SiS USB Touch Driver Porting Guide for sis925x (Android Systems)

Rev. 0.30

Mar. 21, 2014

Revision History

Date	Rev.	Description
Dec. 20, 2011	0.10	1. Initial Release
Jun. 15, 2012	0.20	1. Modify the source code tree structure 2. Modify the access permission from Android 3. Modify issue of Android 4.0
Aug. 01, 2012	0.21	1. Add new configuration
Mar. 21, 2014	0.30	1. Modify the source code tree structure 2. Modify compiling and enclosing driver in kernel

Contents

1. Introduction

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

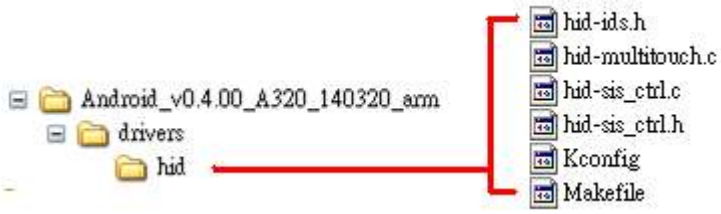
We provide a simple method to integrate and build SiS touch driver within different version of Linux kernel. Additionally, SiS USB touch driver also uses system calls to requests a service from Linux kernel. How to define system calls will be will be described in the chapter, “Patching the Kernel with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

This document contains three parts: source code overview, modifying main program for different operating systems, patching the kernel with SIS USB 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.

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 "<driver source dir>/kernel/drivers/hid".

Use this Makefile to compile SiS USB touch driver. Our SiS USB touch driver can be either enclosed in Linux kernel or compiled independently as a kernel module. Its usage will be described in the chapter, "Compiling and Enclosing Driver in Kernel".

2.2.2. Kconfig

Kconfig is located in "<driver source dir>/kernel/drivers/hid".

This file is used only when enclosing SiS USB touch driver in Linux kernel. Its usage will be described in the chapter, "Compiling and Enclosing Driver in Kernel".

2.2.3. USB Touch Driver

Touch drivers are located in "<driver source dir>/kernel/drivers/hid".

The main program is "hid-multitouch.c". To compile main program for different operating systems will be described in "Modifying main program for different requirements of environment".

2.2.4. ID-Table header file

ID-Table header file is located in "<driver source dir>/kernel/drivers/hid".

The file name is "hid-ids.h". To compile main program for different operating systems will be described in "Modifying main program for different requirements of environment".

2.2.5. Update Firmware Add-On file

Update firmware add-on files are located in "<driver source dir>/kernel/drivers/hid".

The file name is "hid-sis_ctrl.c" and "hid-sis_ctrl.h". To compile main program for different operating systems will be described in "Modifying main program for different requirements of environment".

2.2.6. Set SiS USB device node permission

Applications/Tools use character device driver to requests a service to Linux kernel. The main program is in "hid-sis.c", but we have to set device node permission in "ueventd.rc". How to set the permission will be described in the chapter 3, "Modifying main program for different requirements of environment".

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

- implementation exists
- permission of SiS USB 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/Vendor_XXXX_Product_YYYY.idc

XXXX is vendor ID (VID), YYYY is product ID (PID), depends on different device.

Please check your device's VID and PID number.

3.2. Set SiS USB 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. /media/rootfs/ueventd.rc

Define device name "sis_aegis_hid_touch_device", 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 "hid-sis.c".

```
# sysfs properties
/sys/devices/virtual/input/input*          enable        0660    root      input
/sys/devices/virtual/input/input*          poll_delay    0660    root      input
/sys/devices/virtual/usb_composite/*       enable        0664    root      system

#sis device
/dev/sis_aegis_hid_touch_device                          0666    root      root
/dev/sis_aegis_hid_bridge_touch_device                   0666    root      root
```

3.3. Compiling and Enclosing Driver in Kernel

First copy “hid-sis_ctrl.c” and “hid-sis_ctrl.h” into kernel/drivers/hid/ .

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

Attention:

The value of USB_VENDOR_ID_SIS2_TOUCH and USB_PRODUCT_ID_SIS817_TOUCH depends on your device. Please check your device’s VID and PID number.

```
#define USB_VENDOR_ID_SIS2_TOUCH          0x0457
#define USB_PRODUCT_ID_SIS817_TOUCH       0x0817
#define USB_PRODUCT_ID_SISF817_TOUCH      0xF817
```

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

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

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

```
#////////////////  SIS START  /////////////////
obj-$(CONFIG_HID_SIS_CTRL) += hid-sis_ctrl.o
#////////////////  SIS END  /////////////////
```

Then add in “Include File” and “start-function” into kernel/drivers/hid/hid-multitouch.c.

