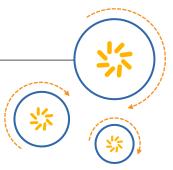


Qualcomm Technologies, Inc.



Qualcomm[®] improveTouch[™] Integration

Guide

80-P6016-1 B

June 29, 2016

Confidential and Proprietary - Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

MSM and Qualcomm Snapdragon are products of Qualcomm Technologies, Inc. Qualcomm improveTouch technology, a feature of Qualcomm Snapdragon processors, is a product of Qualcomm Technologies, Inc. Other Qualcomm products referenced herein are products of Qualcomm Technologies, Inc. or its other subsidiaries.

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

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

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

Revision history

Revision	Date	Description
А	May 2016	Initial release
В	June 2016	Corrected acronym in Section 2.1 and filename in Section 3.2.



Contents

1 Introduction	4
1.1 Purpose	4
1.2 Conventions	
1.3 Technical assistance	
2 improveTouch component overview	5
2.1 improveTouch components	5
2.2 improveTouch process flow	
2.3 Required files	
2.3.1 HBTP daemon binaries	7
2.3.2 JNI/Java libraries	
2.3.3 aDSP/SLPI dynamic loadable libraries	
2.3.4 AFE firmware	
2.3.5 Configuration file	
3 HBTP input drivers	Ω
5 HBTF IIIput unvers	o
3.1 Device nodes	8
3.2 Enable HBTP input drivers	8
4 HBTP service	
4 HBTP service	9
4.1 Register HBTP service	9
4.2 Launch HBTP service at bootup	
4.3 HBTP service security policy	
4.3.1 Security policy files	10
5 Troubleshooting	12
G	
5.1 Touch is not working	
5.2 View HBTP service log	12
A References	13
A.1 Related documents	13
A.2 Acronyms and terms	
	10
Figures	
Figure 2.1 improve Touch prohitecture	6

1 Introduction

1.1 Purpose

This document describes how to integrate an improveTouch device with the following QTI chipsets: MSM8937, MSM8953, and MSM8998.

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:.

1.3 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.

2 improveTouch component overview

The improveTouch module consists of multiple components across different subsystems in MSM8937, MSM8953, and MSM8998 chipsets.

2.1 improveTouch components

The improveTouch Host-Based Touch Processing (HBTP) solution consists of the following components in the touch framework of a Qualcomm[®] SnapdragonTM device:

- improveTouch core framework The improveTouch core framework runs in the application Digital Signal Processor (aDSP) or the Sensor Low Power Island (SLPI). The improveTouch component in the aDSP or SLPI controls the analog front end (AFE) hardware and processes the data that generates the touch coordinates. All advanced use case engines (AUE) also reside in the aDSP or SLPI to save power and prevent the unexpected wake-up of the Application Processor Subsystem (APSS).
- **HBTP daemon** HBTP daemon is the main controller in the APSS and interacts with improveTouch components in the aDSP or SLPI. This daemon manages the lifecycle of improveTouch components in the aDSP or SLPI by delivering touch coordinates and advanced use case-related events from the aDSP or SLPI to Android applications through the improveTouch communication path or the Linux input system through HBTP input driver.
- **HBTP** client HBTP client is a collection of libraries that enable Android applications to access all interfaces and services provided by improveTouch. The collection consists of native client, JNI, and Java libraries. OEMs can develop their own applications using improveTouch features through these client libraries. HBTP client and HBTP daemon communicate with each other through Android Binder interfaces.
- **HBTP input driver** HBTP input driver is the only improveTouch component in the kernel space. This component acts as a bridge between the HBTP daemon, Linux input system, and other kernel components to deliver the following:
 - □ Touch coordinates to the Linux input system
 - □ Events that occur in the kernel space to HBTP daemon

2.2 improveTouch process flow

In the improveTouch process flow, HBTP daemon:

- Initiates the touch core framework in aDSP or SLPI at bootup
- Configures the touch core framework with the setup data
- Starts touch processing

The touch core framework in the aDSP or SLPI configures the AFE hardware with the proper settings, enabling the AFE to begin touch processing. The aDSP or SLPI reads and processes each touch frame. When the aDSP or SLPI detects a valid touch, a proprietary transport/protocol transfers the touch coordinates to HBTP daemon.

