

## RedHawk6.5 マルチタッチパネルについて

版	日付	2016/03/14	記述	加筆訂正				
3	発行	プロフェッショナルサービス	作成	大島龍博	確認		承認	
版	日付	2016/03/14	記述	5章,6章追記				
2	発行	プロフェッショナルサービス	作成	大島龍博	確認		承認	
版	日付	2015/07/07	記述	新規発行				
1	発行	プロフェッショナルサービス	作成	大島龍博	確認		承認	

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	1/36

## 目次

RedHawk6.5 マルチタッチパネルについて .....	1
1 概要 .....	5
1.1 適用範囲 .....	5
2 マルチタッチパネルの構成方法 .....	5
2.1 Dell S2240T 21.5" マルチタッチモニタ .....	5
2.2 EIZO 株式会社 DuraVision FDU2603WT .....	5
2.3 HID-MULTITOUCH カーネルモジュールの生成方法 .....	7
2.4 Dell S2240T を RedHawk6.5 に組み込む方法 .....	9
2.5 EIZO DuraVision FDU2603WT を RedHawk6.5 に組み込む方法 .....	11
3 マルチタッチの試験 .....	13
4 試験プログラム(Dell 用) .....	14
5 マウスポインタとタッチポイントの位置が異なる現象について .....	27
5.1 拡張メソッドについて .....	28
6 キャリブレーションについて .....	32
6.1 認識したデバイス名と ID を表示。 .....	32
6.2 タッチスクリーンのプロパティの表示 .....	32
6.3 キャリブレーション .....	33
6.4 キャリブレーション値の確認 .....	33
6.5 キャリブレーション値の設定 .....	34
6.5.1 デバイス名称の確認 .....	34
6.5.2 /etc/X11/xorg.conf への記述 .....	34
6.5.3 /var/log/Xorg.0.log .....	35

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	2/36

## 図 目次

図 2-1 ccur-config の GUI 画面 .....	7
----------------------------------	---

## 表 目次

表 1-1 改版履歴 .....	4
------------------	---

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	3/36

表 1-1 改版履歴

版数	日付	改版内容
1	2015/07/07	初版発行
2	2016/03/11	マウスポインタとタッチポイントの位置が異なる現象について 追記 キャリブレーションについて追記
3	2016/12/22	カーネルコンパイル用 Makefile, 10-unitec.fdi 追加

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	4/36

## 1 概要

### 1.1 適用範囲

本稿は、Redhawk6.5におけるマルチタッチパネルのコンフィグレーション方法について記述する。

## 2 マルチタッチパネルの構成方法

本稿では、2種類のRedhawk6.5におけるマルチタッチパネルのコンフィグレーション方法について記述する。

### 2.1 Dell S2240T 21.5” マルチタッチモニタ

<http://accessories.apj.dell.com/sna/productdetail.aspx?c=jp&l=ja&s=dhs&sku=391-10116>

- **タッチ機能付きの鮮明な画面:** 1,920 x 1,080(最大)のフル HD 解像度、10 点タッチ機能、鮮明な画像を生み出す 3,000:1(標準)のコントラスト比、驚くほど細部まで表現するクリアな画像など、さまざまな特長を備えています。
- **タッチテクノロジー:** 投影型容量方式
- **センサースタックの厚さ:** 1.1 mm
- **カバーガラスの厚さ:** 1.1 +/- 0.2 mm
- **タッチ方法:** 指、薄い手袋
- **タッチポイント:** 10 点タッチ
- **応答速度:** 25 ms

### 2.2 EIZO 株式会社 DuraVision FDU2603WT

<http://www.eizo.co.jp/products/id/fdu2603wt/index.html>

- **船級と IEC 試験規格に適合する IP65 対応の 25.5 型パネルマウント液晶モニター**  
各国の船級規格に適合  
NK (日本)、LR (イギリス)、ABS (アメリカ)、DNV (ノルウェー)、BV (フランス)、GL (ドイツ)、CCS (中国) の適合要件を満たしています。また、ECDIS に要求される IEC 規格に適合しています。
- **耐久性 / 信頼性**  
耐熱試験や振動試験 (IEC60945 準拠)、防塵・防滴試験 (IP65 準拠) を行い、船舶搭

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	5/36

載用途に求められる耐久性を保証しています。電子基板は防湿対策にコーティング処理を施しているほか、可動部のファンは簡単に取り替え修理ができる設計を行っており、長期間安心して使用できます。

- 広視野角パネル採用

広視野角の液晶パネルを採用。設置場所や画面を見る角度による表示（コントラストや階調特性）の変化が少なく視認性に優れています。

- 個体差のない均一表示

1台ごとにバックライトの輝度、表示ガンマ、RGB各色の色度を測定し、表示が均一になるよう調整の上、出荷しています。個体差が小さいため、機器等への組込用途においてもシステム側の調整が最小限で済みます。

- 高輝度表示 / 調光範囲が広い

日中でも視認性に優れる高輝度表示。また、1cd/m<sup>2</sup>以下まで幅広く輝度調整が行え、昼夜で照度変化の大きい環境でも快適に使用できます。

- バックライト LED
- サイズ 65 cm (25.5)型 (可視域対角 64.9 cm)
- 推奨解像度 1920×1200
- 表示面積 (横×縦) 550.0×343.8 mm
- 画素ピッチ 0.2865 mm×0.2865 mm
- 表示色 1677 万色 : 8 bit 対応
- 視野角 (水平/垂直、標準値) 176 ° / 176 °
- 輝度 (標準値) 470 cd/m<sup>2</sup>
- コントラスト比 (標準値) 1500 : 1
- 応答速度 (標準値) 20ms (黒→白→黒)
- 方式 投影型静電方式
- 通信方式 USB シリアル転送、RS-232C シリアル転送
- タッチ耐久性 5000 万回以上
- 表面硬度 5H

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	6/36

## 2.3 HID-MULTITOUCH カーネルモジュールの生成方法

下記手順で、RedHawk の HID-MOULTITOUCH カーネルモジュールを生成する

```
# cd /usr/src/linux-3.10.??RedHawk6.5
# ./ccur_config -c
```

下記 GUI で、 **HID Multitouch panels** をチェックし、File メニューの Save を選択し、終了する

(注:  を選択した場合には、カーネルにスタティックに組み込まれる)

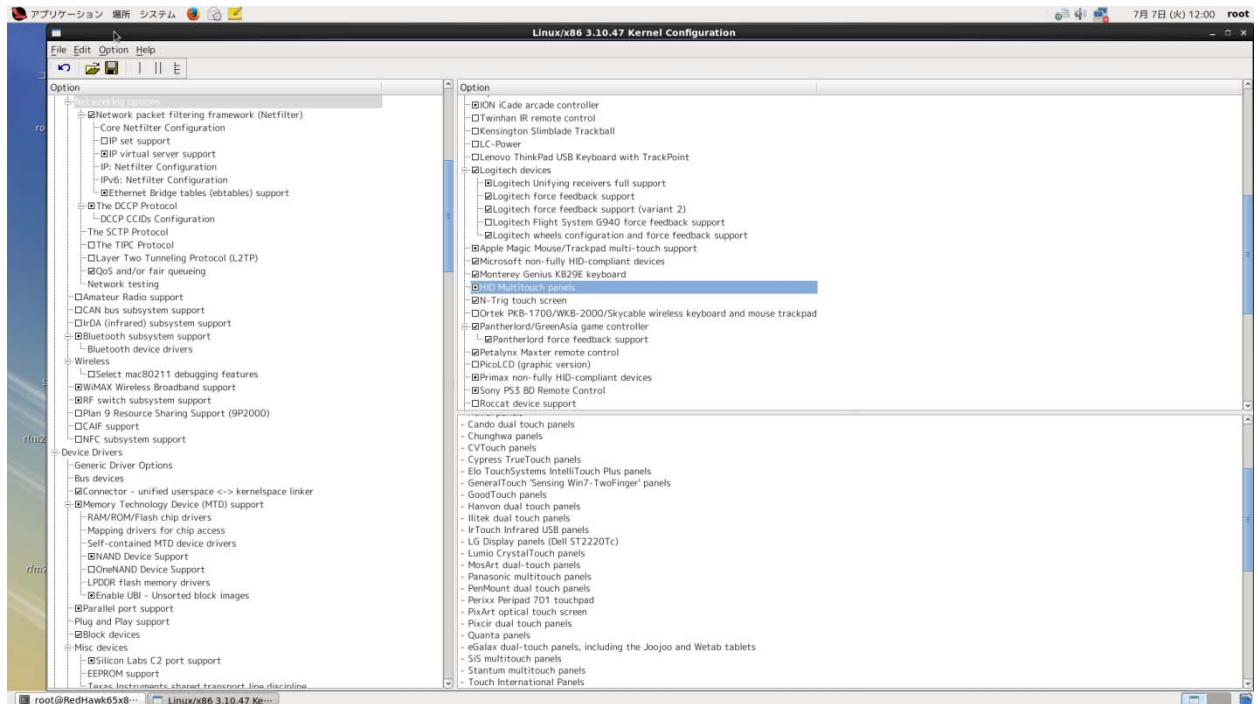


図 2-1 ccur-config の GUI 画面

設定は、.config に保存されているので、これをオリジナルの設定ファイルに上書きする。(ここでは trace カーネル用を想定している)

```
# cp .config configs/x86_64/trace
```

