

# Impractical Computing

Vol 1

By Andrew Roach





## **TOC:**

**REACHING THE WORLD  
FROM WINDOWS 3.1**

**- PG. 4**

**REVIVING THE PORTAPAK**

**- PG. 13**

**DIY CONVERTABLE  
COMPUTING**

**-PG. 18**

# FREQUENT QUESTIONS

## What is this?

Impractical computing is an aperiodic periodical discussing my personal electronic choices, as I explore the edges of practicality and try to reach for a future of computing that is more Personal.

## Hello World?

I'm Andrew Roach. I run the Ellijay Makerspace (and ellijay coffeehouse, and analog revolution records, and retro.social, and mountain town toys, and mountain town radio, and expedition Sasquatch, and ... You get the idea), and I am a Site Reliability Engineer. My job is to make computers as reliable, reproducible, and Boring as possible, so I spend the vast majority of my free time undoing that, reaching in to the void and shaping something new.

## Why?

Because the embedded energy cost of a computer or any other piece of electronics is too high for us to abandon them just because something newer and more shiny was released. Because the best and most useful devices and methodologies are rarely the ones that are most commercially successful, and frankly, because best and

most useful are subjective, and computing should be personal.

A lot of the things I do are silly. Silly is good. Being silly removes the element of self-consciousness that often paralyzes creative people like me, and enables us to focus on doing the work. The artifice, the spectacle, and occasionally the challenge, perform a valuable service in lowering the stakes, freeing the mind, and enabling the creative process.

I spend all day, every day, interacting with computers. They are my gateway to my job, my friends, my hobbies, and many aspects of my life. They're also, frequently, deeply frustrating. It is important to me that I am able to experience something Different from work when I am using computers in my own time. So I make computers tangibly different.

---

# Reaching the World from Windows 3.1

I have an HP omnibook 300. It runs Windows 3.1 from ROM. It's about the coolest computer that was ever made:

- Almost exactly the size of a notebook (8.5x11.2x1.5" or so),
- instant on,
- no moving parts,
- stiff but reasonably satisfying keyboard (rubber domes, but decent ones),
- a non-backlit, sunlight readable, reflective TFT LCD
- it runs for **weeks of daily use on 4 rechargeable AA batteries.**

I could talk about the computer for a while, but I'd rather talk about how I have been using it and how I want to use it in the future. (Tools are, after all, only as interesting as what we use them for.)

My goal is to develop an offline first, asynchronous workflow that will allow me to interact with the world on my terms, at my pace, but won't prevent me from doing any of the things I find important.

## **Now**

It is very nearly a single tasking machine, even though the included Windows 3.1 is technically capable of limited cooperative multitasking, the difficult to upgrade 2MB of RAM on the machine ensures that I am rarely doing two things on it.

This has the unexpected benefit of making a great environment for writing, so I use the box to write a lot. I call it my ZineStation, because I use it to write the text that goes in to the zines I print, but it is also used for writing updates for my various websites, the scripts for Expedition Sasquatch and various other text projects. The things I write are saved to a compact flash card, which I can then stick in any number of other machines to post or email or process.

I also use the box to write simple tools for me to use, often in qBasic or an old version of Python. My first computer ran Windows 3.1, and I cut my teeth there, and even though I know there are better tools for many jobs, the tools available to me in Windows 3.1 are often Good Enough for me to do what it is that I need to do.

I've even managed to get a decent Desktop Publishing workflow going, and I produce a physical newsletter and a handful of pamphlets entirely on this machine.

## **Limitations**

There are a lot of ways that this little box is decidedly living in the past, but not all of them are bad. Some of them, though, have proven to be annoying at one level or another:

- No sound card, just a PC speaker
- No networking to speak of (we'll get to that.)
- Removable storage is limited to very small cards
- Useful and Interesting software for Windows 3.1 and DOS is getting hard and hard to find, and what does exist tends to target powerful 486s and full color VGA (I'm guilty of this myself!)
- No TLS or working SSH

And there are a few other things that I've already got workarounds for or have resolved to fix. I'll list them anyway:

- I have grown accustomed to having the ability to query wikipedia and various other reference sources at will, but the Omnibook doesn't currently allow this
- I frequently use the device to read RSS feeds, emails, etc. I have to remember to prepare these things in advance, and make a point to transfer them to the omnibook ahead of time. This means that, unless I manage to do it every morning "just in case", I'm often stuck with outdated info on the device
- Backups are entirely manual at the moment

- People stop and ask me about the box All The Time. People ask me lots of questions, that's not a big deal, I don't mind talking to people. But while I've been typing this, I've stopped to explain what the thing is Four Times, and this is in a place where I use this laptop several times a week (it's worse in new places.)

