Behind CyberEther:

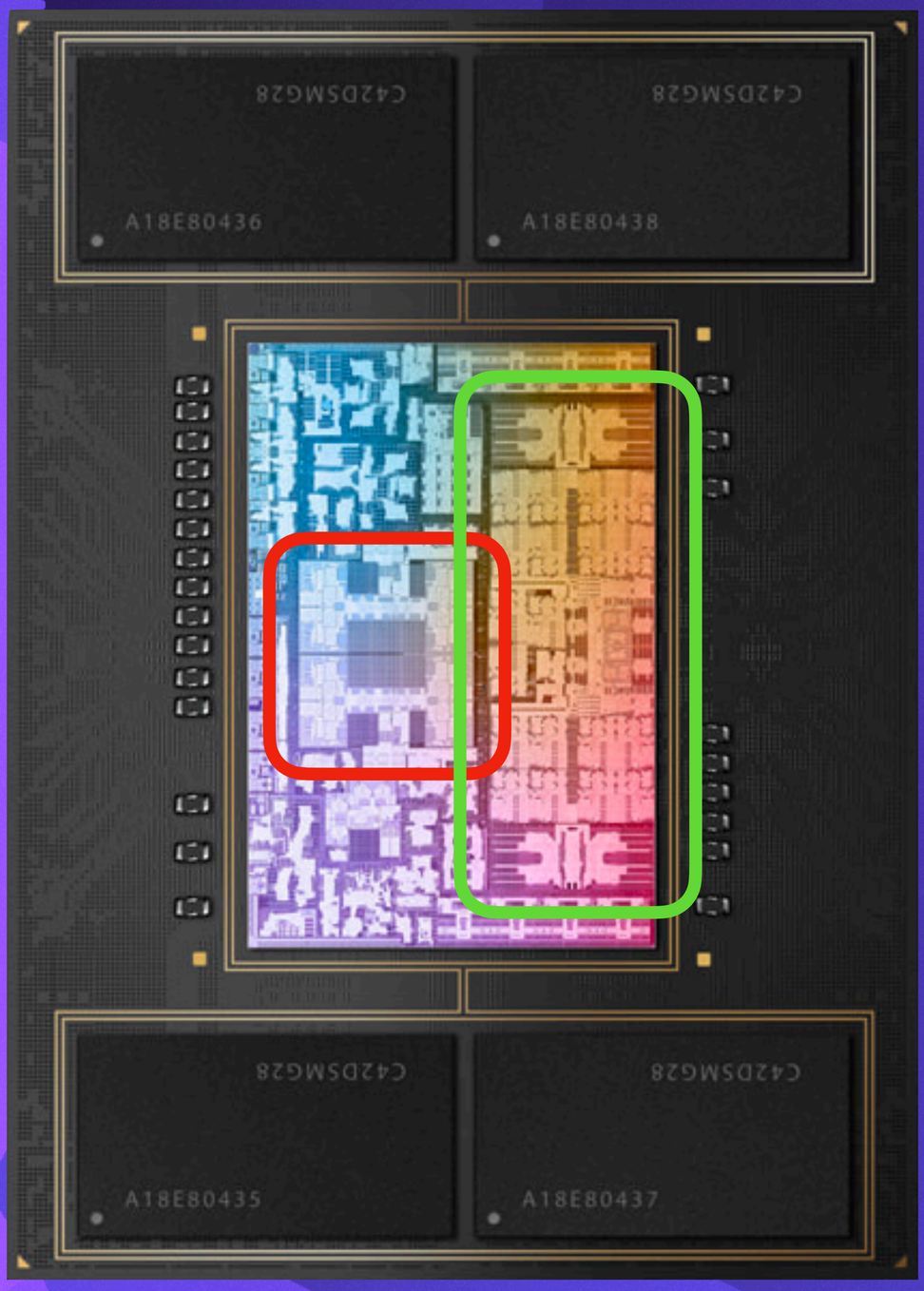
Metal Hardened Portable GPU Accelerated Interface

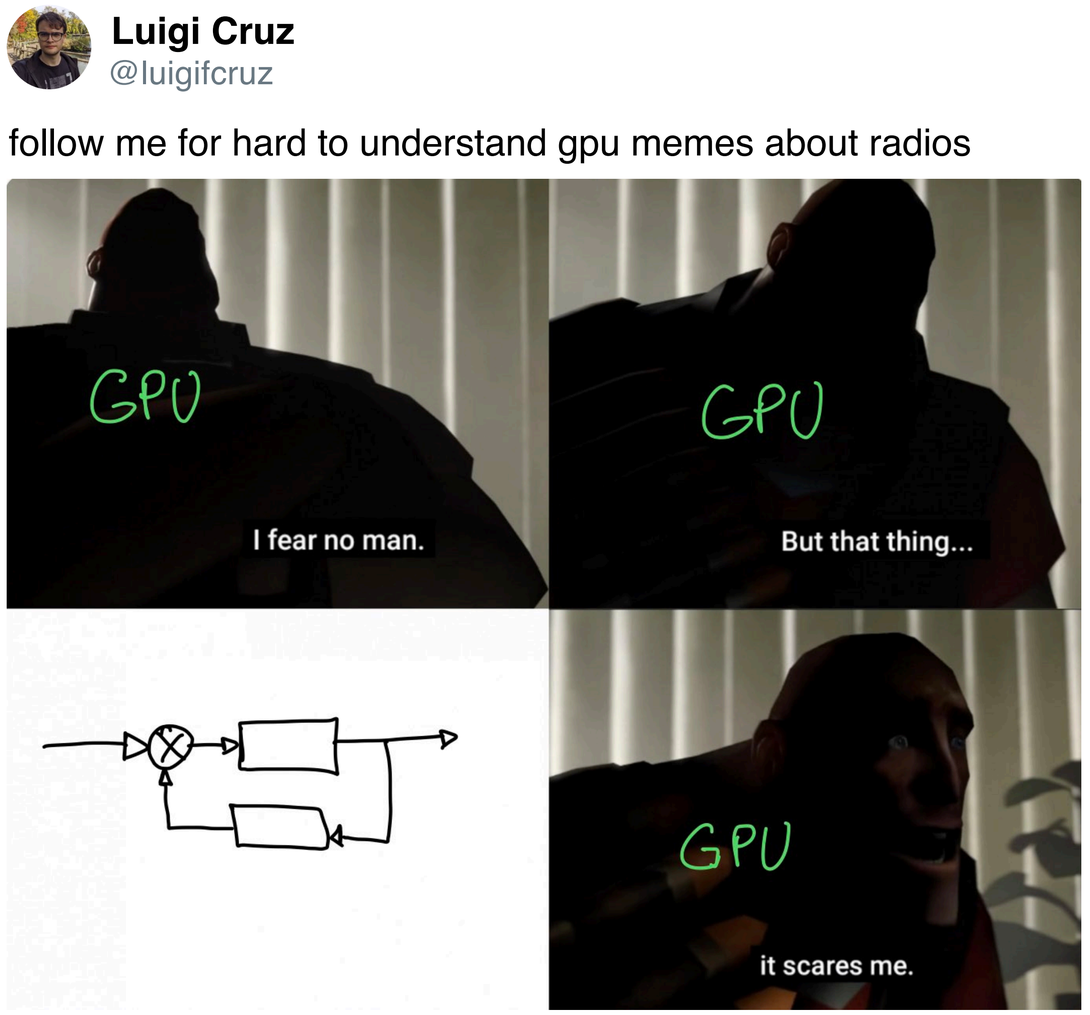
Presented by Luigi Cruz (@luigifcruz) on September 7th

GNU Radio Conference 2023- Tempe, AZ

Summary

- Why heterogeneous first is important.
- How CyberEther came to be.
- What CyberEther is at the moment.
- How it works.
- Future.


