SiS I2C Touch Driver Porting Guide for Linux

Rev. 1.2 2023

Contents

1. Install Requirement

- Linux Kernel Source
- Development tools

2. Steps of Driver Installment

2.1. Kernel source patching

Please copy "hid-sis_ctrl.c" and "hid-sis_ctrl.h" into (kernel_src)/drivers/hid folder.

2.1.1. Modify the authority of driver for update-FW

Copy "99-sis-usb-touch.rules" into the /etc/udev/rules.d

2.1.2. Modify specialized driver list

Attention: Please check your kernel version.

Add "id-table" in hid_have_special_driver[] array. Please do not copy and paste whole file directly.

2.1.2.1. kernel version is < 4.16 or hid-quirks.c does not exist

Copy high-lighted code listed below and paste it into the bottom of **hid_have_special_driver**[] array at (kernel_src)/drivers/hid/**hid-core.c.** Please do not copy and paste whole file directly.

```
static const struct hid_device_id hid_have_special_driver[] = {
    { HID_USB_DEVICE(USB_VENDOR_ID_3M, USB_DEVICE_ID_3M1968) },
    …
    { HID_BLUETOOTH_DEVICE(USB_VENDOR_ID_NINTENDO,
    USB_DEVICE_ID_NINTENDO_WIIMOTE) },
    /* SiSdrv Start */
    { HID_USB_DEVICE(0x0457, HID_ANY_ID) },
    /* SiSdrv End */
    { }
};
```

2.1.2.2. kernel version is >= 4.16 or hid-quirks.c exists

Copy high-lighted code listed below and paste it into the bottom of **hid_have_special_driver**[] array at (kernel_src)/drivers/hid/**hid-quirk.c.** Please do not copy and paste whole file directly.

```
static const struct hid_device_id hid_have_special_driver[] = {
    #if IS_ENABLED(CONFIG_HID_A4TECH)
     { HID_USB_DEVICE(USB_VENDOR_ID_A4TECH,
    USB_DEVICE_ID_A4TECH_WCP32PU) },
     { HID_USB_DEVICE(USB_VENDOR_ID_A4TECH,
    USB_DEVICE_ID_A4TECH_X5_005D) },
     { HID_USB_DEVICE(USB_VENDOR_ID_A4TECH, USB_DEVICE_ID_A4TECH_RP_649) },
    #endif
    …
    #if IS_ENABLED(CONFIG_HID_ZYDACRON)
     { HID_USB_DEVICE(USB_VENDOR_ID_ZYDACRON,
    USB_DEVICE_ID_ZYDACRON_REMOTE_CONTROL) },
    #endif
    /* SiSdrv Start */
    #if IS_ENABLED(CONFIG_HID_SIS92XX) ||
    IS_ENABLED(CONFIG_HID_SIS95XX_SIS98XX)
    { HID_USB_DEVICE(0x0457, HID_ANY_ID) },
    #endif
    /* SiSdrv End */
    { }
    };
```

2.1.3. Modify the start/finish functions in driver

Copy high-lighted code listed below and paste it into (kernel_src)/drivers/hid/hid-multitouch.c. Please do not copy and paste whole file directly.

2.1.3.1. Include File

Add hid-sis_ctrl.h into include file list.

```
#include <linux/string.h>
#include <linux/timer.h>
 /* SiSdrv Start */
#include "hid-sis_ctrl.h"
/* SiSdrv End */
```

2.1.3.2. Class definition

Copy high-lighted code listed below to bottom of vendor specific classes.

```
/* vendor specific classes */
…
#define MT_CLS_VTL                  0x0110
#define MT_CLS_GOOGLE               0x0111
#define MT_CLS_RAZER_BLADE_STEALTH  0x0112
/* SiSdrv Start */
#define MT_CLS_SIS                  0x0457
/* SiSdrv End */
```

Copy high-lighted code listed below to bottom of structure “static const struct mt_class mt_classes[]”

```
static const struct mt_class mt_classes[] = {
…
        { .name = MT_CLS_RAZER_BLADE_STEALTH,
                .quirks = MT_QUIRK_ALWAYS_VALID |
                        MT_QUIRK_IGNORE_DUPLICATES |
                        MT_QUIRK_HOVERING |
                        MT_QUIRK_CONTACT_CNT_ACCURATE |
                        MT_QUIRK_WIN8_PTP_BUTTONS,
```

```
        },
        /* SiSdrv Start */
        { .name = MT_CLS_SIS,
                .quirks = MT_QUIRK_NOT_SEEN_MEANS_UP |
                        MT_QUIRK_CONTACT_CNT_ACCURATE
        },
        /* SiSdrv End */
        { }
};
```

