

---

# Introduction to 5G RAN PHY Simulators in OpenAirInterface



Francesco Mani and Laurent Thomas

31/05/2022

---

# Outline

- ◆ Physical simulators in OAI
- ◆ RFsimulator in OAI
- ◆ Softmodem modes in OAI
- ◆ Utilizing the scope in OAI
- ◆ OAI 5G RAN Roadmap

# Physical Simulators in OAI

- ◆ Physical simulators are a set of tools allowing to simulate mostly physical layer procedures, eg. channel coding and transport channels
  - ◆ Transport channel simulators: PBCHSIM, PRACHSIM, PUCCHSIM, DLSCHSIM, ULSCHSIM
  - ◆ Coding simulators: LDPCtest, polartest, smallblocktest
  - ◆ L1/L2 simulators: DLSIM, ULSIM
- ◆ Physical simulators can be used to test and debug features, evaluate performances and they are used in the CI to verify MRs

# Physical Simulators in OAI

- ◆ How to build physical simulators
  - ◆ `./build_oai --phy_simulators (--sanitize-address)`
  - ◆ `make nr_pbchsim` (from `ran_build/build`)
- ◆ Physical simulators are configurable via command line parameter
  - ◆ `-h` to show help listing the options for given simulator
- ◆ Among the configuration options: SNR and channel modeling
  - ◆ default channel option is AWGN
  - ◆ TODO: implementation of 3GPP TDL channel models to verify standard compliance of receiver performances

# Physical Simulators in OAI

- ◆ Let's run some PHY-SIM examples
  - ◆ pbchsim with and w/o initial sync
  - ◆ pucchsim format 0 and 2
  - ◆ dlsim with 256QAM option and 2 layer MIMO
  - ◆ ulsim with -P
  - ◆ polartest

