Automating My Office

Jason Axelson
ExMex 2025, Austin TX

What we're talking about today

Here's my home office

It's a small nook with a standing desk

Let's automate it!

About me

- Developing in Elixir professionally since 2016
- Member of the Scenic core team
- Maintainer of many libraries
  - MainProxy, ExSync, DepViz, DataTracer, PasswordValidator, etc

Aspiring Yak Shaver

@axelson on GitHub

@axelson@fosstodon.org

Felt

Senior Software Engineer at Felt

The best place to make maps on the Internet

Agenda

- Exposition
- The cast and characters
- The main stage
- Wrap-up and reflections

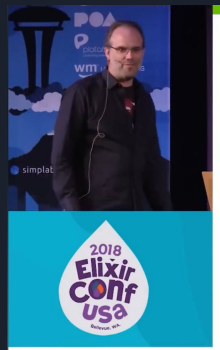
Exposition

How did we get here?

- **Exposition** ←
- The cast and characters
- The main stage
- Wrap-up and reflections

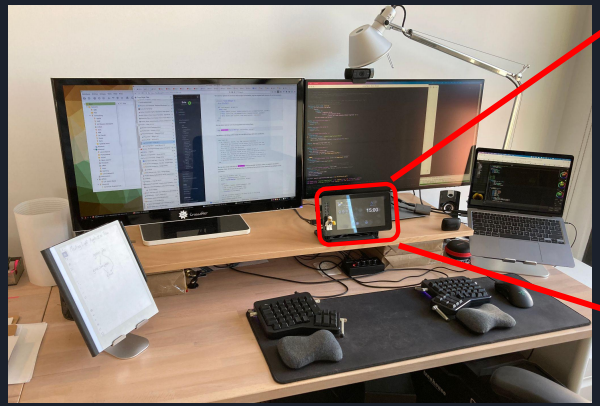
Our story begins in 2018

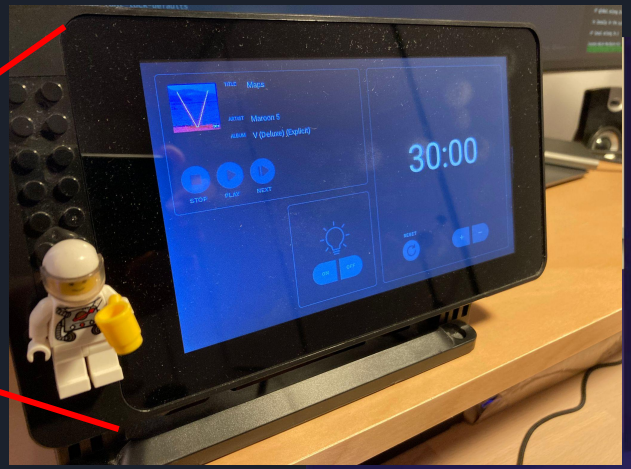
- My first ElixirConf 🥳
- I attended the Nerves training
  - RPI 3B+
  - Official 7-inch touch screen
- Scenic was officially released



2021 - My Scenic Companion

- Another ElixirConf, this time in Austin
- My first Elixir conference presentation 🥳

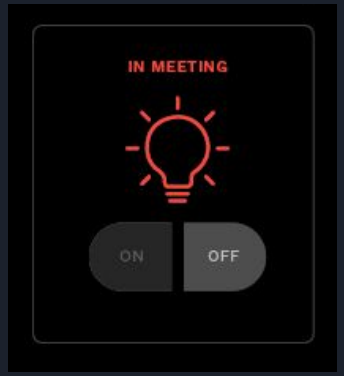


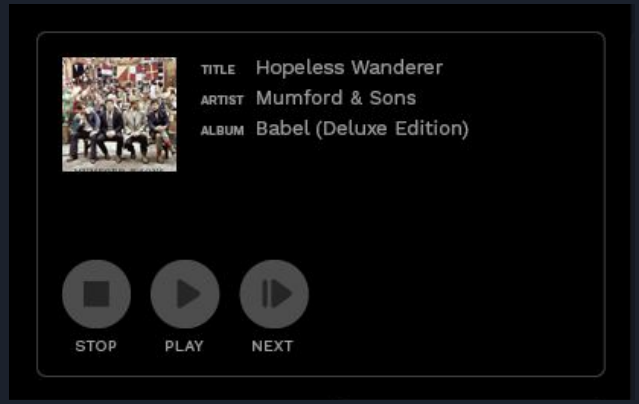


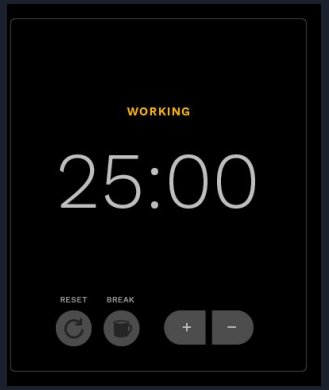
2021 - My Scenic Companion

Functionality in 2021

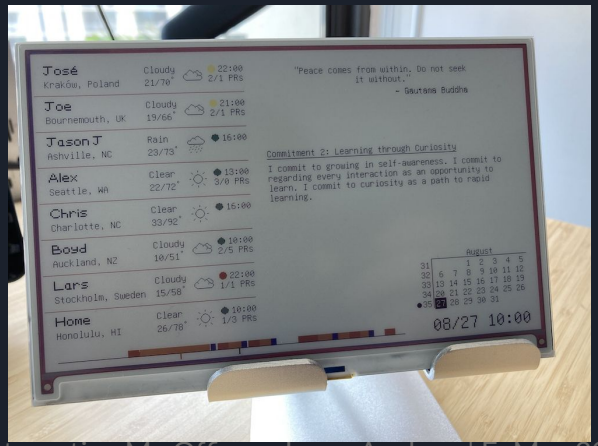
- Govee LED Light - indicates when I'm in a meeting
- Pandora interface - shows my playing music
- Pomodoro interface - tracks focused work sessions

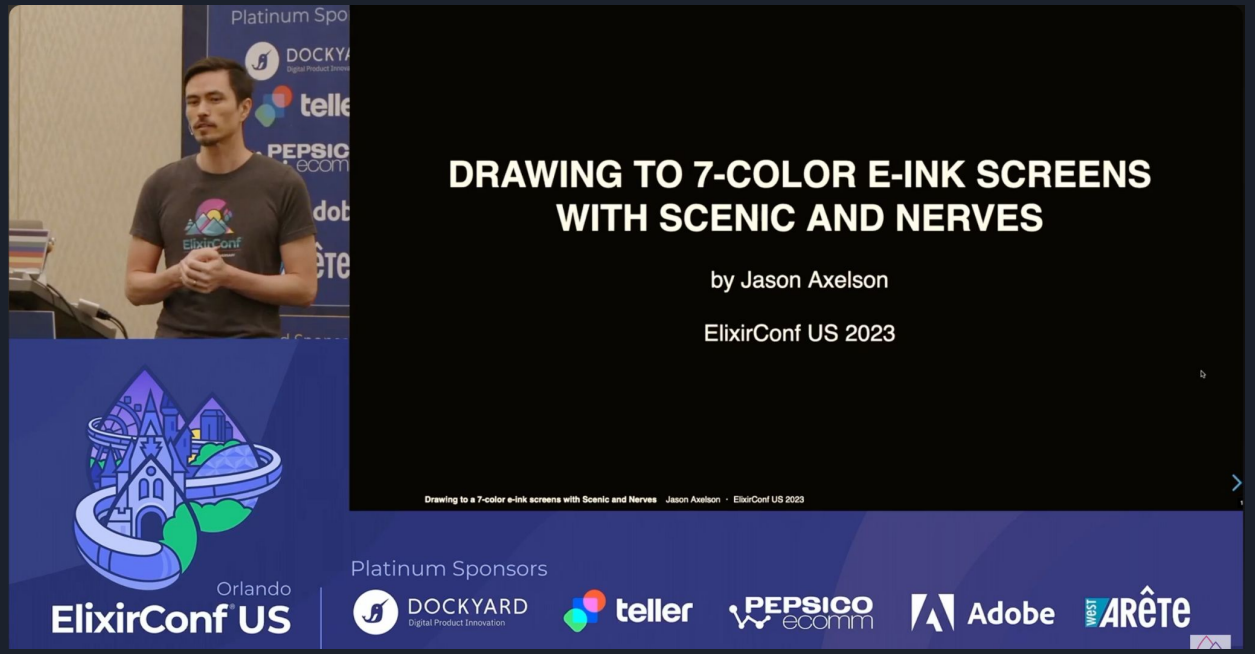






2023 - e-ink


José
Kraków, Poland
Cloudy
21/70°
22:00
2/1 PRs
"Peace comes from within. Do not seek it without."
- Gautama Buddha
Joe
Bournemouth, UK
Cloudy
19/66°
21:00
2/1 PRs
Jason J
Asheville, NC
Rain
23/73°
16:00
Commitment 2: Learning through Curiosity
I commit to growing in self-awareness. I commit to regarding every interaction as an opportunity to learn. I commit to curiosity as a path to rapid learning.
Alex
Seattle, WA
Clear
22/72°
13:00
3/0 PRs
Chris
Charlotte, NC
Clear
33/92°
15:00
Boyd
Auckland, NZ
Cloudy
10/51°
10:00
2/5 PRs
Lars
Stockholm, Sweden
Cloudy
15/58°
22:00
1/1 PRs
Home
Honolulu, HI
Clear
26/78°
10:00
1/3 PRs
August
08/27 10:00


