

USER GUIDE FOR SMARTPHONE BASED REAL-TIME ACOUSTIC FEEDBACK CANCELLATION SYSTEM - ANDROID

Kashyap Patel, Issa Panahi
STATISTICAL SIGNAL PROCESSING LABORATORY (SSPRL)
UNIVERSITY OF TEXAS AT DALLAS

APRIL 2019

This work was supported by the National Institute of the Deafness and Other Communication Disorders (NIDCD) of the National Institutes of Health (NIH) under the award number 1R01DC015430-01. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

Table of Contents

1. SOFTWARE TOOLS.....	4
2. BUILD AN ANDROID APP	5
2.1 Programming Language	5
2.2 Creating Android Apps.....	5
2.3 Adding C File.....	7
3. ACOUSTIC FEEDBACK CANCELLATION APPLICATION.....	10
4. REFERENCES.....	11

INTRODUCTION

The ‘Acoustic Feedback Cancellation (AFC)’ app is designed to cancel the acoustic feedback arising due to the acoustic coupling between the speaker and microphone in real time on smartphone. The application is trained to perform in noisy conditions as well. The contents of this user guide gives you the steps to implement the ‘Acoustic Feedback Cancellation’ algorithm on Android devices and the steps to be followed after installing the app on the smartphone. This app will be an open source and portable research platform for hearing improvement studies.

This user guide covers the software tools required for implementing the algorithm, how to run C codes on Android devices and usage of other tools that are quite helpful in creating audio apps for audio playback in real time. The MATLAB and C codes used for the ‘Acoustic Feedback Cancellation’ algorithm are made available publicly on the following website: <http://www.utdallas.edu/ssprl/hearing-aid-project/>. The codes can be accessed and used with proper consent of the author for further improvements in research activities related to hearing aids. The screenshot of the first look of our app is as shown in Figure 1.

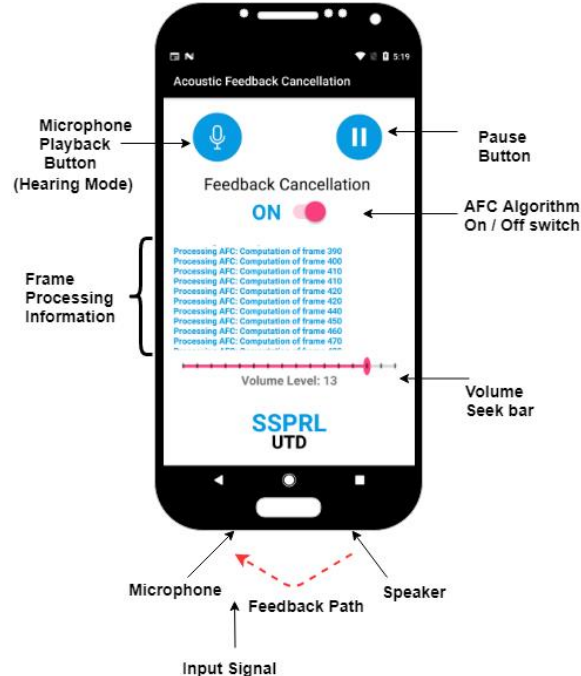


Figure 1. Screenshot of the AFC Application

1. SOFTWARE TOOLS

Android is an open-source operating system developed by Google for mobile phones and tablets. The Android apps are usually coded in Java. In this section, it is shown how to set up the Android Studio IDE (Integrated Development Environment) for developing Android apps.

Android Studio IDE requires either Windows Operating System or Apple Operating System. Android Studio IDE can directly build and upload source codes into Android smartphone or generate a APK file which can be downloaded and installed on the Android smartphone.

To download the latest version of Android Studio

1. Open the Android Studio website to download.
(<https://developer.android.com/studio/index.html>)
2. Click Download Android Studio Button (Figure 2).
3. Install the execution file after download.

