Wireless Communication Systems
@CS.NCTU

USRP Lab 1

Yu-Lin Wei

2016.9.27

Outline

- Intro
  - USRP
  - Environment
- Tasks
  - OFDM signal generator (MATLAB)
  - Tx / Rx (C++ for USRP)
  - Decoding (MATLAB)
- Grading Criteria

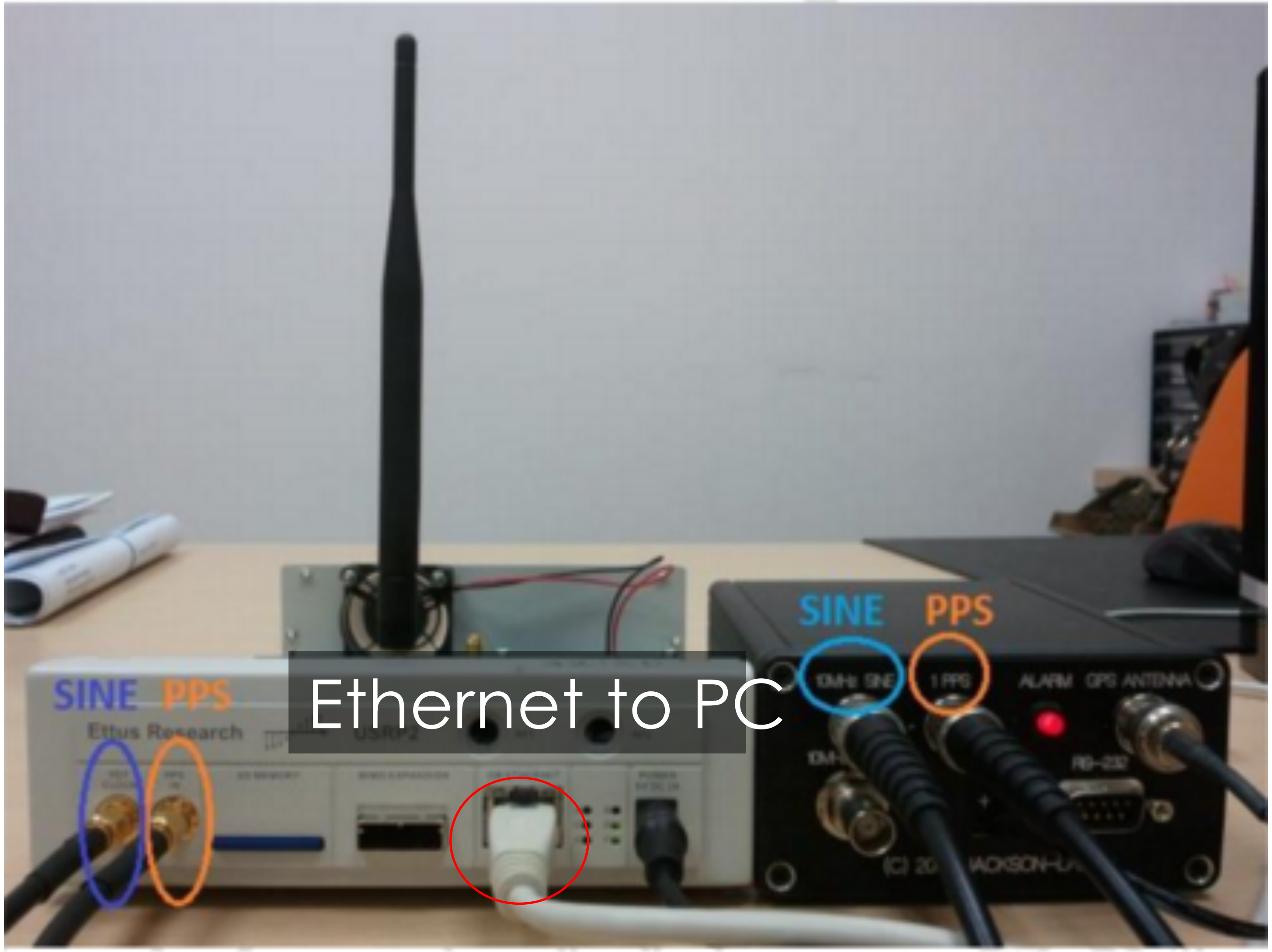
Intro

What is USRP?

USRP

- Universal Software Defined Radio
  - Expensive!
  - Use C++/ python/ GUI to define the radio!!