## **The future**

Let's go over the things I already have solutions for, and then talk about my pie in the sky stuff.

Current solutions:

## **Reference Material**

I mostly use my phone for this, because that's the obvious solution, but I also don't like this! For now, I have an offline copy of most of wikipedia on my phone, and an old copy of microsoft encarta. These are both partial solutions, but they'll do for now.

## **RSS and Email**

I came up with a pretty clever solution for this one, I think. I have a Raspberry Pi 0w. On reboot, it connects to any wifi network it recognizes in range (my home network, my phone's hotspot, the wifi at the local coffee shop) and updates my RSS feeds, downloads a bunch of emails based on some filters I threw together, uploads some files to my home server, and downloads some files from the same server.

Then it disconnects from the wifi network, and broadcasts it's own access point until I reboot it. The Pi no longer has an internet connection at that point, but I can connect to it



from the omnibook (more on that later) and grab the RSS feed or emails, or files that I marked to download, or I can drop files in for upload.

On my home server, I have a periodic job that sweeps the upload folder, and Does Stuff with the files. Making blog posts, posting to social media, sending emails, etc.

## **Backups**

I have a script that will upload backups to my pi0w, which it will push to my server at next reboot. It's not perfect, but it works for now.

## **Questions**

I wrote a quarter page FAQ, printed a bunch of them on my dot-matrix printer, and shoved them in my bag. When people stand expectantly at my table, motioning for me to remove my headphones, I hand them a copy. That's normally enough! Some of them even sign up for my newsletter.

## **Upcoming solutions!**

### **No sound card**

This ... I mean, it bothers me sometimes, but I mostly keep another music playing device on me so it's whatever.

I recently got an OPL3LPT, and I have a music player for DOS which makes use of it. I mostly don't carry it with me because that's not what this box is for, but I have a way to use it to play music if I really want to, and that's enough.

### **No networking to speak of**

So this isn't really fair. I have a Guru Modem, and a couple of other ESP8266 based "wifi modem" devices, and most of them work on this machine without any issues.

I have to power them from an external power source, which is kind of a pain in the ass, but I can connect to BBSs and telnet to my Raspberry Pi 0w, and technically it is networking.

It is networking! But it is limited, because no software for DOS supports modern encryption, so I'm stuck relying on WPA2 as my only source of security. (Not a great place to be!)

**Removable Storage is limited to very small cards**

The guru modem has a microSD card slot, and will allow me to transfer files on to and off of it with xmodem commands.

My Pi0 has a 128GB microSD card in it. I have a couple of spare CF cards in my bag, and a CF reader for my phone.

I also have a CD-RW drive that I can access from this box. It is Huge, and only works over parallel. It sits on my desk near my Dot Matrix printer. If I'm using this box at home, it works. I mostly don't use this box at home, so utility is limited.

**Useful and Interesting Software for Windows 3.1 and DOS is getting hard and hard to find**

This is a huge problem. I'm doing my part to mitigate it by writing about the software that I use, to provide mirrors of that software, and (eventually) to write new software for DOS, but this is a drop in the bucket compared to the stuff that is just Missing. Document and archive your art, friends.

## **No TLS or working SSH**

This is the reason that I say there's not really any networking to speak of. Have you tried to use the web without TLS today? It's almost impossible. There are hardly any DOS web browsers with SSL, and none that I'm aware of with TLS. There are two SSH clients, but neither of them support modern encryption methods.

I have a couple of silly (and some not so silly) ideas about how to fix this. One option would be to take SSH2DOS and backport new encryption schemes in to it. This is probably the correct solution.

The two things I'm most likely to attempt:

1. replace my ESP8266 based wifi modem with a raspberry pi 0w and a serial cable. Spawn a getty on the serial port of the pi, and log in to it as if it was a dialup shell account. I could also have it serve internet over serial via SLIRP, and even configure it to act as an SSL/TLS/JS/CSS stripping proxy. That would give me better solutions for backups, and real time internet, etc. It would be an excellent solution for making the Omnibook act like a modern computer (but I'm certain that's Not what I want.)

2. configure a pi or my phone to provide a torsox proxy that my guru modem will connect to, and telnet over tor to an onion address running on a home machine. Because I'm connecting to a hidden service (and there is no exit node, as a result) this is E2E encrypted. It's just ... kind of ridiculous, and still adds more hardware and more complexity to my set up.

