

Programming in the Sun: A Year with the Daylight Computer

Oskar Wickström — October 10, 2025

I've been [hinting on X/Twitter](#) about my use of the Daylight DC-1 as a programming environment, and after about a year of use, it's time to write about it in longer form. This isn't a full product review, but rather an experience report on coding in sunlight. It's also about the Boox Tab Ultra – which has a different type of display – and how it compares to the DC-1 for my use cases.

This is *not* a sponsored post.

```

68     throw Error("fuzz function not found");
69 }
70
71 await loop(module).catch((err) => {
72     log(err.stack);
73     process.exit(1);
74 });
75
76 function receive(): Promise<protocol.Request> {
77     let message_size_remaining = 0;
78     let message_buffer = Buffer.alloc(0);
79     const disposers: (() => void)[] = [];
80
81     return new Promise((resolve, reject) => {
82         function on_data(data: Buffer) {
83             if (message_size_remaining > 0) {
84                 const to_copy = Math.min(message_size_remaining, data.length);
85                 message_buffer = Buffer.concat([
86                     message_buffer,
87                     Uint8Array.prototype.slice.call(data, 0, to_copy),
88                 ]);
89                 message_size_remaining -= to_copy;
90             } else {
91                 assert(message_size_remaining === 0);
92                 message_size_remaining = data.readUInt32LE(0);
93                 message_buffer = Buffer.alloc(0);
94
95                 if (data.length > 4) {
96                     const to_copy = Math.min(message_size_remaining, data.length);
97                     message_buffer = Buffer.concat([
98                         message_buffer,
99                         Uint8Array.prototype.slice.call(data, 4, 4 + to_copy),
100                     ]);
101                     message_size_remaining -= to_copy;
102                 }
103             }
104
105             if (message_size_remaining === 0) {
106                 disposers.forEach((d) => {
107                     d();
108                 });
109
110                 resolve(protocol.Request.decode(message_buffer));
111             }
112         }
113
114         function on_end() {
115             if (message_size_remaining > 0) {

```

src/worker.ts

91,1

69%

[0] 1:nvim*

"oak" 13:19 09-Oct-25

Neovim in Termux on the Daylight DC-1.

Why do I even bother, you might ask? Sunlight makes me energetic and alert, which I need when I work. Living in the Nordics, 50% of the year is primarily dark, so any direct daylight I

can get becomes really important. I usually run light mode on my Framework laptop during the day, but working in actual daylight with these displays, or plain old paper, is even better.

