

OpenAirInterface Software Installation/Build/Run

Environment

OS: Ubuntu 16.04.6

Kernel : 4.15.0-13-lowlatency

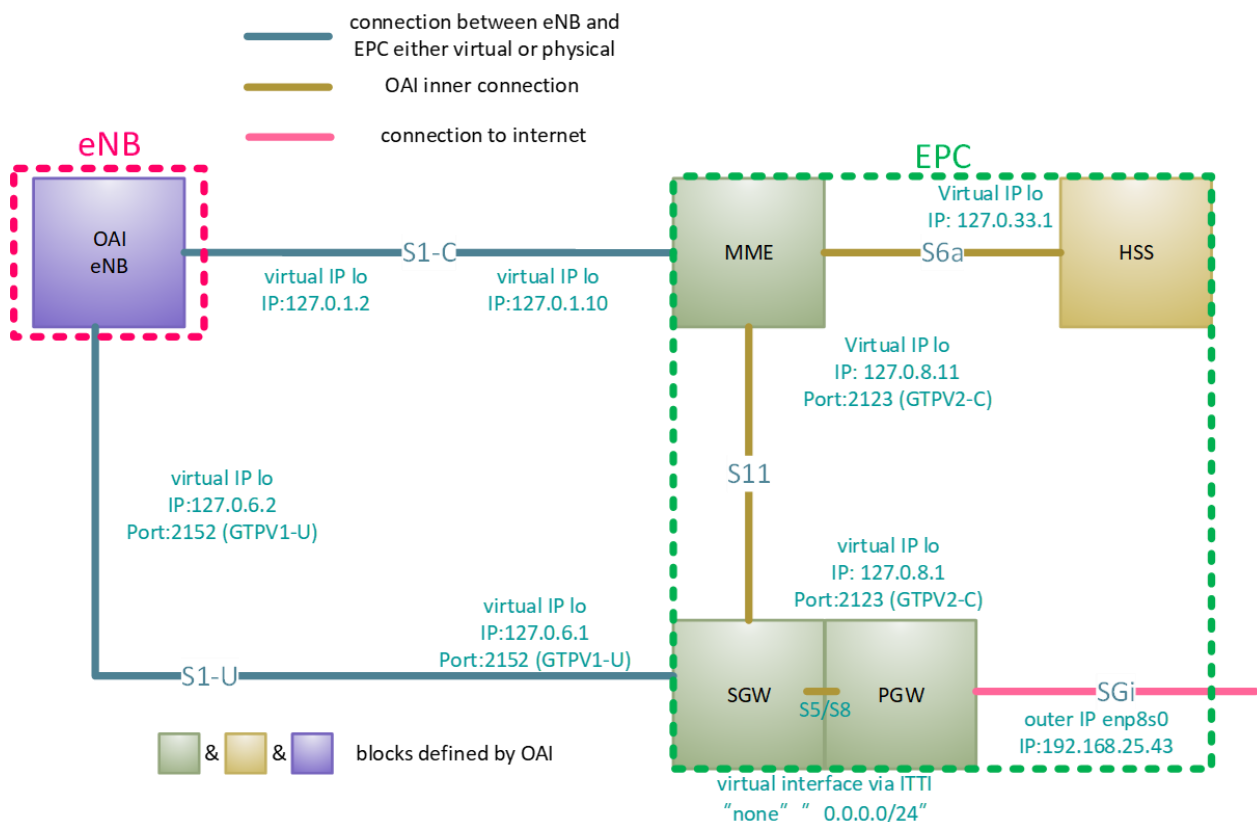
OAI CN branch : 724542d0

OAI eNB branch : 67df8e0e

USRP : B210 / miniB200

USRP drivers : UHD_003.010.002

In this manual, **NIC: enp8s0, IP: 192.168.25.43**



Step 1, Install Ubuntu

Step 2, Change Kernel to Lowlatency

Step 3, Install USRP Drivers

Step 4, Download, Patch, Conf., and Compile OAI EPC

Step 5, Download and Compile OAI eNB

Step 6, Configure eNB and run

Step 1: Install Ubuntu

1. Prepare a machine with at least 4 physical cores, no hyper-threading.

- ✧ All other configuration OAI Wiki (C1 states, ...) describes is about useless.
- ✧ <https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/home>