DRAWING TO 7-COLOR E-INK SCREENS WITH SCENIC AND NERVES
by Jason Axelson
ElixirConf US 2023
Drawing to a 7-color e-ink screens with Scenic and Nerves Jason Axelson · ElixirConf US 2023
Orlando
ElixirConf US
Platinum Sponsors
DOCKYARD
Digital Product Innovation
teller
PEPSICO ecomm
Adobe
ARÊTE

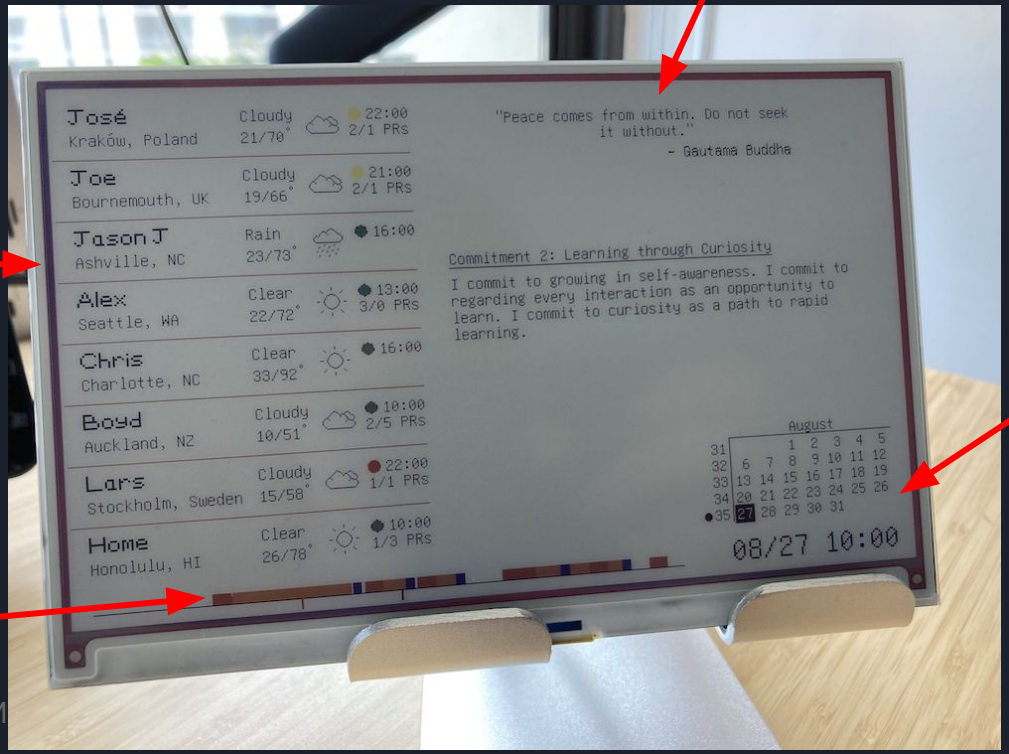
My e-ink screen

Quotes of the day

Info about co-workers

Calendar + Time

Pomodoro visualization



What's happened since 2023?

- I've been continuously maintaining and improving my setup
- My e-ink screen is neglected and dead 💀
  - now I run the software on my desktop, mainly for pomodoro visualization
- I've used my Scenic Companion basically every day since 2021
  - It's also learned minor calendar functionality
- I've changed my monitor setup
- I've added a bunch of hardware ripe for automation

The cast and characters

What are we working with here?

- Exposition
- **The cast and characters** ←
- The main stage
- Wrap-up and reflections

My hardware

Computer-ish

- Primary and secondary monitors
- Two laptops, one dock
- Desktop
- Keyboard and mouse
- USB switcher
- USB microphone

Hardware-ish

- Keylight - Meeting light
- Govee LED light and LED strip
  - Mood lighting
- Office lights
- Macro keypad
- Standing desk

My computer setup

Main monitor

Secondary monitor

Laptop

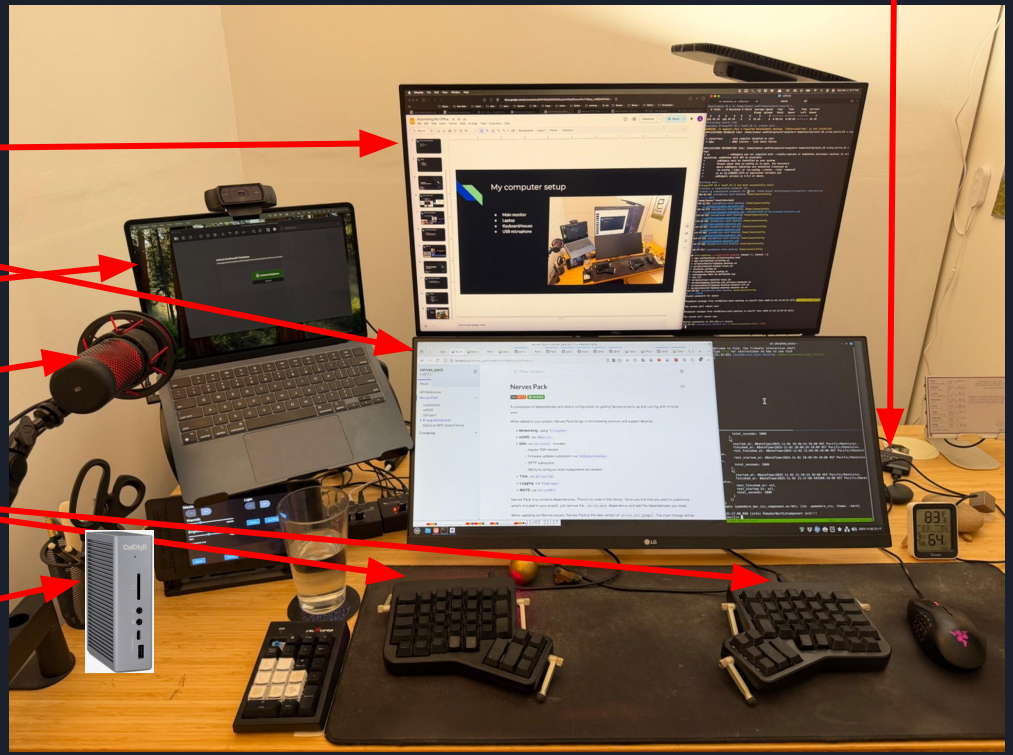
USB microphone

Keyboard/mouse

My desktop (not pictured)

