教育部「5G行動寬頻人才培育跨校教學聯盟計畫」

5G行動網路協定與核網技術聯盟中心

課程:5G垂直應用網路

實驗四
mMTC垂直應用網路實驗

副教授：吳俊興

助教：胡詠翔

國立高雄大學 資訊工程學系

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝
- Stage 3. srsLTE設定及量測
- Stage4. srsLTE參數調整
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
- 總結及問題

實驗目的

1. 在單板電腦(如Raspberry Pi 4)上連結SDR建置低耗能的mMTC UE應用系統
2. 調整訊號及網路參數來分析及量測其對系統及網路效能的影響

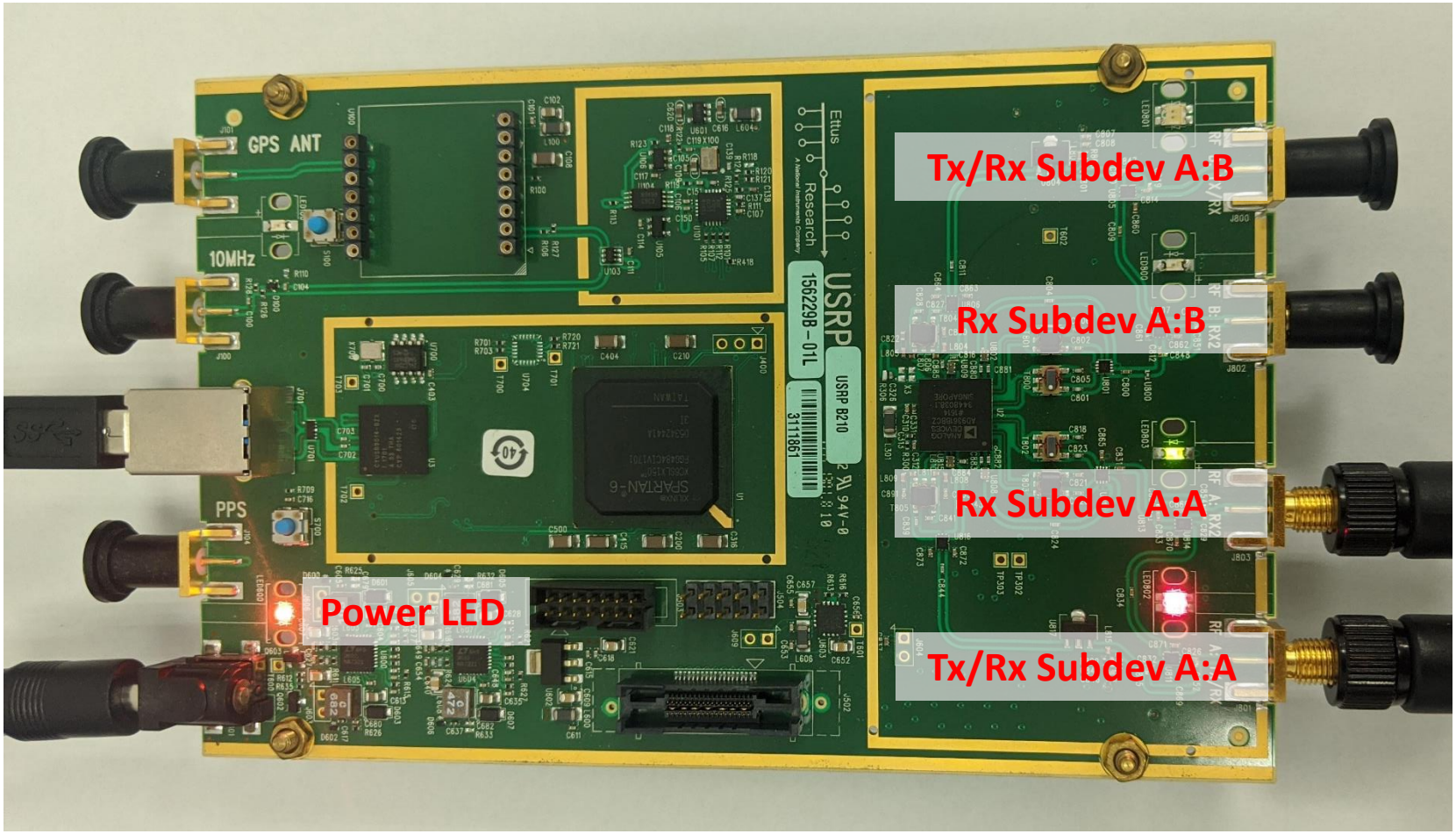
實驗內容

- 架設樹莓派之實驗環境
- 安裝USRP Hardware Driver(UHD)及srsLTE
- 設定srsLTE以讓其在樹莓派上運作並進行相關測試
  - srsLTE內的UHD設定
  - 設定USIM
  - UHD內建的頻譜分析儀使用
  - srsGUI各欄位意義
- 調整參數並觀察對整體Throughput的影響
  - Physical Resource Blocks
  - PDSCH MCS
  - PUSCH MCS
- 利用srsLTE觀察NB-IoT
  - 商用eNB搜尋及其MIB與SIB觀察
  - 自行架設NB-IoT之eNB與UE
- mMTC 之應用 - nukxScan

Outline

- 實驗目的及實驗內容
- 背景知識
  - USRP B210 子裝置配置
  - Physical Resource Block
- 實驗環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝
- Stage 3. srsLTE設定及量測
- Stage4. srsLTE參數調整
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
- 總結及問題

背景知識 - USRP B210 子裝置配置



背景知識 - Physical Resource Block

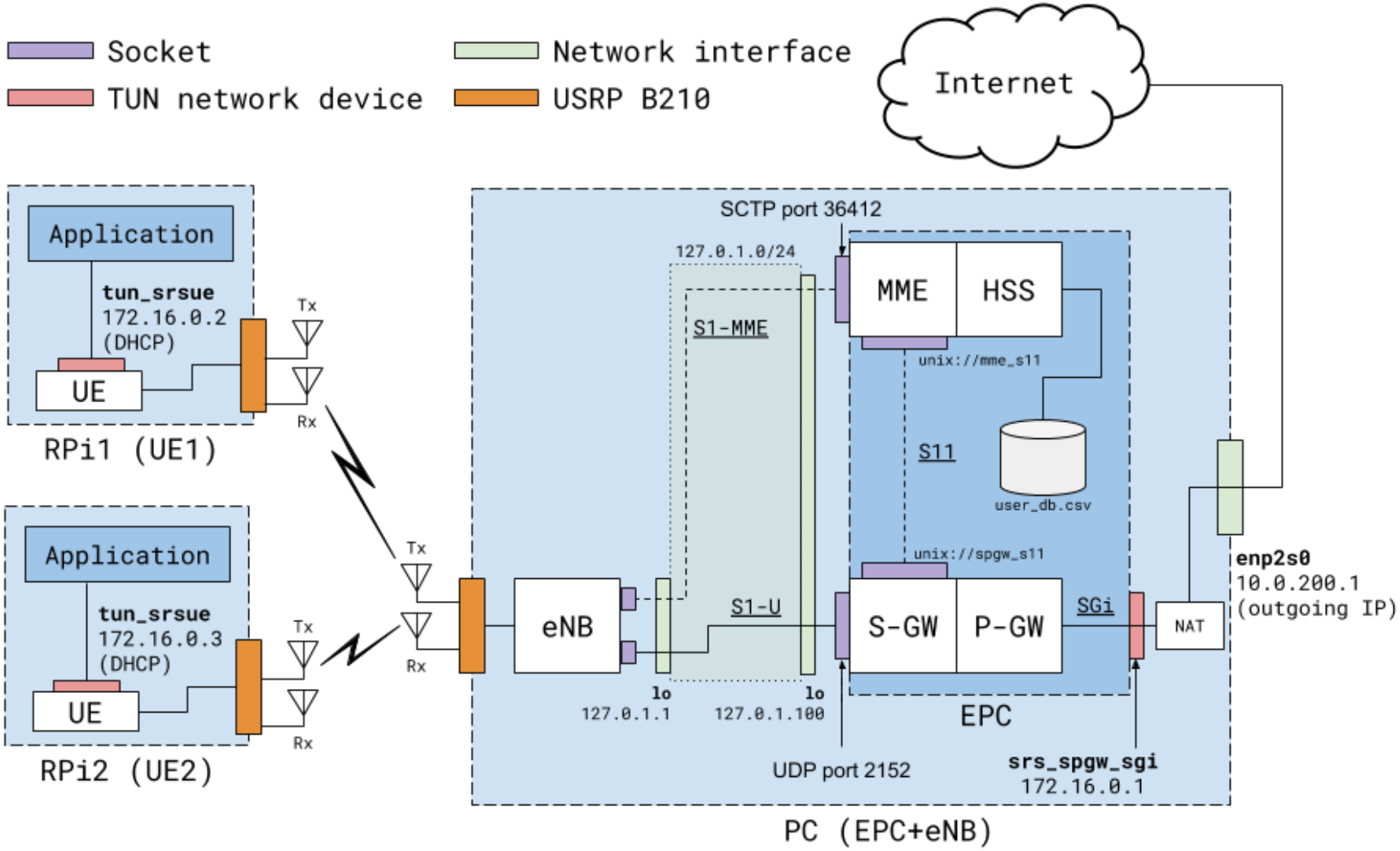
- Resource Block 是 E-UTRA 系統分配無線電資源給使用者的最小單元
- 每個Resource Block在頻域和時域的大小分別是180KHz和1個Slot
- PRB和Bandwidth的對應關係如下表所示
- srsLTE在樹莓派上只能支援15和6個PRBs

Physical Resource Blocks	Bandwidth
6	1.4MHz
15	3MHz
25	5MHz
50	10MHz
75	15MHz
100	20MHz

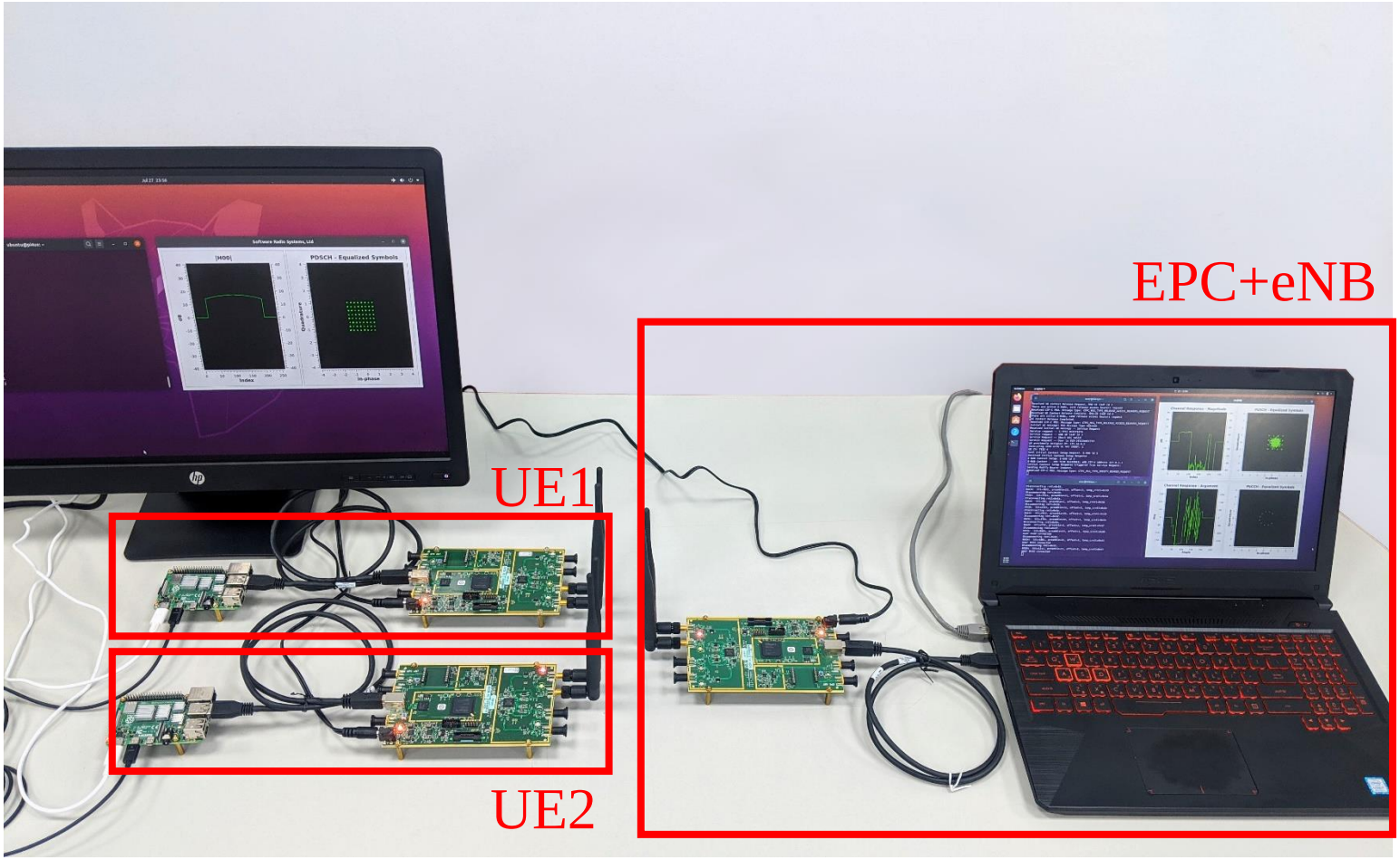
Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
  - 實驗架構
  - 實際架構
  - 軟硬體環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝
- Stage 3. srsLTE設定及量測
- Stage 4. srsLTE參數調整
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
- 總結及問題

實驗架構


Socket
TUN network device
Network interface
USRP B210
Internet
Application
tun_srsue
172.16.0.2
(DHCP)
UE
RPi1 (UE1)
Application
tun_srsue
172.16.0.3
(DHCP)
UE
RPi2 (UE2)
Tx
Rx
SCTP port 36412
127.0.1.0/24
S1-MME
MME
HSS
unix://mme_s11
S11
user_db.csv
unix://spgw_s11
eNB
S1-U
S-GW
P-GW
SGi
NAT
enp2s0
10.0.200.1
(outgoing IP)
lo
127.0.1.1
lo
127.0.1.100
EPC
UDP port 2152
srs_spgw_sgi
172.16.0.1
PC (EPC+eNB)

實際環境



軟硬體環境–硬體

名稱	規格	數量	目的
EPC+ eNB	電腦型號：ASUS MW504G	1	啟動 MME,HSS,S-GW,P-GW,eNB
	USRP B210	1	與srsLTE溝通以傳送與接收無線電訊號
	VERT2450	2	2.4-2.5GHz天線
UE	Raspberry Pi 4B 8G	2	模擬 UE
	USRP B210	2	與srsLTE溝通以傳送與接收無線電訊號
	VERT2450	4	2.4-2.5GHz天線