2. Download Ubuntu 16.04.6 64 bits version ISO file

3. Create a bootable USB with the ISO file download above.

4. Install Ubuntu.

5. Do: *apt update*; *apt upgrade* in order to let the machine is up-to-date

```
sudo apt-get update  
sudo apt-get upgrade
```

6. Install git and other utility packages.

```
sudo apt-get install git vim ssh subversion gitk -y
```

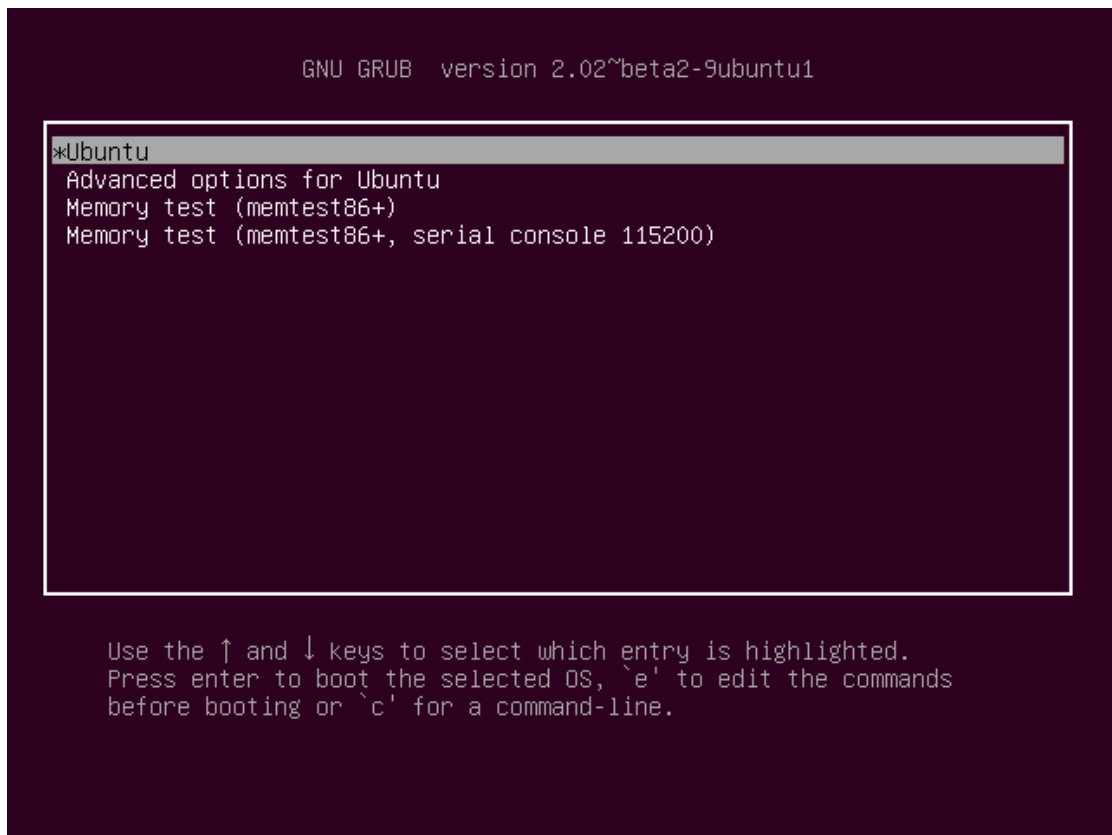
Step 2: Change Kernel to Lowlatency

1. Install low-latency kernel.

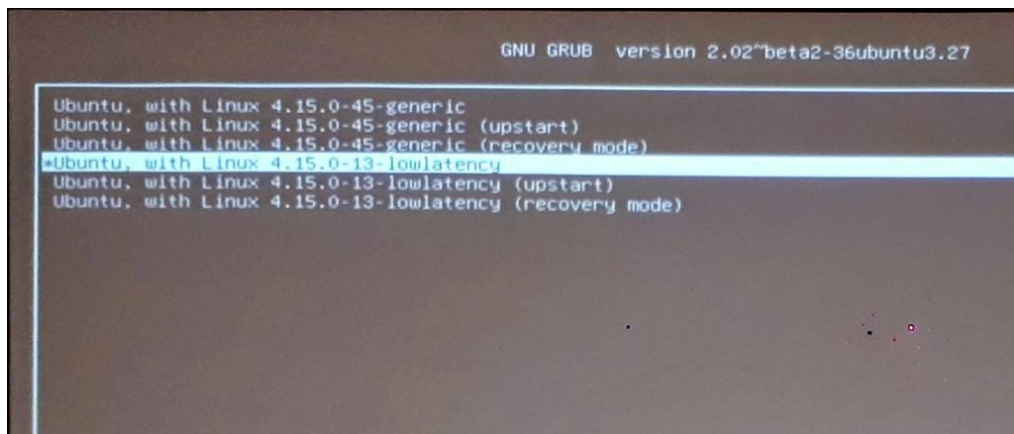
```
sudo apt-get install linux-image-4.15.0.13-lowlatency -y
```

2. Reboot to the low-latency kernel.

- A. Press **ESC** when the boot logo just gone.
- B. Use **↑** and **↓** to select the “Advanced options for Ubuntu”.



- C. Select the lowlatency one.



3. Purge all Linux kernel except for the low-latency one.

- A. Check the current kernel is the low-latency one.

```
uname -a
```

✧ The output of this command should show something like `linux-image-XXX-lowlatency`.

- B. Get the list of installed kernels.

```
dpkg --get-selections | grep linux-image
```

- C. Remove all kernel except for low-latency one. **Replace `linux-image-XXX` with the output of the previous command.**

```
sudo apt-get purge linux-image-XXX
```

- D. Remove unused files.

```
sudo apt-get autoremove
```

4. Configure booting modules.

```
sudo sh -c "echo 'gtp' >> /etc/modules"
```

Step 3: Install USRP Drivers