以下の Makefile を適当なディレクトリで作成する。

Makefile

```
KERNEL_DIR = /lib/modules/`uname -r`/build
MULTITOUCH_DIR = ${KERNEL_DIR}/drivers/hid
obj-m := hid-multitouch.o
```

default:

```
make -C ${KERNEL_DIR} SUBDIRS=${MULTITOUCH_DIR} REDHAWKFLAVOR=`cat /proc/ccur/flavor` modules
```

config:

```
cd ${KERNEL_DIR}; ./ccur-config -n -c
```

install:

```
make -C ${KERNEL_DIR} SUBDIRS=${MULTITOUCH_DIR} REDHAWKFLAVOR=`cat /proc/ccur/flavor`
```

modules\_install

modprobe:

```
modprobe ${obj-m}
```

clean:

```
make -C ${KERNEL_DIR} SUBDIRS=${MULTITOUCH_DIR} REDHAWKFLAVOR=`cat /proc/ccur/flavor` clean
```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	7/36

以下の手順でカーネルモジュールをコンパイルする

```
# make config
```

```
# make modules_install
```

```
# make install
```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	8/36



## 2.4 Dell S2240T を RedHawk6.5 に組み込む方法

### hal から evdev を組み込む

dmesg コマンドを実行し、以下のメッセージを確認する

```
usb 4-1: USB disconnect, device number 8
usb 4-1: new full-speed USB device number 9 using uhci_hcd
usb 4-1: New USB device found, idVendor=2149, idProduct=2122
usb 4-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
usb 4-1: Product: CoolTouch(TM) System
usb 4-1: Manufacturer: Advanced Silicon S.A.
hid-generic 0003:2149:2122.0013: hiddev0,hidraw2: USB HID v1.10 Device [Advanced Silicon S.A. CoolTouch(TM) System] on
usb-0000:00:1a.1-1/input1
```

lshal コマンドを実行し、下記デバイス表示を確認する、

```
udi = /org/freedesktop/Hal/devices/usb_device_2149_2122_noserial'
info.bus = 'usb_device' (string)
info.linux.driver = 'usb' (string)
info.parent = /org/freedesktop/Hal/devices/usb_device_1d6b_1_0000_00_1a_1' (string)
info.product = 'CoolTouch(TM) System' (string)
info.subsystem = 'usb_device' (string)
info.udi = /org/freedesktop/Hal/devices/usb_device_2149_2122_noserial' (string)
info.vendor = 'Advanced Silicon S.A.' (string)
linux.device_file = '/dev/bus/usb/004/008' (string)
linux.hotplug_type = 2 (0x2) (int)
linux.subsystem = 'usb' (string)
linux.sysfs_path = '/sys/devices/pci0000:00/0000:00:1a.1/usb4/4-1' (string)
usb_device.bus_number = 4 (0x4) (int)
usb_device.can_wake_up = true (bool)
usb_device.configuration = 'CoolTouch 速 System 104' (string)
usb_device.configuration_value = 1 (0x1) (int)
usb_device.device_class = 0 (0x0) (int)
usb_device.device_protocol = 0 (0x0) (int)
usb_device.device_revision_bcd = 4913 (0x1331) (int)
usb_device.device_subclass = 0 (0x0) (int)
usb_device.is_self_powered = false (bool)
usb_device.linux.device_number = 8 (0x8) (int)
usb_device.linux.sysfs_path = '/sys/devices/pci0000:00/0000:00:1a.1/usb4/4-1' (string)
usb_device.max_power = 500 (0x1f4) (int)
usb_device.num_configurations = 1 (0x1) (int)
usb_device.num_interfaces = 2 (0x2) (int)
usb_device.num_ports = 0 (0x0) (int)
usb_device.product = 'CoolTouch(TM) System' (string)
usb_device.product_id = 8482 (0x2122) (int)
usb_device.speed = 12.0 (12) (double)
usb_device.vendor = 'Advanced Silicon S.A.' (string)
usb_device.vendor_id = 8521 (0x2149) (int)
usb_device.version = 2.0 (2) (double)
```

この info.product = 'CoolTouch(TM) System' を使用し、下記の

**"/usr/share/hal/fdi/policy/20thirdparty/10-cooltouch.fdi"**を生成する。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.product" contains="CoolTouch(TM) System">
      <merge key="input.x11_driver" type="string">evdev</merge>
    </match>
  </device>
</deviceinfo>
```

再起動し、以下のコマンドで、CoolTouch が組み込まれていることを確認する。

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	9/36

```
# xinput
┌ Virtual core pointer          id=2      [master pointer (3)]
│   ↳ Virtual core XTEST pointer  id=4      [slave pointer (2)]
│   ↳ Dell Dell USB Mouse         id=9      [slave pointer (2)]
│   ↳ Advanced Silicon S.A. CoolTouch(TM) System id=10     [slave pointer (2)]
└ Virtual core keyboard       id=3      [master keyboard (2)]
   ↳ Virtual core XTEST keyboard  id=5      [slave keyboard (3)]
   ↳ Dell Dell USB Keyboard       id=6      [slave keyboard (3)]
   ↳ Power Button                 id=7      [slave keyboard (3)]
   ↳ Power Button
```

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

```
⋮
⋮
I: Bus=0003 Vendor=2149 Product=2122 Version=0110
N: Name="Advanced Silicon S.A. CoolTouch(TM) System"
P: Phys=usb-0000:00:1a.0-2/input0
S: Sysfs=/devices/pci0000:00/0000:00:1a.0/usb3/3-2/3-2:1.0/input/input5
U: Uniq=
H: Handlers=mouse1 event5
B: PROP=2
B: EV=b
B: KEY=400 0 0 0 0 0
B: ABS=260800000000003
⋮
⋮
```

この時点で、タッチパネルは動作している このデバイスには、`/etc/X11/xorg.conf` に特別な記述は必要ない。

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	10/36

## 2.5 EIZO DuraVision FDU2603WT を RedHawk6.5 に組み込む方法

dmesg コマンドを実行し、以下のメッセージを確認する

```
usb 1-2.1: new full-speed USB device number 3 using ehci-pci
usb 1-2.1: New USB device found, idVendor=056d, idProduct=0002
usb 1-2.1: New USB device strings: Mfr=4, Product=14, SerialNumber=0
usb 1-2.1: Product: EIZO USB HID Monitor
usb 1-2.1: Manufacturer: EIZO
usb 1-2.2: new full-speed USB device number 4 using ehci-pci
usb 1-2.2: New USB device found, idVendor=0afa, idProduct=07d2
usb 1-2.2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
usb 1-2.2: Product: USB Touch (WinXP&7)
usb 1-2.2: Manufacturer: UNITEC
:
:
input: UNITEC USB Touch (WinXP&7) Pen as /devices/pci0000:00/0000:00:1a.7/usb1/1-2/1-2.2/1-2.2:1.0/input/input5
input: UNITEC USB Touch (WinXP&7) as /devices/pci0000:00/0000:00:1a.7/usb1/1-2/1-2.2/1-2.2:1.0/input/input6
hid-multitouch 0003:0AFA:07D2:0004: input,hiddev0,hidraw4: USB HID v1.11 Pointer [UNITEC USB Touch (WinXP&7)] on
usb-0000:00:1a.7-2.2/input0
```

以下のコマンドで、UNITEC USB Touch (WinXP&7)の Handlers デバイスを確認する。

"UNITEC USB Touch (WinXP&7) Pen"では動作しないので注意。

```
# cat /proc/bus/input/devices
:
:
I: Bus=0003 Vendor=0afa Product=07d2 Version=0111
N: Name="UNITEC USB Touch (WinXP&7) Pen"
P: Phys=usb-0000:00:1a.7-2.2/input0
S: Sysfs=/devices/pci0000:00/0000:00:1a.7/usb1/1-2/1-2.2/1-2.2:1.0/input/input5
U: Uniq=
H: Handlers=mouse1 event5
B: PROP=0
B: EV=1b
B: KEY=c01 0 0 0 0
B: ABS=7
B: MSC=10

I: Bus=0003 Vendor=0afa Product=07d2 Version=0111
N: Name="UNITEC USB Touch (WinXP&7)"
P: Phys=usb-0000:00:1a.7-2.2/input0
S: Sysfs=/devices/pci0000:00/0000:00:1a.7/usb1/1-2/1-2.2/1-2.2:1.0/input/input6
U: Uniq=
H: Handlers=mouse2 event6
B: PROP=2
B: EV=b
B: KEY=400 0 0 0 0
B: ABS=2608000000000003
```

lshal コマンドを実行し、下記 UDI 表示を確認する

```
udi = /org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0_logicaldev_input_0'
button.has_state = false (bool)
info.addons.singleton = {'hald-addon-input'} (string list)
info.capabilities = {'input', 'button'} (string list)
info.category = 'input' (string)
info.parent = /org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0' (string)
info.product = 'UNITEC USB Touch (WinXP&7)' (string)
info.subsystem = 'input' (string)
info.udi = /org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0_logicaldev_input_0' (string)
input.device = /dev/input/event4' (string)
input.originating_device = /org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0' (string)
input.product = 'UNITEC USB Touch (WinXP&7)' (string)
linux.device_file = /dev/input/event4' (string)
linux.hotplug_type = 2 (0x2) (int)
linux.subsystem = 'input' (string)
linux.sysfs_path = /sys/devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.6/2-1.6.3/2-1.6.3:1.0/input/input4/event4' (string)
```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	11/36

