WCN3680B

Software platform version

The following software platforms are supported in this release:

- Linux kernel – Ver 3.18
- Android Nougat release – 07.00.00
- Android security patch level – PLATFORM_SECURITY_PATCH = 2016-12-05

NOTE: Customers are expected to update the PLATFORM_SECURITY_PATCH variable in platform/build/core/version_defaults.mk file based on the CRs been made part of release. For guideline and date refer the link https://source.android.com/security/bulletin/

Table 2-1 Software maturity of constituent SPs

Software product	Maturity level
MSM8937.LA.2.0.1	ES

The following memory platforms are supported in this release:

- 3 GB LPDDR3
- 2 GB LPDDR3

2.2 Memory configurations and usage

Figure 2-1 shows the memory map configurations for MSM8937.

NOTE: This is a preliminary memory map and may change in the future releases. It is for reference-only in this release and should not be used for optimization.

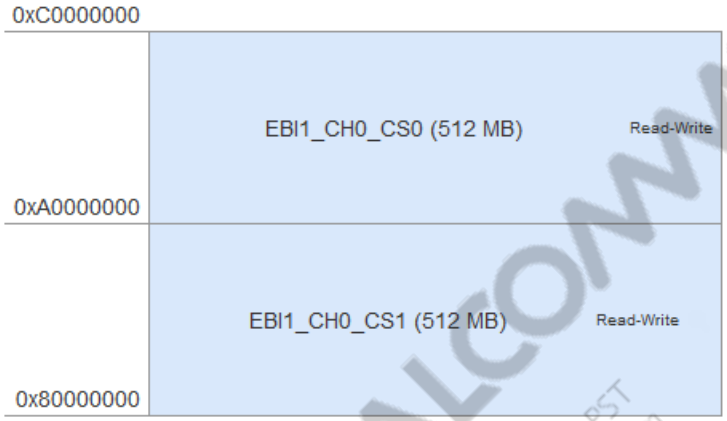


Figure 2-1 MSM8937 memory map diagram

Figure 2-2 EBI1_CH0_CS1 memory map expanded

Figure 2-3 Relocatable partitions of MSM8937 memory map

Figure 2-4 MPSS_EFS/SBL (Boot) partition of MSM8937 memory map

3 New features

The following are the new features applicable to this release:

3.1 New features

3.1.1 Audio

This section is not applicable to this release.

3.1.2 Apps

This section is not applicable to this release.

3.1.3 Boot

This section is not applicable to this release.

3.1.4 Modem

This section is not applicable to this release.

3.1.5 RF

This section is not applicable to this release.

3.1.6 RPM

This section is not applicable to this release.

3.1.7 TZ

This section is not applicable to this release.

3.1.8 WLAN

This section is not applicable to this release.

3.2 Features not part of this release

This section is not applicable to this release.

4 Bugs and Limitations

This chapter lists the bugs and limitations reported for this product line:

- New – Newly reported bugs and limitations
- Ongoing – Previously reported bugs and limitations
- Resolved – Previously reported bugs and limitations that have been resolved and are no longer relevant

NOTE: For a list of all completed and known Change Requests (CRs), refer to the release history tab in Createpoint.

4.1 New

4.1.1 Bugs

This section is not applicable to this release.

4.1.2 Limitations

This section is not applicable to this release.

4.2 Ongoing

4.2.1 Bugs

This section is not applicable to this release.

4.2.2 Limitations

- CR 997480 – Putty console is disabled to avoid camera ping-pong errors

  For information on serial logging, see Section 5.2.3.

4.3 Resolved

4.3.1 Bugs

- For a list of all completed and known Change Requests (CRs), refer to the release history tab in Createpoint.

4.3.2 Limitations

- This section is not applicable to this release.

5 Additional information

5.1 Change request

This section is not applicable to this release.

5.2 Dependency information

5.2.1 Recommendations

5.2.1.1 Enable/exclude NFC and smartcard service

HLOS modifications

The code is part of the build. You need to enable\disable specific flags, in order to build\exclude: the NFC stack and the Smartcard Service. On MSM8917, you need to modify the device/qcom/msm8937_64/ msm8937_64.mk file. On the most recent build, the TARGET_USES_NQ_NFC flag is set to TRUE by default. Also, in those builds, the TARGET_ENABLE_PROPRIETARY_SMARTCARD_SERVICE is defined in vendor/qcom/proprietary/common/msm89XX/BoardConfigVendor.mk.

In the latest builds NFC and smartcard service is independent; if NFC needs #in device/qcom/msm8937/msm8937.mk to be disabled and customer needs smartcard service then:

```
TARGET_USES_NQ_NFC := false
TARGET_ENABLE_PROPRIETARY_SMARTCARD_SERVICE := true
 #in vendor/qcom/proprietary/common/msm89XX/BoardConfigVendor.mk.
```

In kernel, use the following commabds to enable or disable NFC driver:

```
\kernel\arch\arm64\configs\<device config file>
```

- To enable (by default switched ON in MSM8917):

  `CONFIG_NFC_NQ=y`

- To disable:

  Comment or delete this flag – `#CONFIG_NFC_NQ=y`

5.2.1.2 Procedure to enable secure video CMA to support Widevine Level 1 and WFD over HDCP

Secure video CMA of 112 MB is needed to support Widevine Level 1 and WFD over HDCP.

This region is disabled by default in the DT file. Having lower CMA gives better performance on targets with lower RAM sizes, 1 GB and 2 GB.

Customers who need to support these features have to enable this secure CMA region in the DT file. To enable this CMA region, remove the highlighted line from msm8937.dtsi (for MSM8937) and msm8917.dtsi (for MSM8917) in kernel code

```
secure_mem: secure_region@0 {
compat“ble = "shared-d”a-pool";
reusable;
alignment = <0 0x400000>;
size = <0 0x7000000>;
st“tus = "d”sabled";
```

5.2.1.3 Increased EFS size to 1.5 MB as default configuration

NOTE: If you do not want to increase EFS size from 856 KB to 1.5 MB, do not integrate the following CRs:

- CR 1018458 – MPSS
- CR 1024120 – APSS

CAUTION: If you have already launched products with 856 KB EFS size, do not implement the following changes, if implemented, the FOTA will break.

