

# Experiments on Communication Networks\_ Week3

2022-2학기



# Overview of Experiments

---

**WEEK1 – Python Visualization**

**WEEK1 – LABVIEW tutorial**

**WEEK1 – Channel sensor tutorial**

**WEEK2 – Active sensing**

**WEEK2 – Radar generator tutorial**

**WEEK2 – Making datasets**

**WEEK3 – Data labeling**

**WEEK3 – Design CNN model**

# Week 1 -Contents

---

**WEEK1 – Python Visualization**

**WEEK1 – LABVIEW tutorial**

**WEEK1 – Channel sensor tutorial**

# 1주차 실습 개요

❖ USRP 실습은 2인 1조로 총 3주동안 진행됩니다.

## ■ 1주차

- LABview 숙지 및 Python, Pytorch 설치
- LABview: 실험실 컴퓨터, Python 및 Pytorch: 개인 컴퓨터
- Python을 통해 IQ data를 Visualization
- USRP를 연결하여 Tx 예제 실행

## ■ 1주차를 시작하기 전에 앞서 Python 및 Pytorch를 설치합니다.

- 본 작업은 PPT 자료 외의 구글링을 통해 다른 방법으로 설치 및 실행하여도 무방합니다.
- 본 자료는 Window 10 환경에서 제작되었습니다.
- 사용하는 툴은 Anaconda prompt 및 VS code 를 권장합니다.
- Visualization 코드는 제공됩니다.
- 본 자료가 제공하지 않는 실행 오류가 존재할 수 있습니다. 대부분 구글링을 통해 해결할 수 있을 것입니다.

# 1주차 실습 개요

---

## ❖ 추가 실험 및 실험실 이용

- 남혜린 조교에게 문의 010-4154-1988 / [hlnam@ramo.yonsei.ac.kr](mailto:hlnam@ramo.yonsei.ac.kr)
- 수요일 13시~ 이외 사용 가능

# Python 설치

## ❖ Anaconda 다운로드 및 설치

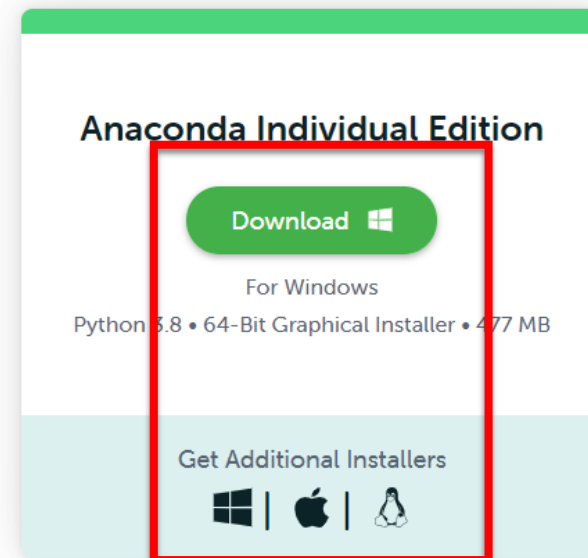
- Anaconda prompt: 원하는 버전의 Python으로 가상 환경 구축에 용이
- <https://www.anaconda.com/products/individual#download-section>
  - 본인 운영체제 확인하여 설치.



