SiS925x USB Touch Driver Porting Guide
For Ubuntu 14.04 LTS

Rev. 1.0

May. 22, 2015

Contents

1. Install Requirement

- Ubuntu 14.04 LTS
- Ubuntu 14.04 Kernel Source (above kernel 3.5)

2. Steps of Driver Installment

2.1. Get Ubuntu 14.04 Kernel Source

2.1.1. Get root privilege

For building kernel conveniently, please get the root privilege. For user who is familiar with Linux operations, we recommend using “sudo” command to replace root privilege.

2.1.2. Update packages

```
cd /usr/src
apt-get update
```

2.1.3. Install essential packages for make Linux kernel

```
apt-get install libncurses5-dev build-essential zip u-boot-tools sharutils kernel-package
```

2.1.4. Get Linux kernel source code

```
apt-get install linux-source
```

The file “linux-source-3.13.x.tar.bz2” will download to /usr/src.

2.1.5. Extract the kernel source

```
tar jxvf linux-source-<version>.tar.bz2 -C /usr/src
```

<version> depends on different computer, for example : 3.13.0

2.2. Kernel source patching

2.2.1. Add in the start functions in driver

1. Add in "Include File", "setup-function", "remove-function" and "id-table" into kernel/drivers/hid/hid-multitouch.c.

   (a) Include File

```
/*
*  HID driver for multitouch panels
*
*  Copyright (c) 2010-2012 Stephane Chatty <chatty@enac.fr>
…
#include <linux/device.h>
…
#include <linux/string.h>
#include "hid-sis_ctrl.h"
MODULE_AUTHOR("Stephane Chatty <chatty@enac.fr>");
…
```

   (b) "setup-function" in `mt_probe()` function

```
static int mt_probe(struct hid_device *hdev, const struct hid_device_id
*id)
{
    int ret, i;
…

    ret = hid_parse(hdev);
    if (ret != 0)
        return ret;
//SiS set noget for not init reports
    if (hdev->vendor == USB_VENDOR_ID_SIS_TOUCH)
    {
        hdev->quirks |= HID_QUIRK_NOGET;
        printk(KERN_INFO "sis:sis-probe: quirk = %x\n", hdev->quirks);

        //SiS FW update
#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
    }
    ret = hid_hw_start(hdev, HID_CONNECT_DEFAULT);
    if (ret)
        return ret;
…
}
```

(c) "remove-function" in `mt_remove()` function

```
static void mt_remove(struct hid_device *hdev)
{
    struct mt_device *td = hid_get_drvdata(hdev);

    //SiS FW update
    #ifdef CONFIG_HID_SIS_CTRL
    if (hdev->vendor == USB_VENDOR_ID_SIS_TOUCH)
    {
        sis_deinit_chardev();
    }
    #endif    //CONFIG_HID_SIS_CTRL

    hid_hw_stop(hdev);
    kfree(td->slots);
    kfree(td);
    hid_set_drvdata(hdev, NULL);
}
```

(d) "id-table" in `mt_devices[]` array

Attenion:

Please skip this step if kernel version is more than 3.5.

```
    static const struct hid_device_id mt_devices[] = {
    /* 3M panels */
```

```
    { .driver_data = MT_CLS_3M,
        HID_USB_DEVICE(USB_VENDOR_ID_3M,
            USB_DEVICE_ID_3M1968) },
    …
    /* XAT */
    { .driver_data = MT_CLS_DEFAULT,
        HID_USB_DEVICE(USB_VENDOR_ID_XAT,
            USB_DEVICE_ID_XAT_CSR) },
    { .driver_data = MT_CLS_DEFAULT,
        HID_USB_DEVICE(USB_VENDOR_ID_SIS_TOUCH,
        USB_DEVICE_ID_SIS817_TOUCH) },
    { .driver_data = MT_CLS_DEFAULT,
        HID_USB_DEVICE(USB_VENDOR_ID_SIS_TOUCH,
        USB_DEVICE_ID_SISF817_TOUCH) },
…
};
```

2.2.2. Copy Update Firmware Add-On file

Copy "hid-sis_ctrl.c" and "hid-sis_ctrl.h" into kernel/drivers/hid/ .

