DMC Co., Ltd.
Interface with Touch

# Linux Multi-touch Serial Mode Driver v1.0 Operation Manual

DMC Co., Ltd.

**--Table of Contents--**

1. Introduction

This document is the user's manual for [Linux Multi-touch Serial Mode Driver] for DUS controller.

2. System Requirements

The descriptions in this document suppose a general environment (x86). Compilation and operation of kernel Ubuntu10.10("v2.6.37") were tested under the [x86] (= AT compatible) environment. The driver also supports [v2.6.37] or greater versions of kernels that support multi-touch. The examples of commands and installation described below may differ according to versions of kernels and distributions.

3. List of the Provided Files

| File Name | Description |
|---|---|
| "hid-unitec.c" | Kernel addition module source code |
| "xf86-input-mtrack-unitec.tar.gz" | Source code for "X" multi-touch driver |
| "10-unitec-serialmode.conf" | Configuration file for "X" multi-touch driver |

4. Multi-touch Driver Overview

4.1. Multi-touch driver for Kernel ("hid-unitec")

"hid-unitec" is a driver that supports serial mode as a report mode, and supports type "A" as multi-touch protocol of "Linux" kernel in "Digitizer Drivers for Windows Touch and Pen-Based Computers" ("http://www.microsoft.com/whdc/device/input/DigitizerDrvs_touch.mspx")

4.2. Multi-touch Driver for X ("xf86-input-mtrack")

"xf86-input-mtrack" is the driver to convert multi-touch protocol to a button event of input device and pass it to "X" server.

The pointer works as a touch pad by default ("https://github.com/BlueDragonX/xf86-inputmtrack"). "xf86-input-mtrack" makes the pointer work as a touch screen (issuing absolute coordinates and button-press events).

5. Build of Multi-touch Driver for Kernel

5.1. Build of Mulit-touch Driver "hid-unitec"

In order to build the driver, get the kernel for the current "Linux" from a kernel source package, and perform patch. The procedures are as follow.

5.1.1. Kernel Source Confirmation

Before starting to patch driver, compile the kernel and check if the kernel from the kernel source package works normally.

Example: How to compile

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

Note, you can also create a package and check operation

5.1.2. Patch

Patch “hid” sources as below.

Add the below in front of the ”endmenu” of “<kernel src dir>/drivers/hid/Kconfig”.

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

Add the below to anywhere in “<kernel src dir>/drivers/hid/Makefile”

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

Add the below to anywhere in ”hid_blacklist” table of “<kernel src dir>/drivers/hid/hid-core.c”.

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

Add the blow to anywhere in “<kernel src dir>/drivers/hid/hid-ids.h”.

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

5.1.3. Implementation of Kernel Addition Module Source for DUS controller

Enter ”hid-unitec.c” to “<kernel src dir>/drivers/hid”.

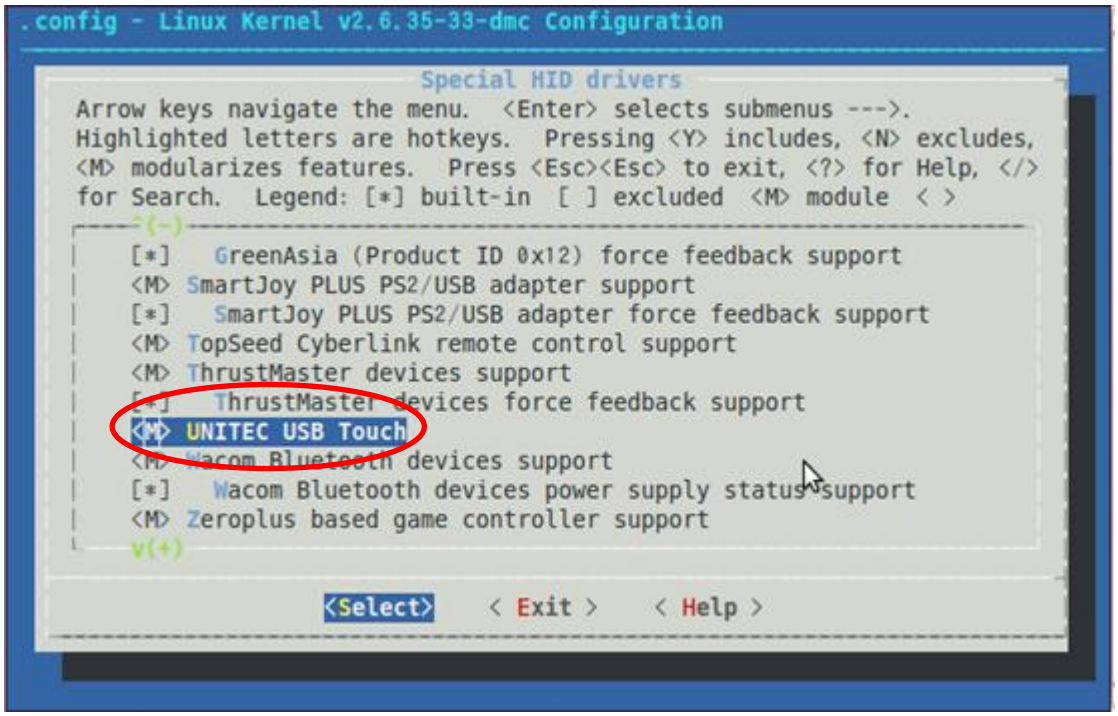
5.1.4.Kernel Configuration

Select CONFIG_HID_UNITEC (<M>)

