



Project Overview/Update

EU GR Days
Josh Morman, GR President
jmorman@gnuradio.org

What is GNU Radio? What Distinguishes GNU Radio as an SDR Framework?



What is GNU Radio?

An SDR Application 

Enables Applications

An SDR Framework 

An SDR Community 

An SDR Ecosystem



Where is GNU Radio Used

Wireless Communications

RADAR

4G/5G/6G

Radio Astronomy

Spectrum Monitoring

IOT and Sensors

Space Comms

!!Particle Accelerators!!

Education

Amateur Radio

Physics Research

Security Research

Public Safety

Citizen Science

Transportation

Recreation (CtF)

Weather

Medical

RFML

...

THEME
GNU Radio enables innovation in
scientific applications

Conferences > 2024 IEEE International Confe... ?

Evaluating the Performance of GMSK Modulation for Measuring SpO2 Using Software-Defined Radio and Universal Software Radio Peripheral

Publisher: IEEE

Cite This

PDF

Namarig Mohamed Taha ; Amin Babiker Abdelnabi ; Ahmed M. Alhassan ; EL Hadj Dogheche ; Iyad Dayoub ; Julio Cesar [All Authors](#)

Conferences > 2024 XVI Congreso de Tecnolog... ?

Building a Radio Community Across Students, Industry and Enthusiasts with Capture the Flag

Publisher: IEEE

Cite This

PDF

Federico Larroca ; Gonzalo Belcredi ; Romina García ; Gastón García González ; Lucas Inglés ; Camilo Mariño ; Martín Randall [All Authors](#)

Deployable 5G emergency networks using software defined radios: a case study

Tamer Omar and Adrien Placentia

Published Online: February 1, 2024 · pp 11-18 · <https://doi.org/10.1504/IJMNDI.2024.136442>



GNU Radio Ecosystem and Community

General Assembly / Board

SETI Institute

Teams

Interest Groups

Codebase - Maintainers

OOT Modules (cgran.org)

Chat

Tutorials

Website

Wiki

Mailing List

Socials

Affiliated Projects

- VOLK

- IQEngine

- GQRX

- SigMF

- SatNogs

...

Conferences

Hackfests



Leads/GA Josh Morman <President> Martin Braun <Vice President> Marc Lichtman <Vice President> Derek Kozel <SETI PI> Andrej Rode Bastian Bloessl Johannes Demel Jeff Long <Maintainer> Marcus Müller <Architect> Nate Temple Philip Balister Samantha Palazzolo Jacob Gilbert Jean-Michel Friedt Ben McCall John Sallay	GR 4.0 Dev Lead: Josh Morman Members: Bastian Bloessl Marcus Müller John Sallay	GRC Lead: Håkon Vågsether - grc@gnuradio.org Members: Derek Kozel
Documentation Lead: Vacant - docs@gnuradio.org Members: Barry Duggan Ben McCall Cyrille Morin (Notou) Bailey Campbell	GRCon Lead: Samantha Palazzolo - grcon@gnuradio.org Members: Josh Morman Barry Duggan Derek Kozel	SigMF (Signal Metadata Standard) Lead: Jacob Gilbert - sigmf@gnuradio.org Marc Lichtman Members: Teque5 gmabey
VOLK (SIMD Optimization) Lead: Johannes Demel - volk@gnuradio.org Members: Michael Dickens	Infrastructure and CI/CD Lead: Andrej Rode - admin@gnuradio.org Members: Marc Lichtman	Education and Academic Outreach Lead: Jean-Michel Friedt Members:

Leadership

Organized under Articles of Association (AoA) - <https://github.com/gnuradio/gr-governance>

General Assembly

Board (3)

