

---

# Linux Multitouch Serial Mode Driver v1.0

## 取扱説明書

株式会社ディ・エム・シー

---

### —目次—

1. はじめに	2
2. システム要件	2
3. 提供ファイル一覧	2
4. マルチタッチドライバの概要	2
5. カーネル用マルチタッチドライバのビルド	2
6. "X"用マルチタッチドライバのビルド	5
7. マルチタッチプロトコルの動作確認	8
8. マルチタッチの動作確認	10
9. 対応"VID" 及び"PID"	11

この取扱説明書で提供するソフトウェアは、使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができます。

### 著作権

Copyright © 2013 DMC Co., Ltd

## 1. はじめに

本書は、”DUS”コントローラ向けの”Linux Multitouch Serial Mode Driver”のユーザーマニュアルです。

## 2. システム要件

一般的な環境(x86)での説明になります。テストでは”x86”(=AT 互換機)環境でのカーネル Ubuntu10.10(”v2.6.37”)でのコンパイル、動作確認しています。カーネルがマルチタッチサポートした”v2.6.37”以降でも有効です。また、下記に記載されているコマンドやインストール等の例は、カーネルやディストリビューションのバージョンにより異なる場合があります。

## 3. 提供ファイル一覧

ファイル名	概要
”hid-unitec.c”	カーネル追加モジュールソースコード
”xf86-input-mtrack-unitec.tar.gz”	”X”用マルチタッチドライバのソースコード
”10-unitec-serialmode.conf”	”X”用マルチタッチドライバの設定ファイル

## 4. マルチタッチドライバの概要

### 4.1. カーネル用マルチタッチドライバ(”hid-unitec”)

”hid-unitec”ドライバは”Digitizer Drivers for Windows Touch and Pen-Based Computers” ([http://www.microsoft.com/whdc/device/input/DigitizerDrvs\\_touch.mspx](http://www.microsoft.com/whdc/device/input/DigitizerDrvs_touch.mspx)) においてレポートモードとしてシリアルモードを、また、”Linux”カーネルでのマルチタッチプロトコルとしてタイプ”A”をサポートしたドライバです。

### 4.2. X 用マルチタッチドライバ(”xf86-input-mtrack”)

”xf86-input-mtrack”ドライバはマルチタッチプロトコルを入力デバイスのボタンイベントに置き換えて”X”サーバーに渡すドライバです。  
(<https://github.com/BlueDragonX/xf86-inputmtrack>)デフォルトではポインタはタッチパッドの動作を行いますが、タッチスクリーンの動作(絶対座標と、タッチ時のボタンプレスイベントの発行)を行うようにしています。

## 5. カーネル用マルチタッチドライバのビルド

### 5.1. マルチタッチドライバ”hid-unitec”のビルド

ドライバのビルドを行うに際して現在ご使用になられている、”Linux”のカーネルをカーネルソースパッケージ等から入手し、パッチ等の作業が必要になります。手順は以下のようになっています。

### 5.1.1.カーネルソースの確認

カーネルソースパッケージ等から入手したカーネルが正常に動作するか、ドライバのパッチ作業前にカーネルをコンパイルして動作確認をしておいてください。

コンパイル例

```
# cd <kernel-src-dir>
# sudo vi Makefile
# sudo make bzImage modules
# sudo make install modules_install
```

注、パッケージを作ったの動作確認でも可能です。

### 5.1.2.パッチ作業

以下の用に”hid”系ソースへのパッチをあてます。

“<kernel src dir>/drivers/hid/Kconfig”の”endmenu”の前に以下を追加

```
config HID_UNITEC
    tristate "UNITEC USB Touch"
    depends on USB_HID
    ---help---
    Support for UNITEC USB Touch.
```

“<kernel src dir>/drivers/hid/Makefile”の適切な位置に以下を追加

```
obj-$(CONFIG_HID_UNITEC) += hid-unitec.o
```

“<kernel src dir>/drivers/hid/hid-core.c”の”hid\_blacklist”テーブルの適切な位置に以下を追加

```
{HID_USB_DEVICE(USB_VENDOR_ID_DMC,USB_DEVICE_ID_DMC_USB_TOUCH_07d2)},
```

“<kernel src dir>/drivers/hid/hid-ids.h”の適切な位置に以下を追加

```
#define USB_VENDOR_ID_DMC                0x0afa
#define USB_DEVICE_ID_DMC_USB_TOUCH_07d2 0x07d2
```

