TP Driver Porting Guide for Qualcomm

TP Driver Porting Guide for Qualcomm	
Project name	Touch panel
Document ref	[Document ref]
Version	1.3
Release date	2017.03.06
Owner	Driver Team, FocalTech
Classification	
Distribution List	
Approval	

3/F,Kingdom Sci-Tech Building,
5th Gaoxinnan Avenue, Hi-Tech Park,
Nanshan District ,Shenzhen, Gungdong, P.R. China

ZIP :518057
T +86 755 26588222
F +86 755 26712499
E support@focaltech-electronics.com

www.focaltech-electronics.com

Revision History

Date	Version	List of changes	Author	Approved by
2017.03.06	1.3	1. Add test step of “Factory Test” 2. Modify upgrade configuration and description	xiaoligen	
2016.12.29	1.2	1. More IC Support	xiaoligen	
2016.10.31	1.1	1. More IC support 2. Extern mode, gesture update	xiaoligen	
2016.08.30	1.0	1. Initial draft.	xiaoligen	

Contents

1 Abstract

This guide introduces the structure and functions of Focaltech Android TP Driver, and a step-by-step porting reference for your Qualcomm platform.

Note: this driver has to be generated by Generator.exe.

2 Basic Information

Basic Information	
IC Supported	FT8716、FT8736、FT8006M、FT8201、FT8606、FT8607、FTE716、FT5416、FT5426、FT5435、FT5436、FT5526、FT5526I、FT5446、FT5346、FT5446I、FT5346I、FT7661、FT7511、FT7421、FT7681、FT3C47U、FT3417、FT3517、FT3327、FT3427、FT5626、FT5726、FT5826B、FT5826S、FT7811、FT3D47、FT3617、FT3717、FT3817B、FT6236U、FT6336G、FT6336U、FT6436U、FT3267、FT3367
Platform Supported	All Qualcomm
APK/ADB Tool	Yes
Other functions	GESTURE、ESD

3 File Structure

Driver Files are included in the focaltech_touch folder. They implement i2c driver initialization, touch events reporting, suspend/resume, gesture wakeup, firmware upgrading, and also APIs for APK & ADB debugging tools. Here is a brief table for them:

File Name	Attribute	Function
Makefile	Required	Makefile file for kernel make
Kconfig	Required	Kconfig file for kernel menuconfig
focaltech_core.c	Required	entrance file, i2c driver init, events reporting, suspend/resume etc
focaltech_core.h	Required	header file for focaltech_core.c
focaltech_i2c.c	Required	i2c communication
focaltech_flash.c	Required	fw upgrading
focaltech_flash.h	Required	header for fw upgrading
focaltech_esdcheck.c	Required	ESD check
focaltech_config.h	Required	configuration for driver
focaltech_common.h	Required	common definitions and declarations
focaltech_gesture.c	Optional	gesture wakeup
focaltech_ex_mode.c	Optional	cover, glove, charger special mode
focaltech_ex_fun.c	Optional	generate API nodes for APK/ADB tools
focaltech_sensor.c	Optional	Sensors
focaltech_test	Optional	mass production test
focaltech_point_report_check.c	Optional	auto-report All Up if UP events timeout
include		
firmware/FT8716_app_sample.i	Required	app file for firmware upgrading (sample one is null, you need define your own)
pramboot	Optional	if a pramboot.i is required for upgrading, that's it

4 Porting to Qualcomm

4.1 Files

Copy focaltech_touch directory under kernel/drivers/input/touchscreen, then add the line below to kernel/drivers/input/touchscreen/Kconfig:

```
source "drivers/input/touchscreen/focaltech_touch/Kconfig"
```

And add the line below to kernel/drivers/input/touchscreen/Makefile:

```
obj-$(CONFIG_TOUCHSCREEN_FOCALTECH_FTS) +=  focaltech_touch/
```

