Android 8.0_Qualcomm_QSEE_指纹移植指导说明V1.1

版本号	修改日期	修改内容	修改人
V1.0	2018.1.1	初始版本	郭松
V1.1	2018.1.12	修正格式与工模相关内容	郭松

一 移植驱动和修改dtsi文件

- 1.1 驱动移植到方案指定目录下
- 1.2 修改dtsi文件，配置GPIO

二 sepolicy权限文件修改

- 2.1 权限文件修改
- 2.2 init.target.rc文件修改

三 mbn文件与TZ memory配置

- 3.1 mbn文件配置
- 3.2 TZ memory配置

四 CA、TA指纹库移植

- 4.1 将编译好的ca ta移植到客户指定目录下，并修改mk文件
- 4.2 指纹库copy路径说明，TA_PATH 路径修改
- 4.3 manifest.xml文件修改

五 工模测试程序移植，不同客户要求不同

- 5.1 android 8.0 测试程序移植(yudeE300)

六 多家指纹兼容

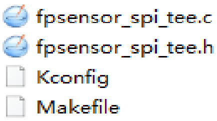
- 6.1 Android O 没有了fingerprintd
- 6.2 修改BiometricsFingerprint.cpp

七 资料说明

一 移植驱动和修改dtsi文件

1.1 驱动移植到方案指定目录下

(参考路径：src/kernel/msm-3.18/drivers/misc/fingerprint/fpsensor/)



fpsensor_spi_tee.c
fpsensor_spi_tee.h
Kconfig
Makefile

修改Kconfig 和Makefile 文件，使可以编译到驱动。修改参考如下：
fpsensor目录下的Kconfig

```
#/*add by fpsensor start*/
config FPSENSOR_FINGERPRINT
    tristate"Fpsensor FingerPrint Driver"
    defaulty
help
     Fpsensor FingerPrint Supported
#/*add by fpsensor end*/
```

fpsensor目录下的Makefile

```
#/*add by fpsensor start*/
obj-y   +=fpsensor_spi_tee.o
#/*add by fpsensor end*/
```

misc目录下的Kconfig添加一行

```
source"drivers/misc/fingerprint/fpsensor/Kconfig"
```

misc目录下的Makefile添加如下

```
# add by fpsensor start
obj-$(CONFIG_FPSENSOR_FINGERPRINT)+=fingerprint/fpsensor/
# add by fpsensor end
```

1.2 修改dtsi文件，配置GPIO

GPIO配置RST和INT(参考路径：src/kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8937-mtp.dtsi)，patch参考如下：

```
diff--gita/arch/arm/boot/dts/qcom/msm8937-mtp.dtsib/arch/arm/boot/dts/qcom/msm8937-mtp.dtsi
index0b56404..679fd3b100755
---a/arch/arm/boot/dts/qcom/msm8937-mtp.dtsi
+++b/arch/arm/boot/dts/qcom/msm8937-mtp.dtsi
@@-59,7+59,17@@
           debounce-interval=<15>
       };
   };
-
+//add by fpsensor fingerprint start
+   fingerprint_gpio{
+       compatible="qcom,fpsensor_finger";
+       interrupt-parent=<&tlmm>
+       interrupts=<120x0>
+
+       fp-gpio-reset=<&tlmm70x00>
+       fp-gpio-int=<&tlmm120x00>
+       fp-gpio-power=<&tlmm860x00>
+       status="okay";
+   };
+//add by fpsensor fingerprint end
   hbtp{
       compatible="qcom,hbtp-input";
       vcc_dig-supply=<&pm8937_l5>
```

以上移植完，可以先编译，看是否有编译到驱动，是否有报错。如无异常，进行下面的步骤。

二 sepolicy**权限文件修改**

2.1 **权限文件修改**

(参考修改路径：src/device/qcom/sepolicy/common/)

device.te

```
#fpsensor fingerprint start
type fpsensor_fp_device, dev_type;
#fpsensor fingerprint end
```

file.te

```
#fpsensor fingerprint start
type fpsensor_data_file, file_type, data_file_type;
#fpsensor fingerprint end
```