Part 1: “Include File”

```
#include "hid-sis_ctrl.h"
```

```
22 /*
23  * This program is free software; you can redistribute it and/or modify it
24  * under the terms of the GNU General Public License as published by the Free
25  * Software Foundation; either version 2 of the License, or (at your option)
26  * any later version.
27  */
28
29 #include <linux/device.h>
30 #include <linux/hid.h>
31 #include <linux/module.h>
32 #include <linux/slab.h>
33 #include <linux/usb.h>
34 #include <linux/input/mt.h>
35 #include "usbhid/usbhid.h"
36 #include "hid-sis_ctrl.h"
37
38 MODULE_AUTHOR("Stephane Chatty <chatty@enac.fr>");
39 MODULE_AUTHOR("Benjamin Tissoires <benjamin.tissoires@gmail.com>");
40 MODULE_DESCRIPTION("HID multitouch panels");
41 MODULE_LICENSE("GPL");
42
43 #include "hid-ids.h"
44
45 /* quirks to control the device */
46 #define MT_QUIRK_NOT_SEEN_MEANS_UP  (1 << 0)
47 #define MT_QUIRK_SLOT_IS_CONTACTID  (1 << 1)
```

Part 2: “start-function” in static int mt_probe()

```
//SiS set noget for not init reports
hdev->quirks |= HID_QUIRK_NOGET;
printk(KERN_INFO "sis:sis-probe: quirk = %x\n", hdev->quirks);

//SiS FW update
ret = sis_setup_chardev(hdev);
if(ret)
{
     printk( KERN_INFO "sis_setup_chardev fail\n");
}
```

```
561     ret = hid_parse(hdev);
562     if (ret != 0)
563       goto fail;
564
565     //SiS set noget for not init reports
566     hdev->quirks |= HID_QUIRK_NOGET;
567     printk(KERN_INFO "sis:sis-probe: quirk = %x\n", hdev->quirks);
568
569     //SiS FW update
570     ret = sis_setup_chardev(hdev);
571     if(ret)
572     {
573       printk( KERN_INFO "sis_setup_chardev fail\n");
574     }
575
576     ret = hid_hw_start(hdev, HID_CONNECT_DEFAULT);
577     if (ret)
578       goto fail;
```

Part 3: “finish-function” in static void mt_remove()

```
//SiS FW update
sis_deinit_chardev();
```

```
593 #ifdef CONFIG_PM
594 static int mt_reset_resume(struct hid_device *hdev)
595 {
596     mt_set_input_mode(hdev);
597     return 0;
598 }
599 #endif
600
601 static void mt_remove(struct hid_device *hdev)
602 {
603     struct mt_device *td = hid_get_drvdata(hdev);
604     //SiS FW update
605     sis_deinit_chardev();
606     hid_hw_stop(hdev);
607     kfree(td->slots);
608     kfree(td);
609     hid_set_drvdata(hdev, NULL);
610 }
611
612 static const struct hid_device_id mt_devices[] = {
613
614     /* 3M panels */
615     { .driver_data = MT_CLS_3M,
616         HID_USB_DEVICE(USB_VENDOR_ID_3M,
617             USB_DEVICE_ID_3M1968) },
618     { .driver_data = MT_CLS_3M,
619         HID_USB_DEVICE(USB_VENDOR_ID_3M,
620             USB_DEVICE_ID_3M2256) },
621     { .driver_data = MT_CLS_3M,
622         HID_USB_DEVICE(USB_VENDOR_ID_3M,
623             USB_DEVICE_ID_3M3266) },
624
625     /* ActionStar panels */
626     { .driver_data = MT_CLS_DEFAULT,
627         HID_USB_DEVICE(USB_VENDOR_ID_ACTIONSTAR,
628             USB_DEVICE_ID_ACTIONSTAR_1011) },
629
```

Part 4: “id-table” in static const struct hid_device_id mt_devices[]

```
    /* SiS device */
    { .driver_data = MT_CLS_DEFAULT,
        HID_USB_DEVICE(USB_VENDOR_ID_SIS2_TOUCH,
            USB_PRODUCT_ID_SIS817_TOUCH) },
    { .driver_data = MT_CLS_DEFAULT,
        HID_USB_DEVICE(USB_VENDOR_ID_SIS2_TOUCH,
            USB_PRODUCT_ID_SISF817_TOUCH) },
```