CAUTION: If you want the EFS size as 1.5 MB, ensure the following:

- Device is fully wiped/erased before flashing the new build otherwise the device will not boot up (As EFS size is being upgraded from 856 KB to 1.5 MB)
- QCN is reprogrammed using RF Calibration.

5. The EFS partitions “modemst1”, “modemst2” and “fsg” have size 1536 Kb. So there is no change required in partition.xml. If EFS size needs to be increased then update EFS partitions size in partition.xml.
6. SCL_MODEM_EFS_RAM_SIZE macro defines the EFS modem RAM buffer size. It has to be defined as 1.5 MB in modem side cust header file.

   ```
   #define SCL_MODEM_EFS_RAM_SIZE 0x00180000
   ```

7. Shared buffer is allocated dynamically by Apps based on the size in device tree file in kernel/arch/arm64/boot/dts/qcom/msm8917.dtsi (For 8917 use : msm8917.dtsi / For 8937 use : msm8937.dtsi)

   ```
   qcom,rmtfs_sharedmem@00000000 {
   compat“ble = "qcom,shared”em-uio";
   reg = <0x00000000 0x00180000>;
   reg-n“mes =””rmtfs";
   qcom,client-id = <0x00000001>;
   };
   ```

5.2.1.4 USB type-C recommendations

For connecting targets that have default USB type C connector, USB type C to standard A cables is required to connect to the legacy host machines(that does not support USB type C). Type C Hosts can provide 500/900 mA, 1500 mA, and 3000 mA by applying appropriate pull-up resistance on CC line.

Legacy host machines can supply 500 mA to 900 mA depending on the speed of communication. As a USB type-C specification, USB type C to standard A cables should have only 56 kΩ pull up resistance so that the devices are connected to legacy host PC and it can draw 500 mA current after the USB is configured. A few cables available in the market have 10 kΩ pull up resistance on CC line. However, these cables violate type C specification.

NOTE: The following are the recommended values from type-C specification.

DFP advertisement	Current source to 1.7 V-5.5 V	Resistor pull up to 4.75 V-5.5 V	Resistor pull up to 3.3 V ± 5 %
Default USB power	80 µA ± 20 %	56 kΩ ± 20 %	36 kΩ ± 20 %
1.5 A at 5 V	180 µA ± 8 %	22 kΩ ± 5 %	12 kΩ ± 5 %
3.0 A at 5 V	330 µA ± 8 %	10 kΩ ± 20 %	4.7 kΩ ± 5 %

Issue observed with the cables that violate the type-C specification:

- Device can wrongly detect charging current as 3 A, which leads to power surge issue.
- It could cause USB ID pin toggling when connected to the wall charger that is not connected to a power socket.

5.2.2 Enable serial logging

Use the following link to enable serial logging:

https://us.codeaurora.org/cgit/quic/la//kernel/msm-3.18/commit/?id=ccb6a9387f8e2fc3b72e8d2441fffd81439bd7a7

5.2.3 Tools recommendations

- Recommended ADB version: 1.0.32 or higher
- Recommended QFIL Version: 2.0.0.9

5.2.4 Reclaim logdump partition

Feature: User Space Log Extraction from RAM Dump, is integrated in this release.

Purpose: Intent of this Feature is to continuously collect logcat logs, dump it into a dedicated partition, so that when a device crashes, logcat logs is extracted from this partition while collecting ramdumps, which would help to analyze the crash. This works only when 'UserDebug + Non-perf Boot image' is used, otherwise feature has no impact.

1. This Feature impacts only the 'UserDebug + Non-perf Boot image' (build flavor).

2. The following changes are made to the META partition.xml file. “A new ""ogdump" partition entry (65536 KB in size) is added and the s“ze of "u”erdata" partition is decreased by equivalent amount 65536 KB (or 64 MB).

```
<partition”label=""ogdump" size”in_kb”"65536” type="5AF80809-AABB-4943-
9168-CDFC3”742598" bo”table”"false" re”donly”"false" fi””name="" />

<partition”label="u”erdata" size”in_kb=""973696” type="1B81E7E6-F50D-
419B-A739-2AEEF”DA3335" bo”table”"false" re”donly”"false"
fi”ename="userd”ta.img" ”pars”="true"/>
```

3. Instructions to disable this feature:
   a. To stop the ‘logdumpd’ service altogether make the following change.

NOTE: This is needed only for ‘UserDebug + Non-perf Boot image’ (build flavor).

In the files indicated below based on Target,

for MSM8917/MSM8937/MSM8940/MSM8920:

device/qcom/msm8937_64/msm8937_64.mk (for 64 bit variant)

device/qcom/msm8937_32/msm8937_32.mk (for 32 bit variant)

for MSM8953:

device/qcom/msm8953_64/msm8953_64.mk (for 64 bit variant)

device/qcom/msm8953_32/msm8953_32.mk (for 32 bit variant)

Remove lines:

```
# Enable logdumpd service only for non-perf bootimage
ifeq ($(findstring perf,$(KERNEL_DEFCONFIG)),)
    ifeq ($(TARGET_BUILD_VARIANT),user)
        PRODUCT_DEFAULT_PROPERTY_OVERRIDES+= \
            ro.logdumpd.enabled=0
    else
        PRODUCT_DEFAULT_PROPERTY_OVERRIDES+= \
            ro.logdumpd.enabled=1
    endif
else
    PRODUCT_DEFAULT_PROPERTY_OVERRIDES+= \
        ro.logdumpd.enabled=0
endif

Add below line:
PRODUCT_DEFAULT_PROPERTY_OVERRIDES+= \
            ro.logdumpd.enabled=0
```

b. Remo“e the ""ogdump" partition entry and add 65536 KB (or 64 MB) “o the "u”erdata" partition as indicated below.