2.2.3. Add in SiS USB Touch Controller PID &VID

1. User can modify 0x0817 behind USB_DEVICE_ID_SIS817_TOUCH to match your device's PID. On the other side, user can create a new PID definition for your device such as the bold string.
2. Modify ID-Table header file

Attenion:

Please check your kernel version and choose the right step,

A. If your kernel version is before or equal to 3.12

Copy scripts listed below and paste it into the bottom of kernel/drivers/hid/hid-ids.h (before #endif).

```
#define USB_VENDOR_ID_SIS_TOUCH                0x0457
```

```
#define USB_DEVICE_ID_SIS817_TOUCH          0x0817
#define USB_DEVICE_ID_SISF817_TOUCH         0xF817
```

B. If your kernel version is equal to 3.13

Copy scripts listed below and paste it into the bottom of kernel/drivers/hid/hid-ids.h (before #endif).

```
#define USB_VENDOR_ID_SIS_TOUCH             0x0457
#define USB_DEVICE_ID_SISF817_TOUCH         0xF817
```

C. If your kernel version is more than 3.13

Copy scripts listed below and paste it into the bottom of kernel/drivers/hid/hid-ids.h (before #endif).

```
#define USB_DEVICE_ID_SISF817_TOUCH         0xF817
```

2.2.4. Add into Kconfig

Copy scripts listed below and paste it into the bottom of kernel/drivers/hid/Kconfig (after config HID_MULTITOUCH).

```
config HID_MULTITOUCH
    tristate "HID Multitouch panels"
…
To compile this driver as a module, choose M here: the
      module will be called hid-multitouch.
config HID_SIS_CTRL
    tristate "SiS Touch Device Controller"
    depends on HID_MULTITOUCH
    ---help---
    Support for SiS Touch devices update FW.
```

2.2.5. Add in Makefile

Copy scripts listed below and paste it into kernel/drivers/hid/Makefile (before obj-$(CONFIG_USB_HID) += usbhid/).

```
obj-$(CONFIG_HID_SIS_CTRL) += hid-sis_ctrl.o
```

2.2.6. Modify the authority of driver for update-FW

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

2.2.7. Add device for hid-core.c

Add "id-table" in `hid_have_special_driver[]` array of kernel/drivers/hid/ hid-core.c.

```
static const struct hid_device_id hid_have_special_driver[] = {
    { HID_USB_DEVICE(USB_VENDOR_ID_A4TECH, USB_DEVICE_ID_A4TECH_WCP32PU) }
…
    { HID_BLUETOOTH_DEVICE(USB_VENDOR_ID_NINTENDO,
USB_DEVICE_ID_NINTENDO_WIIMOTE2) },
    { HID_USB_DEVICE(USB_VENDOR_ID_SIS_TOUCH,
USB_DEVICE_ID_SIS817_TOUCH) },
    { HID_USB_DEVICE(USB_VENDOR_ID_SIS_TOUCH,
USB_DEVICE_ID_SISF817_TOUCH) },
    { }
};
```

2.3. Kernel configure

2.3.1. Setting configure

Set the configure to building kernel.(assume that recent directory is /usr/src/linux-source-3.2.0)

