TP Driver Porting Guide for Qualcomm

TP Driver Porting Guide for Qualcomm	
Project name	Touch panel
Document ref	[Document ref]
Version	1.0
Release date	2023.11.07
Owner	Driver Team, Hynitron
Classification	
Distribution List	
Approval	

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

No.4, Lane 36, Xuelin Road, Pudong New Area, Shanghai.

Tel: 021-5027012

Email: support@hynitron.com.cn

www.hynitron.com

TP Driver Porting Guide for Qualcomm

Revision History

Date	Version	List of changes	Author	Approved by
2023.11.07	1.0	1.More IC Support	ChaoWu	

Contents

1 Abstract

This guide introduces the structure and functions of "Hynitron TP Driver" based on linux environment and a porting reference for your Qualcomm platform, step by step.

Through this guide, you can get how to porting "Hynitron TP Driver" to Qualcomm platform, containing Makefile/DTS modifying, kernel configuration and compiling.

You can also customize your functions of TP, containing MULTI-TOUCH protocol, GESTURE en/dis, ESDCHECK en/dis, FACTORY TEST in driver en/dis, FIRMWARE UPGRADE and so on.

Note: Examples used in the guide are verified in Qualcomm Dragonboard 410c platform, only a reference, maybe different with your platform; please refer to your platform's document when porting the driver into your platform.

2 Interface Setting Recommendation

2.1 I2C interface

I2C speed recommendation: 400K (For all Hynitron's IC)

2.2 SPI interface

SPI speed recommendation: 8M(For CST6x IC)

3 File Structure

Table 1 shows the detail description of driver files

Component	Files	Attribute	Descriptor
Complier	Makefile		Use for kernel compiling and configuration.
Main	hyn_cfg.h hyn_core.c hyn_core.h		The main function of TP driver, that containing driver registration, bus interface (SPI/I2C) initialization, suspend/resume, multi-touch protocol support and so on. You can customize functions of TP driver by modifying hyn_cfg.h.
Interface	hyn_i2c.c		Bus communication with I2C/SPI;

	hyn_spi.c		
esdcheck	hyn_ts_ext.c		Use to process ESD check function.
gesture	hyn_gesture.c		Use to process gesture function.
proc	hyn_fs_node.c		Create sysfs/proc node, use to communicate with APK or ADB.
chip	Hyn_cst***.c		Use for upgrade, factory test, report point.

4 Porting TP Driver to Qualcomm Platform

4.1 Copy driver files into kernel

Copy “hyn_driver_release” directory into kernel directory(kernel/drivers/input/touchscreen);
Then modify Makefile:

Add the line below to kernel/drivers/input/touchscreen/Makefile:

```
obj-y += hyn_driver_release/
```

4.2 Configure DTS

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

I2C Interface Example:

```
hynitron@5A {
    compatible = "hyn,66xx";
    reg = <0x5A>;
    vdd_ana-supply = <&pm8953_l10>;
    vcc_i2c-supply = <&pm8953_l6>;

    interrupt-parent = <&tlmm>;
    interrupts = <65 0x02>;
    reset-gpio = <&tlmm 64 0x01>;
    irq-gpio = <&tlmm 65 0x02>;

    pinctrl-names = "ts_active","ts_suspend";
    pinctrl-0 = <&ts_int_active &ts_reset_active>;
    pinctrl-1 = <&ts_int_suspend &ts_reset_suspend>;

    max-touch-number = <5>;
    display-coords = <0 0 800 1280>;
    pos-swap = <0>;
    posx-reverse = <0>;
    posy-reverse = <0>;

    key-number = <0>;
    keys = <139 102 158>;
    key-y-coord = <2000>;
    key-x-coords = <200 600 800>;
};
```

SPI Interface Example:

```
hynitron@0 {
    compatible = "hyn,66xx";
    reg = <0x0>;
    vdd_ana-supply = <&pm8953_l10>;
    vcc_i2c-supply = <&pm8953_l6>;

    interrupt-parent = <&tlmm>;
    interrupts = <65 0x02>;
    reset-gpio = <&tlmm 64 0x01>;
    irq-gpio = <&tlmm 65 0x02>;

    pinctrl-names = "ts_active","ts_suspend";
    pinctrl-0 = <&ts_int_active &ts_reset_active>;
    pinctrl-1 = <&ts_int_suspend &ts_reset_suspend>;

    max-touch-number = <5>;
    display-coords = <0 0 800 1280>;
    pos-swap = <0>;
    posx-reverse = <0>;
    posy-reverse = <0>;

    key-number = <0>;
    keys = <139 102 158>;
    key-y-coord = <2000>;
    key-x-coords = <200 600 800>;
};
```

pinctrl Example:

```
pmx_ts_int_active {
    ts_int_active: ts_int_active {
        mux {
            pins = "gpio65";
            function = "gpio";
        };

        config {
            pins = "gpio65";
            drive-strength = <8>;
            bias-pull-up;
        };
    };
};

pmx_ts_reset_active {
    ts_reset_active: ts_reset_active {
        mux {
            pins = "gpio64";
            function = "gpio";
        };

        config {
            pins = "gpio64";
            drive-strength = <8>;
            bias-pull-up;
        };
    };
};

pmx_ts_int_suspend {
    ts_int_suspend: ts_int_suspend {
        mux {
            pins = "gpio65";
            function = "gpio";
        };

        config {
            pins = "gpio65";
            drive-strength = <2>;
            bias-pull-down;
        };
    };
};

pmx_ts_reset_suspend {
    ts_reset_suspend: ts_reset_suspend {
        mux {
            pins = "gpio64";
            function = "gpio";
        };

        config {
            pins = "gpio64";
            drive-strength = <2>;
            bias-pull-down;
        };
    };
};
```

You can also refer to readme.txt for detail description.

4.3 Compile kernel and generate boot.img

```
$ make bootimage -j64
```

5 Driver Customization

You can modify the file of hyn_cfg.h to customize your driver components.

I2C_PORT
Select I2C or SPI interface.

I2C_USE_DMA
Set I2C transfer mode.

HYN_UART_DEBUG_EN
Enable/Disable debug log.

HYN_APK_DEBUG_EN
Enable/Disable debug node.

HYN_GESTURE_EN
Enable/Disable gesture function.

MAX_POINTS_REPORT
Set the maximum number of points supported.

ESD_CHECK_EN

Enable/Disable ESD check function.

HYN_MT_PROTOCOL_B_EN

Set the linux multi-touch protocol, Enable:Protocol B, Disable: Protocol A.

6 Sysfs Nodes

6.1 Where are these sysfs nodes

a.I2C interface

When interface is I2C, then sysfs nodes will be generated under /sys/bus/i2c/devices/*-005a/directory, "*" stands for I2C bus number depending on your project design.

b.SPI interface

Similarly, when interface is SPI, then sysfs nodes will be generated under /sys/bus/spi/devices/spi*/ directory, "*" stands for SPI bus number depending on your project design.

Warning: Sysfs nodes may be different with different driver version