Laptop Dock

USB switcher



Not pictured: lots and lots of wires

My hardware

Govee LED strip

Govee LED light

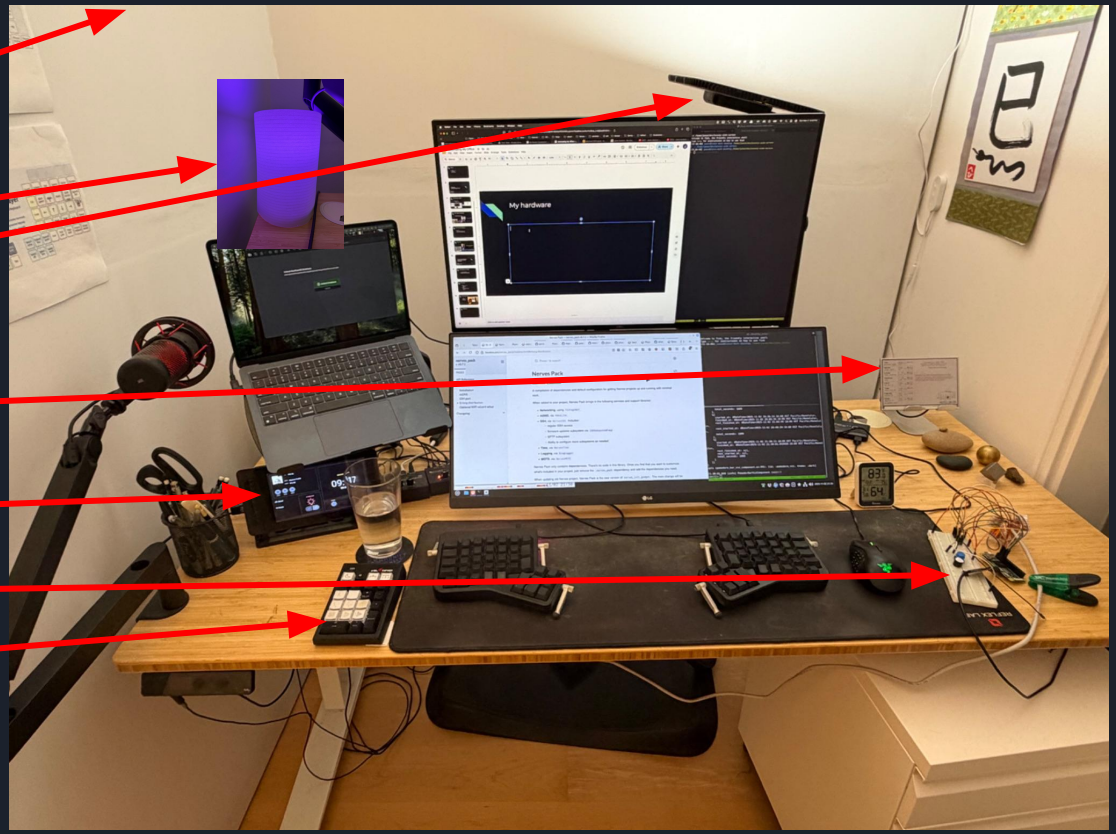
Keylight

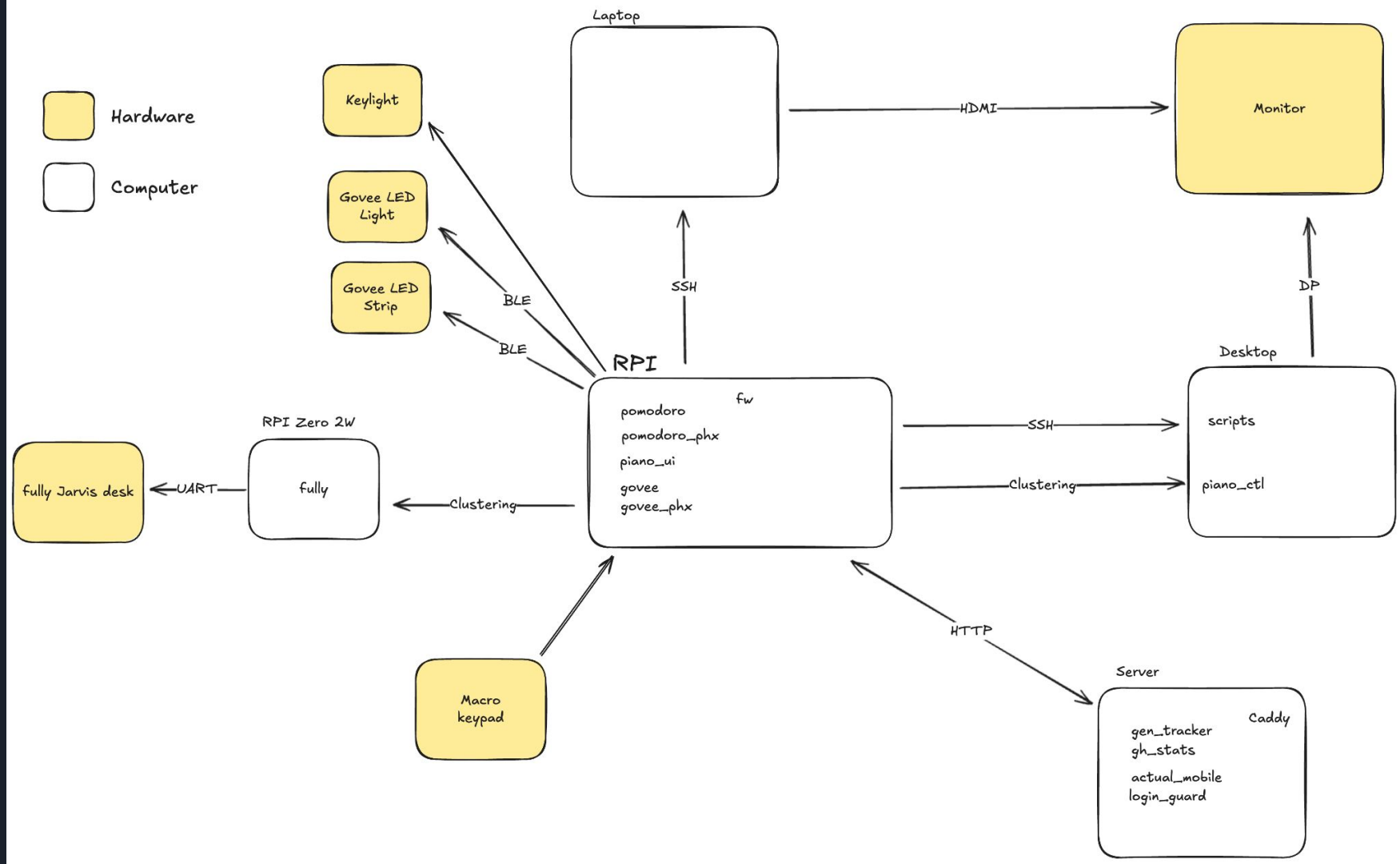
e-ink screen 💀

RPI 3B+ w/Touch screen

RPI Zero 2W w/breadboard

Macro Keypad




Hardware
Computer
Laptop
Keylight
Govee LED Light
Govee LED Strip
BLE
BLE
SSH
HDMI
Monitor
DP
RPI
fw
pomodoro
pomodoro_phx
piano_ui
govee
govee_phx
Desktop
scripts
piano_ctl
SSH
Clustering
RPI Zero 2W
fully
fully Jarvis desk
UART
Clustering
Macro keypad
HTTP
Server
Caddy
gen_tracker
gh_stats
actual_mobile
login_guard

The main stage

Get to the automations already!

- Exposition
- The cast and characters
- **The main stage** ←
- Wrap-up and reflections

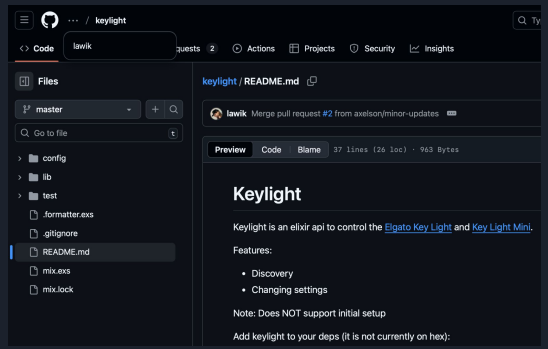
Who here has a light that they use for online meetings?

Who has to reach up and press a button on their meeting light to turn it on?

Let's fix that

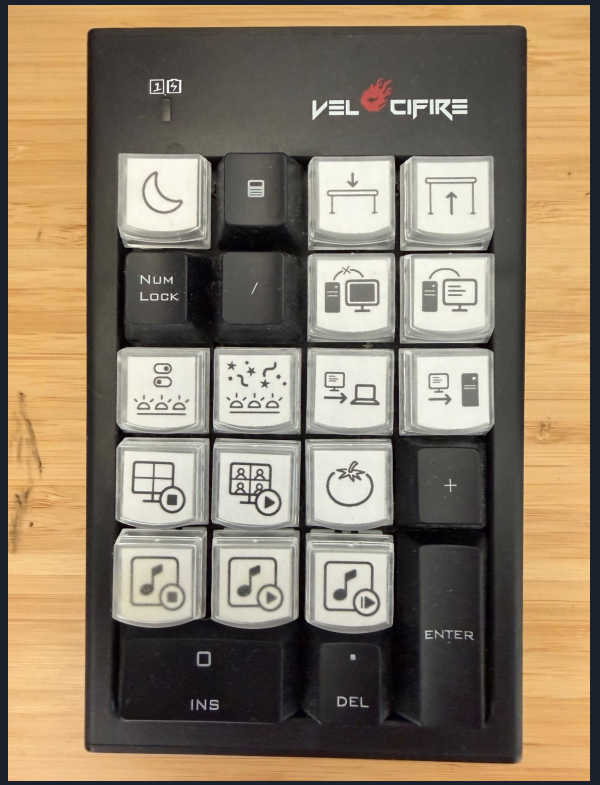
Controlling my meeting light
Elgato KeyLight

- There's an elixir library (thank you Lars!)