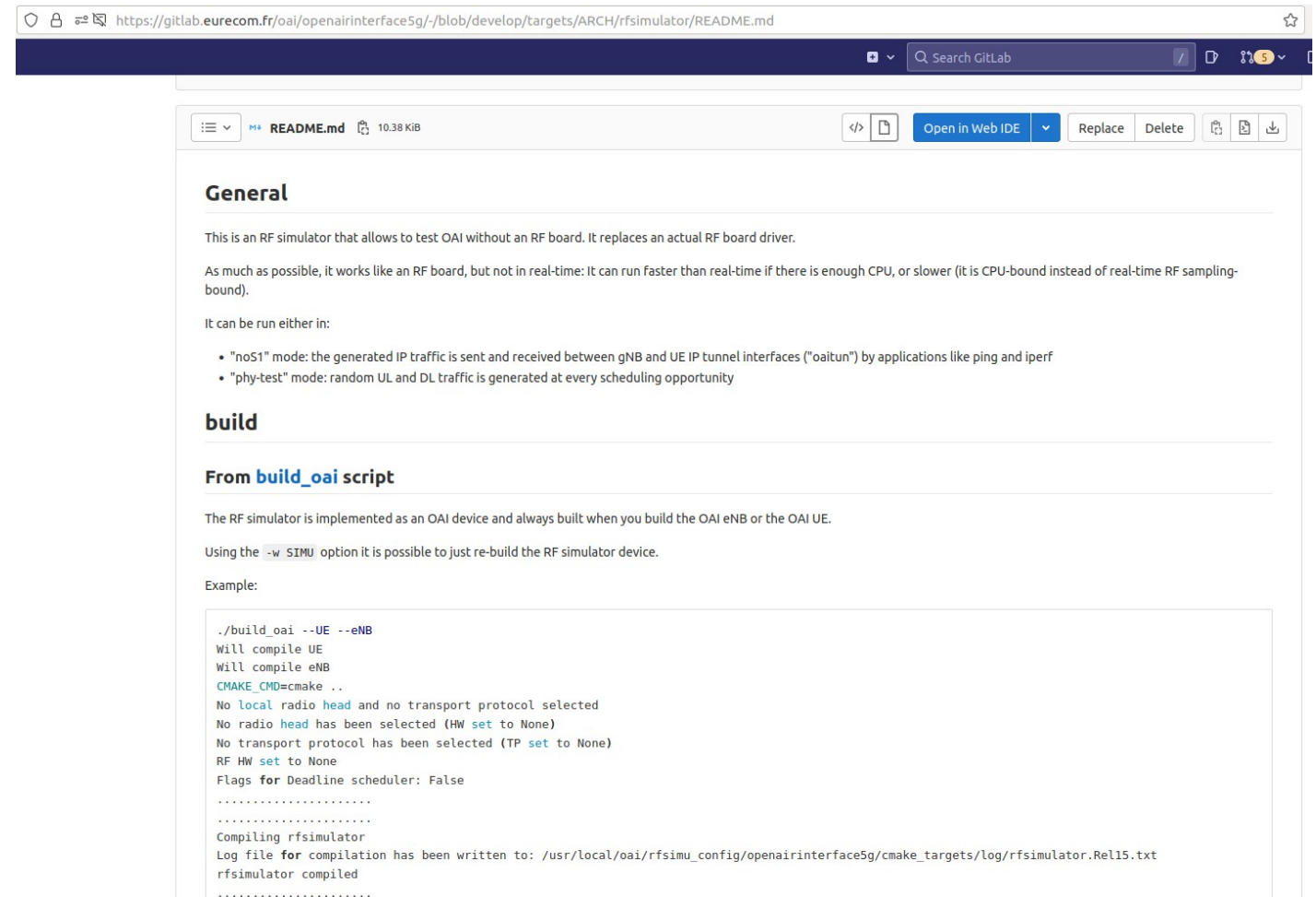
# RF Simulator in OAI

- ◆ OAI without RF board
  - ◆ Sends the I/Q samples in time domain (interface 8) between xNB and UEs
  - ◆ Transparent transmission and channel simulation are available
- ◆ All OAI options and modes are available with RFsim, as with actual RF boards
  - ◆ Important note: operations are not in real time (e.g. it is not possible to test real time throughput in a meaningful way)
  - ◆ Debug is possible: any break point works in gdb
- ◆ Exceptions: OAI UEs are not (yet) thread safe, so it need to run with multi-threading restrictions

# RF Simulator in OAI

## ◆ Documentation

- Module specific documentation is in module source code directory
- In targets/ARCH/rfsimulator
- As the codebase, documentation can be improved and corrected via MRs



The screenshot shows a web browser displaying the README for the RF Simulator in OAI. The URL is <https://gitlab.eurecom.fr/openairinterface5g/-/blob/develop/targets/ARCH/rfsimulator/README.md>. The page title is "README.md" and it is 10.38 KiB in size. The content is organized into sections: "General", "build", and "From build\_oai script".

**General**

This is an RF simulator that allows to test OAI without an RF board. It replaces an actual RF board driver. As much as possible, it works like an RF board, but not in real-time: It can run faster than real-time if there is enough CPU, or slower (It is CPU-bound instead of real-time RF sampling-bound).

It can be run either in:

- "noS1" mode: the generated IP traffic is sent and received between gNB and UE IP tunnel interfaces ("oaitun") by applications like ping and iperf
- "phy-test" mode: random UL and DL traffic is generated at every scheduling opportunity

**build**

**From build\_oai script**

The RF simulator is implemented as an OAI device and always built when you build the OAI eNB or the OAI UE. Using the `-w SIMU` option it is possible to just re-build the RF simulator device.