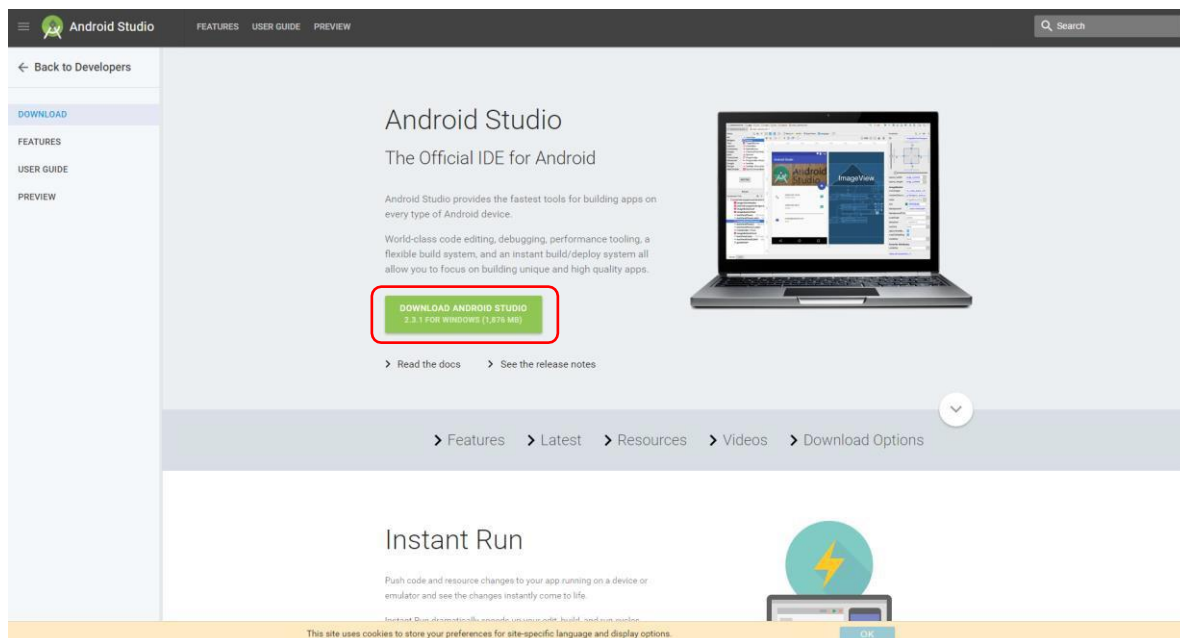


Figure 2

2. BUILD AN ANDROID APP

2.1 Programming Language

For creating Android apps, Java is used to create the required shell. The Java Development Kit (JDK) needs to be firstly installed on your computer. This link contains the latest version of JDK: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2.2 Creating Android Apps

After installations of Android Studio and JDK completed, android apps can be created using Android Studio.

1. Open Android Studio.
2. Under the "Quick Start" menu, select "Start a new Android Studio project." (Figure 3)
3. On the "Create New Project" window that opens, name your project "HelloWorld".(Figure 4)
4. If you choose to, set the company name as desired*.
5. Note where the project file location is and change it if desired.
6. Click "Next."
7. Make sure on that "Phone and Tablet" is the only box that is checked. (Figure 4)
8. If you are planning to test the app on your phone, make sure the minimum SDK is below your phone's operating system level.
9. Click "Next."
10. Select "Blank Activity." (Figure 4)
11. Click "Next."
12. Leave all of the Activity name fields as they are. (Figure 4)
13. Click "Finish."

