

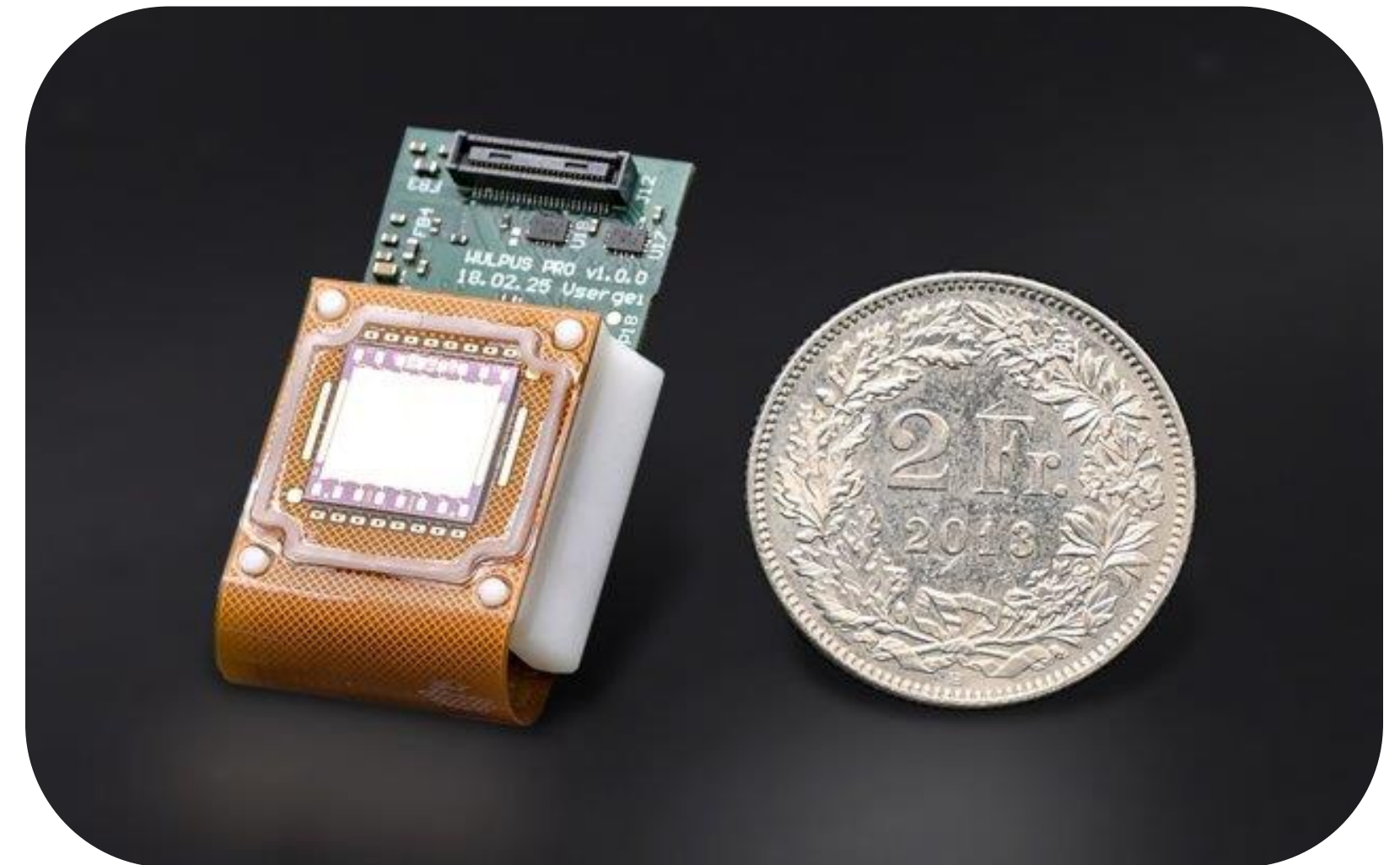


Open-source wearable ultrasound platforms: a rocky path from lab prototype to public knowledge

Dr. Sergei Vostrikov

 Winterthur, Switzerland

 Sugar Land, Texas, USA



About the speaker

From physics and robotics to wearable ultrasound hardware

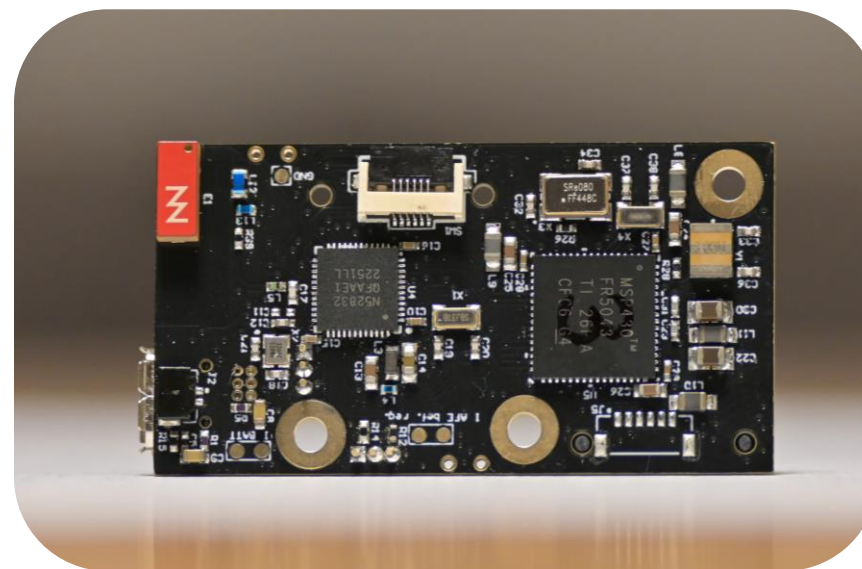
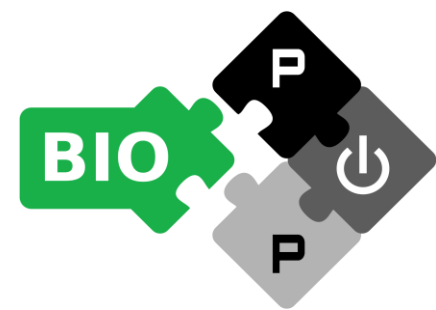
Education

- BSc in Physics
- MSc in Robotics
- PhD in Electrical Engineering (PULP-BIO Group, prof. Luca Benini)

Current role

R&D Electronics Team Lead,
Rheonics

Lead Systems Engineer for TinyProbe and WULPUS



Agenda

- 1 Opening
- 2 Why open source?
- 3 Current state of the field
- 4 The rocky path
- 5 WULPUS as a case study

Opening

Wearable ultrasound hardware

Examples of the systems

Main building blocks

- Electronics
- Transducer
- Housing

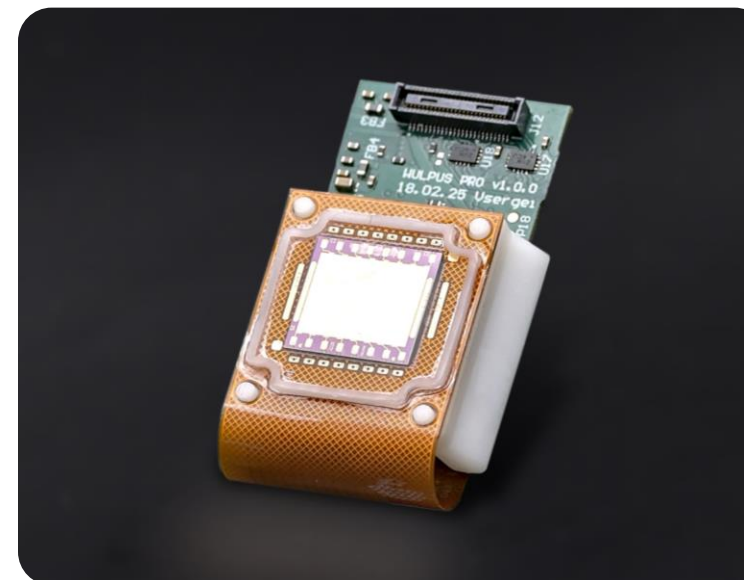
Functionality

- Excitation
- Receive
- Some (pre) processing
- Wireless data transmission

Key Features

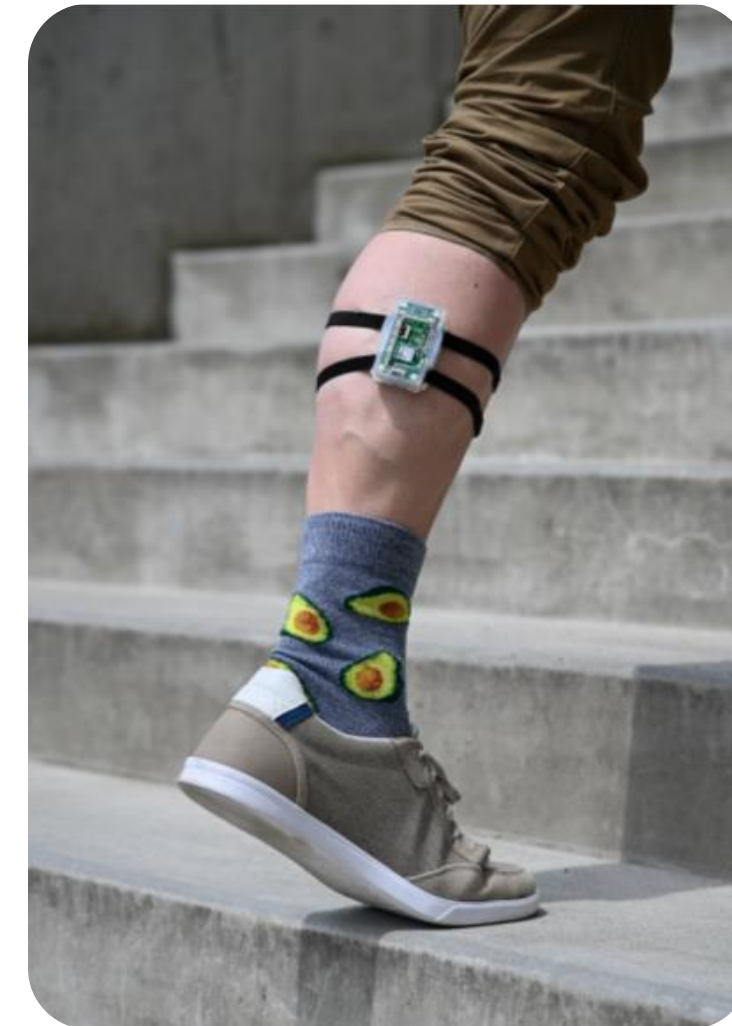
- Small, compact, skin conformal
- Battery powered
- Wireless