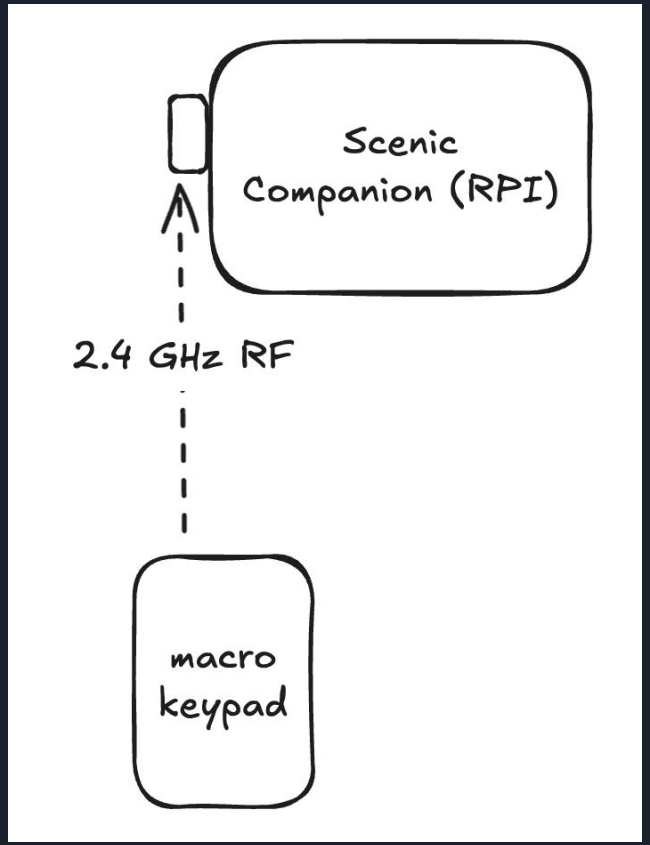
My macro keypad

- Why?
  - Easier to press than the touch-screen
  - Quick and responsive
  - Muscle memory
  - Always on and ready
- Clicky mechanical keys!



My macro keypad

- Keypad is hooked up to RPI via USB
- Scenic program handles key presses

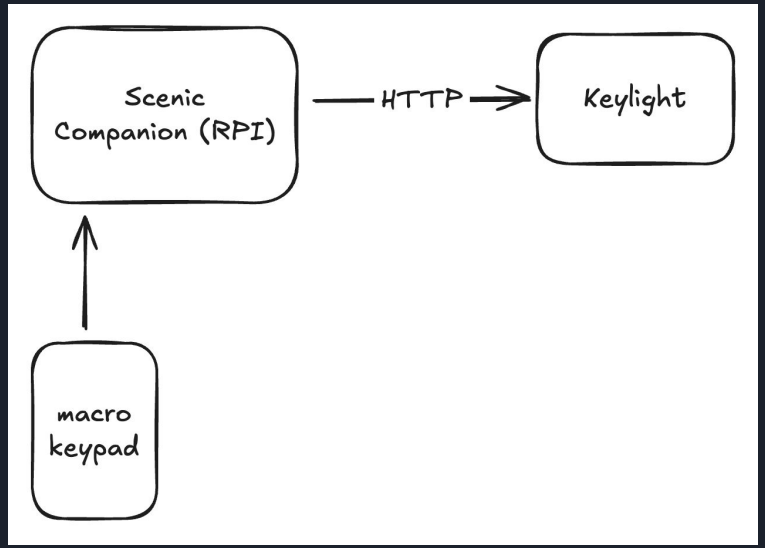


My macro keypad

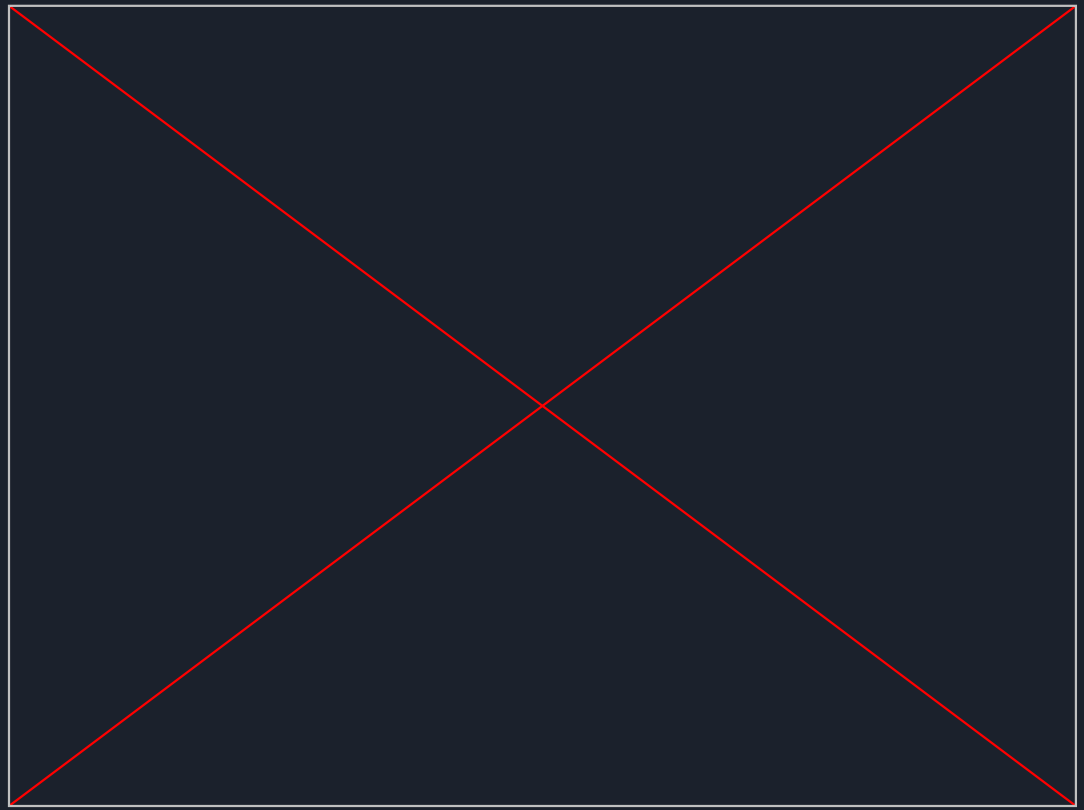
```elixir
def init(_opts) do
  InputEvent.enumerate()
  |> Enum.filter(fn {_dev, %{name: name}} ->
    String.starts_with?(name, "MOSART Semi. 2.4G Keyboard Mouse")
  end)
  # We grab the input because we don't want scenic to see the input as well
  |> Enum.each(fn {dev, _} -> InputEvent.start_link(path: dev, grab: true) end)

  {:ok, []}
end

def handle_info({:input_event, _dev, [_info, {:ev_key, key, 1}]}, scene) do
  case key do
    :key_kp1 ->
      PianoUi.remote_cmd(:stop)

    :key_kp2 ->
      PianoUi.remote_cmd(:play)
```

Controlling my Keylight

```
devices = %{
  ip => Keylight.build_by_ip(ip)
}
Keylight.on(devices)
```



