



SiS I2C Touch Driver Porting Guide

深圳领见科技有限公司

Rev. 2.19

Dec. 12, 2017

This specification is subject to change without notice. Silicon Integrated Systems Corporation assumes no responsibility for any errors contained herein.

Copyright by Silicon Integrated Systems Corp., all rights reserved.



Contents

1. INTRODUCTION..... 2

2. SOURCE CODE OVERVIEW 3

2.1. The Source Code Tree Structure..... 3

2.2. Source Files..... 3

2.2.1. Makefile 3

2.2.2. Kconfig 3

2.2.3. I2C Touch Driver..... 4

2.2.4. Set SiS i2c device node permission 4

3. MODIFYING MAIN PROGRAM FOR DIFFERENT REQUIREMENTS OF ENVIRONMENT 5

3.1. Operation systems..... 5

3.1.1. Android 4.0 Issue 5

3.2. Set SiS i2c device node permission..... 5

3.2.1. System/core/rootdir/uevent.rc..... 5

3.3. Platform..... 6

3.3.1. I2C addresses 6

3.3.2. Board info..... 6

3.3.3. Device Tress Source (DTS)..... 7

3.4. Compiling Driver and set driver options in Kernel..... 8

3.4.1. Modify driver configurations..... 8

3.4.2. Make menuconfig to configure 10



1. Introduction

This document explains how to integrate sis touch driver into Android systems.

We provide a simple method to integrate and build SiS touch driver within different version of Linux kernel. Additionally, SiS I2C touch driver also uses system calls to requests a service from Linux kernel. How to define system calls will be described in the chapter, “Patching the Kernel with SiS I2C Touch Driver”.

This document contains three parts: source code overview, modifying main program for different operating systems, patching the kernel with SIS I2C touch driver. In the first part, we describe the usage of each file in the source code folder. In the second part, we explain an example about how to define system calls in Linux. On the other hand, the proper system call depends on platform.

深圳领见科技有限公司

2. Source Code Overview

2.1. The Source Code Tree Structure

An example of source code tree structure is as follows:



2.2. Source Files

2.2.1. Makefile

Makefile is located in “`$SOURCE_PATH/kernel/drivers/input/touchscreen`”.

Use this Makefile to compile SiS I2C touch driver. Our SiS I2C touch driver can be either enclosed in Linux kernel or compiled independently as a kernel module. Its usage will be described in the chapter, “Patching the Kernel with SiS I2C Touch Driver”.

2.2.2. Kconfig

Kconfig is located in “`$SOURCE_PATH/kernel/drivers/input/touchscreen`”.

This file is used only when enclosing SiS I2C touch driver in Linux kernel. Its usage will be described in the chapter, “Patching the Kernel with SiS I2C Touch Driver”.

2.2.3. I2C Touch Driver

Touch drivers are located in “\$SOURCE_PATH/kernel/drivers/input/touchscreen”.

The main program is “sis_i2c.c”, with a header file, “sis_i2c.h”. To compile main program for different operating systems will be described in “Modifying main program for different operating systems”.

NOTE : In default setting, **COPY** this two file to touchscreen folder, **needn't** to modify them.

2.2.4. Set SiS i2c device node permission

Applications/Tools use character device driver to requests a service to Linux kernel. The main program is in “sis_i2c.c”, but we have to set device node permission in “ueventd.rc”. How to set the permission will be described in the chapter, “Patching the Kernel with SiS I2C Touch Driver”.

Character device driver is implemented in SiS Touch driver. We should check

- implementation exists
- permission of SiS i2c device node is set
- kernel build successfully

3. Modifying main program for different requirements of environment

3.1. Operation systems

3.1.1. Android 4.0 Issue

Android 4.0 has to set attributions in IDC file:

```
touch.deviceType = touchScreen
touch.orentationAware = 1
device.internal = 1
```

IDC file path: /system/usr/idc/<device-name>.idc

<device-name> : default is “sis_touch”.

This part is for Android 4.0 only, the after versions needn't to do this.