WULPUS (PRO)



ETH Zurich, IIS, D-ITET
Image: Sergei Vostrikov

TinyProbe



ETH Zurich, IIS, D-ITET
Image: Sergei Vostrikov

USoP



Image: UC San Diego Jacobs
School of Engineering

Anatomy of a wearable ultrasound probe

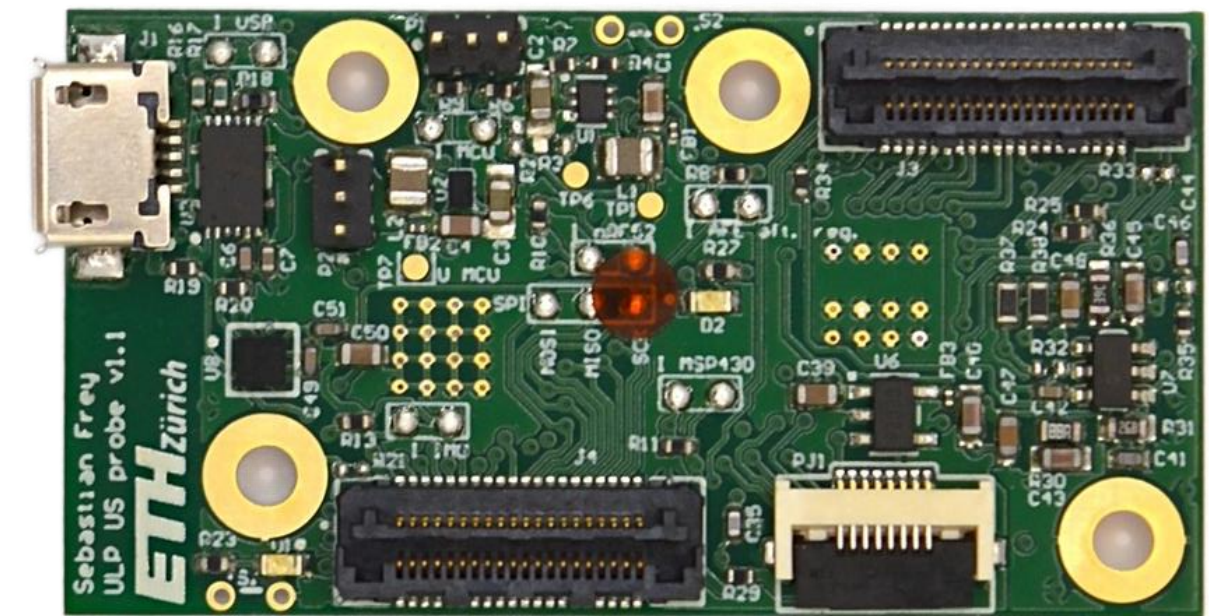
WULPUS as an example platform



WULPUS assembly
(No transducer)



Acquisition PCB



High-voltage PCB

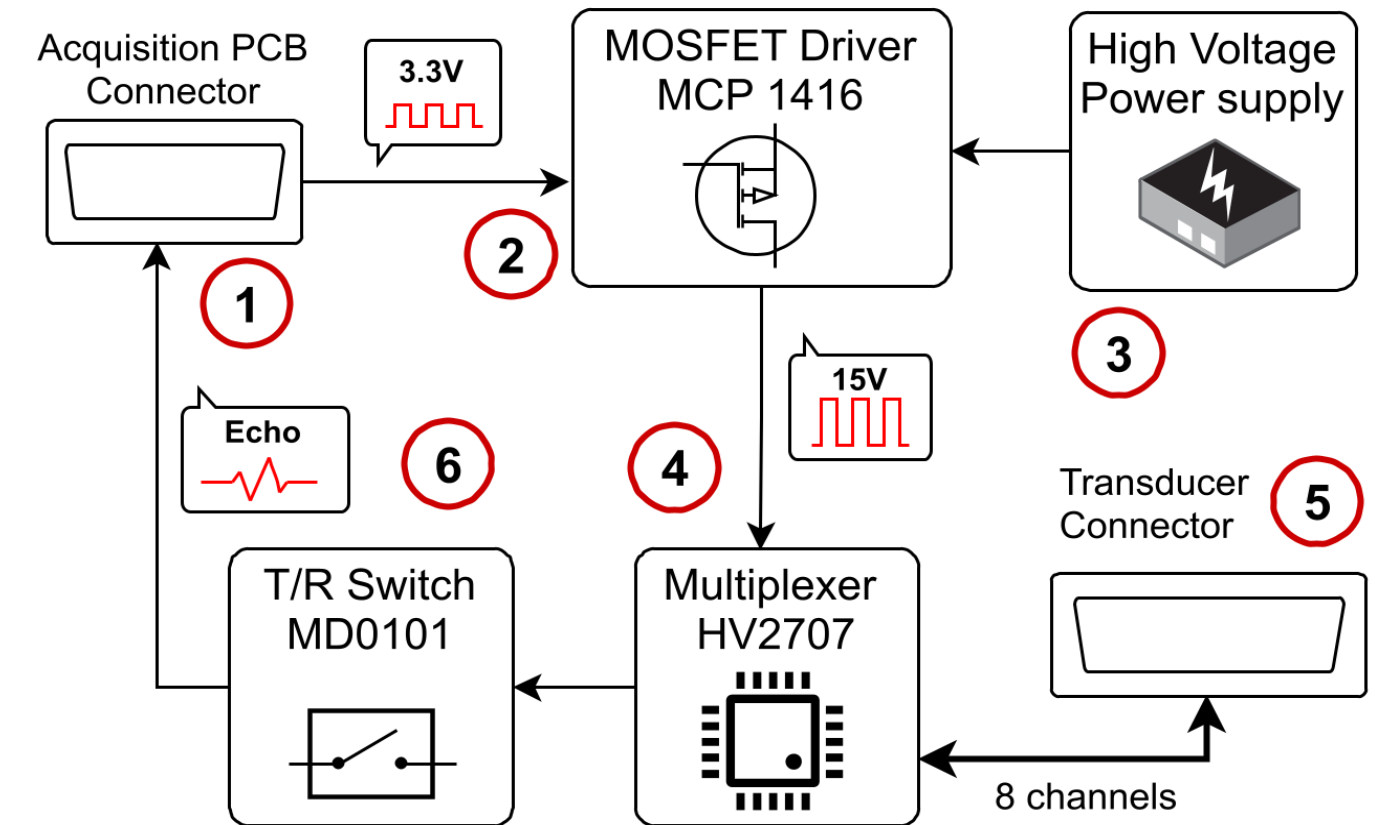
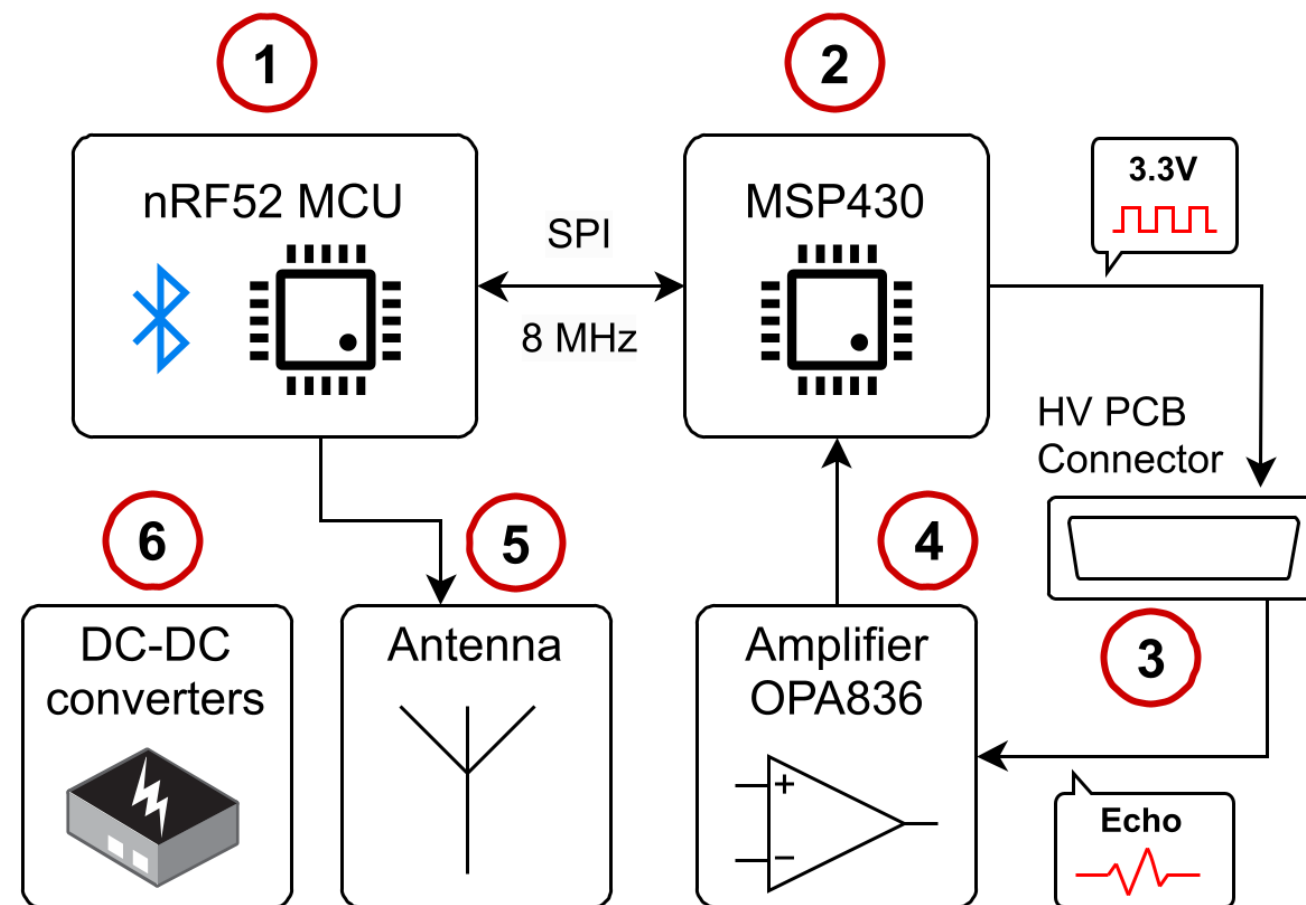
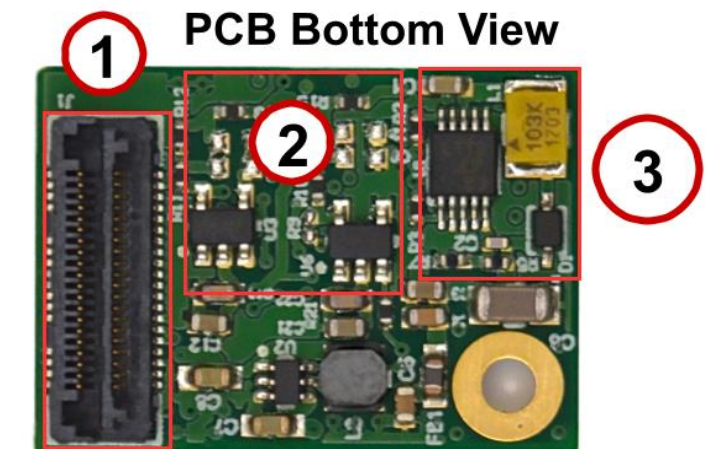
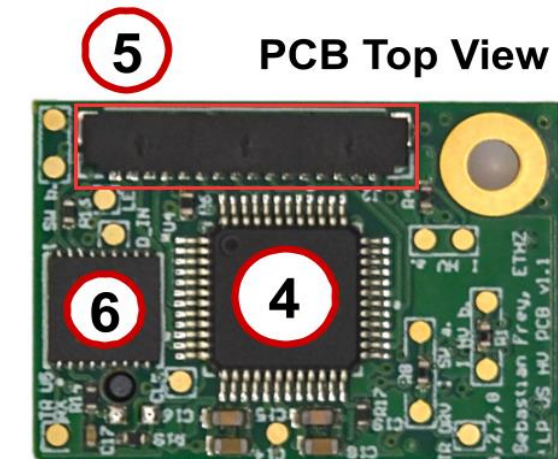
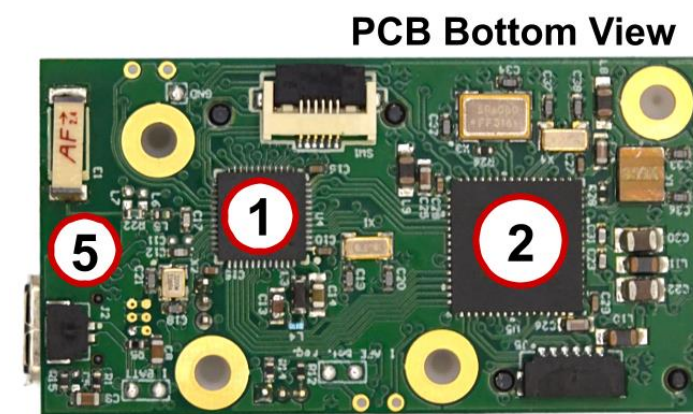
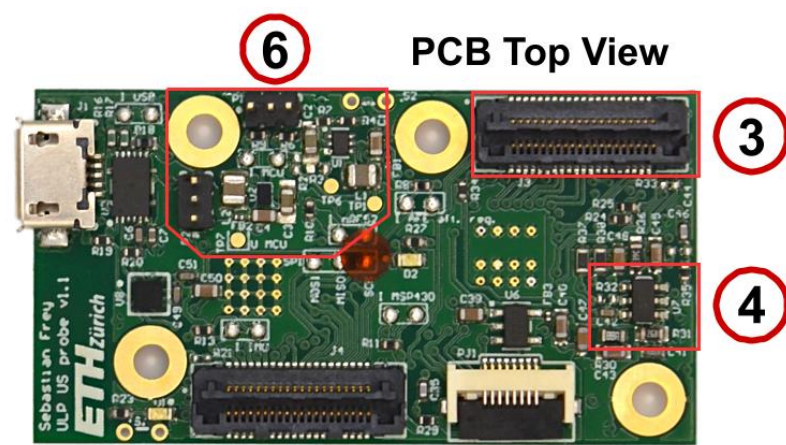


