Elo Touchscreen Linux Driver - MT-USB

Intel i686 (32 bit) or AMD64/Intel (64 bit)

Installation/Calibration/Uninstallation Instructions

Version 2.0.0
February 20, 2014
Elo Touch Solutions

Elo Linux MT USB Driver package contains userspace Linux drivers designed for Linux kernel 3.x and 2.6 (single touch only) and video alignment utility for Elo touchmonitors. This driver requires the presence of libusb-1.0 shared library on the target system for its operation. This driver supports a single touchscreen and single videoscreen setup only(multiple videoscreens with mirroring will work).

This readme file is organized as follows:

1. Supported Touchmonitors and Elo Touchscreen Controllers

- Elo Controllers
  IntelliTouch Plus/iTouch Plus 2515-07(non HID), 2521 (HID), 2515-00(HID)
  PCAP 2 touch, 4 touch and 10 touch controllers

2. System Requirements

- 32 bit Intel i686 (x86) platform (or)
  64 bit AMD/Intel x86_64 platform

- Kernels supported:
  Kernel version 2.6 (single touch only)
  Kernel version 3.x

- Motif versions supported:
  Motif version 3.0 (libXm.so.3)

- libusb versions supported:
  libusb version 1.0.9 or later

3. Installing the Elo Touchscreen USB Driver

Important:

a.) Must have administrator access rights on the Linux machine to install the Elo Touchscreen USB Driver.

b.) Ensure all earlier Elo drivers are uninstalled from the system.

c.) The Elo Touchscreen driver components require new libusb-1.0 library support (older libusb-0.1 library will not work). Most newer Linux distributions have started shipping this library (update to the popular libusb-0.1 library) as a part of their standard release. Customers can also download and compile the libusb-1.0 library from source (requires gcc v4.0.0 or later) available at libusb website. This driver will NOT work with the older libusb-0.1 library.

d.) Do not extract the downloaded binary package on a Windows system.

Step I:

Copy the elo driver files from the binary folder to the default elo folder. Change the permissions for all the elo driver files. These broad permissions are provided to suit most systems. Please change them to tailor it to your access control policy and for specific groups or users.

a.) Copy the driver files to /etc/opt/elo-mt-usb folder location.

```
# cp -r ./bin-mt-usb/  /etc/opt/elo-mt-usb
```

b.) Use the chmod command to set full permissions for all the users for the /etc/opt/elo-mt-usb folder.(read/write/execute). These broad permissions are provided to suit most systems. Please change them to tailor it to your access control policy and for specific groups or users.

```
# cd /etc/opt/elo-mt-usb
# chmod 777 *
# chmod 444 *.txt
```

c.) Copy the udev rules file to /etc/udev/rules.d/ folder location.
Please edit touchscreen device permissions to tailor it to your
access control policy and for specific groups or users.

```
 # cp /etc/opt/elo-mt-usb/99-elotouch.rules /etc/udev/rules.d
```

Step II:

Check if the kernel version for the elo_mt_input_mod kernel module (see the kernel version listed in the kernel module name) matches your current system's kernel version. If the kernel versions match, skip Step III and proceed to Step IV.

```
  [ List your current kernel version ]
  # uname -r
  [ List the kernel module name that contains the kernel version ]
  # ls -l /etc/opt/elo-mt-usb/*.ko
```

Step III:

Compile and build a new kernel module (elo_mt_input_mod.ko) for your system kernel if required. The kernel module source is present in the /etc/opt/elo-mt-usb/elo_mt_input_mod_src folder. Kernel source or header files, gcc, make and other development tools are needed to build a kernel module. Type "make install" to copy the kernel module to the /etc/opt/elo-mt-usb/ folder or use the cp command to manually copy the new kernel module to the /etc/opt/elo-mt-usb/ folder and rename it with your current kernel version.

```
  # cd /etc/opt/elo-mt-usb/elo_mt_input_mod_src
  # make

  # make install
  (or)
  # cp ./elo_mt_input_mod.ko ../elo_mt_input_mod_`uname -r`.ko
```

Step IV:

Configure a script to invoke Elo service at system startup.

Debian, Ubuntu (prior to 6.10) systems:

Copy the elorc script file present in the /etc/opt/elo-mt-usb directory to the /etc/init.d directory.

```
  # cp /etc/opt/elo-mt-usb/elorc /etc/init.d
```

A symbolic link for the elorc script has to be created in the desired runlevel directory (example: rc2.d,rc3.d,....rc5.d). This will allow the elorc script to run at system startup. Ubuntu and Debian systems use runlevel 2 (rc2.d directory) as default.

This directory has startup files (symbolic links) of the form SDDxxxx where DD is the sequence number. Pick a sequence number XX which is larger than the sequence number of the display manager script (xdm, gdm, etc.) found in this directory.