軟硬體環境–軟體

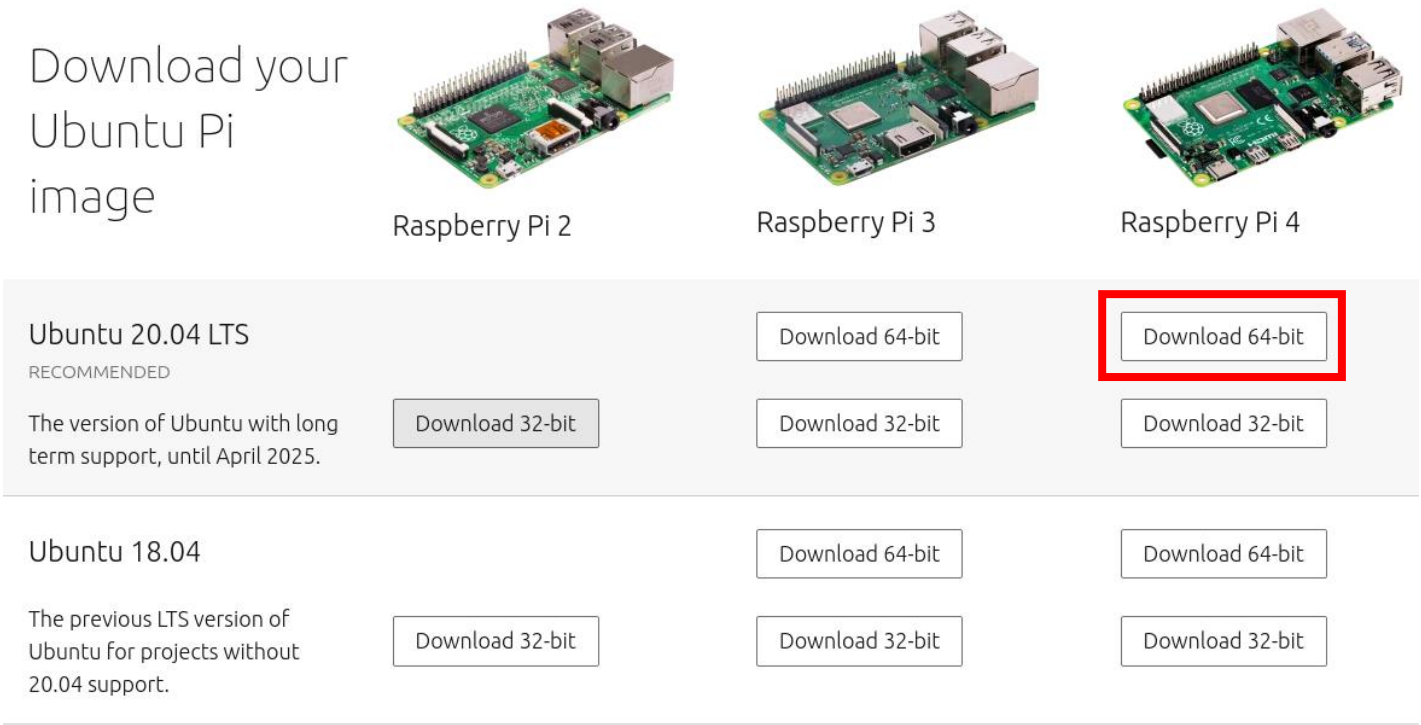
名稱	軟體	版本
EPC+ eNB	OS : Ubuntu	Ubuntu 20.04
	srsLTE	srsLTE 20.04.1 c892ae56be5302eaee5ca00e270efc7a5ce6fbb2
UE	OS : Ubuntu	Ubuntu 20.04
	srsLTE	srsLTE 20.04.1 c892ae56be5302eaee5ca00e270efc7a5ce6fbb2

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 樹莓派環境架設
  - Step1 事前準備
  - Step2 初次使用
  - Step3 設定sudo免密碼
  - Step4 軟體安裝
  - Step5 更改RPi電源組態
- Stage 2. USRP與srsLTE安裝
- Stage 3. srsLTE設定及量測
- Stage 4. srsLTE參數調整
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
- 總結及問題

Step1-1 事前準備(下載映像檔)

至Ubuntu官網的Raspberry Pi專區，點選下載Raspberry Pi 4專用的Ubuntu Server 20.04 LTS 64-bi 的版本



https://ubuntu.com/download/raspberry-pi

Step1-2 事前準備(解壓縮映像檔)

在一台有Linux電腦終端機中輸入xz -dv ubuntu-20.04-preinstalled-server-arm64+raspi.img.xz將下載完畢的映像解壓縮得到完整的Ubuntu Server 20.04 LTS之SD卡映像檔ubuntu-20.04-preinstalled-server-arm64+raspi.img

```
$ xz -dv ubuntu-20.04-preinstalled-server-arm64+raspi.img.xz
ubuntu-20.04-preinstalled-server-arm64+raspi.img.xz (1/1)
  100 %     667.0 MiB / 3,054.4 MiB = 0.218    81 MiB/s       0:37
```

Step1-3 事前準備(燒錄映像檔)

解壓縮完後輸入sudo dd if=ubuntu-20.04-preinstalled-server-arm64+raspi.img of=/dev/sdx status=progress 將解壓縮後的映像檔燒錄至記憶卡，其中/dev/sdx為記憶卡的磁碟代號，請特別注意Ubuntu的映像檔包含磁碟分區，因此將其燒錄至已存在的分區將不會正常動作，整個過程耗時約15分鐘

```
$ sudo dd if=ubuntu-20.04-preinstalled-server-arm64+raspi.img of=/dev/sdd status=progress
1023029760 bytes (1.0 GB, 976 MiB) copied, 285 s, 3.6 MB/s
```

Step2 初次使用

將燒錄完成的記憶卡插入Raspberry Pi並接上鍵盤、螢幕、滑鼠及電源，應可看到Raspberry Pi順利開機，預設之帳號及密碼皆為ubuntu，登入後需輸入新密碼並重新登入

```
You are required to change your password immediately (administrator enforced)
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1008-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Apr  1 17:27:35 UTC 2020

  System load:  0.31              Temperature:           46.3 C
  Usage of /:   3.1% of 58.24GB   Processes:             132
  Memory usage: 3%                Users logged in:       0
  Swap usage:   0%                IPv4 address for eth0: 10.0.0.12


0 updates can be installed immediately.
0 of these updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Wed Apr  1 17:26:59 2020 from 10.0.0.254
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for ubuntu.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

Step3 設定sudo免密碼

在終端機中輸入sudo EDITOR="vim" visudo，並將%sudo ALL = (ALL:ALL) ALL這行改為%sudo ALL = (ALL:ALL) NOPASSWD:ALL以方便未來使用sudo進行提權操作時不需再輸入密碼，改完後請按Esc並輸入:wq以離開並存檔

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) NOPASSWD:ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
                                                    30,1          Bot
```

Step4-1 軟體安裝(更新)

在終端機中輸入sudo apt update && sudo apt upgrade -y，以進行系統軟體之更新

```
Get:43 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 apport all 2.20.11-0ubuntu27.3 [128 kB]
Get:44 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libproxy1v5 arm64 0.4.15-10ubuntu1 [46.2 kB]
Get:45 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 landscape-common arm64 19.12-0ubuntu4.1 [86.7 kB]
Get:46 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libasound2 arm64 1.2.2-2.1ubuntu1 [299 kB]
Get:47 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 libasound2-data all 1.2.2-2.1ubuntu1 [19.1 kB]
Get:48 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 linux-firmware all 1.187.1 [99.0 MB]
Get:49 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 software-properties-common all 0.98.9.1 [10.5 kB]
Get:50 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 python3-software-properties all 0.98.9.1 [25.2 kB]
Get:51 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 snapd arm64 2.45.1+20.04 [24.2 MB]
Get:52 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 sosreport arm64 3.9.1-1ubuntu0.20.04.1 [169 kB]
Get:53 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 ubuntu-server arm64 1.450.1 [2660 B]
Get:54 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 wpasupplicant arm64 2:2.9-1ubuntu4.1 [1087 kB]
Fetched 137 MB in 56s (2467 kB/s)
Extracting templates from packages: 100%
Preconfiguring packages ...
(Reading database ... 66630 files and directories currently installed.)
Preparing to unpack .../login_1%3a4.8.1-1ubuntu5.20.04_arm64.deb ...
Unpacking login (1:4.8.1-1ubuntu5.20.04) over (1:4.8.1-1ubuntu5) ...
Setting up login (1:4.8.1-1ubuntu5.20.04) ...
(Reading database ... 66630 files and directories currently installed.)
Preparing to unpack .../udev_245.4-4ubuntu3.1_arm64.deb ...
Unpacking udev (245.4-4ubuntu3.1) over (245.4-4ubuntu3) ...
```

Step4-2 軟體安裝(測試軟體)

在終端機中輸入sudo apt install -y iperf3 wireshark以安裝後續會使用到的流通量測試軟體iperf3及封包分析軟體wireshark

```
Need to get 19.0 kB of archives.
After this operation, 119 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 iperf3 arm64 3.7
-3 [14.0 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports focal/universe arm64 wireshark arm64
3.2.3-1 [5088 B]
Fetched 19.0 kB in 1s (23.8 kB/s)
Selecting previously unselected package iperf3.
(Reading database ... 225325 files and directories currently installed.)
Preparing to unpack .../iperf3_3.7-3_arm64.deb ...
Unpacking iperf3 (3.7-3) ...
Selecting previously unselected package wireshark.
Preparing to unpack .../wireshark_3.2.3-1_arm64.deb ...
Unpacking wireshark (3.2.3-1) ...
Setting up wireshark (3.2.3-1) ...
Setting up iperf3 (3.7-3) ...
Processing triggers for man-db (2.9.1-1) ...
```

Step4-3 軟體安裝(tasksel)

在終端機中輸入sudo apt install -y lightdm tasksel以安裝Linux的顯示管理器及後續會用到的tasksel

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  acl adwaita-icon-theme-full apg aptdaemon aptdaemon-data aspell aspell-en avahi-daemon avahi-ut
  dbus-x11 dconf-cli desktop-file-utils dictionaries-common dns-root-data dnsmasq-base docbook-xm
  fprintd gcr geoclue-2.0 gir1.2-accountsservice-1.0 gir1.2-atk-1.0 gir1.2-atspi-2.0 gir1.2-freed
  gir1.2-geoclue-2.0 gir1.2-gnomebluetooth-1.0 gir1.2-gnomedesktop-3.0 gir1.2-graphene-1.0 gir1.2
  gir1.2-nma-1.0 gir1.2-notify-0.7 gir1.2-pango-1.0 gir1.2-polkit-1.0 gir1.2-rsvg-2.0 gir1.2-secr
  gnome-accessibility-themes gnome-control-center gnome-control-center-data gnome-control-center-
  gnome-online-accounts gnome-session gnome-session-bin gnome-session-common gnome-settings-daemo
  gnome-themes-extra gnome-themes-extra-data gnome-user-docs gstreamer1.0-clutter-3.0 gstreamer1.
  gstreamer1.0-x gtk2-engines-pixbuf hunspell-en-us ibus ibus-data ibus-gtk ibus-gtk3 iio-sensor-
  libasound2-plugins libaspell15 libavahi-core7 libavahi-glib1 libavc1394-0 libbluetooth3 libcame
  libclutter-1.0-common libclutter-gst-3.0-0 libclutter-gtk-1.0-0 libcogl-common libcogl-pango20
  libdbusmenu-glib4 libdbusmenu-gtk3-4 libdv4 libebackend-1.2-10 libebook-1.2-20 libebook-contact
  libedataserverui-1.2-2 libenchant-2-2 libexif12 libfontenc1 libfprint-2-2 libfprint-2-tod1 libg
  libgeoclue-2-0 libgeocode-glib0 libgjs0g libgles2 libgnome-autoar-0-0 libgnome-bluetooth13 libg
  libgoa-backend-1.0-1 libgphoto2-6 libgphoto2-l10n libgphoto2-port12 libgraphene-1.0-0 libgsound
  libgstreamer-plugins-good1.0-0 libgtop-2.0-11 libgtop2-common libgupnp-1.2-0 libgupnp-av-1.0-2
  libhyphen0 libibus-1.0-5 libical3 libidn11 libiec61883-0 libieee1284-3 libimobiledevice6 libjan
  libmm-glib0 libmozjs-68-0 libmp3lame0 libmpg123-0 libmutter-6-0 libmysqlclient21 libndp0 libnm0
  libpangoxft-1.0-0 libpciaccess0 libphonenumber7 libplist3 libprotobuf17 libpulse-mainloop-glib0
```

Step4-4 軟體安裝(桌面環境)

在終端機中輸入sudo tasksel，並在tasksel的畫面中利用Tab或是Enter鍵選擇Ubuntu desktop並選擇OK以安裝桌面環境

Step5-1 更改RPi電源組態(安裝)

在終端機中輸入sudo apt install -y linux-tools-raspi linux-tools-5.4.0-1008-raspi以安裝Linux下的CPU頻率調整軟體cpupower

```
Unpacking linux-tools-common (5.4.0-40.44) ...
Selecting previously unselected package linux-raspi-tools-5.4.0-1013.
Preparing to unpack .../3-linux-raspi-tools-5.4.0-1013_5.4.0-1013.13_arm64.deb ...
Unpacking linux-raspi-tools-5.4.0-1013 (5.4.0-1013.13) ...
Selecting previously unselected package linux-tools-5.4.0-1013-raspi.
Preparing to unpack .../4-linux-tools-5.4.0-1013-raspi_5.4.0-1013.13_arm64.deb ...
Unpacking linux-tools-5.4.0-1013-raspi (5.4.0-1013.13) ...
Selecting previously unselected package linux-tools-raspi.
Preparing to unpack .../5-linux-tools-raspi_5.4.0.1013.13_arm64.deb ...
Unpacking linux-tools-raspi (5.4.0.1013.13) ...
Setting up libdw1:arm64 (0.176-1.1build1) ...
Setting up libunwind8:arm64 (1.2.1-9build1) ...
Setting up linux-tools-common (5.4.0-40.44) ...
Setting up linux-raspi-tools-5.4.0-1013 (5.4.0-1013.13) ...
Setting up linux-tools-5.4.0-1013-raspi (5.4.0-1013.13) ...
Setting up linux-tools-raspi (5.4.0.1013.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
ubuntu@ubuntu:~$
```

Step5-2 更改RPi電源組態(設定)

在終端機中輸入sudo vim /etc/default/cpufrequtils並輸入如下圖所示的內容，輸入完後按下Esc並輸入:wq存檔並離開，存檔完後請輸入sudo reboot以將Raspberry Pi重新開機

```
ENABLE="true"
GOVORNER="performance"
~
~
~
~
~
~
~
~
"/etc/default/cpufrequtils" 2L, 37C
```

Step5-3 更改RPi電源組態(檢查)

重開機後應可看到剛剛安裝的桌面環境，開啟一個終端機並輸入sudo cpupower frequency-info以確認電源組態使否已正確變更，若current policy為performance即為成功

```
ubuntu@pi4ue2:/lib/uhd/examples$ sudo cpupower frequency-info
analyzing CPU 0:
  driver: BCM2835 CPUFreq
  CPUs which run at the same hardware frequency: 0 1 2 3
  CPUs which need to have their frequency coordinated by software: 0 1 2 3
  maximum transition latency: 355 us
  hardware limits: 600 MHz - 1.50 GHz
  available frequency steps:  600 MHz, 1.50 GHz
  available cpufreq governors: conservative ondemand userspace powersave performance schedutil
  current policy: frequency should be within 600 MHz and 1.50 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency: 1.50 GHz (asserted by call to hardware)
```

Stage 1 Check List

	項目	內容
	Raspberry Pi映像檔	確認映像檔已燒錄進記憶卡並可以使用該記憶卡進行開機
	測試軟體	輸入iperf3 --version及wireshark --version確認必備的測試軟體是否都已正確安裝
	桌面環境	確認ubuntu-desktop已正確安裝且重開機後可以正常啟動
	Raspberry Pi電源組態	輸入sudo cpupower frequency-info並確認已將電源組態設定為performance

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝
  - Step1 安裝Dependency
  - Step2 安裝UHD
  - Step3 下載Image
  - Step4 初始化
  - Step5 安裝srsGUI
  - Step6 安裝srsLTE
- Stage 3. srsLTE設定及量測
- Stage 4. srsLTE參數調整
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
- 總結及問題

Step1 安裝Dependency

在終端機中輸入 sudo apt install -y libfftw3-dev libmbedtls-dev libboost-program-options-dev libconfig++-dev libsctp-dev libczmq-dev cmake build-essential git qtbase5-dev libqwt-qt5-dev 以安裝後續所需要的所有相依程式庫

```
Setting up libstdc++-7-dev:arm64 (7.5.0-6ubuntu2) ...
Setting up libczmq4:arm64 (4.2.0-2) ...
Setting up libfftw3-bin (3.3.8-2ubuntu1) ...
Setting up libboost1.71-dev:arm64 (1.71.0-6ubuntu6) ...
Setting up uuid-dev:arm64 (2.34-0.1ubuntu9) ...
Setting up comerr-dev:arm64 (2.1-1.45.5-2ubuntu1) ...
Setting up libsctp-dev:arm64 (1.0.18+dfsg-1) ...
Setting up libconfig++-dev:arm64 (1.5-0.4build1) ...
Setting up libfftw3-dev:arm64 (3.3.8-2ubuntu1) ...
Setting up krb5-multidev:arm64 (1.17-6ubuntu4) ...
Setting up libboost-program-options1.71-dev:arm64 (1.71.0-6ubuntu6) ...
Setting up libkrb5-dev:arm64 (1.17-6ubuntu4) ...
Setting up libboost-program-options-dev:arm64 (1.71.0.0ubuntu2) ...
Setting up libzmq3-dev:arm64 (4.3.2-2ubuntu1) ...
Setting up libczmq-dev:arm64 (4.2.0-2) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for man-db (2.9.1-1) ...