「Linux Kernel Configuration」(Top Menu)
→「Device Drivers」
→「HID Devices」
→「Special HID drivers」
→ 「UNITEC USB Touch 」(＝CONFIG_HID_UNITEC)

Example: Configuration

```
# sudo make menuconfig
```




5.1.5.Compile the patched kernel and check operation.

Example: How to compile

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

Note, You can also create a package and install to check operation

6. Build of Multi-touch Driver for “X”

6.1. Building of Multi-touch Driver (“xf86-input-mtrack”)

The below is necessary in order to build “xf86-input-mtrack”.

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

”xserver-xorg-dev” and ”xorg-macros” will be not needed after ”xf86-input-mtrack“ is installed.

6.1.1.” xserver-xorg-dev”

To get started, install ”libpciaccess-dev”, ”libxkbfile-dev”, ”x11protodri2-dev”, and ”x11proto-fonts-dev” on which ”xserver-xorg-dev” depends.

Example: Install

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

Install “xserver-xorg-dev” in the same manner.

Example: Install

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

You may uninstall ”xserver-xorg-dev” after ”xf86-input-mtrack” is installed.

Example: Uninstall

```
# 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” or a greater version is necessary, but the version installed from a package may be old. If this is the case, you will need to download the original ”xorg-macros-1.8” or a greater version, and install it manually. ”xorg-macros” is included in ”util-macros”.

Example: Install

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

```
# sudo make install
```

You may uninstall ” xorg-macros” after ”xf86-input-mtrack” is installed.

Example: Uninstall

```
# sudo make uninstall
```

6.1.3. “mtdev”

Download the source code and install it manually.

Example: Manual Install

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

6.1.4.”xf86-input-mtrack”

Use “xf86-input-mtrack” included in the provided file.

Example: Install

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

6.2. Mutli-touch Driver (“xf86-input-mtrack”) Configuration

Put the configuration file in a certain place.

Example: Configuration File Installation

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

By placing this configuration file. ”X” server will load ”xf86-input-mtrack” driver when it recognizes a device matching the criteria (Name of the driver is ”mtrack”). The installed location of the configuration file may differ according to kernels and distribution versions.

6.3.Restarting of ”X” Server

Start the ”X” server in order to enable the multi-touch driver. You may restart the X server only, but it is recommended to restart “Linux” itself just in case.

Example: Restarting

```
# sudo reboot
```

6.4 Loading of Multi-touch Driver (”xf86-input-mtrack”)

Once a device is recognized, the X server will load the driver and output logs (like below) to the log file.

Example: Log Output

```
(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.
```

If the X server recognizes that device gets disconnected, it will unload the driver and output the logs (like below) to the log file.

Example: Log Output

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

7.Operation Checking of Multi-touch Protocol

”evtest” can be used to check operation of multi-touch protocol between “Linux” kernel and user land.

7.1.Building of ”evtest”

A version installed from a package can be too old. If this is the case, multi-touch event cannot be recognized. You will need to build the latest version manually.

Example: Manual Installation

```
# Wget
http://pkgs.fedoraproject.org/repo/pkgs/evtest/evtest-1.29.tar.bz2/1cc914cab3c30fa1e8b6
2005684e8ef3/evtest-1.29.tar.bz2
# tar xpjf evtest-1.29.tar.bz2
# cd evtest-1.29/
# autoreconf -i
# ./configure --prefix=/usr
# make
# sudo make install
```

7.2. Starting of “evtest”

If the multi-touch controller is recognized normally, ”/dev/input/evtestX” will be formed. The number for “X” will depend on the environment (You can check with the “X” server log or “/sys/class/input/eventX/device/name” for the number of “X”). Regarding it as argument,

execute “”evtest”.

Example: Execution

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

Once executed, the below screen will appear and await the input.

Example: Display Screen

```
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)
```

If touch screen is touched at this point, the protocol data sent from kernel will appear like below

Example: Protocol Data

```
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. Operation Checking of Multi-touch

You can use the ”xev” to check the button events sent from “xf86-input-mtrack” to ”X” server.

Start ”xev” from terminal

Example: Starting of ”xev”

```
# xev
```

Once the “Xev” started,” Event Tester” window will appear. Move the cursor into the window and operate the touch screen. The generated events will be displayed in the terminal that started the ”xev”

9. Supported ”VID” and ”PID”

”USB VID/PID” supported by kernel driver (“hid-unitec”) is as below.

| ID | Number |
|---|---|
| VID | 0x0AFA |
| PID | 0x07D2 |