

MTK平台芯片驱动指导文档

芯片驱动指导文档（MTK）	
Project name	Touch panel
Document ref	[Document ref]
Version	1.2
Release date	2016.12.29
Owner	Driver Team, FocalTech
Classification	
Distribution List	
Approval	

This document contains information proprietary to FocalTech Systems, Ltd., and may not be reproduced, disclosed or used in whole or part without the express written permission of FocalTech Systems, Ltd.

**Copyright © 2016, FocalTech Systems, Ltd
All rights reserved**

**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
2016.12.29	1.2	1. More IC Support	xiaoligen	
2016.10.18	1.1	1. extern mode, proximity, upgrade update 2. more ic supported	xiaoligen	
2016.08.12	1.0	first release	xiaoligen	

目录

1	概述.....	4
2	基本信息.....	4
3	驱动文件介绍.....	5
4	整合到平台.....	6
4.1	移植文件.....	6
4.2	如何编译.....	6
5	驱动功能配置.....	8
5.1	DTS 配置.....	8
5.2	功能模块配置.....	9
5.3	升级设置.....	10
5.4	量产测试设置.....	11
6	ADB 调试接口.....	13
6.1	查找节点.....	13
6.2	功能节点.....	14
6.3	调试节点.....	14

1 概述

本文档介绍敦泰科技（FocalTech）触摸屏驱动程序主要功能、文件结构以及如何移植到 MTK 平台。

注：本驱动由 Generator.exe 生成!

2 基本信息

基本信息	
支持芯片系列	FT8716、FT8736、FT8006、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
支持平台	MTK 所有平台
APK/ADB 工具	支持
其他功能	GESTURE、ESD

（以寄存器 0xA3 判断芯片种类，如 FT8716 为 0x87）

3 驱动文件介绍

驱动文件存放在 focaltech_touch 文件夹里，实现了驱动挂载、触摸点上报、休眠唤醒、手势唤醒、FW 升级等功能及 APK 和 ADB 调试调用的接口等功能。下面列表是每一个文件的功能简介：

文件名	属性	功能
Makefile	必选	Makefile 文件
Kconfig	必选	Kconfig 文件
focaltech_core.c	必选	驱动主功能文件，用来实现驱动的挂载、读取触摸数据的上报坐标、休眠唤醒处理等功能
focaltech_core.h	必选	驱动主功能头文件
focaltech_i2c.c	必选	驱动 I2C 通信功能文件
focaltech_flash.c	必选	固件升级功能文件
focaltech_flash.h	必选	固件升级头文件
focaltech_esdcheck.c	必选	ESD 检测功能文件
focaltech_config.h	必选	可配置功能模块的 Eanble/disable 头文件
focaltech_common.h	必选	平台通用的定义和声明
focaltech_gesture.c	可选	手势唤醒功能文件
focaltech_ex_mode.c	可选	手套、皮套、充电器等特殊功能支持
focaltech_ex_fun.c	可选	扩展功能实现文件，主要用于支持ADB, APK 调试。该文件不是必需的，但是推荐在驱动中增加该功能
focaltech_sensor.c	可选	Sensor功能文件
focaltech_proximity.c	可选	接近感应功能文件
focaltech_test	可选	量产测试功能文件夹
focaltech_point_report_check.c	可选	检测无中断后一段时间后全报UP事件
include		
firmware/FT8716_app_sample.i	必选	固件升级所需 app 文件(默认为空,需进行替换)
pramboot	可选	如果升级需要使用到pramboot文件，则会包含相关文件

4 整合到平台

4.1 移植文件

1. 将驱动文件打包到 focaltech_touch 文件夹，并将 focaltech_touch 文件夹复制到 kernel-xx/drivers/input/touchscreen/mediatek 目录下
2. 修改 mediatek 目录下的 Kconfig 文件，在这个文件的末尾增加如下一行：
source “drivers/input/touchscreen/mediatek/focaltech_touch/Kconfig”
3. 修改 mediatek 目录下的 Makefile 文件，在这个文件的末尾增加如下一行：
obj-\$(CONFIG_TOUCHSCREEN_MTK_FOCALTECH_TS) += focaltech_touch/