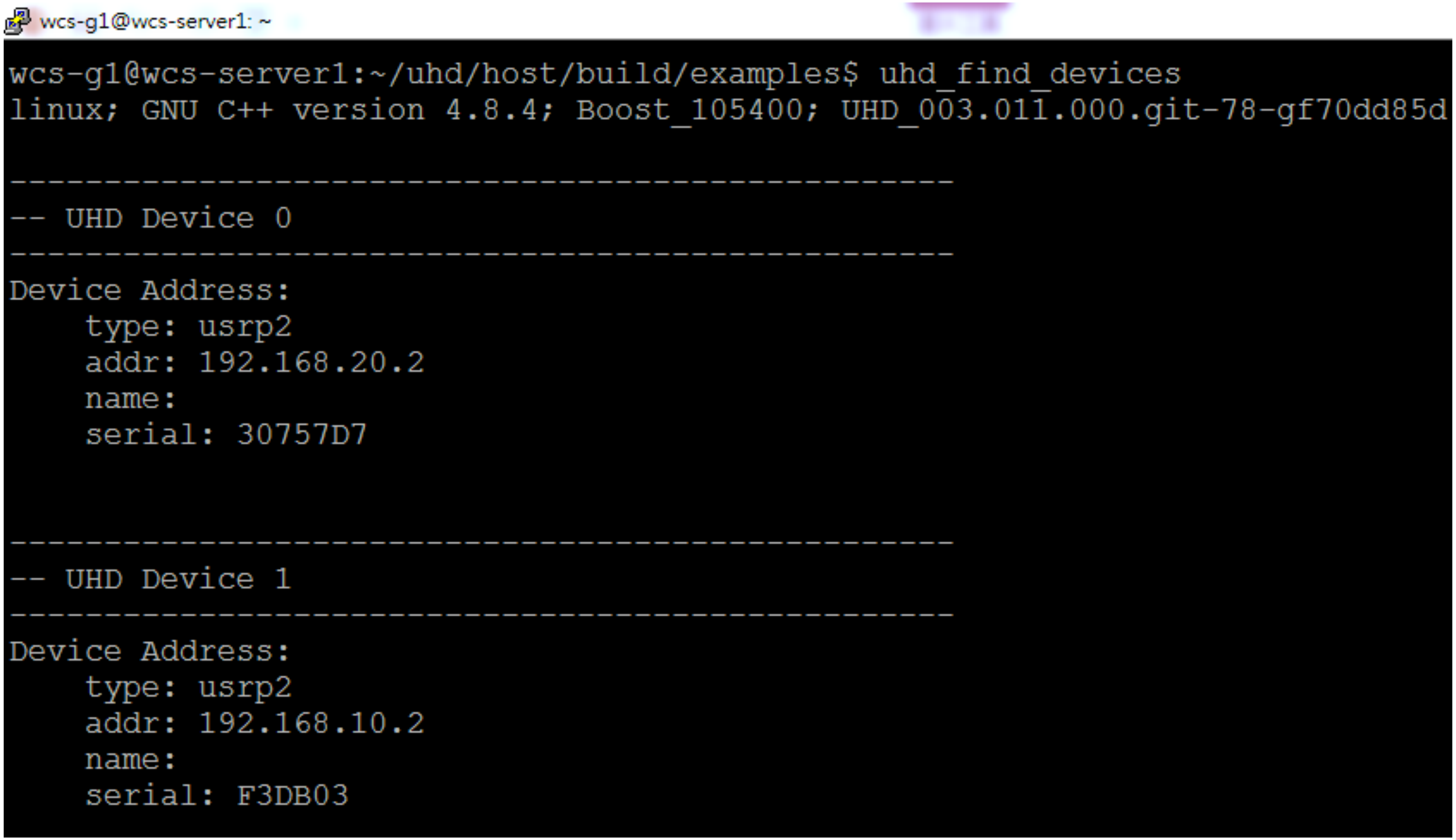
- Official document
  - https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR.pdf


SINE
PPS
Ettus Research
Ethernet to PC
SINE
PPS
10MHz SINE
1 PPS
ALARM
GPS ANTENNA
RS-232

USRP Driver (API)

- UHD
  - USRP Hardware Driver
  - C++ API
  - http://files.ettus.com/manual/
  - https://github.com/EttusResearch/uhd

- UHD tool
  - **host/build/utils/uhd_find_devices**
    - This program scans the network for supported devices and prints out a list of discovered devices and their IP addresses