1. Install required packages.

```
sudo apt-get install libboost-all-dev libusb-1.0-0-dev -y
sudo apt-get install python-mako doxygen python-docutils -y
sudo apt-get install python-requests cmake build-essential -y
```

2. Clone repository.

```
cd ~
```

```
git clone --branch release_003_010_002_000 https://github.com/EttusResearch/uhd.git
```

3. Build USRP drivers

```
cd uhd; sudo mkdir host/build; cd host/build
```

```
sudo cmake -DCMAKE_INSTALL_PREFIX=/usr ..
```

```
sudo make -j4
```

```
sudo make install
```

4. Setup Driver.

```
sudo ldconfig
```

```
sudo /usr/lib/uhd/utils/uhd_images_downloader.py
```

● If you installed the failed version of USRP:

Do not install the latest UHD driver or UHD driver after version 3.10.003. If you installed the UHD driver via command `sudo ./build -I -w USRP`, please uninstall the UHD driver.

```
sudo uhd_usrp_probe
```

```
sudo dpkg --get-selections | grep uhd
```

```
sudo apt-get purge uhd-host libuhd003:amd64 libuhd-dev
```

```
sudo apt-get autoremove
```

Step 4: Download, Patch, Conf. ,and Compile OAI EPC

1. Specify a fully qualified domain name (FQDN) for EPC.

A. Check the host name.

```
cat /etc/hostname
```

- ✧ We will use the output of this command later and we will use the label `<hostname>` to mention this output.
- ✧ Once you see `<hostname>` in the latter instruction, just enter the output of this command.