Individual Edition

## Your data science toolkit

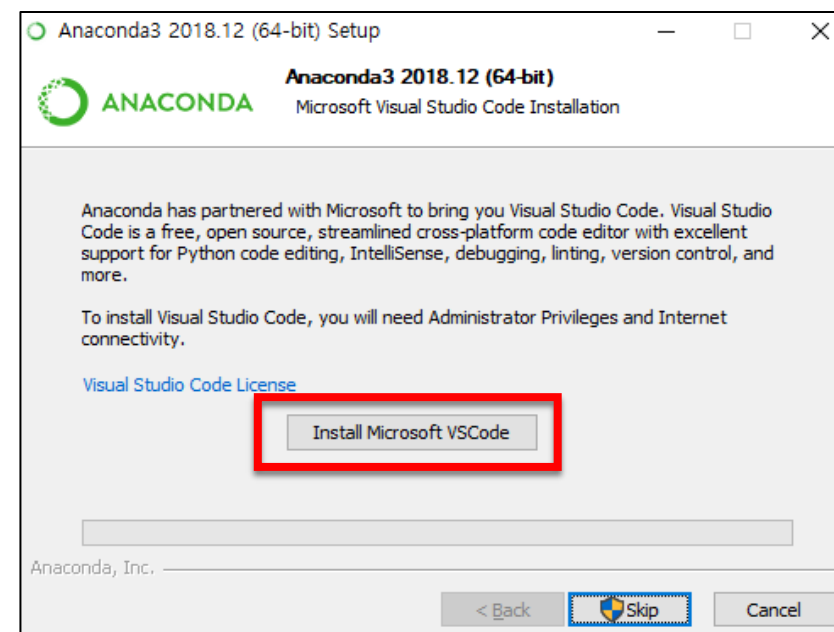
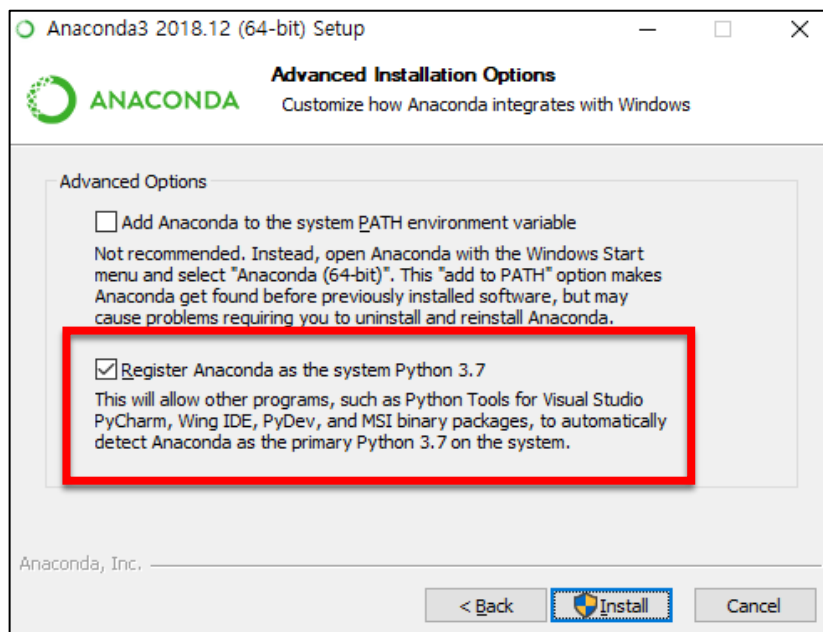
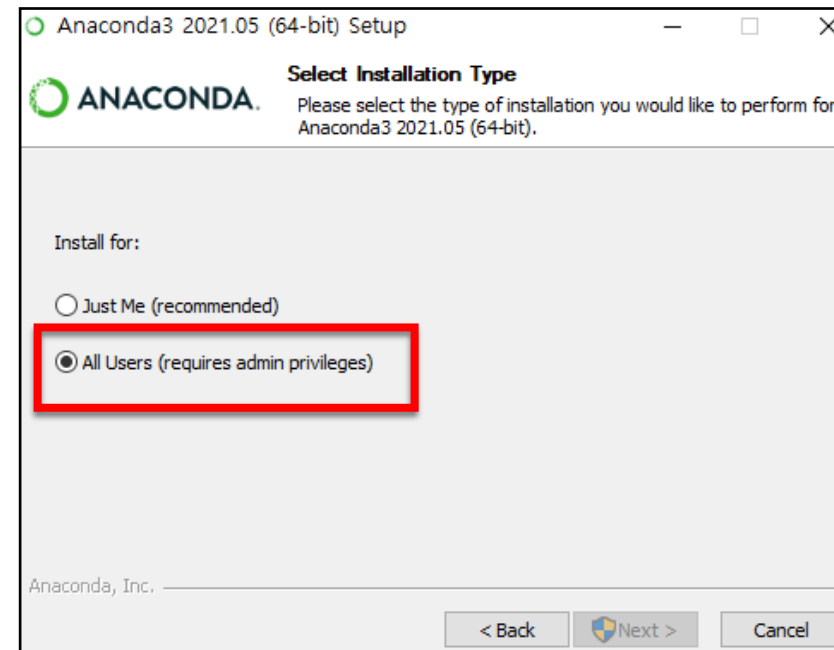
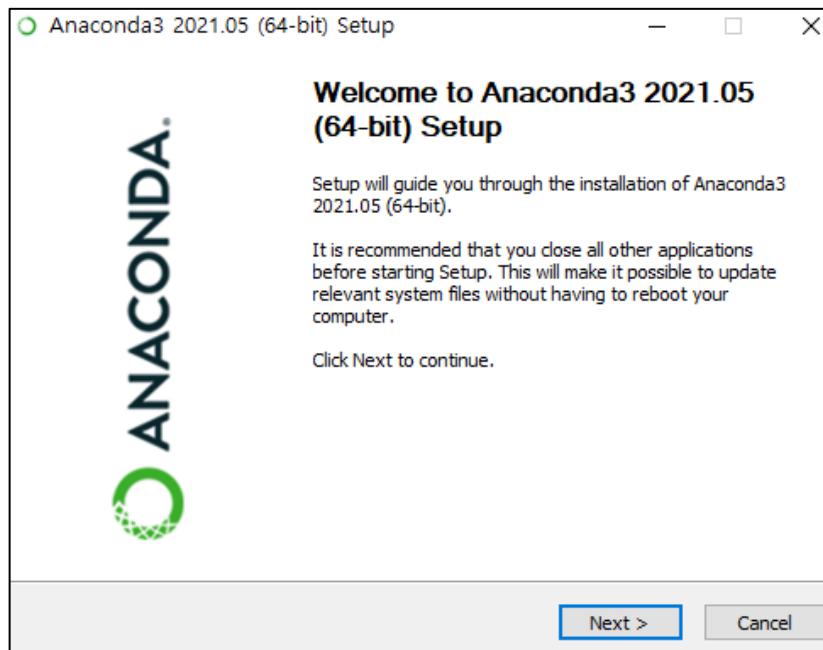
With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.