3.2. Set SiS i2c device node permission

We provide character device driver for AP/Tools to communicate with firmware. To access the character device driver successfully, we should set the device node permission in “ueventd.rc”. To set device node permission, add the setting in Android source code:

3.2.1. System/core/rootdir/ueventd.rc

Define device name “sis_aegis_touch_device” or “sis_hydra_touch_device” which depends on chip type (please reference step 3.3 to check chip type), and set the permission “0666” and file owner to root in ueventd.rc. The device name has to match with **DEVICE_NAME** which is defined in “sis_i2c.c”.

The device name of SiS95xx series is “sis_hydra_touch_device”, The device name of SiS92xx series is “sis_aegis_touch_device”.

```
/dev/binder          0666  root    root
#sis
/dev/sis_aegis_touch_device  0666  root    root
```

3.3. Platform

3.3.1. I2C addresses

For kernel version beyond 2.6.35, address 0x04-0x07 and address above 0x78 are reserved (defined in `kernel/drivers/i2c/i2c-core.c`). We use `0x5c` as default I2C slave address which is defined in firmware.

```
#define SIS_SLAVE_ADDR    0x5c
```

The slave address **must match firmware setting**. Modify the slave address **is not recommended**.

3.3.2. Board info

WARNING! step 3.3.2 and 3.3.3 won't exist at the same time, please choose by platform require.

To register the slave address to I2C bus. Add I2C board info to corresponding file. The setting file depends on platform might have different name. the example is pandaboard. Modify and add the following board info to `kernel/arch/arm/mach-omap2/board-omap4panda.c`. The name `sis_i2c_ts` has to be same as `#define SIS_I2C_NAME "sis_i2c_ts"` in `sis_i2c.h`.

```
static struct i2c_board_info __initdata sis_i2c_boardinfo[] = {
    {
        I2C_BOARD_INFO("sis_i2c_ts", 0x5c),
    },
};
```

Then register I2C bus into initial function of board info,

```
omap_register_i2c_bus(4, 400, sis_i2c_boardinfo, ARRAY_SIZE(sis_i2c_boardinfo));
```

3.3.2.1. Interrupt setting of board info

For different platform, each user should configure GPIO pin for interrupt function first. For example, assume that we use GPIO pin number 0x7b which is set by Kconfig as our GPIO interrupt pin. To setting GPIO by Konfig, please reference [step 3.4.2.2](#) to set GPIO pin.

```
#define _I2C_INT_ENABLE
#define GPIO_IRQ          CONFIG_GPIO_INT_PIN_FOR_SIS
```

Then setting interrupt handler and trigger signal. The default trigger signal of SiS FW is **FALLING EDGE** trigger. And be sure to define the keyword `_I2C_INT_ENABLE`.

```
#define _I2C_INT_ENABLE
request_irq(client->irq, sis_ts_irq_handler, IRQF_TRIGGER_FALLING, client->name, ts);
```

3.3.3. Device Tress Source (DTS)

WARNING! step 3.3.2 and 3.3.3 won't exist at the same time, please choose by platform require.

Some of platform might using DTS as register style. Please follow [step 3.4.2.2](#) to set the option.

For DTS(Device Tree Source), we need to describe our hardware information. Here is an example on firefly-rk3288 and .dts is located in **kernel/arch/arm/boot/dts/firefly-rk3288.dts**.

Each user should configure GPIO pin for interrupt function first. For example, we use GPIO pin number 7 as our GPIO interrupt pin (`&gpio0 GPIO_A7 TRQ_TYPE_LEVEL_LOW`) and **set the i2c path on bus1**. Follow the functions below to configure GPIO pin and assign interrupt service routine to GPIO 7.

```
&i2c1 {
    status = "okay";
    rtc@51 {
        compatible = "nxp,pcf8563";
        reg = <0x51>;
    };
    sis_touchscreen@5c {
        compatible = "sis, sis_touch";
        reg = <0x5c>;
        touch-gpio = <&gpio0 GPIO_A7 IRQ_TYPE_LEVEL_LOW>;
        status = "okay";
    };
};
```