SETI PI

Technical Roles

Community Roles

SETI Institute

Partners in SDR technology development, application, and education

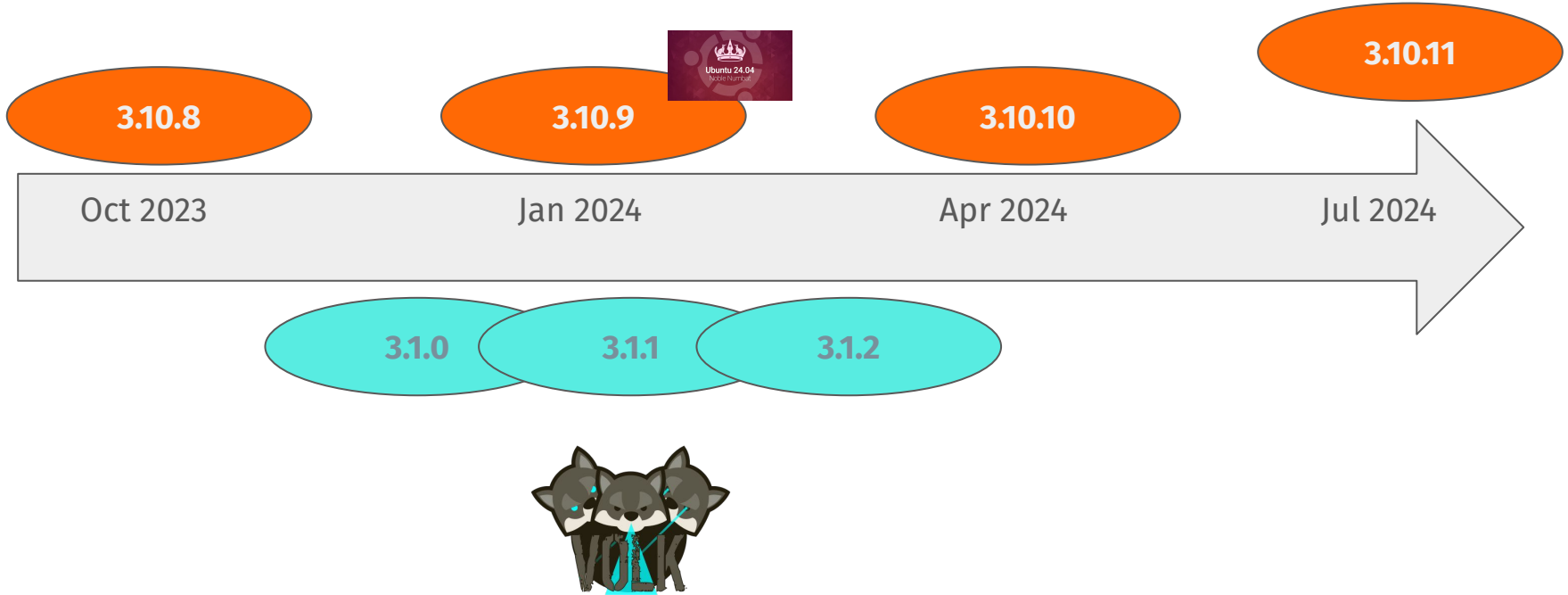
Home of GNU Radio's finances and logistics

Facilitating and enabling access to grants

Outreach and Educational experience and opportunities



Current State of GR3





GR3 Maintenance

Looking for additional maintainers!

- Be a recognized part of the maintenance team
- Own a module or group of modules (e.g. gr-qt, gr-soapy, gr-zeromq)
- Be responsible for ensuring relevant PRs are reviewed and merged
- Help backport merges to maint-3.10

Reduce the number of maintenance releases

- Quarterly has been great, but a lot of overhead
- Release management help appreciated

Email: info@gnuradio.org for more information

Development Activity

ARDC Grant

(<https://www.ardc.net/apply/grants/2022-grants/grant-gnu-radio-usability-enhancements/>)

- Installation of GNU Radio and out-of-tree modules (OOTs)
 - Native installers integrated with CI - Kitware
- Ongoing software maintenance and support
 - Marcus Mueller thrashing the issue tracker
- GNU Radio Companion (GRC) (Haakon V)
 - `gnuradio-companion --qt`
- Updated Tutorials
 - Wavewalker DSP
- Documentation Infrastructure
 - Bailey Campbell
- Documented Packet Modem Example with GR4
 - Dani Estevez



Grant: GNU Radio Usability Enhancements

Date: March 2022

Amount: \$263,011

GNU Radio is a free, open-source software-development toolkit that provides signal processing blocks to implement software radios. GNU Radio is always striving to be accessible to anyone across the globe, regardless of which operating system they are using and how much experience they have with wireless communications and digital signal processing. Historically, Windows operating system users have not had adequate support, despite it being the operating system used by nearly all K-12 students. Increasingly, macOS is becoming the platform of choice for both students and individuals. GNU Radio wants to be more intuitive and make it easier to install third-party modules known as out-of-tree modules (OOTs).

The GNU Radio project has identified a number of improvements to GNU Radio that it hopes will make GNU Radio easier to use, more accessible, and easier to maintain. These improvements are broken down into the following categories:

- Installation of GNU Radio and out-of-tree modules (OOTs)
- Documentation
- Ongoing software maintenance and support
- GNU Radio Companion (GRC)

The projects were carefully chosen to extend and advance work already underway, but hindered by lack of specialist experience. This grant from ARDC will allow GNU Radio to hire experts specialized in each area. We believe the specific GUI projects this ARDC grant would fund will also give the new GRC the final push that it needs to become the interface that ships with GNU Radio and provides users a much-improved experience. For each improvement category described above, one or more mentors (who will be volunteers from the GNU Radio leadership or core developers) will guide the work. These mentors will help the contractors stay aligned to the goals and offer feedback at periodic intervals. This grant will unlock the considerable existing capabilities to a large population in a timely manner and bring new people into the community.

Learn more at <https://www.gnuradio.org/>.