4.2 如何编译

(1) 调出 menuconfig, 选择 fts;

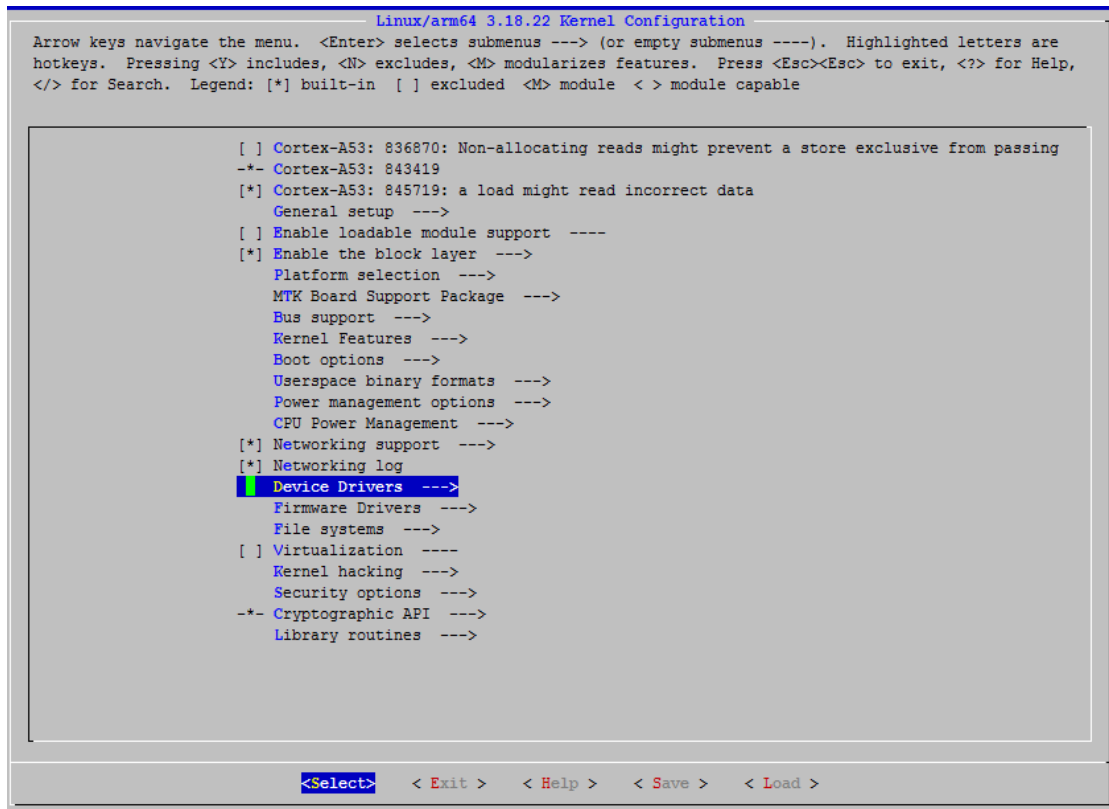
例如:

```
$ source build/envsetup.sh
$ lunch full_amt6797_evb_m-userdebug
$ mmm kernel-3.18:menuconfig-kernel
```

或

```
$ make menuconfig
```

出现下图所示的 Kernel Configuration 界面，选择 Device Drivers -> Input Device Support -> Touchscreens -> Focaltech Ts for Mediatek



如果需要修改驱动的文件名(默认为 focaltech_touch), 定义 "Focaltech Ts for Mediatek" 以下的 "Focaltech ts directory name", 修改为您需要修改的名称.

```
[*] Focaltech Ts for Mediatek
(focaltech_touch) Focaltech ts directory name
```

(2) 编译 boot.img;

\$ make bootimage -j4

5 驱动功能配置

DTS 中对分辨率和按键等进行配置，focaltech_config.h 中对驱动功能模块进行配置.