~~<partition”label=""ogdump" size”in_kb""65536” type="5AF80809-AABB-4943-9168-CDFC3”742598" bo”table""false" re”donly""false" fi””name="" />~~

```
<partition”label="u”erdata" size”in_kb=""039232” type="1B81E7E6-F50D-419B-A739-2AEEF”DA3335" bo”table""false" re”donly""false" fi”ename="userd”ta.img" ”pars”="true"/>
```

5.2.5 How to develop a DVR app

1. This release supports CSI+USB dual camera preview and recording, and support the following standard APIs, such as

   `Camera.getNumberOfCameras() to retrieve system camera number and camera ID, Camera.getCameraInfo() to identify camera characters such as if the camera is front or back.`

2. The main APIs for preview:

   ```
   Camera.open() open camera with required camera ID
   getParameters() get Camera parameters
   setParameters() set Cameraparameters
   startPreview() start preview
   setPreviewCallbackWithBuffer() set camera rawdata callback object
   takePicture() take picture
   ```

3. The main APIs for recording:

   ```
   new MediaRecorder() create recorder object
   setCamera() set camera of recorder
   setProfile() set recorder profile
   setMaxDuration() set recorder max duration for one file
   setOutPutFile() set recorder output file
        setOnInfoListener() set recorder info listener, call
   setOutPutFile to set next recording file when receive the
   MEDIA_RECORDER_INFO_MAX_DURATION_REACHED message.
   setOnErrorListener() set recorder error listener
   prepare() prepare recorder
   start() start recorder
   reset() reset recorder
   release() release recorder
   ```

4. Note: USB camera do not support preview with setPreviewSurface.
5. 2nd camera preview and recording development process is the same as 1st camera.
6. Floating window support example (used to create floating preview in concurrency with other applications):

```
        mWindowManager = (WindowManager) mContext.getSystemService(
                    Context.WINDOW_SERVICE);
                          mLayoutParams = new WindowManager.LayoutParams();
                        mLayoutParams.type =
WindowManager.LayoutParams.TYPE_SYSTEM_OVERLAY;
                        mLayoutParams.format = PixelFormat.RGBA_8888;
                        mLayoutParams[i].width = 320;
                        mLayoutParams[i].height = 240;
                   mLayoutParams[i].gravity = Gravity.LEFT | Gravity.TOP;
                       mWindowManager.addView(mRootView, mLayoutParams);
```

The app developers need to rend raw (YUV) data to surface when retrieving YUV data with camera callback function.

5.2.6 How to enable CarPlay related features

This release has tested CarPlay related features such as Apple Authorization Chip (rev. c) recognition by I2C interface, and USB role switching support for USB controllers. This is verified on MTP with i2c5 interface. Customers need to pay attention to the following:

1. The authorization chip works on 1.8V.
2. During I2C communication, the SCL frequency must set to 100 kHz.
3. Need to follow CarPlay chip timing, and delay 500us before reading register.
4. Need disable low power mode of authorization chip through I2C command if communication with chip is needed continuously.
5. The following are the suggested code change:

```
static struct i2c_client *key_ic_client;

static struct i2c_board_info key_ic_info = {
/*  0x11 is the slave ic addr  */
I2C_BOARD_INFO("carplay_key_ic", 0x11),
}
        static const struct file_operations key_ic_node_fops = {
    .owner = THIS_MODULE,
    .open = simple_open,
    .read = key_ic_show,
    .write = key_ic_store,
};
//register i2c device
key_ic_client = i2c_new_device(i2c_adap, &key_ic_info);
//create usersapce interface
           key_ic_dentry = debugfs_create_file("enable_key_ic",
S_IRUGO, NULL, NULL, &key_ic_node_fops);
               i2c write:
```

```
            i2c_smbus_write_byte_data(key_ic_client,
0x10, 0x33);
           i2c read:
i2c_smbus_write_byte(key_ic_client, 0x00);
           udelay(500);  // this delay is Necessary
           value = i2c_smbus_read_byte(key_ic_client);
           for scl clock change 100khz
        --- a/arch/arm/boot/dts/qcom/msm8937.dtsi
+++ b/arch/arm/boot/dts/qcom/msm8937.dtsi
@@ -576,7 +576,7 @@
          reg = <0x7af5000 0x600>;
          interrupt-names = "qup_irq";
          interrupts = <0 299 0>;
-         qcom,clk-freq-out = <400000>;
+         qcom,clk-freq-out = <100000>;
          qcom,clk-freq-in  = <19200000>;
          clock-names = "iface_clk", "core_clk";
          clocks = <&clock_gcc clk_gcc_blsp2_ahb_clk>

adb shell test this function
// read value from register
#cat /sys/kernel/debug/enable_key_ic
or
// write a value to register
#echo 1 > /sys/kernel/debug/enable_key_ic
Enable OTG Function in defconfig
In order to use switch host and slave function, mast enable otg function
--- a/arch/arm64/configs/msm8937_defconfig
+++ b/arch/arm64/configs/msm8937_defconfig
@@ -437,6 +437,7 @@ CONFIG_HID_MULTITOUCH=y
 CONFIG_USB_HIDDEV=y
 CONFIG_USB_ANNOUNCE_NEW_DEVICES=y
 CONFIG_USB_MON=y
+CONFIG_USB_OTG=y
 CONFIG_USB_XHCI_HCD=y
 CONFIG_USB_EHCI_HCD=y
 CONFIG_USB_EHCI_MSM=y
Change msm8937.dtsi file
--- a/arch/arm/boot/dts/qcom/msm8937.dtsi
+++ b/arch/arm/boot/dts/qcom/msm8937.dtsi
@@ -1109,7 +1109,8 @@

          qcom,hsusb-otg-phy-type = <3>; /* SNPS Femto PHY */
          qcom,hsusb-otg-mode = <3>; /* OTG mode */
```

```
-         qcom,hsusb-otg-otg-control = <2>; /* PMIC */
+         qcom,hsusb-otg-otg-control = <3>; /* USER control */
+         qcom,hsusb-otg-default-mode = <1>; /* USB_PERIPHERAL */
          qcom,dp-manual-pullup;
          qcom,hsusb-otg-mpm-dpsehv-int = <49>;
          qcom,hsusb-otg-mpm-dmsehv-int = <58>;
how to use this function
          In user space mode, you can use commands “echo” to modify this
feature.
          set usb to host mode
echo host > /sys/kernel/debug/msm_otg/mode
          set usb to peripheral mode
echo peripheral > /sys/kernel/debug/msm_otg/mode
```

5.2.7 How to enable OTA

This release tested OTA to be working on QRD and MTP platform. Refer to 80-NL409-1 C doc for details.