B. Add the hosts.

```
sudo vim /etc/hosts
```

- ✧ Modify the file so the file show likes below.

```
127.0.0.1      localhost
127.0.1.1      <hostname>.isip.cs.nctu.edu.tw <hostname>
127.0.33.1     hss.isip.cs.nctu.edu.tw hss
```

C. Check the config is setting correct.

```
hostname -f
```

- ✧ The result should show likes below

```
<hostname>.isip.cs.nctu.edu.tw
```

2. Download EPC source code.

A. Create Directory.

```
sudo mkdir /opt/epc
```

```
cd /opt/epc
```

B. Download the repository from Google Drive.

```
https://reurl.cc/NXjRKe
```

```
tar -xf openair-cn.tar -C /opt/epc/openair-cn
```

```
cd /opt/epc/openair-cn
```

C. Checkout to the required commit.

```
sudo git checkout 724542d0
```

3. Apply Patch

- A. Download the patch file from the provided URL. If you download it from the browser, it should appear in the directory ~/Downloads.

- B. Apply patch.

```
cd /opt/epc/openair-cn
```

```
sudo git apply ~/Downloads/EPC.patch
```

4. EPC (eNB and EPC(MME+SPGW)+HSS on different hosts) Configuration.

- A. Prepare required files.

```
sudo mkdir -p /usr/local/etc/oai/freeDiameter
sudo cp /opt/epc/openair-cn/etc/mme.conf /usr/local/etc/oai
sudo cp /opt/epc/openair-cn/etc/hss.conf /usr/local/etc/oai
sudo cp /opt/epc/openair-cn/etc/spgw.conf /usr/local/etc/oai
sudo cp /opt/epc/openair-cn/etc/acl.conf /usr/local/etc/oai/freeDiameter
sudo cp /opt/epc/openair-cn/etc/mme_fd.conf /usr/local/etc/oai/freeDiameter
sudo cp /opt/epc/openair-cn/etc/hss_fd.conf /usr/local/etc/oai/freeDiameter
```

- B. MME Configuration (/usr/local/etc/oai/mme.conf).

```
24  REALM = "isip.cs.nctu.edu.tw";
...
150  NETWORK_INTERFACES :
151  {
152      # MME binded interface for S1-C or S1-MME communication.....
      # YOUR NETWORK CONFIG HERE
153      MME_INTERFACE_NAME_FOR_S1_MME = "lo";
      # YOUR NETWORK CONFIG HERE
154      MME_IPV4_ADDRESS_FOR_S1_MME   = "127.0.1.10/24";
      # MME binded interface for S11 communication (GTPV2-C)
      # YOUR NETWORK CONFIG HERE
157      MME_INTERFACE_NAME_FOR_S11_MME = "lo";
      # YOUR NETWORK CONFIG HERE
158      MME_IPV4_ADDRESS_FOR_S11_MME   = "127.0.8.11/8";
      # YOUR NETWORK CONFIG HERE
159      MME_PORT_FOR_S11_MME           = 2123;
160  };
204  S-GW :
205  {
206      # S-GW binded interface for S11 communication (GTPV2-C).....
      # YOUR NETWORK CONFIG HERE
207      SGW_IPV4_ADDRESS_FOR_S11       = "127.0.8.1/8";
209  };
```