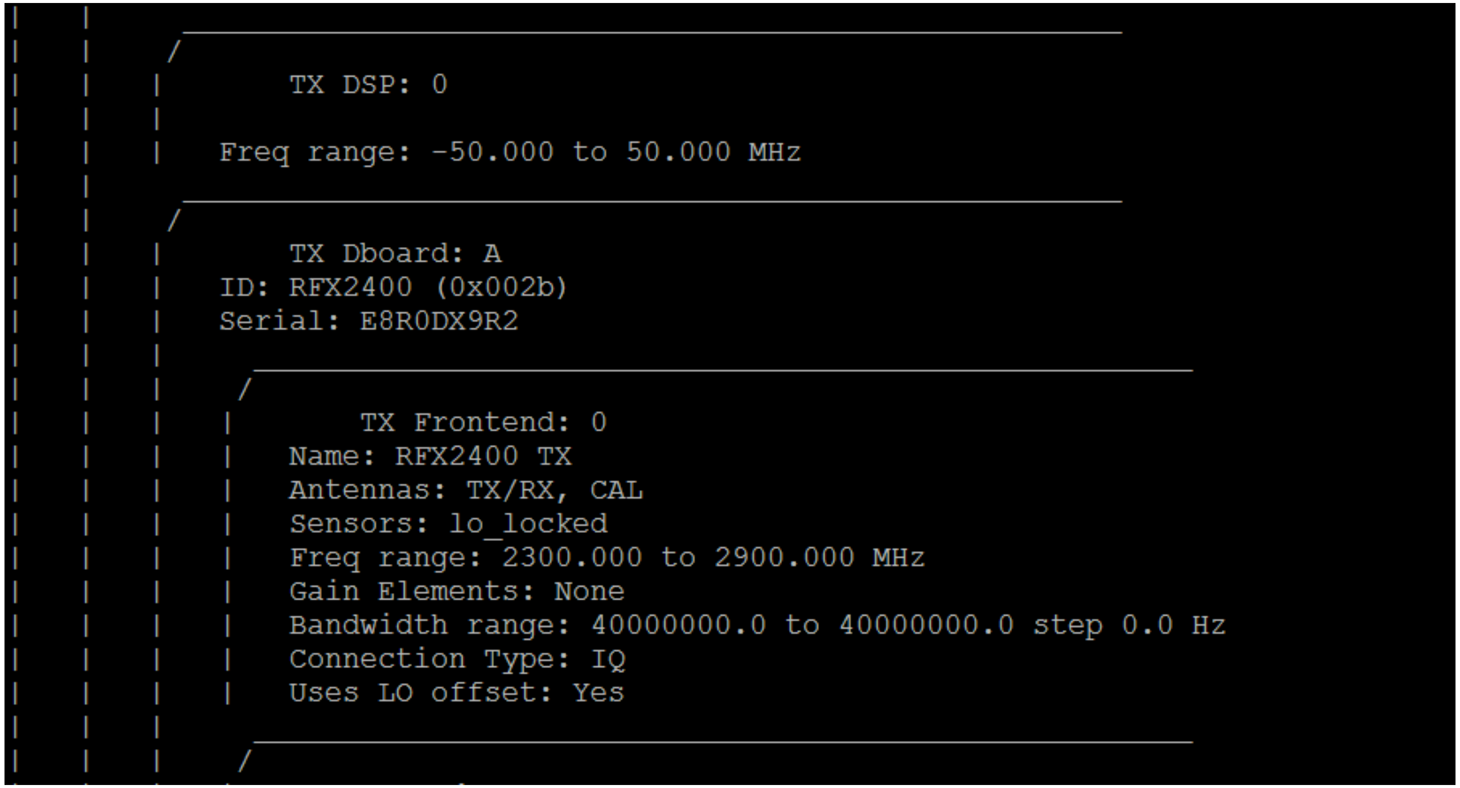


```
wcs-g1@wcs-server1:~/uhd/host/build/examples$ uhd_find_devices
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.011.000.git-78-gf70dd85d

--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    type: usrp2
    addr: 192.168.20.2
    name:
    serial: 30757D7


--------------------------------------------------
-- UHD Device 1
--------------------------------------------------
Device Address:
    type: usrp2
    addr: 192.168.10.2
    name:
    serial: F3DB03
```

- host/build/utils/uhd_usrp_probe
  - This program constructs an instance of the device and prints out its properties, such as detected daughterboards, frequency range, gain ranges, etc