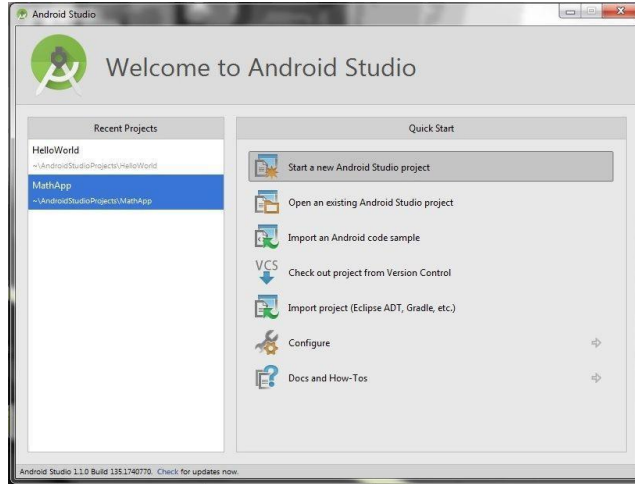


Figure 3

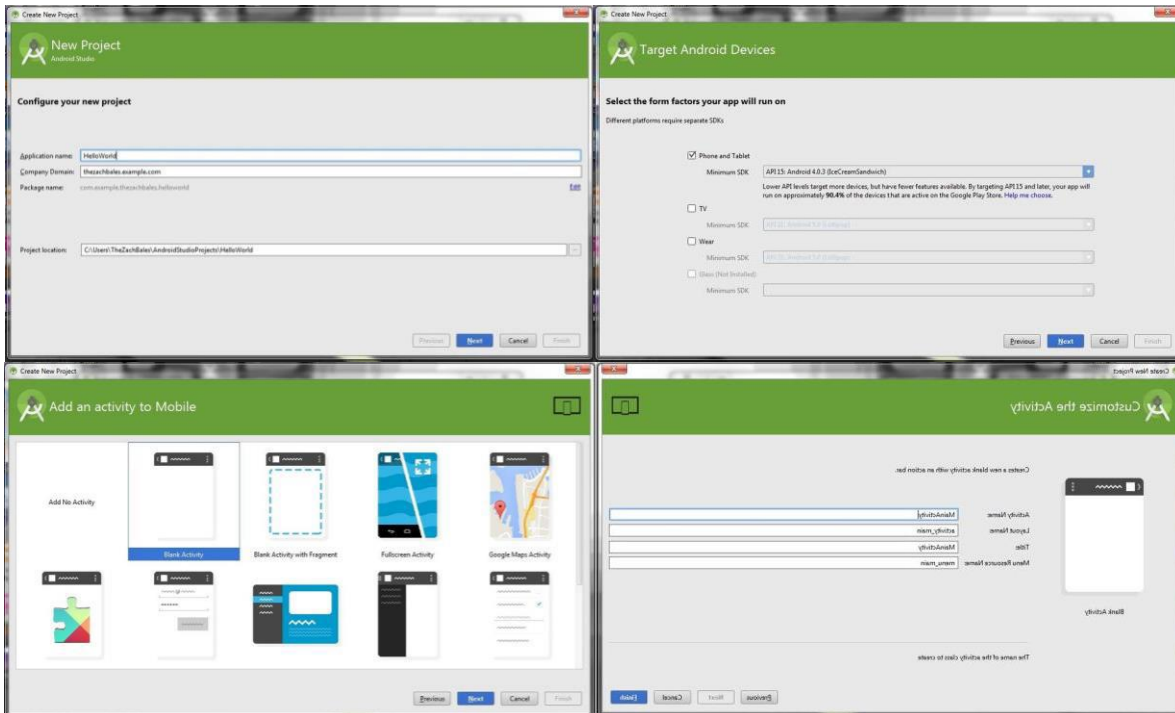


Figure 4

2.3 Adding C File

Android apps are typically written in Java, with its elegant object-oriented design. However, at times, you need to overcome the limitations of Java, such as memory management and performance, by programming directly into Android native interface. Android provides Native Development Kit (NDK) to support native development in C/C++, besides the Android Software Development Kit (Android SDK) which supports Java.

2.3.1 Installing the Native Development Kit (NDK)

1. Menu "Tools" > "Android" > "SDK Manager" (Figure 5)
2. Select tab "SDK Tools"
3. Check "Android NDK"[or "NDK"] if it is not checked
4. Sync and re-build the project.