Progress: [ 99%] [#########################################################################.]
```

Step2 安裝UHD

在終端機中輸入sudo apt install -y libuhd-dev libuhd3.15.0 uhd-host以安裝USRP Hardware driver(UHD)

```
ubuntu@pi4epc:~$ sudo apt install -y libuhd-dev libuhd3.15.0 uhd-host
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme at-spi2-core blt fontconfig fontconfig-config fonts-dejavu-core
  fonts-lyx freeglut3 gnuradio gnuradio-dev gtk-update-icon-cache hicolor-icon-theme
  humanity-icon-theme icu-devtools javascript-common libasyncns0 libatk-bridge2.0-0
  libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data
  libavahi-common3 libblas3 libboost-atomic1.71-dev libboost-atomic1.71.0
  libboost-chrono1.71-dev libboost-chrono1.71.0 libboost-date-time-dev
  libboost-date-time1.71-dev libboost-date-time1.71.0 libboost-filesystem-dev
  libboost-filesystem1.71-dev libboost-filesystem1.71.0 libboost-regex-dev
  libboost-regex1.71-dev libboost-regex1.71.0 libboost-serialization1.71-dev
  libboost-serialization1.71.0 libboost-system-dev libboost-system1.71-dev
  libboost-system1.71.0 libboost-test-dev libboost-test1.71-dev libboost-test1.71.0
  libboost-thread-dev libboost-thread1.71-dev libboost-thread1.71.0 libcaca0
  libcairo-gobject2 libcairo2 libcanberra-gtk-module libcanberra-gtk0 libcanberra-gtk3-0
  libcanberra-gtk3-module libcodec2-0.9 libcolord2 libcppunit-1.15-0 libcppunit-dev libcups2
  libdatrie1 libdouble-conversion3 libdrm-amdgpu1 libdrm-nouveau2 libdrm-radeon1 libegl-mesa0
  libegl1 libepoxy0 libevdev2 libexpat1-dev libflac8 libfontconfig1 libfreetype6
```

Step3 下載Image

安裝完UHD後在終端機輸入sudo /usr/lib/uhd/utils/uhd_images_downloader.py -T以下載USRP SDR中所使用的FPGA映像檔

```
ubuntu@pi4epc:~$ sudo /usr/lib/uhd/utils/uhd_images_downloader.py -T
[INFO] Images destination: /usr/share/uhd/images
[INFO] No inventory file found at /usr/share/uhd/images/inventory.json. Creating an empty one.
19442 kB / 19442 kB (100%) x3xx_x310_fpga_default-gfde2a94e.zip
18697 kB / 18697 kB (100%) x3xx_x300_fpga_default-gfde2a94e.zip
01534 kB / 01534 kB (100%) e3xx_e310_sg1_fpga_default-gfde2a94e.zip
01522 kB / 01522 kB (100%) e3xx_e310_sg3_fpga_default-gfde2a94e.zip
09070 kB / 09070 kB (100%) e3xx_e320_fpga_default-gfde2a94e.zip
23071 kB / 23071 kB (100%) n3xx_n310_fpga_default-gfde2a94e.zip
16072 kB / 16072 kB (100%) n3xx_n300_fpga_default-gfde2a94e.zip
24996 kB / 24996 kB (100%) n3xx_n320_fpga_default-gfde2a94e.zip
00479 kB / 00479 kB (100%) b2xx_b200_fpga_default-gfde2a94e.zip
00464 kB / 00464 kB (100%) b2xx_b200mini_fpga_default-gfde2a94e.zip
00879 kB / 00879 kB (100%) b2xx_b210_fpga_default-gfde2a94e.zip
00523 kB / 00523 kB (100%) b2xx_b205mini_fpga_default-gfde2a94e.zip
00162 kB / 00162 kB (100%) b2xx_common_fw_default-g2bdad498.zip
00007 kB / 00007 kB (100%) usrp2_usrp2_fw_default-g6bea23d.zip
00450 kB / 00450 kB (100%) usrp2_usrp2_fpga_default-g6bea23d.zip
02415 kB / 02415 kB (100%) usrp2_n200_fpga_default-g6bea23d.zip
00009 kB / 00009 kB (100%) usrp2_n200_fw_default-g6bea23d.zip
02757 kB / 02757 kB (100%) usrp2_n210_fpga_default-g6bea23d.zip
00009 kB / 00009 kB (100%) usrp2_n210_fw_default-g6bea23d.zip
```

Step4-1 初始化(查看有無USB裝置)

輸入dmesg --follow並將USRP B210插入RPi上的USB3.0插槽，這時應會在畫面中看到Kernel偵測到有新的USB裝置，請記下idVendor及idProduct以供後續使用，本範例中idVendor為2500，idProduct為0020

```
[ 1007.063404] usb 1-1.1: new high-speed USB device number 8 using xhci_hcd
[ 1007.163854] usb 1-1.1: New USB device found, idVendor=2500, idProduct=0020, bcdDevice= 1.00
[ 1007.163861] usb 1-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 1007.163866] usb 1-1.1: Product: WestBridge
[ 1007.163870] usb 1-1.1: Manufacturer: Cypress
[ 1007.163874] usb 1-1.1: SerialNumber: 0000000004BE
[ 1074.323129] usb 1-1.1: USB disconnect, device number 8
[ 1093.936803] usb 1-1.1: new high-speed USB device number 9 using xhci_hcd
[ 1094.037285] usb 1-1.1: New USB device found, idVendor=2500, idProduct=0020, bcdDevice= 1.00
[ 1094.037292] usb 1-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 1094.037297] usb 1-1.1: Product: WestBridge
[ 1094.037301] usb 1-1.1: Manufacturer: Cypress
[ 1094.037305] usb 1-1.1: SerialNumber: 0000000004BE
```

Step4-2 初始化(查看新USB裝置)

輸入sudo lsusb以查看目前系統中的USB裝置，可以看到剛剛偵測到的裝置2500:0020，此裝置即為我們剛剛接上的USRP B210，但是因板子上的FPGA尚未初始化，因此會被識別為USB2.0 Hub，此為正常現象

```
ubuntu@pi4ue:~$ sudo lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 009: ID 2500:0020  USB2.0 Hub
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Step4-3 初始化(uhd_usrp_probe)

輸入sudo uhd_usrp_probe以進行USRP的硬體偵測，這隻程式會去掃描系統目前所有的USB裝置並初始化所有USRP硬體，可以看到剛剛接上的USRP B210已經順利被偵測到。此時若有外接電源的話USRP B210上的電源指示燈應由熄滅轉為紅色

```
ubuntu@pi4epc:~$ sudo uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200228; Boost_107100; UHD_3.15.0.0-2build5
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
```

Step4-4 初始化(再次查看USB裝置)

再次輸入sudo lsusb即可看到剛剛被識別為USB2.0 Hub的2500:0020已經順利的被辨識為USRP的硬體，說明初始化成功

```
ubuntu@pi4ue:~$ sudo lsusb
Bus 002 Device 002: ID 2500:0020 Ettus Research LLC USRP B200
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Step4-5 初始化(查看USB運作模式)

輸入sudo lsusb -d 2500:0020 -v以查看該USB裝置的詳細運作資訊，請確認bcdUSB為3.0，此為該裝置目前的運作模式，顯示USRP B210目前運作模式為USB3.0

```
Bus 002 Device 002: ID 2500:0020 Ettus Research LLC USRP B200
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               3.00
  bDeviceClass          255 Vendor Specific Class
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0         9
  idVendor           0x2500
  idProduct          0x0020
  bcdDevice            0.00
  iManufacturer           1 Ettus Research LLC
  iProduct                2 USRP B200
```

Step5-1 安裝srsGUI(下載)

輸入 git clone https://github.com/srsLTE/srsGUI.git && cd srsGUI 以下載srsGUI之原始碼

```
ubuntu@pi4ue2:~$ git clone https://github.com/srsLTE/srsGUI.git
Cloning into 'srsGUI'...
remote: Enumerating objects: 48, done.
remote: Counting objects: 100% (48/48), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 333 (delta 18), reused 17 (delta 7), pack-reused 285
Receiving objects: 100% (333/333), 91.89 KiB | 192.00 KiB/s, done.
Resolving deltas: 100% (164/164), done.
ubuntu@pi4ue2:~$ cd srsGUI/
ubuntu@pi4ue2:~/srsGUI$
```

Step5-2 安裝srsGUI(cmake)

輸入mkdir build && cd build && cmake ../以使用cmake產生Makefile，請確認cmake的過程中沒有出現任何錯誤且Makefile正確被產生

```
ubuntu@pi4ue2:~/srsGUI$ mkdir build
ubuntu@pi4ue2:~/srsGUI$ cd build/
ubuntu@pi4ue2:~/srsGUI/build$ cmake ../
-- The C compiler identification is GNU 9.3.0
-- The CXX compiler identification is GNU 9.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
--
-- Configuring Boost C++ Libraries...
-- Found Boost: /usr/lib/aarch64-linux-gnu/cmake/Boost-1.71.0/BoostConfig.cmake (found suitable version "1.71.0", minimum required is "1.37")

-- Boost version: 1.71.0
-- Using Boost Filesystem V2
-- Boost include directories: /usr/include
-- Boost library directories: /usr/lib/aarch64-linux-gnu
-- Boost libraries: Boost::thread;Boost::unit_test_framework;Boost::system
CMake Warning (dev) at cmake/modules/FindQwt.cmake:104 (SET):
  implicitly converting 'DOC' to 'STRING' type.
Call Stack (most recent call first):
  src/CMakeLists.txt:7 (FIND_PACKAGE)
This warning is for project developers.  Use -Wno-dev to suppress it.

-- Found Qwt: /usr/lib/libqwt-qt5.so (found version "6.1.4")
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ubuntu/srsGUI/build
```

Step5-3 安裝srsGUI(make)

輸入make -j4以使用剛剛產生Makefile進行編譯，請仔細注意編譯過程中有沒有出現任何錯誤，若編譯過程順利應可在make結果的最下方看到如下圖所示的字樣

```
[ 96%] Linking CXX executable realplot_test
[ 97%] Linking CXX executable complexplot_test
[ 98%] Linking CXX executable scatterplot_test
[100%] Linking CXX executable waterfallplot_test
[100%] Built target realplot_test
[100%] Built target complexplot_test
[100%] Built target scatterplot_test
[100%] Built target waterfallplot_test
```

Step5-3 安裝srsGUI(make install)

輸入make install以將剛剛編譯好的srsGUI安裝至系統中，成功的安裝過程如下圖所示

```
Install the project...
-- Install configuration: ""
-- Up-to-date: /usr/local/include/srsgui
-- Installing: /usr/local/include/srsgui/srsgui++.h
-- Up-to-date: /usr/local/include/srsgui/plot
-- Installing: /usr/local/include/srsgui/plot/Scatterplot.h
-- Installing: /usr/local/include/srsgui/plot/Waterfallplot.h
-- Installing: /usr/local/include/srsgui/plot/plot_waterfall.h
-- Installing: /usr/local/include/srsgui/plot/plot_complex.h
-- Installing: /usr/local/include/srsgui/plot/text_edit.h
-- Installing: /usr/local/include/srsgui/plot/plot_real.h
-- Installing: /usr/local/include/srsgui/plot/plot_scatter.h
-- Installing: /usr/local/include/srsgui/plot/KeyValue.h
-- Installing: /usr/local/include/srsgui/plot/TextEdit.h
-- Installing: /usr/local/include/srsgui/plot/key_value.h
-- Installing: /usr/local/include/srsgui/plot/Realplot.h
-- Installing: /usr/local/include/srsgui/plot/Complexplot.h
-- Installing: /usr/local/include/srsgui/srsgui.h
-- Up-to-date: /usr/local/include/srsgui/common
-- Installing: /usr/local/include/srsgui/common/Lineplot.h
-- Installing: /usr/local/include/srsgui/common/Events.h
-- Installing: /usr/local/include/srsgui/common/Pointplot.h
-- Installing: /usr/local/include/srsgui/common/WaterfallData.h
-- Installing: /usr/local/include/srsgui/common/Spectrogramplot.h
-- Installing: /usr/local/lib/libsrsgui.so
```

Step6 安裝srsLTE

請參考本課程Lab1的投影片編譯並安裝srsLTE，須注意在cmake的過程中是否成功偵測到UHD及srsGUI，成功偵測到的cmake畫面如下圖所示

```
-- UHD LIBRARIES /usr/lib/aarch64-linux-gnu/libuhd.so
-- UHD INCLUDE DIRS /usr/include
-- Found UHD: /usr/lib/aarch64-linux-gnu/libuhd.so
-- Checking for module 'libbladeRF'
--   No package 'libbladeRF' found
-- libbladeRF not found.
-- FINDING SOAPY.
-- Checking for module 'SoapySDR'
--   No package 'SoapySDR' found
-- libSOAPYSDR not found.
-- FINDING ZEROMQ.
-- Checking for module 'ZeroMQ'
--   No package 'ZeroMQ' found
-- Found libZEROMQ: /usr/include, /usr/lib/aarch64-linux-gnu/libzmq.so
-- Found Boost: /usr/lib/aarch64-linux-gnu/cmake/Boost-1.71.0/BoostConfi
-- SRSGUI LIBRARIES /usr/local/lib/libsrsgui.so
-- SRSGUI INCLUDE DIRS /usr/local/include
-- Found SRSGUI: /usr/local/lib/libsrsgui.so
```

Stage 2 Check List

	項目	內容
	UHD & FPGA Image	確認UHD已經正確被安裝且成功下載FPGA的映像檔
	USRP B210	確認USRP B210已經成功被偵測到且運作模式為USB3.0，若有外接電源的話則電源指示燈應為紅色
	srsGUI	確認srsGUI已經正確被編譯且成功安裝至系統內
	srsLTE	確認srsGUI已經正確被編譯且成功安裝至系統內，且cmake的過程中有成功偵測到UHD及srsGUI

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝
- Stage 3. srsLTE設定及量測
  - Step1 設定eNB
  - Step2 設定UE
  - Step3 執行EPC
  - Step4 執行eNB
  - Step5 執行UE並測試
  - Step6 原始物理通道觀察
  - Step7 測量流通量
- Stage 4. srsLTE參數調整
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
- 總結及問題

Step1-1 設定eNB(enb,gui section)

在eNB上輸入sudo vim /etc/srslte/enb.conf以編輯srsenb的設定檔，將[enb]範圍的設定值改為和下圖一樣的內容，其中n_prb為同一時間使用多少個Physical Resource Block進行資料傳輸，而tm為eNB的Transmission Mode。同時請將[gui]區塊下的enable設為true以啟動srsGUI