Use the maintainer script update-rc.d to create the elorc symbolic link with selected sequence number XX.

```
  # cd /etc/rc2.d
  # update-rc.d elorc start XX 2 .
```

Important:
==========

- Only use the update-rc.d maintainer script to modify these symbolic links. The elorc script will not be run at startup if these symbolic links are manually created.

- Notice that the update-rc.d command syntax has a space and period after the run-level parameter.

- The above example is for runlevel 2. Pick the appropriate folder for the desired runlevel. The default runlevel can be found in the /etc/inittab file.

Note:
=====

The path of the runlevel directories might vary from distribution to distribution. The path for runlevel 5 in Redhat is "/etc/rc.d/rc5.d" while the path for Debian and Ubuntu is "/etc/rc2.d" for runlevel 2.

Locate the corresponding runlevel directory in the system and create the symbolic link for elorc script file in that directory using the update-rc.d maintainer script.

Redhat, Fedora, Mandrake, Slackware, Mint and Ubuntu (6.10 or later) systems:

- -

On the above distributions, there are some restrictions for running the elorc script from /etc/rc.d/rc*.d directory. Hence, add the following line at the end of daemon configuration script in "/etc/rc.local" file.

```
  /etc/opt/elo-mt-usb/loadEloMultiTouchUSB.sh
```

[rc.local file might also be at location /etc/rc.d/rc.local. Use the "# find /etc -name rc.local" command to locate the rc.local file.]

SUSE Systems:

Add the following line at the end of the configuration script in "/etc/init.d/boot.local" file.

```
  /etc/opt/elo-mt-usb/loadEloMultiTouchUSB.sh
```

Step V:

Plug in the USB touchscreen and reboot the system to complete the driver installation process.

```
  # shutdown -r now
```

4. Calibrating the Touchscreen

Important:

Users must have read and write access to "/dev/elo-mt-usb" directory to perform the touchscreen calibration. All long command line options in elova calibration utility use the "--" format. (example: "--help")

Type "# /etc/opt/elo-mt-usb/elova --help" for available command line parameters and usage.

Step I:

Run the calibration utility from a command window in X Windows from the /etc/opt/elo-mt-usb directory for a single or multiple video setup (supports Xorg Xinerama, Xorg non-Xinerama and Nvidia Twinview options).

```
  # cd /etc/opt/elo-mt-usb
  # ./elova
```

In a multiple video setup, the calibration target(s) will be shown on the first video screen and switch to the next video screen after a 30 second default timeout for each target or screen. Once the touchscreen is calibrated the data is stored in a configuration file on the hard disk. To display the calibration targets on just one specific video screen(example:videoscreen[1]) use the command shown below.

```
  # cd /etc/opt/elo-mt-usb
  # ./elova --videoscreen=1
```

To change or disable the default calibration timeout for each target or screen, use the command shown below. [Timeout Range: Min=0 (no timeout), Max=300 secs, Default=30 secs]

```
  # cd /etc/opt/elo-mt-usb
  # ./elova --caltargettimeout=0      [Disable the calibration timeout for all
targets and videoscreens]
  # ./elova --caltargettimeout=45     [Modify the calibration timeout to 45 seconds]
```

To view a list of video and USB touch devices available for calibration, use the command shown below.

```
  # cd /etc/opt/elo-mt-usb
  # ./elova --viewdevices
```

To view all the available options and specific usage for elova calibration program, use the command shown below.

```
  # cd /etc/opt/elo-mt-usb
  # ./elova --help
```

Step II:

Touch the target(s) from a position of normal use. The calibration data is written to the driver at the end of calibration.

5. Uninstalling the Elo Touchscreen USB Driver

Important:

Must have administrator access rights on the Linux machine to uninstall the Elo Touchscreen USB Driver.

Step I:

Delete the script or commands that invoke Elo service at startup.

SUSE systems:

Remove the following entry created in Step IV of Installation section from the configuration script in"/etc/init.d/boot.local" file.

```
  /etc/opt/elo-mt-usb/loadEloMultiTouchUSB.sh
```

Redhat, Fedora, Mandrake, Slackware, Mint and Ubuntu (6.10 or later) systems:
- -

Remove the following entry created in Step IV of Installation section from the configuration script in "/etc/rc.local" file. (or "/etc/rc.d/rc.local" file)

```
  /etc/opt/elo-mt-usb/loadEloMultiTouchUSB.sh
```

Debian, Ubuntu (prior to 6.10) systems:
- -

a.) Remove the symbolic link file created in Step IV of Installation section using the update-rc.d maintainer script.

```
      # update-rc.d -f elorc remove
```

b.) Remove the elo script file "elorc" placed in the "/etc/init.d" directory.

```
      # rm /etc/init.d/elorc
```

Step II:

Delete all the elo driver files from the system.

a.) Delete the main elo driver folder.

```
        # rm -rf /etc/opt/elo-mt-usb
```

b.) Delete the elo related device folder and files.

```
        # rm -rf /dev/elo-mt-usb
        # rm -rf /etc/udev/rules.d/99-elotouch.rules
```

Step III:

Reboot the system to complete the driver uninstallation process.

```
  # shutdown -r now
```

==================
6. Troubleshooting
==================

A. Make sure libusb-1.0 library is installed on the target Linux system. The driver will NOT work with the older libusb-0.1 library. Most Linux distributions come with the newer libusb-1.0 library installed by default. It can also be installed by downloading and compiling the library source (requires gcc v4.0.0 or later) from the libusb-1.0 website.