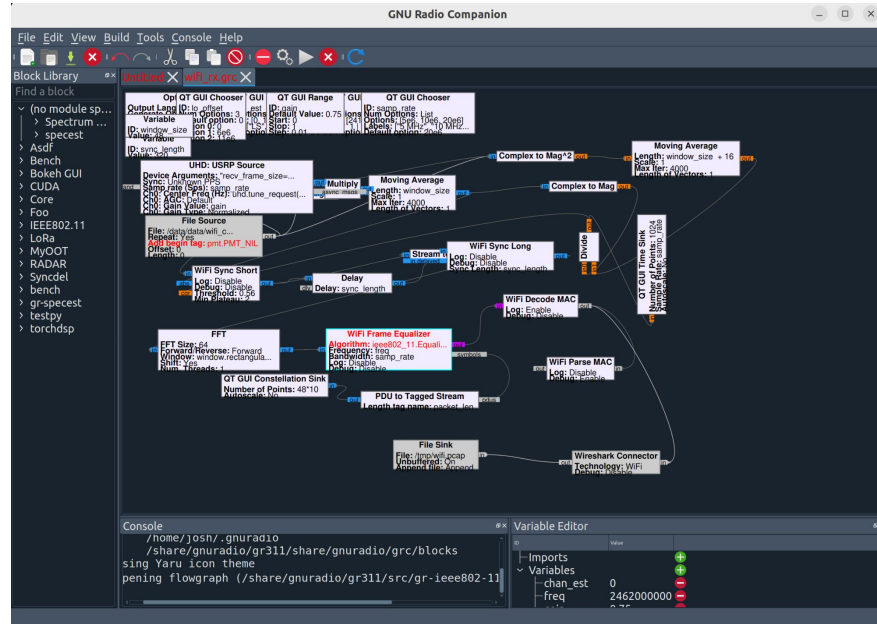


GNU Radio Companion (QT Version)

As of GNU Radio 3.10.10, QT version of GRC is available [not default]

gnuradio-companion --qt

- Clear current search query, filters, and sets
- 3 Open / 43 Closed
- Author Label Projects Milestones Reviews Assignees Sort
- 1 gr-qc: Better solution for #7382 [Bakapillai] [Closed] [2 of 3 tasks]
- gr-qc: Can only connect sink in port to source out port [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: set text color in some widgets depending on light/dark theme [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: Remove Theme and Icontheme settings [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: Use Themes [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: Fix another KeyError in variable_editor [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: AcceptSave mode for save dialog instead of AcceptOpen [Bakapillai] [Closed] [4 of 4 tasks]
- gr-qc: Add Variable Editor content menu to GRC Qt translations [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: Variable editor: fix several bugs [Bakapillai] [Closed] [4 of 4 tasks]
- gr-qc: make gcc use correct config paths [Bakapillai] [Closed] [4 of 4 tasks]
- gr-qc: test_gtbl fixes [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: Don't throw error when missing manifest dir and example paths [Bakapillai] [Closed] [4 of 4 tasks]
- gr-qc: Make more robust for missing manifest dir and example paths [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: Open grc files only once [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: Clear block search field on Ctrl+F [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: Don't close GRC if the user cancels the exiting action [Bakapillai] [Closed] [4 of 4 tasks]
- gr-qc: Save flowgraph: add extension to filename [Bakapillai] [Closed] [3 of 3 tasks]
- gr-qc: grc -gt filename grc doesn't open filename grc [Bakapillai] [Closed] [3 of 3 tasks]
- GRC: Allow for angular connections [Bakapillai] [Closed] [3 of 3 tasks]
- GRC: when passing arguments to terminal emulator, individually [Bakapillai] [Closed] [3 of 3 tasks]



Google Summer of Code



2 GSoC Students - out of *many* proposals

Kayla Comer - <https://kaylacomer.github.io/blog/>

- gr-fec - overhauling GNU Radio's forward error correction (FEC) / error coding package
- Mentors: Andrej Rode, Dani Estevez

Zaky Hermawan - <https://zakyhermawan.github.io//index.html>

- GRC: Standalone application and pluggable workflows
- Mentors: Sebastian Koslowski, Hakon Vagsether

Top GR3.X Technical Priorities



Top Technical Priorities

- SigMF Integration
- Improved Aesthetics of Graphical Widgets
- Native Installers for Windows / Mac
- Improved Documentation Infrastructure
- GRC-QT up to feature parity
- Integration tests using example flowgraphs
- Tracking OOT ecosystem (cgran.org improvements)

Some of these apply to both GR
3.x and GR 4.0

Project Vision

GNU Radio Project Vision

Strengthen

Spread

Sustain



Strengthen

- Make GNU Radio continually improving and feature-rich - from a tech perspective
- Maintain GNU Radio's position as the leading open-source SDR framework for rapid prototyping and real-time signal processing applications.
 - Make the framework better and more accessible
 - Make the framework more performant
 - Make the framework more stable and consistently maintained

WHY?

GNU Radio enables innovation in scientific applications



Spread

- Increase the footprint of GNU Radio in educational programs - University courses, Community College, High School
 - Prepare the next generation of signal processing engineers with the tools to develop real-time applications (as opposed to MATLAB/Python scripts)
 - Usability for beginners, installation, etc.
- Better marketing and outreach
- Training Opportunities

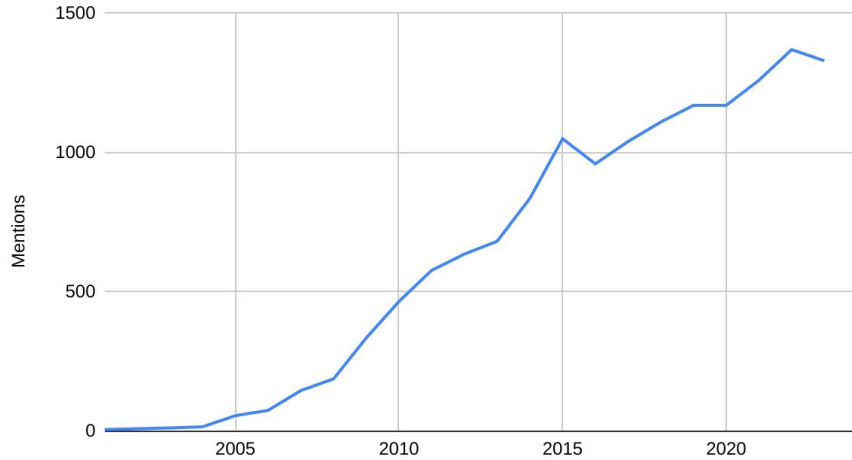
WHY?