C42DSMG28
A18E80436
C42DSMG28
A18E80438
C42DSMG28
A18E80435
C42DSMG28
A18E80437

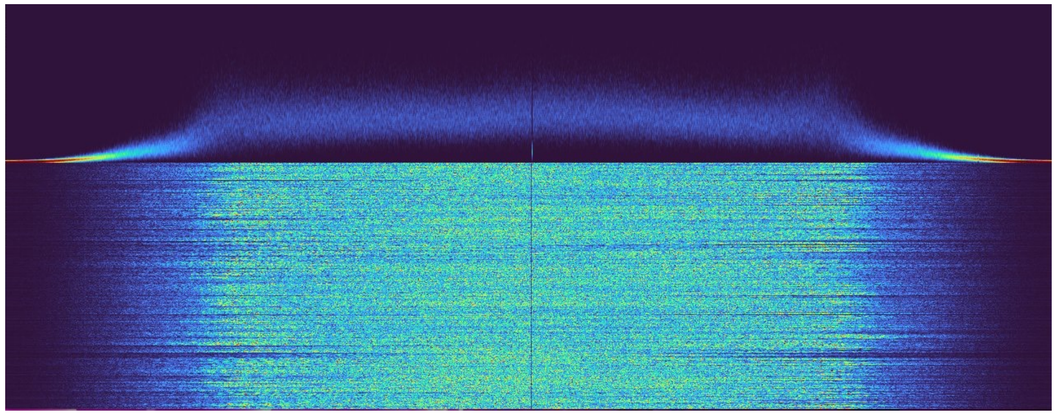

Luigi Cruz
@luigifcruz
follow me for hard to understand gpu memes about radios
GPU
I fear no man.
GPU
But that thing...
GPU
it scares me.
8:17 PM - July 13, 2023 - Twitter Web App

Section One: **History**

Early Development

Luigi Cruz
@luigifcruz

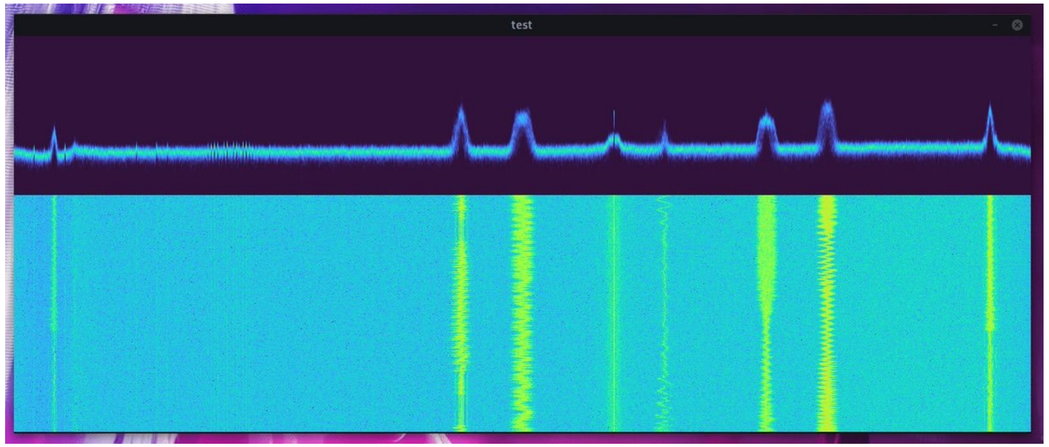
New project! Playing an FM station inside a web-browser with Airspy R2 streaming complex I/Q at 2.5 msps and LiquidDSP demodulating it in real-time. Powered by my libusb translation layer to WebUSB and clever JS threading. No changes are required to the libairspy. #WebAssembly



8:35 PM - Jan 25, 2021 - Twitter Web App

Luigi Cruz
@luigifcruz

This is the spectrum viewer I did for the browser-based SDR console. I'll release an upgraded version that also works on an OS. It'll focus on simplicity, portability, and customizability. Yes, I'm using this as an excuse to learn multiple graphical technologies.

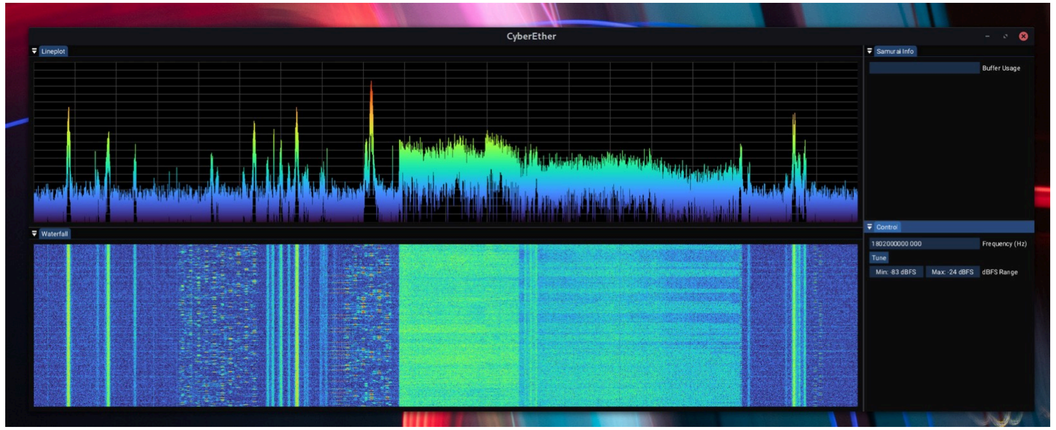


3:32 PM - Mar 15, 2021 - Twitter Web App

Early Development

Luigi Cruz
@luigifcruz

The FFT is now CUDA based.



10:20 AM - Jun 8, 2021 - Twitter Web App

devnulling
@devnulling

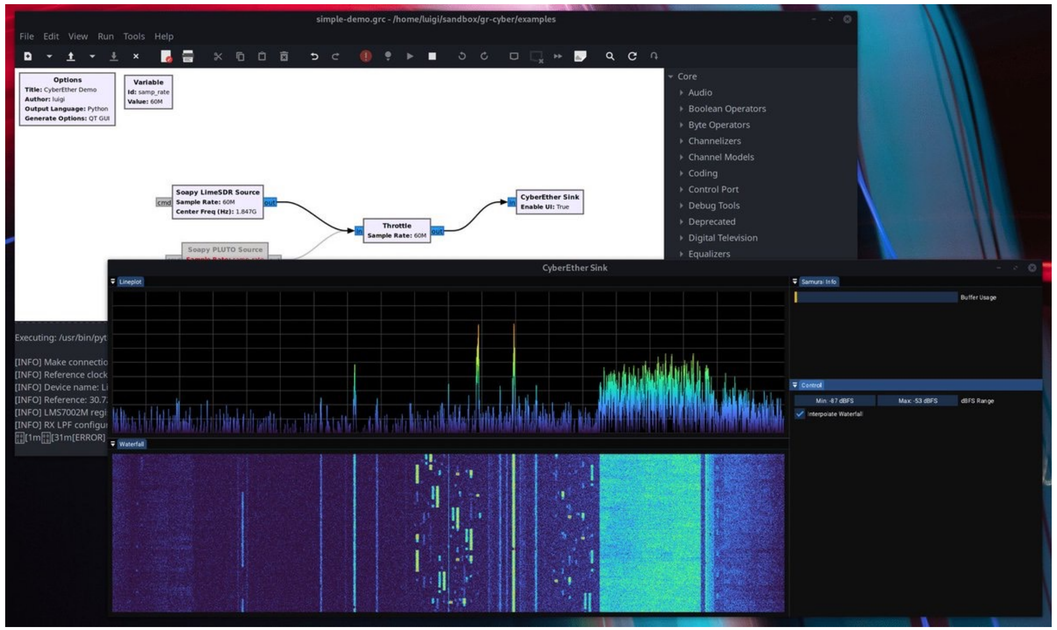
Any change to make this FFT/waterfall into a GR OOT block?
:)