HBTP daemon then pushes touch coordinates to HBTP input driver, and eventually the Android Input Framework receives the touch coordinates through the Linux input system.

Figure 2-1 shows how the improveTouch components communicate with each other and how the touch coordinates are delivered to the Android Input Framework.

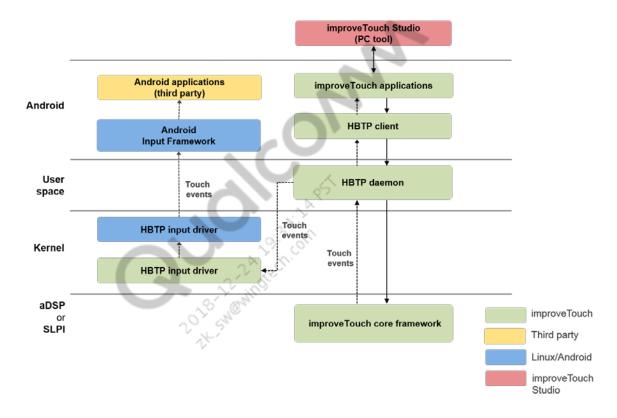


Figure 2-1 improveTouch architecture

2.3 Required files

The improveTouch system requires files in the Android file system. Most of the required files are libraries and executables for Linux and Android, as well as dynamic loadable libraries. These dynamic libraries contain the improveTouch core framework running on the aDSP or SLPI and the configuration/firmware file required to configure the improveTouch system and AFE hardware.

2.3.1 HBTP daemon binaries

Place the following files in the system build location:

```
system/vendor/lib64/libhbtpfrmwk.so
system/vendor/lib64/libhbtpclient.so
system/vendor/lib64/libhbtpdsp.so
system/vendor/lib64/libfastrpc_utf_stub.so
system/vendor/lib64/libfastrpc_aue_stub.so
system/vendor/bin/hbtp_daemon
```

2.3.2 JNI/Java libraries

If you are developing Android applications that use improveTouch features, include the following JNI and Java shared libraries in the system build or application to communicate with the improveTouch HBTP service:

```
system/vendor/lib64/libhbtpjni.so
out/target/common/obj/JAVA_LIBRARIES/touchManager_intermediates/classes.jar
```

2.3.3 aDSP/SLPI dynamic loadable libraries

The following files are shared libraries for the aDSP or SLPI, which loads these files when the improveTouch system is initialized.

Place these files in the exact directory to be loaded to avoid initialization issues:

```
system/vendor/lib/rfsa/adsp/libFastRPC_UTF_Forward_skel.so
system/vendor/lib/rfsa/adsp/libFastRPC_AUE_Forward_skel.so
```

2.3.4 AFE firmware

HBTP daemon passes the following firmware file for AFE hardware to the aDSP or SLPI, which then loads it to AFE hardware:

```
/etc/firmware/hbtp/qtc800s_dsp.bin
```

2.3.5 Configuration file

Each device needs one configuration file that contains all improveTouch parameters.

The configuration file must have the filename hbtpcfg.dat.

/etc/firmware/hbtp/hbtpcfg.dat

3 HBTP input drivers

3.1 Device nodes

HBTP input drivers are enabled in the build by default. improveTouch has two device nodes:

```
/dev/hbtp_input
/dev/hbtp_vm
```

3.2 Enable HBTP input drivers

If HBTP input drivers are not enabled in the build, do the following:

1. Add the CONFIG_INPUT_HBTP_INPUT driver to kernel/drivers/input/misc/Makefile:

```
obj-$(CONFIG_INPUT_HBTP_INPUT) += hbtp_input.o hbtp_vm.o
The HBTP input driver files are compiled by the Makefile for kernel drivers.
```

2. In the defconfig file for the chipset, either kernel/arch/arm/configs/MSMXXXX-perf_defconfig or kernel/arch/arm64/configs/MSMXXXX-perf_defconfig, set the driver to Y:

```
CONFIG INPUT HBTP INPUT Y
```

4 HBTP service

4.1 Register HBTP service

HBTP service initiates the improveTouch system. However, to launch this service you must register HBTP daemon in the init.target.rc file with the appropriate service parameters.