### 5.1.3.”DUS”コントローラ用カーネル追加モジュールソース導入

“<kernel src dir>/drivers/hid”に”hid-unitec.c”を入れる。

#### 5.1.4.カーネルコンフィギュレーション

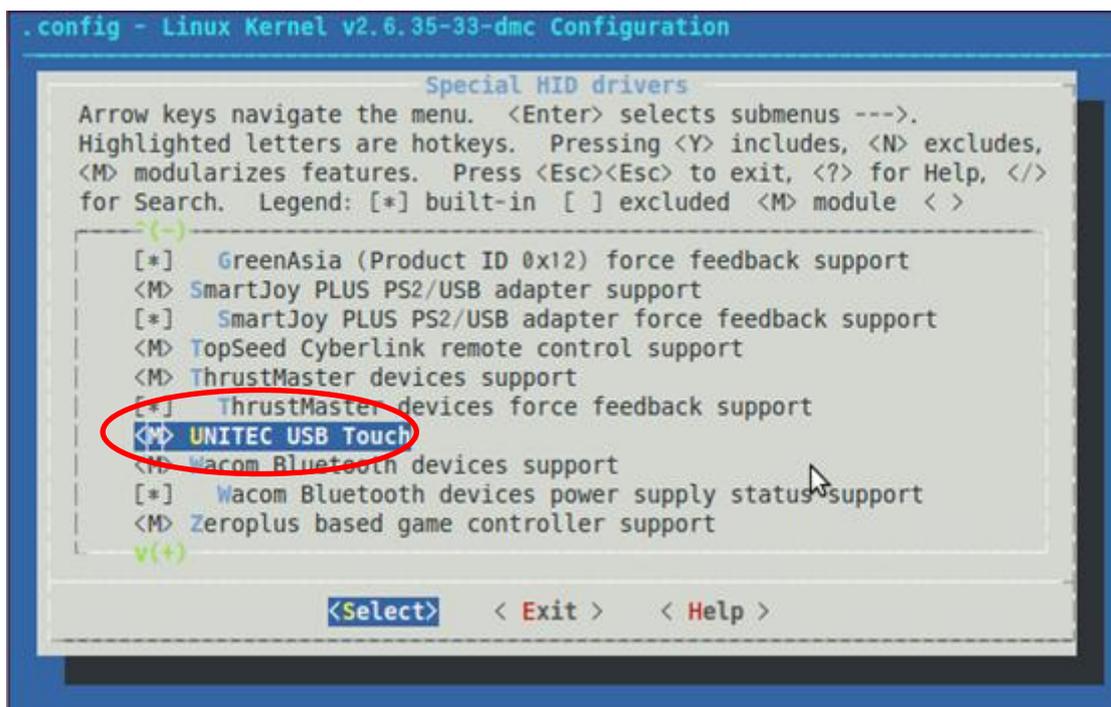
カーネルコンフィギュレーションでは CONFIG\_HID\_UNITEC を選択(<M>指定)してください。

「Linux Kernel Configuration」 (Top Menu)

- 「Device Drivers」
- 「HID Devices」
- 「Special HID drivers」
- 「UNITEC USB Touch 」 (=CONFIG\_HID\_UNITEC)

コンフィグ例

```
# sudo make menuconfig
```



5.1.5.パッチ後のカーネルをコンパイルして動作確認してください。

コンパイル例

```
# sudo make bzImage modules  
# sudo make install modules_install
```

注、パッケージを作ってインストール方法でも可能です。

## 6. “X”用マルチタッチドライバのビルド

### 6.1. マルチタッチドライバ(“xf86-input-mtrack”)のビルド

“xf86-input-mtrack”をビルドするために以下が必要になります。

```
xserver-xorg-dev
xorg-macros
mtdev
```

なお、“xserver-xorg-dev”と“xorg-macros”は“xf86-input-mtrack”のインストール後は不要になります。

#### 6.1.1. “xserver-xorg-dev”

初めに、“xserver-xorg-dev”が依存する“libpciaccess-dev”、“libxkbfile-dev”、“x11proto-dri2-dev”、“x11proto-fonts-dev”をインストールします。

インストール例

```
# sudo apt-get install libpciaccess-dev libxkbfile-dev x11proto-dri2-dev x11proto-fonts-dev
```