4:13 PM - Jun 8, 2021 - Twitter Web App

Early Development

Luigi Cruz
@luigifcruz

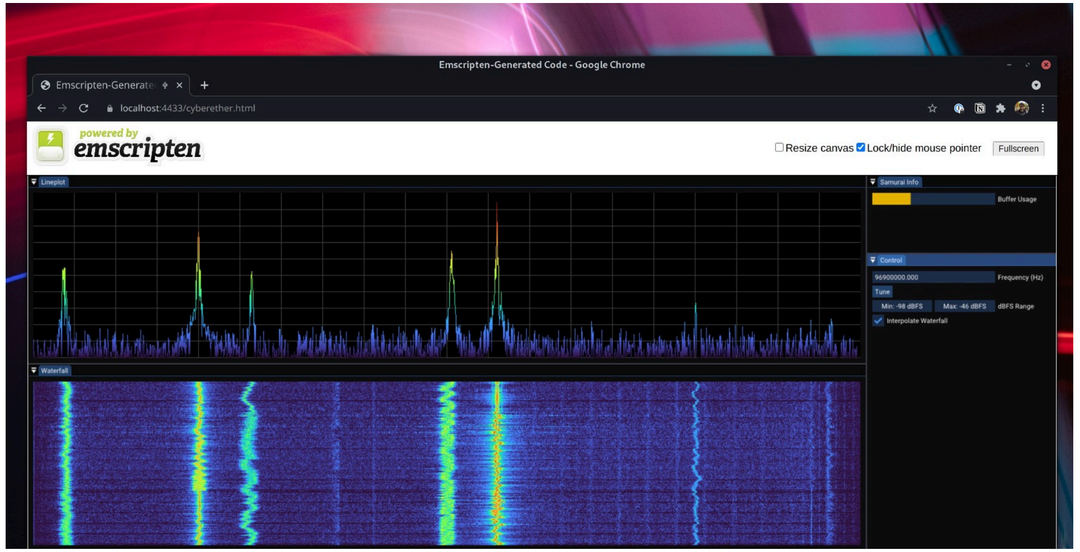
oh hi. cuda spectrogram on gnuradio.



1:47 PM - Jun 17, 2021 - Twitter Web App

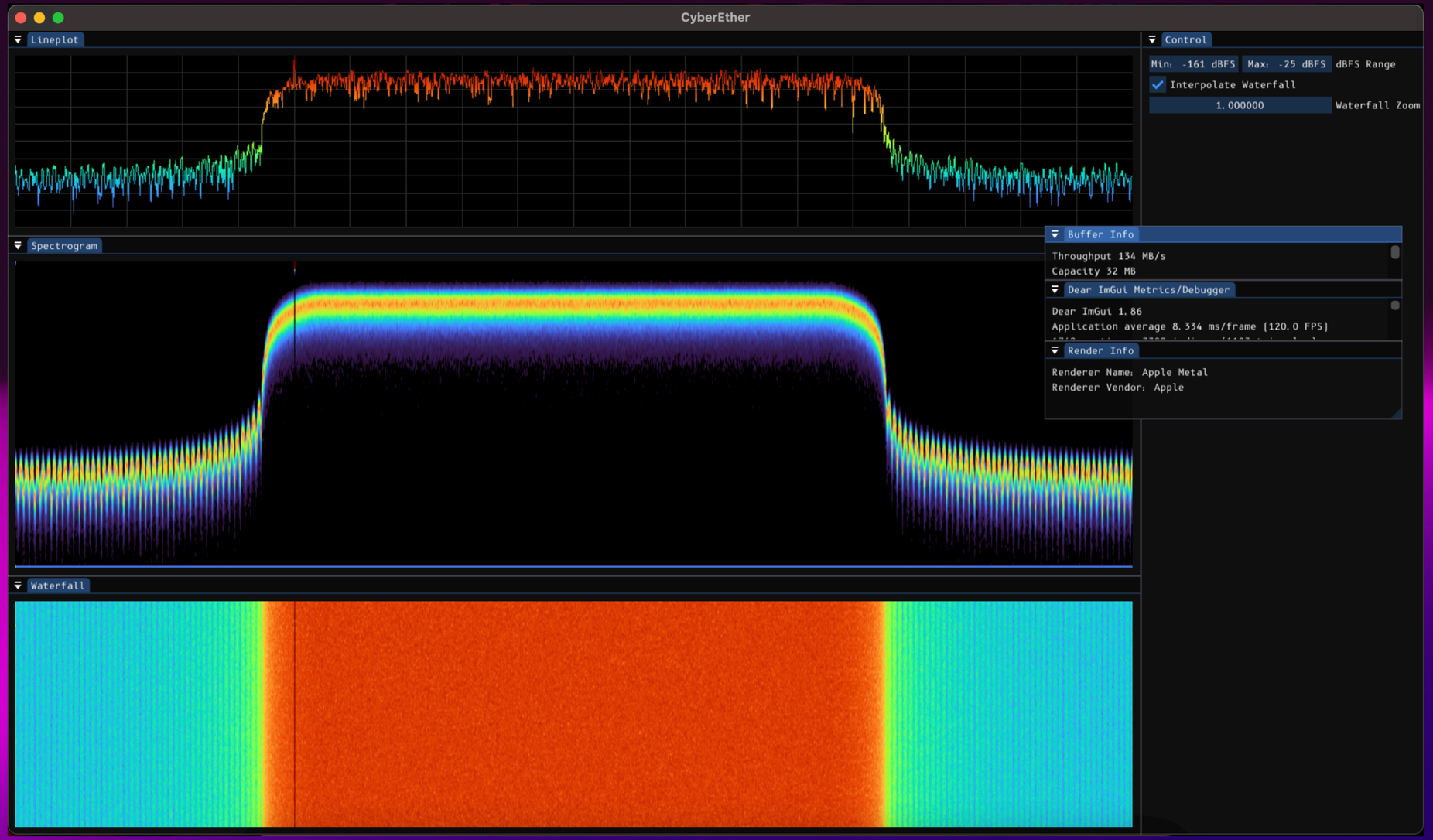
Luigi Cruz
@luigifcruz

This is nowhere near real-time yet but it's running inside Chrome, from USB to UI. I think I created a monster.