Without shared framework, fractured ecosystem, non-portable solutions

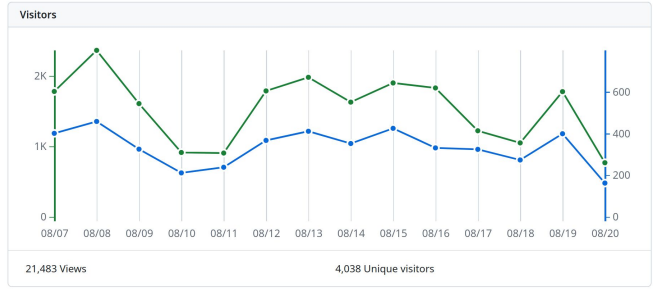


Hard to quantify the pervasiveness of GR Usage

Academic Papers Mentioning GNU Radio



Edit Pins Watch 234 Fork 1.9k Star 5k



@ 22 ☆ 4,975 ⚡ 9.4 <> C++

GNU Radio – the Free and Open Software Radio Ecosystem



ANACONDA.ORG Search Anaconda.org About Anaconda Hel

conda-forge / packages / gnuradio-grc 3.10.11.0

GNU Radio Companion graphical flowgraph interface
copied from cf-staging / gnuradio-grc

Conda Files Labels Badges

- License: GPL-3.0-or-later
- Home: <https://gnuradio.org/>
- Development: <https://github.com/gnuradio/gnuradio>
- Documentation: <https://gnuradio.org/doc/doxygen/>
- 1083209 total downloads
- Last upload: 1 day and 8 hours ago

Sustain



- Future-proof the leadership of the organization
 - Continually build leadership team - great opportunity for developing professionals
 - Move beyond all-volunteer
- Funding beyond GRCon
- Industry and Government partnerships
 - Partnership Program*
- Funding via external development efforts
 - Government funded efforts
 - Who can fund and execute large scale projects
 - E.g. DARPA SDR 4.0
 - Spirit of Public Money / Public Code
 - Actively engage with current and potential users of GNU Radio

WHY?

Maintaining a healthy ecosystem requires a cohesive and active organization able to make good decisions for the broader community

A Brief History of GR 4.0



The need to improve GNU Radio

People want to do even more with GNU Radio!

Moving beyond some of the limitations of GNU Radio has been a hot topic of discussion for a decade

Pain Points:

- Fixed scheduling - thread per block
- Packetized / Asynchronous data processing
- Effective use of Hardware Accelerators
- Ease of Use - boilerplate, duplication, etc.
- Heterogeneous, distributed platforms

Also balance with the reality that
one framework can't solve every
problem:
MODULARITY

2018-2019: Benchmarking and Untangling



<https://www.bastibl.net/bib/bloessl2019benchmarking/bloessl2019benchmarking.pdf>

<https://github.com/bastibl/gr-sched/tree/maint-3.9>

<https://www.bastibl.net/gnuradio-performance-1/>

Benchmarking and Profiling the GNU Radio Scheduler

Bastian Bloessl

Secure Mobile Networking Lab, TU Darmstadt, Mornwegstr. 32, 64293 Darmstadt, Germany

MAIL@BASTIBL.NET

Marcus Müller

Communications Engineering Lab, Karlsruhe Institute of Technology, Kreuzstr. 11, 76133 Karlsruhe, Germany

MUELLER@KIT.EDU

Matthias Hollick

Secure Mobile Networking Lab, TU Darmstadt, Mornwegstr. 32, 64293 Darmstadt, Germany

MHOLLLICK@SEEMOO.TU-DARMSTADT.DE

Abstract

From a technical perspective, GNU Radio has two main assets: its comprehensive block library of optimized, state-of-the-art signal processing algorithms and its runtime environment. The latter manages the data flow and turns GNU Radio in a real-time signal processing framework. In contrast to the block library, where it is easy to replace blocks with more efficient implementations, the runtime grew organically, which resulted in a complex system that is hard to maintain. At the same time, there are concerns about its performance. To understand the current implementation and explore opportunities for future improvements, we provide benchmarking and profiling results. We, furthermore, compare the performance of GNU Radio's default with a manually optimized configuration to show the potential of a more advanced scheduler.

normal PC. To meet the demands of new technologies, GNU Radio focused on optimizing individual transceiver components through more efficient algorithms (e.g., poly-phase filters) and implementations (e.g., vector-optimized math kernels (Rondeau et al., 2013a)). The runtime environment, i.e., the central component handling the data flow and managing parallel signal processing, did not see similar improvements. This is unfortunate, given the fact that it affects performance critical factors like CPU cache efficiency and overhead from thread synchronization.