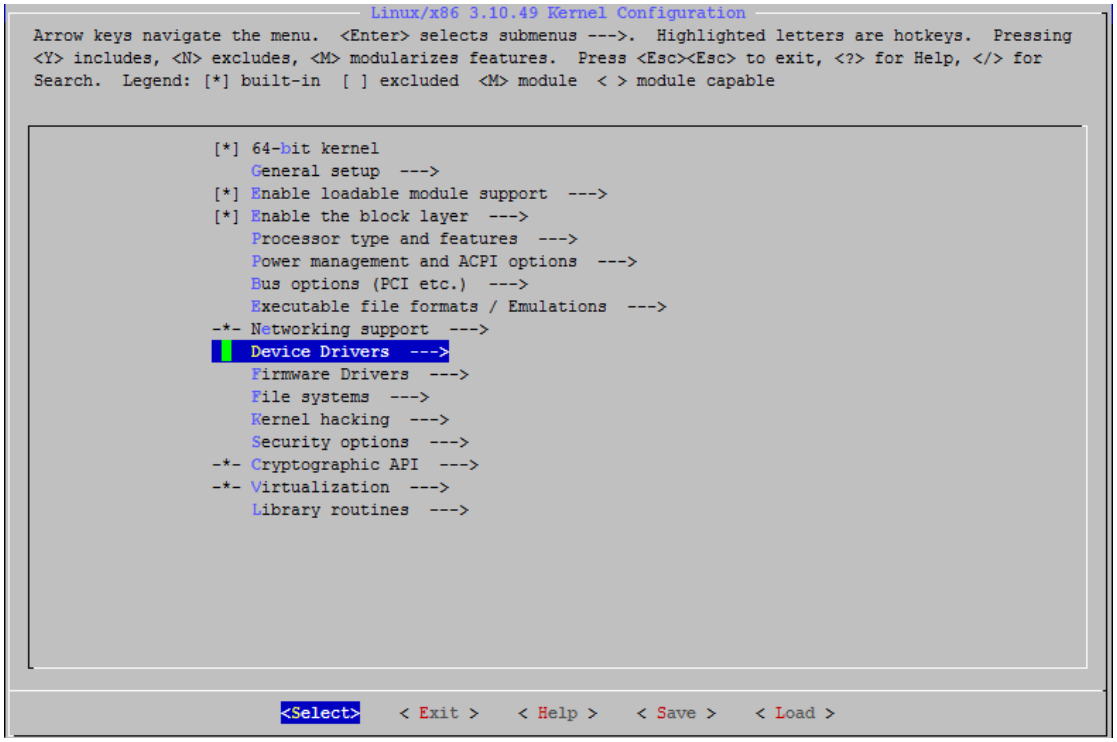
4.2 Compilation

(1) call menuconfig and check FTS;

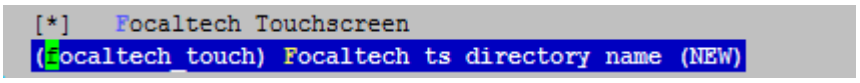
For example:

```
$ source build/envsetup.sh
$ lunch msm8916_64-userdebug
$ cd kernel
$ make menuconfig
```

After the Kernel Configuration dialog showing up as below, check FTS by Device Drivers -> Input Device Support -> Touchscreens -> Focaltech Touchscreen



If you named an alternative TP directory (default: focaltech_touch), then you also need define "Focaltech Touchscreen" -> "Focaltech ts directory name" for your own



(2) Make boot.img;

```
$ make bootimage -j4
```

5 Driver configuration

Configure resolution, keys and pins in DTS, configure driver modules in focaltech_config.h..

5.1 DTS Configuration

For example:

Qualcomm dts file path： arch\arm64\boot\dts\qcom\apq8016-sbc.dtsi

```
 /*
   * KEY_BACK: 158
   * KEY_MENU: 139
   * KEY_HOMEPAGE: 172
   * KEY_SEARCH: 217
   */
focaltech@38{
    compatible = "focaltech,fts";   /* do not modify */
    reg = <0x38>;                   /* do not modify */
    interrupt-parent = <&msm_gpio>;             /* INT pin */
    interrupts = <13 0x2>;
    focaltech,reset-gpio = <&msm_gpio 12 0x01>; /* RST pin */
    focaltech,irq-gpio = <&msm_gpio 13 0x02>;   /* INT pin */
    focaltech,max-touch-number = <5>;
    focaltech,display-coords =  <0 0 1080 1920>;/* resolution */
    /* key settings */
    /*focaltech,have-key;*/  /* no keys */
    /* have key(s) */
    focaltech,have-key;
    focaltech,key-number = <3>;
    focaltech,keys = <139 172 158>; /* key codes*/
    focaltech,key-y-coord = <2000>; /* keys y coordinate */
    focaltech,key-x-coords = <200 600 800>; /* keys x coords */
   /* HD 720P reference settings */
   /*focaltech,display-coords =  <0 0 720 1280>;
    focaltech,key-y-coord = <1350>;
    focaltech,key-x-coords = <180 350 540>;*/
};
You could also take a look at docs/focaltech-ts.txt for reference.
```