7:48 PM - Jun 15, 2021 - Twitter Web App

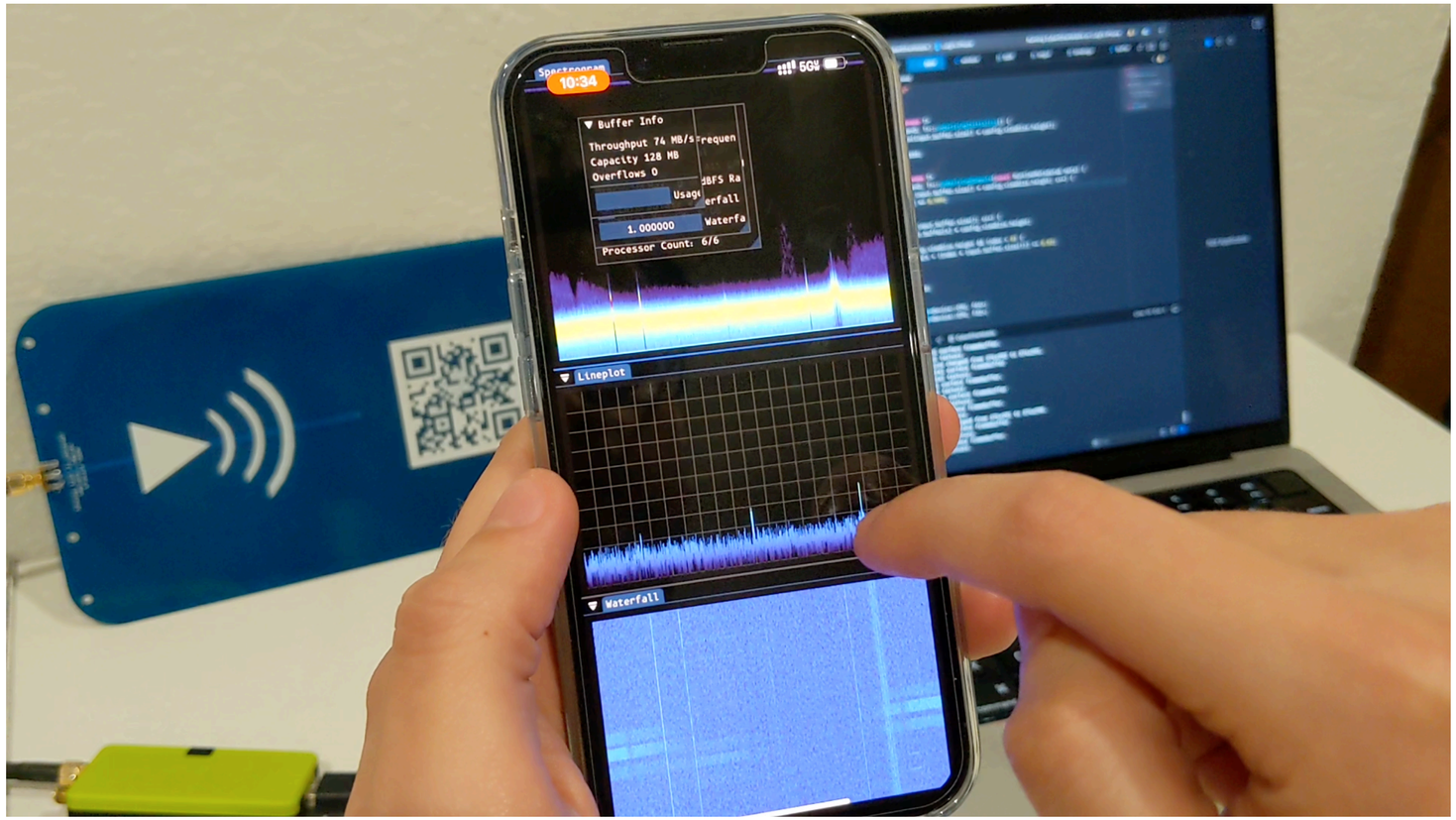
Last Year Conference



Dominion

Luigi Cruz
@luigifcruz

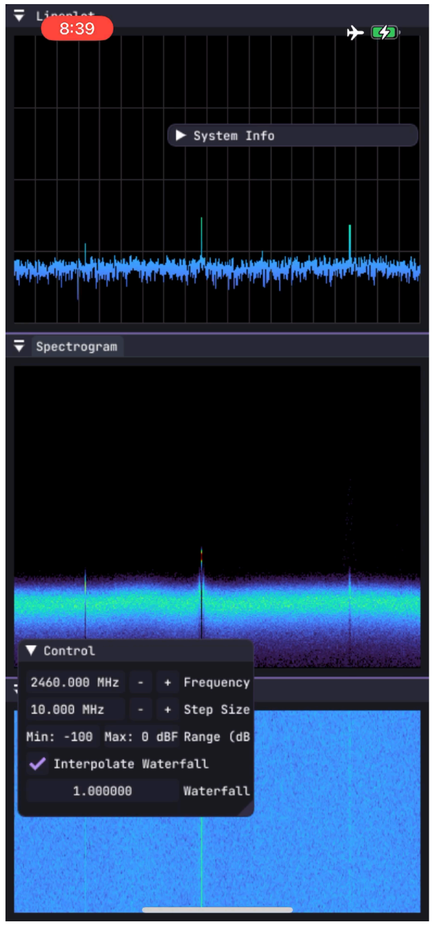
It works! I ported CyberEther to run on iOS using Apple's Metal. 🚀



9:56 AM - Dec 23, 2022 - Twitter Web App

Luigi Cruz
@luigifcruz

I'm at a location where microwave ovens are inside a Faraday cage for RFI reasons. I was curious and I grabbed my SDR to see if I could see any leakage.