## **Other considerations**

I like the idea of having some kind of networking available for the Omnibook, but it isn't actually strictly necessary. I want to be able to do the following with it:

- write
- code
- read and respond to emails
- read and respond to fediverse posts
- read RSS feeds
- write blog posts
- consult reference sources
- request data to be collected

Technically, I could do all of those things with removable storage and some scripts running on another computer!

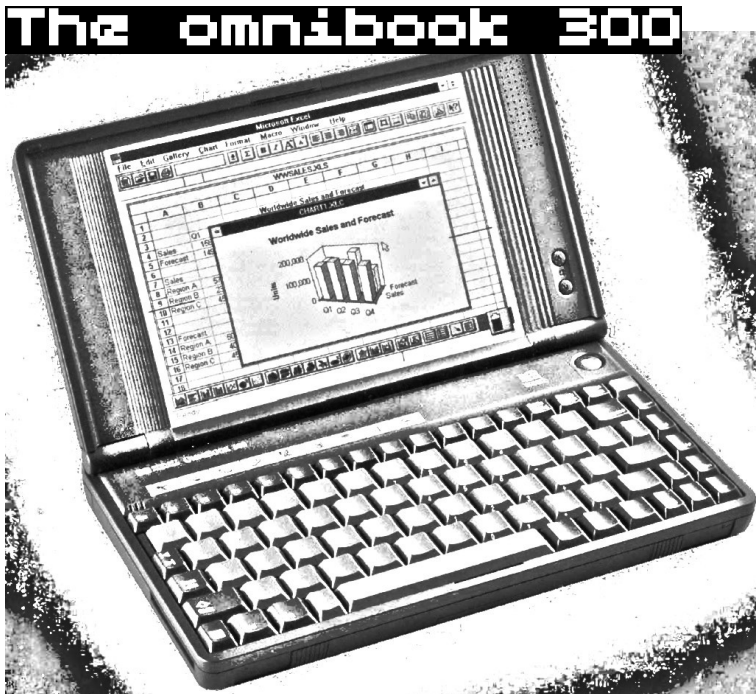
My existing Raspberry Pi solution could be updated to act on files from a Compact Flash card connected over USB, rather than files transmitted via ftp and commands issued via telnet.

It's entirely possible that the best solution for this problem will be to embrace the fully asynchronous nature of the way I currently interact with the omnibook, rather than

trying to get just a little synchronous/real time networking in to my set-up.

That being said, my goal is to have a mostly asynchronous workflow that doesn't prevent me from doing any of the things I need to do and sometimes the thing I need to do is to interact with other humans in something close to real time, so something just a little synchronous is probably in the cards.

This solution is still evolving, but the Omnibook already solves so many of the problems I encounter with computers on a regular basis that I am certain I'll solve them eventually.



# Revisiting the [REDACTED] Portapak

Prior to the advent of VHS and Betamax, most home video cameras were “Portapaks”. These cameras were usually small, agile little things with a tether back to a shoulder slung recording unit. The recording unit used reel-to-reel video tape, and provided power to the camera.

The batteries in these recorders are long dead and mostly no longer manufactured. The recorders are fragile and failing. The tapes are unavailable. Most recorders lack a dedicated video out.

Basically, there are two decades worth of analog video cameras that are functionally useless right now, but which can and should be brought back to life. Remember, the energy costs to make these things were huge, and they can still work!

So! We’re building a little shoulder slung case that houses:

- a 12V battery
- some RCA jacks,
- 5V out via USB

It provides power to the camera, lets us get audio and video out, and connects A/V and power to a recording monitor, DVR, or small computer.

This era of camera has 4 or 5 common plugs that were all essentially electrically identical, and we can 3D print pass through adapters from the most common manufacturer specific plugs back to the standard EIAJ-1 connector.

This means that, by the time we're done, we'll have a little shoulder slung device, (which can be assembled with off the shelf and 3D printed parts,) which will enable more or less any camera made between 1968 and sometime around 1983, and many cameras that were produced as late as 1988 (the portapak format held on long in tot he VHS and Betamax era because it enabled Longer Recording Times) to be used, fully mobile, to capture digital video or even live stream.

“Repair, Reuse, Make Something Ridiculous” is our mantra.

So we're doing it. Right now, we have a fully functional prototype that has been tested and proven functional with cameras from 1968 to 1984 (because that's what I happen to have on hand.) The end result won't win any awards for fidelity or picture quality, but it works!

