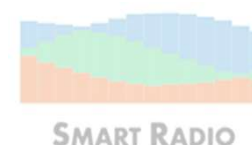


主催：東京都立産業技術研究センター(TIRI)

共催：SR研究会 学習型チュートリアル



インターネット無し
USBインストール版

学習型チュートリアル 第6回

～OpenAirInterface 5G-RAN構築 (B210)～



IDAQS

株式会社 アイダックス

戸部英彦 野上昌雄

目次

1. SIMカード
2. OAIの手順
3. Ubuntu24.04インストール
4. package インストール
5. OAI CN5G インストール
6. gNB インストール
7. 5GCとgNBの起動

1. SIMカード

Program UICC/SIM Card with Open Cells Project programming tool uicc-v3.3.

```
$ cd ~/B210_Tutorial
```

```
# Extract uicc-v3.3.tgz
```

```
$ cd uicc
```

```
# SIMカードをライターに入れて下さい。下記のlsusbで見えるか確認下さい。
```

```
$ lsusb
```

```
Bus 001 Device 006: ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port / Mobile Phone Data Cable
```

```
# あなたのお好きなお名前をiDAQSの代わりに入力下さい。
```

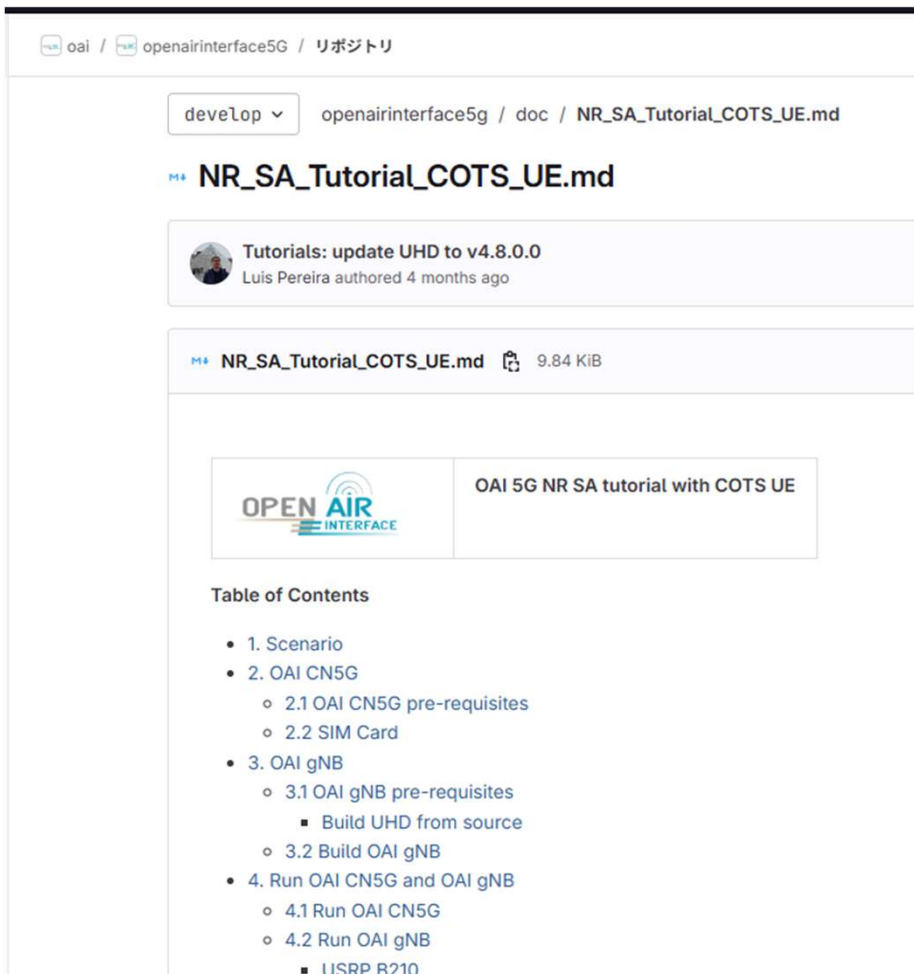
```
$ sudo ./program_uicc --adm 12345678 --imsi 001010000000001 --isdn 00000001 --acc 0001 --key
```

```
fec86ba6eb707ed08905757b1bb44b8f --opc C42449363BBAD02B66D16BC975D77CC1 --spn "iDAQS"
```

```
--authenticate
```