Silicone package



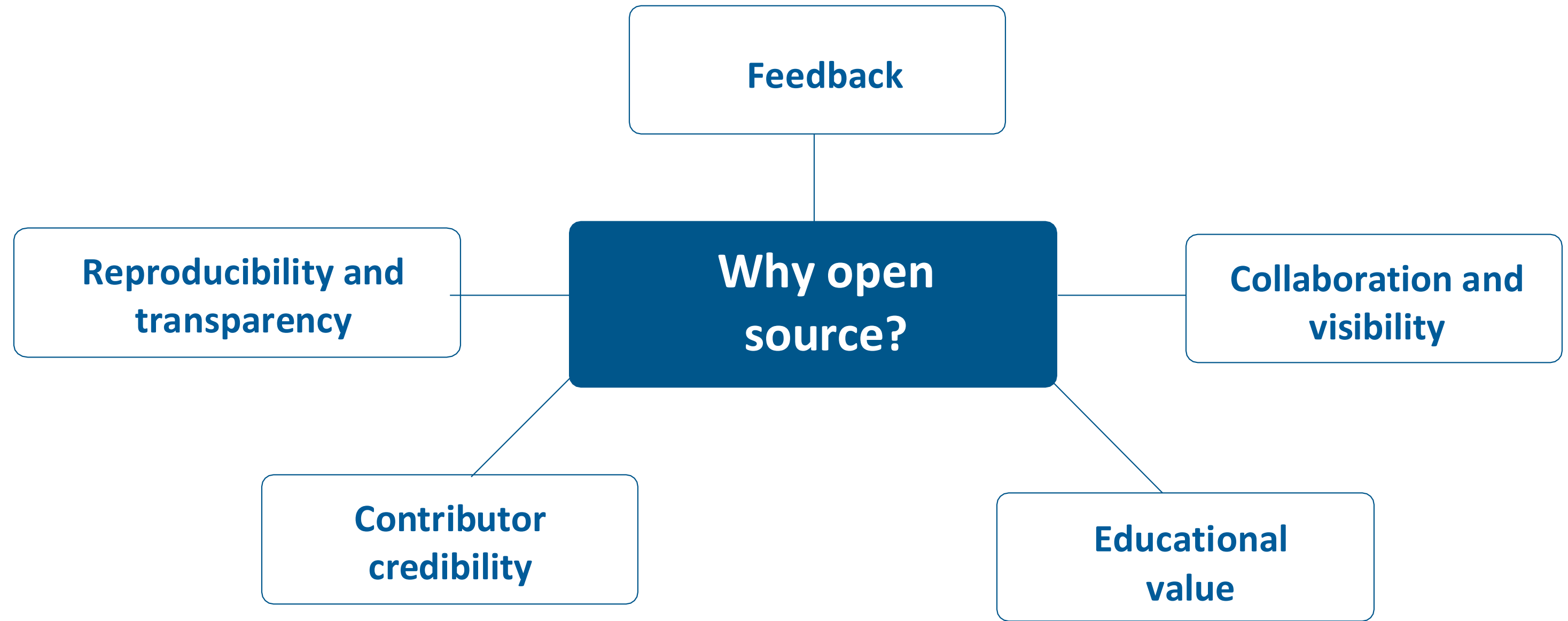
- 1. Opening
- 2. Why open source?
- 3. Current SoA
- 4. The rocky path
- 5. WULPUS as a case study

WULPUS Electronics Anatomy



- 1. Opening
- 2. Why open source?
- 3. Current SoA
- 4. The rocky path
- 5. WULPUS as a case study

Why Open Source?

