

# gr-pocketsdr: Bringing PocketSDR Front-Ends to GNU Radio and GNSS-SDR Ecosystem

**Minhaj Uddin Ahmad**

Ph.D. Student | [mahmad12@crimson.ua.edu](mailto:mahmad12@crimson.ua.edu)

**Connected and Automated Mobility Lab @ UA**

**GNU Radio Conference 2026**



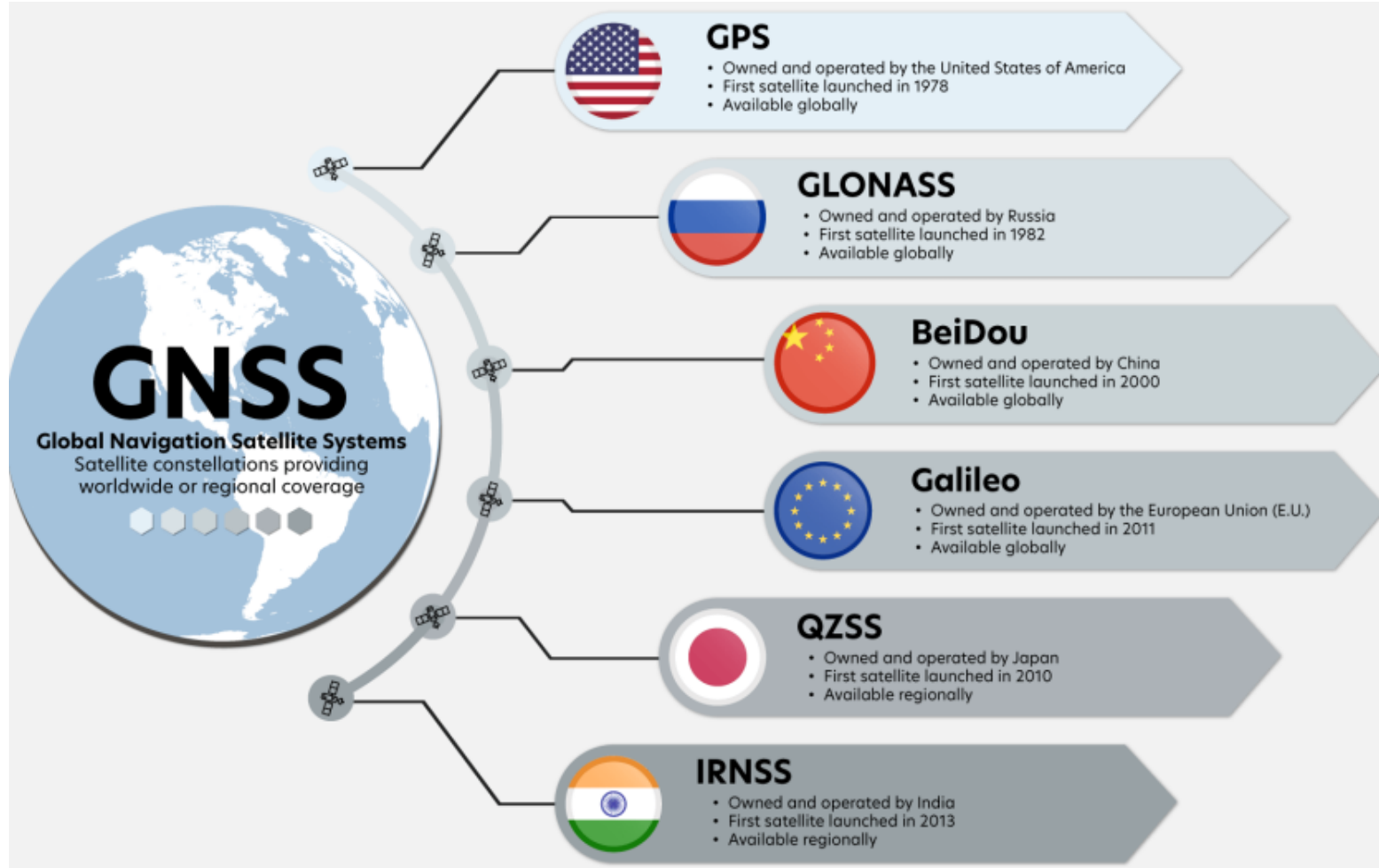
Alabama Transportation  
Institute



**Raleigh, North Carolina**



# Global Navigation Satellite System



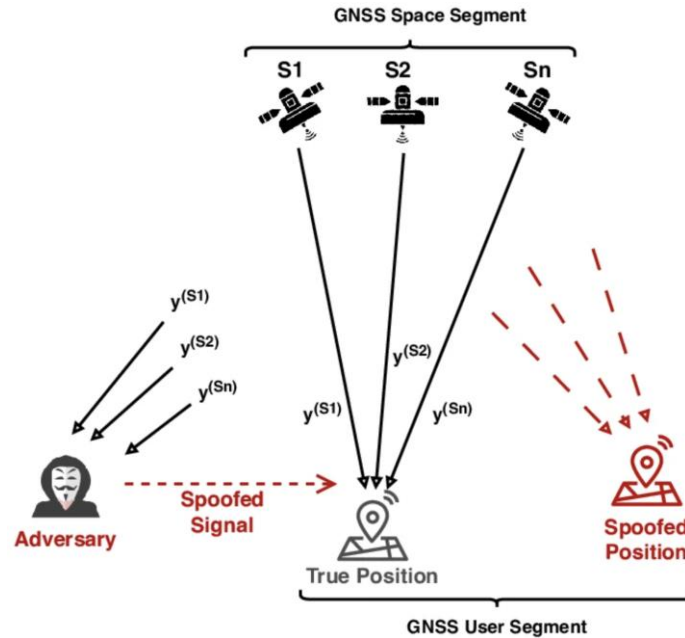
GNSS satellite signals are open for anyone / any device to receive, for free, anywhere on earth.