5.2.8 How to enable acc wake up system

Acc drvier code was executed in ADSP source code in Qualcomm reference design. So we need change ADSP code to support this feature.

adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.c

```
diff --git a/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.c
b/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.c
index 11ee8a7..b8fa7dc 100755
--- a/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.c
+++ b/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.c
@@ -365,7 +365,8 @@ static sns_ddf_status_e bma2x2_set_powerstate(
              byte_reg = BST_CLR_VAL_BIT(byte_reg, 5);
              break;
          case SNS_DDF_POWERSTATE_LOWPOWER:
-             byte_reg = BST_SET_VAL_BIT(byte_reg, 7);
+/*           byte_reg = BST_SET_VAL_BIT(byte_reg, 7);  */
+             byte_reg = BST_CLR_VAL_BIT(byte_reg, 7);
              byte_reg = BST_CLR_VAL_BIT(byte_reg, 6);
              byte_reg = BST_CLR_VAL_BIT(byte_reg, 5);

@@ -619,7 +620,9 @@ static sns_ddf_status_e bma2x2_enable_data_ready_int(
     BMA2X2_DD_CHECK_RETVAL(status, SNS_DDF_SUCCESS);

     if (enable) {
-        regv = BST_SET_VAL_BIT(regv, 4);
```

```
+//      regv = BST_SET_VAL_BIT(regv, 4);
+       regv = BST_CLR_VAL_BIT(regv, 4);
     } else {
         regv = BST_CLR_VAL_BIT(regv, 4);
     }
@@ -1878,6 +1881,13 @@ static sns_ddf_status_e
sns_dd_acc_bma2x2_config_default(
         return stat;
     }

+  stat = sns_dd_acc_bma2x2_config_high_low_g_int(state, 1);
+    if (SNS_DDF_SUCCESS != stat)
+    {
+        return stat;
+    }
+
     /*!set interrupt latch time,  if the interrupt signal can not be
catched easily,
       then set this mode, notice: not set at DRI mode */
     //bma2x2_set_int_mode(state->port_handle, 1); // latched 250 ms
@@ -2680,7 +2690,7 @@ sns_ddf_status_e
sns_dd_acc_bma2x2_config_any_motion_lpm1(
     }

     regv = 0x00;
-    regv = BST_SET_VAL_BIT(regv, 6);
+/*  regv = BST_SET_VAL_BIT(regv, 6);  */
     regv = BST_SET_VAL_BITBLOCK(regv, 1, 4, any_motion_setting-
>sleep_dur);
     status = bma2x2_smbus_write_byte(state->port_handle,
BMA2X2_MODE_CTRL_REG, &regv);
     if (SNS_DDF_SUCCESS != status) {
@@ -3219,6 +3229,12 @@ sns_ddf_status_e sns_dd_acc_bma2x2_set_attr(
             return SNS_DDF_EINVALID_PARAM;
     }

+  if (sensor_type == SNS_DDF_SENSOR_ACCEL)
+  {
+       ret_val = sns_dd_acc_bma2x2_config_high_low_g_int(state, 1);
+  }
+
     return (ret_val);
 }
```

```
diff --git a/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.h
b/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.h
index 4da204e..288d248 100755
--- a/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.h
+++ b/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_bma2x2.h
@@ -944,6 +944,7 @@ struct bst_sensor_nvm_params_entire {
 #define BMA2X2_EN_SOFT_RESET_VALUE        0xB6
 #define BMA2X2_ACCEL_ENABLE_SLOPE_XYZ_VALUE        7
 #define BMA2X2_ACCEL_ENABLE_DATA_READY_VALUE      0x10
+#define BMA2X2_ACCEL_ENABLE_HIGH_LOW_G_VALUE      0x0f

 #define BMA2X2_MAP_SLOPE_INT_TO_INT1      4
 #define BMA2X2_MAP_DR_INT_TO_INT1         1
@@ -1343,6 +1344,10 @@ extern sns_ddf_status_e
sns_dd_acc_bma2x2_config_motion_int(
         sns_dd_acc_bma2x2_state_t  *state,
         bool                       enable);

+extern sns_ddf_status_e sns_dd_acc_bma2x2_config_high_low_g_int(
+        sns_dd_acc_bma2x2_state_t  *state,
+        bool                       enable);
+
 extern sns_ddf_status_e bma2x2_read_accel_x(
         sns_ddf_handle_t port_handle,
         uint8_t sensor_type,
diff --git
a/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_uimg_bma2x2.c
b/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_uimg_bma2x2.c
index 8b4f2c4..b5d162f 100755
--- a/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_uimg_bma2x2.c
+++ b/ADSP.8953.2.8.2/adsp_proc/Sensors/dd/qcom/src/sns_dd_uimg_bma2x2.c
@@ -133,6 +133,13 @@ static sns_ddf_status_e
sns_dd_acc_bma2x2_trigger_fifo_data(
         uint16_t            num_samples,
         bool                trigger_now);

+extern sns_ddf_status_e bma2x2_set_low_g_threshold(
+        sns_ddf_handle_t port_handle,
+        unsigned char threshold);
+
+extern sns_ddf_status_e bma2x2_set_high_g_threshold(
+        sns_ddf_handle_t port_handle,
+        unsigned char threshold);

 /*==========================================================================
```

```
@@ -708,6 +715,109 @@ sns_ddf_status_e sns_dd_acc_bma2x2_config_motion_int(
             rega, &regv);
     BMA2X2_DD_CHECK_RETVAL(status, SNS_DDF_SUCCESS);

+    return status;
+}
+
+/*!
+ *  @brief Configure (enable/disable) the high and low G interrupt
+ *
+ *  @detail
+ *  @param[in] state: Ptr to the driver structure
+ *  @param[in] enable: If true enable the interrupt if false disable the
+ *  interrupt
+ *
+ *  @return
+ *   The error code definition within the DDF
+ *   SNS_DDF_SUCCESS on success; Otherwise SNS_DDF_EBUS
+ *
+ */
+sns_ddf_status_e sns_dd_acc_bma2x2_config_high_low_g_int(
+        sns_dd_acc_bma2x2_state_t  *state,
+        bool                        enable)
+{
+    sns_ddf_status_e status = SNS_DDF_SUCCESS;
+    uint8_t bosch_write_buffer;
+    uint8_t rega;
+    uint8_t regv = 0x00;
+
+
+    bosch_write_buffer = 3;
+    if ((status = bma2x2_smbus_write_byte(state->port_handle,
+                    BMA2X2_RANGE_SEL_REG,
+                    &bosch_write_buffer)) != SNS_DDF_SUCCESS)
+    {
+        return status;
+    }
+
+  if ((status = bma2x2_set_low_g_threshold(state->port_handle, 10))
+       == SNS_DDF_SUCCESS)
+  {
+  }
+
```

```
+  if ((status = bma2x2_set_high_g_threshold(state->port_handle, 130))
+       == SNS_DDF_SUCCESS)
+  {
+  }
+
+  bosch_write_buffer = 0;
+  status = bma2x2_smbus_write_byte(state->port_handle,
+            BMA2X2_LOW_HIGH_HYST_REG,
+            &bosch_write_buffer);
+
+    /*! Perform read, modify & write to set high and low G interrupt bit
*/
+    if ((status = bma2x2_smbus_read_byte(state->port_handle,
+                    BMA2X2_INT_ENABLE2_REG,
+                    &bosch_write_buffer)) == SNS_DDF_SUCCESS)
+    {
+        if (enable)
+        {
+            /*! enable high and low G interrupt */
+            bosch_write_buffer |= BMA2X2_ACCEL_ENABLE_HIGH_LOW_G_VALUE;
+
+        }
+        else
+        {
+            /*! disable high and low G interrupt */
+            bosch_write_buffer &= ~(BMA2X2_ACCEL_ENABLE_HIGH_LOW_G_VALUE);
+
+        }
+
+        status = bma2x2_smbus_write_byte(state->port_handle,
+                BMA2X2_INT_ENABLE2_REG,
+                &bosch_write_buffer);
+    }
+
+
+    if (status == SNS_DDF_SUCCESS)
+    {
+        state->en_md = enable;
+    }
+
+#if BMA2X2_CONFIG_ENABLE_INT_MAP_2
+  rega = BMA2X2_REGA_INT_MAP_2;
+#else
+  rega = BMA2X2_REGA_INT_MAP_0;
```

```
+#endif
+   status = bma2x2_smbus_read_byte(state->port_handle,
+            rega, &regv);
+   BMA2X2_DD_CHECK_RETVAL(status, SNS_DDF_SUCCESS);
+
+   if (enable) {
+        regv = BST_SET_VAL_BIT(regv, 0);
+        regv = BST_SET_VAL_BIT(regv, 1);
+   } else {
+        regv = BST_CLR_VAL_BIT(regv, 0);
+        regv = BST_CLR_VAL_BIT(regv, 1);
+   }
+
+   status = bma2x2_smbus_write_byte(state->port_handle,
+            rega, &regv);
+   BMA2X2_DD_CHECK_RETVAL(status, SNS_DDF_SUCCESS);

     return status;
 }
```