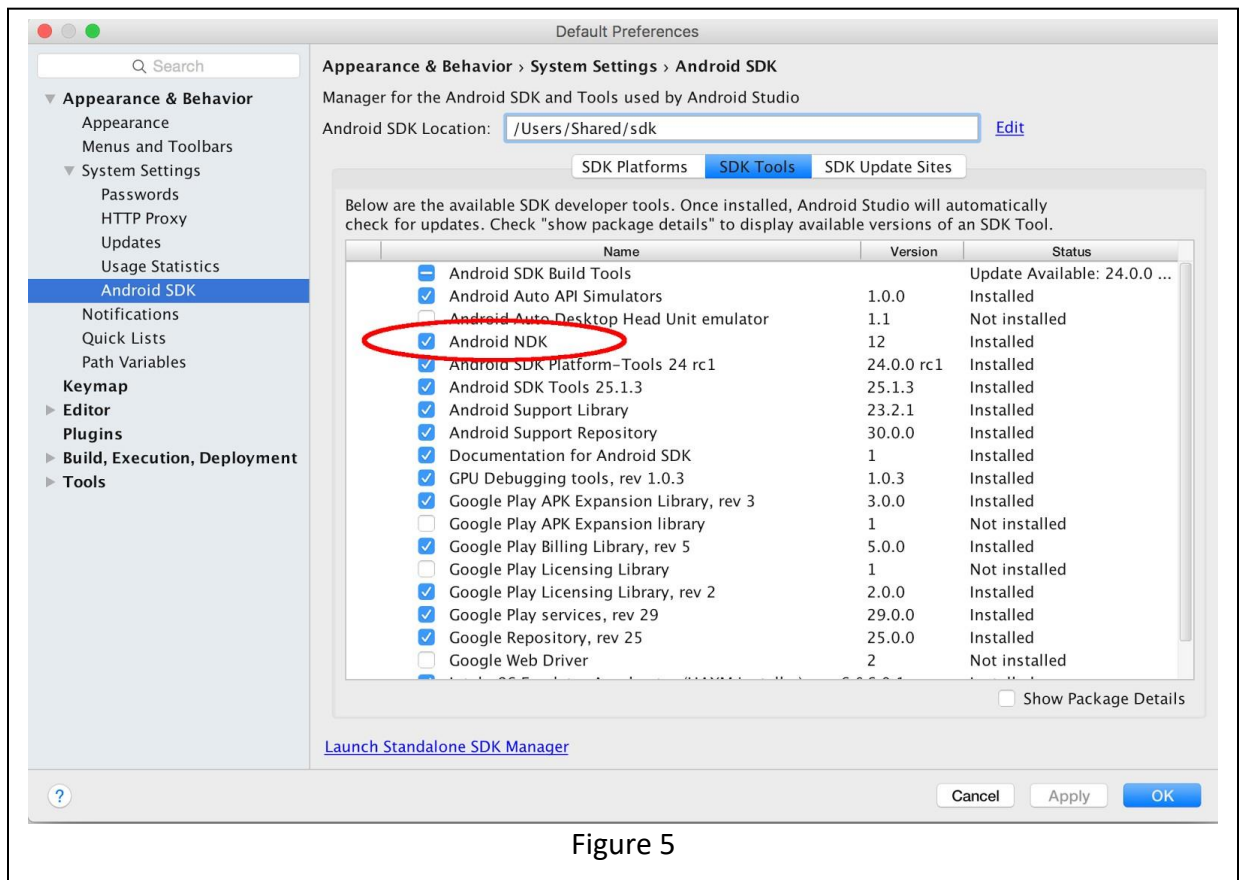


Figure 5

2.3.2 Writing a Hello-World Android NDK Program

Creating a new project with support for native code is similar to creating any other Android Studio project, but there are a few additional steps:

1. In the Configure your new project section of the wizard, check the Include C++ Support checkbox.
2. Click Next.
3. Complete all other fields and the next few sections of the wizard as normal.
4. In the Customize C++ Support section of the wizard, you can customize your project with the following options:
 - C++ Standard: use the drop-down list to select which standardization of C++ you want to use. Selecting Toolchain Default uses the default CMake setting.
 - Exceptions Support: check this box if you want to enable support for C++ exception handling. If enabled, Android Studio adds the `-fexceptionsflag` to `cppFlags` in your module-level `build.gradle` file, which Gradle passes to CMake.
 - Runtime Type Information Support: check this box if you want support for RTTI. If enabled, Android Studio adds the `-frtti` flag to `cppFlags` in your module-level `build.gradle` file, which Gradle passes to CMake.
5. Click Finish.