2.1.3.3. Start function

Copy high-lighted code listed below to function "static int mt_probe()". (Before "hid_hw_start")

```
static int mt_probe(struct hid_device *hdev, const struct hid_device_id *id)
…
/* SiSdrv Start */
if (hdev->vendor == USB_VENDOR_ID_SIS_TOUCH) {
        hdev->quirks |= HID_QUIRK_NOGET;
        printk(KERN_INFO "sis:sis-probe: quirk = %x\n", hdev->quirks);
#ifdef CONFIG_HID_SIS_CTRL
        ret = sis_setup_chardev(hdev);
        if (ret)
                printk( KERN_INFO "sis_setup_chardev fail\n");
#endif    //CONFIG_HID_SIS_CTRL
}
/* SiSdrv End */
ret = hid_hw_start(hdev, HID_CONNECT_DEFAULT);
if (ret)
        return ret;
```

2.1.3.4. Finish Function

Copy high-lighted code listed below to function “static void mt_remove()”. (Before “hid_hw_stop”)

```
static void mt_remove(struct hid_device *hdev)
…
/* SiSdrv Start */
if (hdev->vendor == USB_VENDOR_ID_SIS_TOUCH) {
        sis_deinit_chardev(hdev);
}
/* SiSdrv End */
hid_hw_stop(hdev);
```

2.1.3.5. Device list

Copy high-lighted code listed below to structure “hid_device_id mt_devices[]”. (Before “Google MT devices”).

```
static const struct hid_device_id mt_devices[] = {
…
/* SiSdrv Start */
{ .driver_data = MT_CLS_SIS,
    HID_DEVICE(HID_BUS_ANY, HID_GROUP_ANY, USB_VENDOR_ID_SIS_TOUCH,
                HID_ANY_ID) },
/* SiSdrv End */

/* Google MT devices */
{ .driver_data = MT_CLS_GOOGLE,
        HID_DEVICE(HID_BUS_ANY, HID_GROUP_ANY, USB_VENDOR_ID_GOOGLE,
                USB_DEVICE_ID_GOOGLE_TOUCH_ROSE) },
/* Generic MT device */
{HID_DEVICE(HID_BUS_ANY,        HID_GROUP_MULTITOUCH,        HID_ANY_ID,
HID_ANY_ID) },
/* Generic Win 8 certified MT device */
{    .driver_data = MT_CLS_WIN_8,
        HID_DEVICE(HID_BUS_ANY, HID_GROUP_MULTITOUCH_WIN_8,
                HID_ANY_ID, HID_ANY_ID) },
{ }
```

2.1.4. Modify Kconfig

Copy high-lighted code listed below and paste it into (kernel_src)/drivers/hid/**Kconfig**. (After "config HID_MULTITOUCH") Please do not copy and paste whole file directly.

```
config HID_MULTITOUCH
        tristate "HID Multitouch panels"
        depends on HID
        ---help---
          Generic support for HID multitouch panels.
…
#/////////////////     SiSdrv Start  /////////////////

config HID_SIS_CTRL
        tristate "SiS Touch Device Controller"
        depends on HID_MULTITOUCH
        default y
        help
        Support for SiS Touch devices update FW.

menu "SiS touchscreen series"
choice
        depends on USB_HID
        depends on HID_SIS_CTRL
        prompt "SiS controller select"

config HID_SIS95XX_SIS98XX
        depends on HID_SIS_CTRL
        tristate "SiS 95xx and 98xx series Touch Device"
        help
        Support for SiS Touch devices that are fully compliant with HID standard.

config HID_SIS92XX
        depends on HID_SIS_CTRL
        tristate "SiS 92xx series Touch Device"
        help
```

```
        Support for SiS Touch devices that are fully compliant with HID standard.

endchoice

config DEBUG_HID_SIS_UPDATE_FW
        bool "SiS Touch device update firmware support debug message enable"
        depends on HID_SIS_CTRL
        default n
        help
          Say Y here if you want to enable debug message of
          firmware updating for SiS Touch
              devices.

endmenu
#////////////////     SiSdrv End    ////////////////
```


2.1.5. Modify Makefile

Copy scripts listed below and paste it into (kernel_src)/drivers/hid/**Makefile.**

Please do not copy and paste file directly.


```
…
#////////////////     SiSdrv Start  ////////////////
obj-$(CONFIG_HID_SIS_CTRL)  += hid-sis_ctrl.o
#////////////////     SiSdrv End    ////////////////

obj-$(CONFIG_HID_A4TECH)    += hid-a4tech.o
obj-$(CONFIG_HID_ACCUTOUCH)  += hid-accutouch.o
```

2.1.6. Device Tree Source (DTS)

For DTS (Device Tree Source), we need to describe our hardware information. Here is an example on raspberry pi 3 and .dts is located in **kernel/arch/arm/boot/dts/bcm2710-rpi-3-b.dts**.