With an SDR receiver, you can receive the signals and process it in your laptop, turning your laptop into a GNSS receiver. This allows researchers to develop and test new algorithms and signal processing techniques.

Since GNSS signals are open, without any protection mechanism (mostly), there are safety and security concerns too, requiring active research in this area.

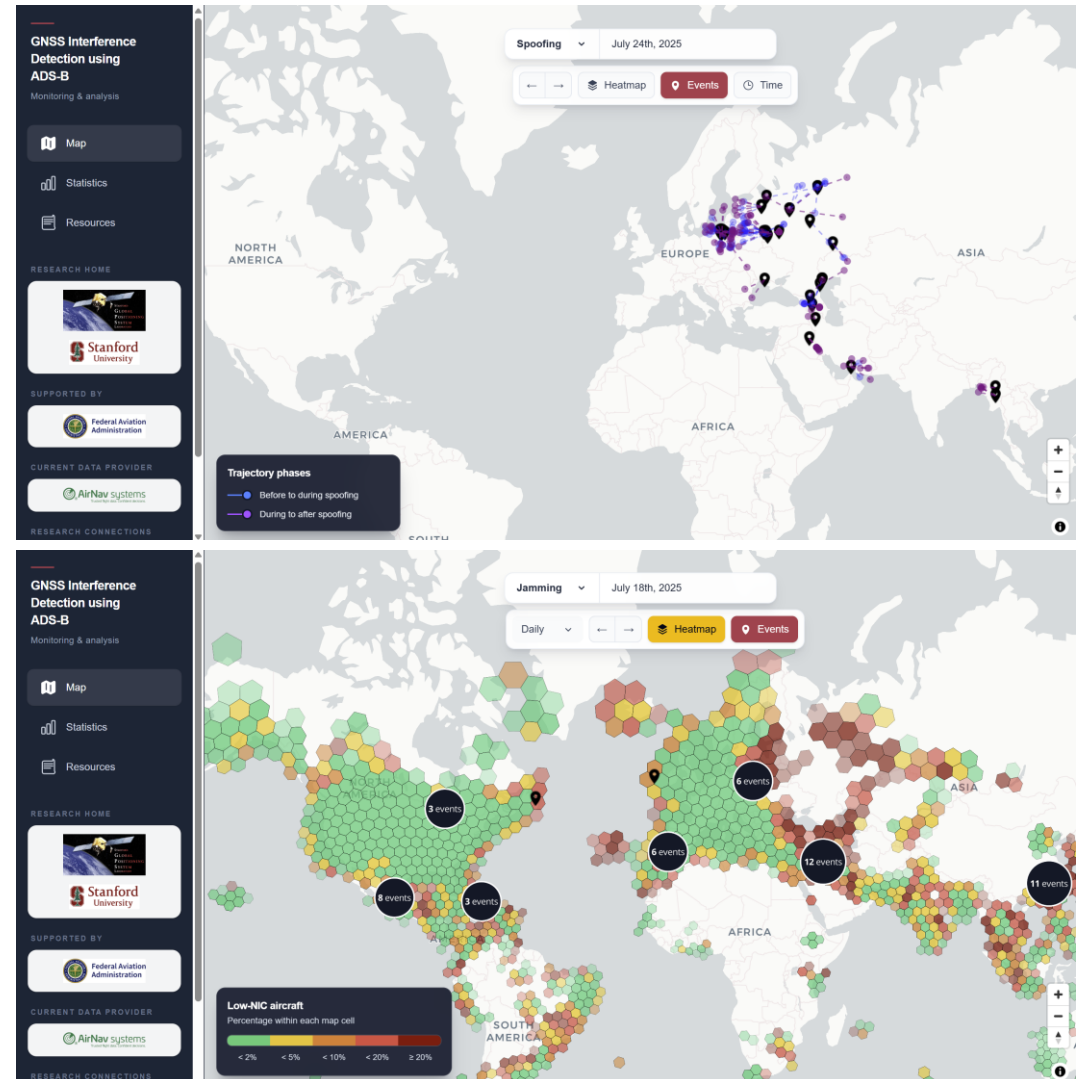
<https://eos-gnss.com/wp-content/uploads/2023/04/GNSS-Infographic-1-1024x576.png>