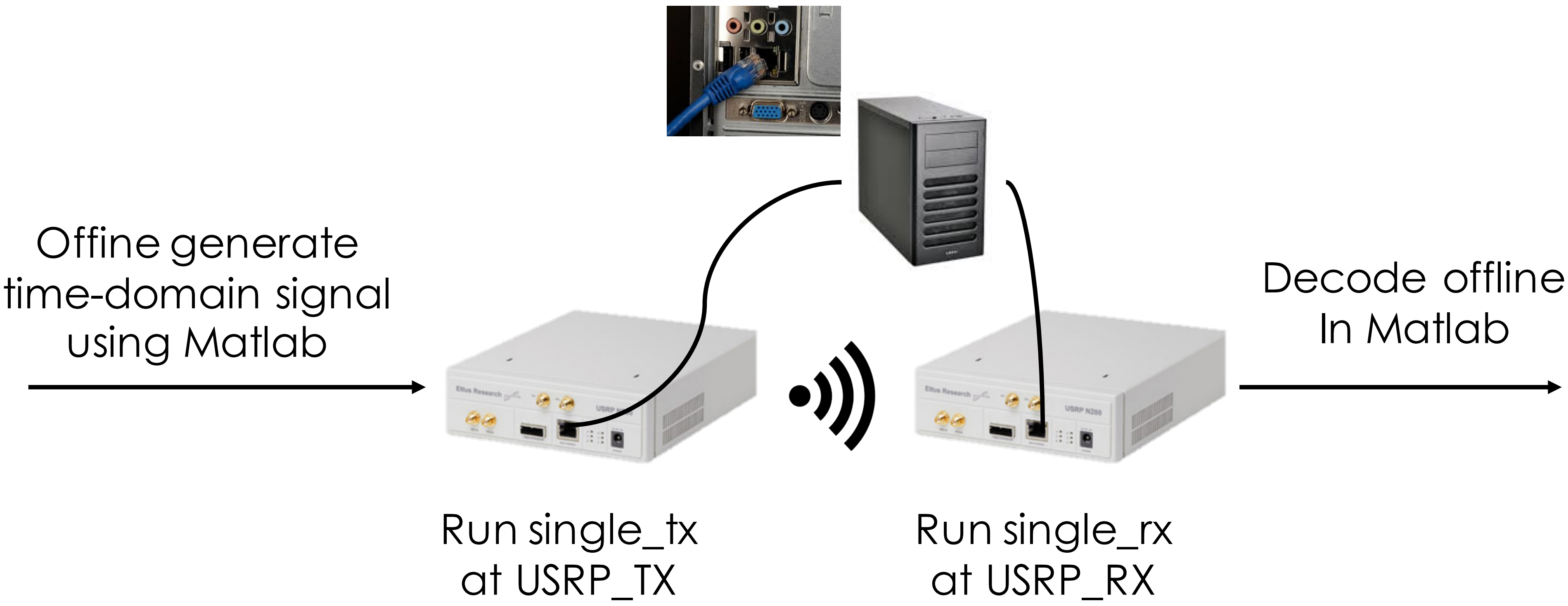
```
|   |     _____________________________________________________
|   |    /
|   |   |       TX DSP: 0
|   |   |
|   |   |   Freq range: -50.000 to 50.000 MHz
|   |     _____________________________________________________
|   |    /
|   |   |       TX Dboard: A
|   |   |   ID: RFX2400 (0x002b)
|   |   |   Serial: E8R0DX9R2
|   |   |     _____________________________________________________
|   |   |    /
|   |   |   |       TX Frontend: 0
|   |   |   |   Name: RFX2400 TX
|   |   |   |   Antennas: TX/RX, CAL
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 2300.000 to 2900.000 MHz
|   |   |   |   Gain Elements: None
|   |   |   |   Bandwidth range: 40000000.0 to 40000000.0 step 0.0 Hz
|   |   |   |   Connection Type: IQ
|   |   |   |   Uses LO offset: Yes
|   |   |     _____________________________________________________
|   |   |    /
```

How to Compile

- File (Source) Directory
  - Use built in Makefile
  - Put your files in ~/uhd/host/examples/
  - Add your files to the Cmakelist.txt in ~/uhd/host/examples

- Compile (Binary) Directory
  - cd ~/uhd/host/build/examples
  - make

Environment



Offine generate
time-domain signal
using Matlab

Run single_tx
at USRP_TX

Run single_rx
at USRP_RX

Decode offline
In Matlab

- USRP Testbed in LAB / office
- Access through ssh (test your single_tx / single_rx)
- Run Matlab in your own machine