```
udi = '/org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0_logicaldev_input'
info.capabilities = {'input', 'input.tablet'} (string list)
info.category = 'input' (string)
info.parent = '/org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0' (string)
info.product = 'UNITEC USB Touch (WinXP&7) Pen' (string)
info.subsystem = 'input' (string)
info.udi = '/org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0_logicaldev_input' (string)
input.device = '/dev/input/event3' (string)
input.originating_device = '/org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0' (string)
input.product = 'UNITEC USB Touch (WinXP&7) Pen' (string)
input.x11_driver = 'evdev' (string)
linux.device_file = '/dev/input/event3' (string)
linux.hotplug_type = 2 (0x2) (int)
linux.subsystem = 'input' (string)
linux.sysfs_path = '/sys/devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.6/2-1.6.3/2-1.6.3.1.0/input/input3/event3' (string)
```

この 2 つの UDI を使用し、下記の

”/usr/share/hal/fdi/policy/20thirdparty/10-unitec.fdi”を生成する。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.udi" string="/org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0_logicaldev_input_0">
      <merge key="input.x11_driver" type="string">evdev</merge>
    </match>
    <!-- match key="info.udi" string="/org/freedesktop/Hal/devices/usb_device_afa_7d2_noserial_if0_logicaldev_input">
      <remove key="input.x11_driver" />
    </match -->
  </device>
</deviceinfo>
```

再起動し、以下のコマンドで、UNITEC USB Touch (WinXP&7)が組み込まれていることを確認する UNITEC USB Touch (WinXP&7) Pen は、動作しない。)

```
# xinput
┌ Virtual core pointer                id=2          [master pointer (3)]
├   ↳ Virtual core XTEST pointer      id=4          [slave pointer (2)]
├   ↳ USB Optical Mouse                id=8          [slave pointer (2)]
├   ↳ UNITEC USB Touch (WinXP&7)       id=9          [slave pointer (2)]
├   ↳ UNITEC USB Touch (WinXP&7) Pen id=10         [slave pointer (2)]
└ Virtual core keyboard              id=3          [master keyboard (2)]
   ↳ Virtual core XTEST keyboard      id=5          [slave keyboard (3)]
   ↳ DELL Dell QuietKey Keyboard      id=6          [slave keyboard (3)]
   ↳ Power Button                     id=7          [slave keyboard (3)]
```

この時点で、タッチパネルは動作している この方法では、/etc/X11/xorg.conf に特別な記述は必要ない。

下記は固定的に割り当てる方法であるが、異なる USB ポートに挿入すると、動作しない。

/etc/X11/xorg.conf に Handlers デバイスを追加する。

```
Section "ServerLayout"
  Identifier      "Layout0"
  Screen          0  "Screen0"
  InputDevice     "Keyboard0" "CoreKeyboard"
  InputDevice     "Mouse0" "CorePointer"
  InputDevice     "Mouse1" "CorePointer"
EndSection

Section "InputDevice"
# generated from default
  Identifier      "Mouse1"
  Driver          "evdev"
  Option          "Protocol" "auto"
  Option          "Device" "/dev/input/event6"
  Option          "Emulate3Buttons" "no"
  Option          "ZAxisMapping" "4 5"
EndSection
```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	12/36

### 3 マルチタッチの試験

以下の Handlers 行から、タッチパネルイベントは"/dev/input/event5"であることが判る。

```
# cat /proc/bus/input/devices
:
:
I: Bus=0003 Vendor=2149 Product=2122 Version=0110
N: Name="Advanced Silicon S.A. CoolTouch(TM) System"
P: Phys=usb-0000:00:1a.0-2/input0
S: Sysfs=/devices/pci0000:00/0000:00:1a.0/usb3/3-2/3-2:1.0/input/input5
U: Uniq=
H: Handlers=mouse1 event5
B: PROP=2
B: EV=b
B: KEY=400 0 0 0 0 0
B: ABS=2608000000000003
:
```

4 に示す試験プログラムをコンパイルし、マルチタッチであることを確認する

```
# cc -o test test.c -lX11
```

```
# ./test /dev/input/event5
```



表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	13/36

#### 4 試験プログラム(Dell 用)

```

/*****/
/* test program */
/*****/

#include <stdio.h>
#include <stdlib.h>
#include <linux/input.h>
#include <bits/time.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>

#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/Xatom.h>

#define BUFF_SIZE 256
#define POINTER 10                                /* 最大認識タッチ数 */

#define WIDTH 1920                                /* 出力画面 X DELL:1920 */
#define HEIGHT 1024                               /* 出力画面 Y DELL:1024 */

#define TOP 52                                    /* 上部削り分 DELL:52 */
#define DEVX 17                                  /* X 割値 DELL:17 */
#define DEVY 31                                  /* Y 割値 DELL:31 */

#define BUTTON_NUM 3                              /* ボタン数 */
#define OBJECT_NUM 64                            /* 物体数 */
#define OBJECT_SIZE_X 100                        /* 物体の初期 X */
#define OBJECT_SIZE_Y 100                        /* 物体の初期 Y */
#define OBJECT_MIN_SIZE 50                       /* 物体の最小サイズ */
#define OBJECT_MAX_SIZE 600                      /* 物体の最大サイズ */

#define STRLEN 16                                /* 表示用文字列幅 */

static int pointer_num;                          /* 読込タッチ数 */
static int mon_x;                                /* 読込モニターX 値 */
static int mon_y;                                /* 読込モニターY 値 */
static int corsor_size;                         /* trace モードのカーソルサイズ */

struct _pointer {
    int x;
    int y;
    int object_x;
    int object_y;
    int diff_x;
    int diff_y;
    int tracking;
    int b1_set;
    int catch;
    int lock;
} pointer[POINTER];

struct _object {
    Window win;
    int x;
    int y;
    int diff_x;
    int diff_y;
    int first_diff_x;
    int first_diff_y;
    int size_x;
    int size_y;
    int now_x;
    int now_y;
} object[OBJECT_NUM];

int bit(int i, char *buff);
void print_abs(int fd);
static void corsor(Display* display, Window window, GC gc, unsigned long color, int x, int y);

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	14/36

```

void color_set(Display* display, int screen, unsigned long *set_color);

int main(int argc, char *argv[])
{
    int i, j, k, l, object_num, slot, flag;
    int fd;
    char buff[BUFF_SIZE];
    struct timeval now, old;
    struct input_event event[64];
    char *str[STRLEN];

    enum {
        FLAG_B0 = 0x0001,
        FLAG_B1 = 0x0002,
        FLAG_B2 = 0x0004
    };

    slot = 0;
    j = 0;
    object_num = 0;

    for(i = 0; i < POINTER; i++)
    {
        pointer[i].x = 0;
        pointer[i].y = 0;
        pointer[i].tracking = 0;
        pointer[i].catch = -1;
        pointer[i].lock = 0;
    }

    /* 画面宣言 */
    Display *display;
    int screen;
    Window mwindow;
    Window sub;
    int button_x[BUTTON_NUM], button_y[BUTTON_NUM];
    unsigned int button_size_x, button_size_y;
    GC gc;
    XEvent xevent;
    unsigned long set_color[POINTER];

    button_size_x = 100;
    button_size_y = 50;

    /* 初期時刻 */
    struct input_event first;
    old.tv_usec = first.time.tv_usec;

    /* イベントファイルの読み込み */
    if (argc <= 1)
    {
        printf("dump_event_file\n");
        return -1;
    }
    fd = open(argv[1], O_RDONLY);

    if (fd < 0)
    {
        printf("open error\n");
        return -1;
    }

    /* 初期値表示 */
    ioctl(fd, EVIOCGNAME(BUFF_SIZE), buff);
    puts(buff);

    ioctl(fd, EVIOCGBIT(0, EV_MAX), buff);
    for (i = 0; i < EV_MAX; i++)
    {
        bit(i, buff);
        switch(i)
        {
            case EV_ABS:
                printf("Absolute Axes:\n");
                print_abs(fd);
        }
    }
}