The Setup

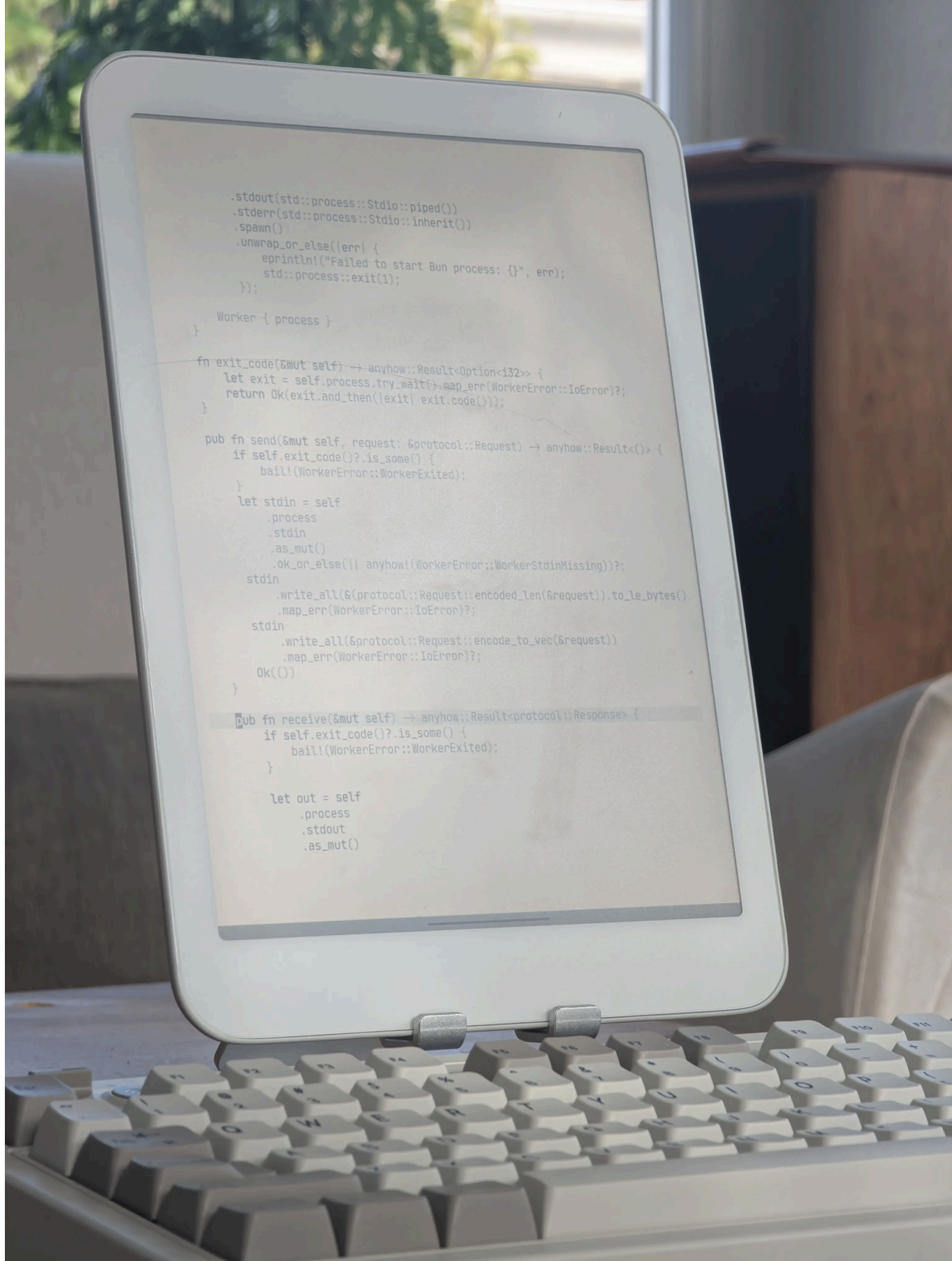
Here are the main components of this coding environment:

- Daylight DC-1: an Android-based tablet with a “Live Paper” display (Reflective LCD, not E-Ink)
- 8BitDo Retro Mechanical Keyboard: a mechanical Bluetooth-enabled keyboard, with Kailh key switches and USB-C charging and optional connection
- Termux: a terminal emulator for Android, with a package collection based on
- SSH, tmux, and Neovim: nothing surprising here

I use a slimmed-down version of my regular dotfiles, because this setup doesn’t use Nix. I’ve manually installed Neovim, tmux, and a few other essentials, using the package manager that comes with Termux. I’ve configured Termux to not show its virtual keyboard when a physical keyboard is connected (the Bluetooth keyboard). The Termux theme is “E-Ink” and the font is JetBrains Mono, all built into Termux. Neovim uses the built-in colorscheme for maximum contrast.