Right now, the entire setup is:

1. A 12v Battery
2. Some RCA jacks
3. A 5v battery
4. A standard definition recording monitor/DVR\
5. Some wires to connect it all together

and it lives in a mid 70s camera bag.

My next steps are:

1. Mount the plug on the outside of the case.
2. Add a power switch
3. Add fuses
4. Add a volt-meter or some other way to monitor charge, and a recharge circuit so the battery can be charged in place rather than swapped.
5. Route the AV jacks to a switch (or a splitter?) that lets me have RCA outs on the outside of the case and also to capture internally.
6. Buck converter for the DVR/Raspberry Pi so I can remove the 5V battery
7. Rig up a raspberry pi to Record or livestream based on the position of a switch and input from the remote recording trigger on the camera and replace the pi with it.
8. Mount that switch.
9. Become Eddison Carter.

Is it silly? Of course. The spectacle is the point.

Imagine a panel van full of weird maker space nerds rolling up on an event, and walking around with these huge, unwieldy cameras.

Picture it! Us in our Hackers.Town gear, our inscrutable home made t-shirts, our collective hair Impressive and Intimidating for various reasons, wielding these



anachronistic, highly visible cameras. It is a Spectacular image.

The spectacle enables a kind of filmmaking that would not otherwise be possible.

Sure, I could just pop my phone out and shoot a video and be invisible, and sometimes that's what I want to do.

Sure, I could take one of our fancy cameras out shoot a modern video in a very visible and serious and professional way, and sometimes that's what I want to do.

Sometimes, though, what we want is to be highly visible, but in a silly and mildly disarming way.

Being kind of silly and highly visible puts people at ease, breaks them out of their shells, makes them feel less self-conscious about being on camera.

This is helpful in a documentary setting, but it's frankly even more helpful in a narrative setting when you're working with amateur actors.

The cheap cameras and bad video footage also temper the expectations of our audience in a way that I think will really serve us well.

While there is a lot of work left to do to make this solution slightly less cumbersome for others to implement, and safer to use in the long term, what we have now is entirely functional, and can be assembled really cheaply.

I'm calling that a win.

---

The original Portapak



# DIY convertible computing

This is an article about building your own Desktop/Portable computer. I'm presenting several options for classes of hardware and methods of interaction. This article is specifically about hardware, I might do another one about software. I'm also trying to meet some specific requirements (low weight, long battery life, ability to run emacs above all else, networking not needed), which means I will be making some suggestions and observations here that differ from what I might say in more general circumstances.

Alright, now that the intro is out of the way, a friend said to me recently:

```
I want a small computer that can sit  
at my desktop and get plugged into  
power and a hdmi'd to a big monitor  
and its wired keyboard and mouse, and  
then I can unplug it and use it off a  
battery with bluetooth keyboard and  
hdmi'd to a small screen.
```

This was at the end of a pretty long discussion about tablets and laptops and the death of the netbook before ultimately alighting on Raspberry Pi, and other DIY options. I felt like this is something other people might also like, and that it merited some longer observations, so here we are. I see three basic methods by which this could

be achieved: 1) fully modular, 2) fully portable (the [NOD E 0 terminal](#) method), 3) Tablet Style.

## **Fully Modular**

The modular method is probably the easiest of the three to understand, and is the closest to the design that my friend requested. For this method you need one of each of the following components:

- A Single Board (SoC) computer
- A case for the computer
- One or more power supplies
- A USB hub, mouse, keyboard, and monitor (for desktop mode)
- A smaller mouse, keyboard, and monitor (for portable mode)
- A battery (preferably one with pass through charging)
- Storage (If you're using a raspberry pi or similar, this will likely be a microSD card)
- Some kind of rig to hold all the portable bits

The beauty of this design is that it can be completely modular. If you don't like the mouse or the keyboard or the screen, sell it and buy another. Each piece can be used on it's own. This modularity comes at the cost of portability, and possibly at the addition of weight.

There are a couple of options available for each piece of hardware mentioned above, and I won't discuss the pros and cons of every USB hub or Keyboard on the market, but I do want to spend a few cycles on the portable monitor, battery, and SoC.

## **SOC**

Depending on your budget, your region, and your needs, you might pick from dozens of single board computers from the Odroid line, RockPro, Pine64, the Atomic Pi, the Orange Pi, the Chip (R.I.P.) and of course, the Raspberry Pi. For the purposes of this article, I'm going to focus on The various Raspberry Pi as they are the most commonly available, and often the most well documented. Your needs will vary, and you should feel free to disregard this advice.