# We need more research for GNSS resiliency

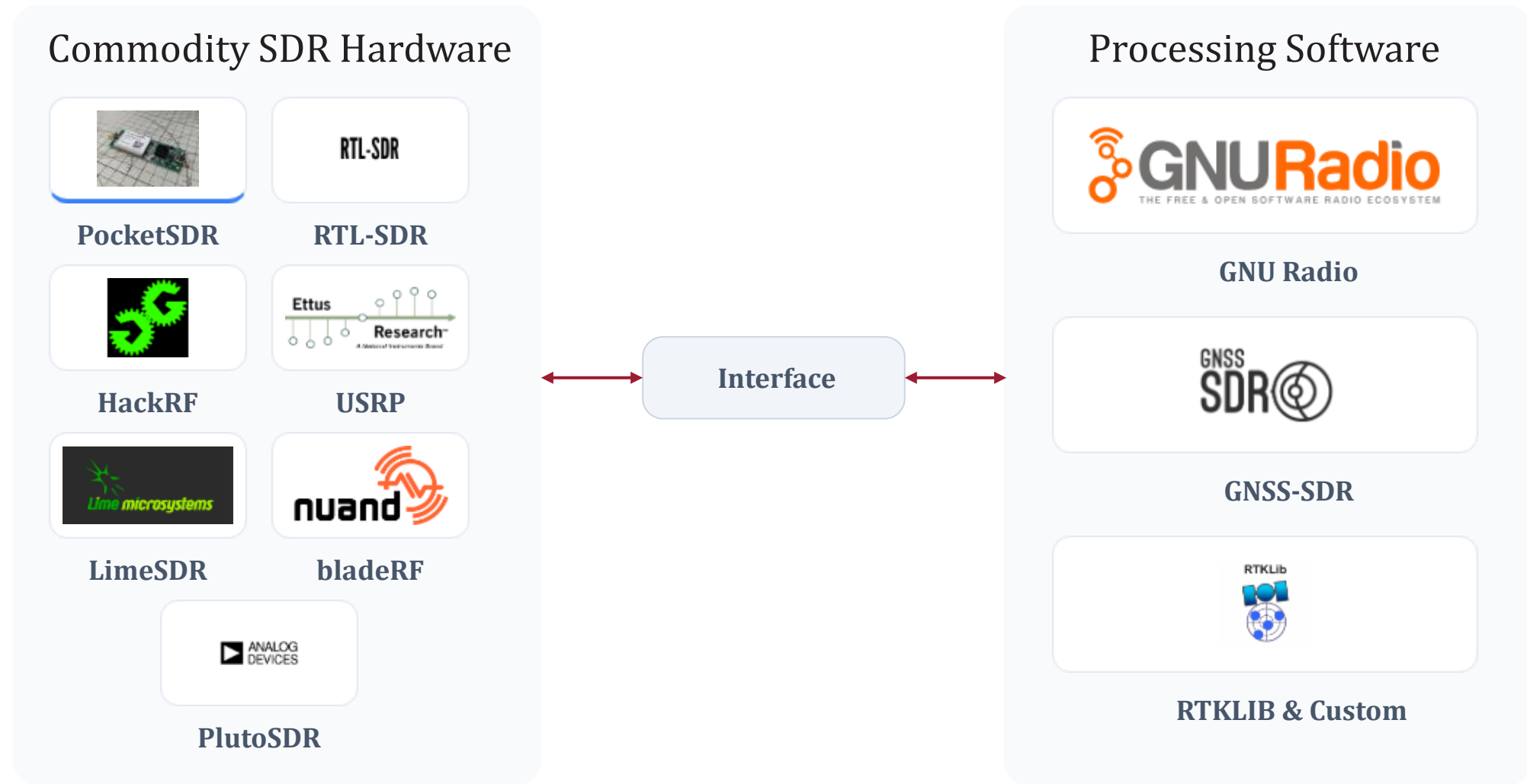


Spoofing concept

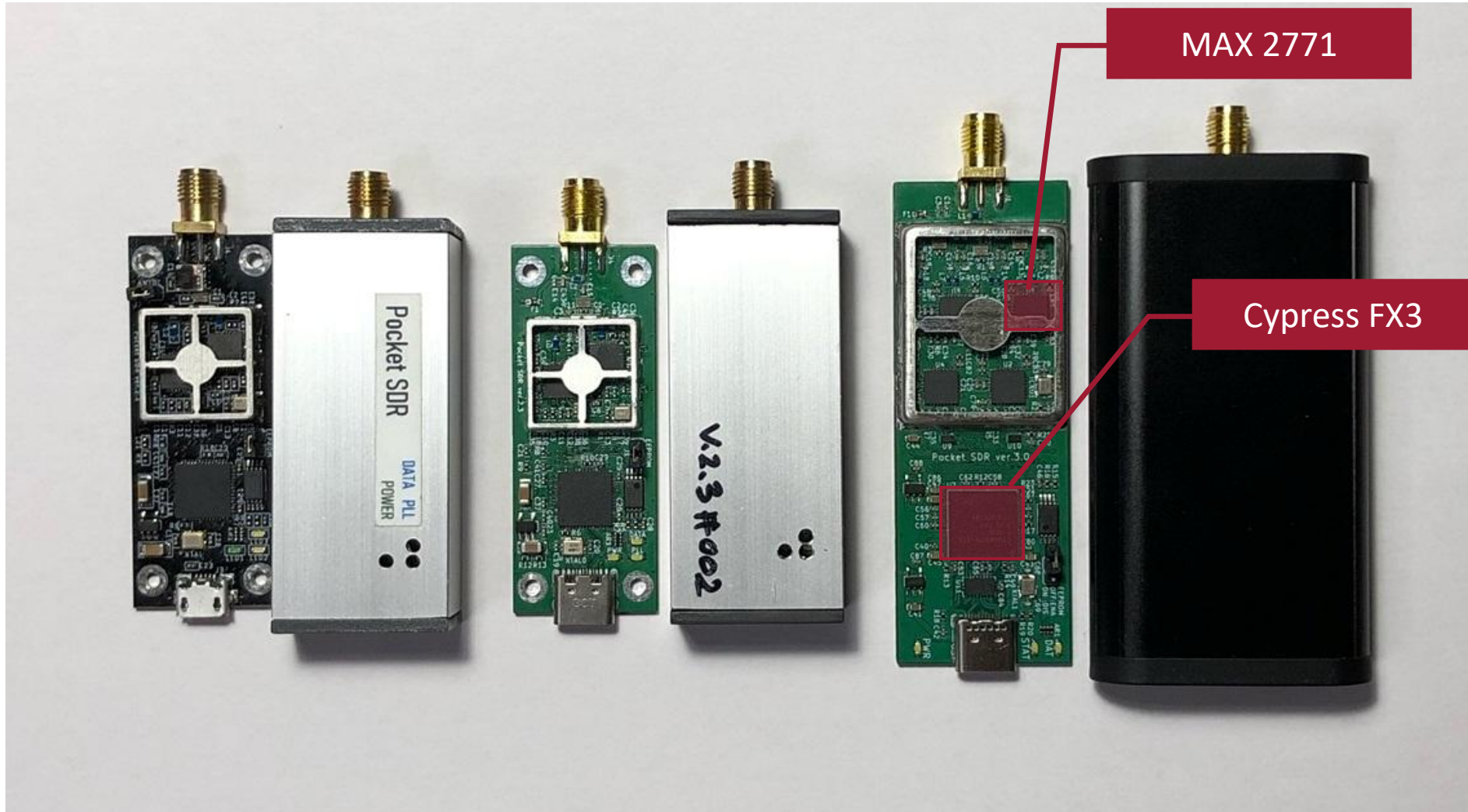
[https://www.researchgate.net/publication/382288564\\_CLPDI-based\\_Fast\\_Secure\\_Code\\_Estimation\\_and\\_Replay\\_Attacks\\_on\\_GNSS\\_Signals](https://www.researchgate.net/publication/382288564_CLPDI-based_Fast_Secure_Code_Estimation_and_Replay_Attacks_on_GNSS_Signals)