Each user should configure GPIO pin for interrupt function first. For example, we use GPIO pin number 18 as our GPIO interrupt pin (18 IRQ_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 18.

```
&i2c1 {
        pinctrl-names = "default";
        pinctrl-0 = <&i2c1_pins>;
        clock-frequency = <100000>;

      /* SiSdrv Start */
       sis_touchscreen@5c {
            compatible = "hid-over-i2c";
            reg = <0x5c>;
            hid-descr-addr = <0x0000>;
            interrupt-parent = <&gpio>;
            interrupts = <18 IRQ_TYPE_LEVEL_LOW>;
            status = "okay";
       };
      /* SiSdrv End */
};
```

2.2. Kernel configures

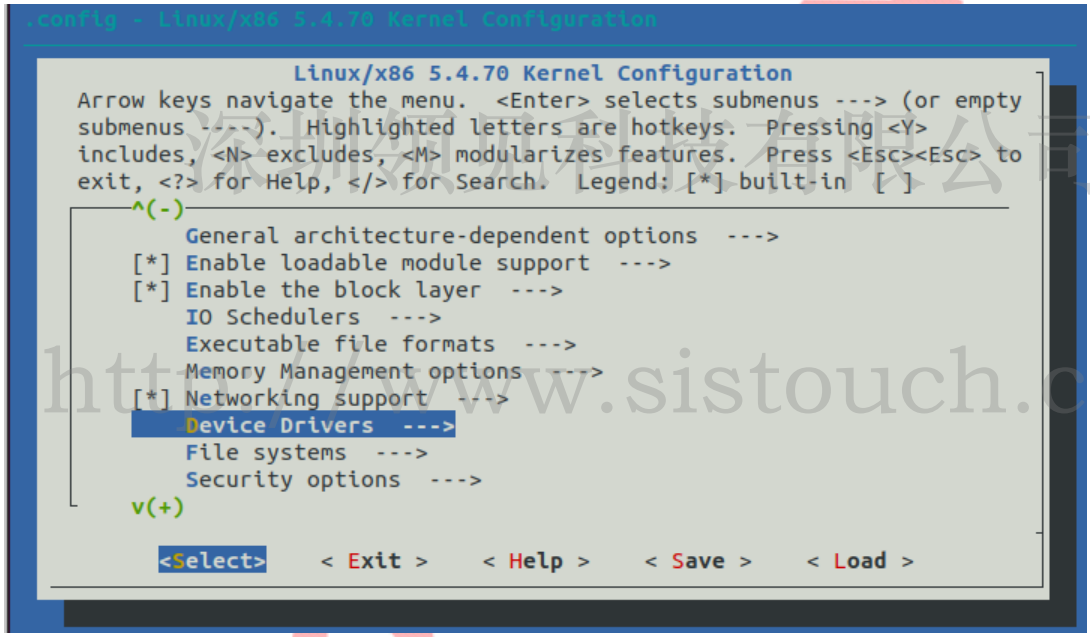
2.2.1. Setting configure

Set the configure to building kernel.

And type in command “make menuconfig” to configure kernel, and then include SiS driver with the procedure in the following subsection.