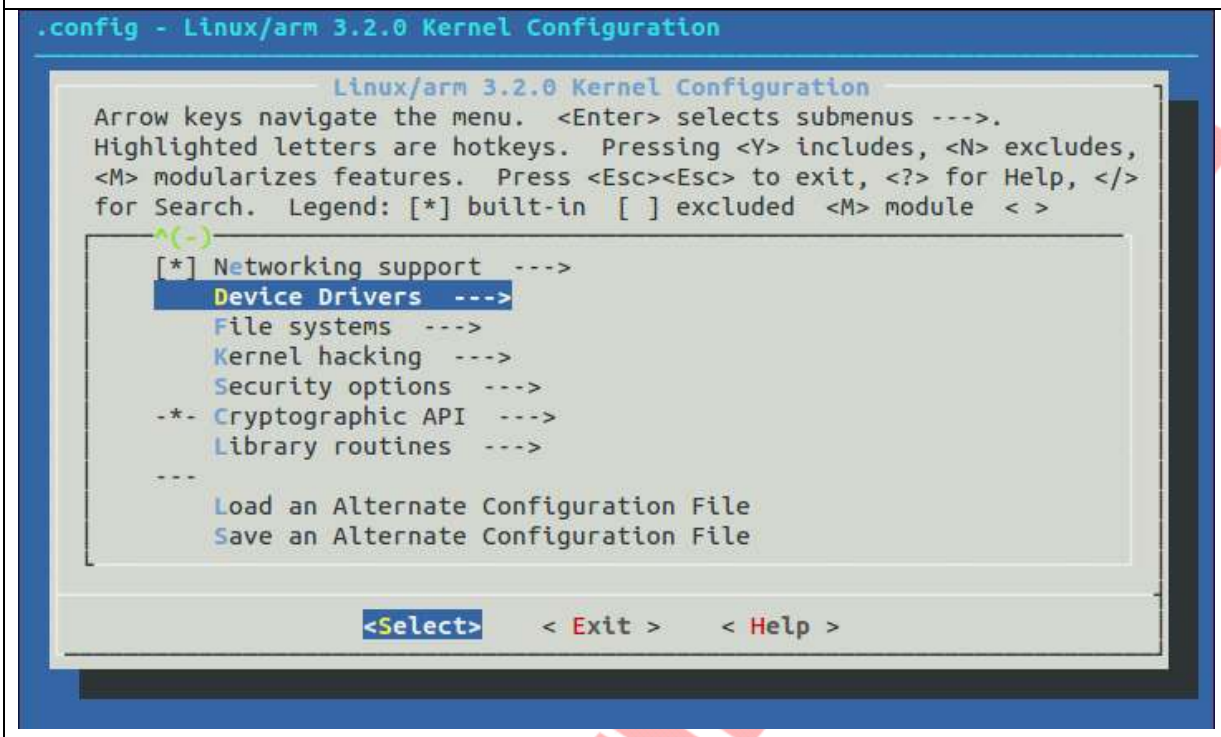
```
652 static const struct hid_device_id mt_devices[] = {
653
654     /* 3M panels */
655     { .driver_data = MT_CLS_3M,
656         HID_USB_DEVICE(USB_VENDOR_ID_3M,
657             USB_DEVICE_ID_3M1968) },
658     { .driver_data = MT_CLS_3M,
659         HID_USB_DEVICE(USB_VENDOR_ID_3M,
660             USB_DEVICE_ID_3M2256) },
661     { .driver_data = MT_CLS_3M,
662         HID_USB_DEVICE(USB_VENDOR_ID_3M,
663             USB_DEVICE_ID_3M3266) },

…
```

```
758     /* Touch International panels */
759     { .driver_data = MT_CLS_DEFAULT,
760       HID_USB_DEVICE(USB_VENDOR_ID_TOUCH_INTL,
761         USB_DEVICE_ID_TOUCH_INTL_MULTI_TOUCH) },
762
763     /* Unitec panels */
764     { .driver_data = MT_CLS_DEFAULT,
765       HID_USB_DEVICE(USB_VENDOR_ID_UNITEC,
766         USB_DEVICE_ID_UNITEC_USB_TOUCH_0709) },
767     { .driver_data = MT_CLS_DEFAULT,
768       HID_USB_DEVICE(USB_VENDOR_ID_UNITEC,
769         USB_DEVICE_ID_UNITEC_USB_TOUCH_0A19) },
770     /* XAT */
771     { .driver_data = MT_CLS_DEFAULT,
772       HID_USB_DEVICE(USB_VENDOR_ID_XAT,
773         USB_DEVICE_ID_XAT_CSR) },
774
775     /* SiS device */
776     { .driver_data = MT_CLS_DEFAULT,
777       HID_USB_DEVICE(USB_VENDOR_ID_SIS2_TOUCH,
778         USB_PRODUCT_ID_SIS817_TOUCH) },
779     { .driver_data = MT_CLS_DEFAULT,
780       HID_USB_DEVICE(USB_VENDOR_ID_SIS2_TOUCH,
781         USB_PRODUCT_ID_SISF817_TOUCH) },
782
783     { }
784 };
785 MODULE_DEVICE_TABLE(hid, mt_devices);
```