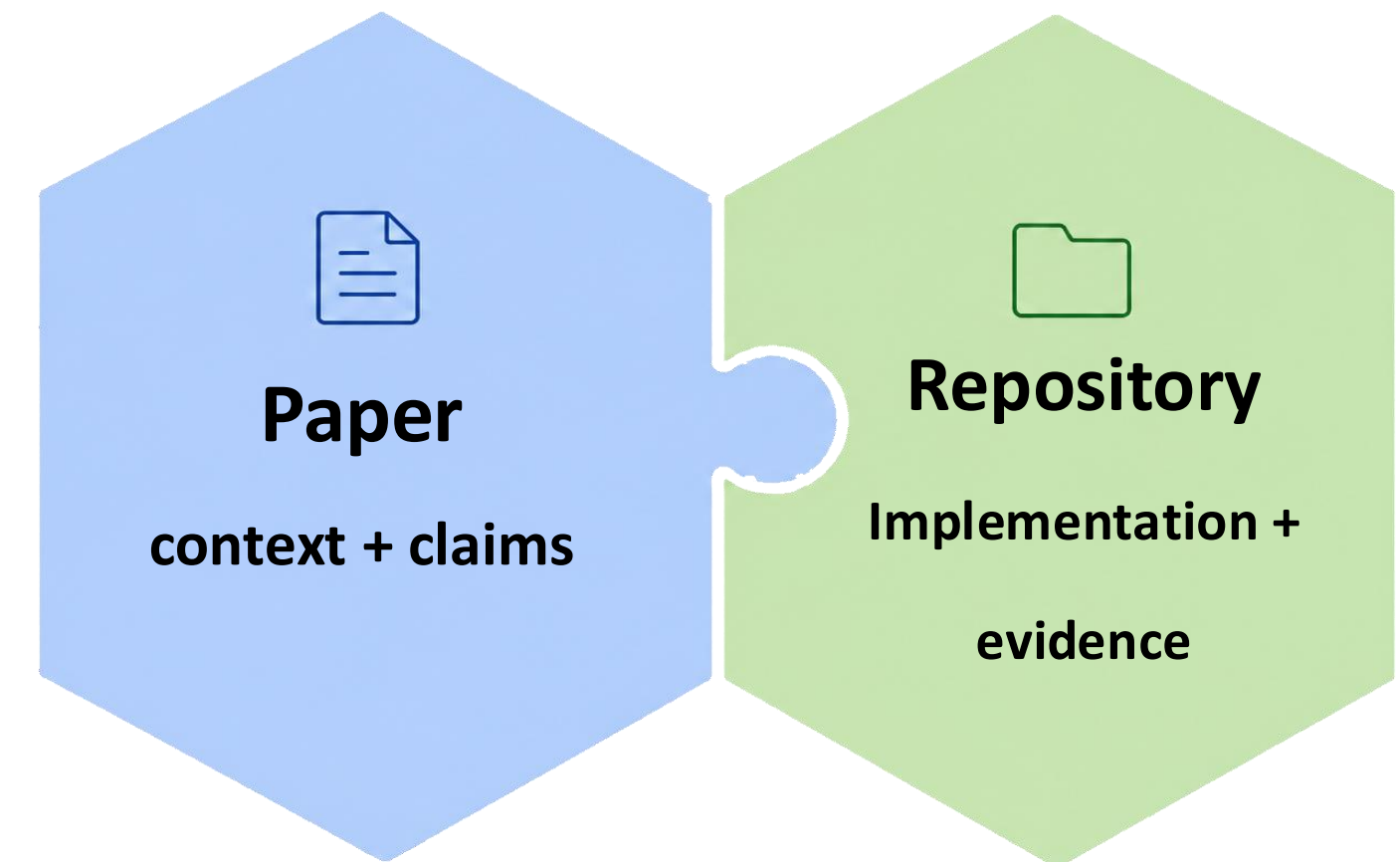


- 1. Opening
- 2. Why open source?**
- 3. Current SoA
- 4. The rocky path
- 5. WULPUS as a case study

Reproducibility and transparency

Papers tell the story but repositories reveal the implementation.

- **Papers are optimized for communication:**
they explain what was built, but rarely how to rebuild it.
- **Hardware reproduction needs complete artifacts:**
PCB files, firmware, gateware, BOMs, and build instructions.
- **Transparency includes limitations:**
open files expose what works, what is fragile, and what assumptions were made.



Together, the paper and repository turn results into reproducible evidence.

Feedback from outside the bubble

The community sees what the lab no longer sees.



Positive feedback fuels momentum and negative feedback steers direction.

Educational value of open-sourcing

Open source as learning infrastructure

Self-education



Releasing outside = explaining to others

You understand fully when you explain to people outside

Turning intuitive understanding -> Documentation

Student education



Good documentation
-> onboarding materials

Students learn from design notes, examples, user guides

The project becomes group memory, not one's private knowledge

Public education

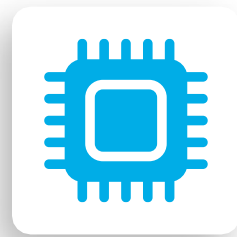


Open files share more than paper

Public money ->
Reusable public knowledge
(code, HW, methods, etc)

Collaboration and visibility

Open design invites to connect



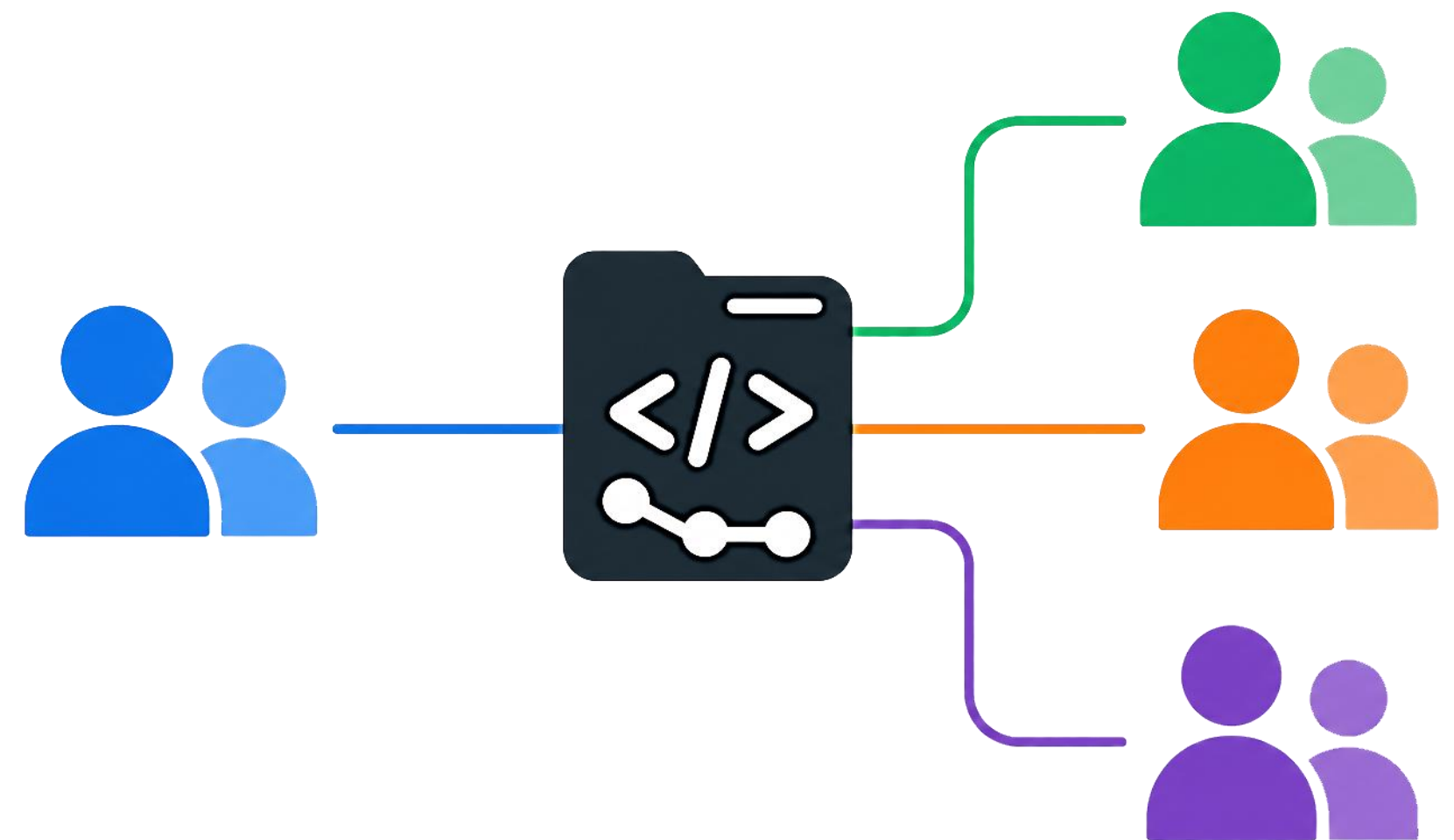
Physical hardware creates trust:
community sees that something really works.



A public project is engaging:
people can react, test, modify,
and build on it.



The repository exposes real design choices:
both the strengths and the weak spots of the design.



Contributor credibility

Open-sourcing helps the contributor, not only the project



Public commits create attribution:
They show who built, changed, and improved the system.



Issues and reviews expose technical reasoning:
Design decisions and trade-offs become traceable.



Open repositories build contributor credibility:
Engineering effort becomes visible beyond the paper author list.



1. Opening
2. Why open source?
3. Current SoA
4. The rocky path
5. WULPUS as a case study

Open development turns invisible engineering work into visible scientific contribution

Current SoA

Representative Open-Source Ultrasound Hardware Projects

Pulse-echo acquisition

Wearable platforms

Ultrasound infrastructure

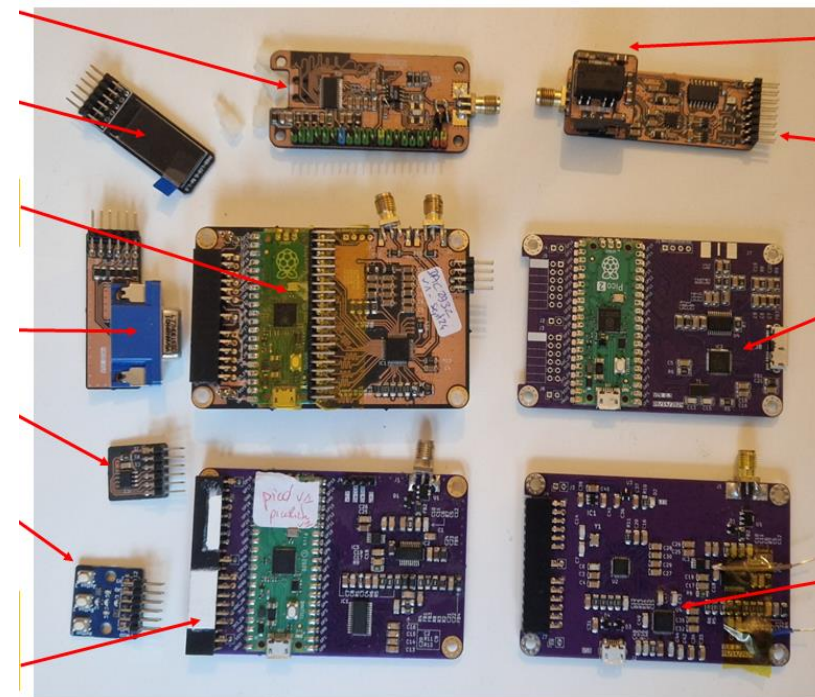
Unmaintained research prototypes

un0rick / lit3rick / pic0rick lineage

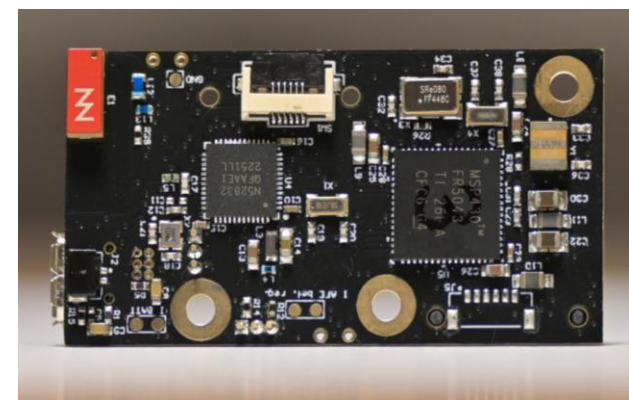
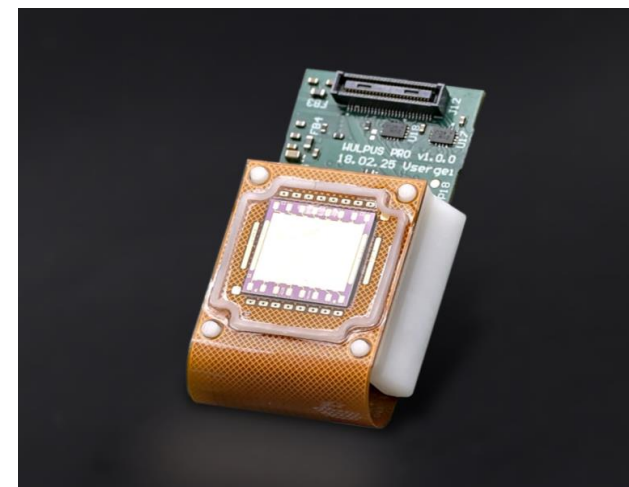
WULPUS family