5.2 Driver Module Configuration

Driver modules can be enabled(1) and disabled(0) in focaltech_config.h, before that you need define your focaltech chip type first:

Macro	Function	Default
FTS_CHIP_TYPE	current focaltech chip type	_FT8716

Then config modules：

Macro	Function	Default
FTS_DEBUG_EN	debug log for debugging, disable it for release	Enable(debug), Disable(release)
FTS_MT_PROTOCOL_B_EN	Linux Multi-Touch protocol, enable(B), disable(A)	Enable
FTS_REPORT_PRESSURE_EN	Multi-Touch A/B report ABS_MT_PRESSURE	Enable
FTS_FORCE_TOUCH_EN	if enable a real-time pressure is reported, otherwise it's a constant one	Disable
FTS_GESTURE_EN	gesture function	Disable
FTS_ESDCHECK_EN	ESD protection, check ESD state every 1 second, if something wrong a hw reset will execute	Disable
FTS_TEST_EN	mass production test, customers can test phones by ADB commands, if enable focaltech_test dir is required	Disable
FTS_GLOVE_EN	glove mode	Disable
FTS_COVER_EN	cover mode	Disable
FTS_CHARGER_EN	USB charger plug-in plug-out detect	Disable
FTS_PSENSOR_EN	proximity sensor	Disable
FTS_SYSFS_NODE_EN	interface nodes for APK/ADB TOOL	Enable
FTS_APK_NODE_EN		Enable
FTS_POWER_SOURCE_CUST_EN	custom power source control	Disable

Upgrade		
FTS_AUTO_UPGRADE_EN	if auto-upgrade after boot	Enable
FTS_AUTO_UPGRADE_FOR_LCD_CFG_EN	auto-upgrade LCD config	Disable
FTS_AUTO_CLB_EN	auto-calibrate CB	Disable
FTS_UPGRADE_LCD_CFG	.i file for upgrading lcd cfg in flash (you need define your own one)	lcd_cfg.i
FTS_GET_VENDOR_ID_NUM	number of vendors , which are compatible by a corresponding unique vendor id for FW upgrading	0
FTS_VENDOR_1_ID	if NUM >=1, the first vendor's Vendor ID	0x0
FTS_VENDOR_2_ID	if NUM >=2, the second vendor's Vendor ID	0x0
FTS_VENDOR_3_ID	if NUM ==3, the third vendor's Vendor ID	0x0
FTS_UPGRADE_FW_APP	default app.i FW for auto-upgrading (need to be replaced, the sample one is invalid).	Sample.i

	If NUM >= 1, it's the FW corresponding with VENDOR_1_ID	
FTS_UPGRADE_FW2_APP	app.i FW corresponding with VENDOR_2_ID	Sample.i
FTS_UPGRADE_FW3_APP	app.i FW corresponding with VENDOR_3_ID	Sample.i
Stress Test		
FTS_UPGRADE_STRESS_TEST	stress upgrading test , a stress test is recommended after driver porting finished	Disable
FTS_UPGRADE_TEST_NUMBER	stress upgrading test number	1000

5.3 Upgrading Settings

FTS_AUTO_UPGRADE_EN is enabled as default, then you need define followings：

FTS_CHIP_TYPE:

Current chip type

FTS_GET_VENDOR_ID_NUM

If there is more than one TP vendor in this case, you need define it for vendor compatibility of FW upgrading, here are values of it:

0- No Check Vendor ID,FW FTS_UPGRADE_FW_APP is the default FW

1- FTS_VENDOR_1_ID is checked, if matched, FW FTS_UPGRADE_FW_APP will be the FW for upgrading, Otherwise no upgrading.

2- two vendors, FTS_VENDOR_1_ID& FTS_VENDOR_2_ID are checked, if matched VENDOR_1_ID, FTS_UPGRADE_FW_APP will be the FW for upgrading. And FTS_UPGRADE_FW2_APP for VENDOR_2_ID.