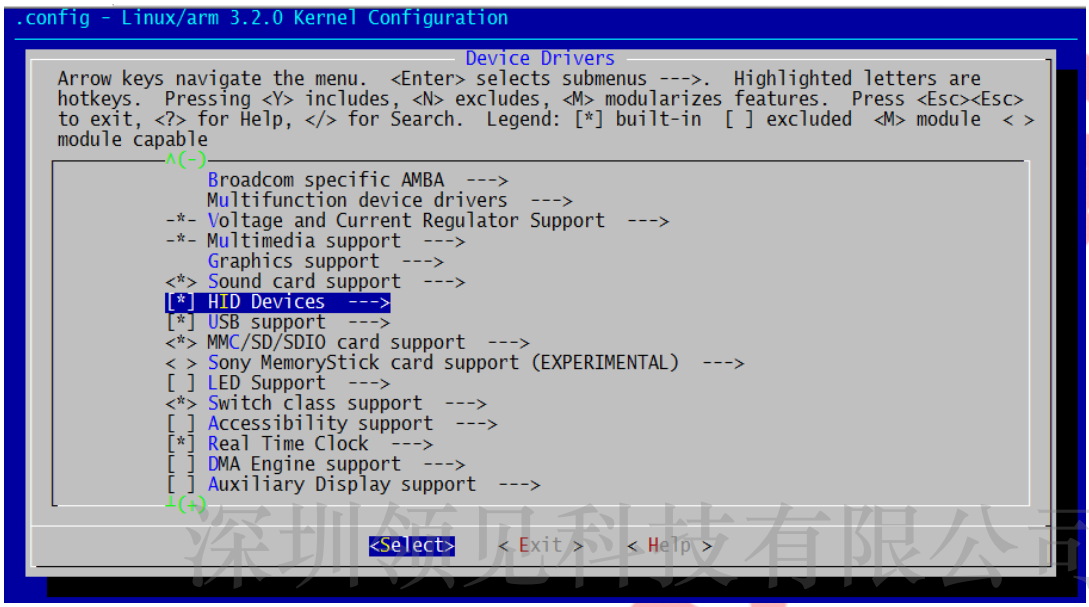
```
cp /boot/config-$(uname -r) .config
make menuconfig
```

2.2.2. Select “Device Drivers”

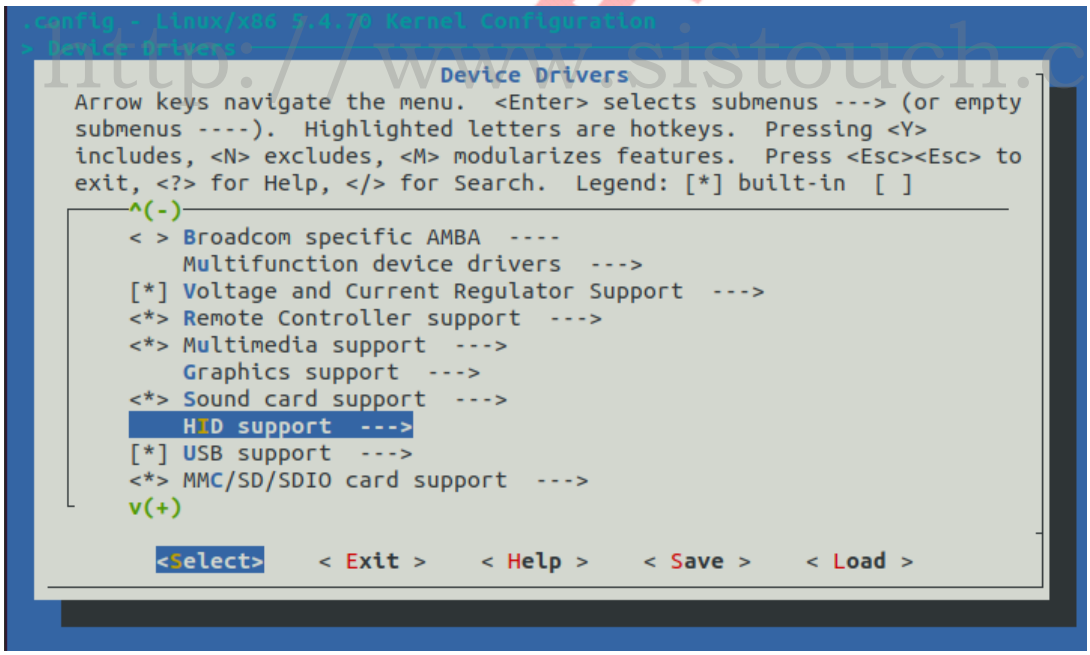


2.2.3. Include and select HID Devices (HID support)

If kernel <3.9, needs to include “HID Devices”

