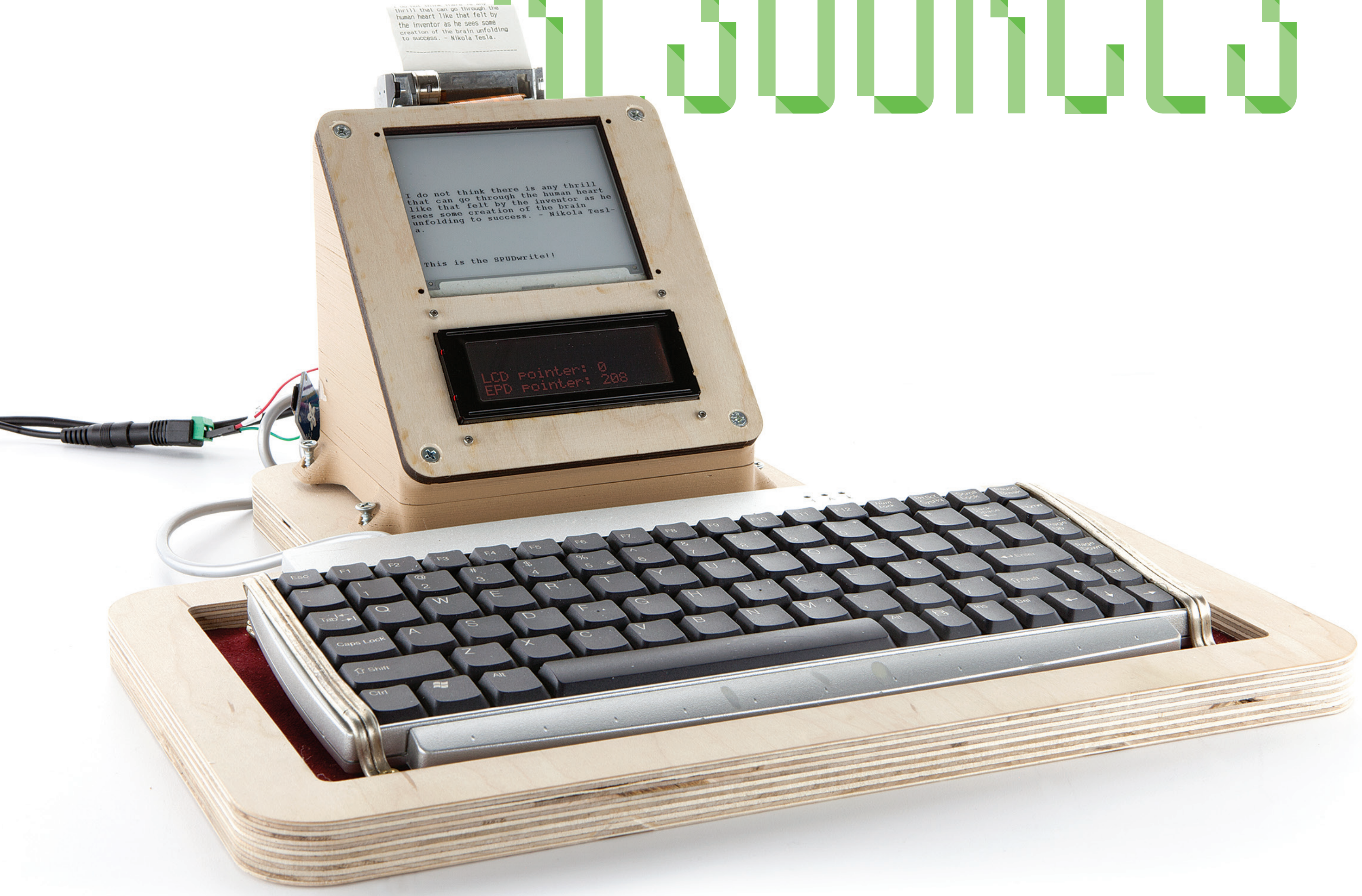

I do not think there is any thrill that can go through the human heart like that felt by the inventor as he sees some creation of the brain unfolding to success. - Nikola Tesla.
This is the SPUDwrite!!
LCD Pointer: 0
EPD Pointer: 208

RANDI KLETT (5)