# Broad hardware support makes research accessible



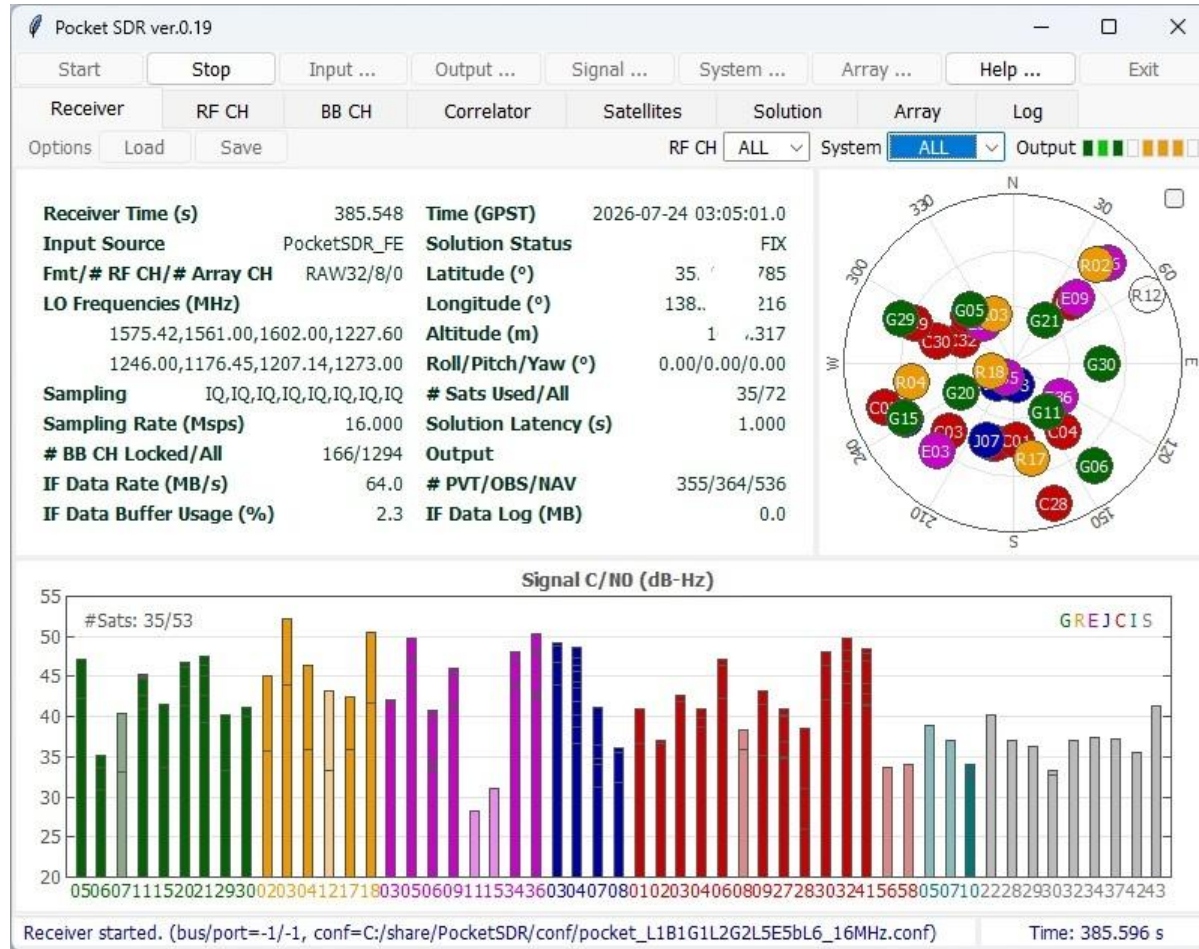
# PocketSDR



<https://github.com/tomojitakasu/PocketSDR>

<https://www.datagnss.com/products/pocketsdr-gnss-receiver>

# General Purpose GNSS Processing



<https://github.com/tomojitakasu/PocketSDR>

The screenshot shows the GNSS-SDR website homepage. The top navigation bar includes links for Quick-Start Guide, Documentation, GeNiusS Place, News, and About Us, along with a search icon. The main header features the GNSS-SDR logo and an "Install Now" button. Below this is a large banner with the text "An open-source Global Navigation Satellite Systems software-defined receiver." and "Current release: v0.0.21". Underneath the banner are three buttons: Watch (119), Star (2,278), and Fork (735).