5.1 DTS 配置

示例:

Touch DTS 路径: alps/kernel-3.18/arch/arm64/boot/dts/amt6797_evb_m.dts

```
&touch {
    tpd-resolution = <1080 1920>;          // 分辨率
    use-tpd-button = <0>;                  // 是否有按键
    tpd-key-num = <3>;                     // 按键数量
    // 按键值, 按键对应的值: KEY_MENU, KEY_HOME, KEY_BACK
    // 定义位于 dts/include/dt-bindings/input/input.h
    tpd-key-local = <139 172 158 0>;
    // 按键对应的矩形区域(中心点x, 中心点y, width, height)
    tpd-key-dim-local = <200 2000 100 40 515 2000 100 40 800 2000 100 40 0 0 0 0>;
    tpd-max-touch-num = <5>;              // 最大触点个数
    tpd-filter-enable = <0>;              // 是否开启防抖
    tpd-filter-pixel-density = <192>;
    tpd-filter-custom-prameters = <0 0 0 0 0 0 0 0 0 0 0>;
    tpd-filter-custom-speed = <0 0 0>;
    pinctrl-names = "default", "state_eint_as_int", "state_eint_output0", "state_eint_output1",
    "state_rst_output0", "state_rst_output1";
    pinctrl-0 = <&ctp_pins_default>;
    pinctrl-1 = <&ctp_pins_eint_as_int>;    // 设为INT
    pinctrl-2 = <&ctp_pins_eint_output0>;  // INT输出低
    pinctrl-3 = <&ctp_pins_eint_output1>;  // INT输出高
    pinctrl-4 = <&ctp_pins_rst_output0>;   // RST输出低
    pinctrl-5 = <&ctp_pins_rst_output1>;   // RST输出高
    status = "okay";
};
// INT & RST操作函数
&pio {
    ctp_pins_default: eint0default {
    };
    ctp_pins_eint_as_int: eint@0 { // 设为INT0
        pins_cmd_dat {
            pins = <PINMUX_GPIO85_FUNC_EINT8>;
            slew-rate = <0>;
            bias-disable;
        };
    };
    ctp_pins_eint_output0: eintoutput0 { // INT输出低
        pins_cmd_dat {
            pins = <PINMUX_GPIO85_FUNC_GPIO85>;
            slew-rate = <1>;
            output-low;
        };
    };
    ctp_pins_eint_output1: eintoutput1 { // INT输出高
        pins_cmd_dat {
            pins = <PINMUX_GPIO85_FUNC_GPIO85>;
            slew-rate = <1>;
            output-high;
        };
    };
};
```

THIS DOCUMENT CONTAINS INFORMATION PROPRIETARY TO FOCALTECH SYSTEMS, LTD., AND MAY NOT BE REPRODUCED, DISCLOSED OR USED IN WHOLE OR PART WITHOUT WRITTEN PERMISSION OF FOCALTECH SYSTEMS, LTD.


```

    };
};
ctp_pins_rst_output0: rstoutput0 { // RST输出低
    pins_cmd_dat {
        pins = <PINMUX_GPIO68_FUNC_GPIO68>;
        slew-rate = <1>;
        output-low;
    };
};
ctp_pins_rst_output1: rstoutput1 { // RST输出高
    pins_cmd_dat {
        pins = <PINMUX_GPIO68_FUNC_GPIO68>;
        slew-rate = <1>;
        output-high;
    };
};
};
};

```

5.2 功能模块配置

驱动中相关模块是可以通过配置 focaltech_config.h 中相关的功能宏进行 enable(1)和 disable(0)。

在此之前，请先选择相应的 IC 类型：

宏	功能	默认
FTS_CHIP_TYPE	当前使用的芯片类型，根据实际使用的IC进行定义，IC列表在宏定义上方“chip type defines”	_FT8716