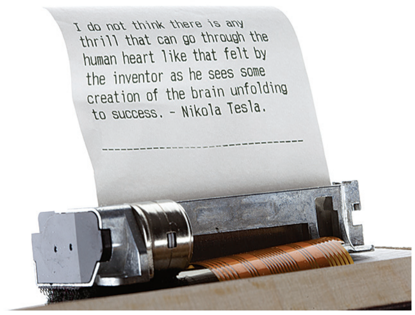
BECOME AN E-INK STAINED WRETCH

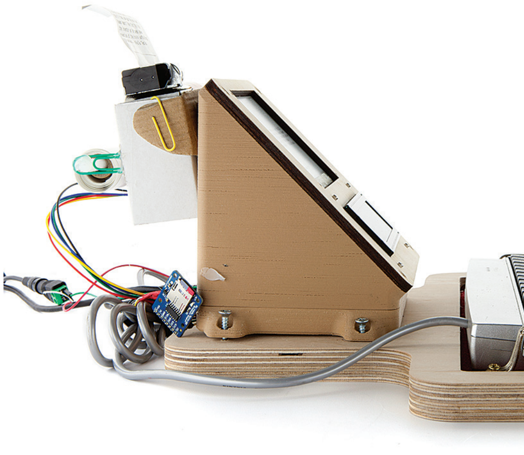
BUILD A SIMPLE TYPEWRITER WITH A MAKER-FRIENDLY DISPLAY

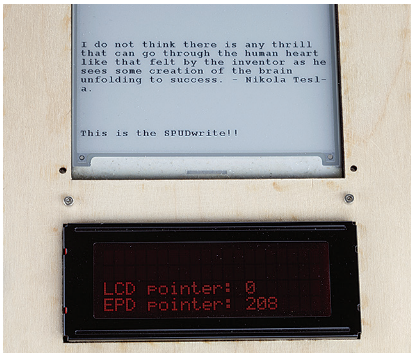
MY ENTIRE PROFESSIONAL LIFE AS AN ENGINEER IS SPENT ON, OR ADJACENT TO, A GLOWING laptop. I use the same computer to code, research, design, and document, and even to waste time. Hour upon hour, every day of the week, every week. So I push to have my creative time be analog: Coming home to do some creative writing, only to sit down at the exact same computer, feels more exhausting than restful—*oh joy, more screen time.* ● But while I find words as easy to read on paper as on a screen, writing does not translate to the analog world so well. After years of typing on a keyboard, writing in a notebook is slow and hand-crampingly painful. With the relatively recent availability of maker-friendly electronic-paper displays of the sort used by devices like the Amazon Kindle, I fantasized about a device that would bring together the best of both worlds, combining a keyboard with the static, daylight-readable surface of an electronic-paper screen. A device that would make me feel the same about writing as I did about picking up a book, with no eyestrain, notifications, or YouTube distractions. And I could do better than the old dedicated word processors of the 1990s—I'd blast past them with 10 gigabytes of SD card storage, triple the display size, and crisper contrast and custom formatting. ● However, I quickly found that there were some serious issues I'd have to resolve before banging out a prototype. The first, and most glaring, was the dismal refresh rate of the e-paper screens I was buying—upwards of 2 seconds, if you used the original firmware. If you tried to go faster with custom firmware, you'd get persistent screen burn. It turned out that this was the major reason I hadn't seen my fantasy word processor already on the market—the few similar commercial attempts had received harsh user reviews regarding the lag between pressing a key and seeing a result.

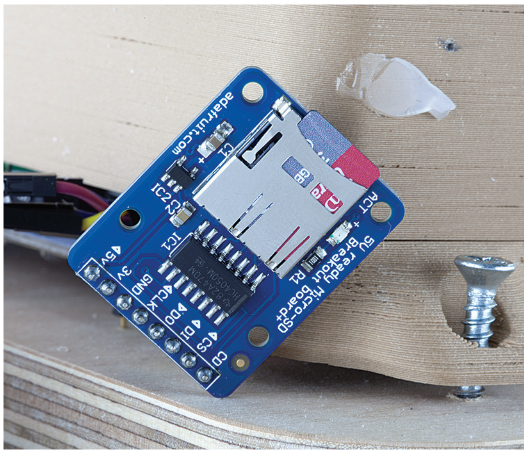
I decided to step back and take a different approach. After all, users need rapid updates only on the text they're directly editing. So why not add a standard character LCD underneath the main display? Then the bulk of the document would be crisp and clear on the main display. Add on a keyboard, SD card file system, and even a little printer to chug out my drafts for redlining with a pen, and I'd have myself an electronic-paper typewriter without the dragging input delays.