Add code to set gpio45 wake up system in dts file

kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8937-mtp.dtsi

```
                sensor_wakeup {
                label = "Gsensor_wakeup";
                gpios = <&tlmm 45 0x1>;
                linux,input-type = <1>;
                linux,code = <249>;
                    gpio-key,wakeup;
                debounce-interval = <15>;
            };
```

kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8937-pinctrl.dtsi

```
   tlmm_gpio_key {
            gpio_key_active: gpio_key_active {
                mux {
                    pins = "gpio91", "gpio127", "gpio128", "gpio45";
                    function = "gpio";
                };

                config {
                    pins = "gpio91", "gpio127", "gpio128", "gpio45";
                    drive-strength = <2>;
```

```
                    bias-pull-up;
               };
          };

          gpio_key_suspend: gpio_key_suspend {
               mux {
                    pins = "gpio91", "gpio127", "gpio128", "gpio45";
                    function = "gpio";
               };

               config {
                    pins = "gpio91", "gpio127", "gpio128", "gpio45";
                    drive-strength = <2>;
                    bias-pull-up;
               };
```

Add code to set gpio45 wake up system in dts file

```
   kernel/msm-3.18/include/dt-bindings/input/input.h
   #define KEY_RFKILL      247  /* Key that controls all radios */
   #define KEY_MICMUTE     248  /* Mute / unmute the microphone */
   #define KEY_GSENSOR_WAKEUP   249
```

Set accelerometer g-range

```
3h -> ±2g range; 5h->±4g range; 8h->±8g range;
12h-> ±16g range; all other settings reserved (do not use)
This set is in function:
sns_ddf_status_e sns_dd_acc_bma2x2_config_high_low_g_int(
        sns_dd_acc_bma2x2_state_t  *state,
        bool                        enable)
{
   /* set G range value: 3,5,8,12 */
    bosch_write_buffer = 3;
    if ((status = bma2x2_smbus_write_byte(state->port_handle,
                    BMA2X2_RANGE_SEL_REG,
                    &bosch_write_buffer)) != SNS_DDF_SUCCESS)
    {
        return status;
}
```

Set low g threshold

```
sns_ddf_status_e sns_dd_acc_bma2x2_config_high_low_g_int(
        sns_dd_acc_bma2x2_state_t  *state,
        bool                        enable)
{
... ... ... ... ...
```

```
/* the setting range is 0 ~ 255 */
   if ((status = bma2x2_set_low_g_threshold(state->port_handle, 10))
        == SNS_DDF_SUCCESS)
   {
   }
}
```

Set high g threshold

```
sns_ddf_status_e sns_dd_acc_bma2x2_config_high_low_g_int(
        sns_dd_acc_bma2x2_state_t  *state,
        bool                       enable)
{
... ... ... ... ...
/* the setting range is 0 ~ 255 */
   if ((status = bma2x2_set_high_g_threshold(state->port_handle, 130))
        == SNS_DDF_SUCCESS){}
... ... ... ... ...
}
```

6 Test reports

6.1 MPSS test report

This section is not applicable to this release.

6.2 CNSS test report

This section is not applicable to this release.

6.3 APSS test report

This section is not applicable to this release.

6.3.1 Multimedia

This section is not applicable to this release.