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	15/36

```

                                break;
                                default:
                                break;
                                }
                                }

/* X11 セット */
if ((display = XOpenDisplay(NULL)) == NULL)          /* Xserver との接続 */
{
    fprintf(stderr, "can not set X server\n");
    return 1;
}

screen = DefaultScreen(display);
color_set(display, screen, set_color);
cursor_size = 6;

/* ウィンドウ設定 */

/* メイン画面 */
mwindow= XCreateSimpleWindow(display, RootWindow(display, 0), 0, 0, WIDTH, HEIGHT, 1, BlackPixel(display, 0),
WhitePixel(display, 0));
XStoreName(display, mwindow, "main test window");
XSelectInput(display,                                mwindow,
ExposureMask|KeyPressMask|ButtonPressMask|ButtonReleaseMask|PointerMotionMask|EnterWindowMask|LeaveWindowMas
k|PointerMotionMask|Button1MotionMask);

/* サブ画面 */
sub = XCreateSimpleWindow(display, mwindow, WIDTH-100, 0, 100, 280, 0, BlackPixel(display, 0), WhitePixel(display,
0));
XSelectInput(display,                                sub,
ExposureMask|KeyPressMask|ButtonPressMask|ButtonReleaseMask|PointerMotionMask|EnterWindowMask|LeaveWindowMas
k|PointerMotionMask|Button1MotionMask);

/* グラフィック設定 */
gc = XCreateGC(display, mwindow, 0, 0);

/* button */
for(i = 0; i < BUTTON_NUM ; i++)
{
    button_x[i] = 0;
    button_y[i] = 100 + (i * 60);
    XSetForeground(display, gc, set_color[1+i]);
    XFillRectangle(display, sub, gc, button_x[i], button_y[i], button_size_x, button_size_y);
}

/* object */
for(i = 0; i < OBJECT_NUM; i++, j++)
{
    if(i == 10)
    {
        j = 0;
    }
    object[i].x = 0;
    object[i].y = 0;
    object[i].now_x = 0;
    object[i].now_y = 0;
    object[i].size_x = OBJECT_SIZE_X;
    object[i].size_y = OBJECT_SIZE_Y;
    object[i].win = XCreateSimpleWindow(display, mwindow, object[i].x, object[i].y, object[i].size_x,
object[i].size_y, 0, 0, set_color[j]);
    XSelectInput(display,                                object[i].win,
ExposureMask|ButtonPressMask|ButtonReleaseMask|EnterWindowMask|LeaveWindowMask|PointerMotionMask|Button1Motion
Mask);
}

/* window 開始 */
XMapWindow(display, mwindow);
XMapWindow(display, sub);

size_t read_size;

/* 最初に一度 event を読み込む? */

/* 取得開始 */

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	16/36



```

for(;;)
{
    while(XPending(display))
    {
        XNextEvent(display, &xevent);
    }

/* タッチパネルイベント操作 */
read_size = read(fd, event, sizeof(event));
for(i = 0; i < (int)(read_size / sizeof(struct input_event)); i++)
{
    if(event[i].time.tv_usec == old.tv_usec)
    {
        now.tv_usec = event[i].time.tv_usec;
    }
    else if (event[i].time.tv_usec != old.tv_usec)
    {
        old.tv_usec = now.tv_usec;
        now.tv_usec = event[i].time.tv_usec;
    }

/* タッチイベント読み込み */
switch(event[i].type)
{
    case EV_ABS:
        switch(event[i].code)
        {
            case ABS_MT_SLOT:
                if(event[i].value >= 0)
                {
                    slot = event[i].value;
                }
                break;
            case ABS_MT_TRACKING_ID:
                if(event[i].value > 0)
                {
                    pointer[slot].tracking = 1;
                }
                else if (event[i].value == -1)
                {
                    pointer[slot].tracking = 0;
                    pointer[slot].x = 0;
                    pointer[slot].y = 0;
                    pointer[slot].catch = -1;
                    pointer[slot].lock = 0;
                    pointer[slot].diff_x = 0;
                    pointer[slot].diff_y = 0;
                    object[slot].first_diff_x = 0;
                    object[slot].first_diff_y = 0;
                    object[slot].diff_x = 0;
                    object[slot].diff_y = 0;
                }
                break;
            case ABS_MT_POSITION_X:
                pointer[slot].x = event[i].value;
                break;
            case ABS_MT_POSITION_Y:
                pointer[slot].y = event[i].value;
                break;
            default:
                break;
        }
        }
        break;
    default:
        break;
}

/* 画面操作 */

while(XPending(display))
{
    XNextEvent(display, &xevent);
    switch(xevent.type)
    {
/* Expose */

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	17/36

```

case Expose:
    if (xevent.xexpose.count == 0)
    {
        for(j = 0; j < BUTTON_NUM; j++)
        {
            button_x[j] = 0;
            button_y[j] = 100+(j*60);
            XSetForeground(display, gc,
                XFillRectangle(display, sub, gc,
                    button_x[j], button_y[j], button_size_x, button_size_y);

            WhitePixel(display, screen);
            button_x[0], button_y[0], button_size_x, button_size_y);
            set_color[0];
            button_x[0], button_y[0], button_size_x, button_size_y);

            WhitePixel(display,screen);
            120, str[0], strlen(str[0]));

            set_color[1];
            button_x[0], button_y[0], button_size_x, button_size_y);

            BlackPixel(display,screen);
            120, str[0], strlen(str[0]));

            set_color[9];
            button_x[1], button_y[1], button_size_x, button_size_y);

            WhitePixel(display,screen);
            180, str[0], strlen(str[0]));

            set_color[2];
            button_x[1], button_y[1], button_size_x, button_size_y);

            BlackPixel(display,screen);
            180, str[0], strlen(str[0]));
        }
        /* button[0] */
        if((flag & FLAG_B0) == FLAG_B0)
        {
            /* trace mode */
            cursor_size = 16;
            XSetForeground(display, gc,
                XFillRectangle(display, sub, gc,
                    XSetForeground(display, gc,
                        XFillRectangle(display, sub, gc,
                            XSetForeground(display, gc,
                                *str= "trace mode";
                                XDrawString(display, sub, gc, 20,
                                    }
            } else if ((flag & FLAG_B0) != FLAG_B0)
            {
                /* paint mode */
                cursor_size = 6;
                XSetForeground(display, gc,
                    XFillRectangle(display, sub, gc,
                        XSetForeground(display, gc,
                            *str = "paint mode";
                            XDrawString(display, sub, gc, 20,
                                }
            }
            /* button[1] */
            if((flag & FLAG_B1) == FLAG_B1)
            {
                XSetForeground(display, gc,
                    XFillRectangle(display, sub, gc,
                        XSetForeground(display, gc,
                            *str= "set mode";
                            XDrawString(display, sub, gc, 20,
                                }
            } else if ((flag & FLAG_B1) != FLAG_B1)
            {
                XSetForeground(display, gc,
                    XFillRectangle(display, sub, gc,
                        XSetForeground(display, gc,
                            *str = "move mode";
                            XDrawString(display, sub, gc, 20,
                                }
            }
    }

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	18/36

```

}
/* button[2] */
XSetForeground(display, gc, BlackPixel(display,
screen));
strlen(str[0]));
}
break;
/* Key Press */
case KeyPress:
XDestroyWindow(display, mwindow);
XCloseDisplay(display);
return 0;
break;
/* Press Button */
case ButtonPress:
/* button[0] */
if((xevent.xbutton.x >= button_x[0] && xevent.xbutton.x <=
&& (xevent.xbutton.y >= button_y[0] && xevent.xbutton.y <=
&& xevent.xany.window == sub)
{
if((flag & FLAG_B0) == FLAG_B0)
{
/* paint mode */
printf("paint mode\n");
cursor_size = 6;
XSetForeground(display, gc,
XFillRectangle(display, sub, gc,
XSetForeground(display, gc,
*str = "paint mode";
XDrawString(display, sub, gc, 20,
}
else if ((flag & FLAG_B0) != FLAG_B0)
{
/* trace mode */
printf("trace mode\n");
cursor_size = 16;
XSetForeground(display, gc,
XFillRectangle(display, sub, gc,
XSetForeground(display, gc,
XFillRectangle(display, sub, gc,
XSetForeground(display, gc,
*str = "trace mode";
XDrawString(display, sub, gc, 20,
}
}
/* button[1] */
if((xevent.xbutton.x >= button_x[1] && xevent.xbutton.x <=
&& (xevent.xbutton.y >= button_y[1] && xevent.xbutton.y <=

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	19/36



```

button_x[0] + button_size_x )
button_y[0] + button_size_y )

set_color[1]);
button_x[0], button_y[0], button_size_x, button_size_y);

BlackPixel(display,screen));

120, str[0], strlen(str[0]));

set_color[0]);
button_x[0], button_y[0], button_size_x, button_size_y);

WhitePixel(display,screen));

120, str[0], strlen(str[0]));

XSetForeground(display,gc,WhitePixel(display,screen));
XFillRectangle(display,mwindow,gc,0,0,WIDTH,HEIGHT);

button_x[1] + button_size_x )
button_y[1] + button_size_y )

set_color[2]);
button_x[1], button_y[1], button_size_x, button_size_y);

BlackPixel(display,screen));

180, str[0], strlen(str[0]));

button_x[2] + button_size_x )
button_y[2] + button_size_y )
&& (xevent.xbutton.y >= button_y[0] && xevent.xbutton.y <=
&& xevent.xany.window == sub)
{
    if((flag & FLAG_B0) == FLAG_B0)
    {
        XSetForeground(display, gc,
        XFillRectangle(display, sub, gc,

        XSetForeground(display, gc,

        *str = "paint mode";
        XDrawString(display, sub, gc, 20,
        flag &= ~FLAG_B0;
    }
    else if ((flag & FLAG_B0) != FLAG_B0)
    {
        XSetForeground(display, gc,
        XFillRectangle(display, sub, gc,

        XSetForeground(display, gc,

        *str= "trace mode";
        XDrawString(display, sub, gc, 20,

        flag |= FLAG_B0;
    }
}
/* button[1] */
if((xevent.xbutton.x >= button_x[1] && xevent.xbutton.x <=
&& (xevent.xbutton.y >= button_y[1] && xevent.xbutton.y <=
&& xevent.xany.window == sub)
{
    if((flag & FLAG_B1) == FLAG_B1)
    {
        XSetForeground(display, gc,
        XFillRectangle(display, sub, gc,

        XSetForeground(display, gc,

        *str = "move mode";
        XDrawString(display, sub, gc, 20,
        flag &= ~FLAG_B1;
    }
    else if ((flag & FLAG_B1) != FLAG_B1)
    {
        flag |= FLAG_B1;
    }
}
/* button[2] */
if((xevent.xbutton.x >= button_x[2] && xevent.xbutton.x <=
&& (xevent.xbutton.y >= button_y[2] && xevent.xbutton.y <=
&& xevent.xany.window == sub)
{

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	21/36