USRP Server

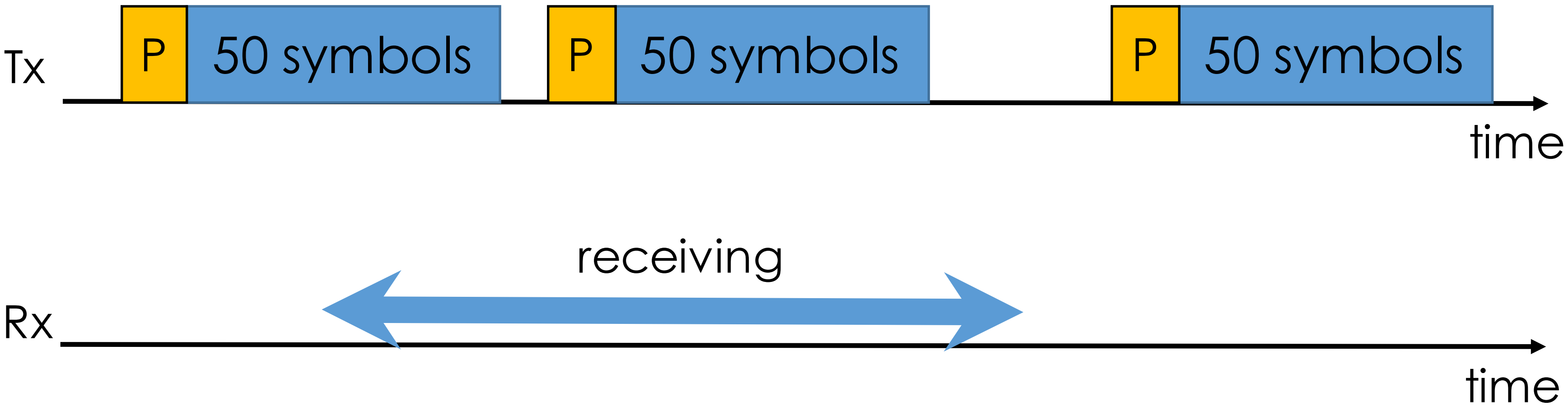
- ssh wcs-g#@140.113.203.6
  ssh wcs-g#@140.113.207.100
  - e.g., wcs-g1@140.113.203.6 default password:
- HW code put in **~/uhd/host/example**
  - single_tx.cpp/ single_tx.h
  - single_rx.cpp/ single_rx.h
- cd ~/uhd/host/build
- cmake .. (only the first time)
- make
- cd example (now in ~/uhd/host/build/example)

USRP Server

- mkdir wcs_trace
- Transimitter :
  - ./single_tx --f=2.49 --i=128
- Receiver:
  - ./single_rx --f=2.49 --i=128
- Received data in ./wcs_trace/recv_signal.bin

TODO

- Tx repetitively sends 50 symbols
- Rx receives at least one batch of 50 symbols
- Matlab offline decoding



Sample Code

- WARP (Wireless Open Access Research Platform)
- 1x1 OFDM example: https://warpproject.org/trac/wiki/WARPLab/Examples/OFDM
    - OFDM symbol generation (pilot/IFFT/cyclic prefix)
    - OFDM Tx/ Rx
    - Decoding
        - Packet detection
        - SFO / CFO correction
        - FFT
- Default setting
    - USE_WARPLAB_TXRX = 0 to see the simulation result
    - Set MOD_ORDER = 2 to use BPSK modulation

Task 1: OFDM Symbol Generator

- Read WARP code
  - NOTE : fft() in MATLAB uses index 1 ~ x to represent the power of frequency $\left[0,\frac{x}{2}\right),\left[-\frac{x}{2},0\right)$ → use fft_shift() to switch the order
- TODO (signal_gen.m)
  - Cut (re-write) the part for generating time-domain signals
  - Save as signal_gen.m
  - signal_gen.m outputs the signal to be transmitted as src_data_1.bin and src_data_1.mat
  - src_data_1.bin – will be fed into single_tx.c for USRP transmission
  - src_data_1.mat – ground truth for decoding / plotting

Task 2: USRP Transmitter

- Login to the testbed (page 12)
- Compile the example code and test (page.13)
- Sample code provided by the TA
  - Transmit random integers on 2.49GHz
  - Please check the IP before transmission
    - Command: uhd_find_device
  - Launch the transmitter (USRP_TX) first
  - ./single_tx --f=2.49 --i=128
  - ./single_rx --f=2.49 --i=128
  - Press ^C to terminate after the receiver finishes receiving
- TODO (single_tx.cpp/ single_tx.h)
  - Modify single_tx.cpp/ singal_tx.h to transmit the message you just generated