The challenge of GPP-based SDRs is to provide a generic runtime environment that offers reasonable performance for any type of transceiver. The requirements are high throughput, low latency, and high stability to avoid occasional data loss. GNU Radio's current approach is to start each transceiver component (like filters or synchronizers) in a separate thread and leave scheduling to the operating system. While the success of GNU Radio shows the practicability of the approach, it also has inherent limitations:

Introducing: The Next Big Thing (NBT)

- ▶ “Scheduling” is actually pretty suboptimal
 - ▶ One thread per block: What if number of blocks \neq cores?
 - ▶ Scheduling is actually by the OS
 - ▶ no feedback of data flow into the scheduling at all
 - ▶ CPU core utilization \gg not thrashing caches
- ▶ Streams and Message are not equal
 - ▶ It's hard to impossible to do no-latency stream-produce-on-async-message blocks (ask Matt!)
 - ▶ Way to many states “I'm done”
 - ▶ can't just apply *work* to the content of a message (invented TSB for that, not an adequate design)





2020: Pre-FOSDEM Hackfest

Identified that the way to go is a fresh redesign

Led to “newsched” project → dev-4.0 branch of gnuradio

Largely same design (microservice) as GR3 with improvements in modularity and usability

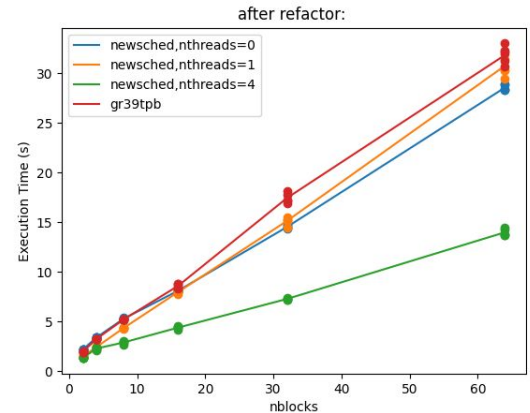
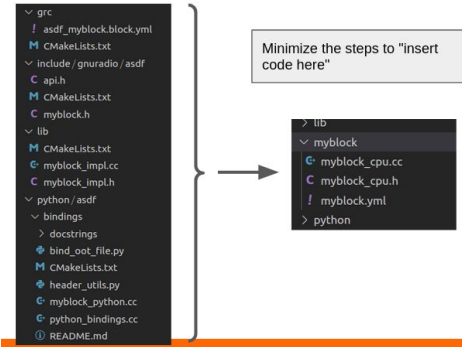
Modular scheduling with ability to set blocks per thread

Features demonstrated at GRCon22

The Future of GNU Radio: Heterogeneous Computing, Distributed Processing, and Scheduler-as-a-Plugin

Publisher: IEEE [Cite This](#) [PDF](#)

Joshua Morman ; Marc Lichtman ; Marcus Müller All Authors



2021: SDR 4.0

https://wiki.gnuradio.org/index.php/Custom_Buffers



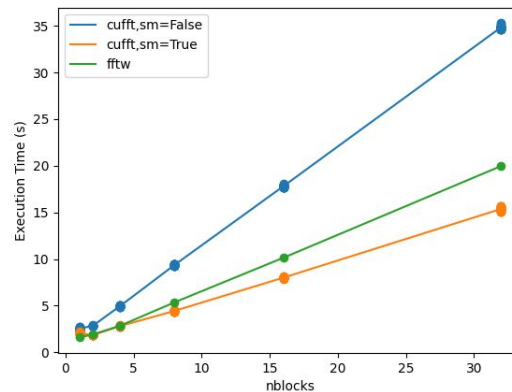
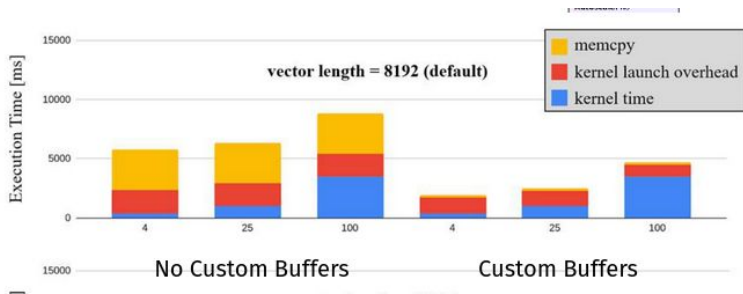
DARPA funded effort



Custom Buffers

Streamlined the data movement to hardware accelerators by use of single mapped buffers and device to device transactions

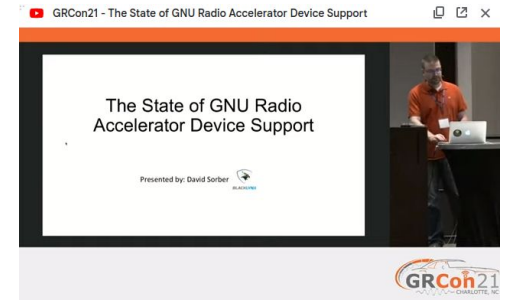
Upstreamed into GR 3.10



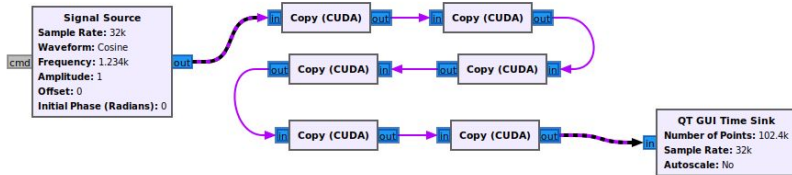


Custom Buffers

- 3.10 Feature introduced by **David Sorber** at **Black Lynx** via the **DARPA SDR 4.0** project
 - Final status given GRCon21
 - <https://www.youtube.com/watch?v=VO1zMXoweZg>
- Device compatible buffer structure (single mapped)
 - https://wiki.gnuradio.org/index.php/Custom_Buffers
- Data able to remain in accelerator memory
 - Streamlined data movement