The main content area is divided into three columns, each with an icon and a title:

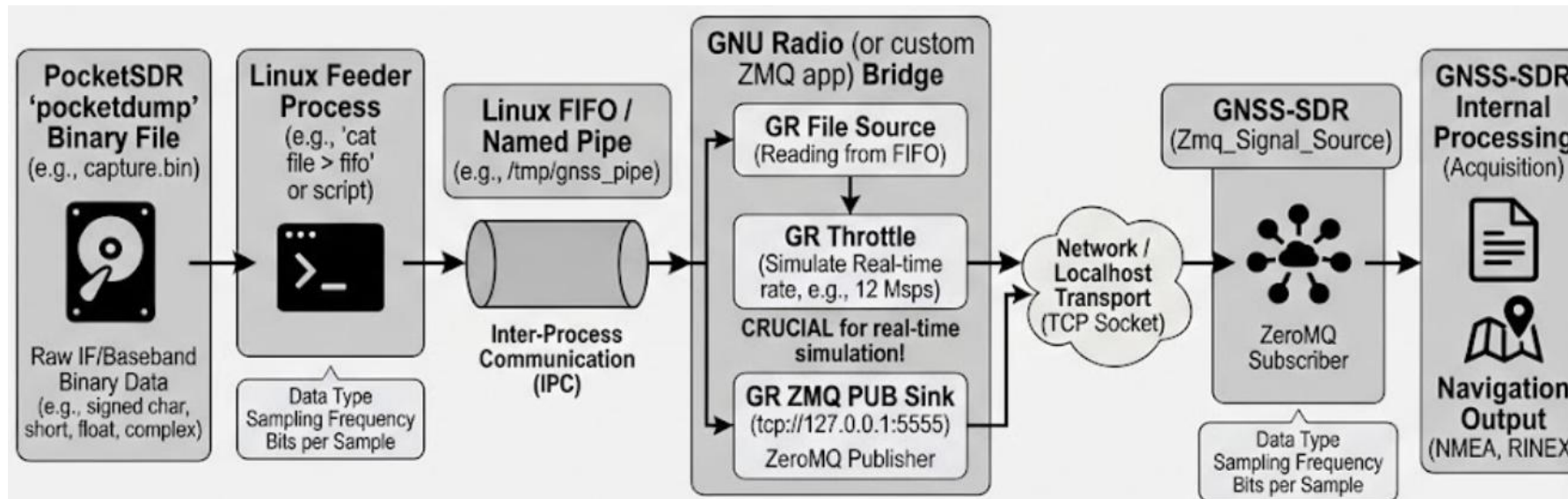
- Using the software receiver**: Get your first position fix using GNSS-SDR and a file containing raw signal samples. Includes a "Read more" button.
- Configuring processing blocks**: Documentation on available signal processing blocks' configuration options. Includes a "Read more" button.
- Key Performance Indicators**: A discussion on the assessment of software-defined GNSS receivers. Includes a "Read more" button.