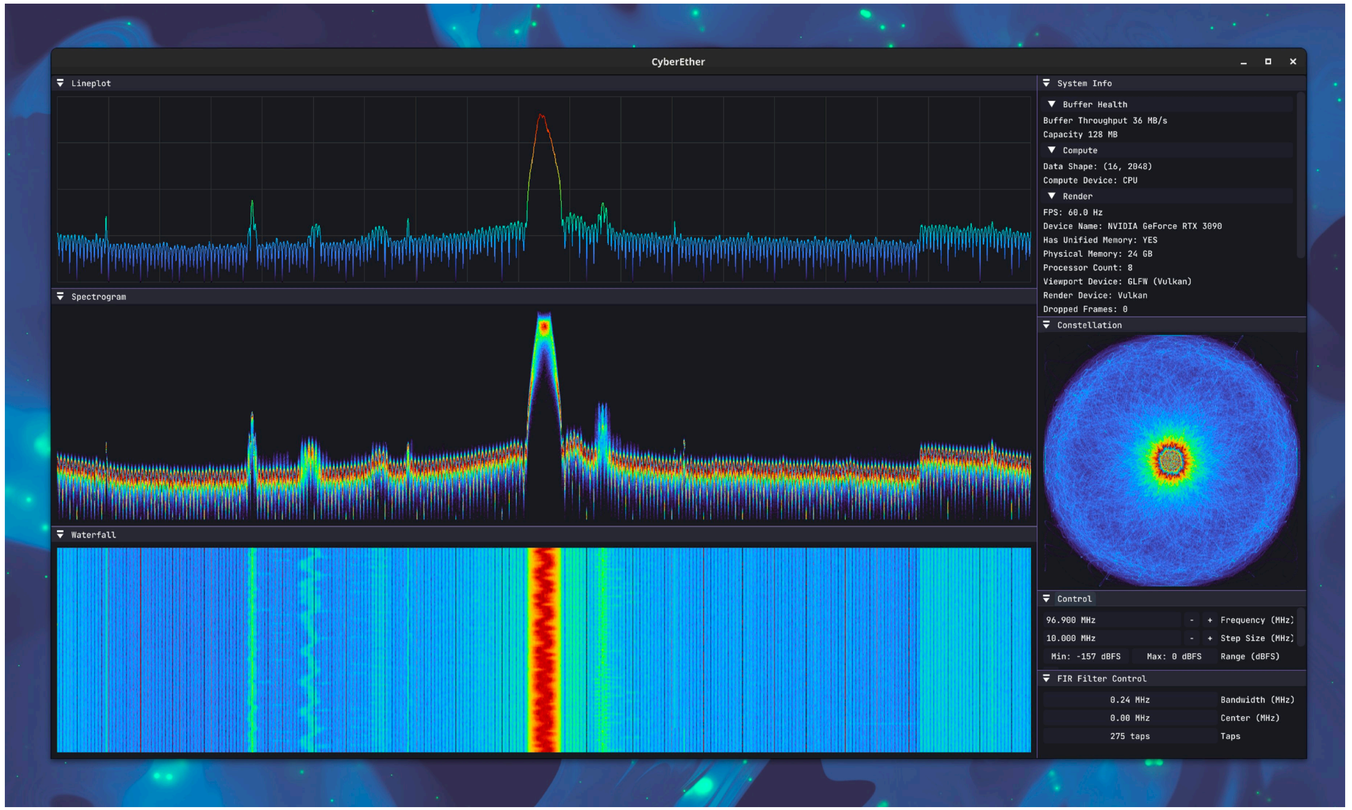


8:41 PM - Apr 12, 2021 - Twitter Web App

Dominion

Luigi Cruz
@luigifcruz

Meet the Orb. CyberEther on Vulkan is ready!

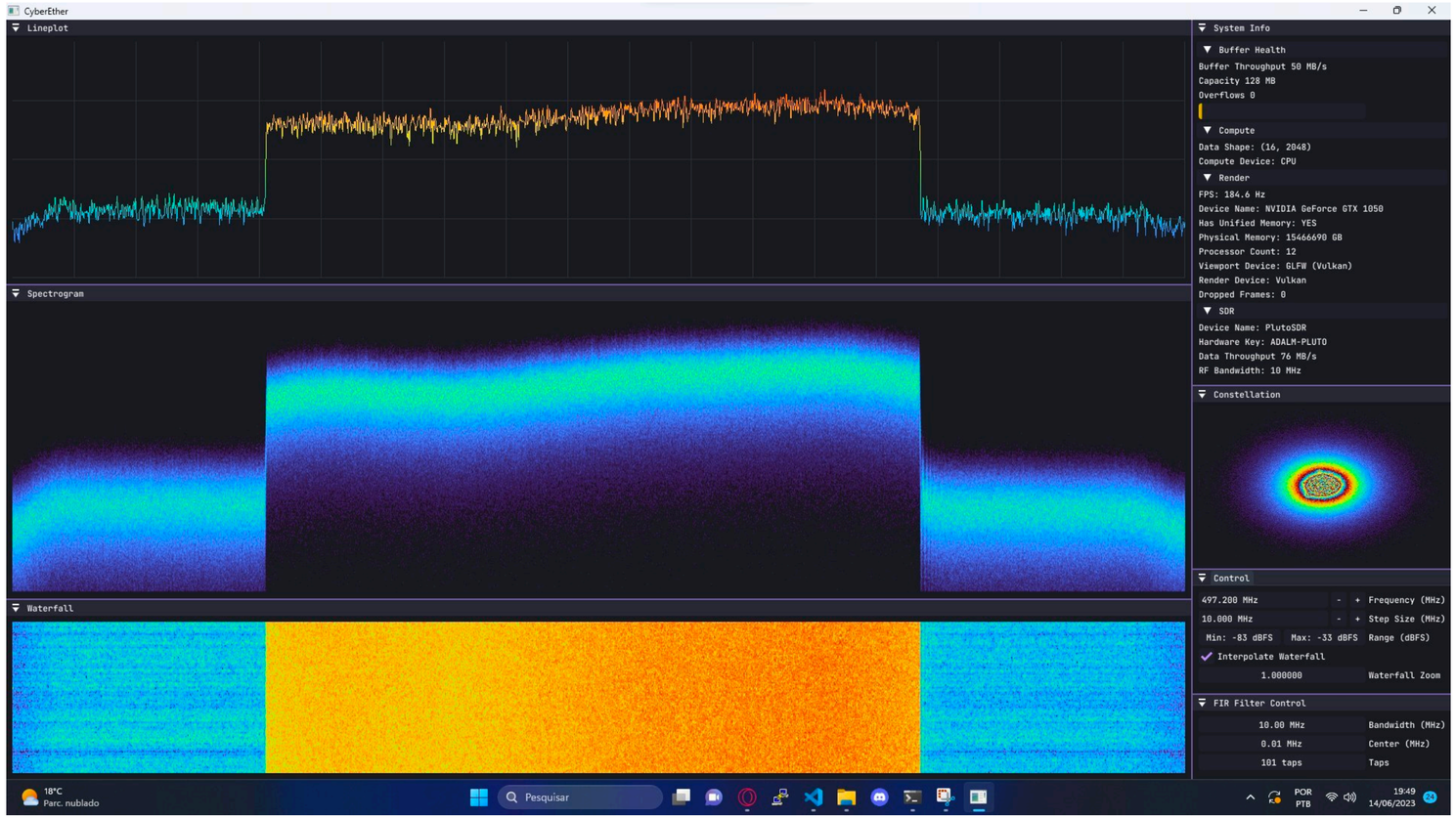


9:52 PM - Jul 11, 2023 - Twitter Web App

Dominion

Paulo Dutra - PU4THZ
@DutraCGI

Its alive!!! #CyberEther on Windows!