```

XSetForeground(display,gc,WhitePixel(display,screen));

XFillRectangle(display,mwindow,gc,0,0,WIDTH,HEIGHT);

screen));

strlen(str[0]));

object[j].win, object[j].size_x, object[j].size_y);
object[j].win);

break;

/* Enter Window */

/* Leave Window */

/* MotionNotify */

root_y=%d\n", xevent.xbutton.x, xevent.xbutton.y, xevent.xbutton.x_root, xevent.xbutton.y_root);

default:
break;

}

}

}

/* 追跡モード */
if((flag & FLAG_B0) == FLAG_B0)
{
XSetForeground(display, gc, WhitePixel(display, screen));
XFillRectangle(display,mwindow, gc, 0, 0, WIDTH, HEIGHT);
}

/* タッチ情報の出力 */

for(j = 0; j < pointer_num; j++)
{

pointer[j].object_x = (int)(pointer[j].x/DEVX);
pointer[j].object_y = (int)(pointer[j].y/DEVY)-TOP;

if(pointer[j].x != 0 && pointer[j].y != 0)
{
printf("tracking[%d]:", j);

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	22/36

```

        cursor(display, mwindow, gc, set_color[j], pointer[j].object_x, pointer[j].object_y);
    }
    for(k = 0; k < object_num; k++)
    {
        /* object 移動判定:1 個目のタッチが枠内に入る */
        if((pointer[j].object_x >= object[k].x)
            && (pointer[j].object_y >= object[k].y)
            && (pointer[j].object_x <= object[k].x + object[k].size_x)
            && (pointer[j].object_y <= object[k].y + object[k].size_y)
            && xevent.xany.window != mwindow)
        {
            /* 2 点以降のタッチについて */
            for(l = j + 1; l < pointer_num; l++)
            {
                pointer[l].object_x = (int)(pointer[l].x/DEVX);
                pointer[l].object_y = (int)(pointer[l].y/DEVY)-TOP;

                printf("%d.catch=[%d],%t          %d.catch=[%d]¥n",
                    j,pointer[j].catch, l, pointer[l].catch);

                /* オブジェクトのサイズ変更 */
                /* 2 点目のタッチが同じ object の枠内に入る */
                if(!*(pointer[l].object_x >= object[pointer[j].catch].x)
                    && (pointer[l].object_y >= object[pointer[j].catch].y)
                    && (pointer[l].object_x <= object[pointer[j].catch].x +
                        object[pointer[j].catch].size_x)
                    && (pointer[l].object_y <= object[pointer[j].catch].y +
                        object[pointer[j].catch].size_y)
                    &&*!(pointer[j].catch == pointer[l].catch)
                    && pointer[j].catch >= 0 && pointer[l].catch >= 0))
                {
                    /* どちらがより(0,0)に近いか */
                    if (pointer[j].diff_x > pointer[l].diff_x)
                    {
                        object[pointer[j].catch].diff_x =
                            pointer[l].diff_x;
                    }
                    else
                    {
                        object[pointer[j].catch].diff_x =
                            pointer[j].diff_x;
                    }
                    if (pointer[j].diff_y > pointer[l].diff_y)
                    {
                        object[pointer[j].catch].diff_y =
                            pointer[l].diff_y;
                    }
                    else
                    {
                        object[pointer[j].catch].diff_y =
                            pointer[j].diff_y;
                    }
                    printf("diff%d,%d¥n",
                        object[pointer[j].catch].diff_x, object[pointer[j].catch].diff_y);

                    /* 最初に 2 点が同じ object に入ったとき */
                    if(pointer[j].lock == 0 && pointer[l].lock == 0)
                    {
                        pointer[j].lock = 1;
                        pointer[l].lock = 1;
                        /* 2 点間の初期距離 */
                        object[pointer[j].catch].first_diff_x =
                            pointer[j].diff_x - pointer[l].diff_x;
                        object[pointer[j].catch].first_diff_y =
                            pointer[j].diff_y - pointer[l].diff_y;
                    }
                    object[pointer[j].catch].size_x =
                        object[pointer[j].catch].now_x + object[pointer[j].catch].diff_x - object[pointer[j].catch].first_diff_x;
                    object[pointer[j].catch].size_y =
                        object[pointer[j].catch].now_y + object[pointer[j].catch].diff_y - object[pointer[j].catch].first_diff_y;

                    /* 最低値及び最高値 */

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	23/36

```

OBJECT_MIN_SIZE)
OBJECT_MIN_SIZE;
OBJECT_MAX_SIZE)
OBJECT_MAX_SIZE;

OBJECT_MIN_SIZE)
OBJECT_MIN_SIZE;
OBJECT_MAX_SIZE)
OBJECT_MAX_SIZE;

pointer[j].object_x)/2 - object[pointer[j].catch].size_x/2;
pointer[j].object_y)/2 - object[pointer[j].catch].size_y/2;

object[pointer[j].catch].size_x, object[pointer[j].catch].size_y);
object[pointer[j].catch].win, object[pointer[j].catch].x,
object[pointer[j].catch].size_y);

object[k].y);

pointer[j].diff_x;
pointer[j].diff_y;
object[pointer[j].catch].win, object[pointer[j].catch].x, object[pointer[j].catch].y);
object[pointer[j].catch].win);
//
//
//
//
}

if(object[pointer[j].catch].size_x <=
{
    object[pointer[j].catch].size_x =
}
else if(object[pointer[j].catch].size_x >=
{
    object[pointer[j].catch].size_x =
}

if(object[pointer[j].catch].size_y <=
{
    object[pointer[j].catch].size_y =
}
else if(object[pointer[j].catch].size_y >=
{
    object[pointer[j].catch].size_y =
}

/* 2点タッチ時のオブジェクト移動 */
object[pointer[j].catch].x = (pointer[j].object_x +
object[pointer[j].catch].y = (pointer[j].object_y +
/**/

printf("size_x[%d]          size_y[%d]",
        size_x, size_y);

XMoveResizeWindow(display,
object[pointer[j].catch].y, object[pointer[j].catch].size_x,
        size_y);

printf("Resize!%n");
}
/* 1点タッチでの移動 */
else if(pointer[j].catch < 0)
{
    /* 初タッチ時 */
    pointer[j].lock = 0;
    pointer[j].catch = k;
    pointer[j].diff_x = pointer[j].object_x - object[k].x;
    pointer[j].diff_y = pointer[j].object_y - object[k].y;
    object[k].x = pointer[j].object_x - pointer[j].diff_x;
    object[k].y = pointer[j].object_y - pointer[j].diff_y;
    XMoveWindow(display, object[k].win, object[k].x,
        object[k].y);

    XMapRaised(display, object[k].win);

    /* 初期サイズ */
    object[k].now_x = object[k].size_x;
    object[k].now_y = object[k].size_y;
}
else if(pointer[j].catch >= 0 && pointer[j].lock == 0)
{
    /* 2回目以降の動き */
    object[pointer[j].catch].x = pointer[j].object_x -
    object[pointer[j].catch].y = pointer[j].object_y -

    XMoveWindow(display,
    XMapRaised(display,

}
else
{
    pointer[j].lock = 0;
}
}

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	24/36





```

                                break;
                                }
                                printf(": value %d minimum %d maximum %d\n", absolute_feature[0], absolute_feature[1],
absolute_feature[2]);
                                }
                                }
}

```

*/\* タッチの出力 \*/*

```

static void corsor(Display* display, Window window, GC gc, unsigned long color, int x, int y)
{
    XSetForeground(display, gc, color);
    printf("x%d y%d\n", x, y);
    XFillArc(display, window, gc, x, y, corsor_size, corsor_size, 0, 360*64);
}

```

*/\* タッチの色設定 \*/*

```

void color_set(Display* display,int screen, unsigned long *set_color)
{
    Colormap cmap;
    XColor color, exact;

    cmap = DefaultColormap(display, screen);
    XAllocNamedColor(display, cmap, "black", &color, &exact);
    set_color[0] = color.pixel;

    XAllocNamedColor(display, cmap, "red", &color, &exact);
    set_color[1] = color.pixel;

    XAllocNamedColor(display, cmap, "blue", &color, &exact);
    set_color[2] = color.pixel;

    XAllocNamedColor(display, cmap, "green", &color, &exact);
    set_color[3] = color.pixel;

    XAllocNamedColor(display, cmap, "purple", &color, &exact);
    set_color[4] = color.pixel;

    XAllocNamedColor(display, cmap, "yellow", &color, &exact);
    set_color[5] = color.pixel;

    XAllocNamedColor(display, cmap, "pink", &color, &exact);
    set_color[6] = color.pixel;

    XAllocNamedColor(display, cmap, "orange", &color, &exact);
    set_color[7] = color.pixel;

    XAllocNamedColor(display, cmap, "gray", &color, &exact);
    set_color[8] = color.pixel;

    XAllocNamedColor(display, cmap, "brown", &color, &exact);
    set_color[9] = color.pixel;
}

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	26/36

## 5 マウスポインタとタッチポイントの位置が異なる現象について

(2016/03/11 追記)

RedHawk6.5.x のみマウスポインタの位置と、タッチスクリーンの位置が異なる問題が確認されました。

この原因は、Xserver と evdev のミスマッチにあり、RedHawk6.3 で使用している版にダウングレードすることで回避することを確認しました。





RedHawk の版	デフォルトインストール evdev 版	動作
6.0	xorg-x11-drv-evdev-2.3.2-8.el6.x86_64	未確認
6.3	xorg-x11-drv-evdev-2.6.0-2.el6.x86_64	○
6.5	xorg-x11-drv-evdev-2.7.3-5.el6.x86_64	△ xorg-x11-drv-evdev-2.6.0-5.el6.x86_64 をインストールする必要あり
7.0	xorg-x11-drv-evdev-2.8.2-5.el7.x86_64	○

下記手順で、ダウングレードしてご使用ください。

root でログイン後、init 1 ; を行うか、起動時の grub オプションで、1 を追加し、起動する。