file_contexts

```
#fpsensor fingerprint start
/dev/fpsensor               u:object_r:fpsensor_fp_device:s0
/data/fpsensor(/.*)?           u:object_r:fpsensor_data_file:s0
#fpsensor fingerprint end
```

fingerprintd.te

```
#fpsensor fingerprint start
#allow hal_fingerprint_default to access /dev/fpsensor
allow hal_fingerprint_default fpsensor_fp_device:chr_file { open read write ioctl};
allow hal_fingerprint_default fuse:dir {search};
allow hal_fingerprint_default mnt_user_file:dir {search};
allow hal_fingerprint_default mnt_user_file:lnk_file {read};
allow hal_fingerprint_default storage_file:lnk_file {read};

#add for fpsensor fingerprint gesture
allow hal_fingerprint_default uhid_device:chr_file {open ioctl write read};

#add for fpsensor test socket connection
allow untrusted_app fpsensor_fp_device:chr_file {open write read ioctl};
allow hal_fingerprint_default fpsensor_data_file:sock_file {create setattr unlink};
allow hal_fingerprint_default fpsensor_data_file:dir {write read add_name remove_name search setattr};

#add for test tool
type fp_ext_svc2_service, hwservice_manager_type;
allow hal_fingerprint_default  fp_ext_svc2_service:hwservice_manager {add};
allow system_app     fp_ext_svc2_service:hwservice_manager {find};
allow hal_fingerprint_default system_data_file:dir {write read};
#fpsensor fingerprint end
```

hwservice_contexts

```
#fpsensor fingerprint start
android.vendor.fpsensorhidlsvc::IFpsensorHidlSvc           u:object_r:fp_ext_svc2_service:s0
#fpsensor fingerprint end
```

init.te

```
#fpsensor fingerprint start
allow init fpsensor_fp_device:chr_file {write};
#fpsensor fingerprint end
```

system_app.te

```
#fpsensor fingerprint start
allow system_app fpsensor_data_file:sock_file { write };
allow system_app fpsensor_data_file:dir { search };
allow system_app hal_fingerprint_default:fd {use};

#qiancheng@wind-mobi.com 20171028 add -s
#allow system_app vendor_file:file { execute read open ioctl getattr};
#fpsensor fingerprint end
```

2.2 init.target.rc**文件修改**

(参考路径：src/device/xxxxx/(project_name)/)

```
on post-fs-data

    #add for fpsensor fingerprint
    mkdir /data/fpsensor
    chown system system /data/fpsensor
    chown system system /data/fpsensor/socket
    chmod 0660  /data/fpsensor
    chmod 0660  /data/fpsensor/socket

on init
    #add for fpsensor fingerprint
    chmod 0660 /dev/fpsensor
    chown system system /dev/fpsensor
```

三 mbn**文件与**TZ memory**配置**

3.1 mbn**文件配置**

需要修改两个地方:
src/amms/TZ.BF.4.0.5/trustzone_images/`apps`/bsp/trustzone/qsapps/build/secimage.xml
src/amms/TZ.BF.4.0.5/trustzone_images/`core`/bsp/trustzone/qsapps/build/secimage.xml
修改是一样的，如下

```
        <!-- added by fpsensor start-->
       <image sign_id="fngap32" name="fngap32.mbn" image_type="elf_has_ht">
            <general_properties_overrides>
                <sw_id>0x000000000000000C</sw_id>
                <app_id>0x0000000000112345</app_id>
            </general_properties_overrides>
        </image>
        <image sign_id="fngap64" name="fngap64.mbn" image_type="elf_has_ht">
            <general_properties_overrides>
                <sw_id>0x000000000000000C</sw_id>
                <app_id>0x0000000000112345</app_id>
            </general_properties_overrides>
        </image>
         <!-- added by fpsensor end-->
```

注: app_id请客户根据情况自己定义，如客户未要求，可以自己定义，但不得与其它 TA 的 app_id相同。fngap32和fngap64分别是指纹TA的32bit和64bit版本名字，在默认情况，只编译64bit版本，即fngap64。secimage.xml文件中提到的文件均需要签名。

3.2 TZ memory**配置**

需要修改三个地方

3.2.1 In QSEE SDK，如下:

src/amms/TZ.BF.4.0.5/trustzone_images/ `core` /securemsm/trustzone/qsee/mink/oem/config/msm8937/oem_co