Example:

```
./build_oai --UE --eNB
Will compile UE
Will compile eNB
CMAKE_CMD=cmake ..
No local radio head and no transport protocol selected
No radio head has been selected (HW set to None)
No transport protocol has been selected (TP set to None)
RF HW set to None
Flags for Deadline scheduler: False
.....
.....
Compiling rfsimulator
Log file for compilation has been written to: /usr/local/oai/rfsimu_config/openairinterface5g/cmake_targets/log/rfsimulator.Rel15.txt
rfsimulator compiled
.....
```

# RF Simulator in OAI

- ◆ Command line option: --rfsim
- ◆ Rfsim options in config file

```
rfsimulator :{  
serveraddr = "server";  
serverport = "4043";  
options = (); #("saviq"); or/and "chanmod"  
modelname = "AWGN";  
IQfile = "/tmp/rfsimulator.iqs";  
};
```

Either server (tcp listen)  
Or a server IP addr

TCP port to use on  
server side

Saviq to record  
transmitted I/Q in a file  
Chanmod enables the  
channel simulation

Channel modelling  
Offers several options

A UE replay node exists for saviq option  
It can use field trial captures or recorded simulation



# Softmodem Modes in OAI

- ◆ **phy-test mode** is an OAI softmodem option that allows the gNB to continuously schedule DL-SCH and UL-SCH even without a UE
- ◆ This mode can be activated both for RFsimulator and with actual RF boards
- ◆ A use case is to test transport channels with a 5G capable signal analyzer
- ◆ Via command line parameter it is possible to configure the slots to be scheduled, the allocated resources, MCS etc.

# Softmodem Modes in OAI

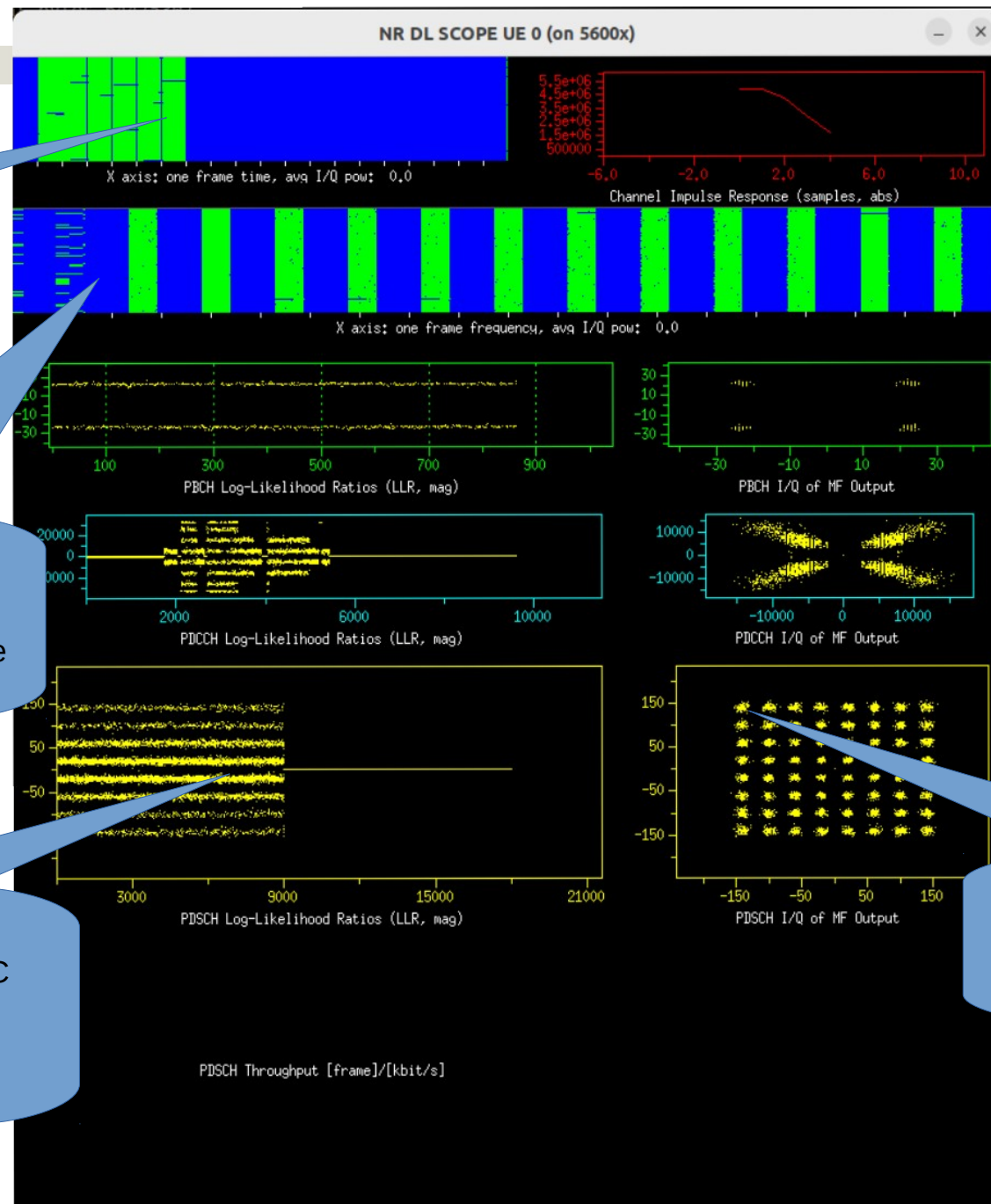
- ◆ **do-ra mode** is an OAI softmodem option that allows a simulated NSA connection with only 5G terminals being present
- ◆ Also this mode can be used both with RFsimulator and actual RF boards
- ◆ It performs synchronization and contention-free random access between OAI gNB and OAI UE before starting to schedule DL and UL traffic channels
- ◆ RRC configuration of UE is done via a .raw file generated by gNB and saved in execution folder