3- three vendors, FTS_VENDOR_1_ID & FTS_VENDOR_2_ID & FTS_VENDOR_3_ID are checked. FTS_UPGRADE_FW_APP for VENDOR_1_ID, FTS_UPGRADE_FW2_APP for VENDOR_2_ID, and FTS_UPGRADE_FW3_APP for VENDOR_3_ID

FTS_VENDOR_1_ID

Vendor ID of the first TP vendor

FTS_VENDOR_2_ID

Vendor ID of the second TP vendor

FTS_VENDOR_3_ID

Vendor ID of the third TP vendor

FTS_UPGRADE_FW_APP

Default FW for auto-upgrading, here is the replacement:

Put your App.i FW file in /include/firmware

Redefine the macro to the full path of the FW file

```
#define FTS_UPGRADE_FW_APP          "include/firmware/FT8607_LGE_K6_V0x3D_D01_20160810_app.i"
```

if FTS_GET_VENDOR_ID_NUM >= 1, it's the FW corresponding with FTS_VENDOR_1_ID

FTS_UPGRADE_FW2_APP

if FTS_GET_VENDOR_ID_NUM >=2, it's the FW corresponding with FTS_VENDOR_2_ID

Replacement same as FTS_UPGRADE_FW_APP

FTS_UPGRADE_FW3_APP

if FTS_GET_VENDOR_ID_NUM == 3, it's the FW corresponding with FTS_VENDOR_3_ID

Replacement same as FTS_UPGRADE_FW_APP

5.4 Factory Test Settings

Enable FTS_TEST_EN in focaltech_config.h, if you are using an driver lower than v1.2, then you should configure the test settings in (2);

(1) Enable FTS_TEST_EN in focaltech_config.h

```
/*
 * Production test enable
 * 1: enable, 0:disable(default)
 */
#define FTS_TEST_EN                     1

/*
 * Nodes for tools, please keep enable
 */
#define FTS_SYSFS_NODE_EN               1
#define FTS_APK_NODE_EN                 1
```

(2) If you are using an old version (before V1.2) driver, you need define FTS_CHIP_TEST_TYPE in red box as the IC to be tested in focaltech_test_config.h. E.g. if it's FT8716, then define FTS_CHIP_TEST_TYPE as FT8716_TEST.

```
/*-----------------------------------------------------------
IC Type Test
-----------------------------------------------------------*/

#define FT3C47_TEST    0x3C47
#define FT3D47_TEST    0x3D47
#define FT5822_TEST    0x5822
#define FT5X46_TEST    0x5422
#define FT6X36_TEST    0x6336
#define FT8606_TEST    0x8606
#define FT8607_TEST    0x8607
#define FT8716_TEST    0x8716
#define FT8736_TEST    0x8736
#define FTE716_TEST    0xE716
#define FTE736_TEST    0xE736
#define FT8006_TEST    0x8006


#ifdef  FTS_CHIP_TYPE

    #if (FTS_CHIP_TYPE == _FT8716)
        #define FTS_CHIP_TEST_TYPE      FT8716_TEST
    #elif(FTS_CHIP_TYPE == _FT8736)
        #define FTS_CHIP_TEST_TYPE      FT8736_TEST
    #elif(FTS_CHIP_TYPE == _FT8006)
        #define FTS_CHIP_TEST_TYPE      FT8006_TEST
    #elif(FTS_CHIP_TYPE == _FT8606)
        #define FTS_CHIP_TEST_TYPE      FT8606_TEST
    #elif(FTS_CHIP_TYPE == _FT8607)
        #define FTS_CHIP_TEST_TYPE      FT8607_TEST
    #elif(FTS_CHIP_TYPE == _FT3D47)
        #define FTS_CHIP_TEST_TYPE      FT3D47_TEST
    #elif(FTS_CHIP_TYPE & 0x000007FF == 0x0001)
        #define FTS_CHIP_TEST_TYPE      FT5822_TEST
    #elif(FTS_CHIP_TYPE == _FT3C47U)
        #define FTS_CHIP_TEST_TYPE      FT3C47_TEST
    #elif((FTS_CHIP_TYPE & 0x000007FF == 0x0002) )
        #define FTS_CHIP_TEST_TYPE      FT5X46_TEST
    #elif((FTS_CHIP_TYPE & 0x000007FF == 0x0003) ||(FTS_CHIP_TYPE & 0x000007FF == 0x0004))
        #define FTS_CHIP_TEST_TYPE      FT6X36_TEST
    #endif

#else
    #define FTS_CHIP_TEST_TYPE          FT8716_TEST
#endif
```