，Patch参考如下:

```
diff --git
a/trustzone_images/core/securemsm/trustzone/qsee/mink/oem/config/msm8937/oem_config.xml
b/trustzone_images/core/securemsm/trustzone/qsee/mink/oem/config/msm8937/oem_config.xml
index fc33bcd..eeb21ed 100755
--- a/trustzone_images/core/securemsm/trustzone/qsee/mink/oem/config/msm8937/oem_config.xml
+++ b/trustzone_images/core/securemsm/trustzone/qsee/mink/oem/config/msm8937/oem_config.xml
@@ -39,12 +39,14 @@
     1
     </props>
     <!-- PIL load region information -->
+    <!-- added by fpsensor start-->
     <props name="OEM_pil_secure_app_load_region_start" type=DALPROP_ATTR_TYPE_UINT32>
-      0x84F00000
+      0x84A00000
     </props>
     <props name="OEM_pil_secure_app_load_region_size" type=DALPROP_ATTR_TYPE_UINT32>
-      0x1400000
+      0x1900000
     </props>
+    <!-- added by fpsensor end-->
     <props name="OEM_pil_subsys_load_region_start" type=DALPROP_ATTR_TYPE_UINT32>
       0x80000000
     </props>
```

3.2.2 In Linux kernel，如下:

src/kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8937.dtsi，Patch参考如下:

```
diff--gita/arch/arm/boot/dts/qcom/msm8937.dtsib/arch/arm/boot/dts/qcom/msm8937.dtsi
oldmode100644
newmode100755
indexddf40d1..d9837cf8
---a/arch/arm/boot/dts/qcom/msm8937.dtsi
+++b/arch/arm/boot/dts/qcom/msm8937.dtsi
@@-59,7+59,9@@
       other_ext_mem:other_ext_region@0{
           compatible="removed-dma-pool";
           no-map;
-           reg=<0x00x85b000000x00xd00000>
+/*add by fpsensor start*/
+           reg=<0x00x84a000000x00x1E00000>
+/*add by fpsensor start*/
       };
       modem_mem:modem_region@0{
@@-1552,9+1554,11@@
       qcom,ce-opp-freq=<100000000>
   };
-   qcom_seecom:qseecom@85b00000{
+   qcom_seecom:qseecom@84A00000{
       compatible="qcom,qseecom";
-       reg=<0x85b000000x800000>
+/*add by fpsensor start*/
+       reg=<0x84A000000x1900000>
+/*add by fpsensor end*/
       reg-names="secapp-region";
       qcom,hlos-num-ce-hw-instances=<1>
       qcom,hlos-ce-hw-instance=<0>
```

3.2.3 In bootloader LK，如下

src/bootable/bootloader/lk/platform/msm8952/include/platform/iomap.h,Patch参考如下:

```
diff --git a/platform/msm8952/include/platform/iomap.h
b/platform/msm8952/include/platform/iomap.h
old mode 100644
new mode 100755
index 6f4b28f..c2f2266
```

```
--- a/platform/msm8952/include/platform/iomap.h
+++ b/platform/msm8952/include/platform/iomap.h
@@ -172,8 +172,10 @@
 #define APP_REGION_SIZE platform_get_tz_app_size()
 #define APP_REGION_ADDR_8952 0x85E00000
 #define APP_REGION_SIZE_8952 0x500000
-#define APP_REGION_ADDR_8937 0x85B00000
-#define APP_REGION_SIZE_8937 0x800000
+/* added by fpsensor start */
+#define APP_REGION_ADDR_8937 0x84a00000
+#define APP_REGION_SIZE_8937 0x1900000
+/* added by fpsensor end */

 /* MDSS */
 #define MIPI_DSI_BASE               (0x1A98000)
```