Prior to 3.10 using custom buffers, each connection between CUDA enabled blocks would require ingress/egress to/from device memory (expensive)





2022 - : enter FAIR

Identified role of modern c++ in maximizing deployments on CPU

Move away from microservice architecture

Reflection, single point of source, ease of developing blocks

Block Merging

...





Our deepest gratitude for
your dedication to
pushing forward the state
of FOSS SDR



PUBLIC MONEY

PUBLIC CODE

Why is software created using taxpayers' money not released as Free Software?

We want legislation requiring that publicly financed software developed for the public sector be made publicly available under a [Free and Open Source Software](#) licence. If it is public money, it should be public code as well.

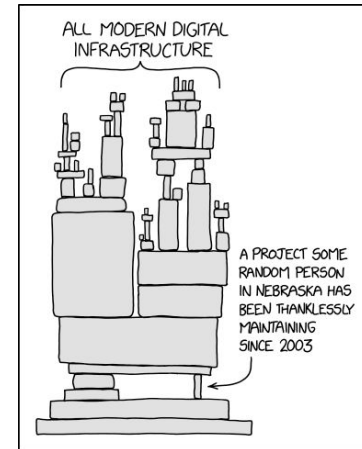
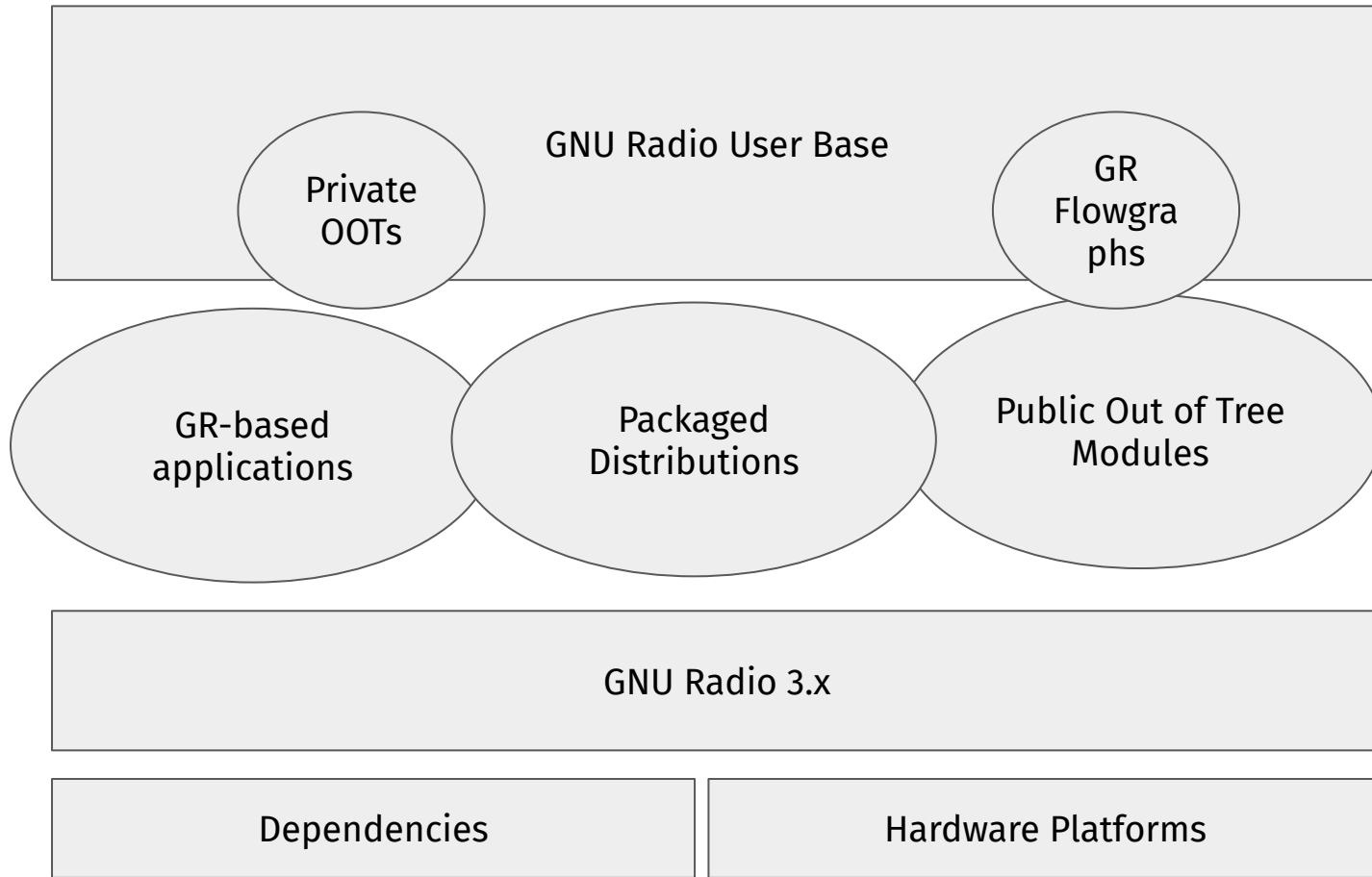
Code paid by the people should be available to the people!