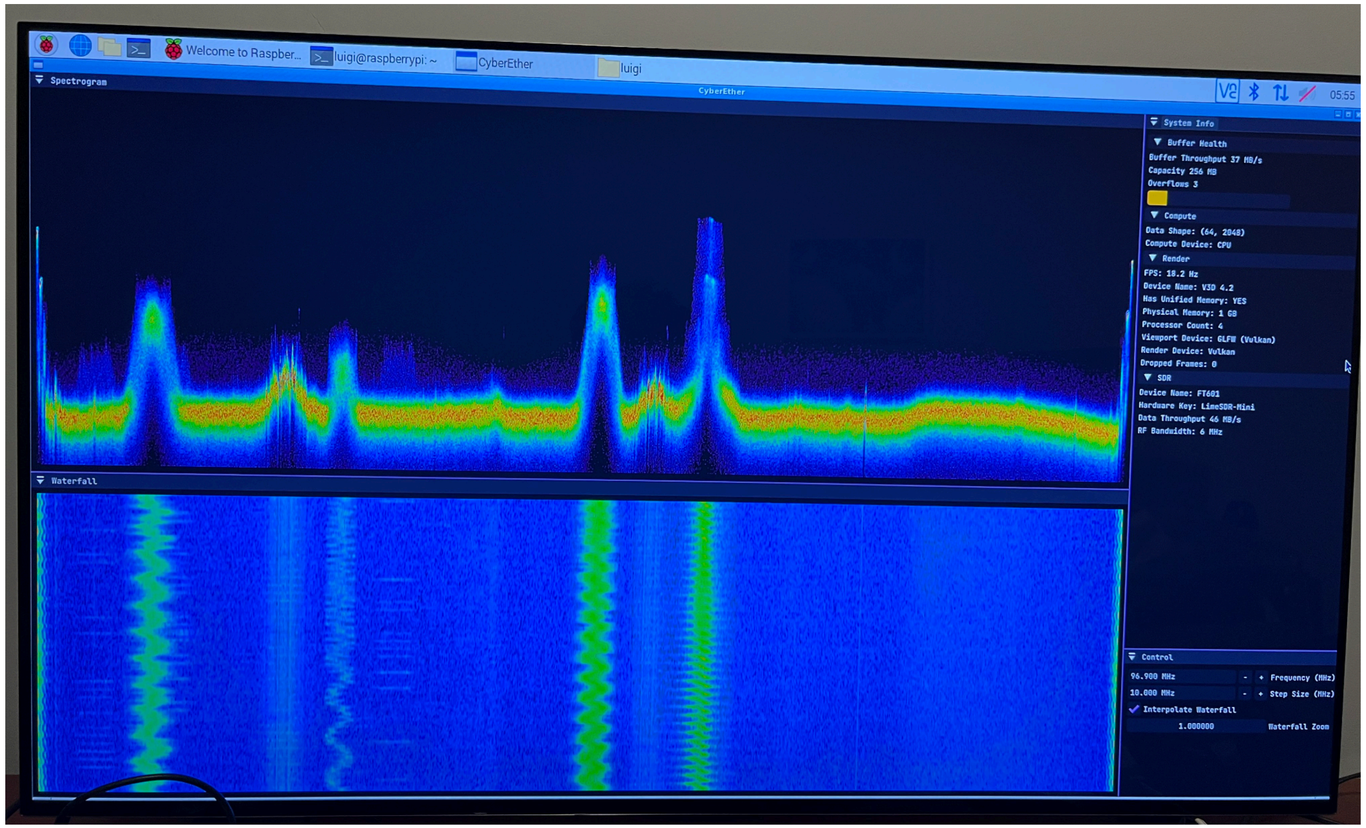


4:04 PM - Jun 14, 2023 - Twitter Web App

Dominion

Luigi Cruz
@luigifcruz

There we go! CyberEther running on the Pi 4 in Full HD.



9:59 PM - Jun 14, 2023 - Twitter Web App

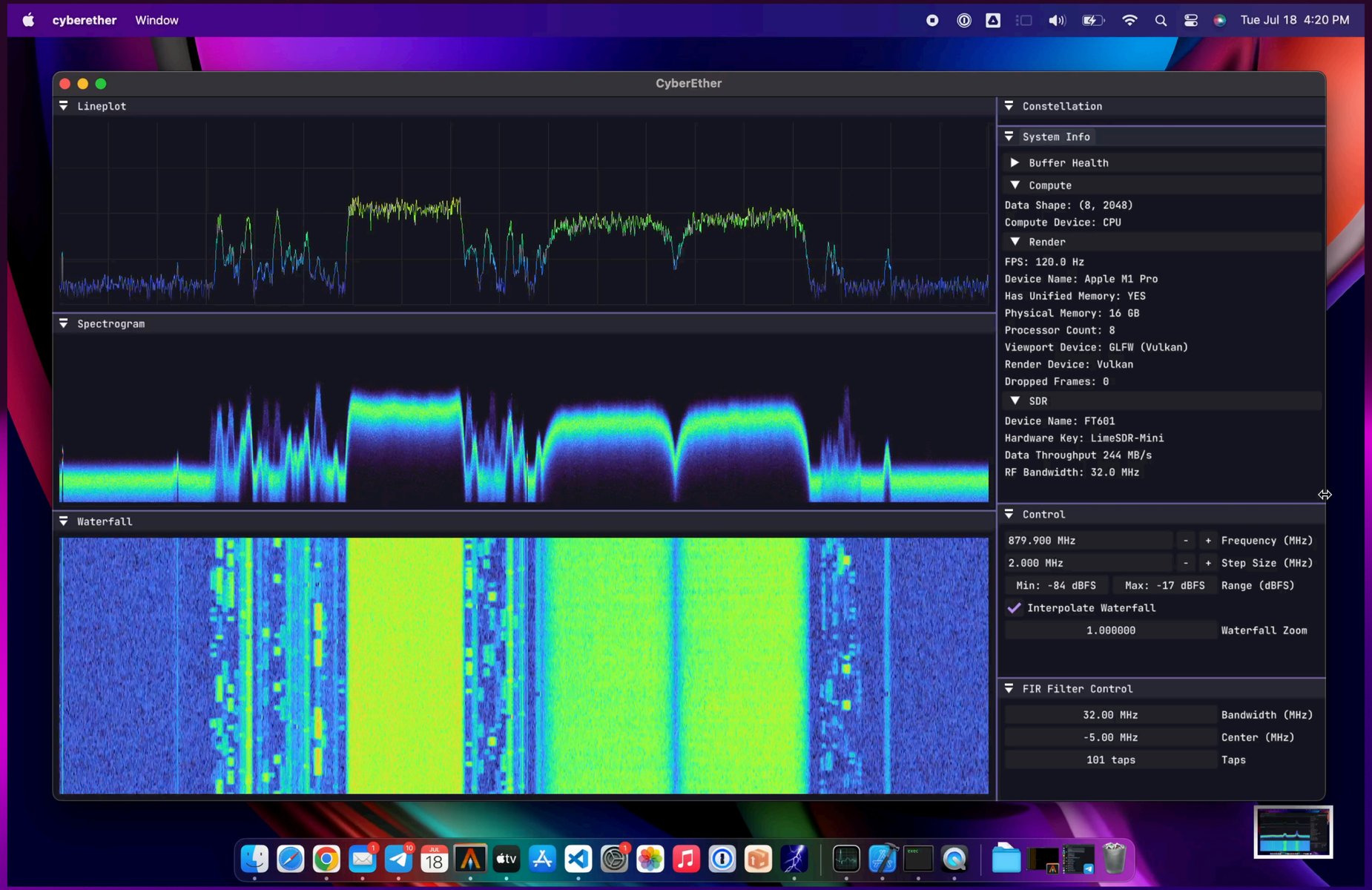
Dominion

Luigi Cruz
@luigifcruz

Here it is! The WebGPU graphical backend for CyberEther on the browser.

9:08 PM - Jul 19, 2023 - Twitter Web App

Dominion


cyberether
Window
Tue Jul 18 4:20 PM
CyberEther
Lineplot
Spectrogram
Waterfall
Constellation
System Info
Buffer Health
Compute
Data Shape: (8, 2048)
Compute Device: CPU
Render
FPS: 120.0 Hz
Device Name: Apple M1 Pro
Has Unified Memory: YES
Physical Memory: 16 GB
Processor Count: 8
Viewport Device: GLFW (Vulkan)
Render Device: Vulkan
Dropped Frames: 0
SDR
Device Name: FT601
Hardware Key: LimeSDR-Mini
Data Throughput 244 MB/s
RF Bandwidth: 32.0 MHz
Control
879.900 MHz
Frequency (MHz)
2.000 MHz
Step Size (MHz)
Min: -84 dBFS
Max: -17 dBFS
Range (dBFS)
Interpolate Waterfall
1.000000
Waterfall Zoom
FIR Filter Control
32.00 MHz
Bandwidth (MHz)
-5.00 MHz
Center (MHz)
101 taps
Taps

DEMO

But wait! There is more!