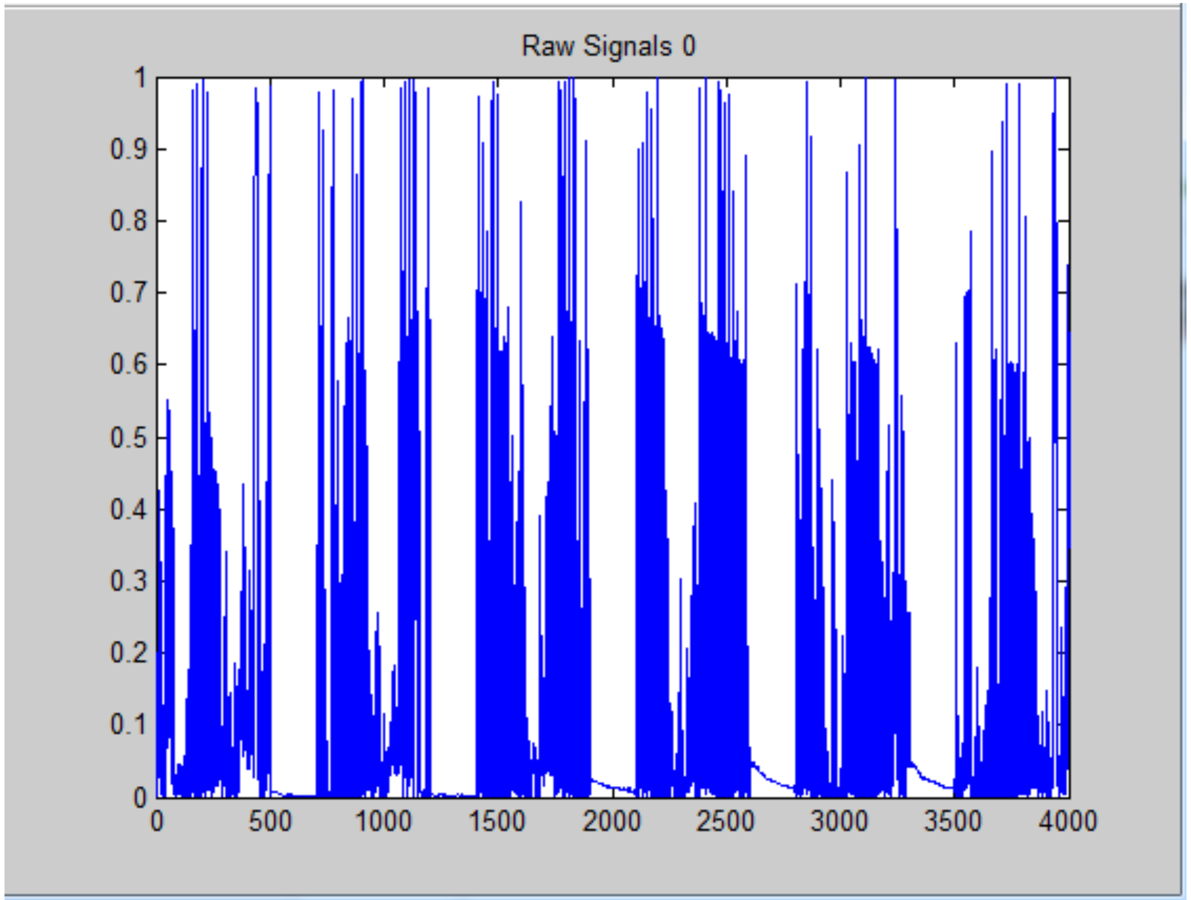
Task 3: USRP Receiver

- Sample code provided by the TA
  - Receive the upcoming signal
  - Save the data at wcs_trace/recv_singal.bin
  - Launch single_rx after single_tx
  - ./single_rx --f=2.49 --i=128

- TODO (single_rx.h)
  - Modify single_rx.h to ensure at least receiving one batch of 50 symbols for offline decoding

TA's sample code

- Send random integers
- Plot of recv_signal.bin

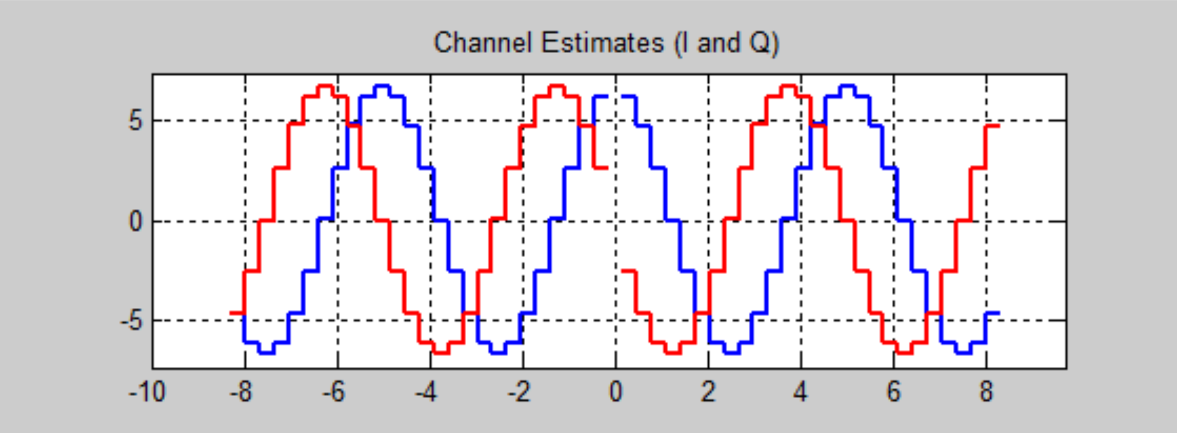


Task 4: Decoding

- Read the WARP code
- TODO (decode.m)
  - Re-implement phase tracking using the regression method mentioned in the lecture
  - Plot the result figures