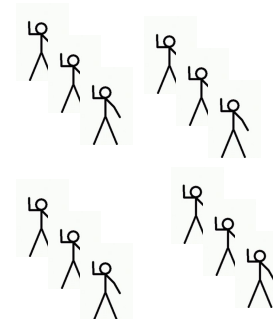
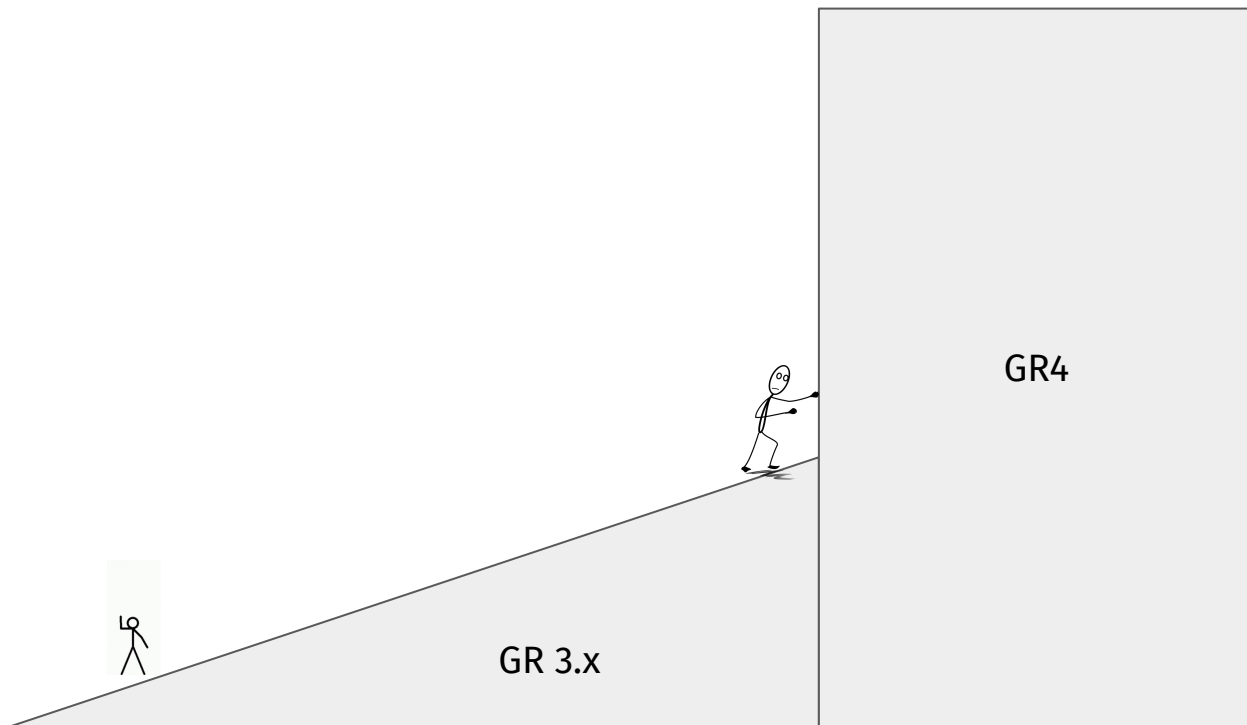
SUPPORTING ORGANISATIONS

The following organisations support our [open letter](#). If your organisation is also interested in joining the call for Public Code, please [contact us](#).



Path to GR 4.0 Adoption







What will it take to facilitate/accelerate adoption

- Benefits outweigh the transition cost
 - What are the most appealing benefits right now?
 - Benefits to one user are different than those to another
 - For some there may be no benefit
 - Low performance applications
 - RTL-SDR processing 100 kHz BW on Raspberry Pi
- Clear examples of benefits
 - Contrast performance between GR3 and GR4
 - Tangible examples > more abstract benchmarks
- New applications enabled by GR4
 - Some may be enabled by licensing changes



What will it take to facilitate/accelerate adoption

What is the effort involved in migration

- GRC-only Users
- OOT Developers
- GR Dependent Applications

Will OOT modules migrate?