# Softmodem Modes in OAI

- ◆ Now some practical examples of modes using RF simulator
  - ◆ phy-test mode selecting MCS, RBs and slot occupation
  - ◆ do-ra mode for FR1
  - ◆ do-ra mode for FR2
  - ◆ SA (without core network)

# Utilizing the Scope in OAI

- ◆ The scope is a graphical interface to display transport channels
- ◆ How to build the shared libs
  - ◆ `ninja/make enbscope uescope nrscope` (one shared lib for NR UE and gNB)
  - ◆ Or `./build_oai ... --build_libs all`
- ◆ Runtime option « `-d` » on command line

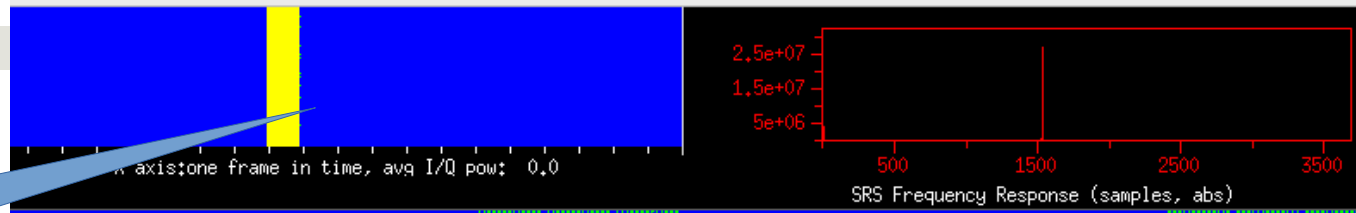


Time domain  
One line is a frame  
Color is the energy relative to average

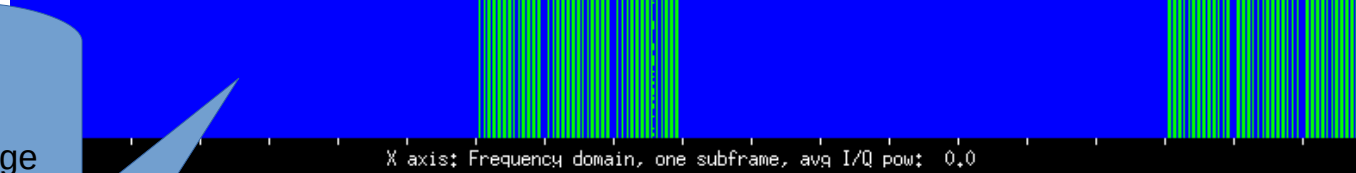
Frequency domain  
One line is a frame  
Color is the energy relative to average