General Purpose - <https://gnss-sdr.org/>

# Traditional workflow

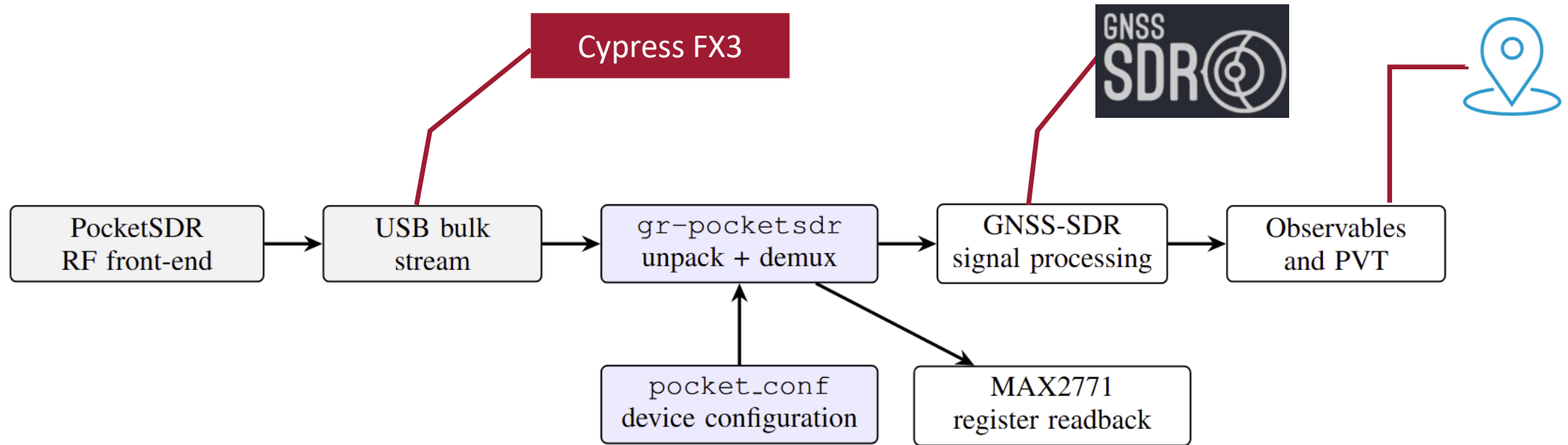


Offline file post-processing



Workaround for real-time processing

# Building the bridge for live reception



# Lets compare the gr-pocketsdr vs. ZMQ workaround

|                      | Pocket_SDR_Signal_Source | ZMQ_Signal_Source |
|----------------------|--------------------------|-------------------|
| Total CPU            | 178.9 %                  | 205.1 %           |
| Latency, median      | 320.6 ms                 | 334.8 ms          |
| Latency, worst case  | 374.9 ms                 | 679.7 ms          |
| Lag spread (jitter)  | 58 ms                    | 236 ms            |
| Wire rate, 1 channel | 16 MB/s                  | 64 MB/s           |
| Reports lost samples | yes                      | no                |

Medians differ by 14 ms — worst cases differ by 305 ms

# Application – Lets test the PocketSDR clock

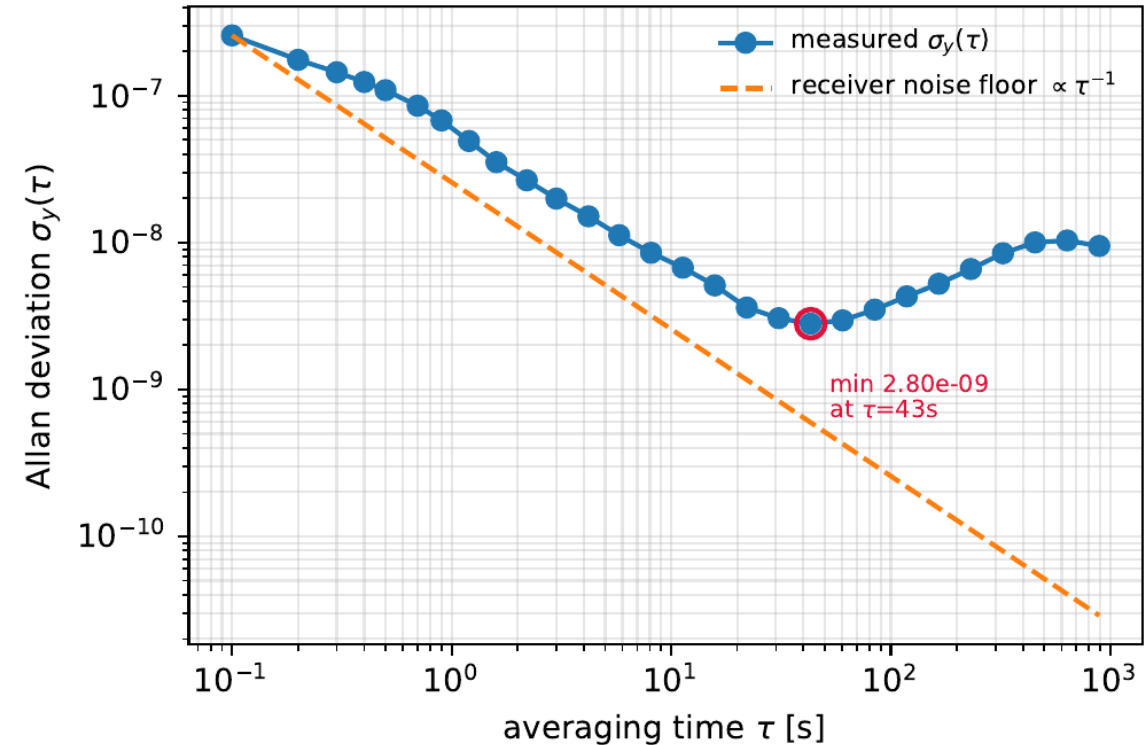
Clock offset = TCXO phase error vs GPS time