四 CA、TA指纹库移植

4.1 **将编译好的**ca ta**移植到客户指定目录下，并修改**mk**文件**

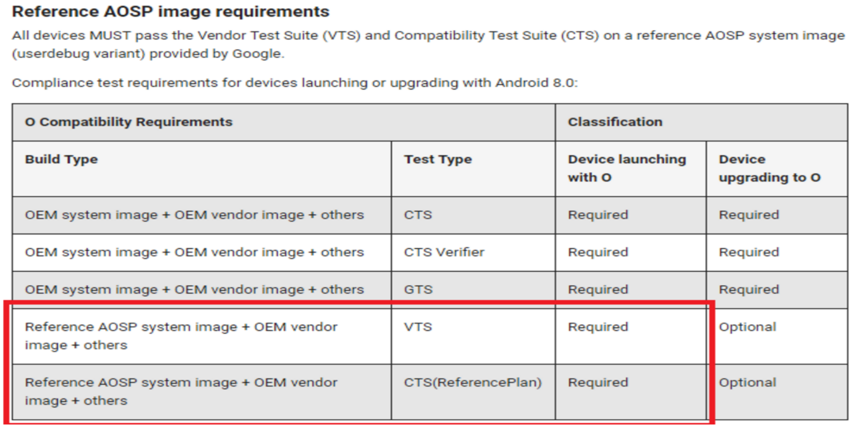
(指纹库参考目录：src/device/common/fingerprint/fpsensor/下；mk参考路径：src/device/xxxx/（project_name）/下)

mk文件修改：

```
# Added by fpsensor fingerprint
# 开启指纹服务和fingerprintservice，如下两条
PRODUCT_PACKAGES += \
    android.hardware.biometrics.fingerprint@2.1-service

PRODUCT_COPY_FILES += \

frameworks/native/data/etc/android.hardware.fingerprint.xml:vendor/etc/permissions/android.hardware.fingerprint.xml

# 添加指纹相关的库(ca、ta、工模的库)
PRODUCT_COPY_FILES += \

device/common/fingerprint/fpsensor/ca/libfpsensor_fingerprint.default.so:vendor/lib64/hw/fpsensor_fingerprint.default.so \
    device/common/fingerprint/fpsensor/ca/fp_ext_svc2.so:vendor/lib64/fp_ext_svc2.so \

device/common/fingerprint/fpsensor/ca/android.vendor.fpsensorhidlsvc@2.0.so:vendor/lib64/android.vendor.fpsensorhidlsvc@2.0.so \
    device/common/fingerprint/fpsensor/ta/fngap64.b00:/vendor/etc/firmware/fngap64.b00 \
    device/common/fingerprint/fpsensor/ta/fngap64.b01:/vendor/etc/firmware/fngap64.b01 \
    device/common/fingerprint/fpsensor/ta/fngap64.b02:/vendor/etc/firmware/fngap64.b02 \
    device/common/fingerprint/fpsensor/ta/fngap64.b03:/vendor/etc/firmware/fngap64.b03 \
    device/common/fingerprint/fpsensor/ta/fngap64.b04:/vendor/etc/firmware/fngap64.b04 \
    device/common/fingerprint/fpsensor/ta/fngap64.b05:/vendor/etc/firmware/fngap64.b05 \
    device/common/fingerprint/fpsensor/ta/fngap64.b06:/vendor/etc/firmware/fngap64.b06 \
    device/common/fingerprint/fpsensor/ta/fngap64.mdt:/vendor/etc/firmware/fngap64.mdt
# Added by fpsensor fingerprint end
```

4.2 **指纹库**copy**路径说明，**TA_PATH **路径修改**

指纹库TA CA以及xml文件也可以copy到系统的system分区，所以如上4.1中mk文件的vendor可以写为system，但要使用相应的ca、ta。因为android 8.0的VTS测试，需要替换为原生的AOSP system镜像来测试，如下图



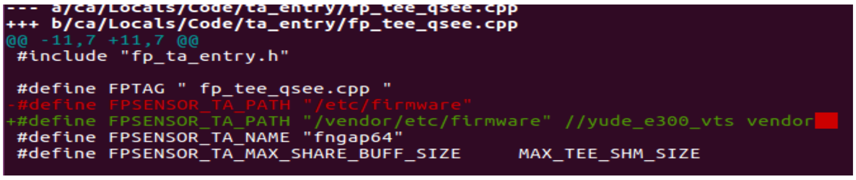
Reference AOSP image requirements

All devices MUST pass the Vendor Test Suite (VTS) and Compatibility Test Suite (CTS) on a reference AOSP system image (userdebug variant) provided by Google.