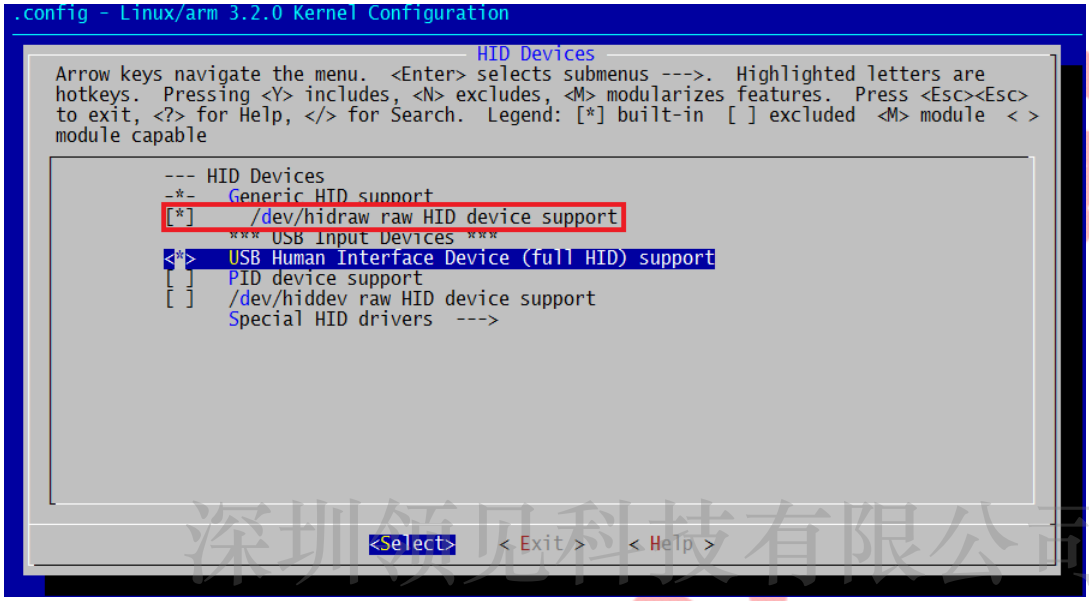


If kernel >= 3.10, select “HID support”

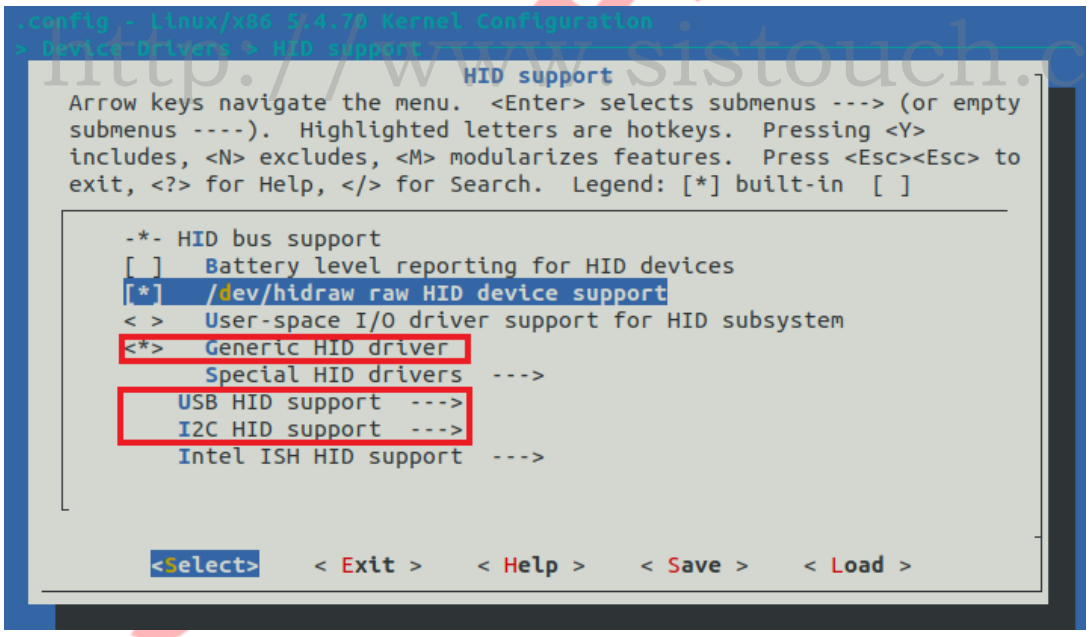


2.2.4. Include USB Human Interface Device (full HID) support

If kernel version<3.9, Please also choose “/dev/hidraw raw HID device support”