3.4. Compiling Driver and set driver options in Kernel

3.4.1. Modify driver configurations

First copy scripts listed below and paste it into the bottom of **kernel/drivers/input/touchscreen/Kconfig** (before the last endif).

```
menu "SiS touchscreen series"
choice
    prompt "SiS controller select"
    default y
config TOUCHSCREEN_SIS_I2C_92XX
    bool "SiS92xx series I2C touchscreen driver"
    depends on I2C
    help
        Say Y here to enable support for I2C connected SiS touchscreen.
        If unsure, say N.
        To compile this driver as a module, choose M here: the
        module will be called sis_i2c.
config TOUCHSCREEN_SIS_I2C_95XX
    bool "SiS95xx series I2C touchscreen driver"
    depends on I2C
    help
        Say Y here to enable support for I2C connected SiS touchscreen.
        If unsure, say N.
        To compile this driver as a module, choose M here: the
        module will be called sis_i2c.
endchoice
```

```
if TOUCHSCREEN_SIS_I2C_92XX || TOUCHSCREEN_SIS_I2C_95XX
config REG_BY_BOARDINFO
    bool "Using boardinfo setting (Disable if using DTS)"
    default y
    help
        This option is for boardinfo, please disable when using DTS.
if REG_BY_BOARDINFO
config GPIO_INT_PIN_FOR_SIS
    int "Enter the interrupt pin"
    default 123
    help
        Configure GPIO pin used for interrupt from SiS
        by entering unnumber in dec in decimal format.
endif
endif
endmenu
```

Then copy scripts listed below and paste it into **kernel/drivers/input/touchscreen/Makefile**.

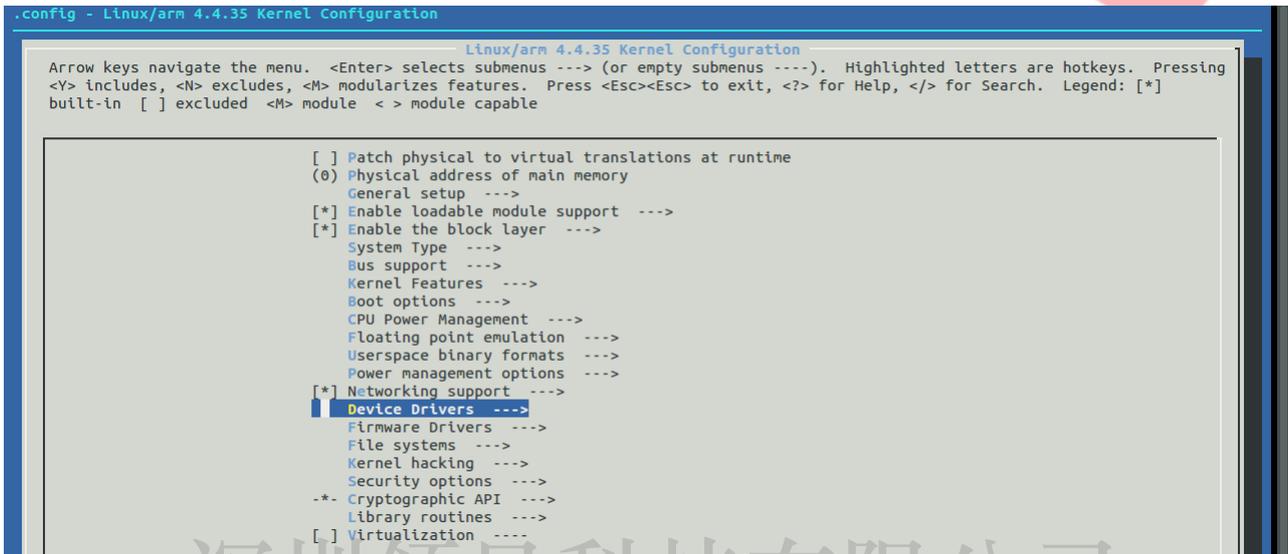
```
obj-$(CONFIG_TOUCHSCREEN_SIS_I2C_92XX) += sis_i2c.o
obj-$(CONFIG_TOUCHSCREEN_SIS_I2C_95XX) += sis_i2c.o
```

3.4.2. Make menuconfig to configure

First, make the configure file at the kernel folder. Command : **make menuconfig**.