Edit /device/qcom/MSMXXX_YY/init.target.rc and set the following service parameters:

```
service hbtp /system/vendor/bin/hbtp_daemon class main user system group system disabled
```

4.2 Launch HBTP service at bootup

HBTP service is not automatically launched at bootup because the service is initially registered with the disabled class parameter. The init.gcom.post boot script launches HBTP service.

To launch HBTP service at bootup, edit /device/qcom/common/rootdir/etc/init.qcom.post_boot.sh and enter the soc_id, hw_platform, and platform_subtype_id for the touch device.

For example, if the soc_id is 239, the hw_platform is Surf or MTP, and the platform_subtype_id is 3, edit the init.qcom.post_boot.sh as follows:

```
if [ -f /sys/devices/soc0/platform_subtype_id ]; then
    platform subtype id=`cat /sys/devices/soc0/platform subtype id`
fi
if [ -f /sys/devices/soc0/hw_platform ]; then
    hw_platform=`cat /sys/devices/soc0/hw_platform`
fi
case "$soc_id" in
    "239")
    case "$hw_platform" in
            case "$platform_subtype_id" in
                "1")
                    start hbtp
                ;;
            esac
        ;;
        "MTP")
```

4.3 HBTP service security policy

Android applications access interfaces in HBTP service to provide certain improveTouch functionality, such as the touch data visualization and debugging features in improveTouch Studio (PC tool), and the gesture and hand biometric advanced use cases.

Specific security policy files must exist in the build so that the Android applications can access HBTP service.

4.3.1 Security policy files

Do not remove the following security policy files from /device/qcom/sepolicy/common since they are related to HBTP service:

- hbtp.te
- file.te
- property.te
- init_shell.te
- platform_app.te
- file_contexts
- service contexts
- property_contexts

4.3.1.1 service_contexts file

The service_contexts file contains the security policies for all interfaces exposed by HBTP service.

Verify that the following four improveTouch services are defined in device/qcom/sepolicy/common/service_contexts:

improveTouch.TouchService
u:object_r:improve_touch_service:s0
improveTouch.TouchManagerService
u:object_r:improve_touch_service:s0
improveTouch.GestureManagerService
u:object_r:improve_touch_service:s0
improveTouch.HandBiometricManagerService
u:object_r:improve_touch_service:s0

5 Troubleshooting

This chapter describes common issues that you may encounter when integrating an improveTouch device with an MSMTM chipset.

5.1 Touch is not working

The touch module does not start.

Solution

Check the process list to verify that the hbtp_daemon process is running:

```
ps aux | grep hbtp_daemon
```

- ☐ If hbtp_daemon is not in the process list, check that HBTP service is registered. For details, see Section 4.1.
- □ If HBTP service is registered but it has not launched, check the init.qcom.post_boot.sh file and verify that the correct soc_id, hw_platform, and platform_subtype are specified for the touch device. See Section 4.2.
- ☐ If HBTP service is running, but you cannot see the hbtp_daemon process, verify that the hbtp_input device nodes exist. See Section 3.1 and Section 3.2.

5.2 View HBTP service log

The HBTP daemon uses the tag, improveTouch, to identify touch output in the adb log.

To view improveTouch adb log output, use the following log options:

```
adb logcat -v threadtime -s improveTouch
```

For details on adb log options, see the logcat information in the Android Developers site.

A References

A.1 Related documents

Title	Number
Qualcomm® Technologies, Inc.	74
improveTouch API Interface Specifications	80-P6016-2
improveTouch Studio User Guide	80-P6016-3
improveTouch Manufacturing Test Support User Guide	80-P6016-4
improveTouch Quick Start	80-P6016-5
	3 4
Resources	>
Android Developers Tools (logcat)	http://developer.android.com/tools/help/logcat.html

A.2 Acronyms and terms

Acronym or term	Definition	
adb	Android Debug Bridge	
AFE	Analog front end	
aDSP	Application digital signal processor	
APSS	Application processor subsystem	
AUE	Advanced use cases	
HBTP	Host-based touch processing	
HW	Hardware	
JNI	Java Native Interface	
SLPI	Sensor Low Power Island	