# Python 설치 (원격X)

## ❖ 주요 박스 체크하여 설치

- 다른 IDE 설치하여 동작하여도 무방하나 VS Code 설치를 권장합니다.
  - 가볍고 빠릅니다.

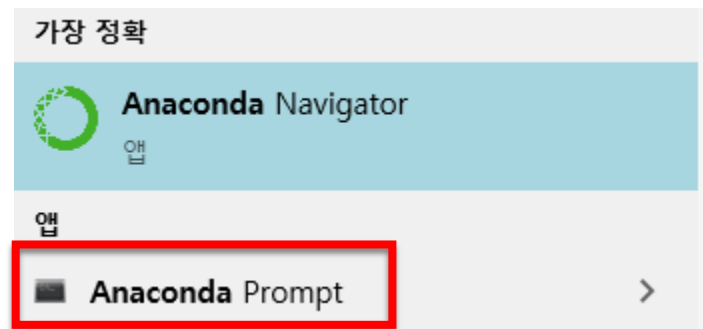


# Python 가상환경 생성

## ❖ Anaconda Prompt 실행

### ■ 명령어 입력: `conda create -n env_name python=3.7`

- env\_name 에는 본인이 원하는 environment 이름 입력
- y 입력하여 가상환경 설치



Anaconda Prompt

```
(base) C:\Users\Kyuwon>conda create test_env python=3.7
```

Anaconda Prompt - conda create -n test\_env python=3.7

The following packages will be downloaded:

package	build	
ca-certificates-2021.7.5	haa95532_1	113 KB
certifi-2021.5.30	py37haa95532_0	139 KB
pip-21.2.2	py37haa95532_0	1.8 MB
python-3.7.11	h6244533_0	14.5 MB
setuptools-52.0.0	py37haa95532_0	711 KB
sqlite-3.36.0	h2bbff1b_0	780 KB
wincertstore-0.2	py37_0	14 KB