Section Two: CyberEther

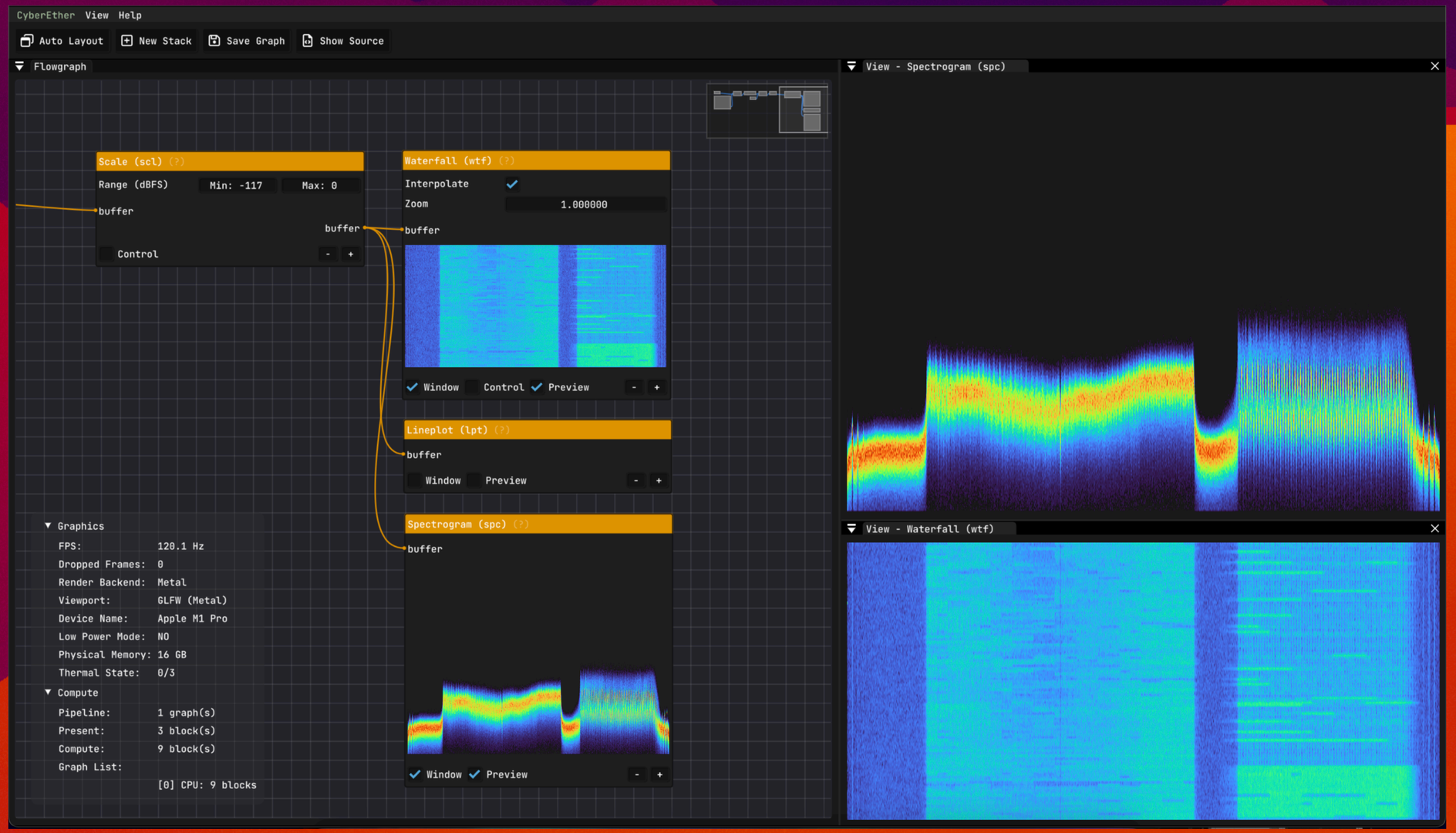
CyberEther

Portable and heterogeneously-accelerated GUI for radio signals.

- Built from the ground up to display signals generated by SDRs.
- Currently offers Lineplot, Waterfall, Spectrogram, and Constellation.
- Runs as close to the metal (literally!) as possible using heterogeneous APIs.
- Minimal dependencies with a modular design.
- Low code duplication by abstracting graphical and compute APIs.
- Easy to implement on third-party projects.

CyberEther

Flowgraph



CyberEther

Lineplot

- A simple and lightweight way to visualize frequency domain signals.
- Low memory requirements. No data retention.

CyberEther

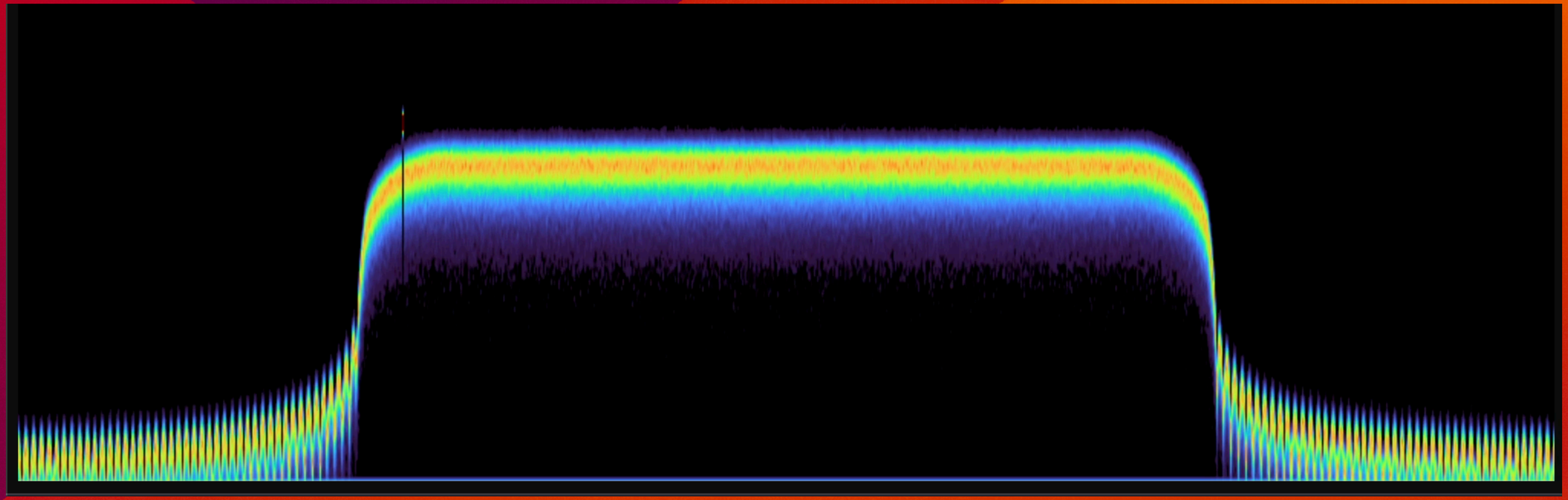
Waterfall

- Standard way to visualize how frequency domain signals change over time.
- Larger memory requirements than Lineplot. Data is retained.
- Output can be fed to Neural Network for inference.