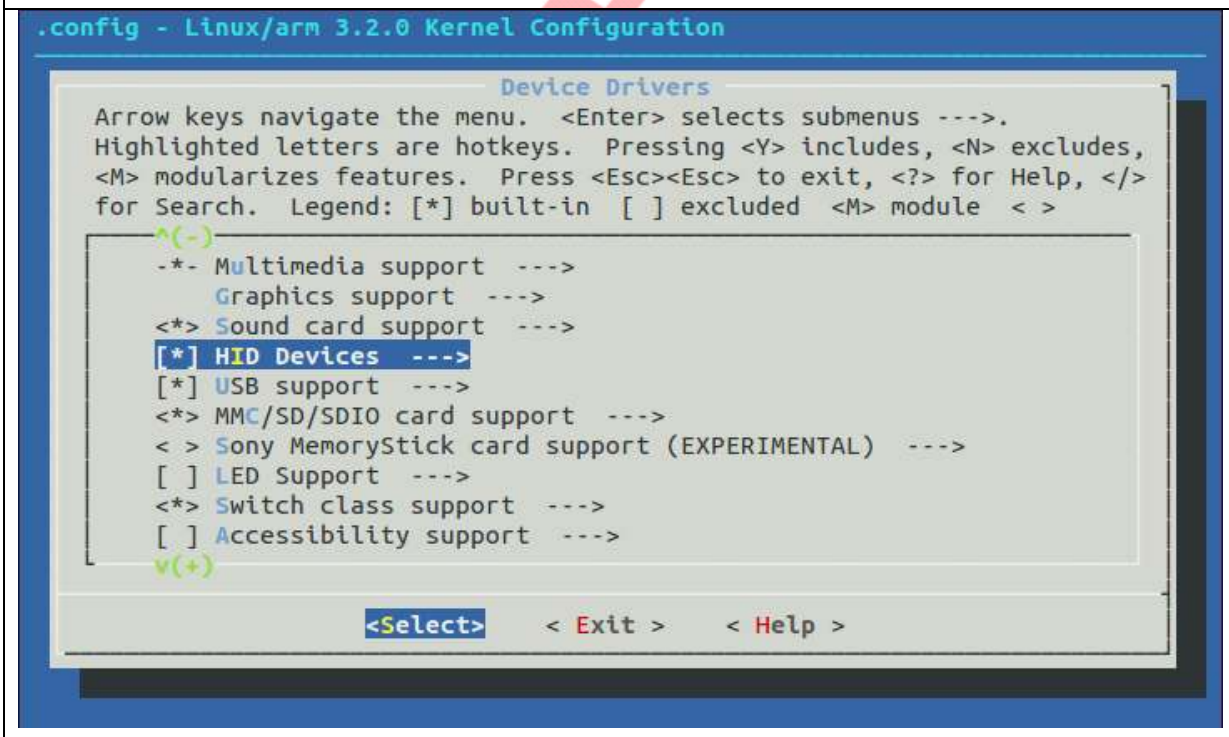
Last, execute “make menuconfig”.

```
$ make menuconfig
```