```
rpm -e --nodeps xorg-x11-drv-evdev-2.7.3-5.el6
rpm -ivh xorg-x11-drv-evdev-2.6.0-5.el6.x86_64.rpm
```

なお、弊社で xorg-x11-drv-evdev-2.6.0 を RedHawk6.5 用に生成したバイナリ及びソース RPM を <https://www.ccur.co.jp/external/TechSup/SRPM/> からダウンロードいただけます。(本バージョンには、下記拡張メソッドが含まれています)

 <a href="#">xorg-x11-drv-evdev-2.6.0-5.el6.src.rpm</a>	11-Mar-2016 16:04 336K
 <a href="#">xorg-x11-drv-evdev-2.6.0-5.el6.x86_64.rpm</a>	11-Mar-2016 16:04 31K
 <a href="#">xorg-x11-drv-evdev-debuginfo-2.6.0-5.el6.x86_64.rpm</a>	11-Mar-2016 16:04 95K
 <a href="#">xorg-x11-drv-evdev-devel-2.6.0-5.el6.x86_64.rpm</a>	11-Mar-2016 16:04 12K

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	27/36

## 5.1 拡張メソッドについて

本 [xorg-x11-drv-evdev-2.6.0-5.el6.src.rpm](#) には、複数画面に対して、USB マウス/キーボードを固定的に割り付けた場合で、USB ポートの挿抜を行っても、再認識する機能を付加しています。

### ①xorg.conf の設定について

evdev を使用する下記の構成において

```
xorg0.conf
DISPLAY0+---KEYBOARD0 (/dev/input/event4)
      |
      +---MOUSE0 (/dev/input/event7)
```

```
xorg1.conf
DISPLAY1+---KEYBOARD1 (/dev/input/event5)
      |
      +---MOUSE1 (/dev/input/event6)
```

USB ポートを固定的に割り当て、互いのポートを認識しないようにするためには、下記の例のように InputClass を使用する。

```
##### xorg0.conf

Section "ServerLayout"
    Identifier      "Layout0"
    Screen          0 "Screen0" 0 0
    #Option         "AutoEnableDevices"      "false"
EndSection

Section "Files"
    FontPath        "/usr/share/fonts/default/Type1"
EndSection

Section "InputClass"
    Identifier      "ignore_other_devices"
    #MatchTag       "garbage"
    MatchDevicePath "/dev/input/event[2356]"
    Option          "Ignore"      "yes"
EndSection

Section "InputClass"
    Identifier      "Device0"
    MatchTag        "Display_0_Devices"
    Driver          "evdev"
EndSection

Section "InputClass"
    Identifier      "KEYBOARD0"
    MatchDevicePath "/dev/input/event4"
    Option          "Ignore"      "no"
    Driver          "evdev"
    Option "GrabDevice" "on"
```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	28/36

```

EndSection

Section "InputClass"
    Identifier "MOUSE0"
    MatchDevicePath "/dev/input/event7"
    Option      "Ignore"      "no"
    Driver      "evdev"
    Option "GrabDevice" "on"
EndSection

Section "Monitor
    Identifier      "Monitor0"
    VendorName      "Unknown"
    ModelName       "Unknown"
    Option          "DPMS"
EndSection

Section "Device"
    Identifier      "Device0"
    Driver          "mga"
    #Driver         "vesa"
    VendorName      "Matrox Graphics"
    BusID           "PCI:01:01:0"
    Screen          0
EndSection

Section "Screen"
    Identifier      "Screen0"
    Device          "Device0"
    Monitor         "Monitor0"
    DefaultDepth    24
    SubSection      "Display"
        Depth       24
    EndSubSection
EndSection

#####xorg1.conf
Section "ServerLayout"
    Identifier      "Layout1"
    Screen          0 "Screen1"
    #Option         "AutoEnableDevices"      "false"
EndSection

Section "Files"
    FontPath        "/usr/share/fonts/default/Type1"
EndSection

Section "InputClass"
    Identifier "ignore_other_devices"
    #MatchTag   "garbage"
    MatchDevicePath "/dev/input/event[2347]"
    Option "Ignore"      "yes"
EndSection

Section "InputClass"
    Identifier      "DEVICE1"
    MatchTag        "Display_1_Devices"
    Option          "Ignore"      "no"
    Driver          "evdev"
EndSection

Section "InputClass"
    Identifier      "KEYBOARD1"
    MatchDevicePath "/dev/input/event5"
    Option          "Ignore"      "no"
    Driver          "evdev"
    Option "GrabDevice" "on"
EndSection

Section "InputClass"
    Identifier      "MOUSE1"
    MatchDevicePath "/dev/input/event6"
    Option          "Ignore"      "no"
    Driver          "evdev"

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	29/36

```
Option "GrabDevice" "on"
EndSection
```

```
Section "Monitor"
    Identifier      "Monitor1"
    VendorName     "Unknown"
    ModelName      "Unknown"
    HorizSync      28.0 - 33.0
    VertRefresh    43.0 - 72.0
    Option         "DPMS"
EndSection
```

```
Section "Device"
    Identifier      "Device1"
    Driver          "nvidia"
    #Driver         "nv"
    VendorName     "NVIDIA Corporation"
    BusID          "PCI:129:0:0"
    Option "CrtcNumber" "1"
EndSection
```

```
Section "Screen"
    Identifier      "Screen1"
    Device          "Device1"
    Monitor         "Monitor1"
    DefaultDepth   24
    SubSection     "Display"
        Depth      24
    EndSubSection
EndSection
```

## ②オリジナル evdev\_drv. so について

弊社での調査の結果、現行の evdev\_drv. so では、USB ポートの抜挿を行った場合、evdev デバイスドライバの read() 関数で ERRNO=19 が発生し、その結果ポートを close() しています。

その後、xinput コマンドによる、デバイスの float, reattach を行っても、evdev デバイスドライバの再組み込みシーケンスが動作することはありません。

USB ポートの再構成を行う唯一の方法は、X サーバーの再起動のみです。

### ③解決方法

以下のように、“InputClass”中に Option として

```
Option "GrabDevice" "on"
```

```
Option "UseSIGIO" "true"
```

を定義すると、FixedPort の再アタッチが可能になります。

```
Section "ServerLayout"
    Identifier      "Layout1"
    Screen          0 "Screen1"
    #Option         "AutoEnableDevices" "false"
EndSection

Section "Files"
    FontPath        "/usr/share/fonts/default/Type1"
EndSection

Section "InputClass"
    Identifier      "ignore_other_devices"
    #MatchTag       "garbage"
    MatchDevicePath "/dev/input/event[2347]"
    Option "Ignore" "yes"
EndSection

Section "InputClass"
    Identifier      "KEYBOARD1"
    MatchDevicePath "/dev/input/event5"
    Option          "Ignore" "no"
    Driver          "evdev"
    Option "GrabDevice" "on"
    Option "UseSIGIO" "true"
EndSection

Section "InputClass"
    Identifier      "MOUSE1"
    MatchDevicePath "/dev/input/event6"
    Option          "Ignore" "no"
    Driver          "evdev"
    Option "GrabDevice" "on"
    Option "UseSIGIO" "true"
EndSection
```

この時、GrabDevice を off で利用すると、通常の evdev と同様の動作になり、GrabDevice を on で再アタッチを繰り返しますが、UseSIGIO を true に設定していない場合には、タイマーを生成しません。

## 6 キャリブレーションについて

RedHawk6.5 におけるキャリブレーションは、自動的には行われなため、以下の手順で行います。また、キャリブレーションには、`xinput_calibrator` をインストールして使用します。

この例で使用したソースコードを以下の URL を <https://www.ccur.co.jp/external/TechSup/SRPM/> からダウンロードいただけます。