C. SPGW Configuration (/usr/local/etc/oai/spgw.conf).

```
21 S-GW :
22 {
23     NETWORK_INTERFACES :
24     {
25         # S-GW binded interface for S11 communication (GTPV2-C).....
26         # YOUR NETWORK CONFIG HERE
27         SGW_INTERFACE_NAME_FOR_S11      = "lo";
28         # YOUR NETWORK CONFIG HERE
29         SGW_IPV4_ADDRESS_FOR_S11        = "127.0.8.1/8";
30         ...
31         # S-GW binded interface for S1-U communication (GTPV1-U).....
32         # YOUR NETWORK CONFIG HERE, USE "lo" if S-GW run on eNB host
33         SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP = "lo";
34         # YOUR NETWORK CONFIG HERE
35         SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP = "127.0.6.1/24";
36         # PREFER NOT CHANGE UNLESS YOU KNOW WHAT YOU ARE DOING
37         SGW_IPV4_PORT_FOR_S1U_S12_S4_UP    = 2152;
38         ...
39     }
40 }
41
42 P-GW =
43 {
44     NETWORK_INTERFACES :
45     {
46         # P-GW binded interface for S5 or S8 communication,.....
47         # DO NOT CHANGE (NOT IMPLEMENTED YET)
48         PGW_INTERFACE_NAME_FOR_S5_S8      = "none";
49         # Add this line. DO NOT CHANGE (NOT IMPLEMENTED YET)
50         PGW_IPV4_ADDRESS_FOR_S5_S8        = "0.0.0.0/24";
51         # P-GW binded interface for SGI.....
52         # YOUR NETWORK CONFIG HERE (your Static IP)
53         PGW_INTERFACE_NAME_FOR_SGI        = "enp8s0";
54         # YOUR NETWORK CONFIG HERE
55         PGW_MASQUERADE_SGI                 = "yes";
56         UE_TCP_MSS_CLAMPING                 = "no";
57         ...
58         ...
59     }
60 }
61
62 # DNS address communicated to UEs
63 DEFAULT_DNS_IPV4_ADDRESS      = "8.8.8.8"; # YOUR NETWORK CONFIG HERE
64 DEFAULT_DNS_SEC_IPV4_ADDRESS = "8.8.4.4"; # YOUR NETWORK CONFIG HERE
```


D. HSS Configuration (/usr/local/etc/oai/hss.conf).

```
25  MYSQL_user = "root";
26  MYSQL_pass = "oailab"; #Put here the root password of mysql database....
...
30  #OPERATOR_key = "1006020f0a478bf6b699f15c062e42b3"; # OP key matching....
31  OPERATOR_key = "11111111111111111111111111111111"; # OP key for
oai_db.sql, Must match to that of UE Sim card, OP_Key
```

E. HSS freeDiameter Configuration

(/usr/local/etc/oai/freeDiameter/hss_fd.conf).

```
7  Identity = "hss.isip.cs.nctu.edu.tw";
...
11 Realm = "isip.cs.nctu.edu.tw";
```

F. MME freeDiameter Configuration

(/usr/local/etc/oai/freeDiameter/mme_fd.conf).

```
4  Identity = "<hostname>.isip.cs.nctu.edu.tw";
5  Realm = "isip.cs.nctu.edu.tw";
...
103 ConnectPeer= "hss.isip.cs.nctu.edu.tw" { ConnectTo = "127.0.33.1";
No_SCTP ; No_IPv6; Prefer_TCP; No_TLS; port = 3868; realm =
"isip.cs.nctu.edu.tw";};
```