Then, follow the below step to set configurations.

1. Move the light bar and Enter **Device drivers**

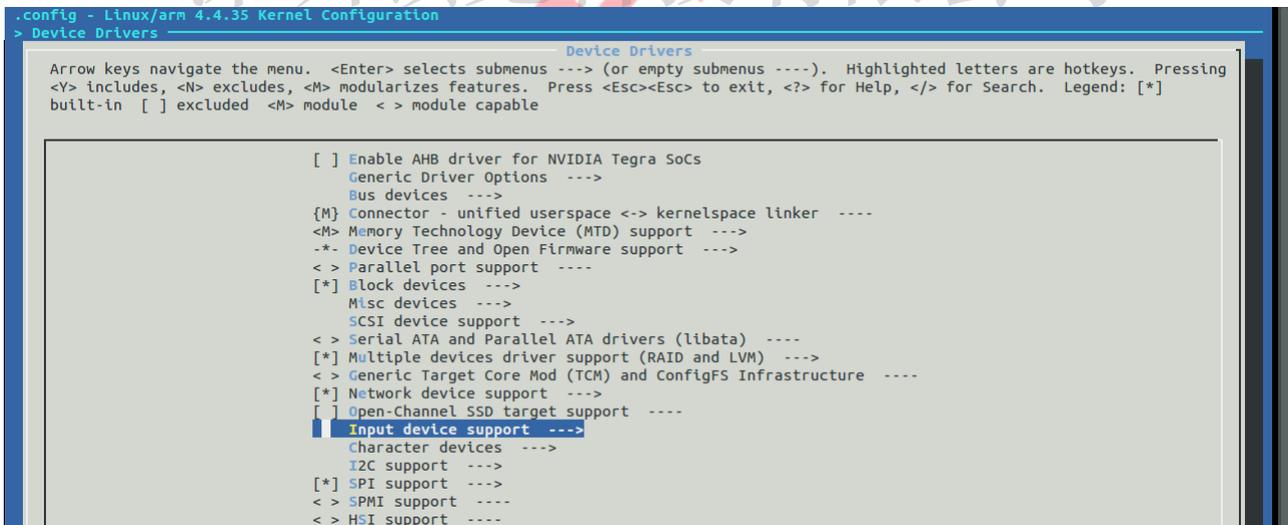


```

.config - Linux/arm 4.4.35 Kernel Configuration
Linux/arm 4.4.35 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
built-in [ ] excluded <M> module < > module capable

[ ] Patch physical to virtual translations at runtime
(0) Physical address of main memory
General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
System Type --->
Bus support --->
Kernel Features --->
Boot options --->
CPU Power Management --->
Floating point emulation --->
Userspace binary formats --->
Power management options --->
[*] Networking support --->
Device Drivers --->
Firmware Drivers --->
File systems --->
Kernel hacking --->
Security options --->
-* Cryptographic API --->
Library routines --->
[ ] Virtualization ----
  
```

2. Move the light bar and Enter **Input device support**



```

.config - Linux/arm 4.4.35 Kernel Configuration
> Device Drivers
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
built-in [ ] excluded <M> module < > module capable

[ ] Enable AHB driver for NVIDIA Tegra SoCs
Generic Driver Options --->
Bus devices --->
{M} Connector - unified userspace <-> kernelspace linker ----
<M> Memory Technology Device (MTD) support --->
-* Device Tree and Open Firmware support --->
< > Parallel port support ----
[*] Block devices --->
Misc devices --->
SCSI device support --->
< > Serial ATA and Parallel ATA drivers (libata) ----
[*] Multiple devices driver support (RAID and LVM) --->
< > Generic Target Core Mod (TCM) and ConfigFS Infrastructure ----
[*] Network device support --->
[ ] Open-Channel SSD target support ----
Input device support --->
Character devices --->
I2C support --->
[*] SPI support --->
< > SPMI support ----
< > HSI support ----
  
