



TouchSense® Android Developer Kit for

DRV2605

Example Haptic Keyboard Integration

1 Haptic Keyboard Integration Overview

The process of modifying the LatinIME keyboard on Android OS to work with DRV2605 device on a mobile handset consists of the following major steps:

1. Add additional modules to the Android OS build system
2. Modify the default LatinIME keyboard code to play desired haptic
3. Build Android images for deployment

2 Adding New Modules

In order for Android to utilize the capabilities of the DRV2605, the Android build system needs to include the shared libraries that are responsible for communicating with the driver on its file system. The 2 shared libraries will need to be located in **/system/lib** so they can be loaded.

To add the shared libraries to the Android build system, follow the steps below:

1. Create a folder named **'hapticapi'** in **'external'** of the Android source
2. Copy **libhapticapi.so** and **libhapticjavaapi.so** to **'hapticapi'**
3. Create a **Android.mk** file and the following lines:

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)
LOCAL_MODULE := libhapticapi.so
LOCAL_MODULE_TAGS := optional
LOCAL_MODULE_CLASS := SHARED_LIBRARIES
LOCAL_MODULE_PATH := $(PRODUCT_OUT)/system/lib
LOCAL_SRC_FILES := $(LOCAL_MODULE)
include $(BUILD_PREBUILT)

include $(CLEAR_VARS)
LOCAL_MODULE := libhapticjavaapi.so
LOCAL_MODULE_TAGS := optional
LOCAL_MODULE_CLASS := SHARED_LIBRARIES
LOCAL_MODULE_PATH := $(PRODUCT_OUT)/system/lib
LOCAL_SRC_FILES := $(LOCAL_MODULE)
include $(BUILD_PREBUILT)
```

After these 3 steps, your **external/hapticapi** should look similar to the following:

```
$ find external/hapticapi/
external/hapticapi/
external/hapticapi/libhapticapi.so
external/hapticapi/libhapticjavaapi.so
external/hapticapi/Android.mk
```

At this point, Android build system has not yet been aware with the new modules although it knows what to do with the modules. We need to define these modules as new packages in **core.mk** located in **build/target/product**.

Edit and add the following lines to the end of **build/target/product/core.mk**:

```
PRODUCT_PACKAGES += \
    libhapticapi.so \
    libhapticjavaapi.so
```

3 Modifying LatinIME Keyboard

Now that `libhapticapi.so` and `libhapticjavaapi.so` are on the file system, modifying Latin IME keyboard to play haptic on keypress is simple.

- Copy the Java API, `api/java/com/*` from the DRV2605 package to `packages/inputmethods/LatinIME/java/src/com`

- Edit `LatinIME.java` in `packages/inputmethods/LatinIME/java/src/com/android/inputmethod/latin` and add the following lines:

1. Import Haptic.java class:

```
import com.android.inputmethod.keyboard.LatinKeyboard;
import com.android.inputmethod.keyboard.LatinKeyboardView;

import com.immersion.Haptic;

import java.io.FileDescriptor;
import java.io.PrintWriter;
import java.util.Locale;
```

2. Initialize the API to communicate with DRV2605 when the keyboard is created:

```
@Override
public void onCreate() {
    Haptic.initialize();

    final SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(this);
    mPrefs = prefs;
    LatinImeLogger.init(this, prefs);
```

3. Terminate the API when the keyboard is destroyed:

```
@Override
public void onDestroy() {
    Haptic.terminate();
    if (mSuggest != null) {
        mSuggest.close();
        mSuggest = null;
```

4. Add code to play effect:

```
public void onPress(int primaryCode, boolean withSliding) {
    final KeyboardSwitcher switcher = mKeyboardSwitcher;
    if (switcher.isVibrateAndSoundFeedbackRequired()) {
        // vibrate();
        Haptic.playEffect(5);
        playKeyClick(primaryCode);
    }
```


4 Building Android Images

After making the changes described above, Android can be rebuilt normally by executing **make** from the root of the Android source code. To rebuild only the above changes, you can use the '**mm**' commands, also issued from the root of the Android source code.

Immersion Corporation

801 Fox Lane
San Jose, CA 95131 USA
T: +1 408.467.1900
F: +1 408.467.1901
www.immersion.com

Copyright 2012 Immersion Corporation. All rights reserved. Immersion, the Immersion logo, and TouchSense are trademarks of Immersion Corporation in the U.S. and other countries. All other trademarks are the property of their respective owners.

This document and the content of this document shall be subject to the terms, conditions, and restrictions of Immersion Corporation's Terms of Use applicable to "Content" (as defined therein) listed at <http://www.immersion.com/legal.html>, including, but not limited to, the terms, conditions, and restrictions relating to Immersion's general disclaimers described therein. The terms, conditions, and restrictions of Immersion Corporation's Terms of Use are hereby incorporated herein by reference. By accessing this document, you hereby agree to follow and be bound by the terms, conditions, and restrictions described in this document and the applicable provisions of Immersion Corporation's Terms of Use.

No warranties, representations or conditions, express or implied, including, without limitation, any representations, warranties or conditions of accuracy, sufficiency, suitability or non-infringement of third party rights, or ownership of any intellectual property rights described or otherwise contained herein, are made by Immersion in this document. All information contained herein is provided "as is," and Immersion shall have no liability whatsoever for any damages, losses or expenses incurred by the recipient of this document (or any other party) as a result of its receipt of this document or use of the information contained herein, whether arising in contract, tort or otherwise.