(3) Test Steps

3.1 Load Test.ini Configurations

Copy the OK Conf_MultipleTest.ini to /sdcard/ by ADB push command. Just like this:

```
C:\Users\luoguojin>adb push E:\Conf_MultipleTest.ini /sdcard/
12 KB/s (3674 bytes in 0.288s)
```

3.2 Locate Test Node

1) > adb root

2) > adb remount

3) > adb shell

4) # cd /sys/bus/i2c/devices/*-0038 (the * stands for the i2c line no. for TP)

Under the directory , fts_test is the node

```
C:\Users\luoguojin>adb remount
remount succeeded

C:\Users\luoguojin>adb shell
root@msm8916_64:/ # cd sys/bus/i2c/devices/6-0038
cd sys/bus/i2c/devices/6-0038
root@msm8916_64:/sys/bus/i2c/devices/6-0038 # ls
ls
driver
fts_charger_mode
fts_cover_mode
fts_driver_version
fts_dump_reg
fts_fw_update
fts_fw_version
fts_gesture_buf
fts_gesture_mode
fts_glove_mode
fts_hw_reset
fts_irq
fts_module_config
fts_rw_reg
fts_show_log
fts_test
fts_upgrade_app
```

3.3 Execute Test Command

fts_test is a node under sysfs, we use “echo” to execute test:

echo Conf_MultipleTest.ini > fts_test

3.4 Check Test Result

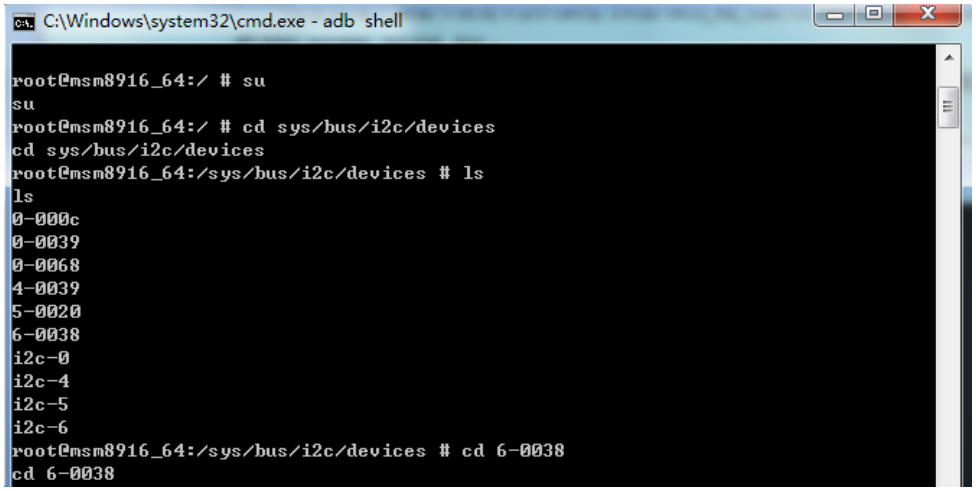
Test log is available by UART serial port and /proc/kmsg. You can get through the process of the test, such as test items, test data and test results. After test, a testdata.csv and a testresult.txt will be generated under /sdcard/, which you can pull to your PC by adb pull command.

```
root@msm8916_64:/sdcard # ls test*
ls test*
testdata.csv
testresult.txt
root@msm8916_64:/sdcard #
```