G. ACL Configuration (/usr/local/etc/oai/freeDiameter/acl.conf).

```
19 ALLOW_OLD_TLS *.isip.cs.nctu.edu.tw
```

5. Install Certificates.

```
cd /opt/epc/openair-cn/scripts
```

```
sudo ./check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter/ hss.isip.cs.nctu.edu.tw
```

```

enb06@enb06:/opt/epc/openair-cn/scripts$ sudo ./check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter/ hss.isip.cs.
nctu.edu.tw
HSS S6A: Did not find valid certificate in /usr/local/etc/oai/freeDiameter/
HSS S6A: generating new certificate in /usr/local/etc/oai/freeDiameter/...
Creating HSS certificate for user 'hss.isip.cs.nctu.edu.tw'
Generating a 1024 bit RSA private key
...+++++
.....+++++
writing new private key to 'hss.cakey.pem'
-----
Generating RSA private key, 1024 bit long modulus
.....+++++
...+++++
e is 65537 (0x10001)
Using configuration from /usr/lib/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 1 (0x1)
    Validity
        Not Before: Feb 18 13:53:15 2019 GMT
        Not After : Feb 18 13:53:15 2020 GMT
    Subject:
        countryName           = FR
        stateOrProvinceName   = PACA
        organizationName      = Eurecom
        organizationalUnitName = CM
        commonName            = hss.isip.cs.nctu.edu.tw
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        Netscape Comment:
            OpenSSL Generated Certificate
        X509v3 Subject Key Identifier:
            DE:10:03:1D:25:D8:DD:A5:C1:E8:57:4E:A0:C1:2D:55:06:AE:E4:91
        X509v3 Authority Key Identifier:
            keyid:26:66:04:91:26:60:76:92:09:E6:29:7D:52:67:D4:E2:9D:91:76:51

Certificate is to be certified until Feb 18 13:53:15 2020 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
/opt/epc/openair-cn/scripts
HSS S6A: Found valid certificate in /usr/local/etc/oai/freeDiameter/
enb06@enb06:/opt/epc/openair-cn/scripts$

```

```

sudo ./check_mme_s6a_certificate /usr/local/etc/oai/freeDiameter/ <hostname>.isip.cs.nctu.edu.tw

```

```

enb06@enb06:/opt/epc/openair-cn/scripts$ sudo ./check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter/ hss.isip.cs.
nctu.edu.tw
HSS S6A: Did not find valid certificate in /usr/local/etc/oai/freeDiameter/
HSS S6A: generating new certificate in /usr/local/etc/oai/freeDiameter/...
Creating HSS certificate for user 'hss.isip.cs.nctu.edu.tw'
Generating a 1024 bit RSA private key
...+++++
.....+++++
writing new private key to 'hss.cakey.pem'
-----
Generating RSA private key, 1024 bit long modulus
.....+++++
...+++++
e is 65537 (0x10001)
Using configuration from /usr/lib/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 1 (0x1)
    Validity
        Not Before: Feb 18 13:53:15 2019 GMT
        Not After : Feb 18 13:53:15 2020 GMT
    Subject:
        countryName           = FR
        stateOrProvinceName   = PACA
        organizationName      = Eurecom
        organizationalUnitName = CM
        commonName            = hss.isip.cs.nctu.edu.tw
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        Netscape Comment:
            OpenSSL Generated Certificate
        X509v3 Subject Key Identifier:
            DE:10:03:1D:25:D8:DD:A5:C1:E8:57:4E:A0:C1:2D:55:06:AE:E4:91
        X509v3 Authority Key Identifier:
            keyid:26:66:04:91:26:60:76:92:09:E6:29:7D:52:67:D4:E2:9D:91:76:51

Certificate is to be certified until Feb 18 13:53:15 2020 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
/opt/epc/openair-cn/scripts
HSS S6A: Found valid certificate in /usr/local/etc/oai/freeDiameter/
enb06@enb06:/opt/epc/openair-cn/scripts$

```

6. Build EPC.

```
cd /opt/epc/openair-cn/scripts
```

A. Build HSS.

```
sudo ./build_hss -i
```

- ✧ Do you want to continue? [Y/n] ==> **y**
- ✧ Enter MySQL Password ==> **oailab (twice)**
- ✧ Do you want to install freeDiameter 1.2.0? <y/N> ==> **y (three times)**
- ✧ Configuration phpmyadmin ==> Select **Apache2**
- ✧ Configuration phpmyadmin ==> **<yes>**
- ✧ PhpMyAdmin Password ==> **oailab (twice)**