open-UST

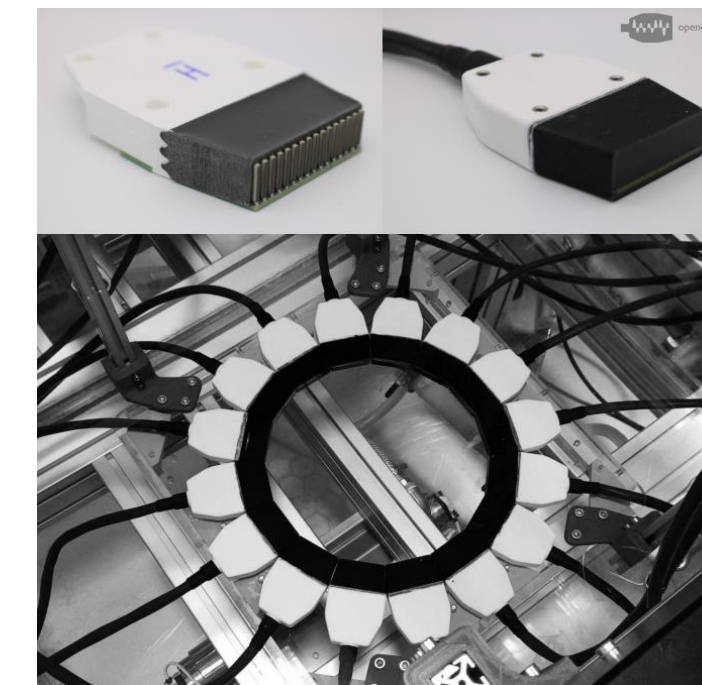
- One-off paper release
- Prototype-level robustness
- Incomplete hardware docs
- Incomplete licensing



Luc Jonveaux (kelu124), 2024



Ph: Sergei Vostrikov, 2026



Morgan Roberts, 2023



1. Opening
2. Why open source?
3. Current SoA
4. The rocky path
5. WULPUS as a case study

Representative Open-Source Ultrasound HW Projects

Open ultrasound exists but complete open platforms are still rare

Project / family	Type	Reproducibility	Docs	Build stack
<i>unOrick / lit3rick / picOrick</i>	Low-cost pulse-echo acquisition	High	Strong	Open from inception
<i>SimpleRick</i>	unOrick-inspired 2D imaging	Prototype	Moderate	Open derivative
<i>WULPUS family</i>	Wearable ultrasound probes	High	Strong / WIP for PRO	Vendor-dependent
<i>TinyProbe</i>	Multimodal wearable probe	Partial / WIP	WIP	Vendor-dependent
<i>open-UST</i>	UST transducer array	High, array only	Strong	Open-source

Selected hardware-oriented projects; software-only frameworks and one-off code/data releases are not shown.



1. Opening
2. Why open source?
3. Current SoA
4. The rocky path
5. WULPUS as a case study

From open repositories to sustainable platforms

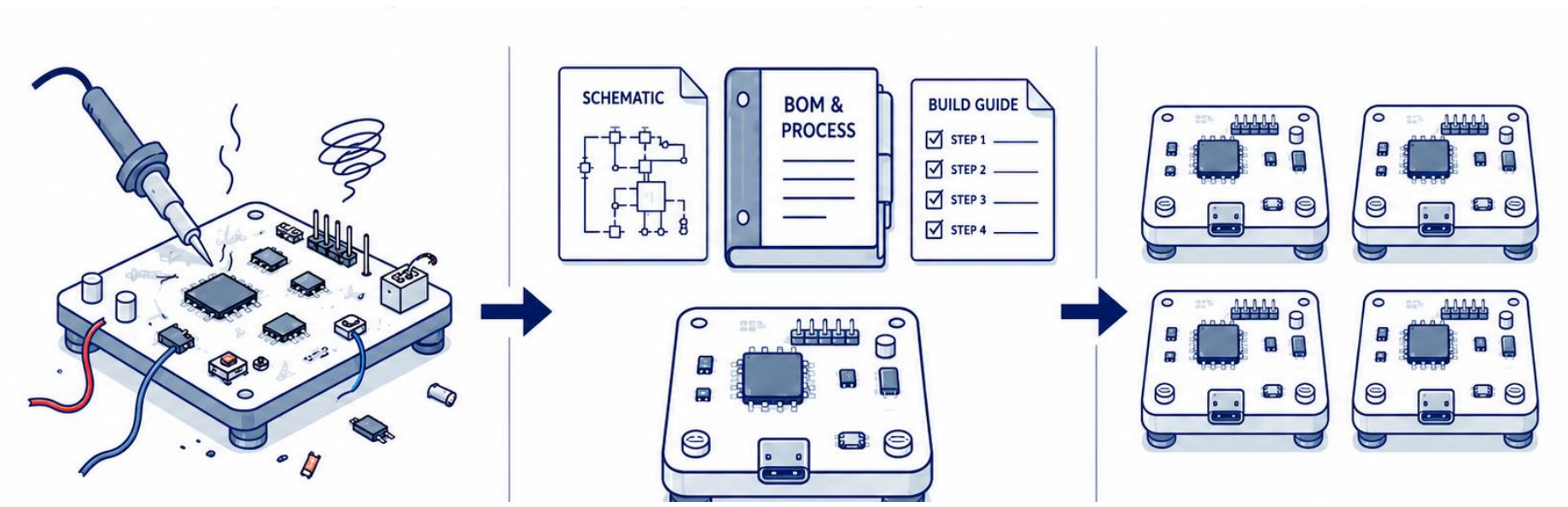
The SoA review reveals different degrees of openness

- 1
Step 1 is common: A public repository makes the work visible.

- 2
Step 2 is harder: Reproducibility requires documentation, BOMs, firmware, and validation.

- 3
Step 3 is rare: Sustainability requires maintenance, releases, and community support.

1. Public repository	2. Reproducible project	3. Sustainable platform
Files are visible	Others can rebuild it	Others can build on it
Often a final snapshot	Docs, BOM, firmware, validation	Issues, releases, maintenance
Common in academia	Harder to achieve	Rare



Why is it so hard to make it sustainable?



- 1. Opening
- 2. Why open source?
- 3. Current SoA**
- 4. The rocky path
- 5. WULPUS as a case study

The rocky path

The probe is not “just hardware”

Reproducibility depends on the full system around the PCB.

Wearable ultrasound is a multi-layer stack:

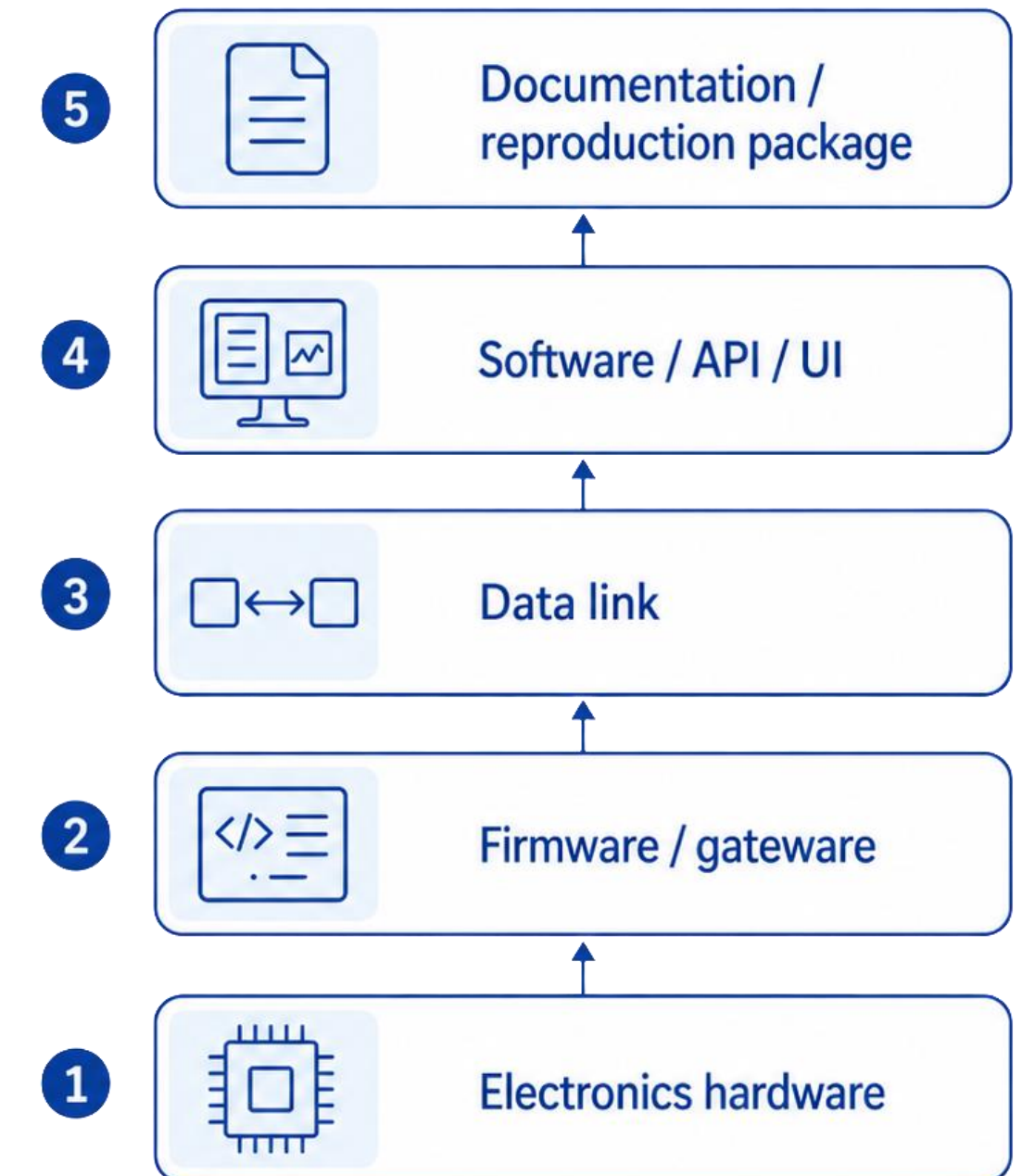
MCU, FPGA, data link, API/UI must be documented as one system