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

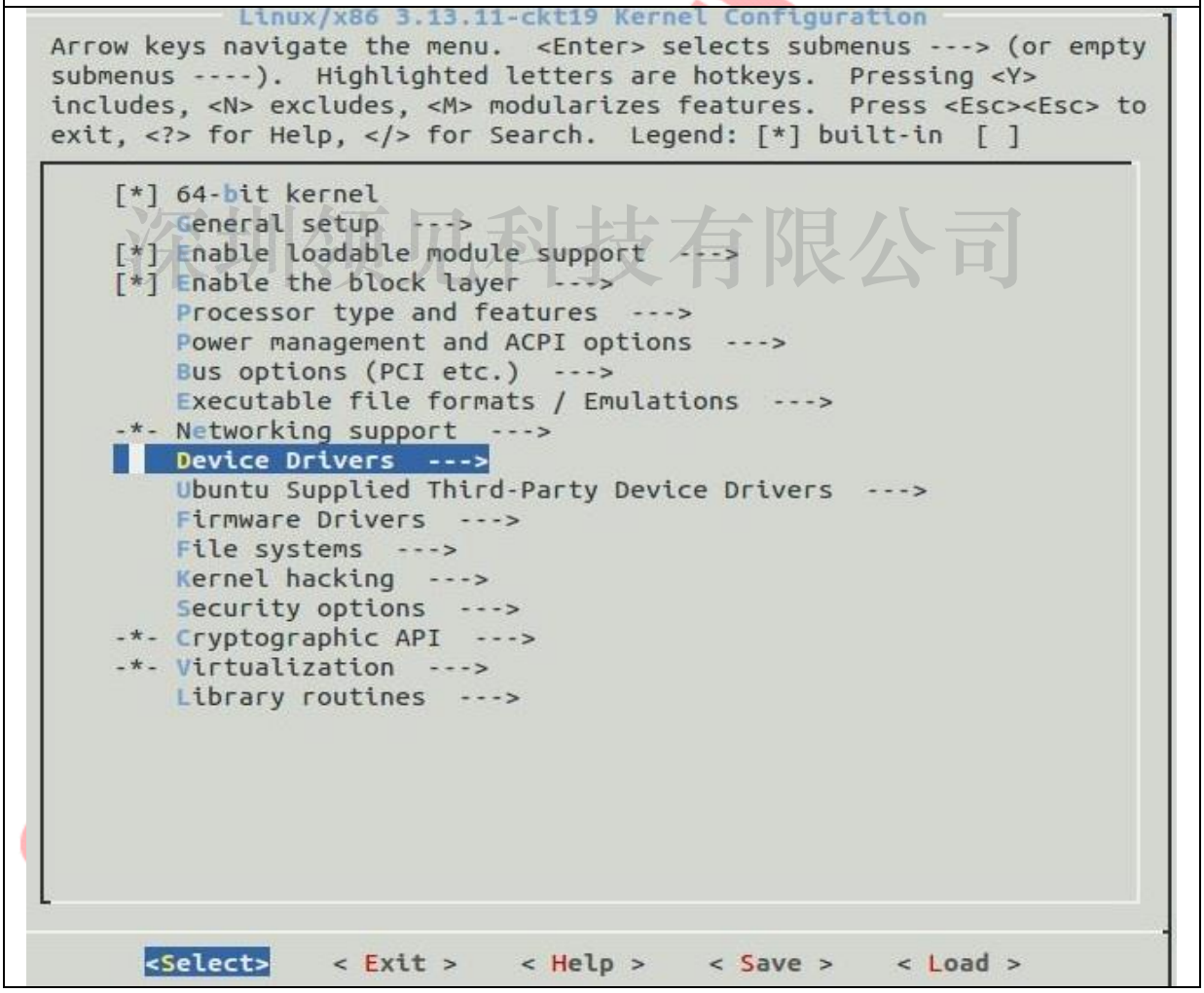
2.3.2. Select the driver for SiS Touch Controller FW update

Select SiS FW update module. (assume that recent directory is /usr/src/linux-source-3.2.0)