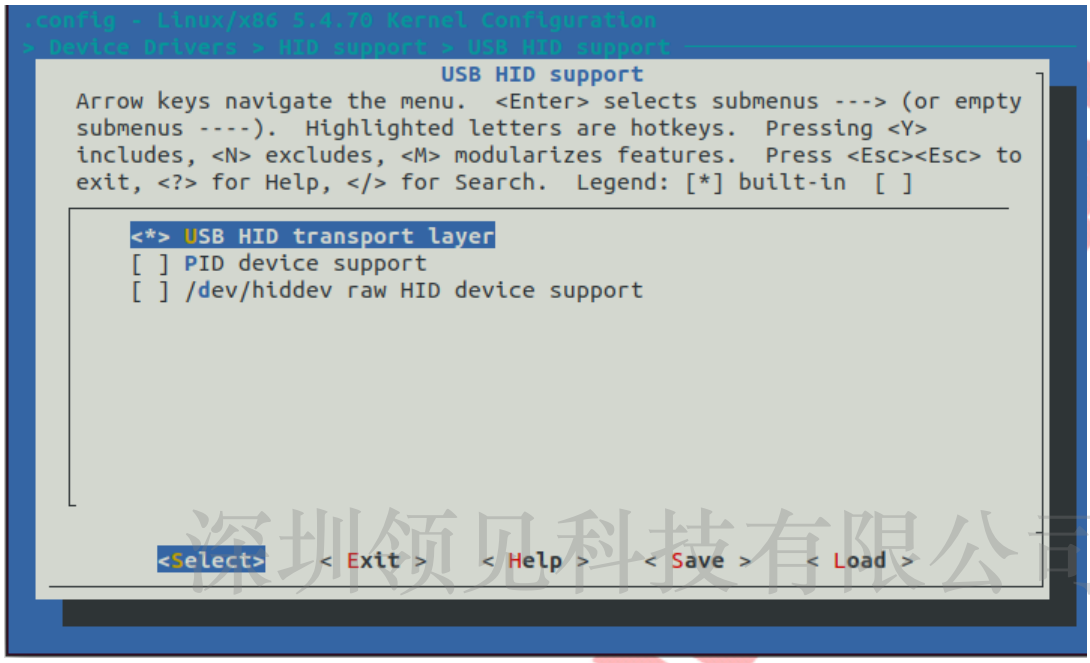


If kernel >= 3.10, Please also choose “/dev/hidraw raw HID device support” and “Generic HID driver”. Then select “USB HID support” and “I2C HID support”.