Reproducibility needs more than source/design files:

build guides, setup instructions, and test procedures.

Hidden infrastructure breaks reproducibility:

custom fixtures, adapter PCBs, cables, and calibration setups are part of the system and must be documented.



A working lab prototype is not yet a reproducible open-source platform.

One closed dependency can break reproducibility

A project is only as open as its least open dependency.

Critical dependencies define real openness:

CAD source files, firmware libraries, FPGA IP cores, and programming tools must be shareable or replaceable.

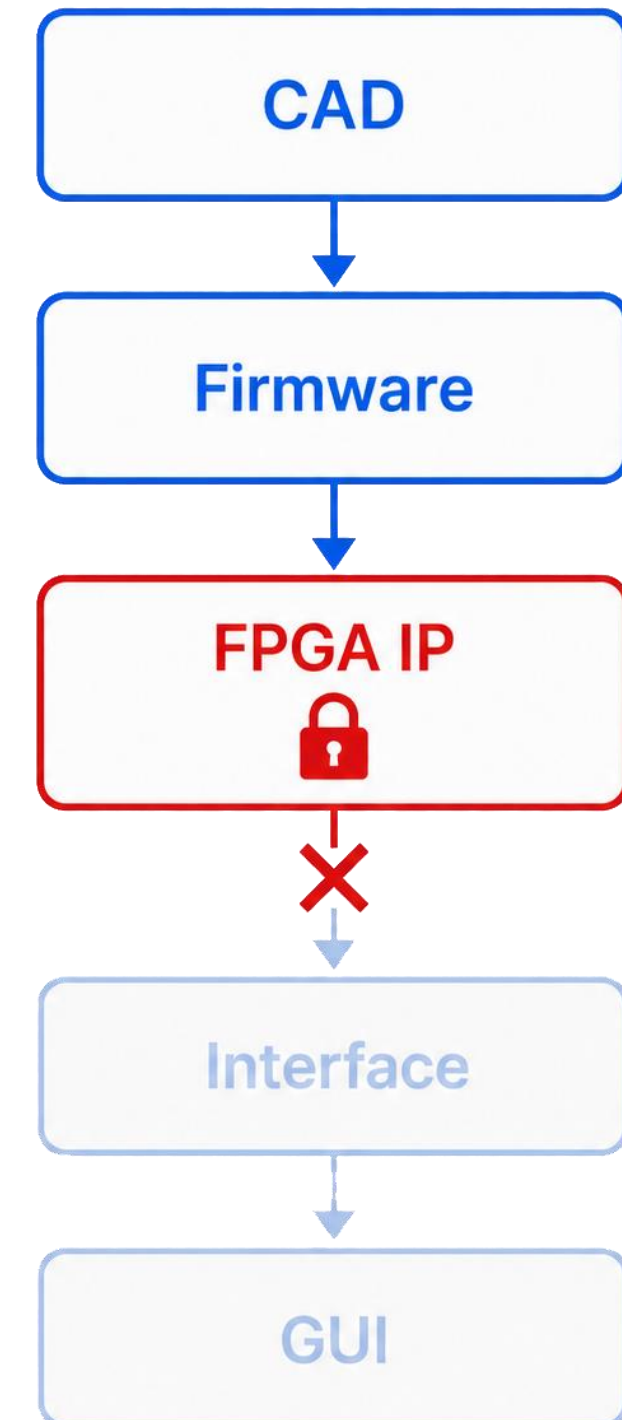
Closed pieces break reproducibility:

one proprietary IP core or toolchain can block reproduction.

Openness is hard to add at the end:

choices made early determine whether the final project can actually be released and rebuilt.

Openness has to be planned from the beginning.



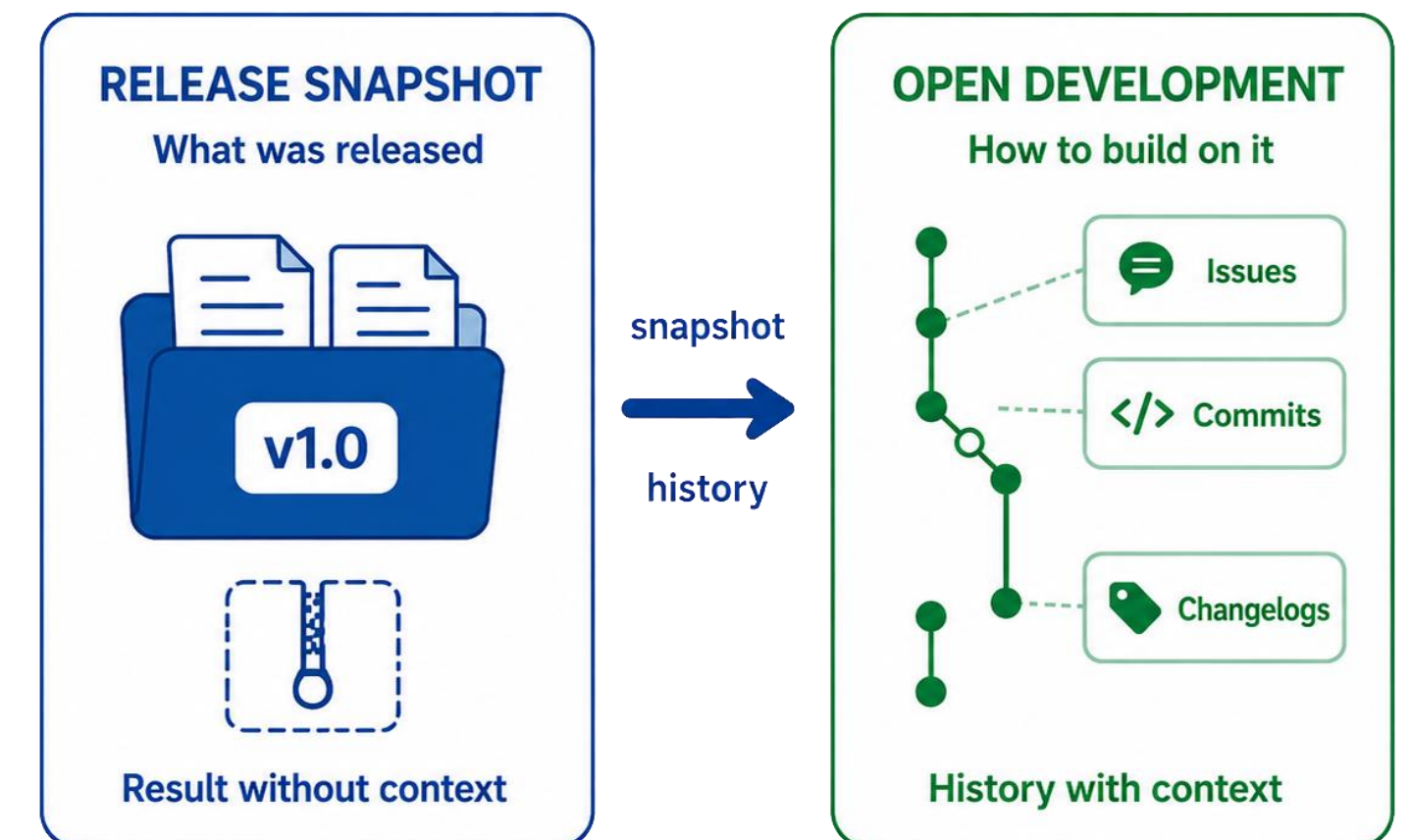
Public repository \neq open development

Files show what was released; history shows how to build on it.

A release snapshot hides design reasoning:
outsiders see the result, but not the trade-offs behind it.

Missing history slows adoption:
failed revisions, bug fixes, and changelogs are hard to reconstruct.

Building on top requires context:
commits, issues, release notes, and known limitations.



Open files let others copy the project; open development helps them to extend it.

The rocky path starts after the release

Sustained maintenance turns a release into something others can build on.

The first release gives a dopamine spike:

paper accepted, repository uploaded, social media post, first positive feedback.

Then the hidden work begins:

missing files, bugs, issues, emails, assembly questions, and component shortages.