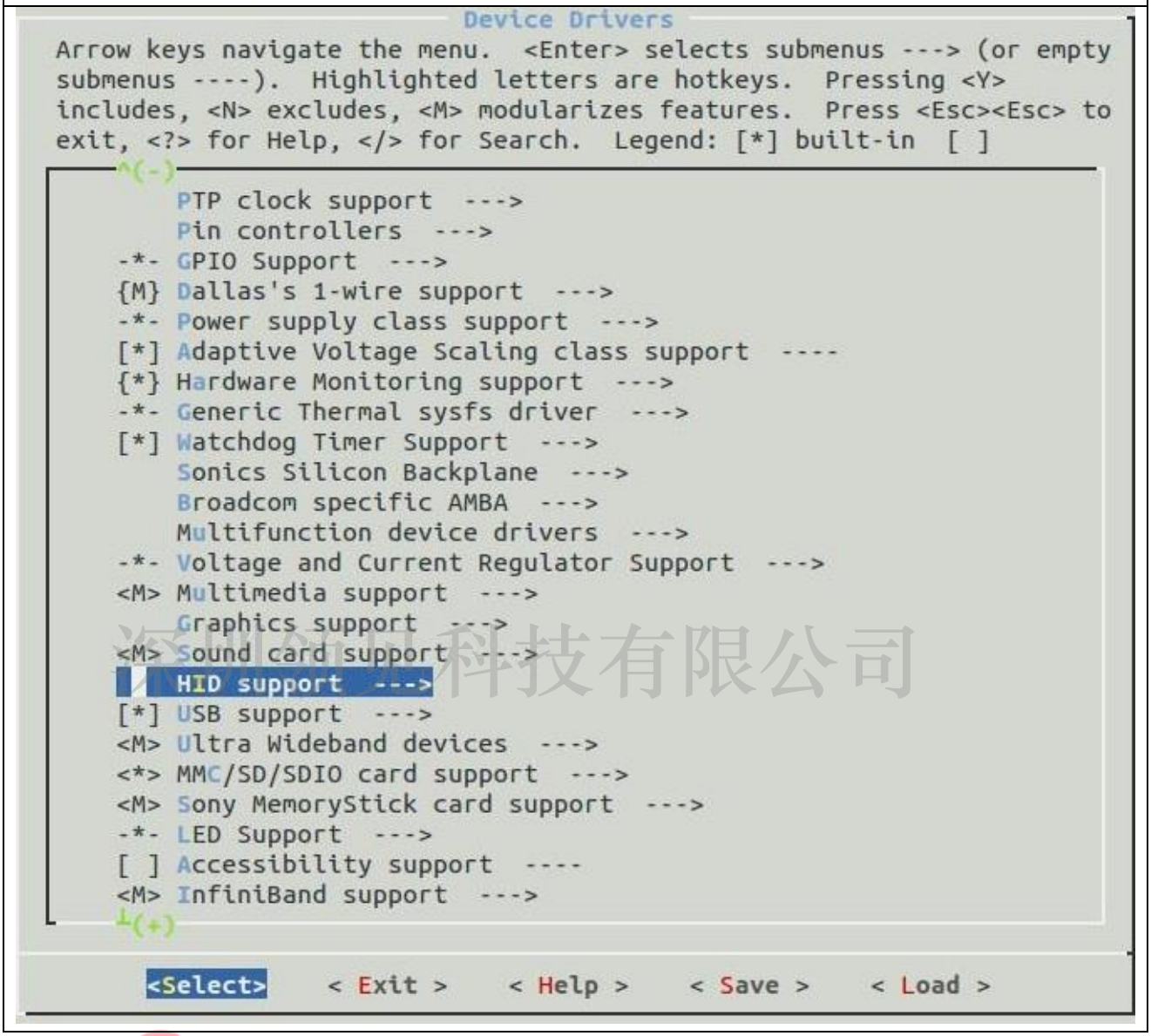
```
make menuconfig
```

Select path :