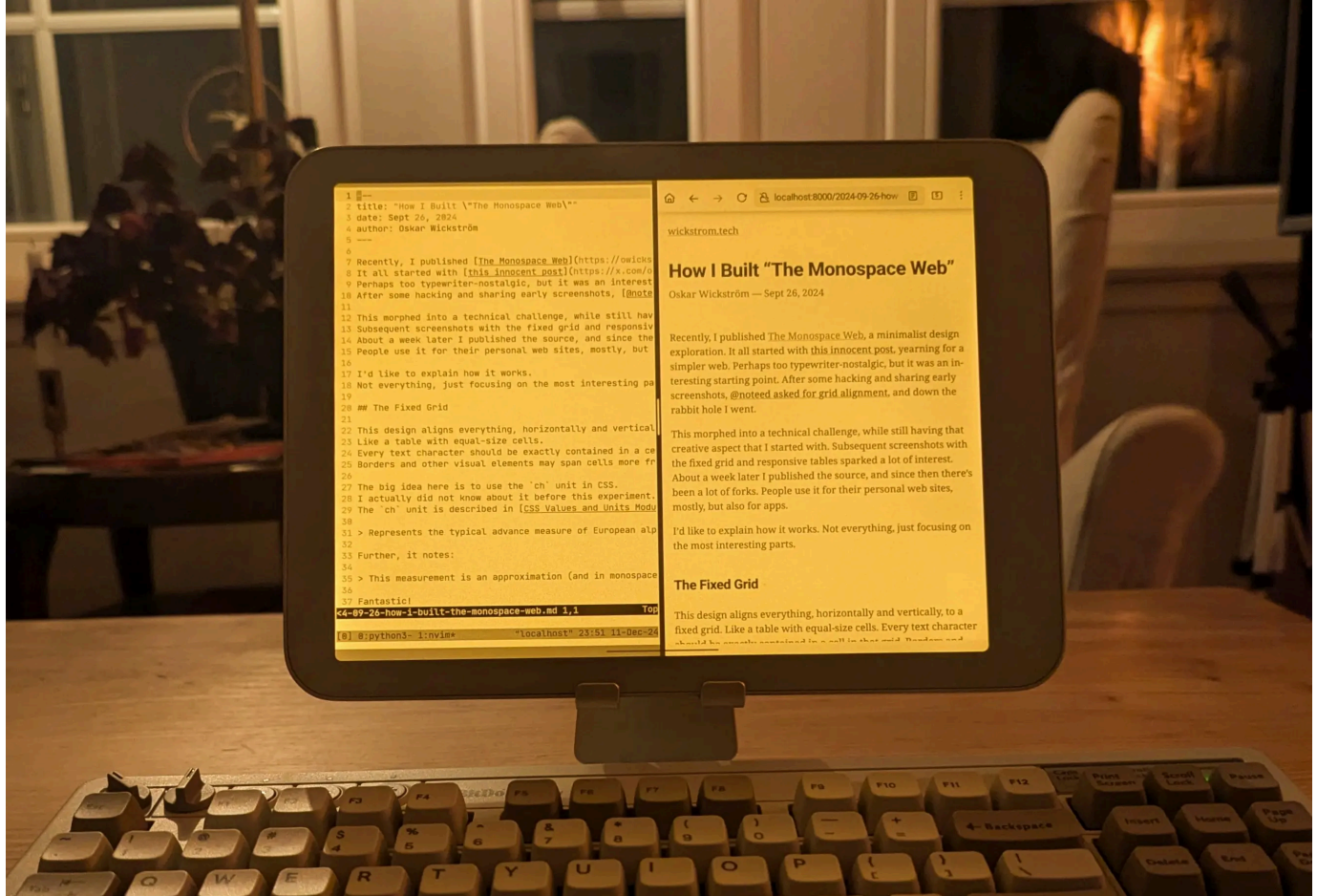
Certain work requires a more capable environment, and in those cases I connect to my workstation using SSH and run tmux in there. For writing or simpler programming projects (I’ve even done Rust work with Cargo, for instance), the local Termux environment is fine.

Sometimes I want to go really minimalist, so I hide the tmux status bar and run ____ in Neovim. *Deep breaths. Feel the fresh air in your lungs.* This is especially nice for writing blog posts like this one.



Minimalist typing with Goyo in Neovim.

My blog editing works locally in Termux, with a live reloading Chrome in a split window, here during an evening writing session with the warm backlight enabled:



Split-screen blogging locally on the Daylight.

There's the occasional Bluetooth connection problem with the 8BitDo keyboard. I also don't love the layout, and I'm considering getting the Kinesis Freestyle2 Blue instead. I already have the wired version for my workstation, and the ergonomics are great.

Daylight DC-1 vs Boox Tab Ultra

What about the Boox? I've had this device for longer and I really like it too, but not for the same tasks. The E-Ink display is, quite frankly, a lot nicer to read on; EPUB books, research PDFs, web articles, etc. The 227 PPI instead of the Daylight's 190 PPI makes a difference, and I like the look of E-Ink better overall.

However, the refresh rate and ghosting make it a bit frustrating for typing. Same goes for drawing, which I've used the Daylight for a lot. Most of my home renovation blueprints are sketched on the Daylight. The refresh rate makes it possible.