Compliance test requirements for devices launching or upgrading with Android 8.0:

O Compatibility Requirements		Classification	
Build Type	Test Type	Device launching with O	Device upgrading to O
OEM system image + OEM vendor image + others	CTS	Required	Required
OEM system image + OEM vendor image + others	CTS Verifier	Required	Required
OEM system image + OEM vendor image + others	GTS	Required	Required
Reference AOSP system image + OEM vendor image + others	VTS	Required	Optional
Reference AOSP system image + OEM vendor image + others	CTS(ReferencePlan)	Required	Optional

为了保证指纹功能正常，需要把ca ta以及工模so库都部署到vendor分区去，所以copy到vendor路径下。此时需要FAE修改TA_PATH以保证CA和TA正常通讯，修改如下



```
--- a/ca/Locals/Code/ta_entry/fp_tee_qsee.cpp
+++ b/ca/Locals/Code/ta_entry/fp_tee_qsee.cpp
@@ -11,7 +11,7 @@
 #include "fp_ta_entry.h"

 #define FPTAG " fp_tee_qsee.cpp "
-#define FPSENSOR_TA_PATH "/etc/firmware"
+#define FPSENSOR_TA_PATH "/vendor/etc/firmware" //yude_e300_vts vendor
 #define FPSENSOR_TA_NAME "fngap64"
 #define FPSENSOR_TA_MAX_SHARE_BUFF_SIZE      MAX_TEE_SHM_SIZE
```

修改完成后，需要重新编译ca、ta并release给客户。

4.3 manifest.xml**文件修改**

(参考路径：src/device/xxxx/（project_name）/)

```
    <!-- added by fpsensor for fingerprint Service-->
    <hal format="hidl">
        <name>android.hardware.biometrics.fingerprint</name>
        <transport>hwbinder</transport>
        <version>2.1</version>
        <interface>
            <name>IBiometricsFingerprint</name>
            <instance>default</instance>
        </interface>
    </hal>
    <!-- added by fpsensor for fingerprint factory mode test-->
    <hal format="hidl">
        <name>android.vendor.fpsensorhidlsvc</name>
        <transport>hwbinder</transport>
        <version>2.0</version>
        <interface>
            <name>IFpsensorHidlSvc</name>
            <instance>default</instance>
        </interface>
    </hal>
```

只调试基本功能,前四项移植好，就可以验证了。

五 工模测试程序移植，不同客户要求不同

5.1 android 8.0 **测试程序移植**(yudeE300)

5.1.1 保证指纹基本功能可用，完成录入、解锁功能
5.1.2 添加selinux 权限(权限修改部分2.1已添加)
  hwservice_contexts 中添加

```
android.vendor.fpsensorhidlsvc::IFpsensorHidlSvc u:object_r:fp_ext_svc2_service:s0
```

hal_fingerprint_default.te(或者fingerprintd.te)中添加

```
type fp_ext_svc2_service, hwservice_manager_type;
allow hal_fingerprint_default  fp_ext_svc2_service:hwservice_manager {add};
allow platform_app     fp_ext_svc2_service:hwservice_manager {find};
```

5.1.3 在src/device/xxxx/（project_name）/manifest.xml 中添加如下信息(4.3中已经添加)

```
<hal format="hidl">
<name>android.vendor.fpsensorhidlsvc</name>
<transport>hwbinder</transport>
<version>2.0</version>
<interface>
    <name>IFpsensorHidlSvc</name>
    <instance>default</instance>
</interface>
</hal>
```

然后重新编译并烧录系统

5.1.4 将so(apk\ext_svc2\libs) push到vendor/lib(64):根据当前安卓8.0的版本，可以选择static mode或者shared mode