“xserver-xorg-dev”も同様にインストールします。

インストール例

```
# sudo apt-get install xserver-xorg-dev
```

もし、必要であれば“xf86-input-mtrack”のインストール後にアンインストールして下さい。

アンインストール例

```
# sudo apt-get purge xserver-xorg-dev libpciaccess-dev libxkbfile-dev
x11proto-dri2-dev x11proto-fonts-dev
```

#### 6.1.2. “xorg-macros”

“xorg-macros-1.8”以降が必要になりますが、パッケージでインストールできるものはバージョンが古い場合があります、その場合オリジナルの“xorg-macros-1.8”以上をダウンロードして手動でインストール必要があります。なお、“xorg-macros”は“util-macros”に含まれています。

インストール例

```
# wget http://xorg.freedesktop.org/releases/individual/util/util-macros-1.8.0.tar.bz2
# tar xpfj util-macros-1.8.0.tar.bz2
# cd util-macros-1.8.0/
# ./configure --prefix=/usr
```

```
# make
# sudo make install
```

もし、必要であれば”xf86-input-mtrack”のインストール後にアンインストールしてください。

アンインストール例

```
# sudo make uninstall
```

### 6.1.3. “mtdev”

ソースコードをダウンロードして手動でインストールして下さい。

手動インストール例

```
# wget http://bitmath.org/code/mtdev/mtdev-1.1.0.tar.bz2
# ar xpf mtdev-1.1.0.tar.bz2
# cd mtdev-1.1.0/
# ./configure --prefix=/usr
# make
# sudo make install
```

### 6.1.4.”xf86-input-mtrack”

“xf86-input-mtrack”は提供ファイルに含まれるものを使用します。

インストール例

```
# tar xpfzf xf86-input-mtrack-unitec.tar.gz
# cd mtrack-unitec/
# autoreconf -fi
# ./configure --prefix=/usr
# make
# sudo make install
```

## 6.2. マルチタッチドライバ(“xf86-input-mtrack”)設定

設定ファイルを所定の場所に置きます。

設定ファイルインストール例

```
# sudo cp 10-unitec-serialmode.conf /usr/share/X11/xorg.conf.d
```

この設定ファイルを用意することで、”X”サーバーは条件に一致するデバイスを認識した際、”xf86-input-mtrack”ドライバをロードします(ドライバ名は”mtrack”)。なお、設定ファイルのインストール場所はカーネルや、ディストリビューションのバージョンで異なる

場合があります。

### 6.3.”X”サーバーの再起動

マルチタッチドライバを有効にするために、”X”サーバーを起動します。X サーバーのみを再起動させることも可能ですが、ここでは念のため、”Linux”自体を再起動させます。

再起動例

```
# sudo reboot
```

### 6.4 マルチタッチドライバ(”xf86-input-mtrack”)のロード

デバイスを認識すると、X サーバーはドライバをロードし、ログファイル(”/var/log/Xorg.0.log”)に以下のようなログを出力します。

ログ出力例

```
(II) config/udev: Adding input device UNITEC USB Touch (WinXP&7)
(/dev/input/mouse2)
(II) No input driver/identifier specified (ignoring)
(II) config/udev: Adding input device UNITEC USB Touch (WinXP&7)
(/dev/input/event7)
(**) UNITEC USB Touch (WinXP&7): Applying InputClass "evdev touchscreen
catchall"
(**) UNITEC USB Touch (WinXP&7): Applying InputClass "UNITEC USB Touch"
(II) LoadModule: "mtrack"
(II) Loading /usr/lib/xorg/modules/input/mtrack_drv.so
(II) Module mtrack: vendor="X.Org Foundation"
        compiled for 1.7.6, module version = 0.1.0
        Module class: X.Org XInput Driver
        ABI class: X.Org XInput driver, version 7.0
(**) UNITEC USB Touch (WinXP&7): always reports core events
(**) Option "MaxTapTime" "400"
(II) XINPUT: Adding extended input device "UNITEC USB Touch (WinXP&7)"
(type: TOUCHSCREEN)
(II) device control: init
(**) Option "Device" "/dev/input/event7"
(II) mtrack: devname: UNITEC USB Touch (WinXP&7)
(II) mtrack: devid: afa 7d2 111
(II) mtrack: caps: mtdata
(II) mtrack: 5: min: 0 max: 9983
```

```
(II) mtrack: 6: min: 0 max: 5631
(II) mtrack: 9: min: 0 max: 65535
(**) UNITEC USB Touch (WinXP&7): (accel) keeping acceleration scheme 1
(**) UNITEC USB Touch (WinXP&7): (accel) acceleration profile 0
(**) UNITEC USB Touch (WinXP&7): (accel) acceleration factor: 2.000
(**) UNITEC USB Touch (WinXP&7): (accel) acceleration threshold: 4
(II) device control: on
(WW) Touchpad has minimal capabilities. Some features will be unavailable.
```