After Android Studio finishes creating your new project, open the Project pane from the left side of the IDE and select the Android view. As shown in figure 6, Android Studio adds the `cpp` and `External Build Files` groups:

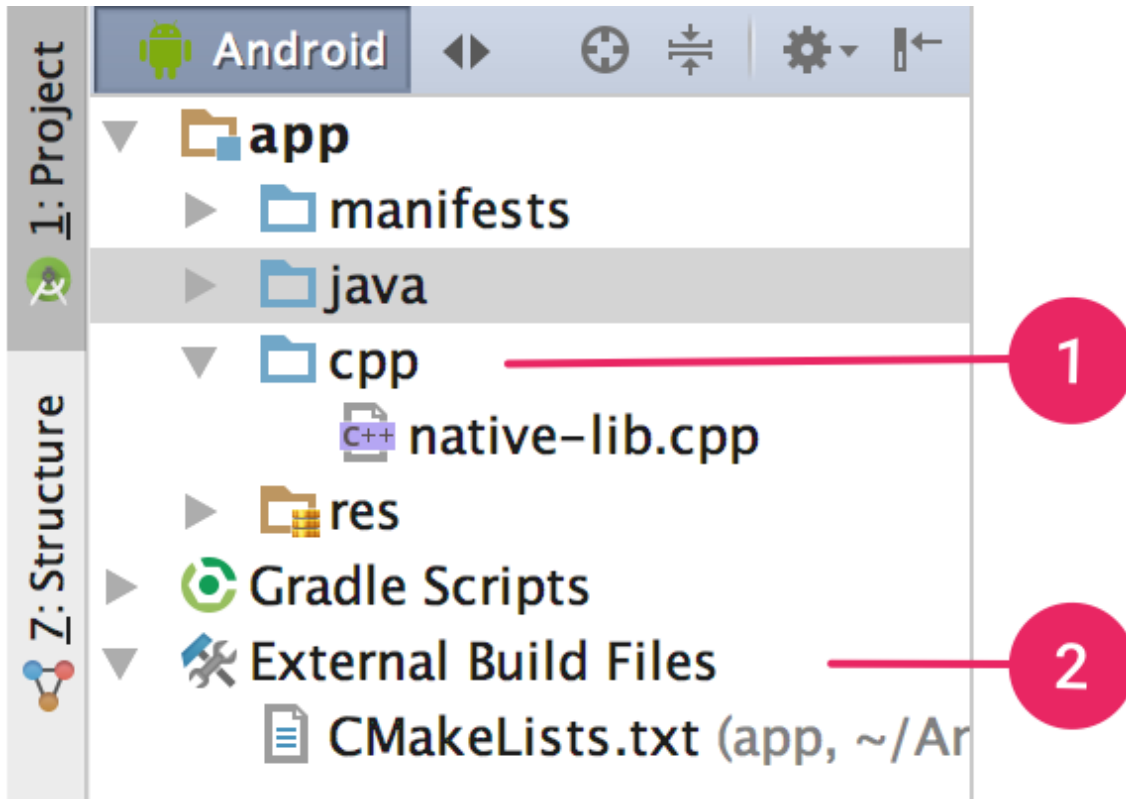


Figure 6

1. The `cpp` group is where you can find all the native source files, headers, and prebuilt libraries that are a part of your project. For new projects, Android Studio creates a sample C++ source file, `native-lib.cpp`, and places it in the `src/main/cpp/` directory of your app module. This sample code provides a simple C++ function, `stringFromJNI()`, that returns the string “Hello from C++”.
2. The External Build Files group is where you can find build scripts for CMake or `ndk-build`. Similar to how `build.gradle` files tell Gradle how to build your app, CMake and `ndk-build` require a build script to know how to build your native library. For new projects, Android Studio creates a CMake build script, `CMakeLists.txt`, and places it in your module’s root directory.

