

cādence[®]

HiKey960 HiFi3 Android SDK

ALPHA SOFTWARE

© 2018 Cadence Design Systems, Inc.

All rights reserved worldwide

This publication is provided “AS IS.” Cadence Design Systems, Inc. (hereafter “Cadence”) does not make any warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Information in this document is provided solely to enable system and software developers to use our processors. Unless specifically set forth herein, there are no express or implied patent, copyright or any other intellectual property rights or licenses granted hereunder to design or fabricate Cadence integrated circuits or integrated circuits based on the information in this document. Cadence does not warrant that the contents of this publication, whether individually or as one or more groups, meets your requirements or that the publication is error-free. This publication could include technical inaccuracies or typographical errors. Changes may be made to the information herein, and these changes may be incorporated in new editions of this publication.

2018 Cadence, the Cadence logo, Allegro, Assura, Broadband Spice, CDNLive!, Celtic, Chipestimate.com, Conformal, Connections, Denali, Diva, Dracula, Encounter, Flashpoint, FLIX, First Encounter, Incisive, Incyte, InstallScape, NanoRoute, NC-Verilog, OrCAD, OSKit, Palladium, PowerForward, PowerSI, PSpice, Purespec, Puresuite, Quickcycles, SignalStorm, Sigrity, SKILL, SoC Encounter, SourceLink, Spectre, Specman, Specman-Elite, SpeedBridge, Stars & Strikes, Cadence, TripleCheck, TurboXim, Vectra, Virtuoso, VoltageStorm Explorer, Xtensa, and Xtreme are either trademarks or registered trademarks of Cadence Design Systems, Inc. in the United States and/or other jurisdictions.

ARM is a registered trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. AMBA, CoreSight, and Cortex are either trademarks or registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

All other trademarks are the property of their respective holders.

Issue Date: 04/2018

Cadence Design Systems, Inc.
2655 Seely Ave.
San Jose, CA 95134
www.cadence.com

HiKey960 HiFi3 Android SDK

HiKey960-HiFi3-Android-SDK-v0.7_Alpha.tgz SDK Package contents are as below

- README: Describes the package contents
- release.txt: Release and version information
- config: HiFi3 Config for Linux and Windows
- docs: Documentation

The following sections describe how to use these SDK packages contents.

Xtensa Audio Framework Brief

The Xtensa Audio Framework, provides a standard software framework to interface audio, voice, and sensor functions running on the HiFi DSP to higher level applications such as media players or recorders. It enables high-level application programmers using Java/C++ to leverage the power-savings and acceleration benefits of the HiFi DSP without having to understand the low-level programming details of a real-time DSP. It also provides OEMs and application developers faster time to market when integrating Tensilica® HiFi DSP based SoCs into their products.

The Xtensa Audio Framework is composed of a DSP Framework and an Application Framework.

DSP Framework

- Implements and runs the data processing chain
- Does all memory management required for the data processing chain
- Receives commands from the Application Framework and sends responses

Application Framework

- Creates, configures and controls the data processing chain on DSP
- Supports runtime creation and tear down of the data processing chains
- Sends commands to the DSP Framework and receives responses

Developer APIs available in Application Framework provide a very simple interface to create, configure and control the data processing chains on DSP. These APIs abstract offloading of audio processing tasks to the DSP while hiding details of memory management and input and output streaming.

The Xtensa Audio Framework supports two main use cases.

1. Hostless Xtensa Audio Framework:

The first use case is targeted at SoCs where the HiFi DSP is the main processor. An example product is a wearable device such as a Bluetooth headset.

2. Hosted Xtensa Audio Framework:

The second use case is targeted at SoCs where a host application processor offloads audio and/or voice tasks to the HiFi DSP processor. Examples include codec chips for ultra-low-power, glitch-free audio playback, noise suppression and echo-cancellation for superior voice experience.