```
[enb]
enb_id = 0x19B
mcc = 001
mnc = 01
mme_addr = 127.0.1.100
gtp_bind_addr = 127.0.1.1
s1c_bind_addr = 127.0.1.1
n_prb = 15
tm = 1
nof_ports = 1
```

```
[gui]
enable = true
```

Step1-2 設定eNB(rf section)

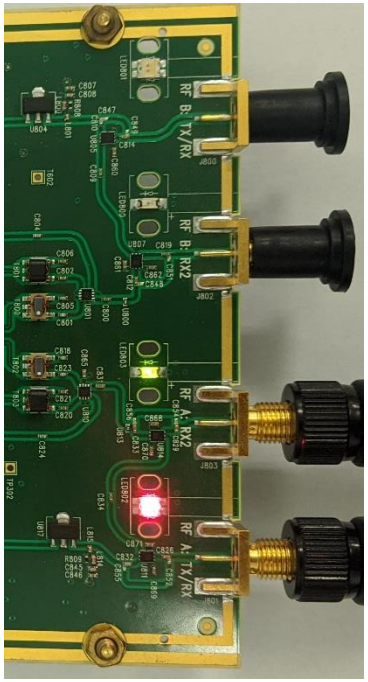
接續前一頁的eNB設定檔，將設定檔內[rf]區塊的設定值改為如下圖所示內容，其中dl_earfcn可以透過查表得知Downlink的頻率範圍，而tx_gain及rx_gain分別為傳送及接收的無線電訊號增益，若是使用同軸電纜連接eNB及UE的話請注意tx_gain盡量不要超過60dB，否則可能會損壞板子

```
[rf]
dl_earfcn = 3400
tx_gain = 80
rx_gain = 40

device_name = UHD

# For best performance in 2x2 MIMO and >= 15 MHz use the following device_args settings:
#     USRP B210: num_recv_frames=64,num_send_frames=64

# For best performance when BW<5 MHz (25 PRB), use the following device_args settings:
#     USRP B210: send_frame_size=512,recv_frame_size=512

device_args = auto
#time_adv_nsamples = auto

# Example for ZMQ-based operation with TCP transport for I/Q samples
#device_name = zmq
#device_args = fail_on_disconnect=true,tx_port=tcp://*:2000,rx_port=tcp://10.0.0.2:2001,id=enb,base_srate=23.04e6
```

Step2-1 設定UE(rf section)

在UE1及UE2上輸入sudo vim /etc/srslte/ue.conf以編輯UE的設定檔，將[rf]區塊中的設定值改為如下圖所示的內容，其中dl_earfcn需與eNB一致。若天線接在A:B的話請將device_args改為 tx_subdev_spec=A:B,rx_subdev_spec=A:B

```
[rf]
dl_earfcn = 3400
freq_offset = 0
tx_gain = 60
rx_gain = 40

#nof_carriers = 1
nof_antennas = 1

# For best performance in 2x2 MIMO and >= 15 MHz use the following dev
#     USRP B210: num_recv_frames=64,num_send_frames=64

# For best performance when BW<5 MHz (25 PRB), use the following devic
#     USRP B210: send_frame_size=512,recv_frame_size=512

device_name = "UHD"
device_args = tx_subdev_spec=A:A,rx_subdev_spec=A:A
#time_adv_nsamples = auto
#continuous_tx     = auto

# Example for ZMQ-based operation with TCP transport for I/Q samples
#device_name = zmq
#device_args = tx_port=tcp://*:2001,rx_port=tcp://localhost:2000,id=ue
```



A:B Tx

A:B Rx

A:A Rx

A:A Tx

Step2-2 設定UE(usim, gui section)

接續前述的UE設定檔，請將UE1的[usim]內容改為如下圖左所示，而UE2的[usim]區塊內容請改為下圖右的數值，其中的參數需與EPC中的user_db.csv一致。同時請將[gui]區塊下的enable設為true以啟動srsGUI

```
[usim]
mode = soft
algo = xor
#opc  = 63BFA50EE6523365FF14C1F45F88737D
k    = 00112233445566778899aabbccddeeff
imsi = 001010123456789
imei = 353490069873319
#reader =
#pin  = 1234
```

UE1設定檔內容

```
[usim]
mode = soft
algo = milenage
opc  = 63BFA50EE6523365FF14C1F45F88737D
k    = 00112233445566778899aabbccddeeff
imsi = 001010123456780
imei = 353490069873320
#reader =
#pin  = 1234
```

UE2設定檔內容

Step3-1 執行EPC

在EPC上輸入sudo srsepc以啟動EPC，成功畫面應如下圖所示

```
user@labepc:~$ sudo srsepc

Built in Release mode using commit c892ae56b on branch master.


---  Software Radio Systems EPC  ---

Reading configuration file /etc/srslte/epc.conf...
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf001, MNC: 0xff01
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
```

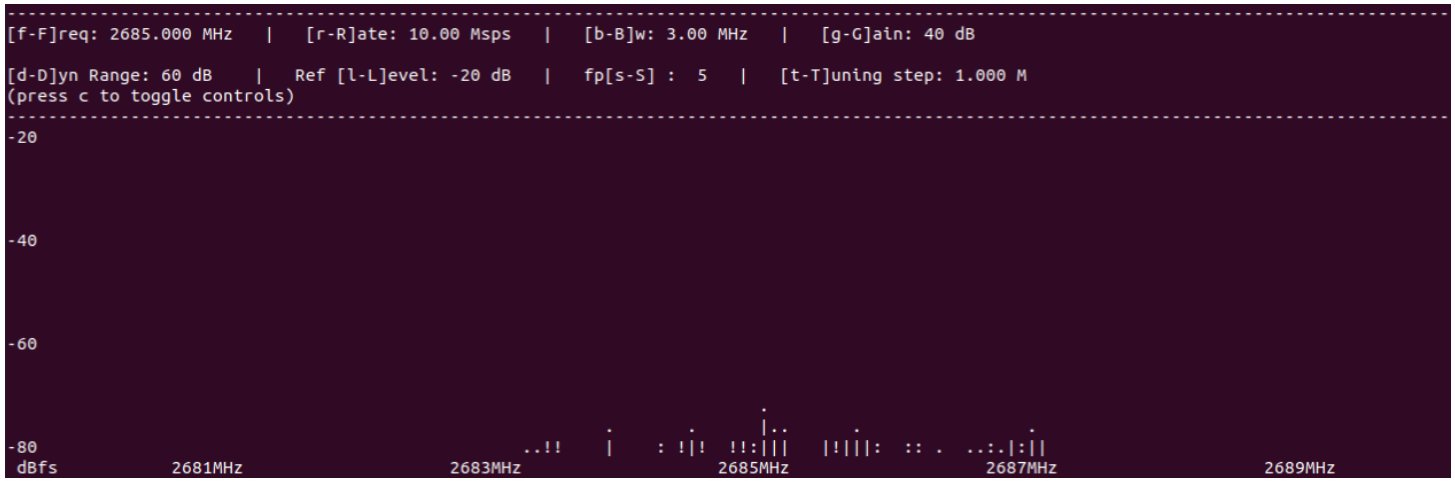
Step3-2 執行EPC(轉送設定)

在EPC上參考Lab1的投影片或是下圖的內容輸入指令以設定EPC的封包轉送，可使用sudo iptables -t nat -L POSTROUTING -v 來檢查所設定之規則是否正確，其中ens2s0為本範例中EPC連接到外部網路的介面名稱

```
user@labepc:~$ echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
1
user@labepc:~$ sudo iptables -L POSTROUTING -t nat -v
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
user@labepc:~$ sudo iptables -A POSTROUTING -t nat -s 172.16.0.0/24 -o enp2s0 -j MASQUERADE
user@labepc:~$ sudo iptables -L POSTROUTING -t nat -v
Chain POSTROUTING (policy ACCEPT 2 packets, 119 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  all  --  any    enp2s0  172.16.0.0/24        anywhere
user@labepc:~$
```

Step4-1 執行eNB(確認頻道)

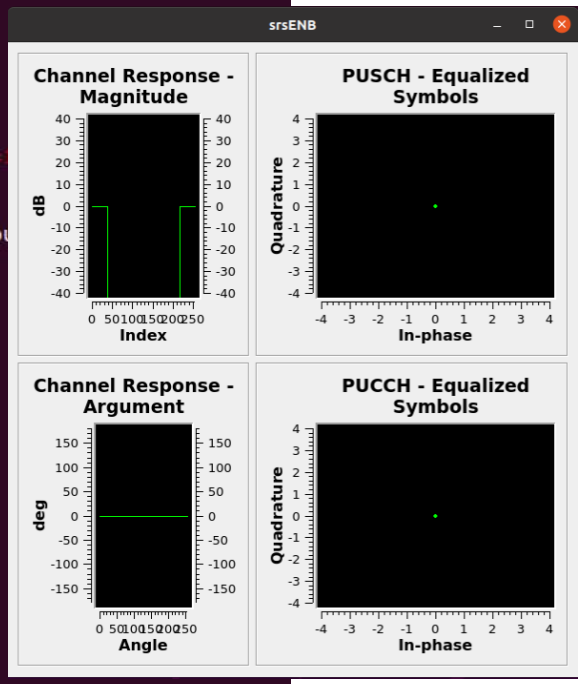
在 eNB 上 輸 入 /lib/uhd/examples/rx_ascii_art_dft --freq 2685e6 --rate 10e6 --gain 40 --bw 3e6 --ref-lvl -20 --subdev A:A以檢查欲使用之無線電頻道是否已經有人使用，其中--freq為檢視的中心頻率，由dl_earfcn查表得知。若顯示的無線電訊號均在-70dB以下即可使用該頻道，否則須更換eNB及UE的dl_earfcn



Step4-2 執行eNB(結果)

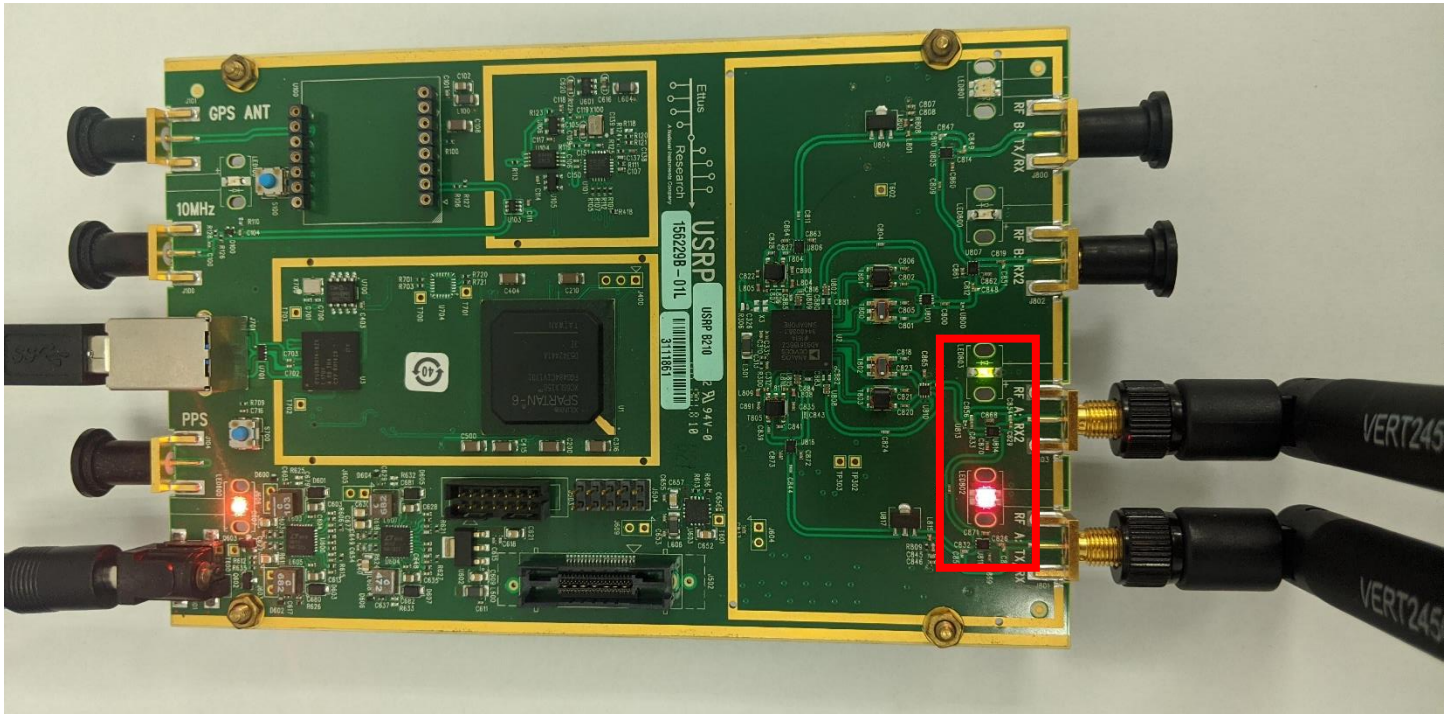
在eNB上輸入sudo srsenb啟動eNB後應可看到eNB上顯示如下圖左的內容，同時會啟動srsGUI顯示如下圖右的畫面

```
user@labepc:~$ sudo srsenb
---  Software Radio Systems LTE eNodeB  ---

Reading configuration file /etc/srslte/enb.conf...

Built in Release mode using commit c892ae56b on branch master.