Required figures

- Figure 1: Channel Estimation H[k] (WARP figure 4-1)



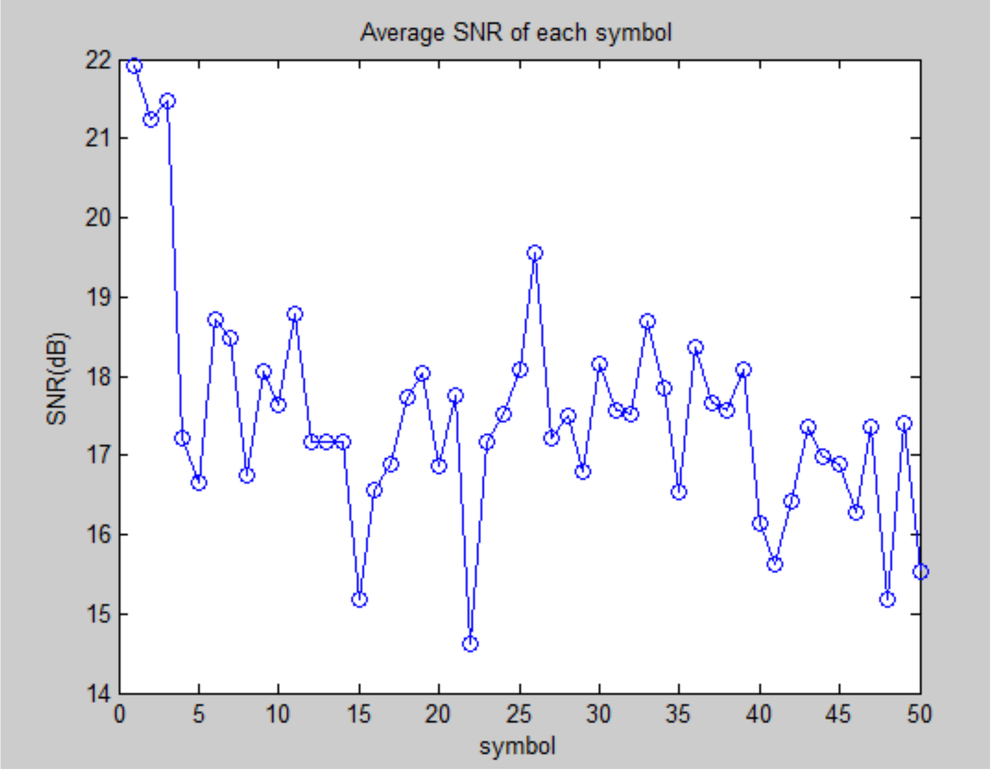
Required figures

- Figure 2: subcarrier SNR
  - average SNR of each data subcarrier among all symbols (bar graph)
  - With and without phase track

- Observation
  - Check if there exists deep fading

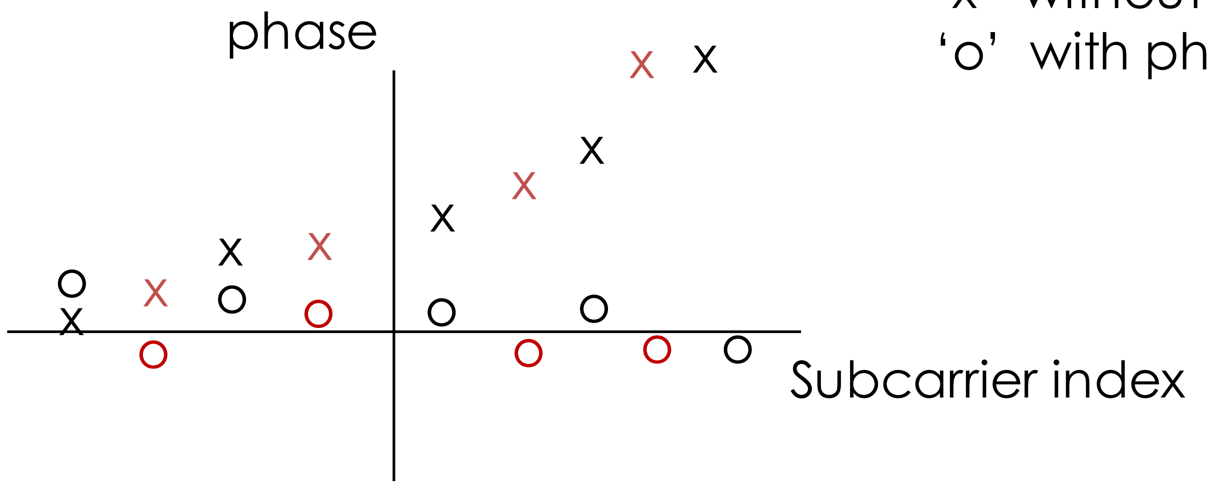
Required figures

- Figure 3: symbol SNR
  - average SNR of all subcarriers for symbols over time (line graph or scatter plot)
  - With and without phase track
- Observation
  - Check if SNR drops over time if phase track is disabled