3. ACOUSTIC FEEDBACK CANCELLATION APPLICATION

The developed algorithm demonstrates the real-time operation to cancel the negative effects of acoustic feedback arising in the hearing aid devices due to the coupling between the speaker and the microphone. The application is trained to perform in noisy conditions as well. See Figure 7 for a screenshot of the Android application.

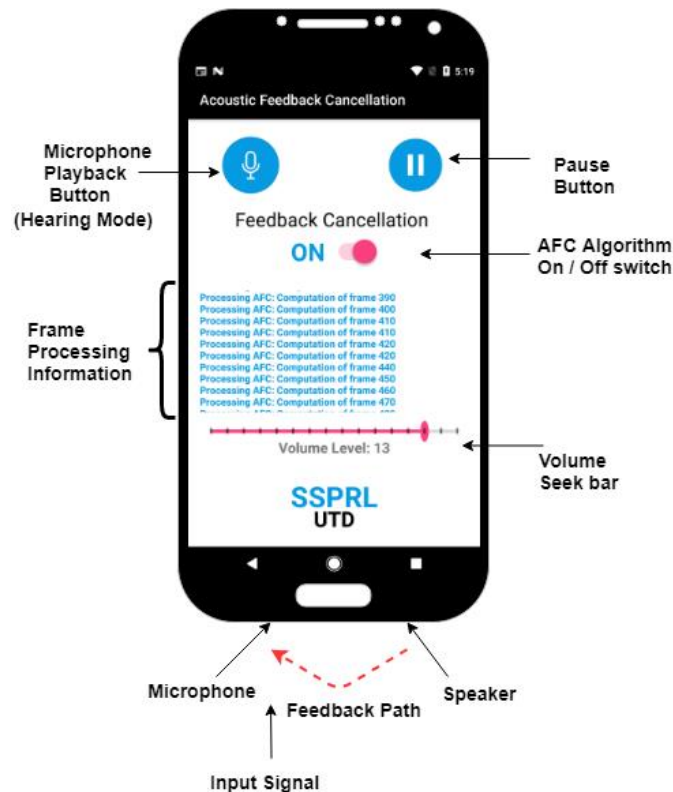


Figure 7 Screenshot of the proposed AFC Application

Following are the key things to keep in mind while using the application:

- When the Mic button is pressed the microphone and the loudspeaker gets switched ON and thus due to the acoustic leakage from the speaker the signal gets picked up by the microphone of the smartphone giving rise to the undesirable irritating “whistling” and “screeching” acoustic feedback sound.

- Pause button is used to switch off the smartphone's microphone and loudspeaker
- When the AFC button is in "OFF" mode, the application merely plays back the audio through the smartphone without processing it.
- On switching "ON" the AFC button, Acoustic feedback cancellation algorithm gets activated and the incoming audio stream degraded by acoustic feedback is processed by applying the noise injection acoustic feedback cancellation procedure.
- As it is computationally efficient, so it consumes very less power of the smartphone while running.

To learn more about the application, please refer to our Audio and video demos on <http://www.utdallas.edu/ssprl/hearing-aid-project/>.

4.References:

- [1] Küçük, Abdullah, et al. "Stereo I/O framework for audio signal processing on android platforms." *The Journal of the Acoustical Society of America* 143.3 (2018): 1955-1956.
- [2] Mar. 2018. [Online]. Available:
<https://developer.android.com/training/articles/perf-jni.html>
- [3] Khoubrouy, Soudeh A., and Issa MS Panahi. "Order selection of the hearing aid Feedback Canceller filter based on its impulse response energy." *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2012.
- [4] <https://utdallas.edu/ssprl/hearing-aid-project/acoustic-feedback-cancellation/>
- [5] Mishra, Parth, Serkan Tokgoz, and Issa MS Panahi. "Robust real-time implementation of adaptive feedback cancellation using noise injection algorithm on smartphone." *Proceedings of Meetings on Acoustics 175ASA*. Vol. 33. No. 1. ASA, 2018.