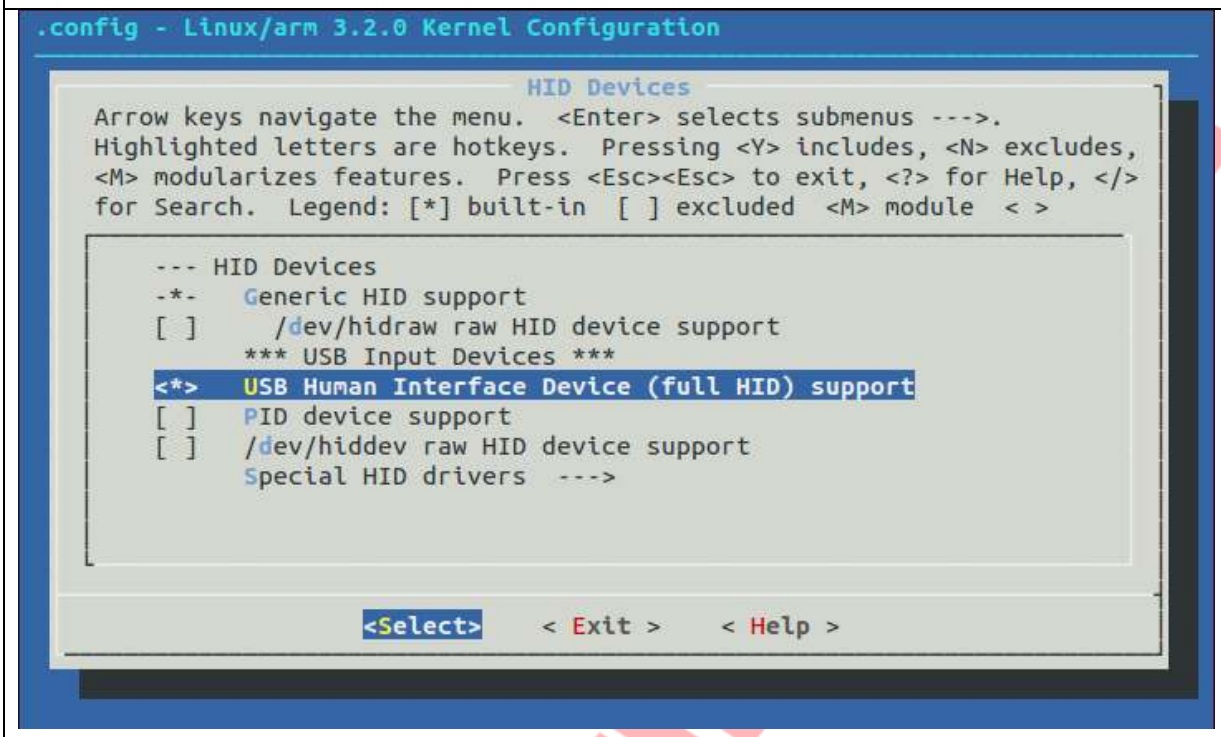
Device Drivers - - - >



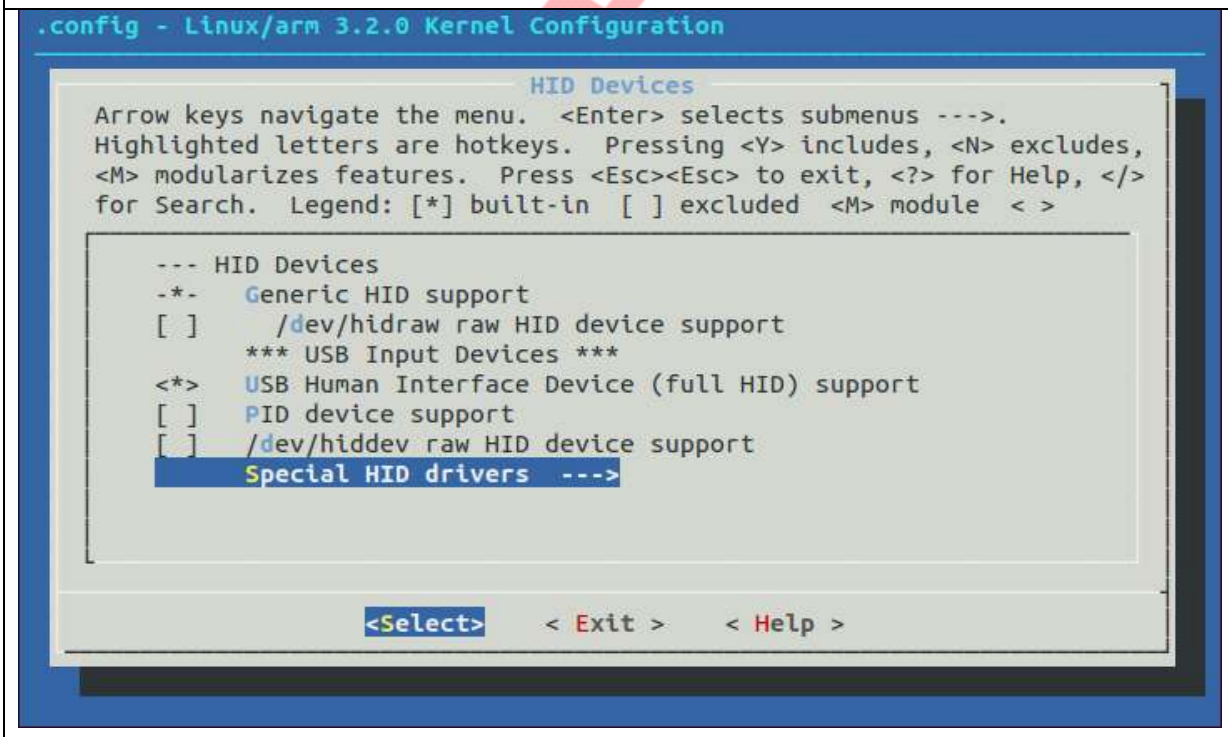
[*] HID Devices - - - >



<*> USB Human Interface Device (full HID) support