Llr graphs  
The soft symbols that will enter the FEC decoder  
2 lines if 1 bit per symbol

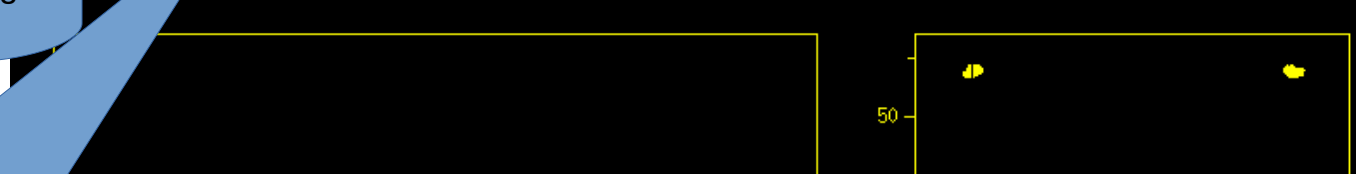
Constellation graphs  
The signal after equalization per physical channel



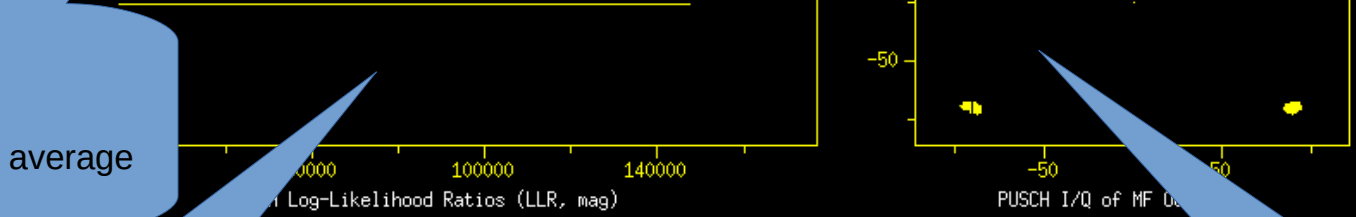
Time domain  
One line is a frame  
Color is the energy relative to average



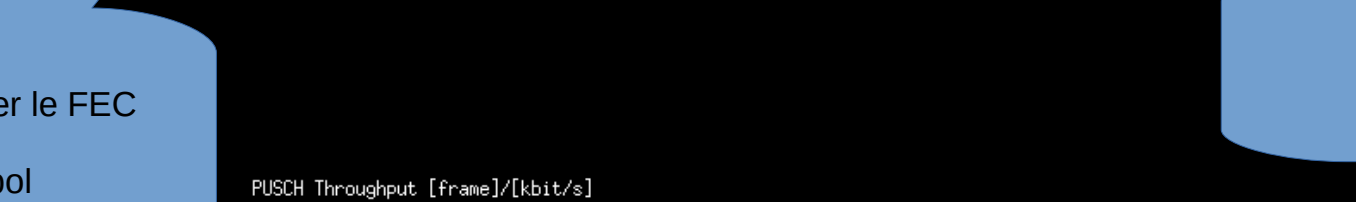
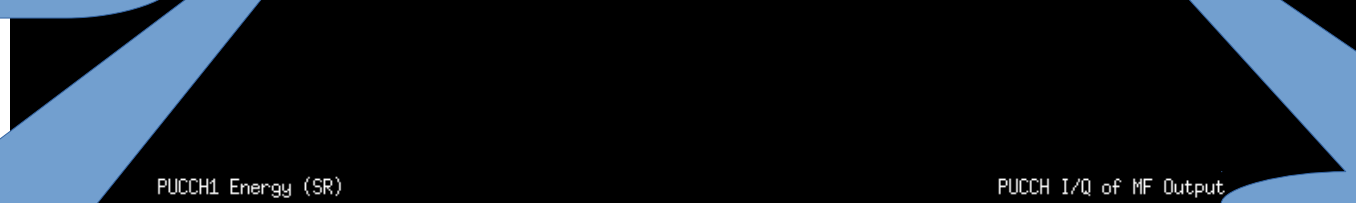
Frequency domain  
One line is a frame  
Color is the energy relative to average