```

3. Move the light bar and Enter **[*]Touchscreens**, if not enabled, press SPACE key to change option to **[*]**.

```
.config - Linux/arm 4.4.35 Kernel Configuration
> Device Drivers > Input device support
Input device support
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----). Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
built-in [ ] excluded <M> module < > module capable

<*> Generic input layer (needed for keyboard, mouse, ...)
<*> Export input device LEDs in sysfs
-* Support for memoryless force-feedback devices
<M> Polled input device skeleton
< > Sparse keymap support library
< > Matrix keymap support library
*** Userland interfaces ***
<*> Mouse interface
[ ] Provide legacy /dev/psaux device
(1024) Horizontal screen resolution
(768) Vertical screen resolution
<M> Joystick interface
<*> Event interface
< > Event debugging
*** Input Device Drivers ***
[*] Keyboards --->
[ ] Mice ----
[*] Joysticks/Gamepads --->
[*] Tablets --->
[*] Touchscreens --->
[*] Miscellaneous devices ----
Hardware I/O ports --->
```

4. Move the light bar and Enter **SiS touchscreen series**

```
.config - Linux/arm 4.4.35 Kernel Configuration
> Device Drivers > Input device support > Touchscreens
Touchscreens
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----). Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
built-in [ ] excluded <M> module < > module capable

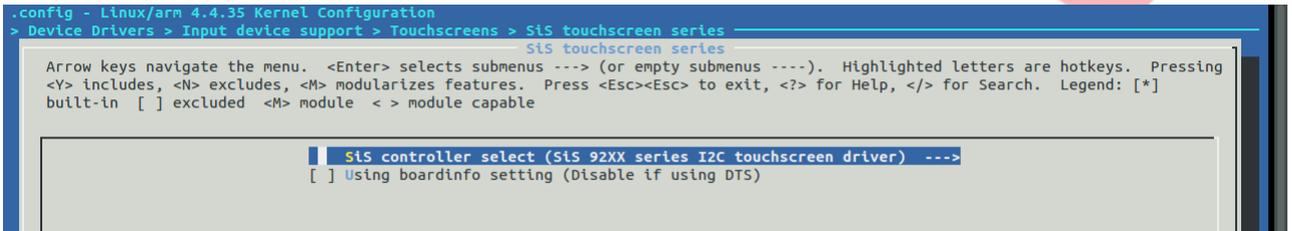
< > Philips UCB1400 touchscreen
< > PIXCIR I2C touchscreens
< > Weida HiTech I2C touchscreen
< > Support for WM97xx AC97 touchscreen controllers
<M> USB Touchscreen Driver
[*] eGalax, eTurboTouch CT-410/510/700 device support
[*] PanJit device support
[*] 3M/Microtouch EX II series device support
[*] ITM device support
[*] eTurboTouch (non-eGalax compatible) device support
[*] Gunze AHL61 device support
[*] DMC TSC-10/25 device support
[*] IRTOUCHSYSTEMS/UNITOP device support
[*] IdealTEK URTC1000 device support
[*] GeneralTouch Touchscreen device support
[*] GoTop Super_Q2/GogoPen/PenPower tablet device support
[*] JASTEC/DigiTech DTR-02U USB touch controller device support
[*] Elo TouchSystems 2700 IntelliTouch controller device support
[*] e2i Touchscreen controller (e.g. from Mimo 740)
[*] Zytronic controller
[*] ET&T USB series TC4UM/TC5UH touchscreen controller support
[*] NEXIO/iNexio device support
[*] EasyTouch USB Touch controller device support
< > Sahara TouchIT-213 touchscreen
< > TSC-10/25/40 serial touchscreen support
< > TSC2004 based touchscreens
< > TSC2005 based touchscreens
< > TSC2007 based touchscreens
< > Sitronix ST1232 touchscreen controllers
<M> STMicroelectronics STMPE touchscreens
< > Samsung SUR40 (Surface 2.0/PixelSense) touchscreen
< > Semtech SX8654 touchscreen
< > TPS6507x based touchscreens
< > Neonode zForce infrared touchscreens
< > ROHM BU21023/24 Dual touch support resistive touchscreens
[*] SiS touchscreen series --->
```

3.4.2.1. Select chip type

Please confirm the chip is 92xx series or 95xx series.

