

# Improve Handling For Touchpads And Mice With Libinput

## # Contact Information

**Name** : Furkan

**Surname** : Tokaç

**Email** : furkantokac34@gmail.com || furkan.tokac@metu.edu.tr

**Blog**: furkantokac.com **GitHub**: @furkantokac **Telegram**: @ftDev7

## # Introduction

Currently KDE is moving from X11 to Wayland and the input device handling methods differ according to these protocols. X11 uses synaptics, evdev directly, or libinput as an input device handling methods, while Wayland only uses libinput. To use libinput in X11, we have to go through the X11 server to interact with it, whereas on Wayland, we have to go through the compositor (for us, KWin). Because of these different code paths, an abstraction layer has to be written that allows the same UI and functionality to be displayed for both the X11 and Wayland cases.

This task is about that; there is a new UI for Wayland which is coded in QML to handle touchpad input devices. Because it is coded for Wayland, its backend is libinput but we also want to use that UI and its functionalities in X11 systems that use libinput so we need an abstraction layer.

My goal involves porting the new Wayland-only touchpad UI to the X11 system. After the porting, I plan to improve the user experience for touchpad and mouse management by implementing new features like advanced scrolling behaviors and more granular speed and acceleration handling.

Part of the current situation of the problem can be viewed from:

<https://phabricator.kde.org/D11469> || <https://phabricator.kde.org/D11468>

KDE input handling structures

### **X11**

- Touchpad: evdev > synaptics or libinput driver > xorg server > System Settings manages parameters via xinput
- Mouse: evdev > [nothing] or libinput driver > xorg server > System Settings manages parameters via xinput

### **Wayland**

- Touchpad: evdev > libinput driver > Wayland compositor (KWin) > System Settings manages parameters via compositor

- Mouse: evdev > libinput driver > Wayland compositor (KWin) > System Settings manages parameters via compositor

## **# Deliverables and Implementation**

At least the following additions and enhancements will be done during the GSoC.

### **1) Porting touchpad UI to X11**

Currently, touchpad UI and its backend is written for Wayland and being used there so I'll port it to X11. Firstly, I'll apply new UI to the X11 system without backend. Then, I'll implement the backend which will be an abstraction layer between libinput and X11. Similar work is being done for mouse UI so I can get benefit of it but of course, mouse backend and touchpad backends are not totally same.

The related directories about this work are;

New mouse UI porting works : kcms/input/

My work will be in : kcms/touchpad/

Current X11 touchpad UI files : plasma-desktop/kcms/touchpad/src/kcm/xlib/ui/

Also, following bug report is related with this issue

[https://bugs.kde.org/show\\_bug.cgi?id=383379](https://bugs.kde.org/show_bug.cgi?id=383379)

### **2) Improving touchpad management**

Firstly, it should be decided to arrangement of new UI elements. According to decisions taken, I'll implement the UI. Related files about this implementation are found in kcms/touchpad/ directory. Then, backend implementation will start. While I'm implementing the backend features, I can benefit from official libinput documentation. Also it is considered that to implement some features, we may need to contact with libinput developers. The related files for this implementation are basically found in kcms/touchpad/ directory.

The features that I'm planning to implement:

- \* Two fingers scroll and touchpad edges scroll can be used at the same time
- \* Mouse click emulation (for ex. two finger tapping means right click)
- \* Corners (for ex. tapping right-top corner simulate middle click)
- \* Speed min, max, acceleration
- \* Disable touchpad when mouse is plugged in

As long as the developers think that the feature is suitable to implement, it will be implemented. Discussions will be made if it is needed. I'll implement different feature if unplanned events happen. Other than that, if there is enough time left after main tasks, bug reports about related issues can be handled.

For ex. [https://bugs.kde.org/show\\_bug.cgi?id=387156](https://bugs.kde.org/show_bug.cgi?id=387156)

3) **Improving mouse management**

The basic process will be same as the previous implementation. The related files for this implementation are basically found in kcms/input/ directory.

The features that I'm planning to implement:

- \* Clicking (double click or single click open files and folders)
- \* Pointer speed min, max, acceleration
- \* Double click interval
- \* Mouse wheel scrolling speed

As an extra work, mouse-click test area can be applied.

4) **If have time, working on future works** : I'll start to implement custom mouse button support. By this feature, extra buttons on the mouses will be able to mapped to another mouse buttons. For ex. custom button with id of 0x10 will be mapped to middle click. The final plan will be done with the mentor at that time according to situation.

Each implementation would be followed by writing relevant unit test (if applicable) and extensive manual testing.

## **# Timeline**

**April 23 - May 14**: Community Bonding Phase

During this period, I'll try to read and hack around the relevant areas mentioned above to clearly understand the code and side by side discuss and finalize the layout for the implementations.

**May 14 - June 15**: Coding Phase 1

I'll work on implementation of "Porting touchpad UI to X11" during this phase.

**June 11 - June 15**: First Evaluations

I'll be submitting a detailed report of my work till that point.

**June 15 - July 9**: Coding Phase 2

I'll work on the implementation of "Improving touchpad management" and "Improving mouse management" during this phase.

**July 9 - July 13**: Second Evaluations

I'll be submitting a detailed report of my work till that point.

**July 13 - Aug 6**: Coding Phase 3

If there are works missing from previous coding phases, or there are extra unplanned works, they will be handled. If not, I'll start to implementation of future works. The decisions will be taken with mentors.

### Aug 6 - Aug 14:

Final testing all the features and wrapping up. Buffer time to tackle unplanned events.

## # Future Work

Free Software is part of my life so after my GSoC tasks, I'll keep contributing to KDE project.

The features that I wish to see in KDE are as follows:

- 1) Custom mouse button support.
- 2) Macro support for mouse buttons.
- 3) Macro support for keyboard.
- 4) Touchscreen management in System Settings>Input Devices.

## # About me

I'm basically Free Software lover/supporter and I have a special relationship with KDE :) I started coding with "Deitel & Deitel C and C++" book in 2012 and since that time, I am trying to improve my art. My main interest is embedded systems, embedded Linux systems. Also, I'm studying Computer Engineering in Middle East Technical University.

## # Some References

LibreOffice contributions

- [My last patch that is merged](#)
- [I'm working on 1](#)
- [I'm working on 2](#)

Localization of GNOME

- [Team page](#)

Qt/PyQt Projects

- [GitHub Repositories](#)
- The Qt projects in Github : Kmousens, Faunus-Market, Fcompressor
- The PyQt projects in Github : KLocalShare, Freg, Fmail, Faunus-Old