展開後、README に記載の手順でコンパイル、インストールしてください。

### 6.1 認識したデバイス名と ID を表示。

Xinput のデバイス名として、"Advanced Silicon S.A. CoolTouch(TM) System"を使用します。

(ID である 9 でも使用可能ですが、常に同じ ID とは限りませんので、曖昧さを回避するために、名称を使用します)

```
# xinput
```

```

{ Virtual core pointer          id=2      [master pointer (3)]
  ↳ Virtual core XTEST pointer  id=4      [slave pointer (2)]
  ↳ Advanced Silicon S.A. CoolTouch(TM) System id=9 [slave pointer (2)]
  ↳ Logitech USB Optical Mouse  id=10     [slave pointer (2)]
[ Virtual core keyboard        id=3      [master keyboard (2)]
  ↳ Virtual core XTEST keyboard  id=5      [slave keyboard (3)]
  ↳ Dell Dell USB Keyboard        id=6      [slave keyboard (3)]
  ↳ Power Button                  id=7      [slave keyboard (3)]
  ↳ Power Button                  id=8      [slave keyboard (3)]

```

### 6.2 タッチスクリーンのプロパティの表示

キャリブレーションデータは、入力されていません。

```
# xinput --list-props "Advanced Silicon S.A. CoolTouch(TM) System"
```

```

Device 'Advanced Silicon S.A. CoolTouch(TM) System':
  Device Enabled (139): 1
  Coordinate Transformation Matrix (141): 1.000000, 0.000000, 0.000000, 0.000000, 1.000000, 0.000000,
0.000000, 0.000000, 1.000000
  Device Accel Profile (267): 0
  Device Accel Constant Deceleration (268): 1.000000
  Device Accel Adaptive Deceleration (269): 1.000000
  Device Accel Velocity Scaling (270): 10.000000
  Evdev Axis Inversion (271): 0, 0
  Evdev Axis Calibration (272): <no items>
  Evdev Axes Swap (273): 0
  Axis Labels (274): "Abs X" (261), "Abs Y" (262), "Abs Misc" (263), "Abs MT Position X" (264), "Abs MT Position Y"
(265), "Abs MT Tracking ID" (266)
  Button Labels (275): "Button Unknown" (260), "Button Unknown" (260), "Button Unknown" (260), "Button Wheel Up"
(145), "Button Wheel Down" (146)
  Evdev Middle Button Emulation (276): 0
  Evdev Middle Button Timeout (277):50
  Evdev Wheel Emulation (278): 0
  Evdev Wheel Emulation Axes (279): 0, 0, 4, 5
  Evdev Wheel Emulation Inertia (280): 10
  Evdev Wheel Emulation Timeout (281): 200
  Evdev Wheel Emulation Button (282): 4
  Evdev Drag Lock Buttons (283): 0

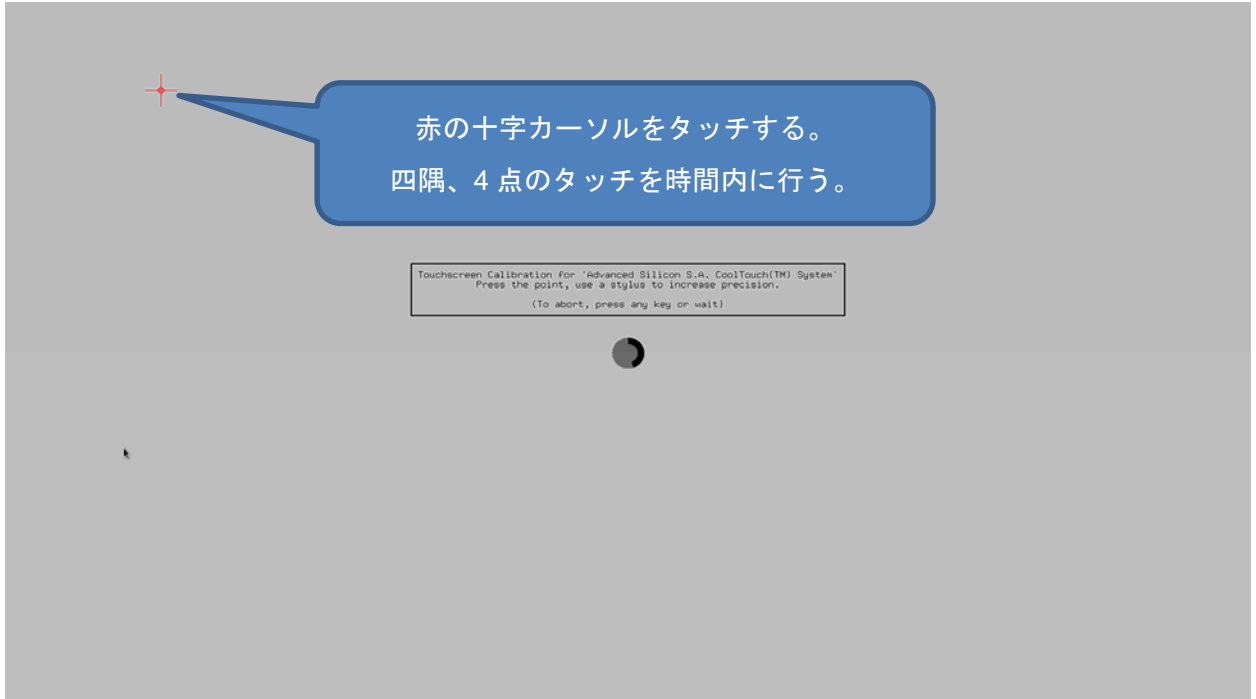
```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	32/36



### 6.3 キャリブレーション

```
# xinput_calibrator --device "Advanced Silicon S.A. CoolTouch(TM) System"
```



### 6.4 キャリブレーション値の確認

```
# xinput --list-props "Advanced Silicon S.A. CoolTouch(TM) System"
Device 'Advanced Silicon S.A. CoolTouch(TM) System':
    Device Enabled (139): 1
    Coordinate Transformation Matrix (141): 1.000000, 0.000000, 0.000000, 0.000000, 1.000000, 0.000000,
0.000000, 0.000000, 1.000000
    Device Accel Profile (267): 0
    Device Accel Constant Deceleration (268): 1.000000
    Device Accel Adaptive Deceleration (269): 1.000000
    Device Accel Velocity Scaling (270): 10.000000
    Evdev Axis Inversion (271): 0, 0
    Evdev Axis Calibration (272): 22, 32618, 248, 32853
    Evdev Axes Swap (273): 0
    Axis Labels (274): "Abs X" (261), "Abs Y" (262), "Abs Misc" (263), "Abs MT Position X" (264), "Abs MT Position Y"
(265), "Abs MT Tracking ID" (266)
    Button Labels (275): "Button Unknown" (260), "Button Unknown" (260), "Button Unknown" (260), "Button Wheel Up"
(145), "Button Wheel Down" (146)
    Evdev Middle Button Emulation (276): 0
    Evdev Middle Button Timeout (277): 50
    Evdev Wheel Emulation (278): 0
    Evdev Wheel Emulation Axes (279): 0, 0, 4, 5
    Evdev Wheel Emulation Inertia (280): 10
    Evdev Wheel Emulation Timeout (281): 200
    Evdev Wheel Emulation Button (282): 4
    Evdev Drag Lock Buttons (283): 0
```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	33/36

## 6.5 キャリブレーション値の設定

キャリブレーションの値は、再起動するとリセットされるため、xinput コマンドを/etc/rc.local などの起動時ファイルで設定するか、/etc/X11/xorg.conf の evdev の Option “Catibration” “min-x max-x min-y max-y” で行います。

```
/usr/bin/xinput --set-prop "Advanced Silicon S.A. CoolTouch(TM) System" "Evdev Axis Calibration" 22 32618 248 32853
```

/etc/X11/xorg.conf への記述は、以下の例のように行います。

### 6.5.1 デバイス名称の確認

xorg.conf で使用する input デバイス名称を以下のコマンドで確認します。

以下の例では、/dev/input/mouse1 か、/dev/input/event5 を使用します。

```
# cat /proc/bus/input/devices |grep "Name\\|Handlers"
N: Name="Power Button"
H: Handlers=kbd event0
N: Name="Power Button"
H: Handlers=kbd event1
N: Name="Logitech USB Optical Mouse"
H: Handlers=mouse0 event2
N: Name="Dell Dell USB Keyboard"
H: Handlers=sysrq kbd event3
N: Name="HDA Digital PCBeep"
H: Handlers=kbd event4
N: Name="Advanced Silicon S.A. CoolTouch(TM) System"
H: Handlers=mouse1 event5
N: Name="PC Speaker"
H: Handlers=kbd event6
```

### 6.5.2 /etc/X11/xorg.conf への記述

evdev は、すべてのポインティングデバイスを同時に管理していますので、個別に値を設定するためには、以下の例のように一旦 event5 を無視させておき、改めて個別設定を行います。

```
Section "ServerLayout"
    Identifier      "Layout0"
    Screen          0  "Screen0"
    InputDevice     "Keyboard0" "CoreKeyboard"
    InputDevice     "Mouse0" "CorePointer"
EndSection

Section "Files"
    FontPath        "/usr/share/fonts/default/Type1"
EndSection

Section "InputClass"
    Identifier      "IgnoreDevices"
    MatchDevicePath "/dev/input/event5"
    Option          "Ignore"          "yes"
EndSection

Section "InputClass"
    Identifier      "TouchScreen"
    MatchDevicePath "/dev/input/event5"
    Option          "Ignore"          "no"
    Driver          "evdev"
Option "Calibration" "22 32675 105 32891"
EndSection
```