B. Build MME.

```
sudo ./build_mme -i
```

- ✧ Do you want to continue? [Y/n] ==> **y (twice)**
- ✧ Do you want to install freeDiameter 1.2.0? <y/N> ==> **N** (We don't need to install the same thing twice.)
- ✧ Do you want to install asn1c rev 1516 patched? <y/N> ==> **y (twice)**
- ✧ Do you want to install libgtpnl? <y/N> ==> **y (twice)**
- ✧ Configuring wireshark-common ==> **No**

C. Build SPGW.

```
sudo ./build_spgw -i
```

- ✧ Do you want to install libgtpnl? [y/N] ==> **N** (We don't need to install the same thing twice.)

7. Compile EPC

A. Install phpMyAdmin.

```
sudo apt-get install phpmyadmin
```

```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf
```

```
sudo a2enconf phpmyadmin
```

```
sudo /etc/init.d/apache2 reload
```

B. Compile HSS.

```
sudo ./build_hss -c
```

C. Compile MME.

```
sudo ./build_mme -c
```

D. Compile SPGW.

```
sudo ./build_spgw -c
```

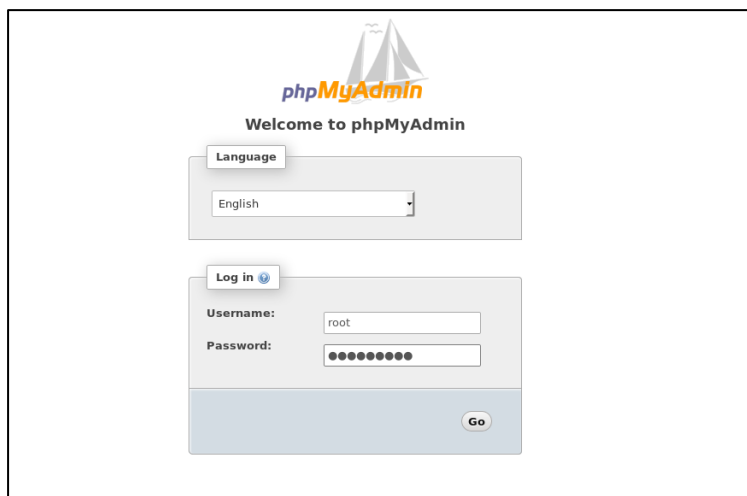
8. Run EPC.

```
cd /opt/epc/openair-cn/scripts
```

A. Prepare Database.

```
sudo ./run_hss -i /opt/epc/openair-cn/src/oai_hss/db/oai_db.sql
```

✧ Launch the browser and go to the site: localhost/phpmyadmin



✧ Add/Modify record into the database.

- I. Modify the record in the database so it would be matched with the settings in the `mme_fd.conf` we configure previous. As a result, in the table ``mmeidentity``, the ``mmehost`` column should be `<hostname>.isip.cs.nctu.edu.tw`, and ``mmerealm`` column should be `isip.cs.nctu.edu.tw`.

	id	mmeidentity	mmehost	mmerealm	UE-Reachability
<input type="checkbox"/>	1		nano.openair4G.eur	openair4G.eur	0
<input type="checkbox"/>	2		mme2.openair4G.eur	openair4G.eur	0
<input type="checkbox"/>	5		abeille.openair4G.eur	openair4G.eur	0
<input type="checkbox"/>	4		epc01.isip.cs.nctu.edu.tw	isip.cs.nctu.edu.tw	0
<input type="checkbox"/>	3		mme3.openair4G.eur	openair4G.eur	0
<input type="checkbox"/>	6		calisson.openair4G.eur	openair4G.eur	0

II. Add the ID of the SIM card in the phone into the database. (suppose ID = 32)

- Add record in `oai_db.pdn` table: copy the data with `id` = `1`, and modify the value of the column `users_imsi` to 208930000000032
- Add record in `oai_db.users` table: copy the data with `imsi` = `208930000000001`, and modify the value of the column `imsi` to 208930000000032, and the value of the column `msisdn` to 33638030032, and modify the value of the column `imei` to 35609204079332.