6.3.2 Graphics

This section is not applicable to this release.

6.3.3 Camera

This section is not applicable to this release.

6.3.4 CBSP test report

This section is not applicable to this release.

6.3.5 Android platform test report

This section is not applicable to this release.

6.3.5.1 AU sanity report

This section is not applicable to this release.

6.3.5.2 AP stability report

This section is not applicable to this release.

6.3.6 CTS/CTSV/GTS report

This section is not applicable to this release.

6.4 Modem data KPI

This section is not applicable to this release.

6.5 Product stability report

This section is not applicable to this release.

6.6 Power dashboard

6.6.1 Modem

This section is not applicable to this release.

6.6.2 Multimedia

This section is not applicable to this release.

6.6.3 Thermal

This section is not applicable to this release.

6.6.4 Connectivity

This section is not applicable to this release.

6.7 Hardware bring up test report

This section is not applicable to this release.

6.8 APPS tools report

This section is not applicable to this release.

6.9 DOU test reports

This section is not applicable to this release.

6.10 Release sanity summary

Area	Test case	Result
BSP	USB enumeration	PASS
	SD card detection and verification	PASS
	EFS read/write	PASS
	DIAG, QPST, QXDM detection, and reset, Dump collection	PASS
	QCN backup/restore	PASS
	USB Charging Enable/Disable	PASS
	EDL using QFIL tool (Firehose protocol)	PASS
Android	Kernel up with 8 cores	PASS
	Android UI	PASS
	Full boot chain working with PIL (Modem, Venus, WCNSS)	PASS
	PMIC	PASS
	ADB, Fast Boot, Home Key/Power Key	PASS
Modem	NV Read/Write	PASS
Connectivity	WLAN: STA	PASS
	▪ Wi-Fi scanning and connection to AP	
	▪ Browsing	
	Bluetooth - Basic Functionality	PASS
	FM – ENABLE, DISABLE, TUNE, SEEK	PASS
GPS	GPS cold start and hot start	PASS
	A-GPS cold start and hot start	PASS
Multimedia	Audio playback	PASS
	Video playback	PASS
	Android UI with graphics	PASS
	Display	PASS
	Touch screen	PASS

A Product definition

A.1 MSM8937

Package Process Technology	12×14 mm2 non-PoP 28 nm LP
CPU	OctaCore - 8x ARM® Cortex™ A53 (4 x Performance Cluster ~ 1.4 GHz / 4 x Power Cluster ~ 1.1 GHz)
Memory	LPDDR3 800 MHz
Modem	Integrated X6 LTE LTE category 4 Up to 150 Mbps DL Up to 50 Mbps UL Downlink features: 2x10 MHz carrier aggregation 64-QAM Uplink features: 1x20 MHz Global Mode LTE FDD and TDD WCDMA (DC-HSDPA, DC-HSUPA) TD-SCDMA EV-DO and CDMA 1x GSM/EDGE Additional features include: LTE Broadcast LTE Dual SIM HD Voice over 3G and VoLTE Wi-Fi calling with LTE call continuity EVS Codec Support EVS IMS support 1xSCH + G and G + L Multi SIM support 1XSCH → G and G → L IRAT support
Location	Qualcomm® IZat™ Gen8C
GPU	Adreno 505 GPU OpenGL ES 3.1 OpenCL 2.0 full DirectX 11.2 HW Tessellation Geometry Shading

Display	Up to Full HD 1920x1200@60 fps UBWC 2x DSI 4 Lanes
Camera	Up to 21 MP camera and ZSL Dual Image Signal processor (ISP)
PMIC	PM8937, PMI8937/PMI8952
Video	Up to 1080p capture and playback 1080p@30 fps H.264 (AVC) H.265 (HEVC)
Audio	Fluence HD noise cancellation technology
Sensor	Hexagon ADSP based
Storage	eMMC 5.1 SD 3.0 (UHS-I)
Peripherals	1xUSB 2.0, SLIMBus
Security	Qualcomm Haven™ security suite: Snapdragon StudioAccess™ Content Protection Qualcomm® Safeswitch™ Technology Qualcomm® SecureMSM™ hardware and software foundation
BT/WLAN/FM	Qualcomm® VIVE™ 2-stream 802.11n/ac with MU-MIMO Bluetooth 4.1 Bluetooth Low Energy
NFC	Supported

B References

B.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Snapdragon Software Product Family user guide	80-P3255-31
MSM8937 Linux Android Software User Manual	SP80-P2485-4
MSM8937 Linux Android Software Debug Manual	SP80-P2485-5
MSM8937 Linux Android Software Porting Manual	SP80-P2485-6
Debug policy with serial number skip	80-P2485-39
Ensuring DSP Partition /dsp is Read-only	80-P5157-1
Hexagon Tools Installation Guide	80-VB419-25
USB Host Driver for Windows 2000/Windows XP User Guide	80-V4609-1
USB Host Driver Installation Instructions for Microsoft Windows	80-VP092-1
Qualcomm CreatePoint User Guide	80-NC193-2
Qualcomm Flash Image Loader (QFIL) User Guide	80-NN120-1
Hexagon LLVM C/C++ Compiler User Guide	80-VB419-89
LLVM Compiler For Hexagon Processor Deployment Plan	80-VB419-87
DSI Programing Guide for B-Family Android Devices	80-NA157-174
Linux Android Display Driver Porting Guide	80-NN766-1
WCN36X0 WLAN/BT/FM FTM Guide With QRCT Test Example	80-WL300-27
MSM8916 External MI2S Interface Overview	80-NL239-42
Linux Camera Debugging Guide	80-NL239-33
Widevine DRM	80-N9340-1
Subsystem Restart User Guide	80-N5609-2
MSM8x10 Android Subsystem Restart Overview	80-NC839-21
MSM8974 Android Subsystem Restart	80-NA157-31
SGLTE Device Configuration	80-NJ017-11
MSM8937 Linux Android Thermal Management Overview	80-P2485-13
Segment Loading Feature	80-NL239-46
MSM89x7 Modem Software Overview	80-P2485-12
MSM89x7 RF Software Overview	80-P2485-3
MSM8952/MSM8937 Audio Bringup Guide	80-NU323-11
Android Display Debug Guide	80-NP925-1
Multimedia Driver Development and Bringup Guide – Display	80-NU323-3
Multimedia Driver Development and Bringup Guide – Camera	80-NU323-2
Multimedia Driver Development and Bringup Guide – Video	80-NU323-5

