

Free-flow USB Mouse Single/Multi-monitor Driver Build and Install Instructions

Pete Arnold

26 February 2015 / 13:00

OPTION A: COPY PRE-BUILT DRIVERS

Drivers built on an x86_64 system with CentOS 5.8 are available. The following assumes the same system with default locations and kernel numbers for the files.

You can either copy/modify the files manually as described below or use the install script. To use the install script, as root, navigate to the directory containing the script and type:

```
./install
```

Then reboot the PC. The Free-Flow mouse should be working on the X log-in screen.

To perform a manual install:

1. The Kernel Driver

The driver is provided in the PreBuilt/CentOS-5.8/kernel folder. Proceed as follows:

```
cp freeflow.ko /lib/modules/2.6.18-308.el5/extra
/sbin/depmod -a
cp 62-freeflow.rules /etc/udev/rules.d
```

2. The Xorg Driver

The driver is located in the PreBuilt/CentOS-5.8/xorg folder. Proceed as follows:

```
cp freeflow_drv.* /usr/lib64/xorg/modules/input
```

Then modify the xorg.conf to include the Free-Flow InputDevice section and a reference to it in the ServerLayout section.

Then reboot the PC and the mouse should work!

OPTION B: BUILD DRIVERS FROM SOURCE

The following instructions have been tried on CentOS 5.2, 5.5 and 5.8 and in all cases resulted in multi-monitor behaviour in X.

Under 5.8 the cursor also works on the text screen in runlevel 3 but hasn't worked in 5.2/5.5 yet.

These drivers will not compile on CentOS 7. They haven't yet been tried on CentOS 6. For both 6 & 7, the modern kernel driver is recommended.

Note: the CentOS build instructions for a kernel module suggest that the files will be installed at ~/rpmbuild. This is the case in 5.2 but on 5.5 and 5.8 the default instructions resulted in a build tree at /usr/src/redhat.

1. Get and Extract the Kernel Source Code.

As root:

For 5.2 and 5.5, the build process wants to place the source at /usr/src/redhat: it is probably worth making that directory now, so that it can use it if it needs it:

```
mkdir /usr/src/redhat
```

```
rpm -i http://vault.centos.org/5.11/os/SRPMS/kernel-2.6.18-398.el5.src.rpm 2>&1 | grep -v exist
```

Note that you need to replace 5.11 and -398 as follows:

CentOS 5.2	5.11 -> 5.2	-398 -> -92
CentOS 5.5	5.11 -> 5.5	-398 -> -194
CentOS 5.8	5.11 -> 5.8	-398 -> -308.

As user:

```
cd ~/rpmbuild/SPECS
rpmbuild -bp --target=$(uname -m) kernel-2.6.spec
```

(Note that for 5.8 this is kernel.spec)

(if there is no code at ~/rpmbuild, it may have been placed at /usr/src/redhat - just work there).
If you don't have the following, you will need to install (as root):

```
yum install rpm-build
yum install redhat-rpm-config
yum install unifdef
yum install gcc
```

Then back to the source extraction:

```
rpmbuild -bp --target=$(uname -m) kernel-2.6.spec
```

(Note that for 5.8 this is kernel.spec)

Source is now at ~/rpmbuild/BUILD (or at /usr/src/redhat/BUILD).

2. Add the Free-Flow Kernel Diver to the Build.

Copy freeflow.c to both the BUILD/kernel-2.6.18/vanilla/drivers/usb/input and the BUILD/kernel-2.6.18/linux-2.6.18.x86_64/drivers/usb/input folders. At 5.8, this is vanilla-2.6.18.4.

You will then need to edit the Kconfig and Makefiles in the linux-2.6.18.x86_64/drivers/usb/input folder and its parent (which I think are linked into the vanilla folder, so these are the same file in both trees).

The Kconfig-section file provided contains only the section for freeflow, you will need to add this to the existing Kconfig file (don't replace it). Likewise with the Makefile-section.

You may also need to add a line to the Makefile in the parent directory (linux-2.6.18.x86_64/drivers/usb) to += input/ on CONFIG_USB_FREEFLOW.

Finally, move to the linux-2.6.18.x86_64 directory from where you will need to do all the building and add the following line to the .config file (preferably close to the other USB INPUT devices --- look for CONFIG_USB_HID).

```
CONFIG_USB_FREEFLOW=m
```

3. Configure / Prepare the Kernel Driver Build

Do the following (the menuconfig failed on my builds so I left it out - this shouldn't be a problem unless you want to change the original configuration, so it may be best ignored):

```
make oldconfig
make menuconfig ---> failed
```

```
make prepare
make scripts
```

4. Make the Kernel Driver

You can build the driver with (which will also build the other usb/input drivers ...).

```
make M=drivers/usb/input
```

A further few steps are required to build the different modules for the different types of kernel. Edit the main Makefile in the **root directory of the kernel source (the linux-2.6.18.x86_64 folder)**. For CentOS 5, find:

```
EXTRAVERSION = -prep
```

and replace it with (for 5.2, I had kernel 92, 5.5 kernel 194 and 5.8 kernels 308 and 398 (depending on the update state)):

```
EXTRAVERSION = -92.el5
```

You may need to touch the file to make it build:

```
touch drivers/usb/input/freeflow.c
```

Make the driver:

```
make M=drivers/usb/input
```

Strip out the debug:

```
strip --strip-debug drivers/usb/input/freeflow.ko
```

Copy the driver to the extra lib folder (note that if this does not work, you may need to copy the driver into the /lib/modules/2.6.18-92.el5/drivers/usb/input folder).

```
cp drivers/usb/input/freeflow.ko /lib/modules/2.6.18-92.el5/extra
```

Rebuild the module dependency file:

```
/sbin/depmod -a
```

5. Attach the Driver to the Device

Determine where your device is attached from the dmesg output (look for the Adder Technology Free-Flow USB Mouse and get the USB address). You may need to adder ":1.0" to the address found.