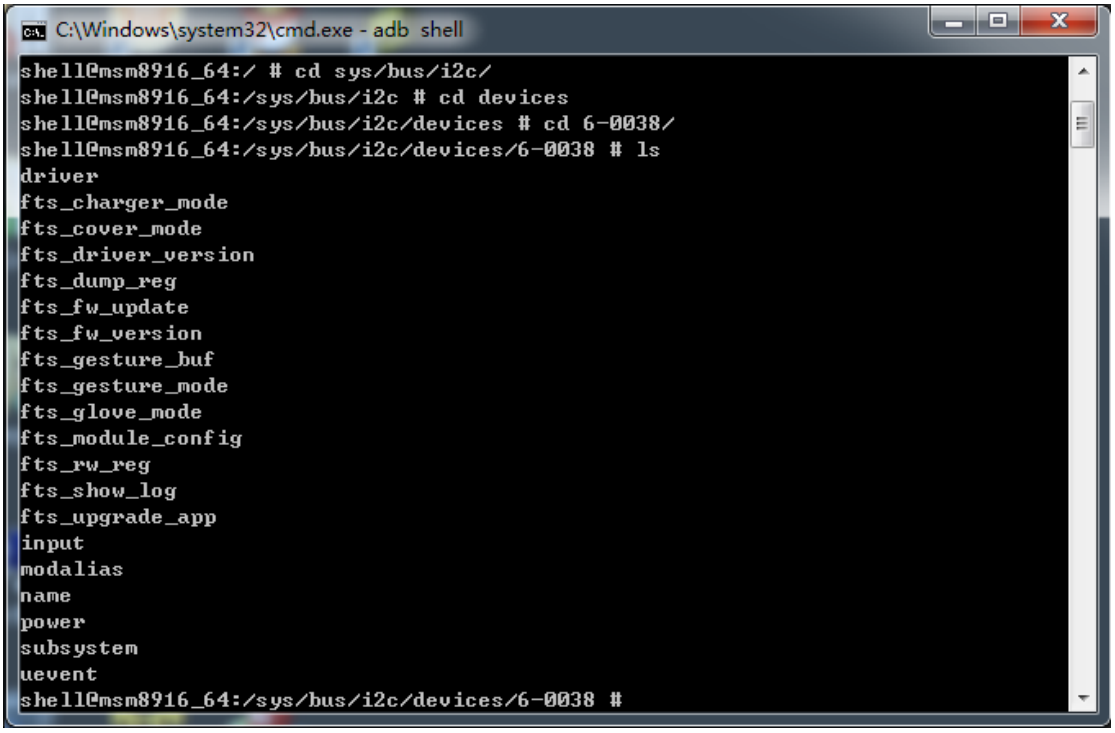
6 ADB Interface Nodes

6.1 Where are nodes

The debugging interface nodes are located at /sys/bus/i2c/devices/*-0038/, the specific number that * stands for depends on the i2c line you put your tp on.



use “ls” to list all nodes:



6.2 Module Nodes

1. Gesture module, an enabled FTS_GESTURE_EN is required

#echo 0 > fts_gesture_mode //Disable Gesture
#echo 1 > fts_gesture_mode //Enable Gesture
#cat fts_gesture_mode // show current Gesture status

2. Glove mode node, an enabled FTS_GLOVE_EN is required

#echo 0 > fts_glove_mode //Disable glove mode
#echo 1 > fts_glove_mode //Enable glove mode
#cat fts_glove_mode // show current glove status

3. Cover mode node, an enabled FTS_COVER_EN is required

#echo 0 > fts_cover_mode //Disable cover mode
#echo 1 > fts_cover_mode //Enable cover mode
#cat fts_cover_mode //read current cover status

4. Charger mode node, an enabled FTS_CHARGER_EN is required

#echo 0 > fts_charger_mode //Disable charger mode
#echo 1 > fts_charger_mode //Enable charger mode
#cat fts_charger_mode //show current charger mode

6.3 Debugging Nodes

ADB Interface nodes make driver debugging easier and more convenient.An enabled FTS_SYSFS_NODE_EN. is required.

1) #cat fts_driver_version // show driver version
2) #cat fts_fw_version // show firmware version
3) #cat fts_dump_reg // show key registers' values
4) #cat fts_show_log // show current FTS_INFO/FTS_ERROR log status
   #echo on > fts_show_log // enable FTS_INFO/FTS_ERROR
   #echo off > fts_show_log // disable FTS_INFO/FTS_ERROR
5) #echo 00 > fts_rw_reg // read register 0x00
   #echo 0040 > fts_rw_reg // write 0x00 as 0x40
   #cat fts_rw_reg // show result of last reading/writing
6) #cat fts_esd_check // show ESD check status
   #echo on > fts_esd_check // enable ESD
   #echo off > fts_esd_check // disable ESD
7) #echo *_app.bin > fts_upgrade_app // upgrade FW by .bin, you need push *_app.bin into /sdcard
8) #echo 1 > fts_fw_update //upgrade FW by .i in driver
9) Enable/Disable IRQ Node
   #echo 1 > fts_irq // Enable irq

#echo 0 > fts_irq //Disable irq

10) hardware reset TP Node

#cat fts_hw_reset //Reset TP