接下来配置功能模块：

宏	功能	默认
FTS_DEBUG_EN	打印 debug log 信息,debug 驱动时建议打开,release 版本时建议关闭	Enable(debug), Disable(release)
FTS_MT_PROTOCOL_B_EN	Linux 多点触控协议开关, enable(B 协议), disable(A 协议)	Enable
FTS_REPORT_PRESSURE_EN	Multi-Touch A/B 上报 pressure 值, 默认打开	Enable
FTS_FORCE_TOUCH_EN	如果打开则上报实际的 pressure, 否则上报恒定的 pressure	Disable
FTS_GESTURE_EN	手势功能开关, enable(开启), disable(关闭)	Disable
FTS_ESDCHECK_EN	ESD 静电保护机制,每隔 1s 检测一次 ESD 状态, 异常则复位 IC, enable(开启), disable(关闭)	Disable
FTS_TEST_EN	量产测试, 提供整机测试功能, 用户可以通过 ADB 命令来执行测试, 以验证 TP 模组之好坏, 开启需加入 focaltech_test 文件夹	Disable
FTS_GLOVE_EN	手套模式支持	Disable
FTS_COVER_EN	皮套模式支持	Disable
FTS_CHARGER_EN	USB 充电器插入拔出检测	Disable

FTS_PSENSOR_EN	proximity 接近感应功能支持	Disable
FTS_SYSFS_NODE_EN	APK/PC 工具调试结点	Enable
FTS_APK_NODE_EN		Enable
FTS_POWER_SOURCE_CUST_EN	自定义 TP 电源控制	Disable

升级相关		
FTS_AUTO_UPGRADE_EN	自动升级开关	Enable
FTS_AUTO_UPGRADE_FOR_LCD_CFG_EN	自动升级 LCD config	Disable
FTS_AUTO_CLB_EN	自动 CB 校验	Disable
FTS_UPGRADE_FW_APP	升级 app.i 文件(需要替换,sample.i 是无效的)	Sample.i
FTS_UPGRADE_LCD_CFG	升级 FLASH 中 lcd cfg 区域所使用的.i 文件(需要替换)	lcd_cfg.i
FTS_GET_VENDOR_ID	升级是否判断 Vendor ID	Disable
FTS_VENDOR_1_ID	LCD Vendor 1 的 ID	0x8D
FTS_VENDOR_2_ID	如果有 2 家 Vendor, 定义之, 如果只 1 家, 定义为与 1 一样	0x8D
压力测试		
FTS_UPGRADE_STRESS_TEST	压力测试, 使能后, 会重复升级指定的次数。驱动调好后, 建议做压力测试, 以检测升级功能的可靠性	Disable
FTS_UPGRADE_TEST_NUMBER	压力测试次数	1000

5.3 升级设置

自动升级 FTS_AUTO_UPGRADE_EN 是默认开启的, 移植时升级所使用到的文件按照以下进行配置:

FTS_CHIP_TYPE:

选择对应的 IC 类型

FTS_UPGRADE_FW_APP

把 App.i 文件放置到/include/firmware 目录下

修改宏中 app.i 的名称, 使其与放置到/include/firmware 目录下的文件同名

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

THIS DOCUMENT CONTAINS INFORMATION PROPRIETARY TO FOCALTECH SYSTEMS, LTD., AND MAY NOT BE REPRODUCED, DISCLOSED OR USED IN WHOLE OR PART WITHOUT WRITTEN PERMISSION OF FOCALTECH SYSTEMS, LTD.

FTS_GET_VENDOR_ID

如果需要判断 vendor ID，则需打开此宏，并设置 FTS_VENDOR_1_ID & FTS_VENDOR_2_ID

FTS_VENDOR_1_ID

TP 所对应屏厂 1 玻璃的 ID，需要所属屏厂提供，以区别不同的屏

FTS_VENDOR_2_ID

- TP 所对应屏厂 2 玻璃的 ID，需要所属屏厂提供，以区别不同的屏
- 定义两个 vendor id 是考虑到有些项目同时使用两家屏厂的屏

5.4 量产测试设置

打开 focaltech_config.h 中的宏 FTS_TEST_EN，然后配置 IC 对应的宏。

- (1) 在 focaltech_config.h 中把测试功能的使能宏 FTS_TEST_EN 打开。

```
/*
 * 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) 在 focaltech_test_config.h 中把红框中 FTS_CHIP_TEST_TYPE 的宏改为所需要测试的 IC，例如 8716 就把宏 FTS_CHIP_TEST_TYPE 改为 FT8716_TEST，#ifdef FTS_CHIP_TYPE 下的内容是为兼容新版驱动，不用修改。不用再增加具体 IC 类型，只需确定项目所用到的具体 IC 是属于哪一系列的 IC。