Total: 18.0 MB

The following NEW packages will be INSTALLED:

ca-certificates	pkgs/main/win-64::ca-certificates-2021.7.5-haa95532_1
certifi	pkgs/main/win-64::certifi-2021.5.30-py37haa95532_0
openssl	pkgs/main/win-64::openssl-1.1.1k-h2bbff1b_0
pip	pkgs/main/win-64::pip-21.2.2-py37haa95532_0
python	pkgs/main/win-64::python-3.7.11-h6244533_0
setuptools	pkgs/main/win-64::setuptools-52.0.0-py37haa95532_0
sqlite	pkgs/main/win-64::sqlite-3.36.0-h2bbff1b_0
vc	pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime	pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel	pkgs/main/noarch::wheel-0.36.2-pyhd3eb1b0_0
wincertstore	pkgs/main/win-64::wincertstore-0.2-py37_0

Proceed ([y]/n)? y



# Python 가상환경 생성

## ❖ 가상환경 실행

- conda activate `env_name` 입력
- 왼쪽 아래에 해당 가상환경 이름으로 변경되었으면 성공

```
Anaconda Prompt
done
#
# To activate this environment, use
#
#     $ conda activate test_env
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base) C:\Users\kyuwon>conda activate test_env
(test_env) C:\Users\kyuwon>_
```

# Pytorch 설치

❖ Pytorch를 설치하기 위해서는 CUDA Toolkit 설치가 선행되어야 합니다.

## ■ CUDA Toolkit 설치

- <https://developer.nvidia.com/cuda-downloads>
  - 본인 운영체제 확인하여 설치

CUDA Toolkit 11.4 Update 1 Downloads

[Home](#)

### Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

Operating System	<a href="#">Linux</a> <a href="#">Windows</a>
Architecture	<a href="#">x86_64</a>
Version	<a href="#">10</a> <a href="#">Server 2016</a> <a href="#">Server 2019</a>
Installer Type	<a href="#">exe (local)</a> <a href="#">exe (network)</a>

### Download Installer for Windows 10 x86\_64

The base installer is available for download below.

>Base Installer	Download [2.8 GB] <a href="#">Download</a>
-----------------	--

Installation Instructions:

1. Double click cuda\_11.4.1\_471.41\_win10.exe
2. Follow on-screen prompts

The checksums for the installer and patches can be found in [Installer Checksums](#).  
For further information, see the [Installation Guide for Microsoft Windows](#) and the [CUDA Quick Start Guide](#).

# Pytorch 설치

## ❖ Pytorch 설치

### ■ <https://pytorch.org/>

- 본인 운영체제 확인하여 체크 후 아래 해당되는 Command를 Anaconda prompt에 입력
- 본인의 가상 환경에서 제대로 설치되었는지 체크.

#### INSTALL PYTORCH

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.10 builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.

Additional support or warranty for some PyTorch Stable and LTS binaries are available through the [PyTorch Enterprise Support Program](#).

PyTorch Build	Stable (1.9.0)		Preview (Nightly)		LTS (1.8.1)	
Your OS	Linux		Mac		Windows	
Package	Conda	Pip		LibTorch		Source
Language	Python			C++ / Java		
Compute Platform	CUDA 10.2	CUDA 11.1		ROCm 4.2 (beta)		CPU
Run this Command:	<div><b>NOTE:</b> 'conda-forge' channel is required for cudatoolkit 11.1</div> <pre>conda install pytorch torchvision torchaudio cudatoolkit=11.1 -c pytorch -c conda-forge</pre>					

Anaconda Prompt

```
(base) C:\Users\Kyuwon>conda activate test_env
```

```
(test_env) C:\Users\Kyuwon>conda install pytorch torchvision torchaudio cudatoolkit=11.1 -c pytorch -c conda-forge
```

# 기타 모듈 설치

❖ 배포한 Visualization 코드가 실행되기 위해서는 몇가지 모듈이 추가적으로 설치되어야 합니다.