Llr graphs  
The soft symbols that will enter the FEC decoder  
2 lines if 1 bit per symbol



Constellation graphs  
The signal after equalization per physical channel



# Utilizing the Scope in OAI

- ◆ Configuration file need default channel definition
- ◆ Add in your configuration file the prepared default file
  - @include "channelmod\_rfsimu.conf"
  - Example file for UE
- ◆ Launch the gNB and the UE
  - sudo ./nr-softmodem  
-O ../../ci-scripts/conf\_files/gnb.band78.tm1.106PRB.usrpn300.conf  
--phy-test --rfsim -d
  - -m for DLSCCH MCS (27), -D downlink slots bit map (126)
  - sudo ./nr-uesoftmodem --phy-test --nokrnmod -O ./ue.conf  
--rfsim -d --telnetsrv --rfsimulator.options chanmod

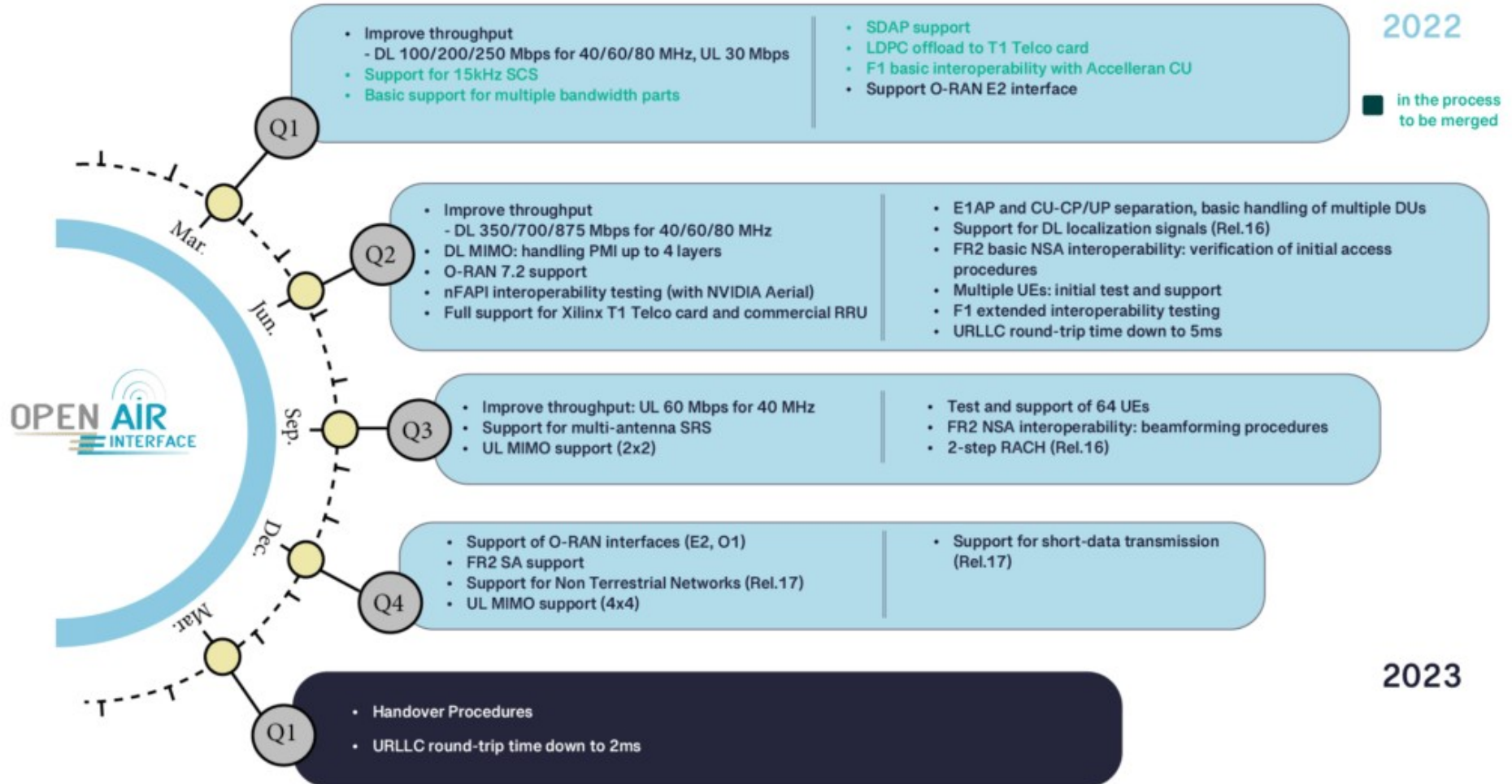
```
uicc0 = {  
  imsi = "001010000000002";  
  key = "6874736969202073796d4b2079650a73";  
  opc = "504f20634f6320504f50206363500a4f";  
  dnn = "oai";  
  nssai_sst=1;  
  nssai_sd=1;  
}  
  
log_config :  
{  
  global_log_level      = "info";  
  hw_log_level          = "info";  
  phy_log_level         = "info";  
  mac_log_level         = "info";  
  mac_log_verbosity    = "high";  
  rlc_log_level         = "info";  
  pdcp_log_level        = "info";  
  rrc_log_level         = "info";  
  nr_rrc_log_level      = "info";  
  ngap_log_level        = "info";  
};  
@include "../../ci-scripts/conf_files/channelmod_rfsimu.conf"
```