/home/user/srsLTE/srsenb/src/enb_cfg_parser.cc.883: Force DL EARFCN for cell PCI=

Opening 1 channels in RF device=UHD with args=default
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200304; Boost_107100; UHD_3.15.0.0-2bu
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP channels=1, args: type=b200,master_clock_rate=23.04e6
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 2.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Setting frequency: DL=2685.0 Mhz, UL=2565.0 MHz for cc_idx=0

==== eNodeB started ===
Type <t> to view trace
Starting plot for worker_id=0
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```



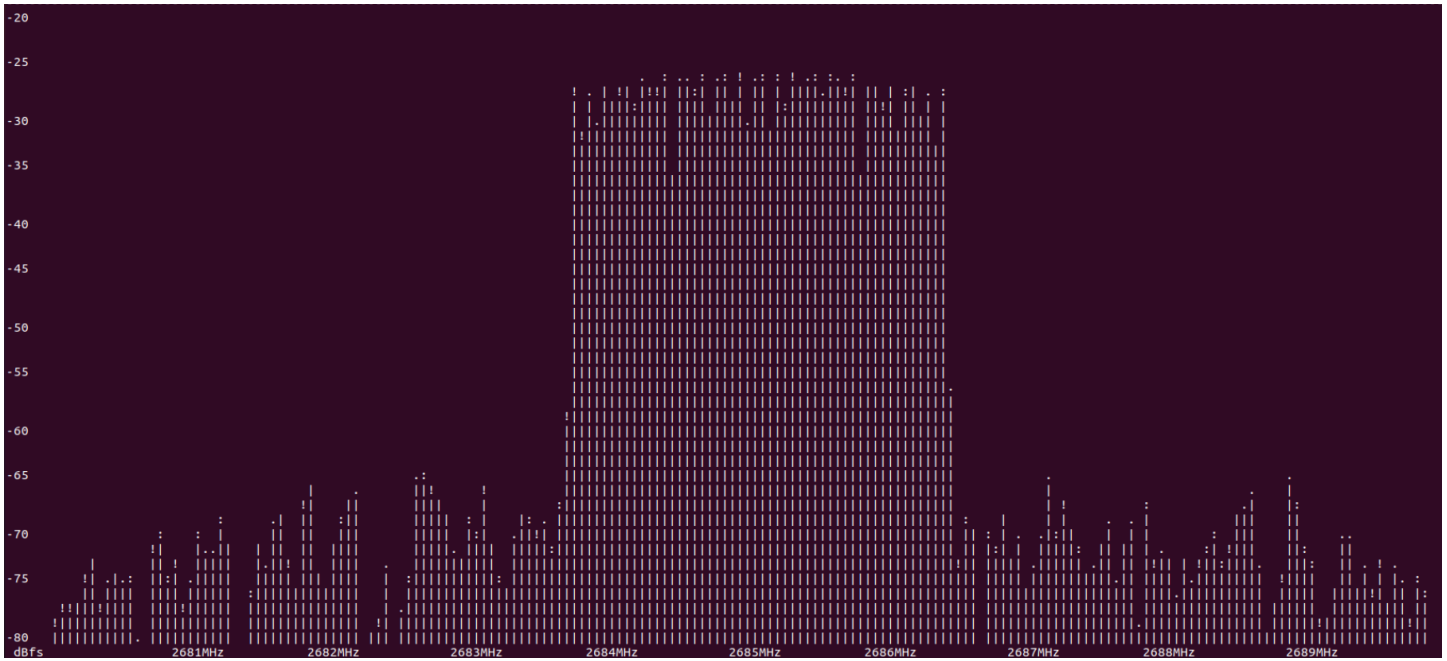
Step4-3 執行eNB(確認USRP狀態)

eNB啟動後連接在eNB上的USRP B210的狀態應會如下圖所示，因我們設定的sub_dev為A:A，因此A:A上的Rx及Tx指示燈會亮起，綠燈表示Rx，紅燈表示Tx



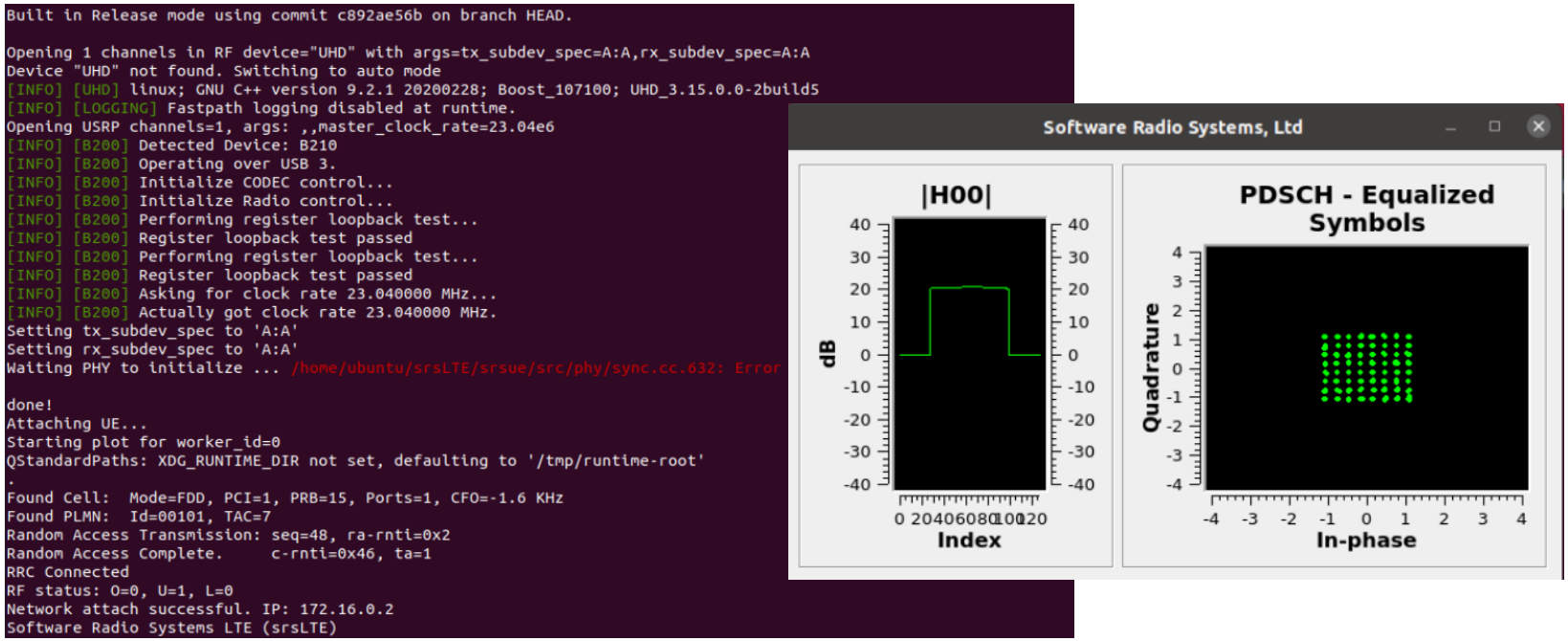
Step5-1 執行UE並測試(載波偵測)

在UE1或是UE2執行/lib/uhd/examples/rx_ascii_art_dft --freq 2685e6 --rate 10e6 --gain 40 --bw 3e6 --ref-lvl -20 --subdev A:B，其中--freq為由dl_earfcn推算出來的Downlink中心頻率，subdev為接收天線所接的位置，若有順利收到載波則頻譜儀應顯示如下圖所示之畫面



Step5-2 執行UE並測試(srsue)

在UE1和UE2分別執行sudo srsue，成功執行的話兩個UE理應順利連接上eNB及EPC並啟動一個srsGUI顯示PDSCH的symbol狀態，成功連接的畫面如下圖所示



UE連接上eNB的訊息

每當一個UE Attach上eNB時，eNB應該都會顯示一個 User 0xXX connected 的訊息，而當UE和eNB的無線連接中斷時會顯示 Disconnecting rnti=0xXX 的訊息

```
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200304; Boost_107100; UHD_3.15.0.0-2build5
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP channels=1, args: type=b200,master_clock_rate=23.04e6
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 2.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Setting frequency: DL=2685.0 Mhz, UL=2565.0 MHz for cc_idx=0

==== eNodeB started ===
Type <t> to view trace
RACH:  tti=8361, preamble=39, offset=3, temp_crnti=0x46
Disconnecting rnti=0x46.
RACH:  tti=3861, preamble=27, offset=1, temp_crnti=0x47
User 0x47 connected
RACH:  tti=3921, preamble=6, offset=3, temp_crnti=0x48
Disconnecting rnti=0x48.
RACH:  tti=9461, preamble=33, offset=1, temp_crnti=0x49
User 0x49 connected
```

UE連接上EPC時的訊息

每當一個UE Attach上EPC時，SPGW都會分配給該UE一個GTP通道的TEID以及IP位址，請驗證EPC所顯示的IMSI與UE端所設定的IMSI是否一致，若一致即為UE成功Attach上EPC的證明

```
SPGW: Allocated Ctrl TEID 1
SPGW: Allocated User TEID 1
SPGW: Allocate UE IP 172.16.0.2
Received Create Session Response
Create Session Response -- SPGW control TEID 1
Create Session Response -- SPGW S1-U Address: 127.0.1.100
SPGW Allocated IP 172.16.0.2 to IMSI 001010123456789
Adding attach accept to Initial Context Setup Request
Sent Initial Context Setup Request. E-RAB id 5
Received Initial Context Setup Response
E-RAB Context Setup. E-RAB id 5
E-RAB Context -- eNB TEID 0x460003; eNB GTP-U Address 127.0.1.1
UL NAS: Received Attach Complete
Unpacked Attached Complete Message. IMSI 1010123456789
```

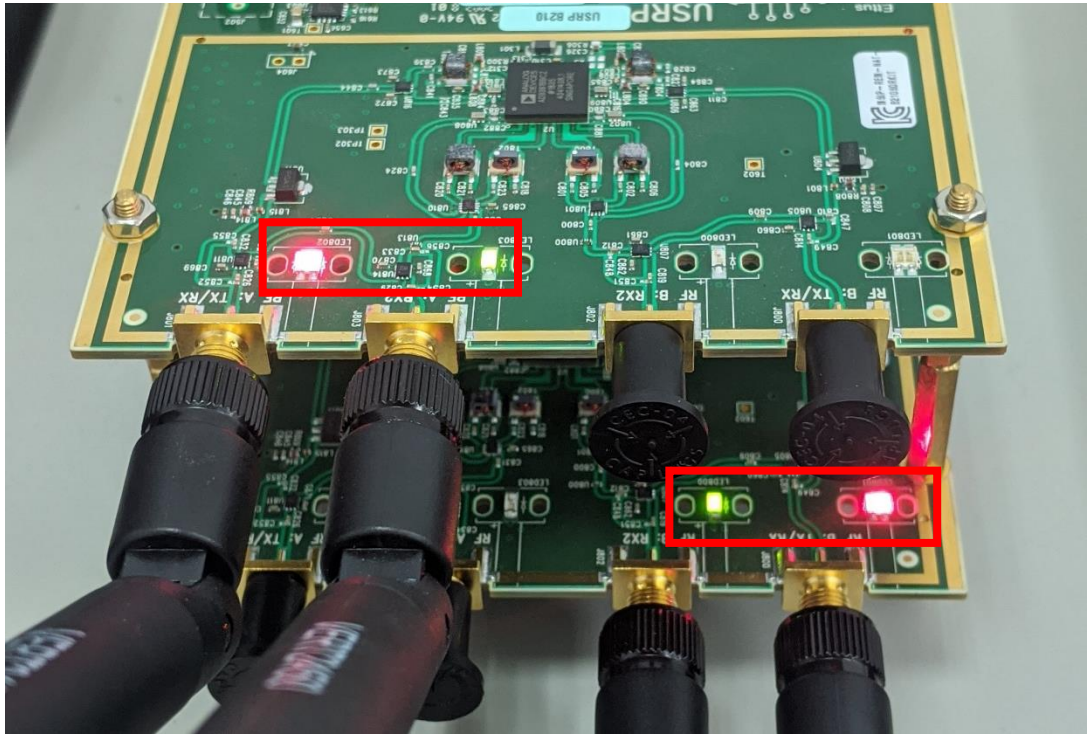
UE1

```
SPGW: Allocated Ctrl TEID 2
SPGW: Allocated User TEID 2
SPGW: Allocate UE IP 172.16.0.3
Received Create Session Response
Create Session Response -- SPGW control TEID 2
Create Session Response -- SPGW S1-U Address: 127.0.1.100
SPGW Allocated IP 172.16.0.3 to IMSI 001010123456780
Adding attach accept to Initial Context Setup Request
Sent Initial Context Setup Request. E-RAB id 5
Received Initial Context Setup Response
E-RAB Context Setup. E-RAB id 5
E-RAB Context -- eNB TEID 0x490003; eNB GTP-U Address 127.0.1.1
UL NAS: Received Attach Complete
Unpacked Attached Complete Message. IMSI 1010123456780
```

UE2

Step5-3 執行UE並測試(USRP狀態)

當UE成功Attach上eNB後其燈號應如下圖所示。下圖上方的為本範例的UE1，天線連接在A:A的位置。下方的USRP B210為UE2，天線連接在A:B的位置



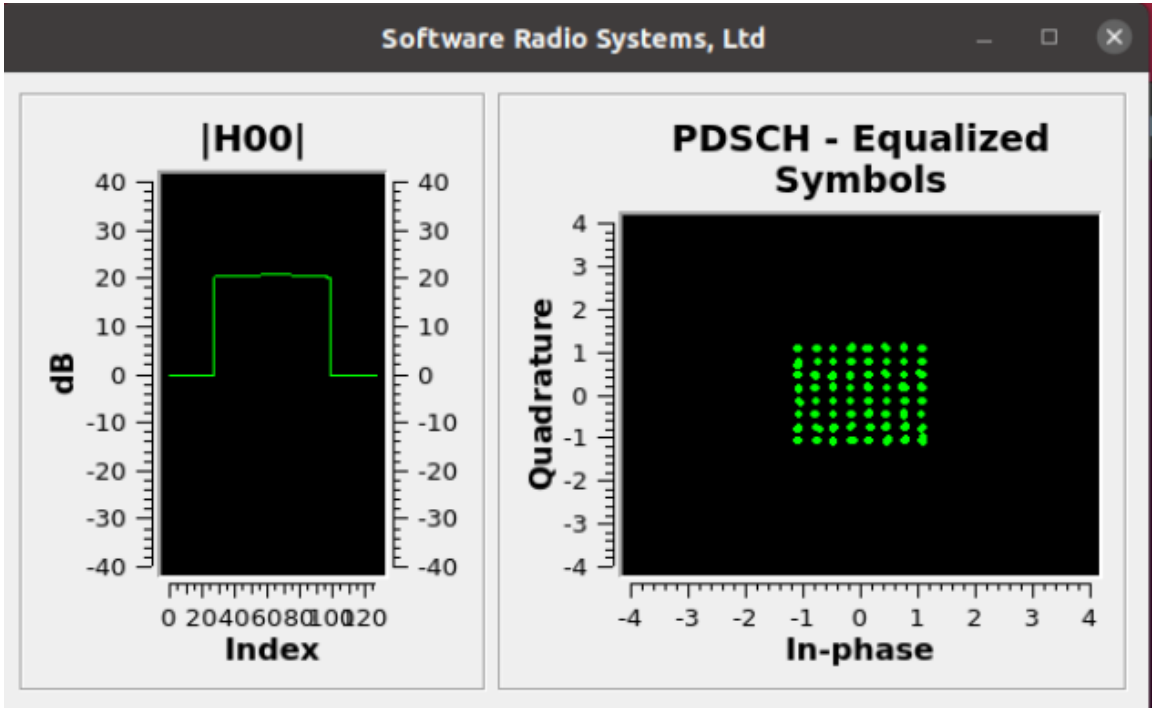
Step5-4 執行UE並測試(連通測試)

參考Lab1的投影片設定UE1及UE2的預設路由後在兩台裝置上執行mtr -tn 8.8.8.8，應可看到如下圖所示的畫面，證明兩台UE確實可以透過eNB及EPC連到Internet

```
                              My traceroute  [v0.93]
pi4ue2 (172.16.0.3)                                    2020-07-26T12:50:32+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                     Packets                 Pings
 Host                              Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 172.16.0.1                      0.0%     6   29.4  23.7  15.7  36.4   8.3
 2. 10.0.200.254                    0.0%     6   32.3  27.4  18.7  34.7   6.1
 3. 140.127.208.254                 0.0%     5   25.9  22.3  14.7  32.2   7.5
 4. 192.168.1.254                   0.0%     5   28.7  25.5  15.5  35.4   7.4
 5. 140.127.160.193                 0.0%     5   30.9  25.4  15.7  32.4   7.3
 6. 192.192.61.154                  0.0%     5   15.3  28.7  15.3  38.8  10.0
 7. 192.192.61.21                   0.0%     5   20.7  27.4  18.6  37.4   7.7
 8. 192.192.61.185                  0.0%     5   23.0  32.2  23.0  40.0   7.3
 9. 192.192.61.198                  0.0%     5   25.4  30.4  19.0  42.8   9.0
10. 72.14.196.229                   0.0%     5   28.0  32.8  17.2  43.5  10.4
11. 108.170.244.33                  0.0%     5   30.5  28.4  18.6  40.4   8.0
12. 209.85.254.233                  0.0%     5   31.3  34.4  29.4  40.8   4.6
13. 8.8.8.8                         0.0%     5   36.7  31.2  19.0  45.4  10.7
```