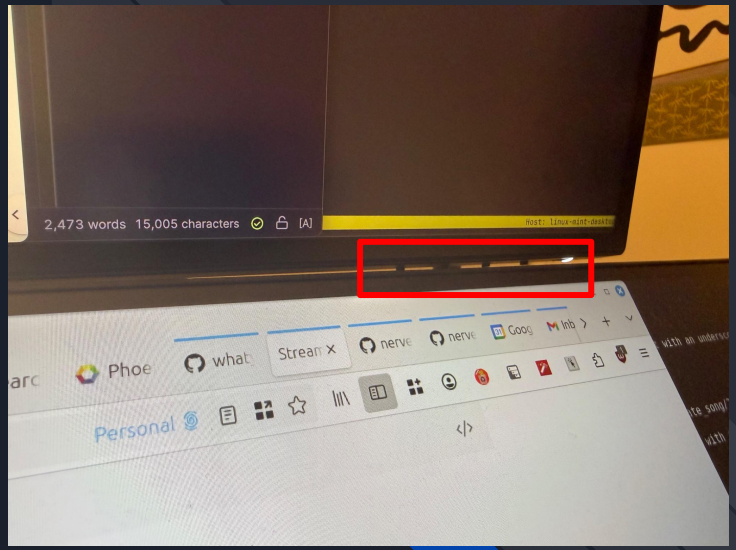
In Action



I have a problem:
One main monitor but two computers

Yes, it's fair to call this self-inflicted

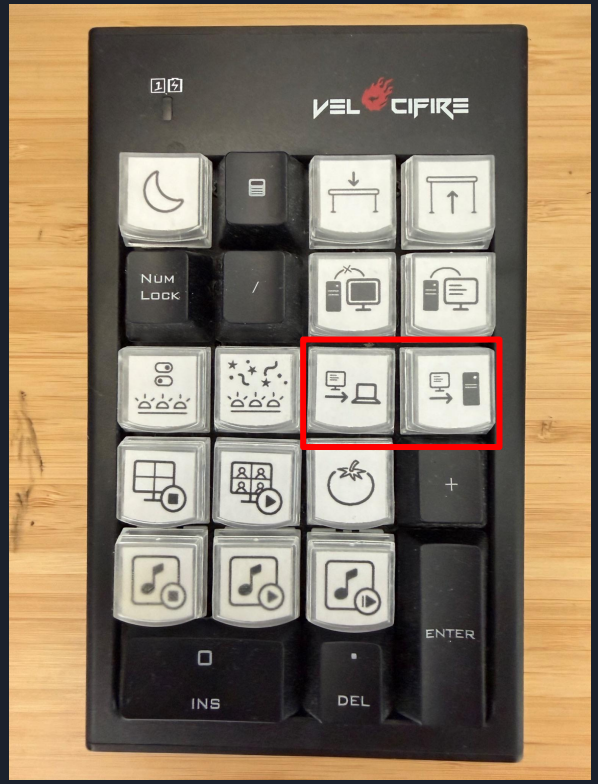
Who would want to press these tiny buttons?



Solution: Press a button to control the monitor

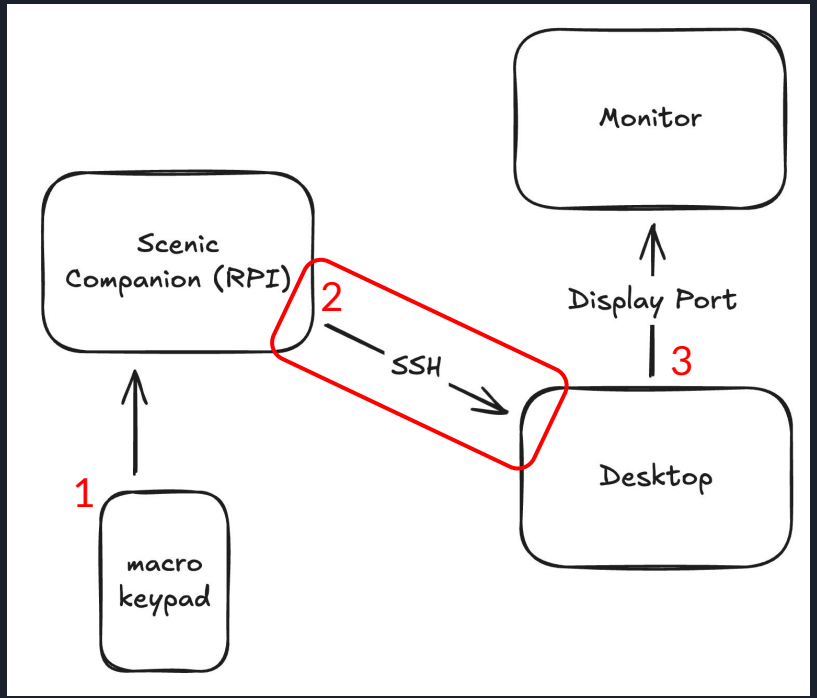
Controlling my monitor

- Change the input on the monitor
- Uses `ddcutil` to change the monitor settings
  - Sends DDC commands over I2C
  - without pressing a button on the monitor



How ddcutil is run

1. Button pressed on macro keypad
2. Scenic companion uses SSH to run command on Desktop
3. Desktop runs ddcutil to modify the monitor
    - specifically `ddcutil -d 2 setvcp 0x60 0x1b` (or 0x0f)



SSH from RPI to Desktop

- RPI uses `:ssh` to send the SSH command
  - RPI has a private key that the Desktop trusts
  - (with labzero/ssh_client_key_api)
- Desktop trusts the key by placing it in `~/.ssh/authorized_keys`

Entry looks like:

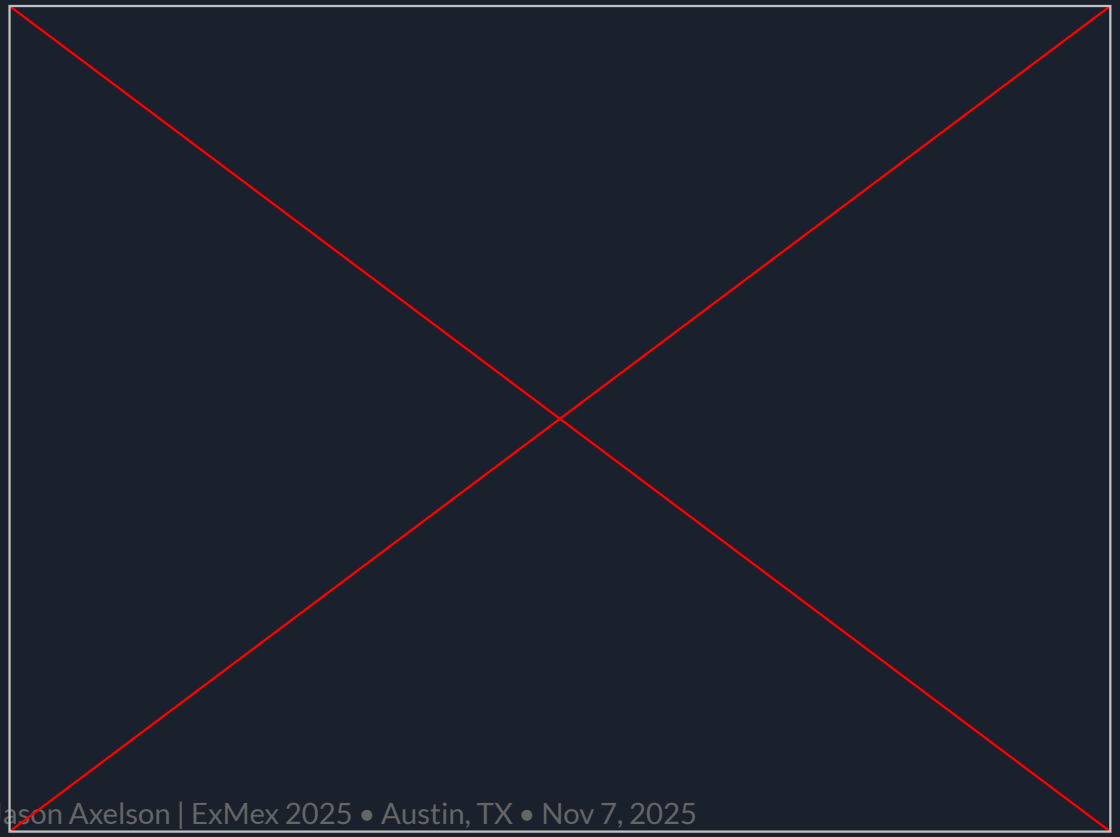
```
command="/home/jason/config/scripts/desktop_ssh_allowed_commands.sh"
ssh-ed25519 <MY_PUBLIC_KEY>
```

This is an extensible way to allow running a limited set of commands!

desktop_ssh_allowed_commands.sh

```
COMMAND=$(echo "$SSH_ORIGINAL_COMMAND" | awk '{print $1}')
ARGS=$(echo "$SSH_ORIGINAL_COMMAND" | awk '{$1=""; print substr($0,2)}')

case "$COMMAND" in
    "TurnOnMonitor")
        ~/config/scripts/monitor/desktop-monitor-on.sh
        ;;
    "TurnOffMonitor")
        ~/config/scripts/monitor/desktop-monitor-off.sh
        ;;
    "SwitchMonitorToDock")
        ~/config/scripts/monitor/switch-monitor-to-dock.sh
        ;;
    "SwitchMonitorToDesktop")
        ~/config/scripts/monitor/switch-monitor-to-desktop.sh
        ;;


    *)
        echo "Unrecognized command: $SSH_ORIGINAL_COMMAND" >> $ERROR_LOG
        exit 1
        ;;
esac
```

Putting it all together



Wait, what about that RPI Zero with all the wires?

Also known as the TDD section (Talk-Driven-Development)

Controlling my standing desk with Nerves!