# Utilizing the Scope in OAI

- ◆ Connect to the telnet server in the UE
  - Telnet 127.0.0.1 9090
  - Help
    - module 4 = channelmod:
      - channelmod help
      - channelmod show <predef,current>
      - channelmod modify <channelid> <param> <value>
    - module 5 = rfsimu:
      - rfsimu setmodel <model name> <model type>
    - If you don't get the help on channemod, you missed to enable it in rfsimulator
      - Help is contextual
  - Command
    - channelmod modify 0 noise\_power\_dB -50



# OAI 5G RAN Roadmap



# OAI 5G RAN Roadmap

The screenshot shows a web browser window displaying a GitLab wiki page. The browser's address bar shows the URL: <https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/OpenAirDevMeetings>. The page header includes the EURECOM logo and a search bar. The left sidebar contains a navigation menu for the 'openairinterface5G' project, with 'Wiki' selected. The main content area shows the title 'OpenAirDevMeetings', the last editor 'Camille Lerda' from 2 weeks ago, and buttons for 'Page history' and 'New page'. The text of the page describes weekly OAI Developer Calls and provides specific times for odd and even weeks, along with a note about the transition from Webex to Zoom.

OpenAirDevMeetings · W X +

← → ↻ 🔒 🔑 🔍 <https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/OpenAirDevMeetings>

Menu +

**openairinterface5G**

- Project information
- Repository
- Issues 54
- Merge requests 40
- CI/CD
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Analytics
- Wiki**

oai > openairinterface5G > Wiki > **OpenAirDevMeetings**

Last edited by **Camille Lerda** 2 weeks ago Page history New page

## OpenAirDevMeetings

We are holding a weekly OAI Developer Calls open to all OAI community members.

### General developer calls

**Odd weeks:** [Thursday 11:00 CET \(for Asia\)](#)

**Even weeks:** [Thursday 16:00 CET \(for Americas\)](#)

Previously these meetings were managed via the Webex platform. In order to harmonize with the other OAI regular meetings, kindly note that starting on the 2022/05/12 the Developer Calls will be held on Zoom.

The new calendar invitations can be downloaded with the links above.