## **Raspberry Pi 0W**

The 0w has a couple of benefits: Low power consumption, low price (\$10 MSRP), small size, built in wifi and bluetooth, and low heat output.

If you don't need the wifi and bluetooth you can also look at the 0, which can be found for as little as \$5. The 0W isn't perfect, though. It's slow, and it does not have any full sized ports, requiring an adapter to connect a full sized USB or HDMI cable. These adapters are commonly available, and not terribly expensive (less than \$5 each) but they do add additional complexity and cost, undercutting the low price of the 0 and making the 3 or 4 seem more attractive.

A Pi 0W with SD card, power supply, case, and the requisite adapters will cost around \$30, assuming you have access to a Microcenter or another source for low cost components. (Recently, the raspberry pi foundation released the 0W 2, which is a more attractive choice, but didn't exist when I wrote this originally.)

## **Raspberry Pi 4**

The Pi 4 is new and fancy! It's much, much faster than previous models, and supports dual monitors and up to 4k resolutions. There's wifi, bluetooth, USB 3, and much faster ethernet. The base model is \$35, but it's available in more expensive packages that include more RAM. In order to support dual monitors, the Pi 4 makes a similar concession to the Pi 0, it uses a much smaller HDMI port (in this case, even smaller than on the 0) meaning you'll need a special cable or an adapter to use it with a standard monitor.

Additionally, and importantly, the 4 is reported to run pretty hot, with the official guidelines calling for at least a heatsink, and many people suggesting that active cooling may be needed.

A Pi 4 with SD card, power supply, case, and video adapter will cost around \$55 on the low end.

## **Raspberry Pi 3+ and other Pi models**

The Pi 3+ is my personal choice for projects like this. They also have an MSRP of \$35, but are frequently available for less (at least, in my area.) They have built in wifi and bluetooth. They run a little hot, and need a little more power than a 0, but not as hot or as hungry as a 4. Importantly, they have a full sized HDMI port, and 4 full sized USB 2 ports, and do not require adapters.

A Pi 3 with SD card, power supply, and case can be had for around \$45.

## Batteries

There are several options for powering a pi on the go. They fall in to, broadly, three categories:

1. basic cellphone backup batteries,
2. nice cellphone backup batteries,
3. internal batteries + power boost circuits.

I'm not going to spill many bits here:

- Basic batteries don't have charge pass through, so you can't charge the battery while it is powering the device, and you must shut it down before switching from wall power to battery power or vice versa. This is a bad option. It is much more of a hassle than it sounds like it would be, and it already sounds like a hassle. Don't do this to yourself.
- Nice batteries do have charge pass through, so you can charge the battery while it is powering the device, and you can plug in/un-plug the battery without shutting down the device. Many devices that offer this don't advertise it, so it can be hard to find a reasonably priced battery with pass through. This is generally the best option. Do your research.
- Internal batteries are what it says on the tin. You get something like a [Power Boost](#) and hook up a battery to it. The power boost up converts the voltage from the battery to match what the Pi needs and handles charge pass through. If you're doing anything other than a fully modular design, I

recommend seriously considering this route, but it does add cost and complexity (the charge circuit alone is usually \$20. Paying \$20 for the power boost plus another \$30 for the battery seems kind of silly if you're going to connect it to a \$5 computer.)

## **Monitor**

There are Many monitors, but they broadly fall in to four categories: High-res HDMI, Low-res HDMI, Composite, SPI. Unless you have a good reason to, don't bother with composite monitors or SPI monitors. The first is *very* low res and often full of chromatic aberrations, the second updates the screen pretty slowly. The benefits of both are entirely lost when working on a fully modular design.

As for the HDMI options, they are available in sizes from 5" up to 11", and resolutions from from 800x480 up to a full 1080p. They vary in cost depending on where you buy, and who manufactured the display, but I would expect to pay between \$40 and \$80 for a good monitor without touch at any size under 10 inches, and double that for one above inches. These can usually be powered from the USB port on the pi directly. I'm a fan of Adafruits HDMI4PI line, but there are other cheaper options available as well.

Resistive and capacitive touch screen models are also available, but I wouldn't bother with one of those for this use case.