- I have a "fully" Jarvis standing desk
- The handset interface works, but it is finicky
- Sometimes you have to press the screen to wake it up 😱
- We can do better!

This is what we're replacing

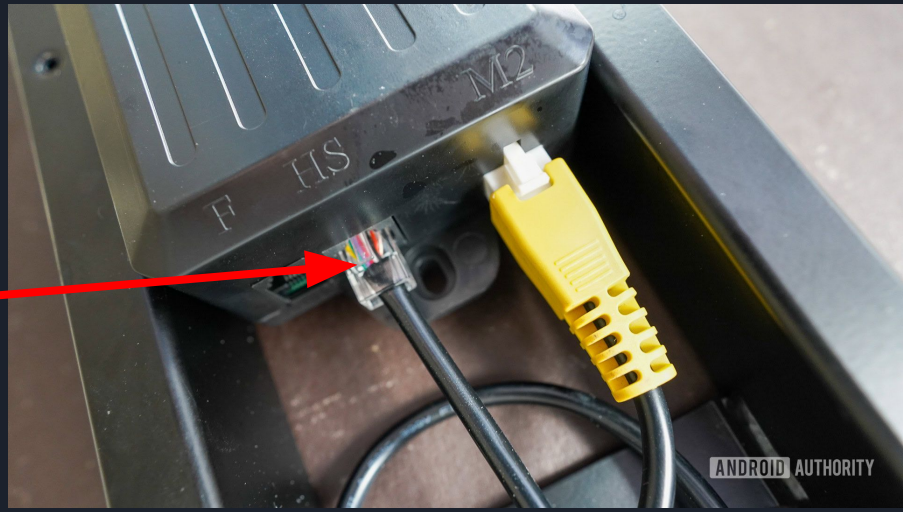
fully Jarvis standing desk

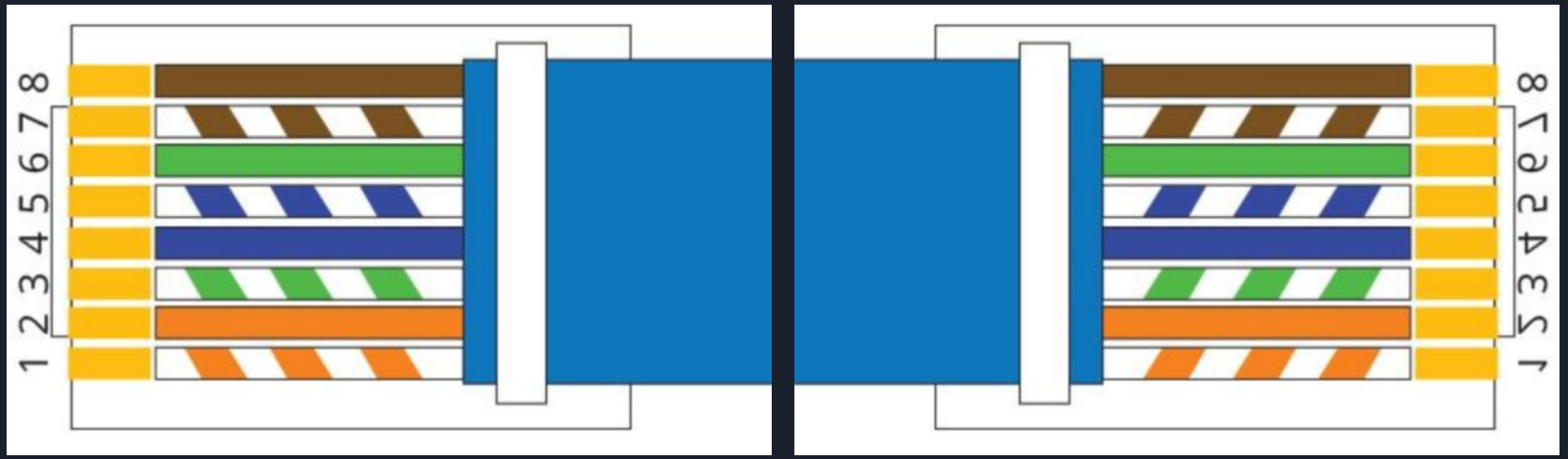
Controlling my standing desk with Nerves!

- Uses an ethernet cable (RJ-45 jack)
- But the signals are not Ethernet!

Handset interface

Desk controller

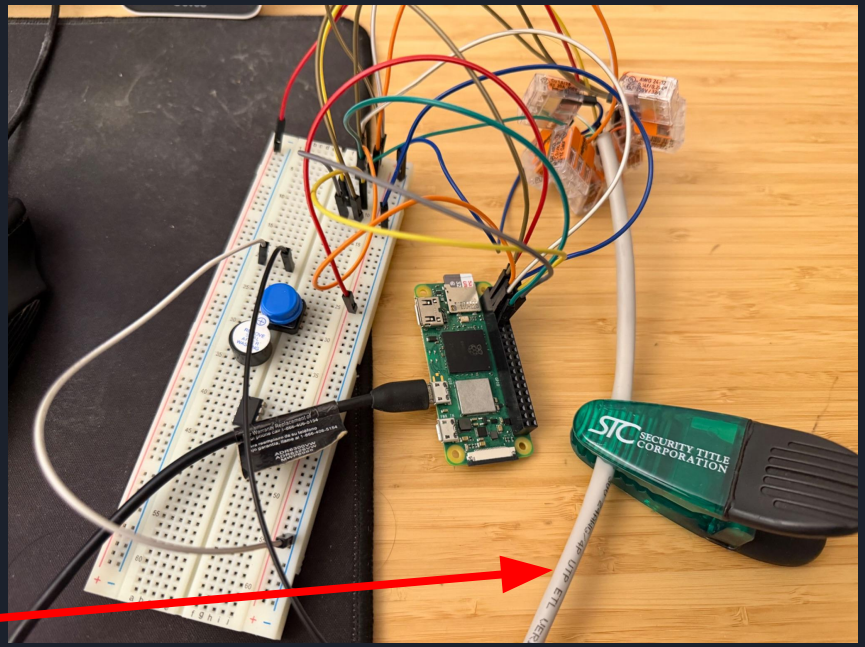




Ethernet cables are just wires

Controlling my standing desk with Nerves!

- These wires are an RJ-45 "breakout" for the desk controller
- RPI Zero 2 W



Ethernet cord

This is the Proof-of-Concept, I'll clean this up later

Controlling my standing desk with Nerves!

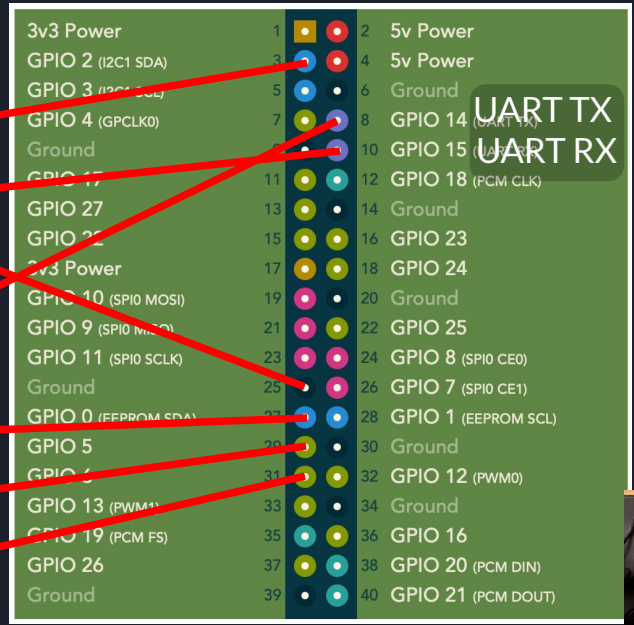
Pin	Label	Description
1	HS3	Handset control line 3 [1]
2	DTX	Serial control messages from controller to handset [2]
3	GND	Ground
4	HTX	Serial control messages from handset to controller [2]
5	VCC	Vcc (5vdc) supply from desk controller [3]
6	HS2	Handset control line 2 [1]
7	HS1	Handset control line 1 [1]
8	HS0	Handset control line 0 [1]

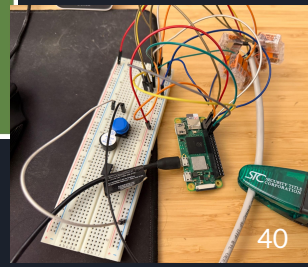
github.com/phord/Jarvis

Reverse engineering instructions ❤️

Controlling my standing desk with Nerves!