また、デバイスが切り離されたことを認識すると、X サーバーはドライバをアンロードし、ログファイルに以下のようなログを出力します。

ログ出力例

```
(II) config/udev: removing device UNITEC USB Touch (WinXP&7)
(II) device control: off
(II) device control: close
(II) UnloadModule: "mtrack"
```

## 7. マルチタッチプロトコルの動作確認

“Linux”カーネルとユーザーランド間のマルチタッチプロトコルの動作確認に“evtest”を用いることができます。

### 7.1. “evtest”のビルド

パッケージでインストールできるものはバージョンが古い場合があります、その場合マルチタッチのイベントを識別できないため、手動で最新バージョンをビルドする必要があります。

手動インストール例

```
# Wget
http://pkgs.fedoraproject.org/repo/pkgs/evtest/evtest-1.29.tar.bz2/1cc914cab3c30fa1e8
b62005684e8ef3/evtest-1.29.tar.bz2
# tar xpf evtest-1.29.tar.bz2
# cd evtest-1.29/
# autoreconf -i
# ./configure --prefix=/usr
# make
# sudo make install
```

## 7.2.“evtest”の起動

マルチタッチコントローラを正常に認識すると、“/dev/input/evtestX”が生成されます(“X”が何番になるかは環境に依存します。“X”サーバーのログや、/sys/class/input/eventX/device/name”等を見ることで、該当するデバイスの番号が何番になっているか確認することができます)。これを引数として“evtest”を実行してください。

実行例

```
# sudo evtest /dev/input/evtestX
```

実行すると、以下の画面が表示され、入力待ちとなります。

表示例

```
Input driver version is 1.0.0
Input device ID: bus 0x3 vendor 0xafa product 0x7d2 version 0x111
Input device name: "UNITEC USB Touch (WinXP&7)"
Supported events:
Event type 0 (EV_SYN)
Event type 1 (EV_KEY)
Event code 330 (BTN_TOUCH)
Event type 3 (EV_ABS)
Event code 0 (ABS_X)
Value 0
Min 0
Max 9983
Event code 1 (ABS_Y)
Value 0
Min 0
Max 5631
Event code 53 (ABS_MT_POSITION_X)
Value 0
Min 0
Max 9983
Event code 54 (ABS_MT_POSITION_Y)
Value 0
Min 0
Max 5631
Event code 57 (ABS_MT_TRACKING_ID)
Value 0
```

```
Min 0
Max 65535
Testing ... (interrupt to exit)
```

ここでタッチパネルにタッチすることで、以下のようなカーネルから送られるプロトコルデータが表示されます。

プロトコルデータ例

```
Event: time 527.233241, type 3 (EV_ABS), code 57
(ABS_MT_TRACKING_ID),
value 0
Event: time 527.233264, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X),
value 5136
Event: time 527.233266, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y),
value 2723
Event: time 527.233268, ++++++ SYN_MT_REPORT
+++++
Event: time 527.233291, ----- SYN_REPORT -----
Event: time 527.237242, type 3 (EV_ABS), code 57
(ABS_MT_TRACKING_ID),
value 0
Event: time 527.237263, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X),
value 5136
Event: time 527.237265, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y),
value 2723
Event: time 527.237267, ++++++ SYN_MT_REPORT
+++++
Event: time 527.237284, ----- SYN_REPORT -----
```

## 8. マルチタッチの動作確認

“xf86-input-mtrack”が”X”サーバーに投げるボタンイベントの確認に”xev”を用いることができます。

ターミナルから”xev”を起動します。

起動例

```
# xev
```

“Xev” を起動すると”Event Tester”というウィンドウが表示されます。そのウィンドウ内にカーソルを移動し、タッチパネルの操作を行うことで、”xev”を起動したターミナルに発生したイベントが表示されます。

#### 9.対応”VID”及び”PID”

カーネル用ドライバ(“hid-unitec”)がサポートする”USB VID/PID”を以下に示します。

ID名	番号
VID	0x0AFA
PID	0x07D2