```

/*-----
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

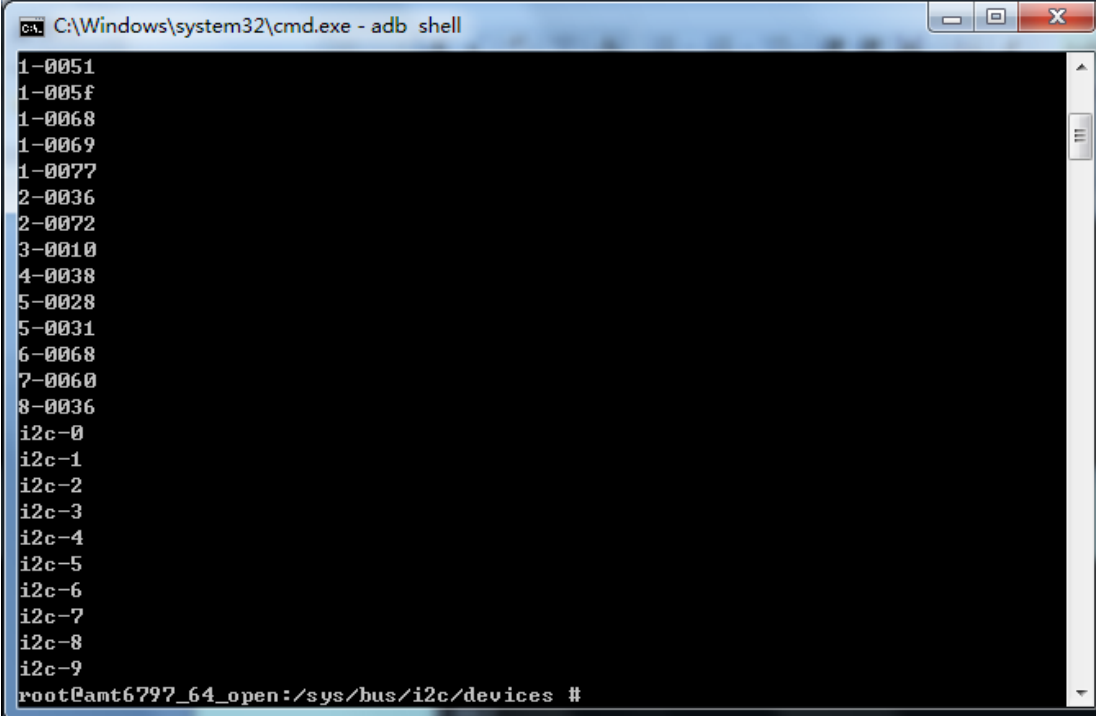
#endif

```

6 ADB 调试接口

6.1 查找节点

生成的节点在系统的/sys/bus/i2c/devices/*-0038/这个目录下，具体的结点要看设备挂载在哪一路I2C上，以下示例为4-0038：



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - adb shell". The window displays a list of I2C devices connected to the system. The list includes addresses like 1-0051, 1-005f, 1-0068, 1-0069, 1-0077, 2-0036, 2-0072, 3-0010, 4-0038, 5-0028, 5-0031, 6-0068, 7-0060, 8-0036, and a series of i2c-0 through i2c-9. The prompt at the bottom is "root@amt6797_64_open:/sys/bus/i2c/devices #".