2. OAIの手順



oai / openairinterface5g / リポジトリ

develop ▾ openairinterface5g / doc / NR_SA_Tutorial_COTS_UE.md

NR_SA_Tutorial_COTS_UE.md

Tutorials: update UHD to v4.8.0.0
Luis Pereira authored 4 months ago

NR_SA_Tutorial_COTS_UE.md 9.84 KiB

OPEN AIR INTERFACE OAI 5G NR SA tutorial with COTS UE

Table of Contents

- 1. Scenario
- 2. OAI CN5G
 - 2.1 OAI CN5G pre-requisites
 - 2.2 SIM Card
- 3. OAI gNB
 - 3.1 OAI gNB pre-requisites
 - Build UHD from source
 - 3.2 Build OAI gNB
- 4. Run OAI CN5G and OAI gNB
 - 4.1 Run OAI CN5G
 - 4.2 Run OAI gNB
 - USRP B210

基本的に下記のgitlabの解説の作業ですが、
そのままでは動作しないため
この手順でインストール下さい。

https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/NR_SA_Tutorial_COTS_UE.md

1. Scenario

- Minimum hardware requirements:
Laptop/Desktop/Server for OAI CN5G and OAI gNBOperating
System: Ubuntu 24.04 LTS
CPU: 8 cores x86_64 @ 3.5 GHzRAM: 32 GB
- Laptop for UE
Windows driver for Quectel MUST be equal or
higher than version 2.4.6 => 2.8
- USRP B210
- Quectel RM500Q => RM520 技適あり ローカル5Gで使える??

3. Ubuntu24.04インストール

1. **重要！バックアップ下さい！** Windows11 NMVe-SSD, Bit-Locker code and licenses.

2. Download Ubuntu24.04 iso file and create Installer USB Memory.

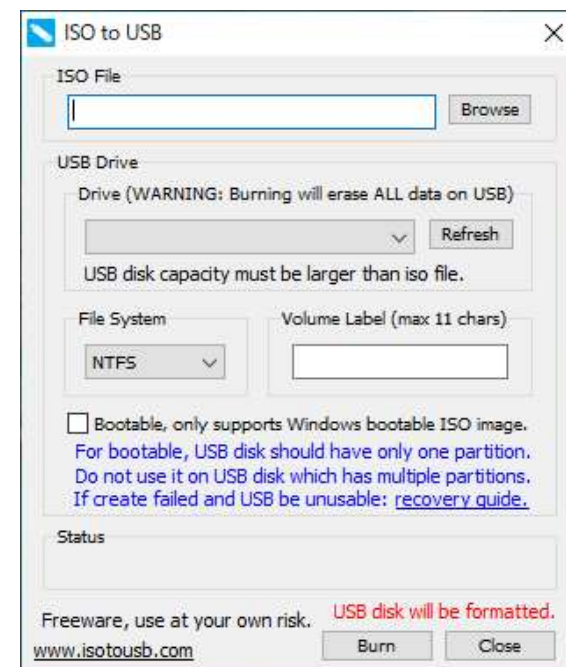
3. Insert "Ubuntu24.04 Installer USB Memory" and "Boot-SSD" to your PC.

4. Boot from the USB Memory. For example, push F12 in the case of DELL-PC.

- Choose your language: English
- Accessible in Ubuntu: No Change > Next
- Select your keyboard layout: Japanese > Next
- Connect to the internet: Check "Use wired connection" > Next
- Update now > Skip
- What do you want to do with Ubuntu? Check Install Ubuntu > Next
- How would you like to install Ubuntu? Check Interactive installation > Next
- What apps would you like to start with? Check Default selection > Next
- Install recommended proprietary software? No Check > Next
- How do you want to install Ubuntu?
 - Check "Erase disk and install Ubuntu" > Next
- Choose where to install Ubuntu? Select drive: sda-256GB!!!
NEVER SELECT Win11 NVMe!! > Next
- Create your account, oai,oai,oai,
 - Remove check "Require my password to login" > Next
- Select your time zone: Tokyo > Next – Next