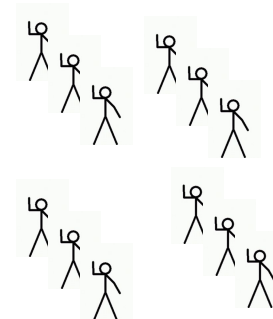
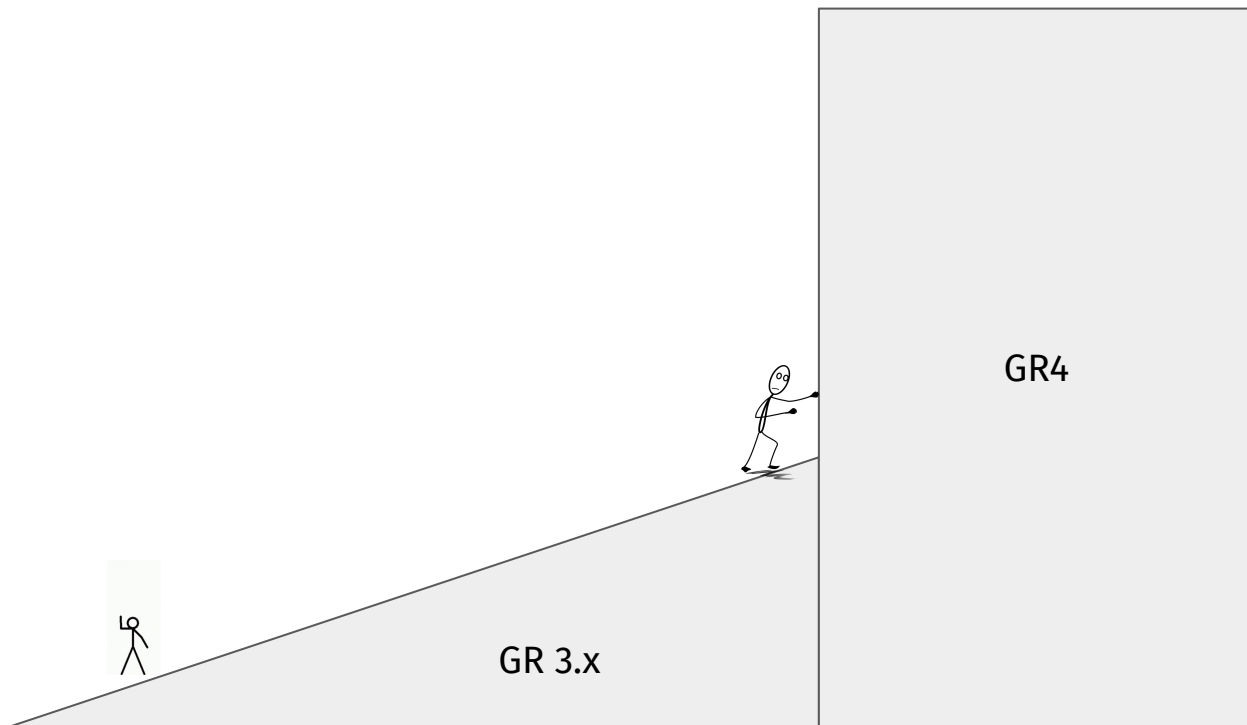
- Lost a number of OOTs in 3.7 ... 3.10 transition

Even a small effort could be a dealbreaker for some.



When can GR 4.0 become “GNU Radio.”

- To call something an official GR 4.0 release, it needs
 - Some Basic Feature Parity
 - Python bindings, GRC integration
 - Equivalent block set
 - Example applications
 - Migration guide
 - Active maintenance
- Can't/won't stop support of GR 3.x until ...
 - Enough users are comfortable migrating over
 - No one feels like maintaining GR 3.x (developer momentum)
 - Need maintenance help from user base





GRCCon24

Knoxville Tennessee



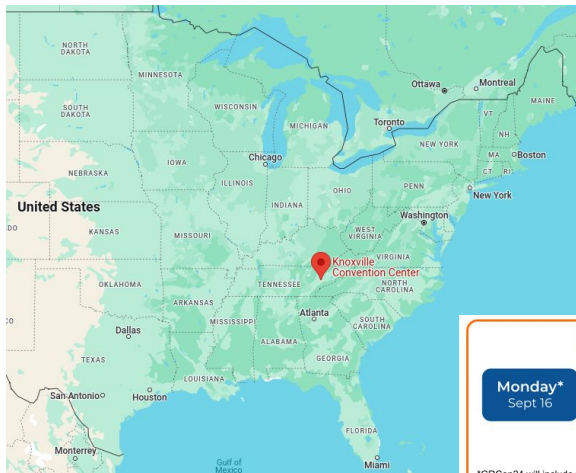
Sept. 16* - 20, 2024

Knoxville, TN
Knoxville Convention Center

gnuradio.org/grcon24



KNOXVILLE
CONVENTION CENTER



Main Track				
Monday* Sept 16	Tuesday Sept 17	Wednesday Sept 18	Thursday Sept 19	Friday Sept 20
		Meet & Greet	Dinner Social	Hands-on*

*GRCcon24 will include two half day (starting @ 1300 ET) hands-on segments on Monday Sept 16 and Friday Sept 20



How you can help:

Maintainers - URGENT NEED

Conference Planning and Execution

Outreach and Networking

Social Media

PR Reviewers

Block Development

Sponsor Engagement

Further Info

- **Main project info, blog, event info**
 - <https://gnuradio.org>
- **Documentation, Tutorials, project info**
 - <https://wiki.gnuradio.org>
- **Videos and slides from annual conference**
 - <https://youtube.com/gnuradioproject>
- **Mailing List – discuss-gnuradio**
- **Live discussion chat**
 - <https://chat.gnuradio.org>

Thanks and Questions?



@gnuradio

www.gnuradio.org