B. If touch is not working, check if the elomtusbd driver is loaded and currently available in memory. Some Xorg Xserver versions terminate the touchscreen driver upon user logout. The current workaround in this situation is to startup the driver from Xwindows startup script or reboot the system.

```
  # ps -e |grep elo
```

Check the driver log file for any errors that have been reported.

```
  # gedit /etc/opt/elo-mt-usb/EloUsbErrorLog.txt
```

If the driver is not present then load the driver again. Root access is needed to load the driver manually. Normal users will have to restart the system so that the elomtusbd daemon is loaded again during system startup. Normal users may be able to load the driver manually depending on access control and file permissions that are setup.

```
  # /etc/opt/elo-mt-usb/elomtusbd
```

C. If starting the Elo touchscreen driver from the normal startup locations like rc.local or boot.local does not work, first test if the touchscreen is working by manually launching the driver from a terminal window within XWindows GUI.

```
  # /etc/opt/elo-mt-usb/loadEloMultiTouchUSB.sh
```

If the touchscreen works when the driver is launched manually, try to add the touchscreen driver startup line to the end of one of the XWindows startup scripts. The Xwindows startup scripts are located usually in the following path /etc/X11/xinit/xinitrc.d/. Running the touchscreen driver from the Xwindows startup script will provide touch input ONLY after the user has logged in successfully at the GUI Login screen.

D. While trying to load the driver manually, if you get an error "Error opening USB_ERROR_LOG_FILE", check the file permissions for the /etc/opt/elo-mt-usb/EloUsbErrorLog.txt file. The user needs to have read and write access to this log file to launch the driver.

E. In a multi video setup, the touchscreen can be mapped to just one videoscreen. First find the name of the video port (example: VGA-1, HDMI-0, DVI-0) that connects to the desired videoscreen, using the xrandr command in a terminal window.

```
  # xrandr
```

Next, find the device ID (id=xx) of the Xinput pointer device "Elo MultiTouch(MT) Device Input Module" using the xinput command in a terminal window.

```
  # xinput
```

Finally, map the touchscreen device ID to the desired video port using the xinput command's --map-to-output option.

```
  # xinput --map-to-output 22 VGA-1   [Map input device ID 22 to VGA-1 port]
```

The input device ID and video port name are stable across system reboots. The above mapping command can be added to a startup script to perform the mapping at every boot after the Elo MTUSB driver have been loaded.

F. In some Linux distributions (example: Ubuntu 12.04) the desktop does not respond to clicks after some time, while the pointer still follows the touch input. This is a know bug in Xwindows which has been fixed on newer versions. To solve this issue, either upgrade to newer version of Xwindows or download the bug fix, patch and recompile current version of Xserver.

G. The default mode of the touch driver is multitouch mode with the primary touch events duplicated as mouse events to support applications that listen to mouse events only. The touch driver's kernel module code(elo_mt_input.c) detects if the target system's kernel supports Multitouch Protocol (Linux kernel version 2.6.38 or 3.x.x) and then enables the _MULTITOUCH_KERNEL_ flag to report multitouch events to the kernel. On older Linux kernels the driver reports only the primary touch events(as mouse events) and discards the other touches.

On multitouch capable kernels(2.6.38, 3.x.x), disabling the _MULTITOUCH_KERNEL_ flag in the kernel module code, rebuilding and installing a modified kernel module will force the driver to send mouse events(single touch events) based on primary touch information, similar to the behavior described above for old Linux kernels.

If the target application or system (example: POS system) does not require multitouch events or gestures on a multitouch capable kernel, forcing the driver to send mouse events might provide better results.

7. Contacting Elo Touch Solutions

Website: http://www.elotouch.com

E-mail: customerservice@elotouch.com

Mailing Address:

```
  Elo Touch Solutions Inc,
  1033 McCarthy Blvd.
  Milpitas, CA 95035
  USA

  Phone:   (800) 557-1458
           (650) 361-4800
  Fax:     (650) 361-4722


================================================================================

                      Copyright (c) 2014 Elo Touch Solutions Inc

                                All rights reserved.

================================================================================
```