WARNING! The driver MUST MATCH WITH THE CHIP TYPE ON CONTROL BOARD.

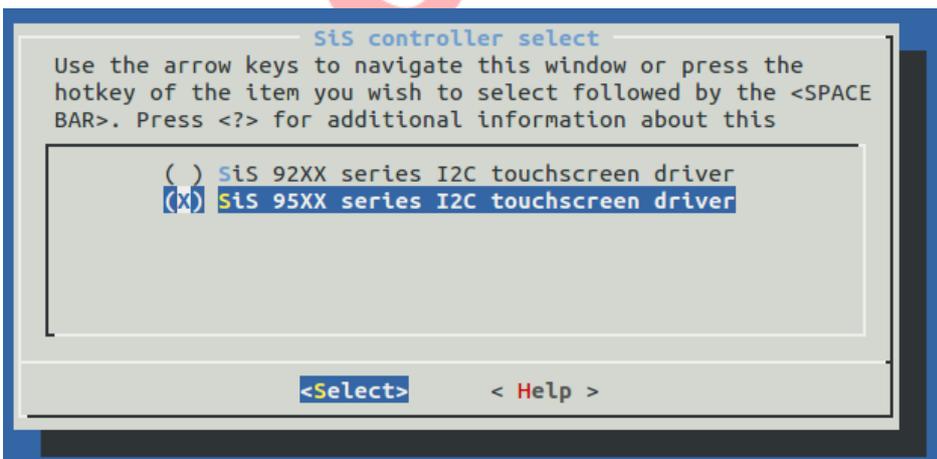
1. Move the light bar and Enter **SiS controller select**



2. Choose the chip type, if chip is 92xx series select **SiS92xx series I2C touchscreen driver**



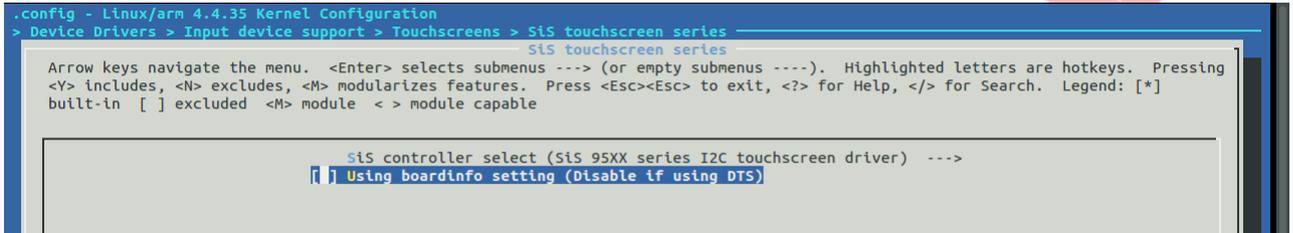
- Or if chip is 95xx series select **SiS95XX series I2C touchscreen driver**



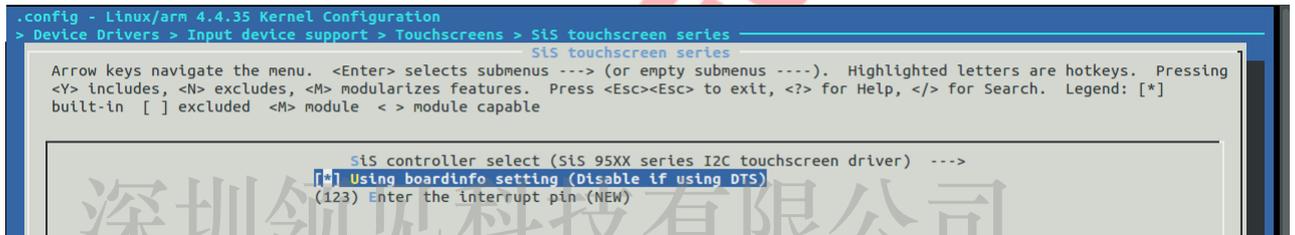
3.4.2.2. Select register type and set interrupt pin

There is two different ways to register I2C device on OS, they won't exist at the same time, please choose by platform require.