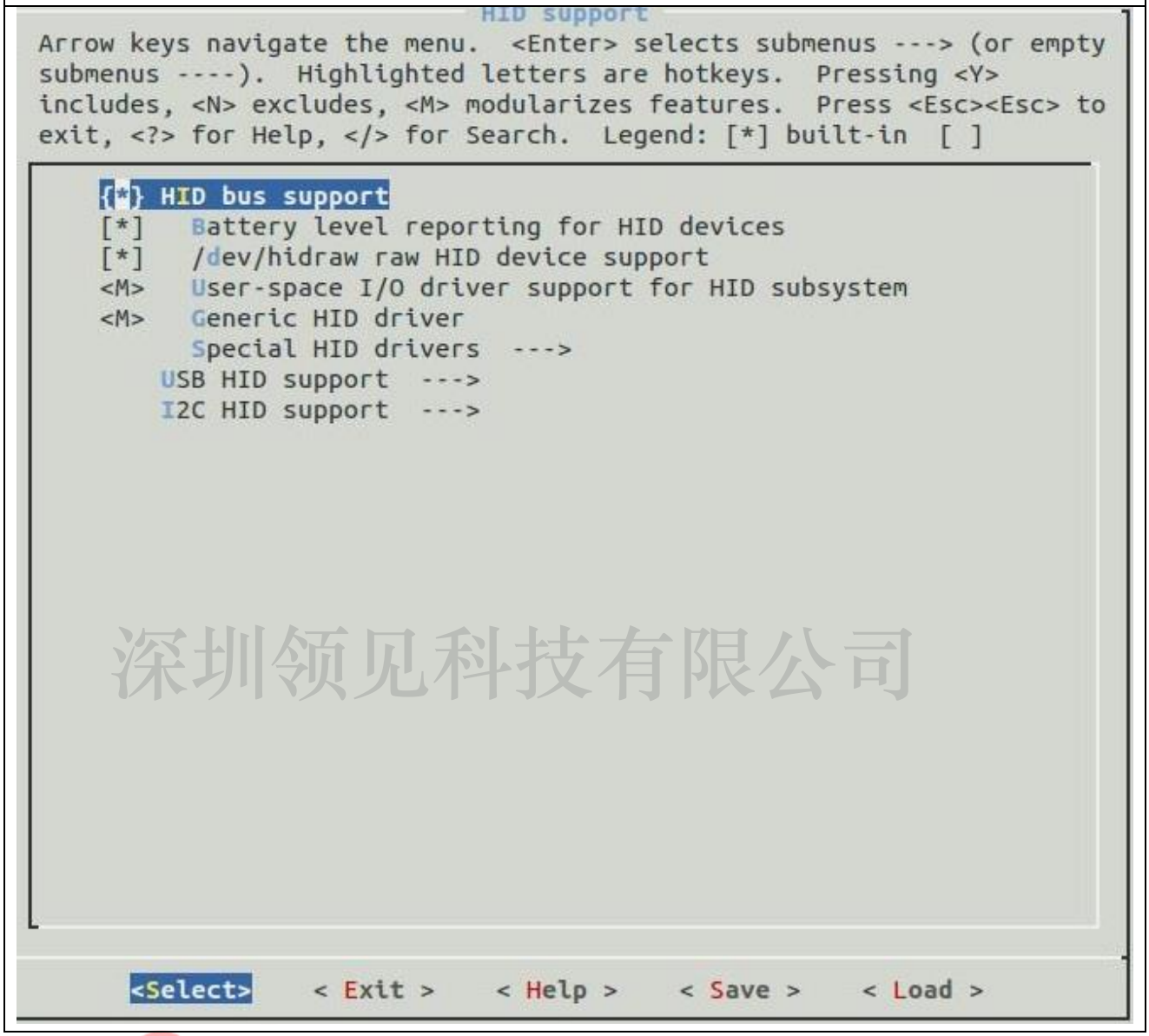
```
Device Drivers →
```



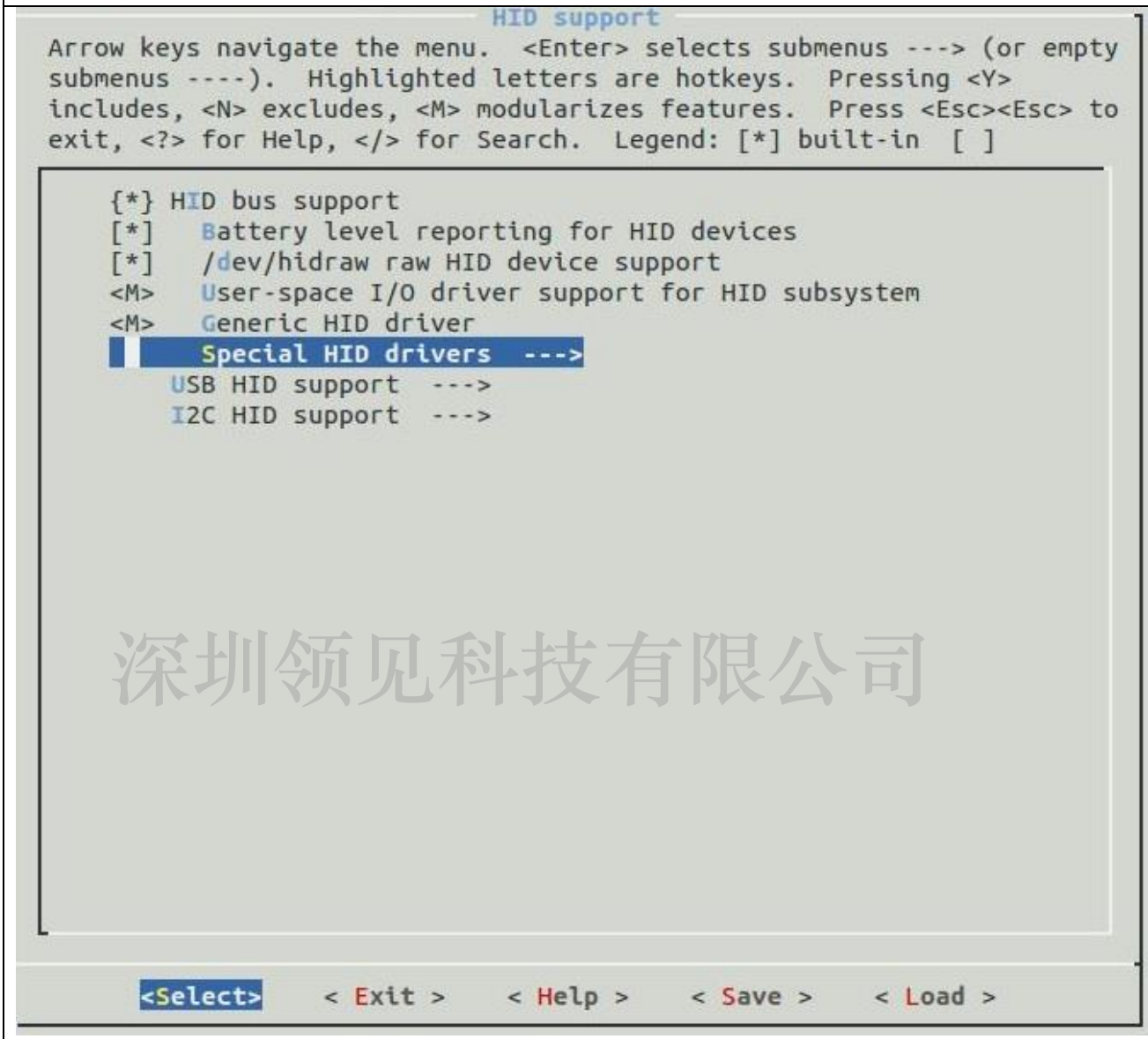
HID support →



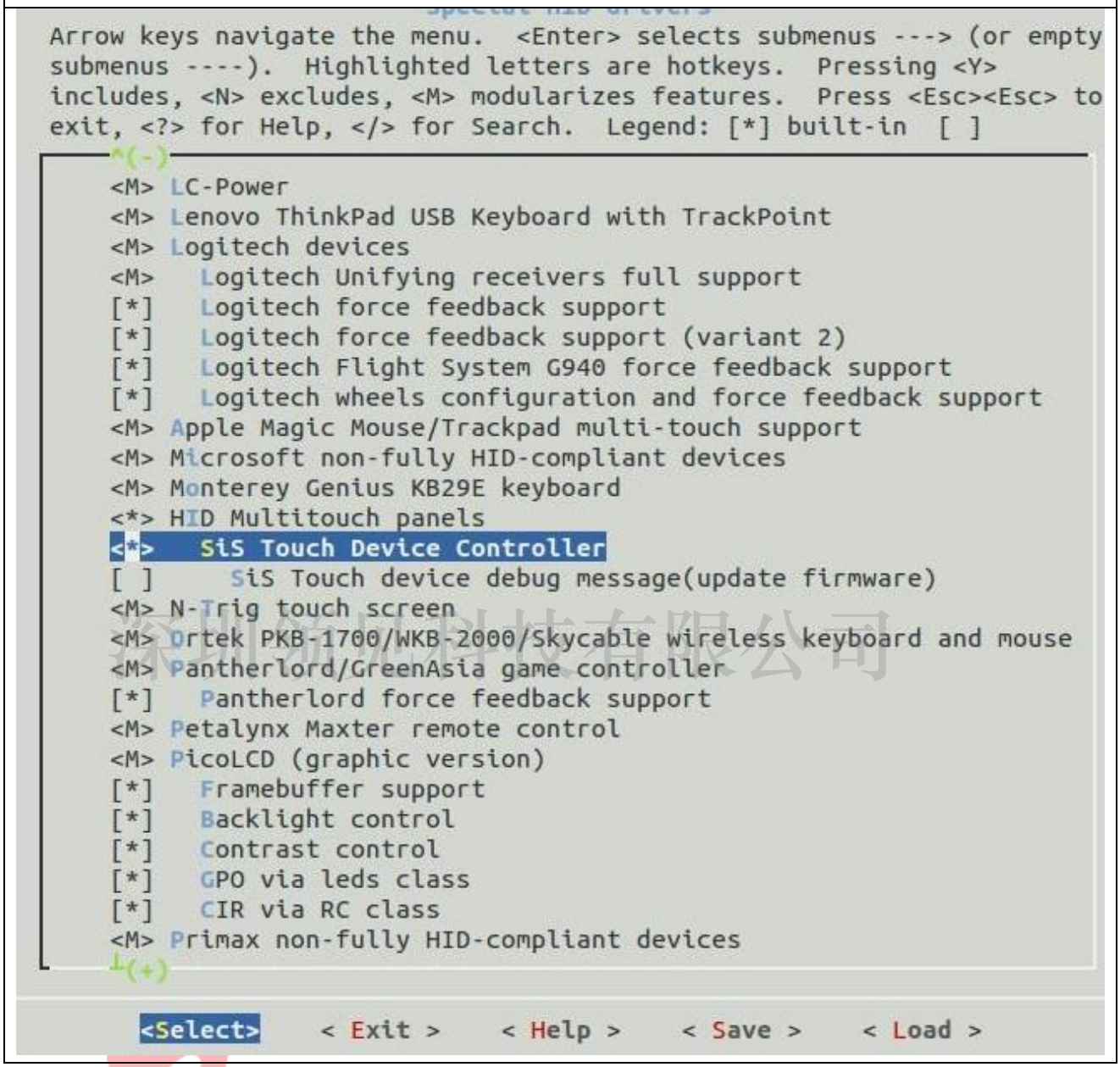
[*] HID bus support



Special HID drivers →



<*> HID Multitouch panels → <*> SiS Touch Device Controller



2.4. Debug message setting (optional)