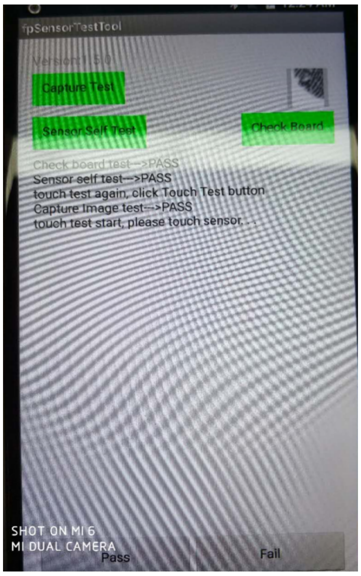
static mode：只要一个so，out\soong.intermediates\hardware\interfaces\biometrics\fingerprint\2.1\android.vendor.biometrics.fingerprint@2.1目录下是否有目录android_arm_armv8-a_cortex-a73_core_static

shared mode：需要三个so。（此种方式通用性更好，不过文件较多)另外需注意，sharedmode下，android.hidl.base@1.0.so 是系统自动生成的，不是我们的代码生成的，如果系统已经有这个库，可以不用push。

5.1.5 将apk进行platform签名，然后push到/system/app/fpExtensionSvc2/fpExtensionSvc2.apk

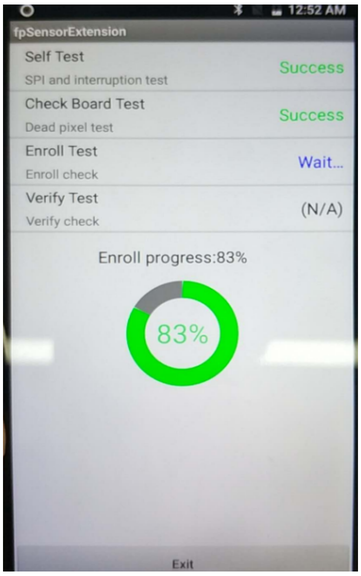
5.1.6 重启后，执行命令启动测试程序

```
//默认界面命令
adb shell am start
com.fpsensor.fpSensorExtensionSvc2/com.fpsensor.sensortesttool.sensorTestActivity
adb shell am start
com.fpsensor.fpSensorExtensionSvc2/com.fpsensor.sensortesttool.sensorTestActivity
```



```
Check board test--->PASS
Sensor self test--->PASS
touch test again, click Touch Test button
Capture Image test--->PASS
touch test start, please touch sensor. . .
```

```
//yude定制界面命令
adb shell am start
com.fpsensor.fpSensorExtensionSvc2/com.fpsensor.fpenrollauthtest.fpTestActivity
```



fpSensorExtension

Self Test — Success
SPI and interruption test

Check Board Test — Success
Dead pixel test

Enroll Test — Wait...
Enroll check

Verify Test — (N/A)
Verify check

Enroll progress:83%

5.1.7 上面验证成功后，要把两个so（android.vendor.fpsensorhidlsvc@2.0.so和fp_ext_svc2.so）放到src/device/common/fingerprint/fpsensor/ca/下，并修改mk文件（4.1中已修改）

六 多家指纹兼容

6.1 Android O **没有了**fingerprintd

fingerprintService.java调用HIDL接口，HIDl接口的实现类可以由指纹厂家自行去实现。BiometricsFingerprint.cpp 文件是IBiometricsFingerprint接口的实现类。android.hardware.biometrics.fingerprint@2.1-service.rc启动fps_hal 服务。BiometricsFingerprint.cpp 文件，会在构造函数中去打开HAL（CA）。