B. Run HSS.

```
sudo ./run_hss
```

C. Run MME. (in a new terminal)

```
cd /opt/epc/openair-cn/scripts  
sudo ./run_mme
```

D. Run SPGW. (in a new terminal)

```
cd /opt/epc/openair-cn/scripts  
sudo ./run_spgw
```

Step 5: Download and Compile OAI eNB

1. Download OAI eNB.

```
cd ~
```

```
git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git
```

2. Compile OAI eNB.

```
cd openairinterface5g
```

```
git checkout 67df8e0e
```

✧ Checkout to the required commit.

```
source oaienv
```

✧ Very Important!! It will set the required environment variables.

```
cd cmake_targets
```

```
sudo ./build_oai -I
```

✧ ./build_oai options:

- **-I**: installs required packages.
- **-w**: Add the hardware support, which is USRP in our case.
- **-x**: Add a software oscilloscope feature to the produced binaries.
- **--eNB**: Installs eNB, i.e., lte-softmodem.
- **--install-system-files**: Installs OAI required files in Linux system.
- **--install-optional-packages**: Install optional packages.

✧ If build failed ==> **AttributeError: 'module' object has no attribute 'SSL_ST_INIT'**

```
rm -rf /usr/lib/python2.7/dist-packages/OpenSSL
```

```
rm -rf /usr/lib/python2.7/dist-packages/pyOpenSSL- 0.15.1.egg- info
```

```
sudo pip install pyopenssl
```

3. Build OAI eNB

A. For commonly used.

```
sudo ./build_oai -c --eNB -w USRP
```

B. Compilation with showing QAM figure later.

```
sudo ./build_oai -c --eNB -w USRP -x
```

C. Compilation with showing VCD figure later.

```
sudo ./build_oai -c --eNB -w USRP -V
```

Step 6: Configure eNB and Run

1. eNB Configuration (~/.openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tml.usrpb210.conf).

```
18  mobile_country_code = "208";
20  mobile_network_code = "93";
...
138  ////////// MME parameters:
140  mme_ip_address      = ( { ipv4      = "127.0.1.10";
141                          ipv6      = "192:168:30::17";
142                          active    = "yes";
143                          preference = "ipv4";
144                          }
145                          );
147  NETWORK_INTERFACES :
...
150  ENB_INTERFACE_NAME_FOR_S1_MME      = "1o";
151  ENB_IPV4_ADDRESS_FOR_S1_MME        = "127.0.1.2/8";
152  ENB_INTERFACE_NAME_FOR_S1U        = "1o";
153  ENB_IPV4_ADDRESS_FOR_S1U          = "127.0.6.2/8";
154  ENB_PORT_FOR_S1U                  = 2152; # Spec 2152
```

2. Run eNB in a new terminal. **(Remember to connect the USRP first.)**

```
cd ~/.openairinterface5g/cmake_targets/lte_build_oai/build
```

- A. Run with original setting.

```
sudo ./lte-softmodem -O ../../../../targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tml.usrpb210.conf
```

- B. Run with QAM figure showing.

```
sudo ./lte-softmodem -O ../../../../targets/PROJECT/GENERIC-LTE-EPC/CONF/enb.band7.tml.usrpb210.epc.local.conf -d
```

- C. Run with VCD figure showing.

```
sudo ./lte-softmodem -O ../../../../targets/PROJECT/GENERIC-LTE-EPC/CONF/enb.band7.tml.usrpb210.epc.local.conf -V
```

3. Mobile Setting.

- A. Setting/Connections/Mobile network/APN/

Add the apn = oai.ipv4, name = oai (Save)

- B. Turnoff “Airplane mode” and turn on “roaming” and “cellular data”.