```
C:\Windows\system32\cmd.exe - adb shell
1-0051
1-005f
1-0068
1-0069
1-0077
2-0036
2-0072
3-0010
4-0038
5-0028
5-0031
6-0068
7-0060
8-0036
i2c-0
i2c-1
i2c-2
i2c-3
i2c-4
i2c-5
i2c-6
i2c-7
i2c-8
i2c-9
root@amt6797_64_open:/sys/bus/i2c/devices #
```

查看所有节点列表



A screenshot of an ADB shell window showing the contents of the directory /sys/bus/i2c/devices/4-0038. The prompt is "root@amt6797_64_open:/sys/bus/i2c/devices/4-0038 #". The user has entered the command "ls", and the output lists various files and directories: driver, fts_driver_version, fts_dump_reg, fts_fw_update, fts_fw_version, fts_module_config, fts_rw_reg, fts_show_log, fts_upgrade_app, modalias, name, power, subsystem, uevent, and _.

```
root@amt6797_64_open:/sys/bus/i2c/devices/4-0038 # ls
driver
fts_driver_version
fts_dump_reg
fts_fw_update
fts_fw_version
fts_module_config
fts_rw_reg
fts_show_log
fts_upgrade_app
modalias
name
power
subsystem
uevent
root@amt6797_64_open:/sys/bus/i2c/devices/4-0038 # _
```

6.2 功能节点

1. 手势功能节点, 需要打开宏 FTS_GESTURE_EN

```
#echo 0 > fts_gesture_mode //Disable Gesture功能
#echo 1 > fts_gesture_mode //Enable Gesture功能,只有向节点fts_gesture_mode写1, 手势功能才会生效
#cat fts_gesture_mode // 查看Gesture的状态
```

2. 手套功能节点, 需要打开宏 FTS_GLOVE_EN

```
#echo 0 > fts_glove_mode //Disable手套功能
#echo 1 > fts_glove_mode //Enable手套功能
#cat fts_glove_mode //读取手套功能状态
```

3. 皮套功能节点, 需要打开宏 FTS_COVER_EN

```
#echo 0 > fts_cover_mode //Disable皮套功能
#echo 1 > fts_cover_mode //Enable皮套功能
#cat fts_cover_mode //读取皮套功能状态
```

4. 充电器功能节点, 需要打开宏 FTS_CHARGER_EN

```
#echo 0 > fts_charger_mode //Disable充电器功能
#echo 1 > fts_charger_mode //Enable充电器功能
#cat fts_charger_mode //读取充电器功能状态
```

5. 接近感应功能节点, 需要打开宏 FTS_PSENSOR_EN

```
#echo 0 > fts_touch_proximity_mode //Disable接近感应功能
#echo 1 > fts_touch_proximity_mode //Enable接近感应功能
#cat fts_touch_proximity_mode //读取接近感应功能状态
```

6.3 调试节点

为了方便项目的调试我们在驱动端提供了供 ADB 命令调用的接口。与生成相关调试节点有关的代码在 focaltech_ex_fun.c, 请务必打开 focaltech_config.h 中的宏 FTS_SYSFS_NODE_EN.

- 1) #cat fts_driver_version // 查看driver的版本号
- 2) #cat fts_fw_version // 查看firmware的版本号
- 3) #cat fts_dump_reg // 查看某些特征寄存器的值
- 4) #cat fts_show_log // 查看底层FTS_INFO/FTS_ERROR log状态

```
#echo on > fts_show_log // 开启 FTS_INFO/FTS_ERROR
#echo off > fts_show_log // 关闭 FTS_INFO/FTS_ERROR
```
- 5) #echo 00 > fts_rw_reg // 读取寄存器0x00的值

```
#echo 0040 > fts_rw_reg // 写0x00为0x40
#cat fts_rw_reg // 查看上一步读写操作的结果
```
- 6) #cat fts_esd_check // 查看ESD check的状态

```
#echo on > fts_esd_check // 开启ESD
```

```
#echo off > fts_esd_check // 关闭ESD
```

7) #echo *_app.bin > fts_upgrade_app // 通过.bin文件来升级FW， 必须先把*_app.bin放入sdcard目录下

8) #echo 1 > fts_fw_update //通过驱动中.i文件来升级FW