The hidden work is maintenance:

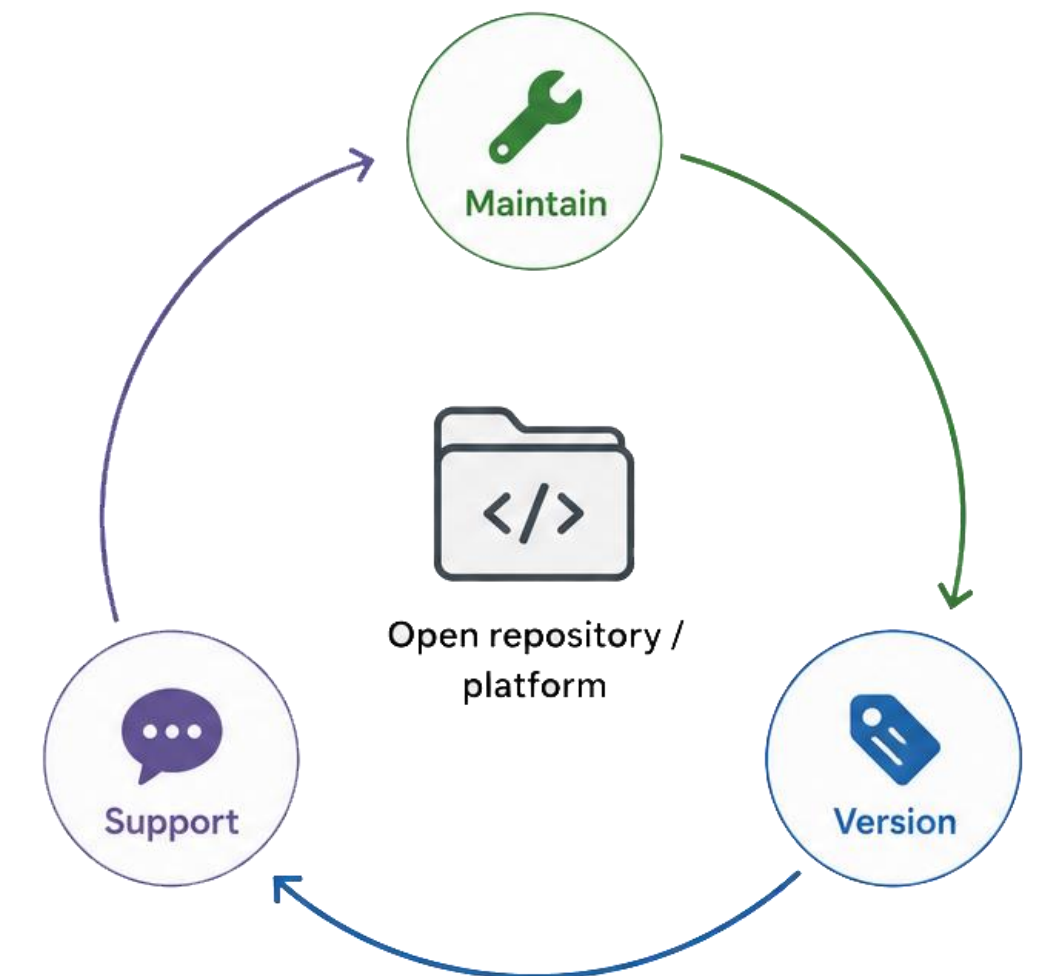
fixes, documentation updates, part substitutions, and user support make the project reusable.



A sustainable platform needs operating processes

A release becomes reusable only when it is maintained, versioned, and supported.

Maintenance	Versioning	Support
Bug fixes	Releases	Issues and questions
Documentation drift	Changelogs	Pull requests
Part substitutions	Known-good builds	User feedback
Toolchain changes	HW/SW compatibility	New use cases



The repository is just a visible artifact but the maintenance behind it makes it sustainable.

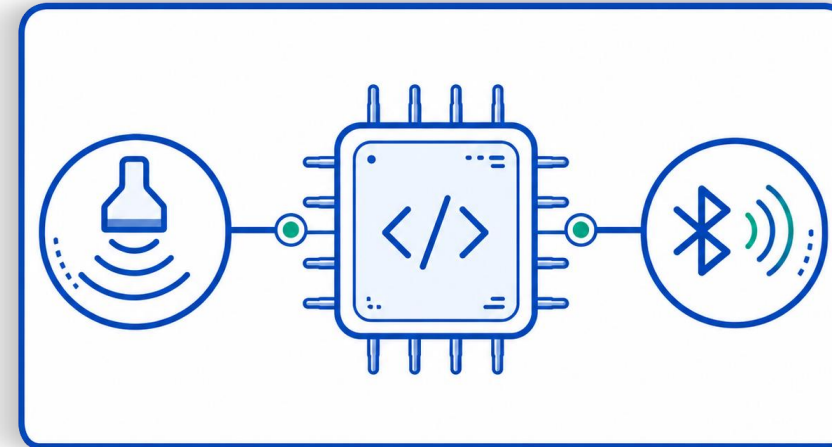
WULPUS as a case study

What is WULPUS Project?

Electronics Hardware



Firmware



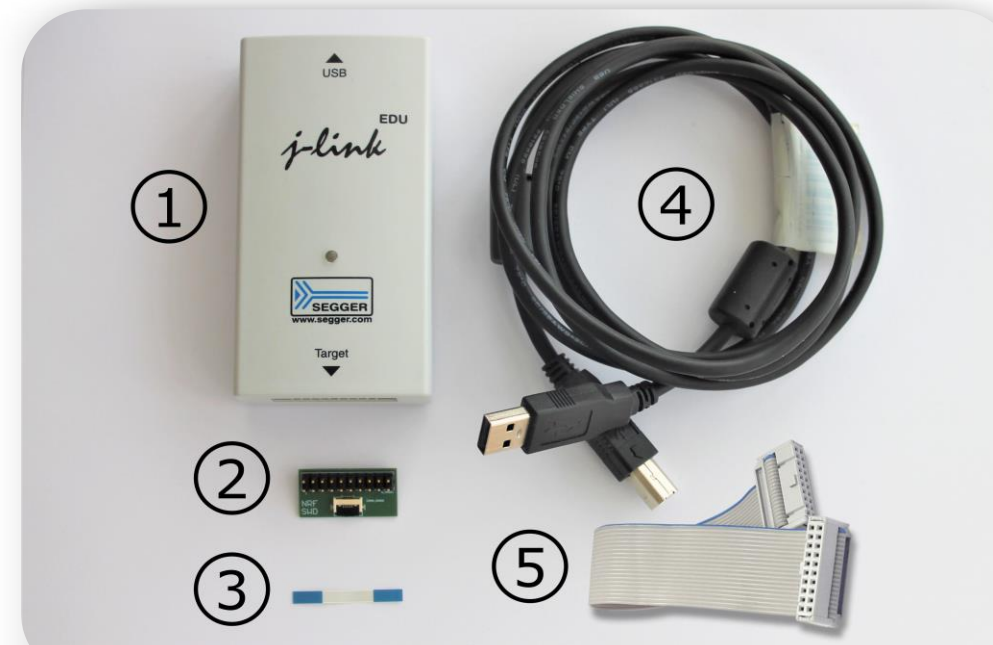
GUI

Measurement settings		Advanced settings	
Number of acquisitions	100	HV-MUX RX start time [us]	494
Acquisition Period [us]	321965	DC-DC turn on time [us]	100
Sampling frequency [Hz]	8000000.0	PPG start time [us]	500
Number of samples	400	ADC turn on time [us]	5
Receive (RX) gain [dB]	18.2	PGA in bias start time [us]	5
Excitation settings		ADC sampling start time [us]	509
Pulse frequency [Hz]	2250000	Capture restart time [us]	3000
Number of pulses	2	Capture timeout time [us]	3000
Filename:	uss_config_hv_artifact_5	Save	Load