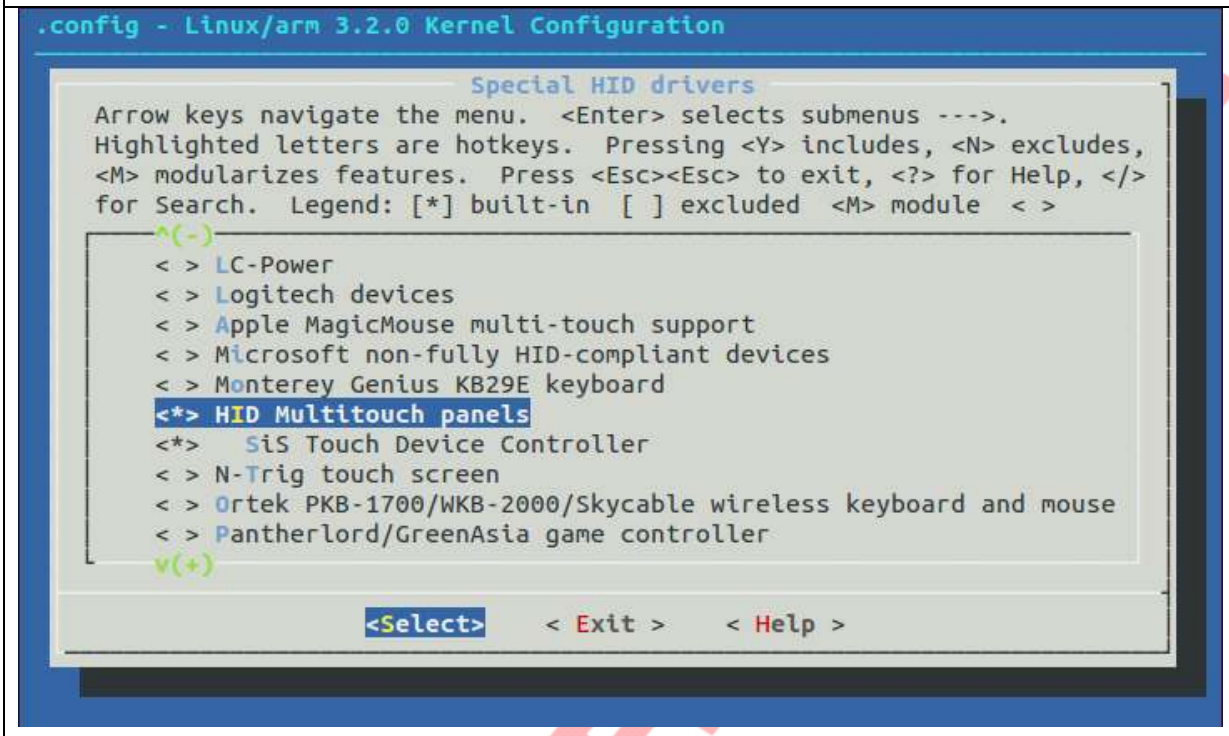


Special HID drivers - - - >



<*> HID Multitouch panels

<*> SiS Touch Device Controller



3.4. Rebuild kernel