Title	Number
Multimedia Video Debug Guide	80-NU339-1
TZ.BF.4.0 TrustZone Architecture Overview for MSM8937/MSM8953	80-P2485-21
MSM8937/MSM8953 Boot Architecture Overview	80-P2485-1
MSM8937 PMIC Linux Software Driver Overview	80-P2485-2
MSM8937 System Power Overview	80-P2485-4
MSM8937 Linux Android Audio Overview	80-P2485-5
MSM8937 Linux Android Camera Overview	80-P2485-6
MSM8937 Linux Android Display Overview	80-P2485-7
MSM8937 Linux Android Graphics Overview	80-P2485-8
MSM8937 Linux Android Video Overview	80-P2485-9
MSM8937 Android Camera KPIS Overview	80-P2485-10
MSM8937 Android Performance Overview	80-P2485-11
MSM89X7 RF Bringup User Guide	80-P2485-14
MSM89X7 RFC Customization User Guide	80-P2485-15
MSM89X7 Modem Data Services Overview	80-P2485-16
MSM8937 Linux USB Overview	80-P2485-17
MSM8937 System Drivers PMIC Overview	80-P2485-18
MSM8937/MSM8917 Software Architecture Overview	80-P2485-19
MSM8937 Clock Plan	80-P2485-20
MSM8937 Linux Audio KPIS	80-P2485-24
MSM8937 Software Migration Overview	80-P2485-26
MSM8937 Linux Android Wi-Fi Display Overview	80-P2485-27
MSM8937 PMIC Software Design Review	80-P2485-59
MSM8937 BSP Software Design Review Questionnaire	80-P2485-82
WCN36X0 WLAN/BT/FM Platform Development Guide	80-Y8887-1
QCA WCN36X0 Software Architecture	80-Y0513-1
RF Reference Schematic using RF360	80-NP237-49
WTR2965 Device Specification	80-NP237-1
RF chipset training/Design Guidelines	80-NP237-5
PMI8952 Device Specification	80-NT391-1
PMI8937 Device Specification	80-P2563-1
WCN3660B/3680B Device Specification	80-WL007-1
WCN3615 Device Specification	80-WL009-1
PMI8937 Device Revision Guide	80-P2563-4
PMI8952 Device Revision Guide	80-NT391-4
WCN3680B Device Revision Guide	80-WL007-4
WCN3615 Device Revision Guide	80-WL009-4
WCN3615 training/Design Guidelines	80-WL009-5A
MSM8937_LA_SW_Architecture (MSM8937 Chipset SW Overview)	80-P2485-19
MSM8937 Linux Android Software User Manual	80-P2485-4

Title	Number
MSM8937 Linux Android Software porting manual	80-P2485-6
MSM8937 Linux Android Software Migration document	80-P2485-26
MSM8937_Boot_Architecture_Overview	80-P2485-1
MSM8937_Clock_Plan	80-P2485-20
MSM8937_GPIO	80-P2468-1B
MSM8937_PMIC_SW_Driver_Overview	80-P2485-18
MSM8937_Peripherals_Overview	80-NA157-24 (K)
MSM8937 RPM Overview and Debug	80-NU154-10
MSM8937_System_Power_Overview	80-P2485-4
MSM8937 Security TrustZone QSEE Overview	80-P2485-21
MSM8937 Linux Android Software Debug overview	80-NV396-72
MSM8937/MSM8940 Thermal_Mgmt overview	80-P2485-13
MSM8937/MSM8917/MSM8940 Audio Overview	80-P2485-5
MSM8937/MSM8917/MSM8940 Video Overview	80-P2485-9
MSM8937/MSM8917/MSM8940 Camera Overview	80-P2485-6
MSM8937/MSM8917/MSM8940 Display Overview	80-P2485-7
MSM8937/MSM8917/MSM8940 Graphics Overview	80-P2485-8
MSM8937/MSM8917/MSM8940 Android Camera KPIs Overview	80-P2485-10
MSM8937/MSM8917/MSM8940 Linux Android Wi-Fi Display Overview	80-P2485-27
MSM8937/MSM8917/MSM8940 Android Video KPIs Overview	80-P2485-35
MSM8937/MSM8917/MSM8940 Android Audio KPIs Overview	80-P2485-24
MSM8937/MSM8917/MSM8940 Audio Quick Start	80-P2485-34
MSM8937/MSM8917/MSM8940 Video Quick Start	80-P2485-36
MSM8937/MSM8917/MSM8940 Camera Quick Start	80-P2485-37
MSM8937/MSM8917/MSM8940 Display Quick Start	80-P2485-38
WCN36x0_Connectivity_SW_Overview	80-Y0513-1
WCN36X0 WLAN-BT-FM Platform Development guide	80-Y8887-1
WCN36x0 Software Debug Guide	80-Y0513-2
Bluetooth Debug Guide	80-Y8113-9
MSM8937 Multi Button Headset Control	80-NL239-27
PMIC_SBL_API_Interface_Spec_ PM8937/PM8940	80-NV610-42
MSM8937/MSM8040 Linux USB Overview	80-P2485-17
Resources	
Android Open Source Project Page	http://source.android.com/
Android Developer Resources	http://developer.android.com/index.html
Android Source Download and System Setup	http://source.android.com/source/index.html
Code Aurora Forum	https://www.codeaurora.org/
Installing Repo	http://source.android.com/source/downloading.html
Qualcomm ChipCode Website	https://chipcode.qti.qualcomm.com/

Apache license

When applying a patch to a file that is licensed under the Apache License, Version 2.0, be aware of Section 4(b) thereof, which states:

"b. You must cause any modified files to carry prominent notices stating that You changed the files; and"

Nothing in this release notes file changes any terms and conditions of any agreement you have with Qualcomm Technologies, Inc.

OZPLA license

Some portions of the software distributed in this release are subject to the OZPLA license agreement, which is as follows.