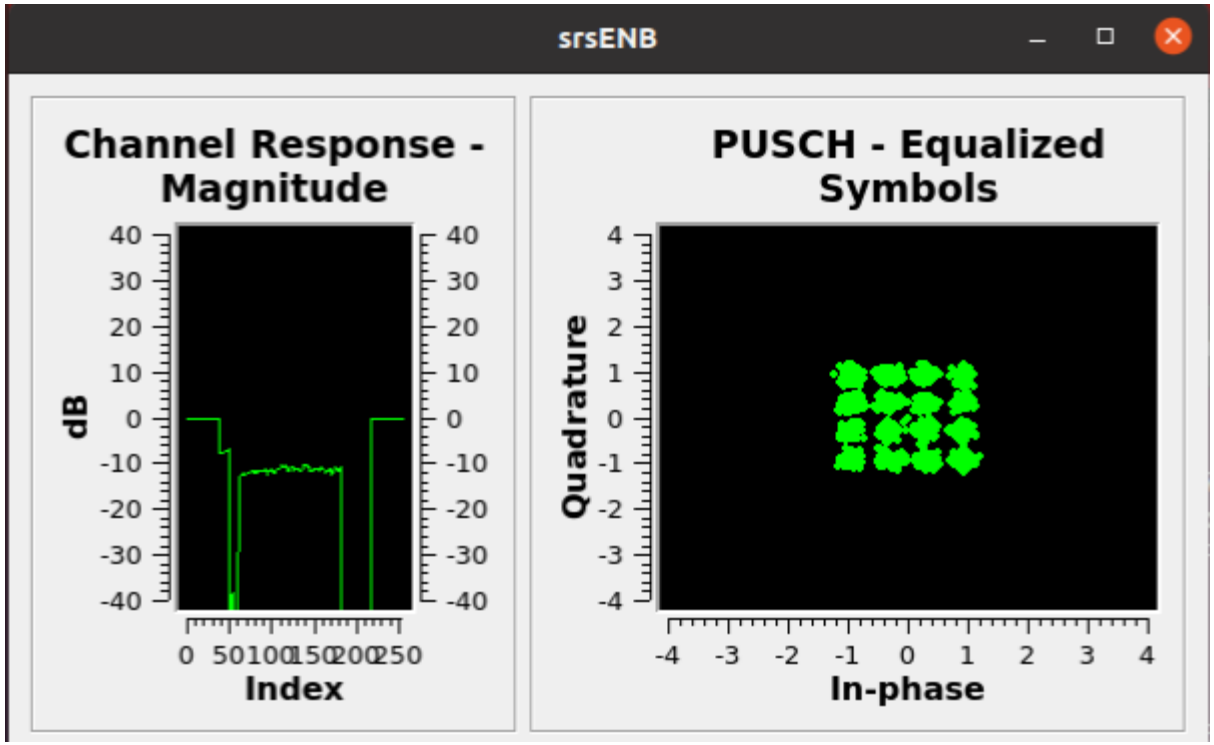
Step6-1 原始物理通道觀察(PDSCH)

下圖為UE上的srsGUI畫面，設定檔為前述投影片調整完後的設定。左邊為發射端到接收端的信道增益，右邊為繪製各個Symbol的I/Q平面。可以觀察到目前是以64QAM的方式進行調變



Step6-2 原始物理通道觀察(PUSCH)

下圖為eNB上的srsGUI部分畫面，設定檔為前述投影片調整完後的設定。右邊為無線信道對於不同頻率的增益，左邊為繪製PUSCH Symbol的I/Q平面。可以觀察到目前是以16QAM的方式進行調變



Step7 測量流通量

在EPC上執行iperf3 -s並在其中一個UE上執行iperf3 -c 172.16.0.1 -t 60 -i 10 -b 1m --bidir執行雙向流通量測試，其結果如下圖所示，Downlink流通量為450Kbps，Uplink流通量為657Kbps

```
Connecting to host 172.16.0.1, port 5201
[  5] local 172.16.0.16 port 51492 connected to 172.16.0.1 port 5201
[  7] local 172.16.0.16 port 51494 connected to 172.16.0.1 port 5201
[ ID][Role] Interval           Transfer     Bitrate         Retr  Cwnd
[  5][TX-C]   0.00-10.00  sec   795 KBytes   652 Kbits/sec   65   2.83 KBytes
[  7][RX-C]   0.00-10.00  sec   482 KBytes   395 Kbits/sec
[  5][TX-C]  10.00-20.00  sec   602 KBytes   493 Kbits/sec   60   2.83 KBytes
[  7][RX-C]  10.00-20.00  sec   725 KBytes   594 Kbits/sec
[  5][TX-C]  20.00-30.00  sec   370 KBytes   304 Kbits/sec   49   2.83 KBytes
[  7][RX-C]  20.00-30.00  sec   983 KBytes   805 Kbits/sec
[  5][TX-C]  30.00-40.00  sec   520 KBytes   426 Kbits/sec   58   1.41 KBytes
[  7][RX-C]  30.00-40.00  sec   853 KBytes   699 Kbits/sec
[  5][TX-C]  40.00-50.00  sec   525 KBytes   430 Kbits/sec   54   2.83 KBytes
[  7][RX-C]  40.00-50.00  sec   868 KBytes   711 Kbits/sec
[  5][TX-C]  50.00-60.00  sec   594 KBytes   487 Kbits/sec   59   4.24 KBytes
[  7][RX-C]  50.00-60.00  sec   901 KBytes   738 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID][Role] Interval           Transfer     Bitrate         Retr
[  5][TX-C]   0.00-60.00  sec  3.33 MBytes   465 Kbits/sec  345             sender
[  5][TX-C]   0.00-60.02  sec  3.22 MBytes   450 Kbits/sec                  receiver
[  7][RX-C]   0.00-60.00  sec  4.78 MBytes   669 Kbits/sec  337             sender
[  7][RX-C]   0.00-60.02  sec  4.70 MBytes   657 Kbits/sec                  receiver

iperf Done.
```

Stage 3 Check List

	項目	內容
	eNB設定	確認eNB設定與投影片一致
	UE設定	確認UE1和UE2設定與投影片一致，且UE1和UE2設定與EPC中的user_db.csv一致
	欲使用無線電頻段	確認欲使用之無線電頻段無人占用
	eNB無線電發射	在UE端使用頻譜分析儀確認是否能收到eNB所發射的無線電訊號
	UE連接	確認UE1和UE2都能正常與eNB及EPC連接
	網路連通	確認在UE上可以連接到Internet
	srsGUI	確認srsGUI會正常啟動並顯示對應圖形
	流通量	以iprf3確認Downlink與Uplink之流通量

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝
- Stage 3. srsLTE設定及量測
- Stage 4. srsLTE參數調整
  - Step1 調整RB
  - Step2 調整PDSCH_MAX_MCS
  - Step3 調整PUSCH_MAX_MCS
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
- 總結及問題

Step1 調整RB

在eNB輸入sudo vim /etc/srslte/enb.conf以編輯eNB的設定檔，將[enb]區塊下的n_prb由15改為6，更改後的設定檔如下圖所示。此改動會變更Resource Block的數量，連帶的會將系統的頻寬由3MHz降為1.4MHz

```
[enb]
enb_id = 0x19B
mcc = 001
mnc = 01
mme_addr = 127.0.1.100
gtp_bind_addr = 127.0.1.1
s1c_bind_addr = 127.0.1.1
n_prb = 6
tm = 1
nof_ports = 1
```

調整RB的影響

調整Resource Block的數量後重新執行eNB及UE，並使用iperf3重新測量一次流通量，其測量結果如下圖所示。Downlink流通量為601Kbps，為原始流通量的1.3倍。Uplink流通量為42.8Kbps，為原始流通量的0.063倍

```
Connecting to host 172.16.0.1, port 5201
[  5] local 172.16.0.17 port 41198 connected to 172.16.0.1 port 5201
[  7] local 172.16.0.17 port 41200 connected to 172.16.0.1 port 5201
[ ID][Role] Interval           Transfer     Bitrate         Retr  Cwnd
[  5][TX-C]   0.00-10.00  sec   836 KBytes   685 Kbits/sec   45   7.07 KBytes
[  7][RX-C]   0.00-10.00  sec  52.3 KBytes  42.9 Kbits/sec
[  5][TX-C]  10.00-20.00  sec   652 KBytes   534 Kbits/sec   35   5.66 KBytes
[  7][RX-C]  10.00-20.00  sec  91.9 KBytes  75.3 Kbits/sec
[  5][TX-C]  20.00-30.00  sec   751 KBytes   615 Kbits/sec   35   9.90 KBytes
[  7][RX-C]  20.00-30.00  sec  12.7 KBytes  10.4 Kbits/sec
[  5][TX-C]  30.00-40.00  sec   796 KBytes   652 Kbits/sec   27   9.90 KBytes
[  7][RX-C]  30.00-40.00  sec  48.1 KBytes  39.4 Kbits/sec
[  5][TX-C]  40.00-50.00  sec   850 KBytes   696 Kbits/sec   31   5.66 KBytes
[  7][RX-C]  40.00-50.00  sec  32.5 KBytes  26.6 Kbits/sec
[  5][TX-C]  50.00-60.00  sec   619 KBytes   508 Kbits/sec   36   4.24 KBytes
[  7][RX-C]  50.00-60.00  sec  76.4 KBytes  62.6 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID][Role] Interval           Transfer     Bitrate         Retr
[  5][TX-C]   0.00-60.00  sec  4.40 MBytes   615 Kbits/sec  209             sender
[  5][TX-C]   0.00-60.02  sec  4.30 MBytes   601 Kbits/sec                  receiver
[  7][RX-C]   0.00-60.00  sec   397 KBytes  54.3 Kbits/sec  178             sender
[  7][RX-C]   0.00-60.02  sec   314 KBytes  42.8 Kbits/sec                  receiver

iperf Done.
```

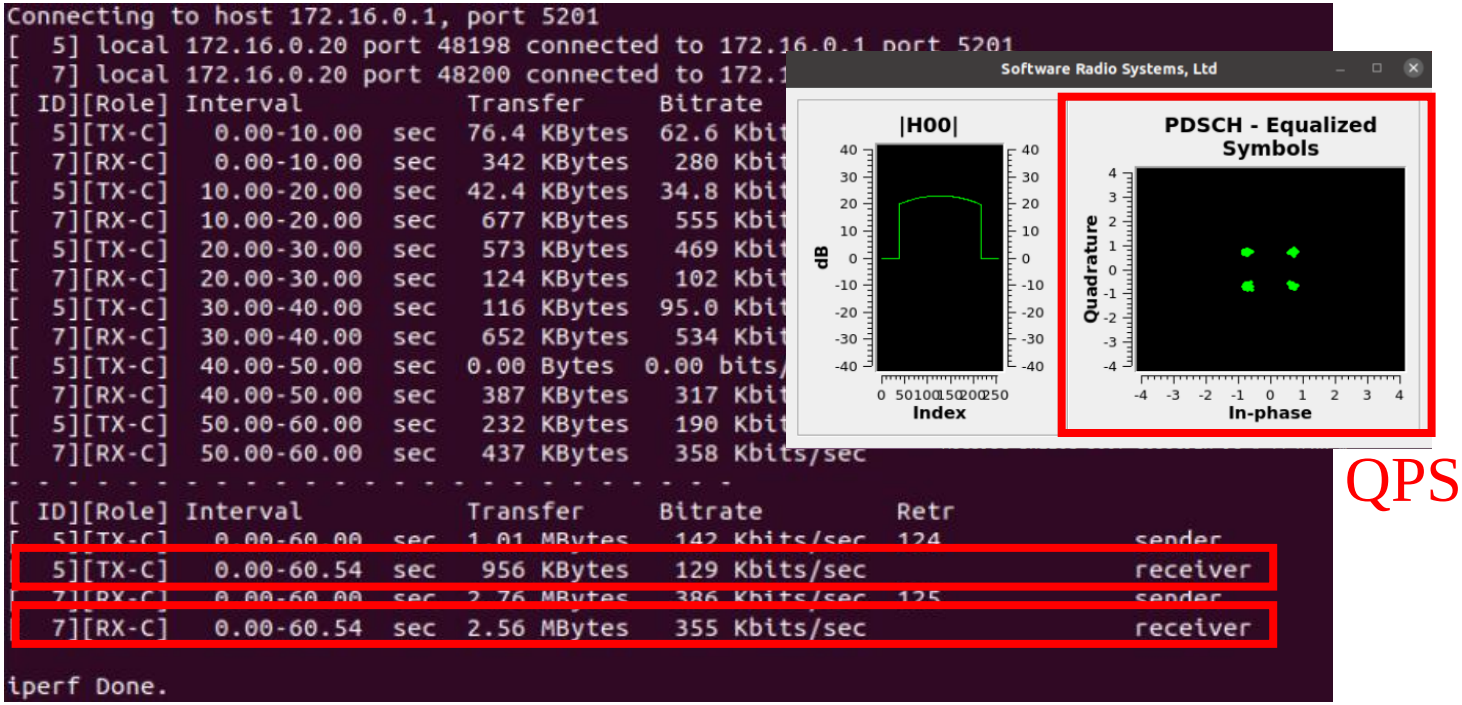
Step2 調整PDSCH_MAX_MCS

在eNB輸入sudo vim /etc/srslte/enb.conf以編輯eNB的設定檔，將[scheduler]區塊下的pdsch_max_mcs設定為5，其他設定值維持與Stage3相同，更改後的設定檔如下圖所示。此改動將PDSCH的Modulation Coding Scheme限制在QPSK

```
[scheduler]
#max_aggr_level    = -1
#pdsch_mcs         = -1
pdsch_max_mcs     = 5
#pusch_mcs         = -1
#pusch_max_mcs     = -1
#min_nof_ctrl_symbols = 2
#max_nof_ctrl_symbols = 2
```

調整PDSCH_MAX_MCS的影響

調整PDSCH_MAX_MCH後重新執行eNB及UE，並使用iperf3重新測量一次流通量，其測量結果如下圖所示。Downlink流通量為129Kbps，為原始流通量的0.29倍。Uplink流通量為355Kbps，為原始流通量的0.53倍



QPSK

Step3 調整PUSCH_MAX_MCS

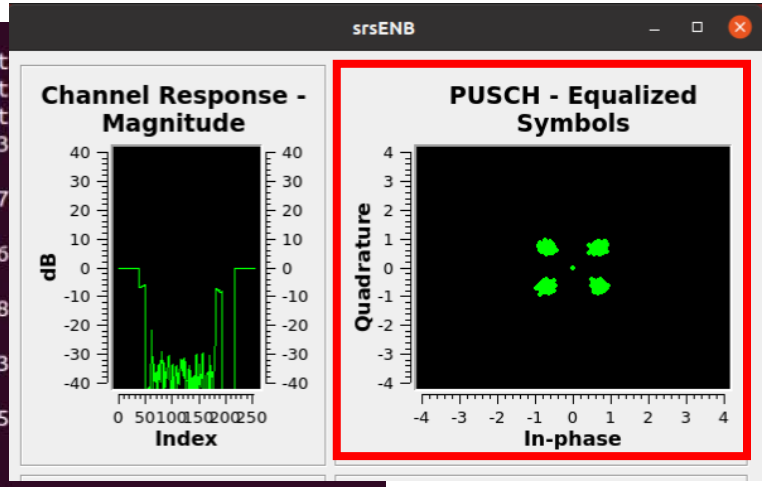
在eNB輸入sudo vim /etc/srslte/enb.conf以編輯eNB的設定檔，將[scheduler]區塊下的pusch_max_mcs設定為5，其他設定值維持與Stage3相同，更改後的設定檔如下圖所示。此改動將PUSCH的Modulation Coding Scheme限制在QPSK

```
[scheduler]
#max_aggr_level   = -1
#pdsch_mcs        = -1
#pdsch_max_mcs    = -1
#pusch_mcs        = -1
pusch_max_mcs    = 5
#min_nof_ctrl_symbols = 2
#max_nof_ctrl_symbols = 2
```