Pin	Label	Description
1	HS3	Handset control line 3 [1]
2	DTX	Serial control messages from controller to handset [2]
3	GND	Ground
4	HTX	Serial control messages from handset to controller [2]
5	VCC	Vcc (5vdc) supply from desk controller [3]
6	HS2	Handset control line 2 [1]
7	HS1	Handset control line 1 [1]
8	HS0	Handset control line 0 [1]





Let's try it!

	Down	Up	1	2	3	4
HS3						
HS2				X	X	X
HS1		X	X			X
HS0	X		X		X	

Let's move the desk up

We need to pull down HS1

```
{:ok, gpio} = Circuits.GPIO.open("GPIO5", :output)

Circuits.GPIO.set_pull_mode(gpio, :pulldown)

Circuits.GPIO.write(gpio, 0)
```

And then nothing happened 😵

The next day

What about UART?

- UART is the other way to control the desk
- Rather then pulling lines low you send messages
- The fully desk uses "9600 bps, 8 data bits, no parity (9600/8N1)"
- `{:ok, pid} = Circuits.UART.start_link(opts)`
- `Circuits.UART.open(pid, "ttyAMA0", speed: 9600, active: false)`

Circuits.UART

- Circuits.UART and binary pattern matching made short work of the protocol

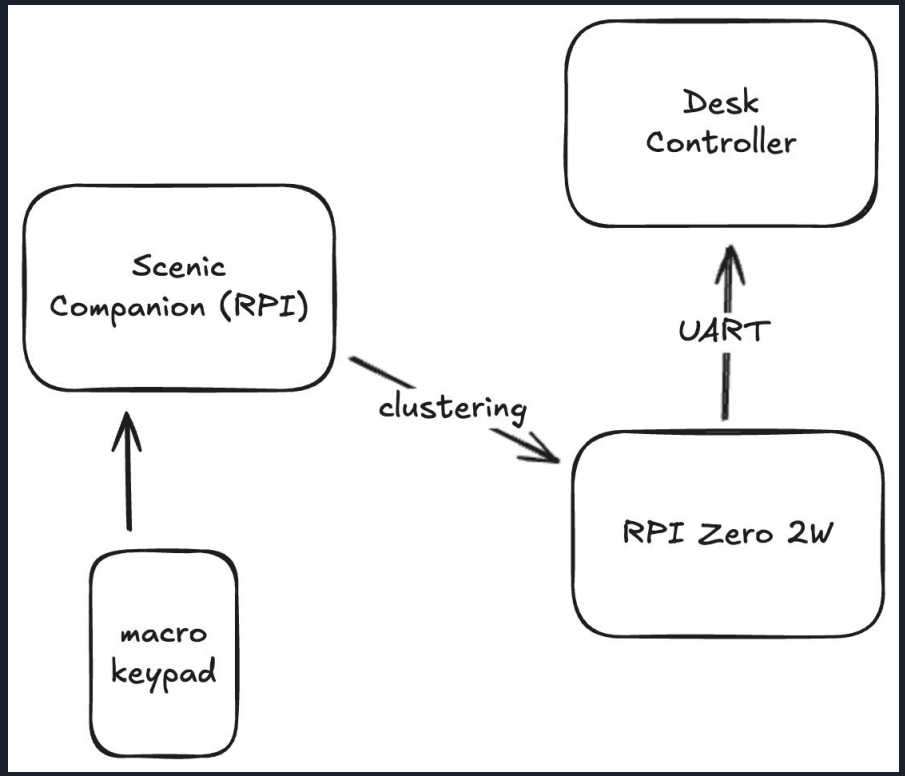
```elixir
def build_command(command_name, params) when is_binary(params) do
  command_bytes = command_name_to_bytes(command_name)

  <<
    @handset_address,
    @handset_address,
    command_bytes,
    byte_size(params),
    params::binary,
    checksum(command_bytes, params),
    @eom # 0x7e
  >>
end
```

```elixir
def send_command(pid \\ @me, message) when is_binary(message) do
  Circuits.UART.write(pid, message)
end
```

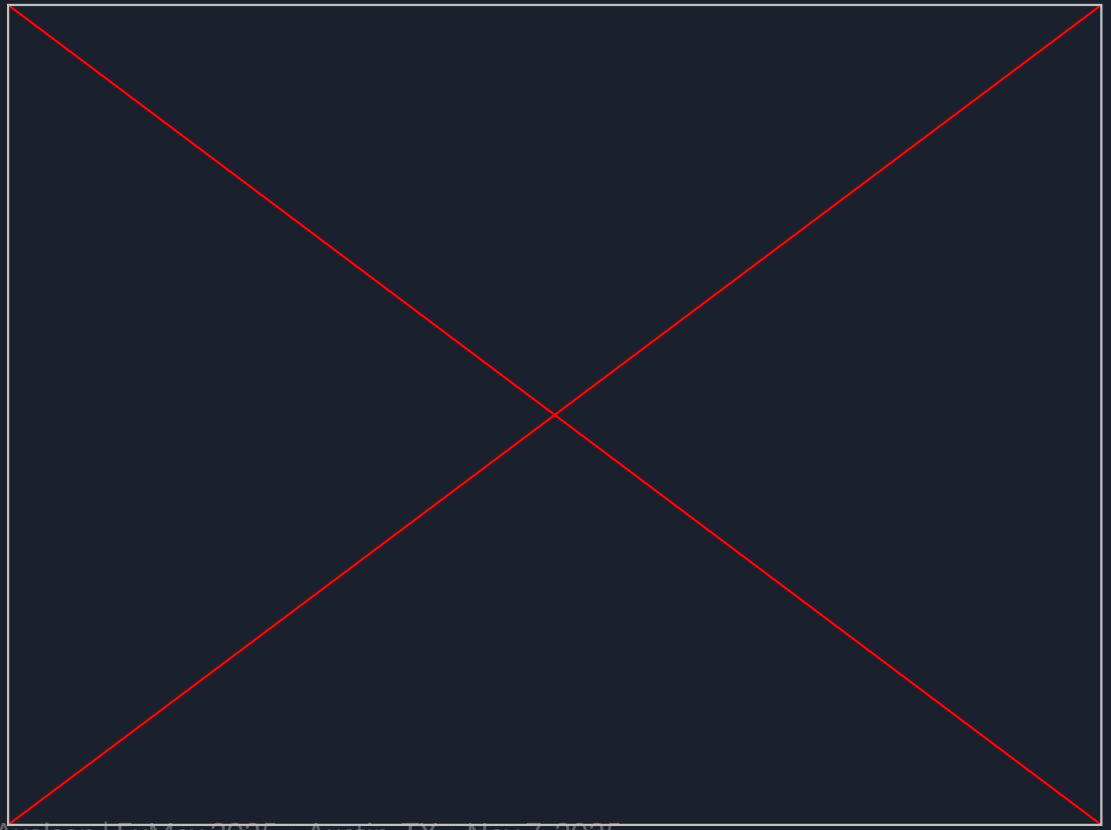
```elixir
def inspect_response(
      <<@desk_address, @desk_address, height_resp(), 3, height::integer-size(16), _unknown,
        _checksum, @eom, rest::binary>>
    ) do
  # NOTE: Assumes that we're set in inches and not meters
  height = height / 10
  IO.puts("Height in inches: #{height}")
  rest
end
```

Sending the command to the desk

```
Task.async(fn ->
  :rpc.call(
    fully_node,
    Fully,
    :move_to_position,
    [:pos1]
  )
end)
```