5. Settings Power Mode:Performance, Dim Screen:OFF, Screen Blank: Never

Ubuntuのダウンロード



4. package インストール

ネットに繋がらない苦肉の策！他により良い方法があるかも？バージョンが固定される。

```
$ cd ~
# Copy and Extract B210_Tutorial.zip
$ cd /B210_Tutorial

# Extract packages.zip
$ cd packages
$ sudo dpkg -i *.deb
.....
libSDL1.2debian:amd64 libSDL1.2-dev:amd64
Processing was halted because there were too many errors.
$

# Stop install and try again! independency has changed!
$ sudo dpkg -i *.deb
.....
xorg
libglapi-mesa:amd64
$
# FINISH without ERRORS!
```

5. OAI CN5G インストール(1/2)

```
$ cd ~
# Copy and Extract B210_Tutorial.zip
$ cd /B210_Tutorial
# Extract docker.zip
$ cd docker
$ sudo install -m 0755 -d /etc/apt/keyrings
$ sudo cp docker.asc /etc/apt/keyrings/$ sudo chmod a+r /etc/apt/keyrings/docker.asc
$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | sudo
tee /etc/apt/sources.list.d/docker.list > /dev/null

# Extract oai-cn5g.zip
$ cd ~/B210_Tutorial/oai-cn5g/
$ mv openairinterface5g-develop-doc-tutorial_resources-oai-cn5g/doc/tutorial_resources/oai-
cn5g ~/oai-cn5g
$ cd ..
$ rm -r oai-cn5g
$ cd ~/oai-cn5g
$ mv ~/B210_Tutorial/docker/*.tar .
$ sudo docker images => NO IMAGES
```

5. OAI CN5G インストール(2/2)

Load Network Functions(NFs) NFとは？

\$ cd ~/oai-cn5g

\$ sudo docker image load -i ims.tar

.....

\$ sudo docker images => See 10 NFs.

\$ sudo docker compose up -d => Run NFs.

\$ sudo docker logs oai-amf -f => See log!

\$ sudo docker compose stop

\$ sudo docker compose down

\$ sudo docker ps -a => NO docker!!

```
oai@oai-Dell-G15-5530: ~/oai-cn5g
[2025-09-12 08:43:17.473] [amf_sbi] [info] Send HTTP message to http://oai-nrf:8080/nrf-nfm/v1/nf-instances/a0e6a22f-01e9-402c-be29-5d2c7f2aadb3
[2025-09-12 08:43:17.473] [amf_sbi] [info] HTTP message Body: [{"op":"replace","path":"/nfStatus","value":"REGISTERED"}]
[2025-09-12 08:43:17.473] [amf_sbi] [debug] Send a simple HTTP request
[2025-09-12 08:43:17.474] [amf_sbi] [info] Get response with HTTP code (204)
[2025-09-12 08:43:17.474] [amf_sbi] [info] Could not get JSON content from the response
[2025-09-12 08:43:17.474] [amf_app] [debug] Received SBI_UPDATE_NF_INSTANCE_RESPONSE
[2025-09-12 08:43:17.474] [amf_app] [debug] Handle NF Update response
[2025-09-12 08:43:17.474] [amf_app] [debug] Set a timer to the next Heart-beat (10)
[2025-09-12 08:43:27.275] [amf_app] [info]

-----gNBs' Information-----
| Index | Status | Global Id | gNB Name | PLMN |
|-----|-----|-----|-----|-----|
| - | - | - | - | - |

-----UEs' Information-----
| Index | 5GMM State | IMSI | GUTI | RAN UE NGAP ID | AMF UE NGAP ID | PLMN | Cell Id |
|-----|-----|-----|-----|-----|-----|-----|-----|
| - | - | - | - | - | - | - | - |

[2025-09-12 08:43:27.474] [amf_app] [debug] Send ITTI msg to SBI task to trigger NRF Heartbeat
[2025-09-12 08:43:27.474] [amf_sbi] [info] Receive Update NF Instance Request, handling ...
[2025-09-12 08:43:27.474] [amf_sbi] [debug] Send NF Update to NRF
[2025-09-12 08:43:27.474] [amf_sbi] [info] Send HTTP message to http://oai-nrf:8080/nrf-nfm/v1/nf-instances/a0e6a22f-01e9-402c-be29-5d2c7f2aadb3
[2025-09-12 08:43:27.474] [amf_sbi] [info] HTTP message Body: [{"op":"replace","path":"/nfStatus","value":"REGISTERED"}]
[2025-09-12 08:43:27.474] [amf_sbi] [debug] Send a simple HTTP request
[2025-09-12 08:43:27.477] [amf_sbi] [info] Get response with HTTP code (204)
[2025-09-12 08:43:27.477] [amf_sbi] [info] Could not get JSON content from the response
[2025-09-12 08:43:27.477] [amf_app] [debug] Received SBI_UPDATE_NF_INSTANCE_RESPONSE
[2025-09-12 08:43:27.477] [amf_app] [debug] Handle NF Update response
[2025-09-12 08:43:27.477] [amf_app] [debug] Set a timer to the next Heart-beat (10)
oai@oai-Dell-G15-5530:~/oai-cn5g$
```