37.6 min unbroken fix, 22 584 epochs at 10 Hz

Frequency offset: -0.265 ppm

Stability floor  $2.8 \times 10^{-9}$  at  $\tau = 43$  s

Above the receiver noise floor throughout



# Open-source!

minhaj6 / gr-pocketsdr Public

Notifications Fork 4 Star 3

<> Code Issues Pull requests Actions Projects Security and quality Insights

main Go to file <>

minhaj6 better document registers 798a747 · 3 days ago

.conda init 3 months ago

apps init 3 months ago

cmake init 3 months ago

conf typical pocketsdr configs fro... 2 weeks ago

docs datasheets for reference 2 weeks ago

examples updated about gr-pocketsdr s... 3 months ago

grc init 3 months ago

include/gnuradio/pocketsdr better document registers 3 days ago

lib better document registers 3 days ago

python/pocketsdr init 3 months ago

tests init 3 months ago

.clang-format init 3 months ago

.cmake-format.py init 3 months ago

.gitignore typical pocketsdr configs fro... 2 weeks ago

GNU Radio block for DataGNSS PocketSDR (Max 2771) front-ends. Can be used as GNSS-SDR signal source.

Readme

GPL-3.0, License licenses found

Activity

3 stars

1 watching

4 forks

Contributors 1

minhaj6 Minhaj Ahmad

C++ 54.4% CMake 27% Python 11.6% C 5.3% Other 1.7%

Report repository

Source code available at GitHub

## Implementation: Pocket\_SDR\_Signal\_Source

Pocket SDR is an open-source GNSS software-defined radio platform. Its Pocket SDR FE 2CH/4CH/8CH front-ends can stream RF samples to GNSS-SDR through the gr-pocketsdr GNU Radio out-of-tree module.

**Warning:** This Signal Source is only available from the `next` branch of the upstream GNSS-SDR repository. It will be included in the next stable release.

In order to make use of this block implementation, you need to build GNSS-SDR from the source code after installing `gr-pocketsdr`. Once `gr-pocketsdr` is installed, build GNSS-SDR passing the flag `-DENABLE_POCKETSDDR=ON` at configure time:

```
$ cd gnss-sdr && mkdir build && cd build
$ git checkout next
$ git pull upstream next
$ cmake -DENABLE_POCKETSDDR=ON ..
$ make && sudo make install
```

The device is configured with a standard Pocket SDR configuration file (`pocket_conf` format) passed with the `conf_file` parameter. If `conf_file` is empty, the device keeps its current settings. GNSS-SDR reads back the

|                                   |                                 |
|-----------------------------------|---------------------------------|
| IMPLEMENTATION:                   | MULTICHANNEL_FILE_SIGNAL_SOURCE |
| READING DATA FROM A PIPE          |                                 |
| IMPLEMENTATION:                   | FIFO_SIGNAL_SOURCE              |
| READING DATA FROM A ZEROMQ STREAM |                                 |
| IMPLEMENTATION:                   | ZMQ_SIGNAL_SOURCE               |
| RADIO FREQUENCY FRONT-ENDS        |                                 |
| IMPLEMENTATION:                   | UHD_SIGNAL_SOURCE               |
| IMPLEMENTATION:                   | OSMOSDR_SIGNAL_SOURCE           |
| IMPLEMENTATION:                   | RTL_TCP_SIGNAL_SOURCE           |
| IMPLEMENTATION:                   | FMCOMMS2_SIGNAL_SOURCE          |
| IMPLEMENTATION:                   | PLUTOSDR_SIGNAL_SOURCE          |
| IMPLEMENTATION:                   | LIMESDR_SIGNAL_SOURCE           |
| IMPLEMENTATION:                   | BLADERF_SIGNAL_SOURCE           |
| IMPLEMENTATION:                   | POCKET_SDR_SIGNAL_SOURCE        |
| MULTIPLE RADIO FREQUENCY CHAINS   |                                 |
| MULTIPLE SOURCES                  |                                 |

Merged with GNSS-SDR upstream, development branch

# Conclusions and future work

---

It works! (few quirks to work out)

14.6 % less CPU than ZMQ workaround, 305 ms better at worst case jitter

4× less data transmission compared to ZMQ workaround

Get reports of ring-buffer overruns

Multi-frequency (GPS L1, L2, L5, Galileo) testing is open for future work

# Thank You Questions?

Supported by the National Center for Transportation Cybersecurity and Resiliency (TraCR), a U.S. DOT National University Transportation Center, and by NSF Award #2340456.

**Contact:** Minhaj Uddin Ahmad | [mahmad12@crimson.ua.edu](mailto:mahmad12@crimson.ua.edu)



THE UNIVERSITY OF  
**ALABAMA**<sup>®</sup>

Alabama Transportation  
Institute