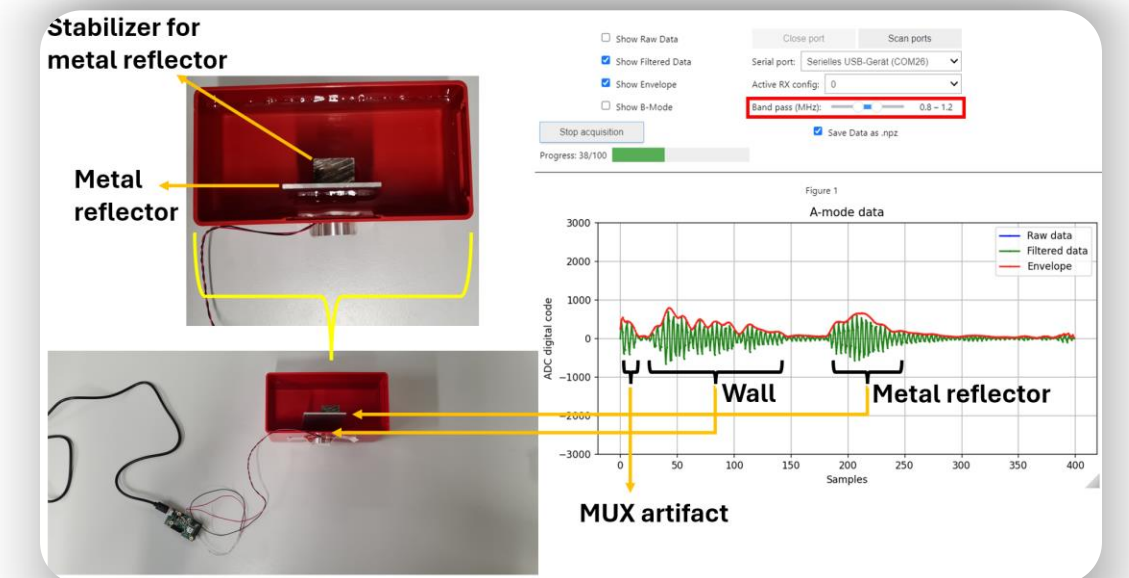
Housing



Reproduction Instructions

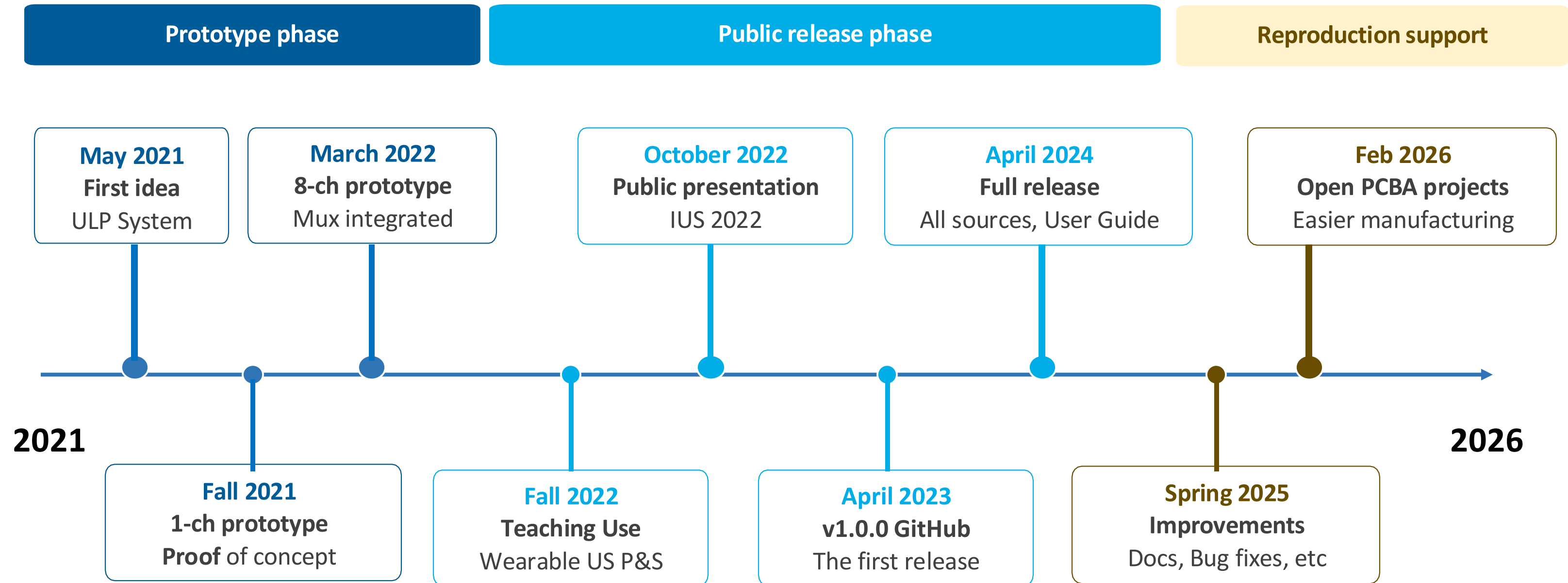


User Manual & Examples



1. Opening
2. Why open source?
3. Current SoA
4. The rocky path
5. WULPUS as a case study

WULPUS project timeline



WULPUS became reusable through repeated documentation, fixes, and manufacturing-oriented updates

Open source is a long-term investment

2

years from idea to initial release

3

years of development and user support

12+

academic/industrial group partners

10+

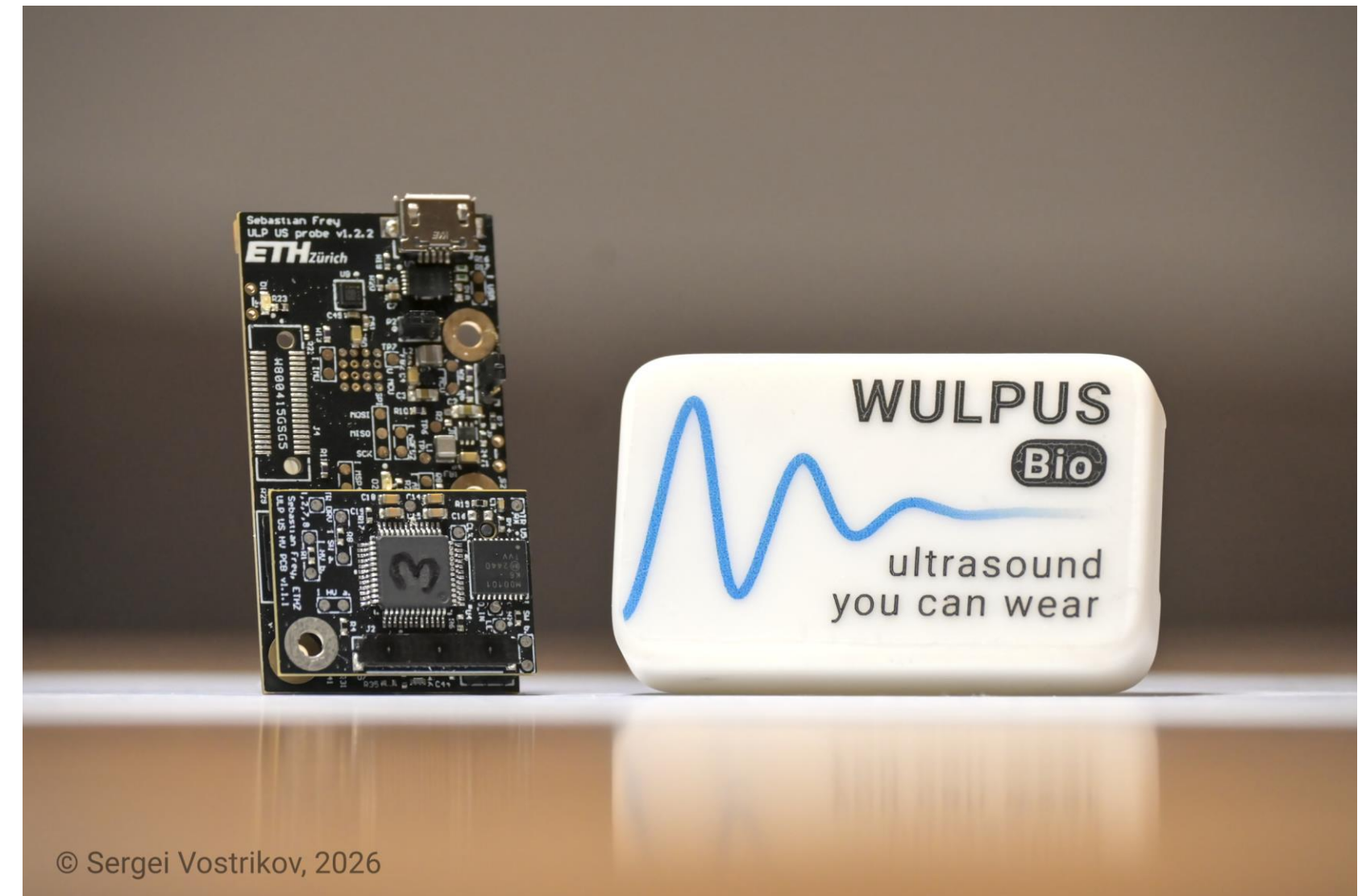
student projects

5+

derivative systems

100+

probes manufactured worldwide



1. Opening
2. Why open source?
3. Current SoA
4. The rocky path
5. WULPUS as a case study

Lessons from WULPUS Project

What worked

1. Grew beyond a conference paper
2. Used in teaching and by partners
3. Public release matured over time
4. Feedback improved reproducibility
5. Open PCBA projects eased manufacturing

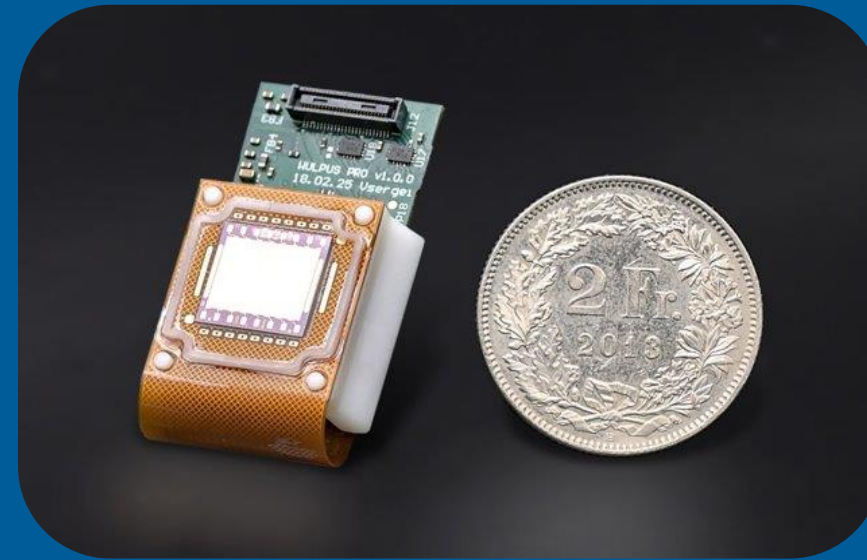
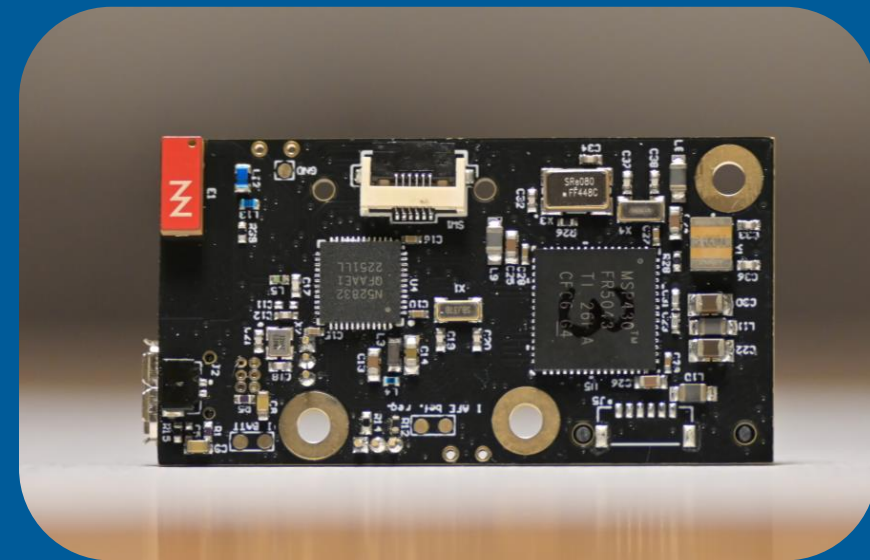
What I would improve

1. Reduce proprietary CAD dependency
2. Simplify debugging/programming
3. Improve transducer accessibility
4. Publish PCBA projects earlier
5. Move toward open development from day one



1. Opening
2. Why open source?
3. Current SoA
4. The rocky path
5. **WULPUS as a case study**

Questions?



Repositories:

<https://github.com/pulp-bio/>

<https://github.com/Sergio5714/>

Email:

sergei.vostrikov@rheonics.com