6.2 **修改**BiometricsFingerprint.cpp

使用我们提供的BiometricsFingerprint.cpp对比修改即可。下面只贴一个openhal，注意最后的break，不能丢。

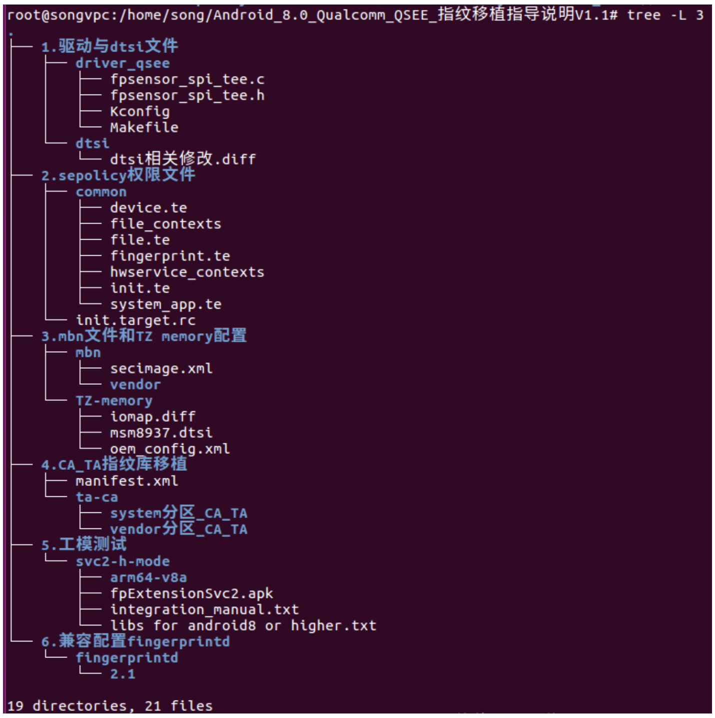
```
fingerprint_device_t* BiometricsFingerprint::openHal() {
    int err;
    const hw_module_t *hw_mdl = nullptr;
    fingerprint_device_t* fp_device = nullptr;
    ALOGD("Opening fingerprint hal library...");
    for(int i = 0; i < FP_VARIANT_KEYS_COUNT; i++) {
        if (0 != (err = hw_get_module(variant_keys[i], &hw_mdl))) {
            ALOGE("Can't open fingerprint HW Module, error: %d", err);
            continue;
        }

        if (hw_mdl == nullptr) {
            ALOGE("No valid fingerprint module");
            continue;
        }

        fingerprint_module_t const *module =
            reinterpret_cast<const fingerprint_module_t*>(hw_mdl);
        if (module->common.methods->open == nullptr) {
            ALOGE("No valid open method");
            continue;
        }

        hw_device_t *device = nullptr;

        if (0 != (err = module->common.methods->open(hw_mdl, nullptr, &device))) {
            ALOGE("Can't open fingerprint methods, error: %d", err);
            continue;
        }

        if (kVersion != device->version) {
            // enforce version on new devices because of HIDL@2.1 translation layer
            ALOGE("Wrong fp version. Expected %d, got %d", kVersion, device->version);
            continue;
        }
        fp_device = reinterpret_cast<fingerprint_device_t*>(device);
        if (0 != (err =
                fp_device->set_notify(fp_device, BiometricsFingerprint::notify))) {
            ALOGE("Can't register fingerprint module callback, error: %d", err);
            fp_device = nullptr;
            continue;
        }
     break;//必须有，逻辑才ok
    }
    return fp_device;
}
```

七 资料说明

7.1 适用平台MSM8917/MSM8937

7.2 所提供的代码和库仅用于Android 8.0_Qualcomm_QSEE平台指纹基本功能移植成功，最终效果要以FAE所提供的最新代码和库为准。

7.3 Release资料目录结构如下：



```
root@songvpc:/home/song/Android_8.0_Qualcomm_QSEE_指纹移植指导说明V1.1# tree -L 3
.
├── 1.驱动与dtsi文件
│   ├── driver_qsee
│   │   ├── fpsensor_spi_tee.c
│   │   ├── fpsensor_spi_tee.h
│   │   ├── Kconfig
│   │   └── Makefile
│   └── dtsi
│       └── dtsi相关修改.diff
├── 2.sepolicy权限文件
│   ├── common
│   │   ├── device.te
│   │   ├── file_contexts
│   │   ├── file.te
│   │   ├── fingerprint.te
│   │   ├── hwservice_contexts
│   │   ├── init.te
│   │   └── system_app.te
│   └── init.target.rc
├── 3.mbn文件和TZ memory配置
│   ├── mbn
│   │   ├── secimage.xml
│   │   └── vendor
│   └── TZ-memory
│       ├── iomap.diff
│       ├── msm8937.dtsi
│       └── oem_config.xml
├── 4.CA_TA指纹库移植
│   ├── manifest.xml
│   └── ta-ca
│       ├── system分区_CA_TA
│       └── vendor分区_CA_TA
├── 5.工模测试
│   └── svc2-h-mode
│       ├── arm64-v8a
│       ├── fpExtensionSvc2.apk
│       ├── integration_manual.txt
│       └── libs for android8 or higher.txt
└── 6.兼容配置fingerprintd
    └── fingerprintd
        └── 2.1

19 directories, 21 files
```