調整PUSCH_MAX_MCS的影響

調整PUSCH_MAX_MCH後重新執行eNB及UE，並使用iperf3重新測量一次流通量，其測量結果如下圖所示。Downlink流通量為243Kbps，為原始流通量的0.54倍。Uplink流通量為378Kbps，為原始流通量的0.56倍

```
Connecting to host 172.16.0.1, port 5201
[  5] local 172.16.0.21 port 43432 connected to 172.16.0.1 port
[  7] local 172.16.0.21 port 43434 connected to 172.16.0.1 port
[ ID][Role] Interval           Transfer     Bitrate         Ret
[  5][TX-C]   0.00-10.00  sec   571 KBytes   468 Kbits/sec   53
[  7][RX-C]   0.00-10.00  sec   221 KBytes   181 Kbits/sec
[  5][TX-C]  10.00-20.00  sec   448 KBytes   367 Kbits/sec   47
[  7][RX-C]  10.00-20.00  sec   132 KBytes   108 Kbits/sec
[  5][TX-C]  20.00-30.00  sec   373 KBytes   306 Kbits/sec   46
[  7][RX-C]  20.00-30.00  sec   332 KBytes   272 Kbits/sec
[  5][TX-C]  30.00-40.00  sec   291 KBytes   239 Kbits/sec   48
[  7][RX-C]  30.00-40.00  sec   608 KBytes   498 Kbits/sec
[  5][TX-C]  40.00-50.00  sec  73.5 KBytes  60.3 Kbits/sec   33
[  7][RX-C]  40.00-50.00  sec   850 KBytes   696 Kbits/sec
[  5][TX-C]  50.00-60.00  sec   151 KBytes   124 Kbits/sec   15
[  7][RX-C]  50.00-60.00  sec   629 KBytes   515 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID][Role] Interval           Transfer     Bitrate         Retr
[  5][TX-C]   0.00-60.00  sec  1.86 MBytes   261 Kbits/sec  242             sender
[  5][TX-C]   0.00-60.09  sec  1.74 MBytes   243 Kbits/sec                  receiver
[  7][RX-C]   0.00-60.09  sec  2.71 MBytes   378 Kbits/sec                  receiver

iperf Done.
```



QPSK

Stage 4 Check List

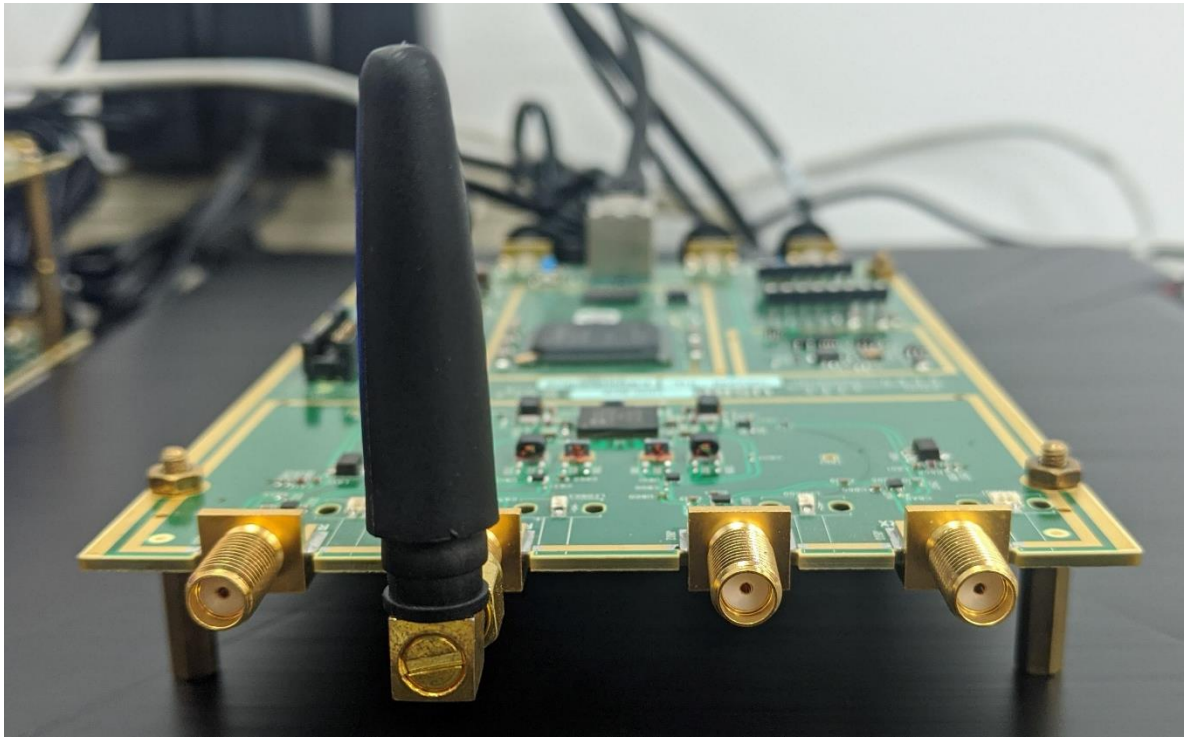
	項目	內容
	Resource Block	調整n_prb並測量其流通量，可使用頻譜分析儀觀測其頻寬有無變化
	PDSCH MAX MCS	調整pdsch_max_mcs並確認UE端的PDSCH調變方式有沒有改變，同時測量其流通量
	PUSCH MAX MCS	調整pusch_max_mcs並確認eNB端的PUSCH調變方式有沒有改變，同時測量其流通量

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝
- Stage 3. srsLTE設定及量測
- Stage 4. srsLTE參數調整
- Stage 5. NB-IoT
  - Step1 接上天線
  - Step2 搜尋附近的eNB
  - Step3 商用eNB的資訊
  - Step4 更換天線
  - Step5 NPDSCH eNB
  - Step6 NPDSCH UE
- Stage 6. mMTC 應用
- 總結及問題

Step1 接上天線

將其中一台USRP B210位於A:A位置上的Rx天線換成對應頻率為700MHz到900MHz左右的天線，更換天線會使得後面的搜尋步驟較容易成功



Step2 搜尋附近的eNB

在前一步驟的USRP所連接之裝置上先進入srsLTE的原始碼目錄，並執行./build/lib/examples/cell_search_nbiot -b 1，其中-b 1代表要搜尋的頻段，台灣目前NB-IoT可能出現的頻代為Band1、Band3、Band8及Band28，請依序搜尋這四個頻帶。若找到eNB的話會顯示如下圖的訊息

Band1

```
Found 2 cells
Found CELL 2110.9 MHz, EARFCN=9, PHYID=331, NPSS power=-1.4 dBm
Found CELL 2146.6 MHz, EARFCN=366, PHYID=313, NPSS power=-12.8 dBm
```

Band3

```
Found 3 cells
Found CELL 1816.1 MHz, EARFCN=1311, PHYID=444, NPSS power=-10.3 dBm
Found CELL 1849.0 MHz, EARFCN=1640, PHYID=26, NPSS power=-15.8 dBm
Found CELL 1856.1 MHz, EARFCN=1711, PHYID=467, NPSS power=-7.4 dBm
```

Band8

```
Found 2 cells
Found CELL 940.7 MHz, EARFCN=3607, PHYID=18, NPSS power=1.4 dBm
Found CELL 945.3 MHz, EARFCN=3653, PHYID=450, NPSS power=4.0 dBm
```

Band28

```
Found 2 cells
Found CELL 777.6 MHz, EARFCN=9406, PHYID=470, NPSS power=15.5 dBm
Found CELL 783.9 MHz, EARFCN=9469, PHYID=181, NPSS power=21.1 dBm
```

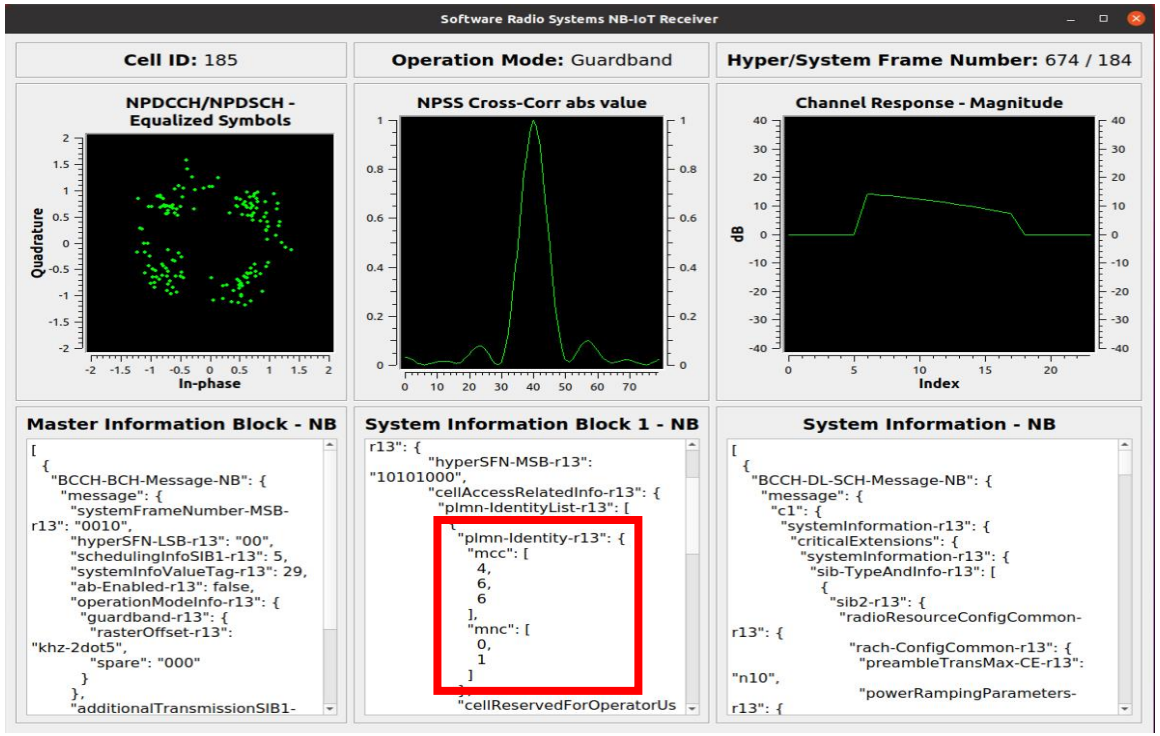
Step3-1 商用eNB的資訊(連接)

接下來請執行./build/lib/examples/npdsch_ue -f 783.9e6，其中 -f 783.9e6 為欲連接eNB的Downlink頻率，本範例為783.9MHz。由於前一步驟有能會找到非NB-IoT的eNB，因此若連接失敗的話請嘗試下一個所找到的eNB。若連接成功的話會出現如*Found n_id_ncell … 的字樣

```
Opening RF device...
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200228; Boost_107100; UHD_3.15.0.0-2
build5
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP channels=1, args: type=b200,master_clock_rate=23.04e6
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Set RX gain: 70.0 dB
Set RX freq: 783.900000 MHz
Setting sampling rate 1.92 MHz
NSSS with peak=39.214897, cell-id: 104, partial SFN: 0
*Found n_id_ncell: 104 DetectRatio= 0% PSR=6.13, Power=80.0 dBm
Finding PSS... Peak:      2.5, FrameCnt: 0, State: 1
```

Step3-2 商用eNB的資訊(srsGUI)

連接成功的同時會透過srsGUI顯示如下的資訊，可以看到這個eNB的MCC為466且MNC為01，代表這是屬於遠傳電信的eNB



Ref: https://www.ncc.gov.tw/chinese/opendata_item.aspx?menu_function_sn=163

Step3-3 商用基地台的資訊(MIB)

收到MIB及SIB後即可關閉UE並以Wireshark打開/tmp/npdsch.pcap以觀察詳細資訊，參考Lab2的投影片設定封包解析器後點選MIB即可看到如下的資訊，可以看到這個NB-IoT eNB的Operation Mode為Guard band

No.	Time	Source	Destinatio	Protocol	Length	Info
1	0.000000			LTE RR…	69	MasterInformationBlock-NB
2	1.431870			LTE RR…	74	SystemInformationBlockType1-NB
3	3.848528			LTE RR…	74	SystemInformation-NB [SIB2 SIB3]

```
▸ Frame 1: 69 bytes on wire (552 bits), 69 bytes captured (552 bits)
  DLT: 147, Payload: mac-lte-framed (mac-lte-framed)
▾ MAC-LTE BCH PDU (50 bytes, on BCH transport)
  ▸ [Context]
    [Transport channel: BCH (0)]
  ▾ LTE Radio Resource Control (RRC) protocol
    ▾ BCCH-BCH-Message-NB
      ▾ message
          systemFrameNumber-MSB-r13: 20 [bit length 4, 4 LSB pad bits, 0010 .... decimal value 2]
          hyperSFN-LSB-r13: 00 [bit length 2, 6 LSB pad bits, 00.. .... decimal value 0]
          schedulingInfoSIB1-r13: 16 NPDSCH repetitions - TBS 328 bits (5)
          systemInfoValueTag-r13: 29
                  0 ab-Enabled-r13: False
          ▾ operationModeInfo-r13: guardband-r13 (2)
            ▾ guardband-r13
                rasterOffset-r13: khz-2dot5 (1)
                spare: 00 [bit length 3, 5 LSB pad bits, 000. .... decimal value 0]
          .... ...0 additionalTransmissionSIB1-r15: False
          spare: 0000 [bit length 10, 6 LSB pad bits, 0000 0000  00.. .... decimal value 0]
```

Step3-4 商用基地台的資訊(SIB)

可以在SIB2中看到NPRACH、NPDSCH及NPUSCH的相關參數，如NPRACH的週期、用來估測信道品質的NRS參考訊號之發射強度...等

```
▾ sib-TypeAndInfo-r13 item: sib2-r13 (0)
  ▾ sib2-r13
    ▾ radioResourceConfigCommon-r13
      ▸ rach-ConfigCommon-r13
      ▸ bcch-Config-r13
      ▸ pcch-Config-r13
      ▾ nprach-Config-r13
          nprach-CP-Length-r13: us266dot7 (1)
        ▸ rsrp-ThresholdsPrachInfoList-r13: 2 items
        ▾ nprach-ParametersList-r13: 3 items
          ▾ Item 0
            ▾ NPRACH-Parameters-NB-r13
                nprach-Periodicity-r13: ms320 (4)
                nprach-StartTime-r13: ms8 (0)
                nprach-SubcarrierOffset-r13: n36 (3)
                nprach-NumSubcarriers-r13: n12 (0)
                nprach-SubcarrierMSG3-RangeStart-r13: twoThird (2)
                maxNumPreambleAttemptCE-r13: n10 (6)
                numRepetitionsPerPreambleAttempt-r13: n1 (0)
                npdcch-NumRepetitions-RA-r13: r8 (3)
                npdcch-StartSF-CSS-RA-r13: v2 (1)
                npdcch-Offset-RA-r13: zero (0)
          ▸ Item 1
          ▸ Item 2
```

```
▾ npdsch-ConfigCommon-r13
    nrs-Power-r13: 28dBm
▾ npusch-ConfigCommon-r13
  ▾ ack-NACK-NumRepetitions-Msg4-r13: 3 items
    ▾ Item 0
        ACK-NACK-NumRepetitions-NB-r13: r1 (0)
    ▾ Item 1
        ACK-NACK-NumRepetitions-NB-r13: r2 (1)
    ▾ Item 2
        ACK-NACK-NumRepetitions-NB-r13: r32 (5)
  ▾ dmrs-Config-r13
      threeTone-CyclicShift-r13: 0
      sixTone-CyclicShift-r13: 1
  ▾ ul-ReferenceSignalsNPUSCH-r13
      .... .0.. groupHoppingEnabled-r13: False
      groupAssignmentNPUSCH-r13: 0
```