表題	RedHawk6.5 マルチタッチパネルについて	No	版 2
			ページ 34/36

```

Section "InputDevice"
    # generated from default
    Identifier "Mouse0"
    Driver "mouse"
    Option "Protocol" "auto"
    Option "Device" "/dev/input/mice"
    Option "Emulate3Buttons" "no"
    Option "ZAxisMapping" "4 5"
EndSection

Section "InputDevice"
    # generated from data in "/etc/sysconfig/keyboard"
    Identifier "Keyboard0"
    Driver "kbd"
    Option "XkbLayout" "jp"
    Option "XkbModel" "jp106"
EndSection

Section "Monitor"
    Identifier "Monitor0"
    VendorName "Unknown"
    ModelName "Unknown"
    HorizSync 28.0 - 33.0
    VertRefresh 43.0 - 72.0
    Option "DPMS"
EndSection

Section "Device"
    Identifier "Device0"
    Driver "nvidia"
    VendorName "NVIDIA Corporation"
EndSection

Section "Screen"
    Identifier "Screen0"
    Device "Device0"
    Monitor "Monitor0"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
    EndSubSection
EndSection

```

### 6.5.3 /var/log/Xorg.0.log

```

[ 5906.888] (II) config/hal: Adding input device Dell Dell USB Keyboard
[ 5906.888] (II) LoadModule: "evdev"
[ 5906.888] (II) Loading /usr/lib64/xorg/modules/input/evdev_drv.so
[ 5906.888] (II) Module evdev: vendor="X.Org Foundation"
[ 5906.888]         compiled for 1.13.0, module version = 2.6.0
[ 5906.888]         Module class: X.Org XInput Driver
[ 5906.888]         ABI class: X.Org XInput driver, version 18.1
[ 5906.888] (II) Using input driver 'evdev' for 'Dell Dell USB Keyboard'
[ 5906.888] (**) Dell Dell USB Keyboard: always reports core events
[ 5906.888] (**) Dell Dell USB Keyboard: Device: "/dev/input/event3"
[ 5906.888] (-- ) Dell Dell USB Keyboard: Found keys
[ 5906.888] (II) Dell Dell USB Keyboard: Configuring as keyboard
[ 5906.888] (**) Option "config_info" "hal:/org/freedesktop/Hal/devices/usb_device_413c_2105_noserial_if0_logicaldev_input"
[ 5906.888] (II) XINPUT: Adding extended input device "Dell Dell USB Keyboard" (type: KEYBOARD, id 6)
[ 5906.888] (**) Option "xkb_rules" "evdev"
[ 5906.888] (**) Option "xkb_model" "jp106"
[ 5906.888] (**) Option "xkb_layout" "jp"
[ 5906.888] (WW) Option "xkb_variant" requires an string value
[ 5906.888] (WW) Option "XkbVariant" requires an string value
[ 5906.888] (**) Option "xkb_options" "terminate:ctrl_alt_bksp"
[ 5906.906] (II) config/hal: Adding input device Power Button
[ 5906.906] (II) Using input driver 'evdev' for 'Power Button'
[ 5906.906] (**) Power Button: always reports core events
[ 5906.906] (**) Power Button: Device: "/dev/input/event0"
[ 5906.906] (-- ) Power Button: Found keys
[ 5906.906] (II) Power Button: Configuring as keyboard
[ 5906.906] (**) Option "config_info" "hal:/org/freedesktop/Hal/devices/computer_logicaldev_input_0"

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	35/36

```

[ 5906.906] (II) XINPUT: Adding extended input device "Power Button" (type: KEYBOARD, id 7)
[ 5906.906] (**) Option "xkb_rules" "evdev"
[ 5906.906] (**) Option "xkb_model" "jp106"
[ 5906.906] (**) Option "xkb_layout" "jp"
[ 5906.906] (WW) Option "xkb_variant" requires a string value
[ 5906.906] (WW) Option "XkbVariant" requires a string value
[ 5906.906] (**) Option "xkb_options" "terminate:ctrl_alt_bksp"
[ 5906.910] (II) config/hal: Adding input device Power Button
[ 5906.910] (II) Using input driver 'evdev' for 'Power Button'
[ 5906.910] (**) Power Button: always reports core events
[ 5906.910] (**) Power Button: Device: "/dev/input/event1"
[ 5906.910] (-) Power Button: Found keys
[ 5906.910] (II) Power Button: Configuring as keyboard
[ 5906.910] (**) Option "config_info" "hal:/org/freedesktop/Hal/devices/computer_logicaldev_input"
[ 5906.910] (II) XINPUT: Adding extended input device "Power Button" (type: KEYBOARD, id 8)
[ 5906.910] (**) Option "xkb_rules" "evdev"
[ 5906.910] (**) Option "xkb_model" "jp106"
[ 5906.910] (**) Option "xkb_layout" "jp"
[ 5906.910] (WW) Option "xkb_variant" requires a string value
[ 5906.910] (WW) Option "XkbVariant" requires a string value
[ 5906.910] (**) Option "xkb_options" "terminate:ctrl_alt_bksp"
[ 5906.912] (II) config/hal: Adding input device Advanced Silicon S.A. CoolTouch(TM) System
[ 5906.912] (**) Advanced Silicon S.A. CoolTouch(TM) System: Applying InputClass "IgnoreDevices"
[ 5906.912] (**) Advanced Silicon S.A. CoolTouch(TM) System: Applying InputClass "TouchScreen"
[ 5906.912] (II) Using input driver 'evdev' for 'Advanced Silicon S.A. CoolTouch(TM) System'
[ 5906.912] (**) Advanced Silicon S.A. CoolTouch(TM) System: always reports core events
[ 5906.912] (**) Advanced Silicon S.A. CoolTouch(TM) System: Device: "/dev/input/event5"
[ 5906.922] (-) Advanced Silicon S.A. CoolTouch(TM) System: Found absolute axes
[ 5906.922] (-) Advanced Silicon S.A. CoolTouch(TM) System: Found x and y absolute axes
[ 5906.922] (-) Advanced Silicon S.A. CoolTouch(TM) System: Found absolute touchscreen
[ 5906.922] (II) Advanced Silicon S.A. CoolTouch(TM) System: Configuring as touchscreen
[ 5906.922] (**) Advanced Silicon S.A. CoolTouch(TM) System: YAxisMapping: buttons 4 and 5
[ 5906.922] (**) Advanced Silicon S.A. CoolTouch(TM) System: EmulateWheelButton: 4, EmulateWheelInertia: 10,
EmulateWheelTimeout: 200
[ 5906.922] (**) Option "config_info" "hal:/org/freedesktop/Hal/devices/usb_device_2149_2122_noserial_if0_logicaldev_input"
[ 5906.922] (II) XINPUT: Adding extended input device "Advanced Silicon S.A. CoolTouch(TM) System" (type: TOUCHSCREEN, id
9)
[ 5906.923] (II) Advanced Silicon S.A. CoolTouch(TM) System: initialized for absolute axes.
[ 5906.923] (**) Advanced Silicon S.A. CoolTouch(TM) System: (accel) keeping acceleration scheme 1
[ 5906.923] (**) Advanced Silicon S.A. CoolTouch(TM) System: (accel) acceleration profile 0
[ 5906.923] (**) Advanced Silicon S.A. CoolTouch(TM) System: (accel) acceleration factor: 2.000
[ 5906.923] (**) Advanced Silicon S.A. CoolTouch(TM) System: (accel) acceleration threshold: 4
[ 5906.925] (II) config/hal: Adding input device Logitech USB Optical Mouse
[ 5906.925] (II) Using input driver 'evdev' for 'Logitech USB Optical Mouse'
[ 5906.925] (**) Logitech USB Optical Mouse: always reports core events
[ 5906.925] (**) Logitech USB Optical Mouse: Device: "/dev/input/event2"
[ 5906.925] (-) Logitech USB Optical Mouse: Found 3 mouse buttons
[ 5906.925] (-) Logitech USB Optical Mouse: Found scroll wheel(s)
[ 5906.925] (-) Logitech USB Optical Mouse: Found relative axes
[ 5906.925] (-) Logitech USB Optical Mouse: Found x and y relative axes
[ 5906.925] (II) Logitech USB Optical Mouse: Configuring as mouse
[ 5906.925] (II) Logitech USB Optical Mouse: Adding scrollwheel support
[ 5906.925] (**) Logitech USB Optical Mouse: YAxisMapping: buttons 4 and 5
[ 5906.925] (**) Logitech USB Optical Mouse: EmulateWheelButton: 4, EmulateWheelInertia: 10, EmulateWheelTimeout: 200
[ 5906.925] (**) Option "config_info" "hal:/org/freedesktop/Hal/devices/usb_device_46d_c05a_noserial_if0_logicaldev_input"
[ 5906.925] (II) XINPUT: Adding extended input device "Logitech USB Optical Mouse" (type: MOUSE, id 10)
[ 5906.925] (II) Logitech USB Optical Mouse: initialized for relative axes.
[ 5906.925] (**) Logitech USB Optical Mouse: (accel) keeping acceleration scheme 1
[ 5906.925] (**) Logitech USB Optical Mouse: (accel) acceleration profile 0
[ 5906.925] (**) Logitech USB Optical Mouse: (accel) acceleration factor: 2.000
[ 5906.925] (**) Logitech USB Optical Mouse: (accel) acceleration threshold: 4

```

表題	RedHawk6.5 マルチタッチパネルについて	No	版	2
			ページ	36/36