Remove the default driver from the device:

```
echo -n "2-1.4.1.3:1.0" > /sys/bus/usb/drivers/usbhid/unbind
```

Load the freeflow module:

```
/sbin/modprobe freeflow
```

Install the driver on the device:

```
echo -n "2-1.4.1.3:1.0" > /sys/bus/usb/drivers/freeflow/bind
```

Note that the driver may bind automatically on the modprobe. You can check this by looking for the device ID in the driver directory:

```
ls -l /sys/bus/usb/drivers/freeflow
```

This has failed on 5.2 and 5.5 but that did not stop the X system working. It may be that the X system will load or associate evdev making the drivers work in X - not sure what is happening here yet, but the dmesg output includes:

```
dmesg | grep freeflow ---> error loading evdev module
```

You can make this happen automatically using a udev rule. Create a file in the udev rules directory, e.g. /etc/udev/rules.d/62-freeflow.rules, containing:

```
SYSFS{idVendor}=="21d1", SYSFS{idProduct}=="0001", MODE="0666", PROGRAM="/bin/sh -c 'echo -n $id:1.0 >/sys/bus/usb/drivers/usbhid/unbind; /modprobe freeflow; echo -n $id:1.0 >/sys/bus/usb/drivers/freeflow/bind'"
```

It isn't yet clear whether we also need to echo to the same id but with :1.1 following as well.

If you want to see which ports have devices, you can use:

```
ls -l /sys/bus/usb/devices
```

If you want to find out if a particular device number is bound to a driver, you can use:

```
find 2.1.3 /sys/bus/usb/drivers | grep 2.1.3
```

6. Make the Xorg Driver.

Make a copy of the xf86-input-freeflow-1.0.0-release-backup folder to xf86-input-freeflow-1.0.0 and change to that directory.

```
cp -pr xf86-input-freeflow-1.0.0-release-backup xf86-input-freeflow-1.0.0
cd xf86-input-freeflow-1.0.0
```

You can build the X11 driver in that folder (wherever it is, so you can move it if you wish) as follows:

```
./configure
```

This may fail with no package xorg-server and/or xproto. You will need to install the Xorg development files with:

```
yum install xorg-x11-server-devel
```

or maybe ...

```
yum install xorg-x11-proto-devel
```

or in desperation ...

```
yum install xorg-x11*
```

You will need to re-run:

```
./configure
```

Then build the driver with:

```
make clean
make
make install
```

7. Install the Xorg Driver.

Copy the driver to the current library folder (the make install places it in the local libraries but this doesn't seem to be recognised by X).

```
cp /usr/local/lib/xorg/modules/input/freeflow_drv.* /usr/lib64/xorg/modules/input
```

8. Update the xorg.conf Configuration File

In /etc/X11/xorg.conf, add the InputDevice section for the Free-Flow mouse and add a reference to it in the ServerLayout section - see the xorg.conf-sample for the file I have used with my 5.x's.

```
Section "InputDevice"
    Identifier "Free-Flow"
    Driver "freeflow"
    Option "Device" "/dev/input/event4"
    Option "DebugLevel" "7"
    Option "SendCoreEvents"
    Option "Screen1" "0 0 1920 1080"
    Option "Screen2" "1920 0 3840 1080"
    Option "Screen3" "1920 1080 3840 2160"
    Option "Screen4" "0 1080 1920 2160"
EndSection
```

The SendCoreEvents value can be added to the Free-Flow entry in the ServerLayout or in the InputDevice section.

9. Start X and Debug

Start Xorg with

```
startx
```

X specific debug can be found in /var/log/Xorg.0.log.

The cursor originally failed. I then returned to runlevel 3 and unbound the device from freeflow and bound it again, returned to X and the cursor was working (started X as root this time) as follows:

Boot the PC to runlevel 3, then:

```
echo -n "2-1.4.1.3:1.0" > /sys/bus/usb/drivers/usbhid/unbind
/sbin/modprobe freeflow
echo -n "2-1.4.1.3:1.0" > /sys/bus/usb/drivers/freeflow/bind
startx
```

The layout in the CCSx should be the same as that specified in the xorg.conf file.

To install the source package and tools for CentOS-6:

As root, install the following packages:

```
yum install rpm-build redhat-rpm-config asciidoc hmaccalc perl-ExtUtils-Embed xmlto  
yum install audit-libs-devel binutils-devel elfutils-devel elfutils-libelf-devel  
yum install newt-devel python-devel zlib-devel
```

```
yum install gcc redhat-rpm-config patchutils bison
```

As a user get the source using:

```
rpm -i http://vault.centos.org/6.5/os/Source/SPackages/kernel-2.6.32-431.el6.src.rpm 2>&1  
| grep -v exist
```

Note that you need to replace 6.5 and -431 as follows:

CentOS 6.4 6.5 -> 6.4 -431 ->