LogにgNB,UEの状態が判る！

6. gNB インストール(1/3): ライブラリ入替

スコープのためにライブラリを入れ替えます。スコープが古いため。

3) Copy asn1c

```
$ cd ~/B210_Tutorial
```

```
# Copy asn1c.zip and extract it.
```

```
$ sudo cp -rv asn1c /opt
```

4) Copy simde

```
# Copy simde.tgz and extract it.
```

```
$ sudo cp -rv simde /usr/include
```

5) Install cmake

```
# Copy cmake-3.22.6-linux-x86_64.tar.gz
```

```
$ sudo mv cmake-3.22.6-linux-x86_64 /opt/cmake-3.22
```

```
$ export PATH="/opt/cmake-3.22/bin:$PATH"
```

```
$ echo 'export PATH="/opt/cmake-3.22/bin:$PATH"' >> ~/.bashrc
```

```
# Check cmake version
```

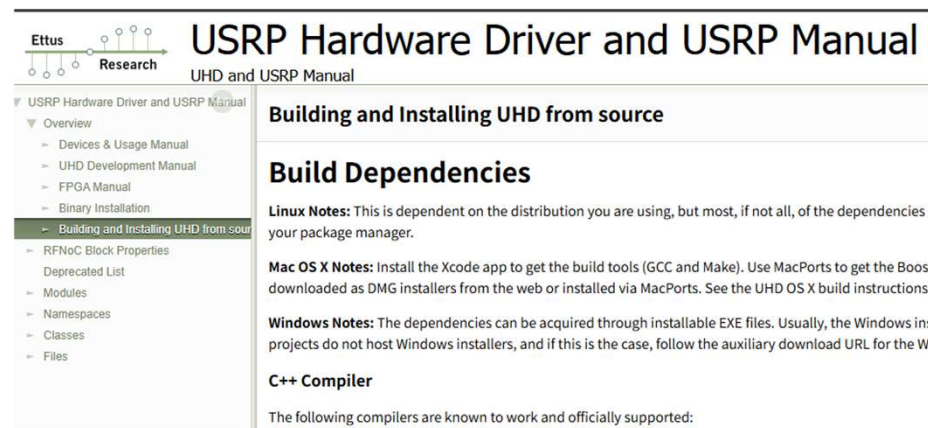
```
$ cmake --version
```

```
cmake version 3.22.6CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

6. gNB インストール(2/3): UHD

6) Install UHD