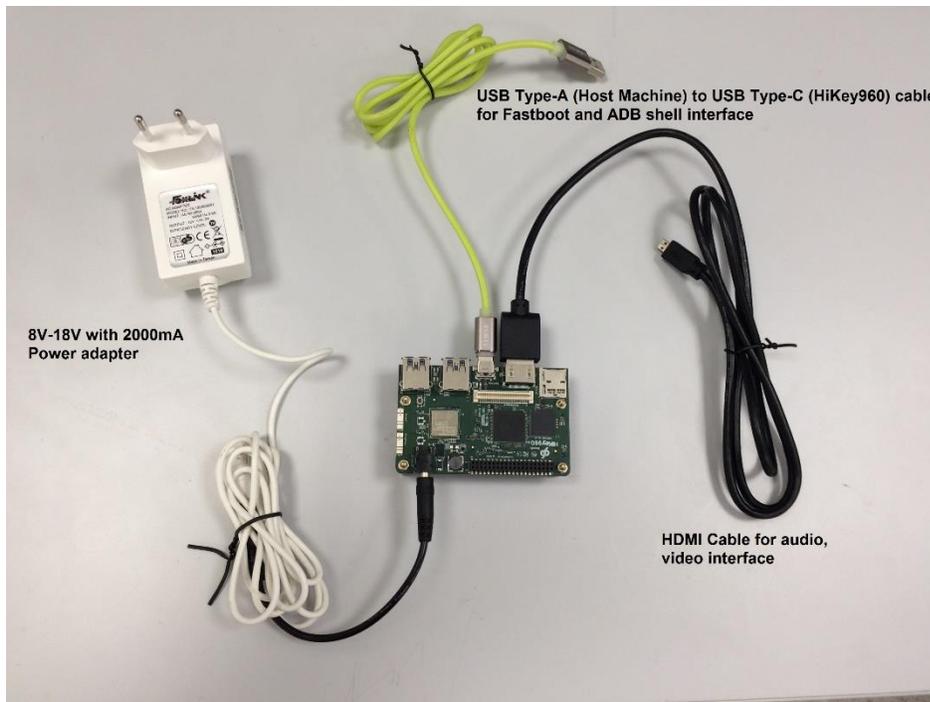
Xtensa Audio Framework on HiKey960 with Android demonstrates the second use case i.e. Hosted Xtensa Audio Framework.

HiKey960 Board Brief

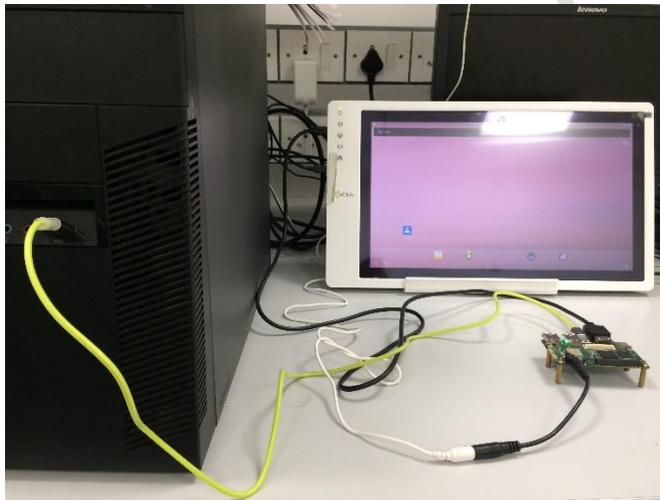
The HiKey960 development platform is based around the **Huawei Kirin 960** processor (*Hi3660*), which embeds Audio subsystem with one **Tensilica HiFi 3 DSP**. The board is provided by Archermind and LeMaker through multiple channels. Initial software support for the board is provided in the AOSP source tree based on the Android Common Kernel using the Linux 4.9 kernel release. A brief introduction about the Hikey960 Board and buying details can be found here: <https://www.96boards.org/product/hikey960/>

How to set up HiKey960 Board

Please refer to following picture to know the cables required to flash images, run ADB shell interface and run Android on HiKey960 Board. Note, these cables are not provided with HiKey960 Board and you would need to arrange them separately.



HiKey960 Board generally comes with pre-loaded Android. You should be able to connect the board and run Android right away. Refer to following picture for connections.



Note, instructions given in following sections about working with board (flashing images using fastboot, running applications using adb shell), assume that you have required adb tools on your host machine. These tools are available with AOSP source tree and these instructions will readily run from your Ubuntu host machine (required to build Android). If you are connecting the HiKey960 board to Windows host machine, you may need to install these tools separately. See this section for instructions.

How to build Android for HiKey960 Board

Follow instructions from the below link to build system and kernel images

<https://source.android.com/setup/build/devices>

Build XAF DSP Firmware

Please register from the below link to get Xtensa tools to build DSP firmware

<https://www.tensilicatools.com/platform/huawei-kirin-960/>

After successful installation of Xtensa tools go to

```
$cd AOSP_ROOTDIR/device/Linaro/hikey/hifi/xaf/hifi-dpf/build_hikey/
```

Note, the XAF DSP firmware build is supported only on Linux systems. Make sure you have installed RG2017.5 Xtensa tools and hifi3_hikey960 core.