Required figures

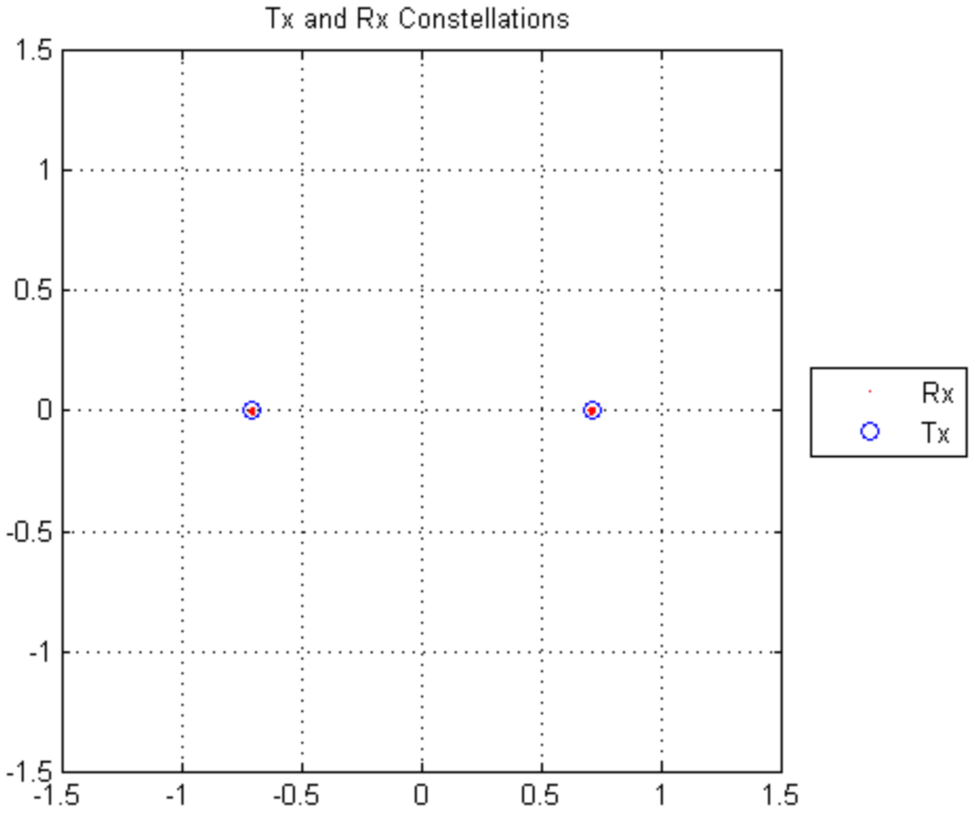
- Figure 4: Phases of decoded signal of different subcarriers in the first symbol
  - with and without phase track



'x' without phase tracking
'o' with phase tracking

Required figures

- Figure 5: constellation points (WARP figure 6)



Demo

- Time/Location
  - Oct. 18(Tue.) 17:00~18:15 in EC-538
  - Or by appointment
  - Contact with TX (張威竣) to sign up the time slot
- Flow
  - Run signal_gen.m
  - Get src_data_1.bin / src_data_1.mat
  - Upload src_data_1.bin to wcs-g[#] account
  - Run ./single_tx and ./single_rx under ~/uhd/host/build/example
  - Download example/wcs_trace/recv_signal.bin
  - Put recv_signal.bin in program/trace
  - Run decode.m to get the figures

Grading

- signal_gen.m: 10%
- Tx/Rx: 20%
- decode.m: 40%
    - Each figure: 8%
- Code readability: 10%
- Q&A: 10%

90%

- Peer review: ±15%

Peer Review

- 15% group member peer review
  - Anonymous
  - Range from -15 ~ 15
  - Grade for each peer, excluding yourself
  - Zero mean

	Alice	Bob	Chris	David
Alice	N/A	-10	-5	+15

- Total score: up to 105

Code Submission

- Deadline: Oct. 18 (Tue.) 23:59
- Email to
  - msn.nctu@gmail.com
  - Email subject: [WCS] lab1_gX
  - WCS_lab1_gX.zip
    - source code (single_tx.cpp/ single_rx.cpp/ decode.m/ signal_gen.m)
    - Report (.pdf): include all figures along with captions and **short** discussion

Q&A