```
# Extract uhd.zip
$ cd ~/B210_Tutorial/uhd/host
$ mkdir build
$ cd build
$ cmake ../
$ make -j $(nproc)
# IT TAKES 20 min, long time!!
$ make test
$ sudo make install
$ sudo ldconfig$ export LD_LIBRARY_PATH=/usr/local/lib
$ lsusbBus 001 Device 006: ID 2500:0020 Ettus Research LLC USRP B210
## Copy /usr/local/share/uhd/images from USB Memory and unzip.
$ sudo mv images /usr/local/share/uhd/images
$ sudo uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 13.3.0; Boost_108300; UHD_4.8.0.HEAD-0-g308126a4
[INFO] [B200] Loading firmware image: /usr/local/share/uhd/images/usrp_b200_fw.hex...
-----
-- UHD Device 0
-----
Device Address:
  serial: 32711B7
  name: MyB210
  product: B210
  type: b200
```



USRP Hardware Driver and USRP Manual
UHD and USRP Manual

Building and Installing UHD from source

Build Dependencies

Linux Notes: This is dependent on the distribution you are using, but most, if not all, of the dependencies : your package manager.

Mac OS X Notes: Install the Xcode app to get the build tools (GCC and Make). Use MacPorts to get the Boost downloaded as DMG installers from the web or installed via MacPorts. See the UHD OS X build instructions

Windows Notes: The dependencies can be acquired through installable EXE files. Usually, the Windows ins projects do not host Windows installers, and if this is the case, follow the auxiliary download URL for the Wi

C++ Compiler

The following compilers are known to work and officially supported:

6. gNB インストール(3/3): gNB

8) Build gNB

```
$ cd ~/B210_Tutrial/  
# Copy openairinterface5g.zip and extract it.  
$ mv openairinterface5g/ ~/openairinterface5g/  
$ cd ~/openairinterface5g/  
$ git checkout develop
```

・ OAIのバージョンチェックを省略。このためHASH値を合わせています。USBのライブラリが一致しない。

```
#Copy CPM.cmake before build_oai executes
```

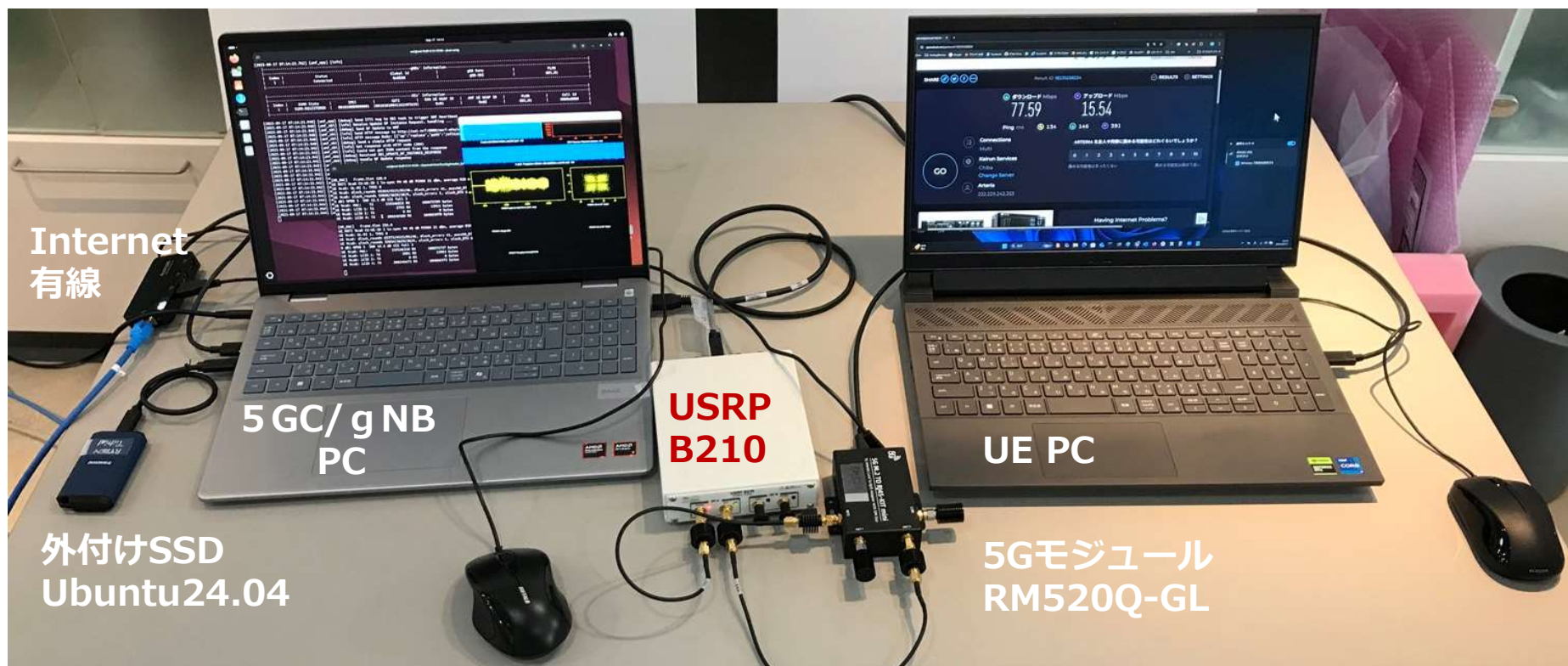
```
$ cd ~/B210_Tutrial/  
$ cp ~/openairinterface5g/cmake_targets/CPM.cmake ~/openairinterface5g/cmake_targets/CPM.cmake.bak  
$ cp CPM.cmake ~/openairinterface5g/cmake_targets/  
$ cd ~/openairinterface5g/cmake_targets
```

ビルドの実行