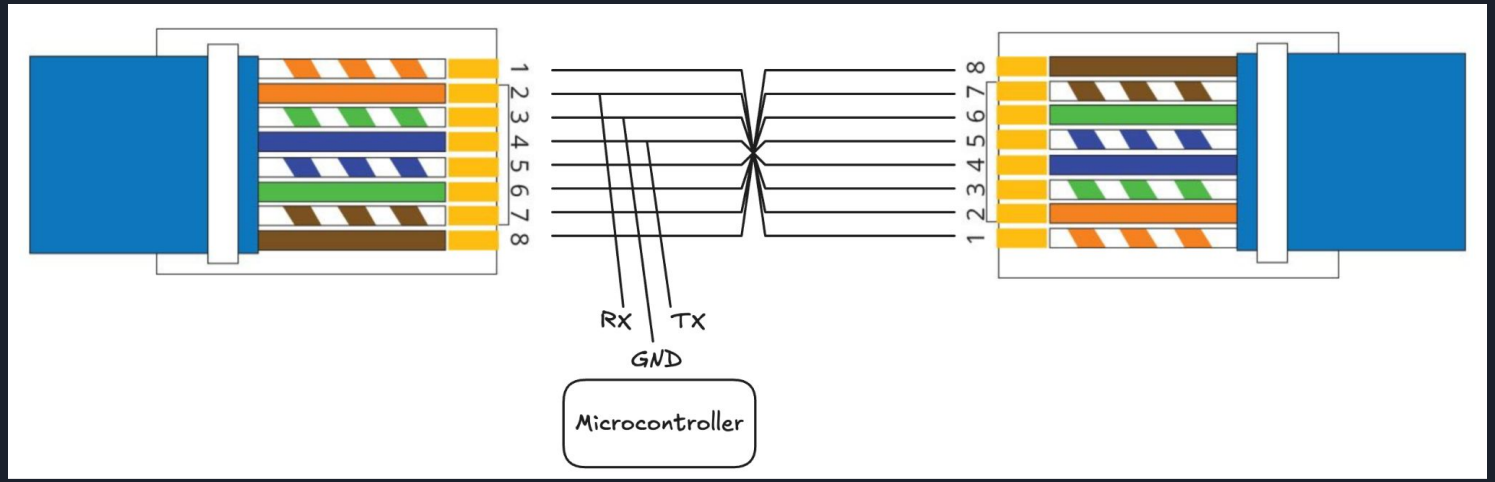


Demo



Ideal wiring

- In the future the handset will also be connected
- The microcontroller will "tap" off the UART lines

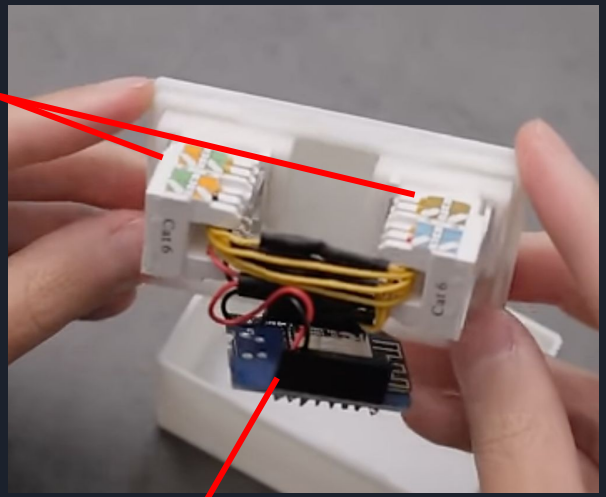


Ideal wiring

David Zhang
YouTube video

- In the future the handset will also be connected
- The microcontroller will "tap" off the UART lines

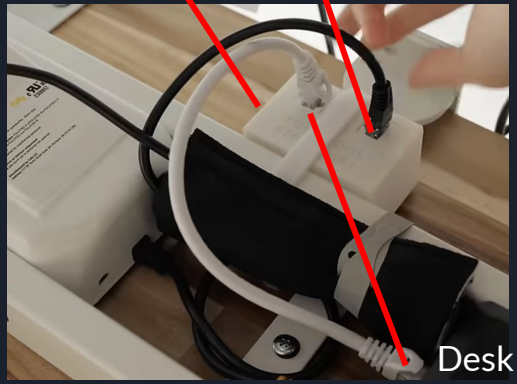
Ethernet passthrough



Microcontroller

Microcontroller (inside box)

Handset



Desk controller

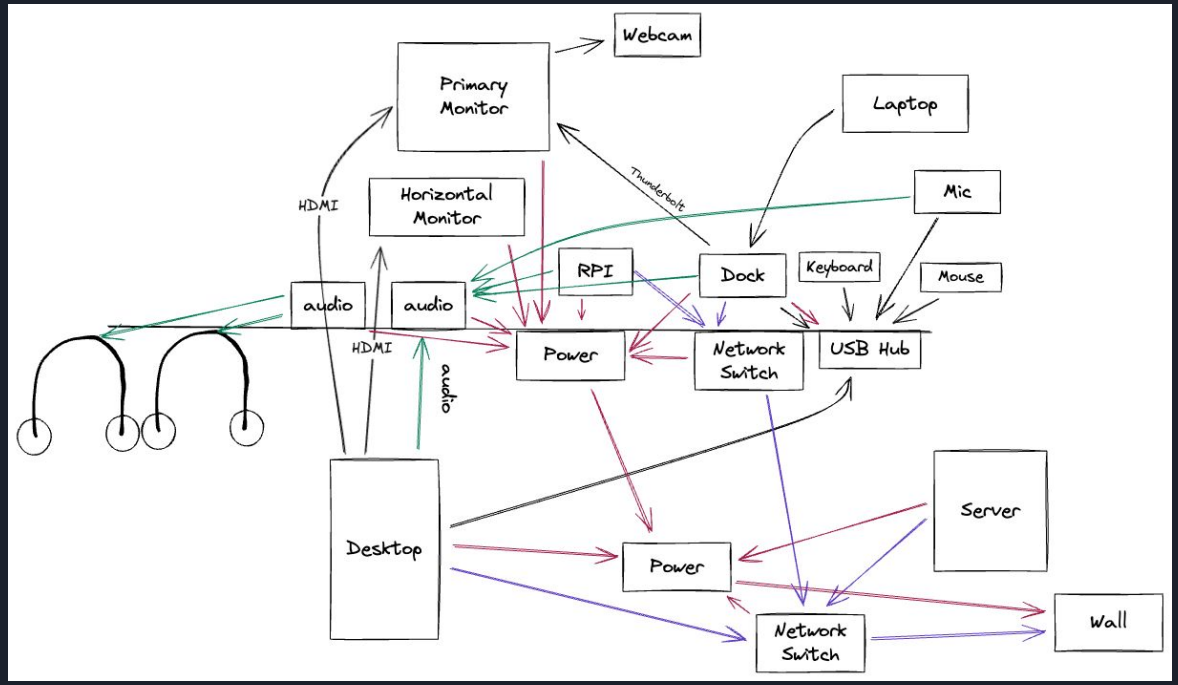
Wrap-up and reflections

- Exposition
- The cast and characters
- The main stage
- **Wrap-up and reflections** ←

Other fun things

- When I start a pomodoro I add a line to my daily Obsidian note
- I can control the lights through Lutron switches
  - Integration powered by HomeAssistant
- Buttons to advance my Pomodoro
- Button to turn off all my devices
- Web interfaces for Pomodoro
- I can remotely access Phoenix apps running on my PI

We didn't even get to talk about the wires!



The RPI Zero 2 W is actually missing from this diagram

Reflections

- Software requires love and care to avoid bitrot
  - RIP my e-ink screen 🪦
- Developing your own software is rewarding
  - You know the audience
  - I love that I've used my own software for 5 years
  - I love when my software acts on the physical world

My repositories

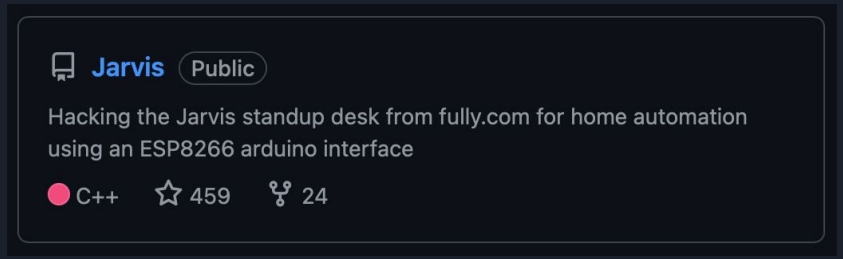
- https://github.com/axelson/scenic-side-screen
- https://github.com/axelson/piano_ex
- https://github.com/axelson/pomodoro
- https://github.com/axelson/pomodoro_phx
- https://github.com/axelson/govee
- https://github.com/axelson/govee_phx
- https://github.com/axelson/govee_semaphore
- https://github.com/axelson/scenic_launcher
- https://github.com/axelson/scenic_asteroids
- https://github.com/axelson/scenic_live_reload
- https://github.com/axelson/lutron_ui
- https://github.com/axelson/fully
- https://github.com/axelson/impression_dash
- https://github.com/axelson/dash_phx

Everything is open source

Libraries/Resources Used - Thank you Maintainers!

Main libraries:

- Nerves
- Keylight
- Scenic
- Phoenix
- Circuits.UART
- ddcutil
- blue_heron
- beam_notify
- muontrap
- ssh_client_key_api



Reverse engineering info by Phil Hord

Inspirational office automation video by David Zhang

Let's automate all the things!

Thank you!

Sleep all devices

Toggle LEDs/Party mode

Stop/Start meeting

Stop/Play/Next song

Raise/Lower desk

Connect/Disconnect main monitor

Switch main monitor to laptop/desktop

"Advance" Pomodoro