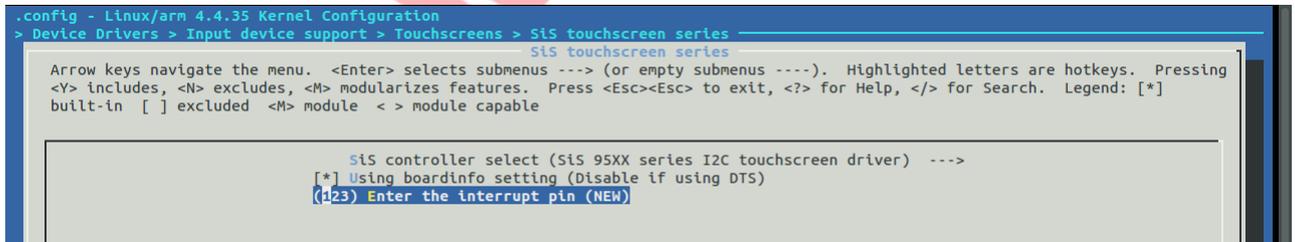
1. Back to **SiS touchscreen series** menu and select interrupt setting option.



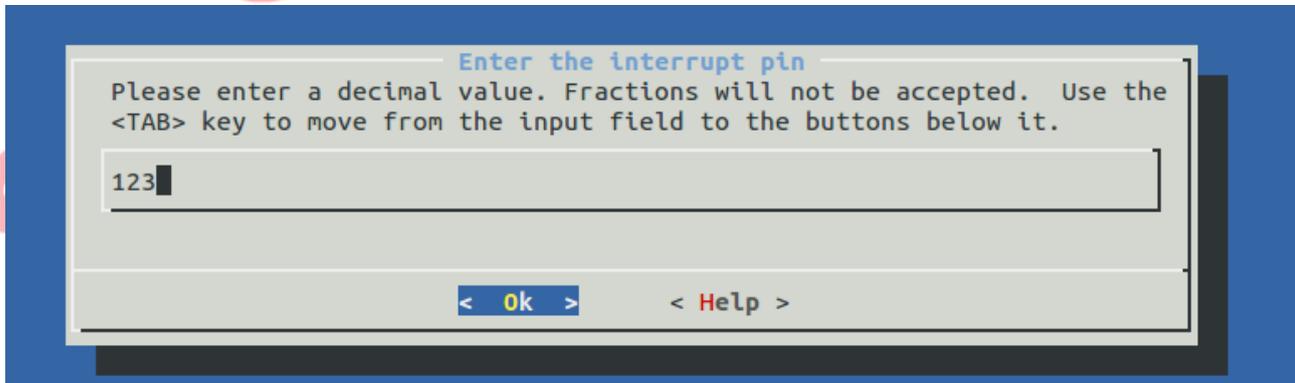
- 1.1 If use **boardinfo** as the register style, **Enable** the option **[*]Using boardinfo setting**, Change the option to **[*]** by press SPACE, then set the GPIO pin



Move the light bar to **Enter the interrupt pin**. The value of pin is using **decimal** format



Then enter the number of GPIO pin,



- 1.2 If the register style is DTS, keep and make sure the option []Using boardinfo setting disabled. If not, press SPACE to adjust it to disable [].

```

.config - Linux/arm 4.4.35 Kernel Configuration
> Device Drivers > Input device support > Touchscreens > SiS touchscreen series
SiS touchscreen series
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----). Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
built-in [ ] excluded <M> module < > module capable

SiS controller select (SiS 95XX series I2C touchscreen driver) --->
[ ] Using boardinfo setting (Disable if using DTS)
  
```

The GPIO pin setting of DTS is not here, please reference [step 3.3.3](#) to set the interrupt for DTS.

2. Finally, exit and save the configurations, rebuild kernel.

深圳领见科技有限公司

SiS CONFIDENTIAL