CyberEther

Spectrogram

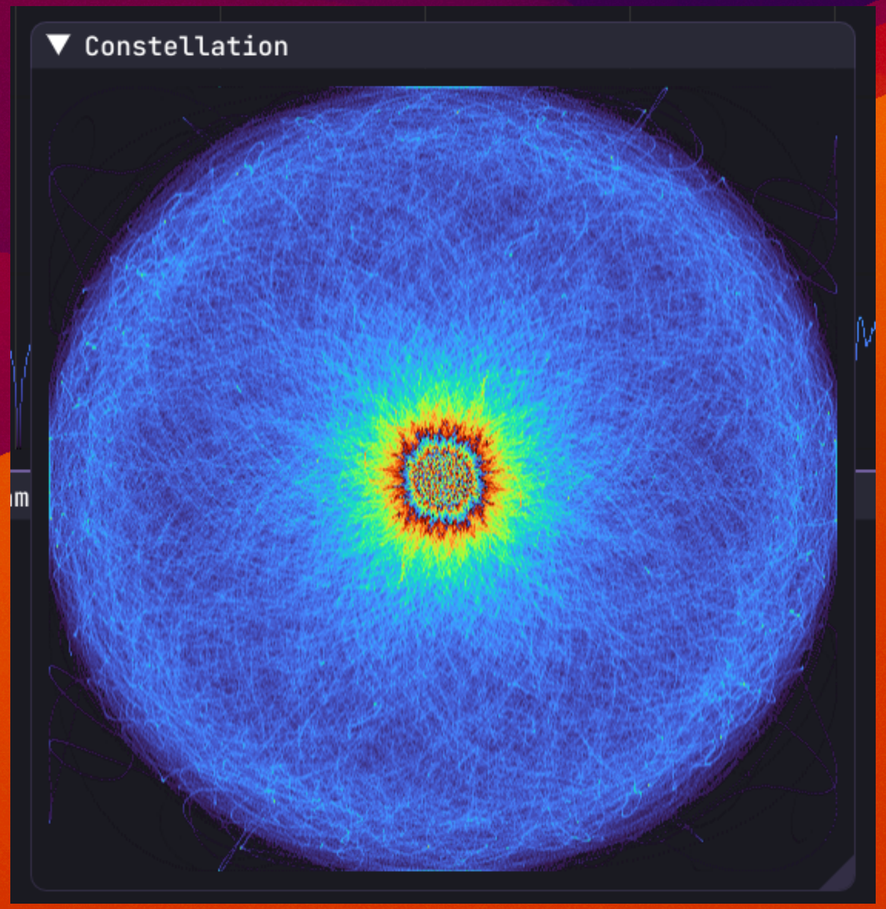
- A **wholesome** way to visualize how frequency domain signals change over time.
- Lower memory requirements than Waterfall but much more compute-intensive.



CyberEther

Constellation

- Ideal to visualize modulations.



Section Three: **Why & How?**

CyberEther

Runs as close to the metal as possible using heterogeneous APIs.

- Accelerated **graphics** with low-level frameworks (Metal, Vulkan, WebGPU).
- Accelerated **compute** with heterogeneous APIs (CUDA, Metal, Vulkan).
- Fallback to CPU-based compute when no hardware acceleration is available.
- Dependencies aren’t mandatory, it will compile with the available ones.

CyberEther

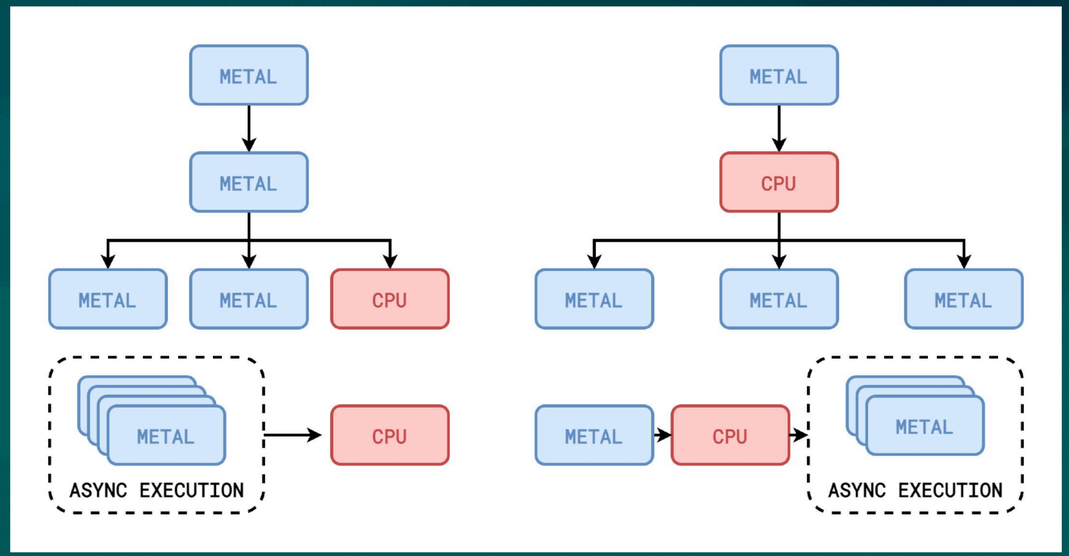
Low code duplication by abstracting graphical and compute APIs.

- Graphical modules are written on top of a abstraction layer.
- Shaders are translated from GLSL to SPIR-V then WGL and MSL.
- This abstraction layer for graphics is called **Jetstream::Render**.
- The graphical backend can be selected during runtime.
- Compute modules are also abstracted using **Jetstream::Module**.
- Module backend can be selected in runtime.
- Mix-match of backends completely supported!

CyberEther

Heterogeneous compute graph solver.

- Utilizes progressive lowering of the original graph.
- Main graph is broken into a dependency list.
- Each sub-graph is broken into backend graphs (Metal, CPU, etc).
- Synchronization between backend graphs determined and executed.



CyberEther

Easy to implement on third-party projects.

- Front-end agnostic.
- All visualizations are rendered on a headless frame buffer.
- Rendered frame can be attached to any window (Qt, Cocoa, MTKView, etc).
- CyberEther uses ImGui only for windowing, no visualization is handled by it.
- Rendered frame can even be shared with another process using DMA-BUF.
- TL;DR: It’s easy to use on iOS, Android, macOS, or even a browser.

Section Four: **Future**

CyberEther

Current status of supported devices.

Device	Metal	Vulkan	WebGPU
macOS (Apple Silicon)	✅ (Native)	✅ (via MoltenVK)	✅ (Dawn*)
iOS/iPadOS	✅ (Native)	✅ (via MoltenVK)	N/A
Linux (NVIDIA/AMD/Intel)	N/A	✅ (Native)	✅ (Dawn*)
Windows (NVIDIA/AMD/Intel)	N/A	✅ (Native*)	✅ (Dawn*)
Android	N/A	✅ (Native*)	✅ (Dawn*)
Browser (WebKit/Chrome/Firefox)	N/A	N/A	✅ (Chrome)

CyberEther

Current status of **graphical** modules.

Module	Metal	Vulkan	WebGPU	CPU+Render
Lineplot	✅ (Full)	✅ (Graphical)	✅ (Graphical)	✅ (Full)
Waterfall	✅ (Full)	✅ (Graphical)	✅ (Graphical)	✅ (Full)
Spectrogram	✅ (Full)	✅ (Graphical)	✅ (Graphical)	✅ (Slow but full)
Constellation	❌ (Porting)	❌ (Porting)	❌ (Porting)	✅ (Slow but full)

CyberEther

Current status of **compute** modules.

Module	CPU	CUDA	Metal	Vulkan	Description
Amplitude	✅	❌ (Porting)	✅	❌ (Next)	Complex data to power.
FFT	✅	❌ (Porting)	✅	❌ (Next)	Channelization.
Multiply	✅	❌ (Porting)	✅	❌ (Next)	Vector multiplication.
Scale	✅	❌ (Porting)	✅	❌ (Next)	Scaling vector by factor.
Window	✅	❌ (Porting)	✅	❌ (Next)	Apply a window to vector.
Filter	✅	❌ (Porting)	✅	❌ (Next)	Apply a FIR Filter to vector.
Soapy	✅	❌ (Porting)	✅	❌ (Next)	Simple SoapySDR tap.

CyberEther

Future development

- Make CyberEther available as a iOS/iPadOS application. VERY SOON!
- Make CyberEther available on the browser. VERY SOON!
- Unified compute backend with additional CUDA, Vulcan, WebGPU support.
- Increase reliability and DSP fidelity.

DEMO

It hopefully works (twice)!

Thanks for listening!

https://github.com/luigifcruz/CyberEther

Questions?

Contact me!

https://luigi.ltd/contact/