Debug message does not necessary in SiS touch driver, but it is very useful when there is something wrong on your touch device. User should decide whether add these debug messages or not. If user does not use debug message, please skip this subsection.

2.4.1. Set kconfig file for kernel configuration

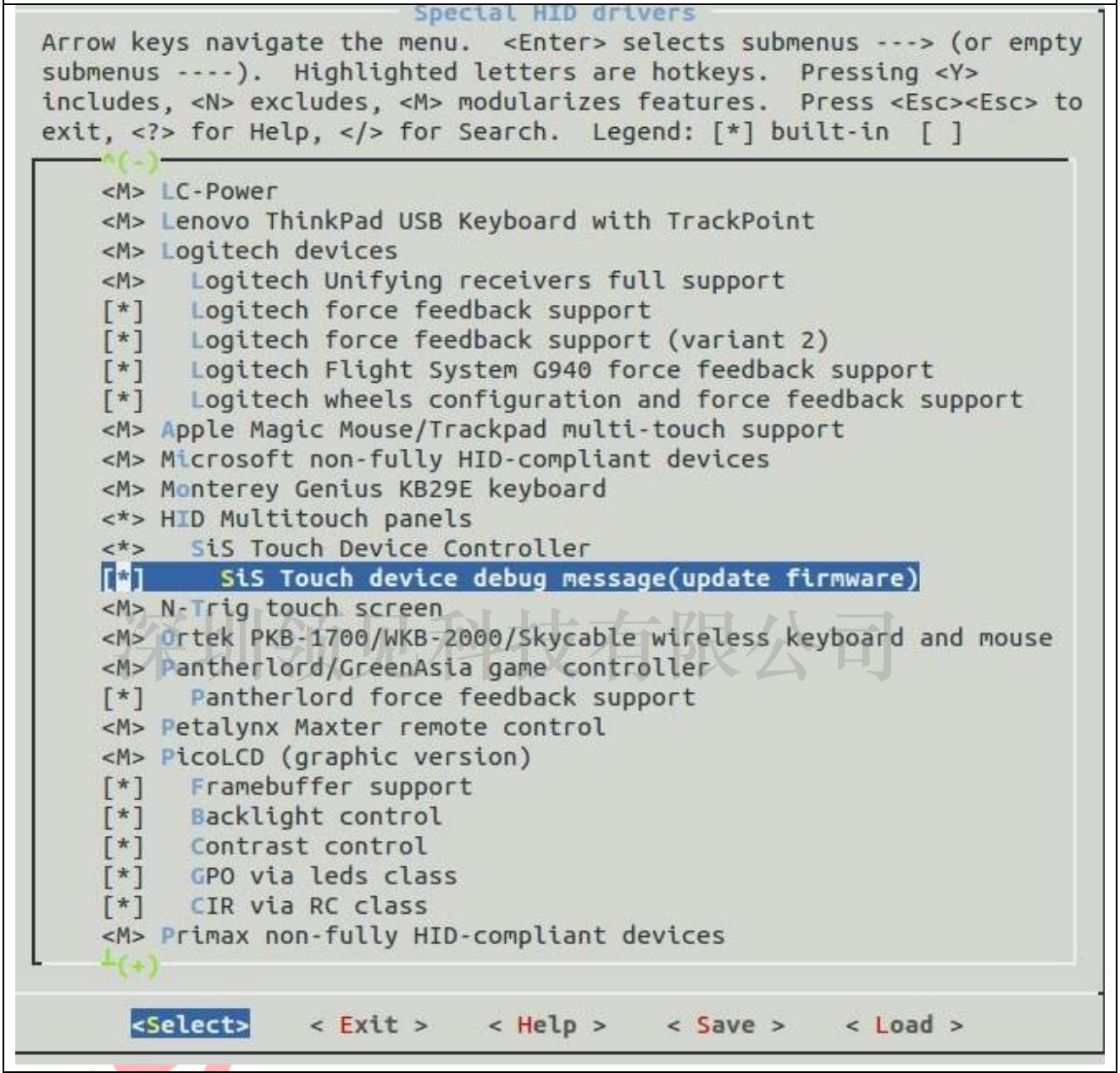
Copy scripts listed below and paste it into the bottom of kernel/drivers/hid/Kconfig (after config HID_MULTITOUCH).

```
config DEBUG_HID_SIS_UPDATE_FW
    bool "SiS Touch device debug message(update firmware)"
    depends on HID_SIS_CTRL
    default n
    ---help---
    Say Y here if you want to enable debug message(update firmware) for SiS
    Touch devices. Must enable config HID_SIS_UPDATE_FW first.
```