## To Sum Up!

- A Single Board (SoC) computer (\$10 - \$50)
- A case for the computer (\$5 - \$10)
- One or more power supplies (\$5 - \$10 each)
- A USB hub, mouse, keyboard, and monitor (for desktop mode) *I am working under the assumption that you already have these, or will be acquiring them separately, and will only be discussing the portable operation. This stuff is expensive*
- Adapters (if needed) (up to \$15)
- A smaller keyboard (\$10 +)
- A monitor (for portable mode) (\$40 - \$80)
- A battery (preferably one with pass through charging) (\$30+)
- Storage (If you're using a raspberry pi or similar, this will likely be a microSD card) (\$5 or less from microcenter)
- Some kind of rig to hold all the portable bits (up to you)

for a base price of \$105 - \$245

## Fully Portable

I first saw this [a few years ago from n-o-d-e](#) (see QR code). The basic idea is to stick a Pi 0W, a battery with a charge circuit, and a [PiTFT Plus 480x320](#) in to a sliding BT keyboard phone case.



The end result is surprisingly slick and usable. I built one of my own shortly after seeing this article, although I sadly don't have any pictures of it, and I've long since deconstructed it for parts. I don't love the SPI screen, but it makes for a really slick finished product that is actually pretty fun to use.

I didn't have a 3D printer when I built mine, so I encased the top in some scrap cardboard and some duct tape. It worked well enough, and I carried it with me for a few weeks, before realizing that it met the needs I had in previous years when I was working as a web developer, but it did not meet my needs at the time.

iPhone 5 keyboard cases are hard to find these days, but any sliding BT keyboard case will do, and they can be found for less than \$20 all day.

Total for this model: \$120 (including the USB and HDMI adapters you'll need for desktop operation.)

## **Tablet style**

This is probably the most appealing method on first glance, and I've toyed with it on more than one occasion, but I've been consistently disappointed with the results, mostly on a software level. The basics: A pi with the needed bits (\$30 - \$55) an [official RasPi foundation touch screen](#)(\$80) and a case (several options, \$10+). You add to that a wireless keyboard, and that's it!

Of course, you'll probably want a battery too. See Batteries in the modular section for thoughts on that, and add it to your \$120.

You end up with a weirdly thick and kind of fragile tablet running at 800x480 with a pretty nice capacitive touch screen.

Depending on what software you end up using on it, this actually works pretty well. I used a cellphone backup battery with charging pass through and 3D printed a little stand for the touch screen that ended up looking something like an Apple IIc. In practice, though, I found that 800 x 480 was too low of a resolution to use for most graphical applications, and that I could get a better non-touch monitor for a text terminal for the same amount of money, so that's the route I'm going next.

## **In Practice**

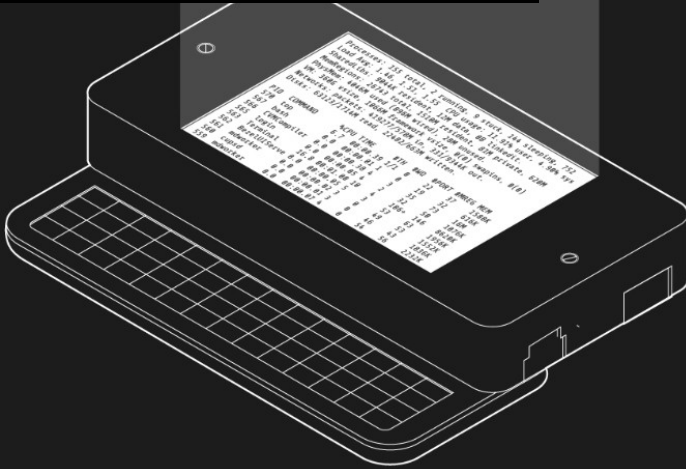
The Pi line, through the pi 3 at least, is not powerful enough for a modern desktop and web browser in most circumstances. If you want to run a CLI/TUI environment, it'll do just fine. If you're willing to go for lighter desktop apps, it can be very snappy. If you want to watch youtube, expect to wait often.

The Pi 4, can deliver something closer to a modern desktop experience. I've used mine for everything from basic web browsing to video editing. It works well enough.

For my modular pi rig (which features a pi0, backup battery with charging pass through, and a 10" HDMI monitor clamped to a mechanical keyboard) I mostly use it to access other machines on my network over SSH, or to access email (via Mutt) and a text editor. I do this when real computers get to be overwhelming. I've used it with

omxplayer to watch videos and cmus for music. I've used it as a mobile server and I frequently use it as a serial gateway for my vintage computers like my omnibook 300. It does what it's told, always. Sometimes, it's just a little slow.

**Original illustration for  
the n-o-d-e terminal**







Learn the wrong way to  
use your computer in  
this aperiodic  
periodical by Andrew  
Roach from

The Ellijay MakerSpace