Step4 更換天線

將USRP的天線換回原本的VERT2450，若有兩隻以上的GSM天線亦可使用GSM天線替代

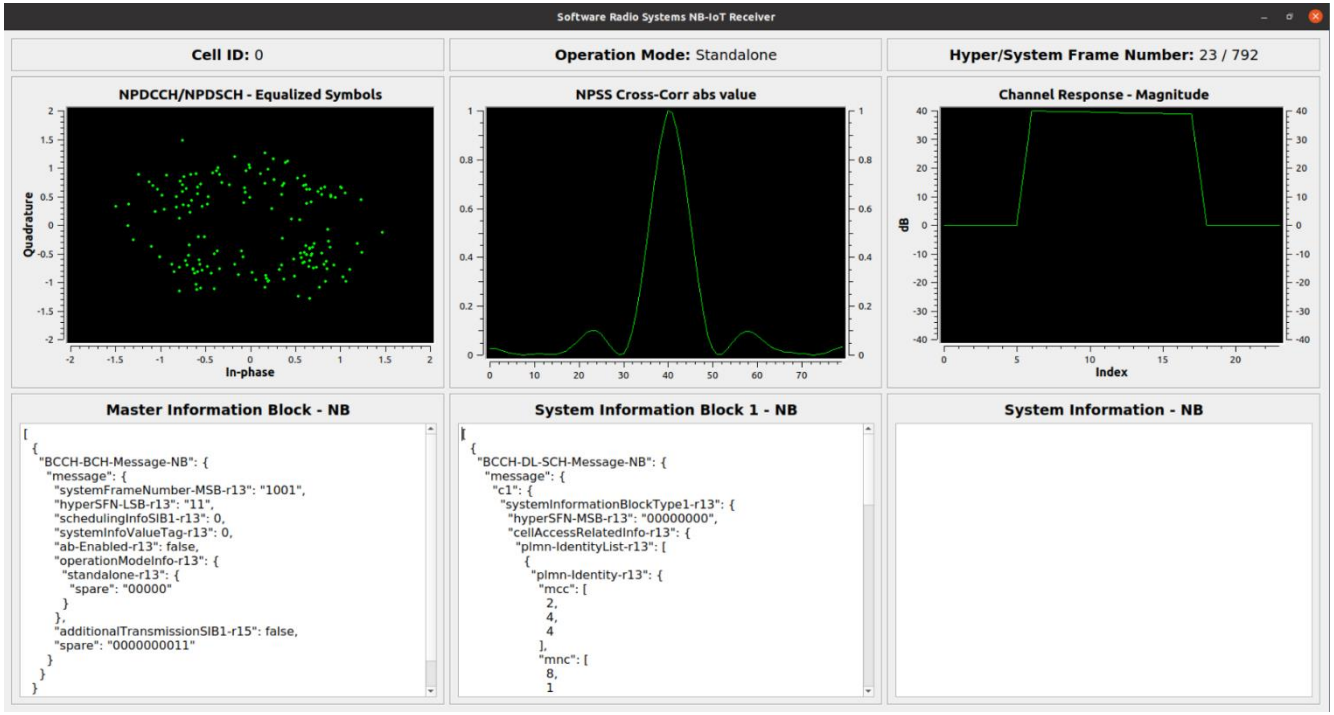
Step5 NPDSCH eNB

於 eNB 之 終 端 機 進 入 srsLTE 的 原 始 碼 目 錄 後 輸 入./build/lib/examples/npdsch_enodeb -f 2100e6以架設一個NB-IoT的eNB，其中-f 2100e6 為欲開設之頻率，可依自身環境選擇干擾較少的頻段，成功執行之畫面如下圖所示

```
Setting sampling rate 1.92 MHz
Set TX gain: 70.0 dB
Set TX freq: 2100.00 MHz
NB-IoT DL DCI:
 - Format flag:                          1
   + FormatN1 DCI:                       Downlink
 - PDCCH Order:                          0
 - Scheduling delay:                     0 (0 subframes)
 - Resource assignment:                  0
   + Number of subframes:                1
 - Modulation and coding scheme index:   1
 - Repetition number:                    0
   + Number of repetitions:              1
 - New data indicator:                   0
 - HARQ-ACK resource:                    1
 - DCI subframe repetition number:       0
DL grant config:
 - Number of subframes:                  1
 - Number of repetitions:                1
 - Total number of subframes:            1
 - Starting SFN:                         0
 - Starting SF index:                    6
 - Modulation type:                      QPSK
 - Transport block size:                 24
Type new MCS index and press Enter: NB-IoT HFN: 1
NB-IoT HFN: 2
NB-IoT HFN: 3
```

Step6 NPDSCH UE

於 UE 之 終 端 機 進 入 srsLTE 的 原 始 碼 目 錄 後 輸 入./build/lib/examples/npdsch_ue -f 2100e6以啟動NB-IoT的UE，其中-f 2100e6 為欲接收之頻率，需與前一步驟之頻率吻合，成功執行之srsGUI畫面如下圖所示



Stage 5 Check List

	項目	內容
	搜尋商用eNB	是否能順利搜尋到商用的eNB
	連接商用eNB	是否能順利從商用eNB取得其MIB與SIB
	建立NB-IoT eNB	確認npdsch_enodeb的執行沒有任何錯誤，可在UE端以頻譜分析儀查看是否收到eNB的無線電訊號
	連接到自己的eNB	確認npdsch_ue是否能連接到自己架設的eNB並順利取得MIB及SIB

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝
- Stage 3. srsLTE設定及量測
- Stage 4. srsLTE參數調整
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
  - Step1 安裝nukxScan server
  - Step2 安裝nukxScan client
  - Step3 執行nukxScan server
  - Step4 執行nukxScan client
- 總結及問題

Step1 安裝nukxScan server

於EPC的終端機輸入

git clone http://github.com/Nukicslab/nukxScan.git

cd nukxScan/server

sudo apt install -y nodejs npm mongodb

sudo npm install

即可完成nukxScan伺服端的安裝

```
user@labepc:~/nukxScan/server$ sudo npm install
npm WARN deprecated core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage d
ue to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.

> core-js@2.6.11 postinstall /home/user/nukxScan/server/node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standar
d library!
```

Step2 安裝nukxScan client

於兩台UE的終端機輸入

git clone http://github.com/Nukicslab/nukxScan.git

cd nukxScan/rpi_client

sudo apt install -y python-is-python3 python3-pip

sudo pip3 install -r requirements.txt

即可完成nukxScan負責掃描之客戶端的安裝

```
Requirement already satisfied: pywifi==1.1.12 in /home/ubuntu/.local/lib/python3
.8/site-packages (from -r requirements.txt (line 1)) (1.1.12)
Collecting requests==2.24.0
  Downloading requests-2.24.0-py2.py3-none-any.whl (61 kB)
     |[redacted]| 61 kB 125 kB/s
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/lib/python3/dist-packag
es (from requests==2.24.0->-r requirements.txt (line 2)) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/lib/python3/dist-packages (f
rom requests==2.24.0->-r requirements.txt (line 2)) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-packa
ges (from requests==2.24.0->-r requirements.txt (line 2)) (2019.11.28)
```

Step3 啟動nukxScan server

於EPC之終端機輸入

sudo systemctl status mongodb

以確認mongodb的運作情形，若正常運作應如下圖所示，接下來即可在server的目錄下輸入

npm start

來啟動nukxScan之伺服端以接收來自UE所收集的資料

```
● mongodb.service - An object/document-oriented database
     Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2020-07-30 15:24:48 CST; 4min 17s ago
       Docs: man:mongod(1)
   Main PID: 4068 (mongod)
      Tasks: 23 (limit: 9311)
     Memory: 43.8M
     CGroup: /system.slice/mongodb.service
             └─4068 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /etc/mongodb.conf
```

Step4 啟動nukxScan client

於兩台UE的終端機輸入

sudo python rpi_client.py

以啟動nukxScan之客戶端，客戶端啟動後會列出目前系統所有的無線網卡介面選擇，請選擇wlan0以使用RPi4內建的無線網卡。接下來程式應會持續掃描附近所有AP的資訊並回傳至伺服端，周期約10秒

```
ubuntu@pi4ue2:~/rpi_client$ sudo python3 rpi_client.py
0 p2p-dev-wlan0
1 wlan0
Please select the interface number:1
Selected interface wlan0
[08:56:21] Begin scanning...
```

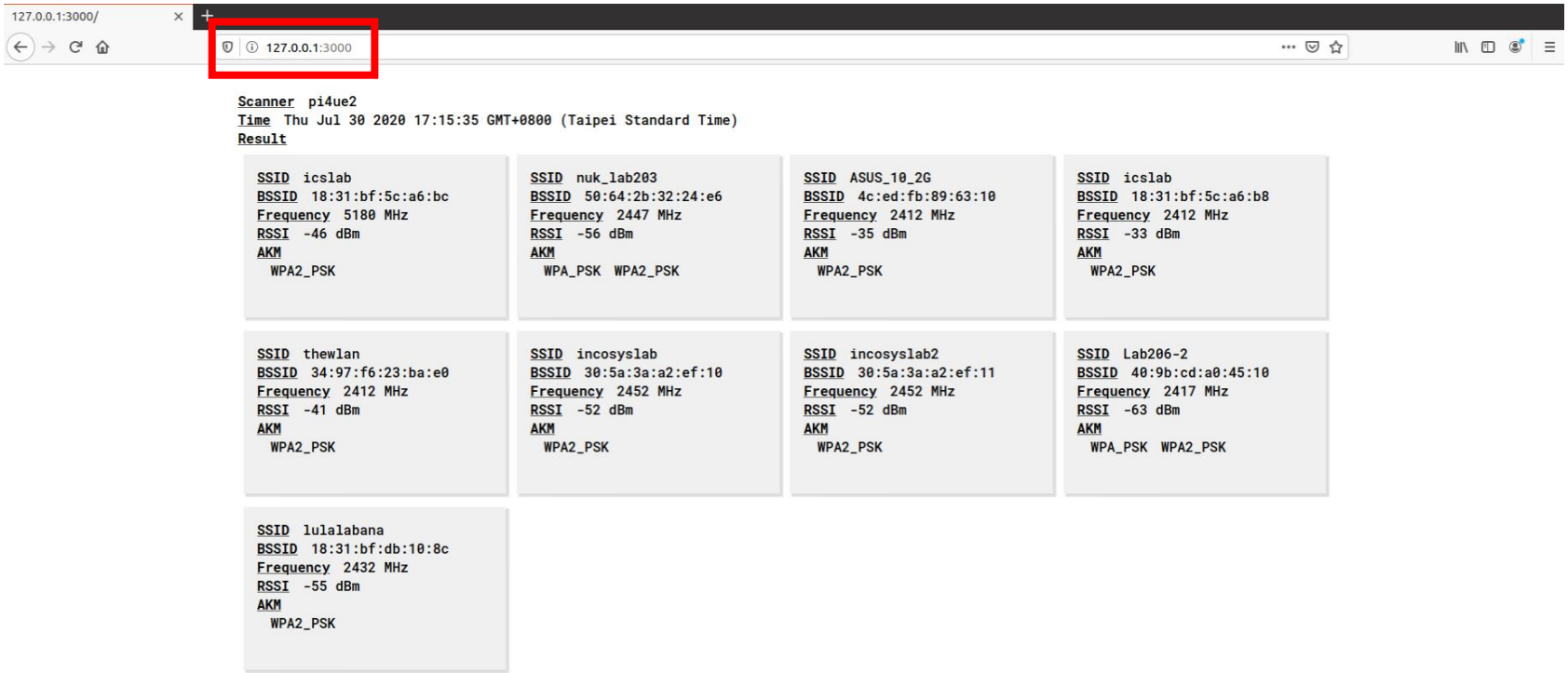
Client顯示結果

Client開始掃描附近AP資訊後會將其顯示在終端機上並同時上傳至Server，若成功上傳至Server會顯示 “Uploading result was completed” 字樣

```
0 p2p-dev-wlan0
1 wlan0
Please select the interface number:1
Selected interface wlan0
[09:15:24] Begin scanning...
Scanning was completed. Result as follow.
        {'ssid': 'icslab', 'bssid': '18:31:bf:5c:a6:bc', 'akm': ['WPA2_PSK'], 'freq': 5180, 'rssi': -46}
        {'ssid': 'nuk_lab203', 'bssid': '50:64:2b:32:24:e6', 'akm': ['WPA_PSK', 'WPA2_PSK'], 'freq': 2447, 'rssi': -56}
        {'ssid': 'ASUS_10_2G', 'bssid': '4c:ed:fb:89:63:10', 'akm': ['WPA2_PSK'], 'freq': 2412, 'rssi': -35}
        {'ssid': 'icslab', 'bssid': '18:31:bf:5c:a6:b8', 'akm': ['WPA2_PSK'], 'freq': 2412, 'rssi': -33}
        {'ssid': 'thewlan', 'bssid': '34:97:f6:23:ba:e0', 'akm': ['WPA2_PSK'], 'freq': 2412, 'rssi': -41}
        {'ssid': 'incosyslab', 'bssid': '30:5a:3a:a2:ef:10', 'akm': ['WPA2_PSK'], 'freq': 2452, 'rssi': -52}
        {'ssid': 'incosyslab2', 'bssid': '30:5a:3a:a2:ef:11', 'akm': ['WPA2_PSK'], 'freq': 2452, 'rssi': -52}
        {'ssid': 'Lab206-2', 'bssid': '40:9b:cd:a0:45:10', 'akm': ['WPA_PSK', 'WPA2_PSK'], 'freq': 2417, 'rssi': -63}
        {'ssid': 'lulalabana', 'bssid': '18:31:bf:db:10:8c', 'akm': ['WPA2_PSK'], 'freq': 2432, 'rssi': -55}
Uploading result was completed.
```

Server顯示結果

於EPC開啟瀏覽器並輸入http://127.0.0.1:3000/，應會顯示伺服端從客戶端所蒐集的所有AP資訊



Stage 6 Check List

	項目	內容
	nukxScan Server 安裝	確認Server所需要的所有相依套件都已安裝完畢
	nukxScan Client 安裝	確認Client所需要的所有相依套件都已安裝完畢
	MongoDB	確認Server端的MongoDB安裝成功且順利執行
	nukxScan Server 運作	確認nukxScan Server 可正常收到Client的資料且正確顯示在網頁中
	nukxScan Client 運作	確認nukxScan Client可以正常捕捉附近AP的資訊並將其回傳至Server

Outline

- 實驗目的及實驗內容
- 背景知識
- 實驗環境
- Stage 1. 樹莓派環境架設
- Stage 2. USRP與srsLTE安裝與測試
- Stage 3. srsLTE設定及量測
- Stage 4. srsLTE參數調整
- Stage 5. NB-IoT
- Stage 6. mMTC 應用
- 總結及問題

總結

- 學習如何在樹莓派上安裝UHD及srsLTE
- 學習如何設定srsLTE以讓其在樹莓派上運作
  - 學習srsLTE內的UHD及USIM設定
  - 學習使用UHD內建的頻譜分析儀
  - 了解srsGUI各欄位意義
- 調整參數並了解其對整體Throughput的影響
  - 了解Physical Resource Blocks數量的影響
  - 了解PDSCH MCS的影響
  - 了解PUSCH MCS的影響
- 利用srsLTE觀察NB-IoT
  - 學習如何搜尋商用eNB搜尋及觀察其MIB與SIB
  - 了解如何自行架設NB-IoT之eNB與UE
- 了解mMTC 之應用

問題

- 若同時將n_prb設為6、pdsch_max_mcs及pusch_max_mcs皆設為5的話能否成功執行iperf3雙向流通量測試
- 若可以，請給出流通量的雙向測試結果並與原始設定檔的流通量做比較，說明造成差異的可能原因
- 若不行，請試著說明無法成功進行流通量測試的原因