But this led to another problem: My device was now going to incorporate at least five discrete subsystems, each with a different communications protocol. On top of that, the dual-display setup would make it impractical to use an existing text editor like Nano or Emacs running on a Raspberry Pi. So I was also looking at writing my own text editor on a microcontroller from scratch. If I had to reinvent the word-processing wheel from device drivers on up, the project could take years. I needed to find a platform that would let me use some existing code.


I do not think there is any thrill that can go through the human heart like that felt by the inventor as he sees some creation of the brain unfolding to success. - Nikola Tesla.




I do not think there is any thrill that can go through the human heart like that felt by the inventor as he sees some creation of the brain unfolding to success. - Nikola Tesl-a.
This is the SPUDwrite!!
LCD pointer: 0
EPD pointer: 208



STRIPPED-DOWN SYSTEM: The prototype SPUDwrite chassis is made of wood and cardboard [top right]. It uses a fast LCD display for editing and an e-ink display for reading [above]. Text can be printed [top] or saved to an SD card [right]. A more polished assembly is intended for the next version, with a reflective LCD display.

An obvious source of such code is the Arduino ecosystem. However, the Atmel AVR chips that power most Arduino microcontrollers are too limited for word processing; for example, the Arduino Uno can barely fit two paragraphs of text in its 2-kilobyte RAM. I landed on Mbed instead, a firmware framework I'd used at a previous startup. Mbed has enough similarities to Arduino to make it possible to port over existing LCD and keyboard utility code. But instead of using Atmel AVR chips, Mbed is designed for the Arm architecture, allowing me to use an embedded system based on the far more powerful Cortex M4 family. A US $10 board from ST Microelectronics has enough RAM to hold a full chapter at a time, along with flash memory for longer documents. Mbed also had code for my typewriter's communication systems using I2C and SPI, and it included the crucial C++ libraries I needed to load and save on the SD card.

The project had a reasonable time frame again, and for the next six months I focused on getting the core features I needed into the text editor, such as selecting and creating files, handling the trade-off between the LCD and the e-paper displays, and formatting text. The printer—which outputs text onto a roll of 5.7-centimeter-wide thermal paper—added an especially goofy finish, and felt fitting to the theme: My device was a hack, but it was a *fun* hack, with lots of playfully unnecessary parts and work-arounds that nonetheless hit a basic level of functionality.

As it neared completion, I also determined the name: SPUDwrite, or Single Purpose User Device, writing edition. SPUDwrite could be the first of a whole line of purposefully anachronistic distraction-free devices—restricted yet full of surprises. SPUDphone, SPUDcode, SPUDtv.... Even as I finished coding the text editor, I was already thinking of the fixes I'd bring into version 2—swapping the transmissive LCD I'd ordered by mistake for a daylight-readable reflective one, adding a tactile scroll wheel and a huge red "print" button—more features, more work-arounds, more ways to improve my Internet-free little ecosystem.

But what was it really for? As I started actually using my SPUD for creative writing, I thought about where the incentive had come from. People who tried the SPUDwrite for themselves had sharply split reactions; some leaped on the concept excitedly, while others seemed almost offended at its restrictiveness compared with a modern PC. I found the split was most prominent with age: Those who liked it the least were parents and mentors who had never felt the impact of Internet addiction, while it got a better response from my peers and younger relatives who were resentful of the ever-present grip that their devices had on their lives.

I had built the project for myself. But I was surprised by the investment others had in the idea. Like me, they saw SPUD as an opportunity to escape and step back a bit, without abandoning their modern skill sets and base requirements for functionality, such as digital file storage. They wanted devices that held back on delivering everything in favor of delivering one thing, properly. They wanted the modern trappings of our online world—just strictly one at a time. **—LUCIAN COPELAND**

POST YOUR COMMENTS at https://spectrum.ieee.org/spudwrite0419