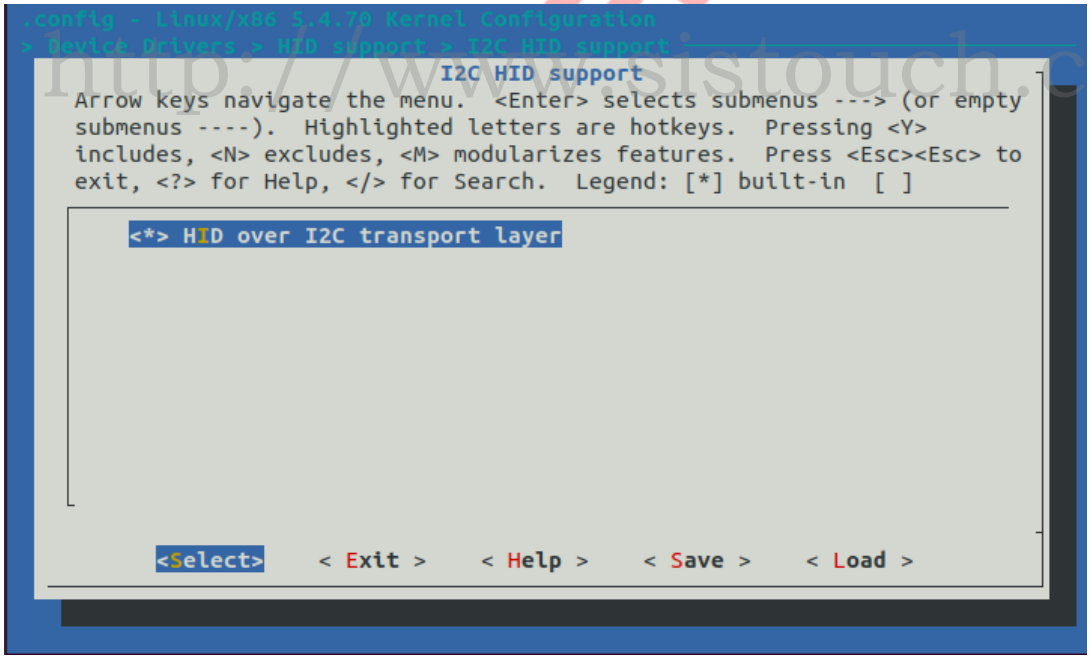


2.2.5. USB HID support and I2C HID support

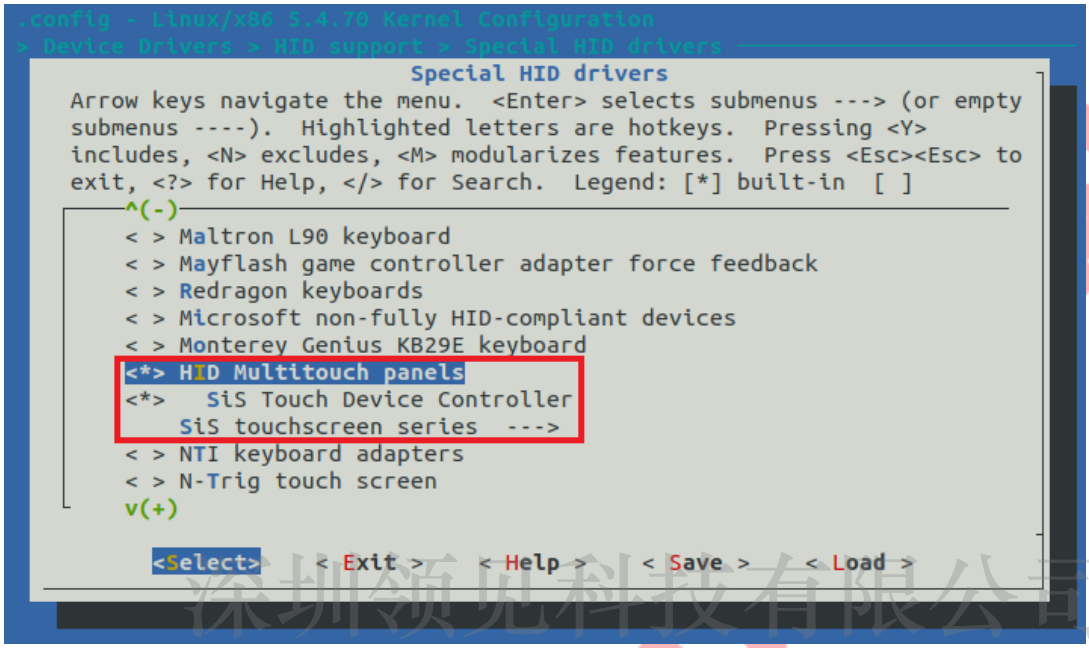
Choose “USB HID transport layer” in USB HID support.



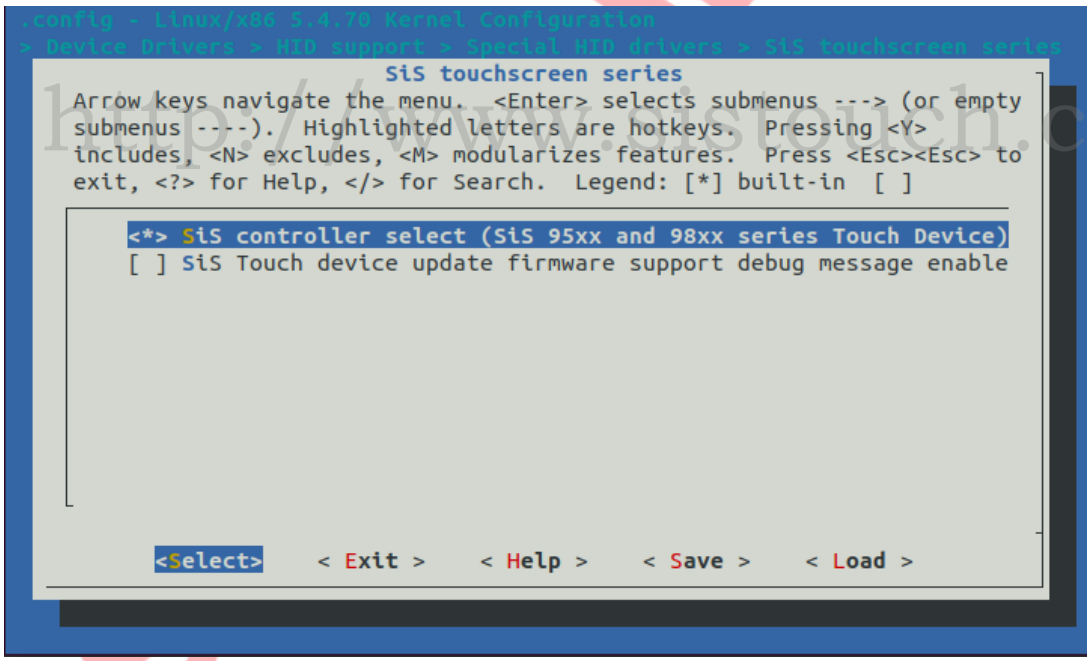
Choose “HID over I2C transport layer”.