```
$ ./build_oai -w USRP --ninja --gNB --nrUE --build-lib "nrscope uescope" -C
```

```
.....  
[10854/10854] Linking CXX executable nr-softmodem  
BUILD SHOULD BE SUCCESSFUL  
$
```

7. 5GCとgNBの起動(1/3):接続



B210 RF A
Tx/Rx:ATT30dB →Cable→
Rx: ATT:30dB ←Cable←
USBバスパワーのみ

Quectel
ATT30dB: ANT0
ATT30dB: ANT2
終端: ANT1
終端: ANT3

外部電源

7. 5GCとgNBの起動(2/3): 画面例

9) Run 5GC and gNB

Run 5GC

```
$ cd ~/oai-cn5g$ sudo docker compose up -d
```

```
$ sudo docker logs oai-amf -f
```

Down 5GC

```
$ ctrl + C
```

```
$ sudo docker compose down
```

Run gNB

USRP B210

```
$ cd ~/openairinterface5g/cmake_targets/ran_build/build
```

```
$ sudo ./nr-softmodem -O ../../targets/PROJECTS/GENERIC-NR-
```

```
5GC/CONF/gnb.sa.band78.fr1.106PRB.usrbp210.conf -E --continuous-tx -d
```

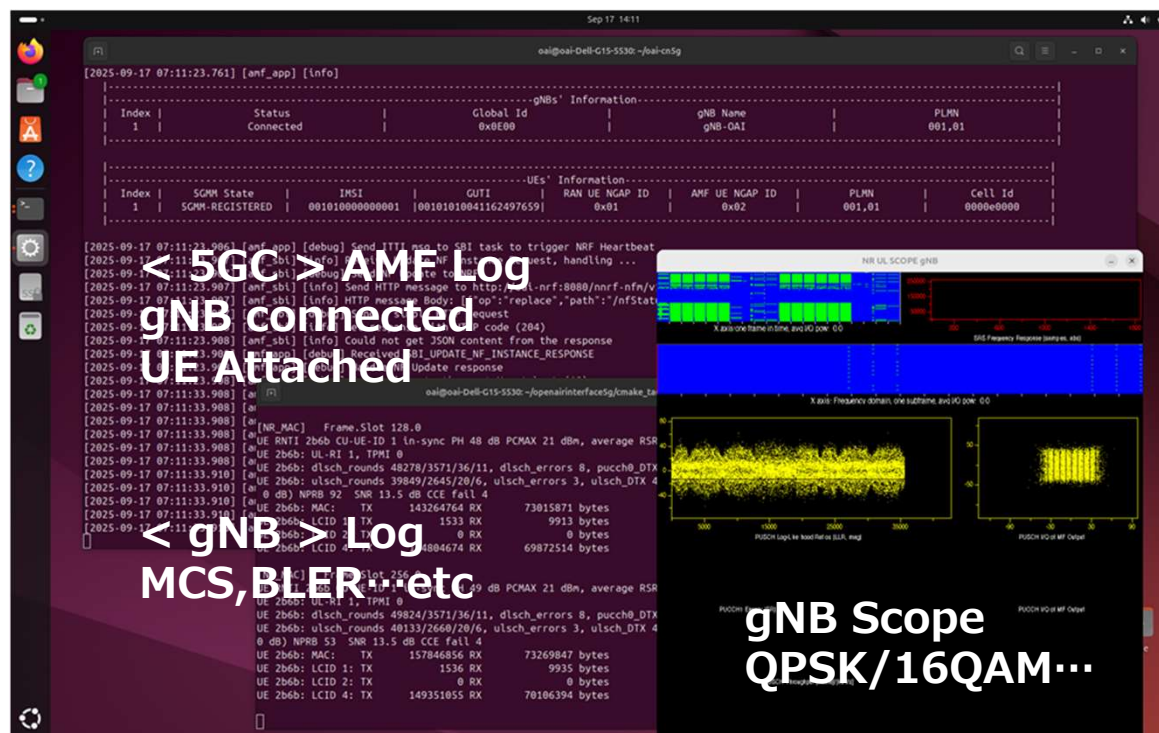
See Scope!!

Connect Internet

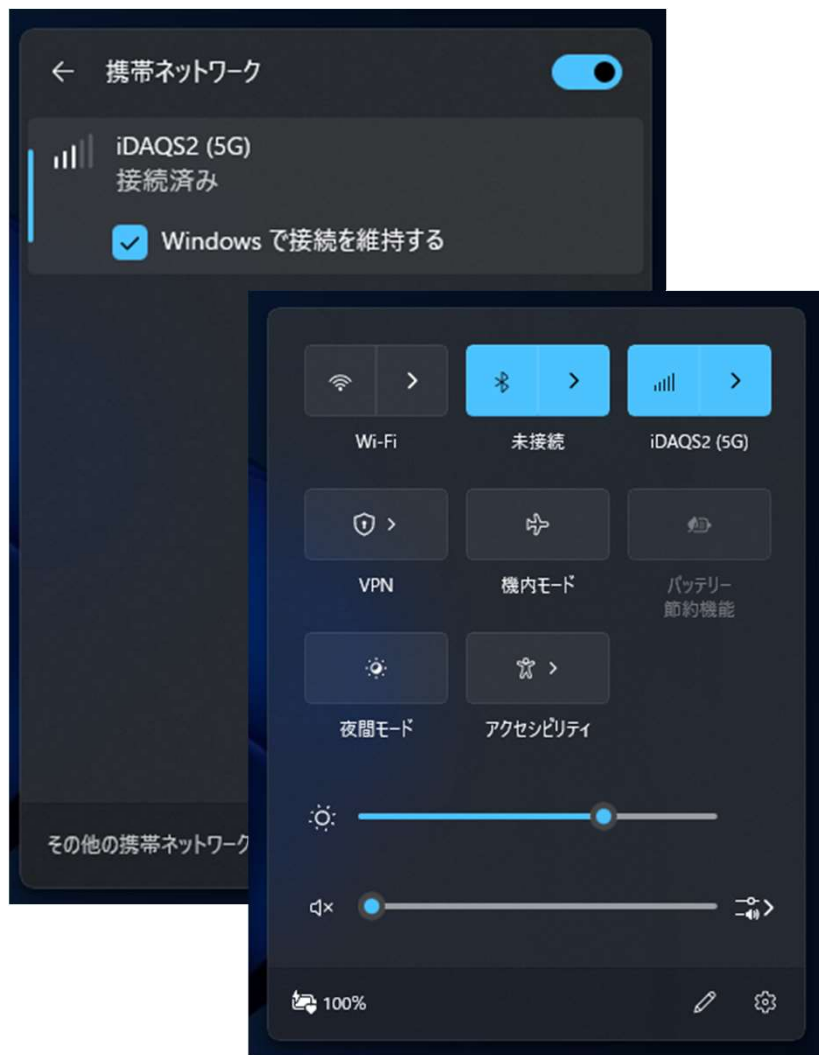
UE Start!

confファイル -d(スコープ表示)

5C-SA,Band78,FR1(SCS30kHz),106PRB(40MHz帯域)



7. 5GCとgNBの起動(3/3) : UE疎通



SIMカードに書き込んだSPNの名前が見える。
DL: 77Mbps, UL: 15Mbps程度