Australian Public License A (OZPLA)

Version 0-1

Developed by: Embedded, Real-Time and Operating Systems Program (ERTOS)

National ICT Australia Ltd (NICTA)

http://ertos.nicta.com.au

Permission is granted by NICTA, free of charge, to any person obtaining a copy of this software and any associated documentation files (the "Software") to deal with the Software without restriction, including (without limitation) the rights to use, copy, modify, adapt, merge, publish, distribute, communicate to the public, sublicense, and/or sell, lend or rent out copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- Neither the names of NICTA, nor the names of its contributors, may be used to endorse or promote products derived from this Software without specific prior written permission.

EXCEPT AS EXPRESSLY STATED IN THIS LICENCE AND TO THE FULL EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE IS PROVIDED "AS-IS", AND NICTA AND ITS CONTRIBUTORS MAKE NO REPRESENTATIONS, WARRANTIES OR CONDITIONS OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY REPRESENTATIONS, WARRANTIES OR CONDITIONS REGARDING THE CONTENTS OR ACCURACY OF THE SOFTWARE, OR OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, THE ABSENCE OF LATENT OR OTHER DEFECTS, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE.

TO THE FULL EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL NICTA OR ITS CONTRIBUTORS BE LIABLE ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHERWISE) FOR ANY CLAIM, LOSS, DAMAGES OR OTHER LIABILITY, INCLUDING (WITHOUT LIMITATION) LOSS OF PRODUCTION OR OPERATION TIME, LOSS, DAMAGE OR CORRUPTION OF DATA OR RECORDS; OR LOSS OF ANTICIPATED SAVINGS, OPPORTUNITY, REVENUE, PROFIT OR GOODWILL, OR OTHER ECONOMIC LOSS; OR ANY SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES, ARISING OUT OF OR IN CONNECTION WITH THIS LICENCE, THE SOFTWARE OR THE USE OF OR OTHER DEALINGS WITH THE SOFTWARE, EVEN IF NICTA OR ITS CONTRIBUTORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH CLAIM, LOSS, DAMAGES OR OTHER LIABILITY.

If applicable legislation implies representations, warranties, or conditions, or imposes obligations or liability on NICTA or one of its contributors in respect of the Software that cannot be wholly or partly excluded, restricted or modified, the liability of NICTA or the contributor is limited, to the full extent permitted by the applicable legislation, at its option, to:

a. in the case of goods, any one or more of the following:

i. the replacement of the goods or the supply of equivalent goods;
ii. the repair of the goods;
iii. the payment of the cost of replacing the goods or of acquiring equivalent goods;
iv. the payment of the cost of having the goods repaired; or

b. in the case of services:

i. the supplying of the services again; or
ii. the payment of the cost of having the services supplied again.

The construction, validity and performance of this license shall be governed by the laws in force in the State of New South Wales, Australia.

OZPLB License

Some portions of the software distributed in this release are subject to the OZPLB license agreement, which is as follows.

Australian Public Licence B (OZPLB)

Version 1-0

Developed by: Embedded, Real-time and Operating Systems Program (ERTOS)

National ICT Australia

http://www.ertos.nicta.com.au

Permission is granted by National ICT Australia, free of charge, to any person obtaining a copy of this software and any associated documentation files (the "Software") to deal with the Software without restriction, including (without limitation) the rights to use, copy, modify, adapt, merge, publish, distribute, communicate to the public, sublicense, and/or sell, lend or rent out copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the name of National ICT Australia, nor the names of its contributors, may be used to endorse or promote products derived from this Software without specific prior written permission.

EXCEPT AS EXPRESSLY STATED IN THIS LICENCE AND TO THE FULL EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE IS PROVIDED "AS-IS", AND NATIONAL ICT AUSTRALIA AND ITS CONTRIBUTORS MAKE NO REPRESENTATIONS, WARRANTIES OR CONDITIONS OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY REPRESENTATIONS, WARRANTIES OR CONDITIONS REGARDING THE CONTENTS OR ACCURACY OF THE SOFTWARE, OR OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, THE ABSENCE OF LATENT OR OTHER DEFECTS, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE.

TO THE FULL EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL NATIONAL ICT AUSTRALIA OR ITS CONTRIBUTORS BE LIABLE ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHERWISE) FOR ANY CLAIM, LOSS, DAMAGES OR OTHER LIABILITY, INCLUDING (WITHOUT LIMITATION) LOSS OF PRODUCTION OR OPERATION TIME, LOSS, DAMAGE OR CORRUPTION OF DATA OR RECORDS; OR LOSS OF ANTICIPATED SAVINGS, OPPORTUNITY, REVENUE, PROFIT OR GOODWILL, OR OTHER ECONOMIC LOSS; OR ANY SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES, ARISING OUT OF OR IN CONNECTION WITH THIS LICENCE, THE SOFTWARE OR THE USE OF OR OTHER DEALINGS WITH THE SOFTWARE, EVEN IF NATIONAL ICT AUSTRALIA OR

ITS CONTRIBUTORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH CLAIM, LOSS, DAMAGES OR OTHER LIABILITY.

If applicable legislation implies representations, warranties, or conditions, or imposes obligations or liability on National ICT Australia or one of its contributors in respect of the Software that cannot be wholly or partly excluded, restricted or modified, the liability of National ICT Australia or the contributor is limited, to the full extent permitted by the applicable legislation, at its option, to:

a. in the case of goods, any one or more of the following:
   i. the replacement of the goods or the supply of equivalent goods;
   ii. the repair of the goods;
   iii. the payment of the cost of replacing the goods or of acquiring equivalent goods;
   iv. the payment of the cost of having the goods repaired; or

b. in the case of services:
   i. the supplying of the services again; or
   ii. the payment of the cost of having the services supplied again.

The construction, validity and performance of this license is governed by the laws in force in New South Wales, Australia.

OKL4 software

This product contains OKL4 software, which is:

Copyright(C) 2009 Open Kernel Labs

Copyright(C) 2004-2006 National ICT Australia (NICTA)

Copyright(C) 2002-2003 University of New South Wales

Copyright(C) 2002-2004 Karlsruhe University

OKL4 are subject to a license nominated by the relevant copyright holder. License terms nominated by the copyright holders listed above are available from http://ok-labs.com/licenses.