2.4.2. Include debug message of kernel configuration

The preceding flow is the same with SiS touch driver described in section 2.3.2. At the end,User includes the **SiS Touch device debug message(update firmware)** item to enable what the debug message you want to show.

<*> SiS Touch device debug message(update firmware)



2.5. Build kernel

2.5.1. Build kernel, install kernel image & update grub menu, reboot

```
make bzImage –j8
```

Build module (Only do one time if you have not modify modules)

```
make modules –j8
make modules_install
```

```
make install
reboot
```

Update the firmware of SiS touch Controller can be operated by console mode after reboot with the built kernel.

3. Touch driver test

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

3.1. lsusb

Type command “lsusb” check the VID/PID (Here PID:10b1 is our test touch panel)

```
Bus 005 Device 002: ID 04f2:b084 Chicony Electronics Co., Ltd
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 002: ID 0a5c:2101 Broadcom Corp. BCM2045 Bluetooth
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 002: ID 046d:c058 Logitech, Inc. M115 Mouse
Bus 002 Device 001: ID VID   PID  Linux Foundation 1.1 root hub
Bus 001 Device 002: ID 0457:10b1 Silicon Integrated Systems Corp.
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

3.2. xinput

Type command “xinput” and check if the device is an INPUT device.

```
zerget@zerget-usb:~$ xinput
⎡ Virtual core pointer                          id=2    [master pointer  (3)]
⎜   ↳ Virtual core XTEST pointer                id=4    [slave  pointer  (2)]
⎜   ↳ Logitech Unifying Device. Wireless PID:4007       id=8    [slave  pointer  (2)]
⎜   ↳ USBest Technology SiS HID Touch Controller        id=10   [slave  pointer  (2)]
⎣ Virtual core keyboard                         id=3    [master keyboard (2)]
    ↳ Virtual core XTEST keyboard               id=5    [slave  keyboard (3)]
    ↳ Power Button                              id=6    [slave  keyboard (3)]
    ↳ Power Button                              id=7    [slave  keyboard (3)]
    ↳ AT Translated Set 2 keyboard              id=9    [slave  keyboard (3)]
zerget@zerget-usb:~$
```

3.3. USB touch device handlers

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

```
I: Bus=0003 Vendor=0457 Product=0817 Version=0111
N: Name="USBest Technology SiS HID Touch Controller"
P: Phys=usb-ehci-omap.0-1.2/input0
S:      Sysfs=/devices/platform/usbhs-omap.0/ehci-omap.0/usb1/1-1/1-1.2/1-1.2:1.0
    /input/input1
```

```
U: Uniq=
H: Handlers=mouse0 event1
B: PROP=2
B: EV=b
B: KEY=400 0 0 0 0 0 0 0 0 0 0
B: ABS=2608000 3
```

3.4. Char device node

Type command “ls /dev/sis*” to check device node exist or not.
If device node exists, you will find the messages below.

```
/dev/sis_aegis_hid_touch_device
```