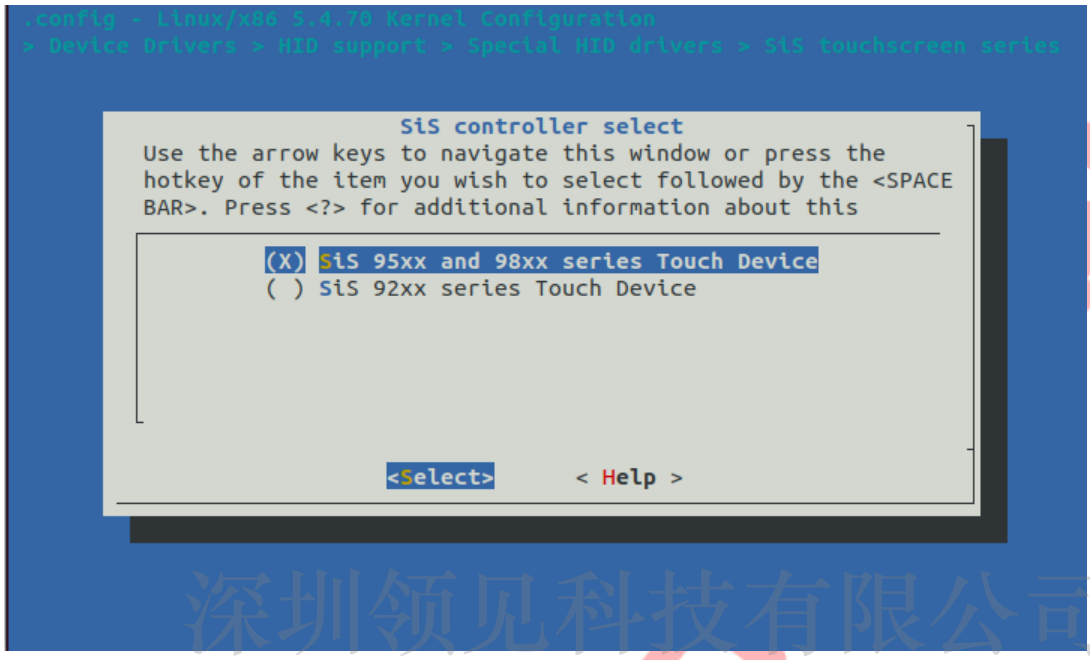
2.2.6. Select HID Multitouch panels <*> and SiS Touch Device Controller <*>



In SiS touchscreen series:



Select type of SiS controller:



2.3. Build kernel image

3. Touch driver test

After the touch driver is installed and built, there are some steps below to confirm them.

3.1. Kernel message (dmesg)

Type command “dmesg > log.txt” to record full kernel message.

```
dmesg > log.txt
```

Type command “dmesg | grep sis” to check sis driver loading.

```
sis@sis-desktop:~$ sudo dmesg | grep sis
[    0.000000] Linux version 5.11.0-1007-raspi (root@sis-X580VD) (aarch64-linux-
gnu-gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0, GNU ld (GNU Binutils for Ubuntu) 2
.34) #7 SMP PREEMPT Tue Jan 4 15:47:41 CST 2022 (Ubuntu 5.11.0-1007.7-raspi 5.11
.12)
[    3.585428] sis:sis-probe: quirk = 80000848
[    3.585450] sis_setup_chardev.
[    3.585467] sis_hydra_hid_touch_device driver(major 236) installed.
[    8.080140] systemd[1]: Hostname set to <sis-desktop>.
[    9.170979] systemd[1]: Condition check resulted in Platform Persistent Stora
ge Archival being skipped.
```

3.2. Char device node

Type command “ls -al /dev/sis*” to check device node and permission.

```
ls -al /dev/sis*
```

If device node exists, you will find the messages below. (for SiS95xx)

```
/dev/sis_hydra_hid_touch_device
```

Or (for SiS92xx)

```
/dev/sis_aegis_hid_touch_device
```

3.3. USB touch device handlers

Type command “cat /proc/bus/input/devices” and find the messages below.

```
cat /proc/bus/input/devices
```

```
sis@sis-desktop:~$ cat /proc/bus/input/devices
I: Bus=0018 Vendor=0457 Product=1905 Version=0100
N: Name="hid-over-i2c 0457:1905"
P: Phys=1-005c
S: Sysfs=/devices/platform/soc/fe804000.i2c/i2c-1/1-005c/0018:0457:1905.0001/inp
ut/input0
U: Uniq=
H: Handlers=mouse0 event0
B: PROP=2
B: EV=1b
B: KEY=400 0 0 0 0 0
B: ABS=673800001000003
B: MSC=20
```