1. Set-up environment variables as:

```
$ setenv XTENSA_CORE hifi3_hikey960
$ setenv XTENSA_SYSTEM $INSTALL_DIR/xtensa/XtDevTools/install/tools/RG-
2017.5-linux/XtensaTools/config

$ setenv PATH $XTENSA_SYSTEM/./bin:$PATH
```

Above instructions are for csh. Use respective instructions for bash, if you are using bash shell to build XAF DSP firmware.

2. Go to

```
$cd AOSP_ROOT_DIR/device/linaro/hikey/hifi/xaf/hifi-dpf/build_hikey/
$ make clean all
```

It creates `xa_hikey.img` in the current directory.

3. Copy `xa_hikey.img` DSP firmware image from the above build steps to AOSP source tree as following (note, this would overwrite default `hifi-hikey960.img`):

```
$ cp xa_hikey.img to
AOSP_ROOT_DIR/device/linaro/hikey/hifi/firmware/hifi-hikey960.img
```

4. To include `hifi-hikey960.img` firmware, rebuild AOSP tree as following
\$ cd \$AOSP_ROOT_DIR
\$ make -j32

Flash the new AOSP images to HiKey960

1. Enter fastboot mode by turning ON switch 1 and 3.
2. Flash images by running the following commands:

```
fastboot flash boot AOSP_ROOT_DIR/out/target/product/hikey960/boot.img
fastboot flash dts AOSP_ROOT_DIR/out/target/product/hikey960/dt.img
fastboot flash system AOSP_ROOT_DIR/out/target/product/hikey960/system.img
fastboot flash cache AOSP_ROOT_DIR/out/target/product/hikey960/cache.img
fastboot flash userdata AOSP_ROOT_DIR/out/target/product/hikey960/userdata.img
```

3. Turn OFF switch 3 and power on the board.

xa_hikey.img details

DSP firmware includes following Audio processing components from Cadence

- Ogg-Vorbis Decoder (v1.11p3)
- PCM-Gain component (implementation available in XAF DSP firmware source code)
- Mixer component (implementation available in XAF DSP firmware source code)

Build XAF Host Application

The SDK package contains `xaf-host-v0.7_Alpha.tgz`

1. Set-up environment variables as:

```
$ cd $AOSP_ROOT_DIR
$ . ./build/envsetup.sh
$ lunch hikey960-userdebug
```

2. To build XAF Host application, execute following commands

```
$ cd AOSP_ROOT_DIR/device/linaro/hikey/hifi/xaf/host-apf/
$ mm
```

3. It creates `xaf-dec-test` and `xaf-dec-mix-test` binaries in

`$AOSP_ROOT_DIR/out/target/product/hikey960/system/bin/` directory

4. Push `xaf-dec-test` and `xaf-dec-mix-test` binaries to Hikey960

```
$ adb root
$ adb remount
$ adb push $AOSP_ROOT_DIR/out/target/product/hikey960/system/bin/xaf-
dec-test /system/bin/
$ adb push $AOSP_ROOT_DIR/out/target/product/hikey960/system/bin/xaf-
dec-mix-test /system/bin/
```

Note: `dhifimesg` (utility for printing HiFi3 DSP execution logs) is available in XAF Host application `AOSP_ROOT_DIR/device/linaro/hikey/hifi/xaf/host-apf/tools/` directory.

```
$ adb push dhifimesg /system/bin
```

Run XAF Host Application

```
$ adb push <test-Ogg1-file> /sdcard/Music/  
$ adb push <test-Ogg2-file> /sdcard/Music/  
$ adb push <test-PCM-file> /sdcard/Music/  
$ adb shell  
# cd /system/bin/  
# xaf-dec-test /sdcard/Music/<test-Ogg1-file>  
# xaf-dec-test /sdcard/Music/<test-PCM-file>  
# xaf-dec-mix-test /sdcard/Music/<test-Ogg1-file> /sdcard/Music/<test-Ogg2-file>  
# xaf-dec-mix-test /sdcard/Music/<test-Ogg1-file> /sdcard/Music/<test-PCM-file>
```

The steps above, should playback the audio.