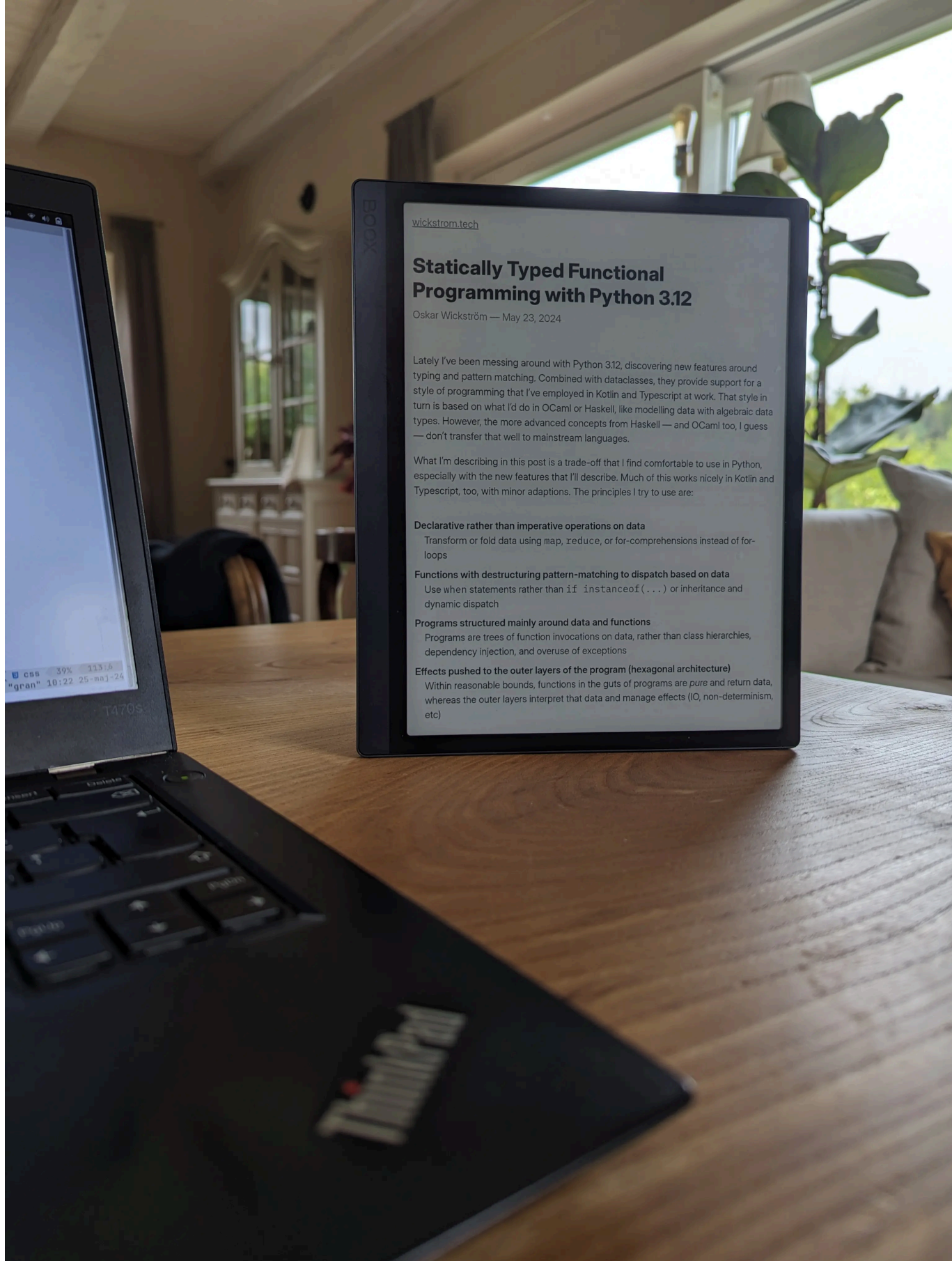
When reading at night with a more direct bedside lamp, often in combination with a subtle backlight, the Boox is much better. The Daylight screen can glare quite a bit, so the only option is backlight only. And at that point, a lot of the paperlike quality goes away.

You can also get some glare when there's direct sunlight at a particular angle:



You may get glare in direct sunlight or from lamps at some angles.

Even if I don't write or program directly on the Boox, I've experimented with using it as a secondary display, like for the live reload blog preview:



Using the Boox Tab Ultra as a secondary display by browsing the live reload HTTP server.

To sum up, these devices are good for different things, in my experience. I've probably spent more time on the Boox, because I've had it for longer and I've read a lot on it, but the Daylight has been much better for typing and drawing.

Another thing I'd like to try is a larger E-Ink monitor for my workstation, like the one Zack is hacking on. I'm hoping this technology continues to improve on refresh rate, because I love E-Ink. Until then, the Daylight is a good compromise.



Touch grass, as they say.

[Home](#) [Atom feed](#) [X](#) [Bluesky](#) [Mastodon](#) [CV](#)