To view XAF Host application logs, use following command:

```
$ logcat
```

To view HiKey IPC driver logs, use following command:

```
$ dmesg |grep -i hifi
```

To view DSP Firmware logs, use following command:

```
$ dhifimesg
```

xaf-dec-test Application details

- Configured to playback 16 bit, stereo PCM data at 48 kHz sample rate
- Supports Ogg-Vorbis decode, PCM-Gain playback
- Accepts `.ogg` or `.pcm` as input file
- Plays back decoded output through audio device connected through HDMI interface
- Also writes PCM output to `dec-out.pcm` file

xaf-dec-mix-test Application details

- Configured to playback 16 bit, stereo PCM data at 48 kHz sample rate
- Supports Ogg-Vorbis decode, PCM-Gain playback and mixing
- Accepts two input files *.ogg* or *.pcm* as input
- Plays back decoded and mixed output through audio device connected through HDMI interface
- Writes PCM output to *dec-mix-out.pcm* file

Refer to application source files available in `$AOSP_ROOT_DIR/external/xaf/host-apf/utest/` directory for more details.

More about Xtensa Audio Framework on HiKey960 Board

Host and HiFi 3 DSP shared memory

Following is the memory map of shared memory between Host and DSP. Note, 256 kB from 'Music data' area (starting from 0x8B300000) is used as shared memory for IPC communication (messages and audio data) between Host and DSP.

```
/**Non Secure 3.5M **/  
/*  
|0x8B300000|0x8B432000|0x8B532000|0x8B5B1000|0x8B5B2000|0x8B5C5000|0x8B5C6000|0x8B5C7000|0x8B5F9800|~0x8B609800~|~0x8B618800~|~0x8B618880~|0x8B627880|~0x8B629880~|0x8B62C880~|~| */  
/* |Music data|~PCM data|~hifi uart|panicstack|icc debug~|flag data~|DDRsethead|~AP NV  
~|AP&HIFIMB~|codec dma buff|codecdmaconfig|soundtrigger|pcm upload|~hikey share|unsec  
reserve| */  
/*  
|~~~~1.2M~~|~~~~1M~~~~|~~508k~~~~|~~~~4k~~~|~~76k~~~~|~~4k~~~~|~~4k~~~~  
~|~~202k~~~~|~~64k~~~~|~~60k~~~~|~~128b~~~~|~~60k~~~~|~~8k~~~~  
|~~~~12k~~~~|~~334k-128b~~| */  
/*  
|0x8B431fff|0x8B531fff|0x8B5B0fff|0x8B5B1fff|0x8B5C4fff|0x8B5C5fff|0x8B5C6fff|0x8B5F9  
7ff|0x8B6097ff|~0x8B6187FF~|~0x8B61887F~|~0x8B62787F~|0x8B62987F|0x8B62C87F  
~|~0x8B67FFFF~| */  
  
/** Secure9.5M **/  
/* |~~0x89200000~~|~~0x89800000~~|~~0x89830000~~|~~0x89864000~~| */  
/* |~~HIFI RUNNING~~|~OCRAM img bak~~|~TCM img bak~~|~~IMG bak~~~~| */  
/* |~~~~6M~~~~|~~~~192K~~~~|~~~~208k~~~~|~~~~3.1M ~~~~| */  
/* |~~0x897fffff~~|~~0x8982ffff~~|~~0x89863fff~~|~~0x89B80000~~| */
```

Notes about how XAF DSP Firmware gets loaded on HiFi3 runtime

Audio HAL (audio_hw.c) from userspace as part of audio device open adev_open(), opens HiKey IPC driver.

While opening HiKey IPC driver, it calls `load_hifi_img_by_misc()` function to load XAF DSP Firmware on to HiFi3 core. The said function does following:

- Read XAF DSP Firmware image from `/system/etc/firmware/hifi/hifi.img`
- Load sections onto DSP instruction memory

Known Issues / Limitations of XAF

Following is the known issue or limitation of XAF which will be resolved in next release.

- FATAL errors from XAF component (e.g. Vorbis Decoder) are ignored and may cause XAF Host application to hang.

Guidelines to Set up HiKey960 Board with Windows Host Machine

Following instructions are provided as guidelines for setting up HiKey960 Board with Windows host machine.

1. Installing Google USB driver
 1. Download Google USB driver from <https://developer.android.com/studio/run/win-usb.html>
 2. Install the same as per steps in <https://developer.android.com/studio/run/oem-usb.html#InstallingDriver>
2. Installing ADB tools - adb is included in the Android SDK Platform-Tools package.
 1. Download the standalone Android SDK Platform-Tools package, from <https://developer.android.com/studio/releases/platform-tools.html> for Windows.
 2. Unzip the downloaded file and Change the directory to where adb executable is (...platform-tools-latest-windows\platform-tools) or add this directory to PATH.