- 필요한 모듈 목록

- pandas
- matplotlib
- opencv
- imageio
- datetime
- numpy

- 모듈 설치하는 다음의 명령어 입력을 통해 진행합니다.

- conda install **module**
  - 파란색 칸에 다른 모듈 이름 입력. ex:) conda install numpy
  - 본인의 가상 환경에서 설치되었는지 **반드시** 체크.
    - conda activate **env\_name**
  - 위 모듈 외 필요한 모듈이 있으면 개별적으로 설치.

# Yonsei CBRS testbed

## ❖ Radar signal generator

- 1차 사용자. 연방, 위성 시스템 신호를 송출

## ❖ CBSD LTE signal

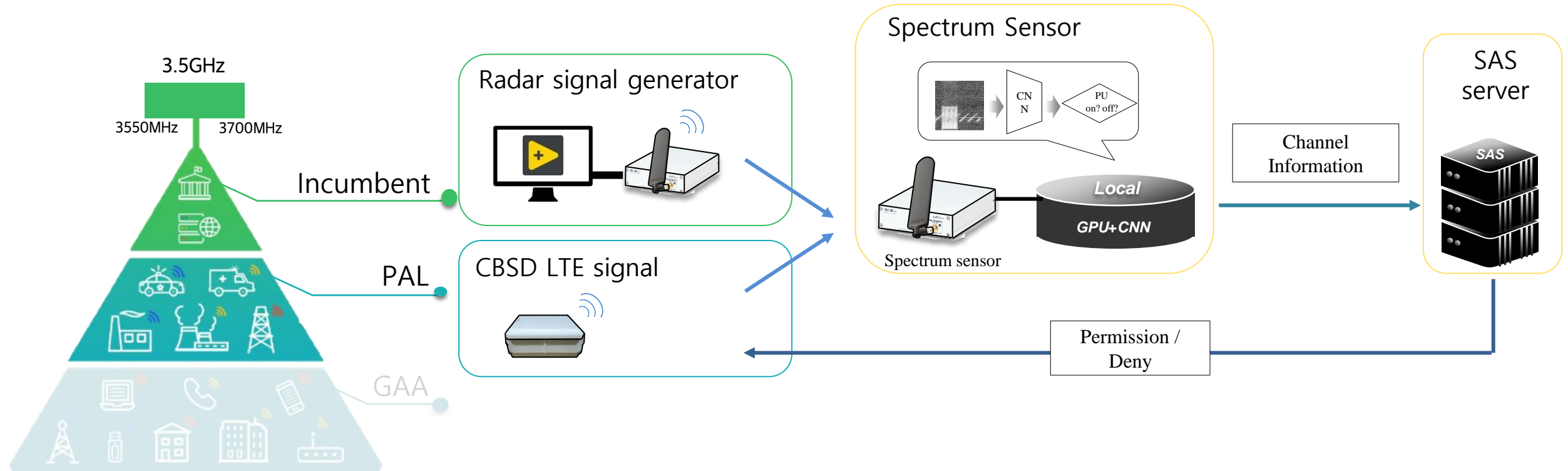
- 2차 사용자. LTE 신호를 송신

## ❖ Spectrum Sensor

- 3.5GHz 대역에 보내지고 있는 신호를 수신하여 1,2차 사용자의 사용 여부를 감지,
- SAS서버에 이러한 정보를 전송

## ❖ SAS server

- 2차 사용자에게 주파수 사용 (신호 송신) 권한을 실시간으로 부여
- Ex) 1차 사용자와 동시에 사용할 경우 주파수 사용 금지 권고



# Yonsei CBRS testbed

## ❖ Radar signal generator

- 1차 사용자. 연방, 위성 시스템 신호를 송출

## ❖ CBSD LTE signal

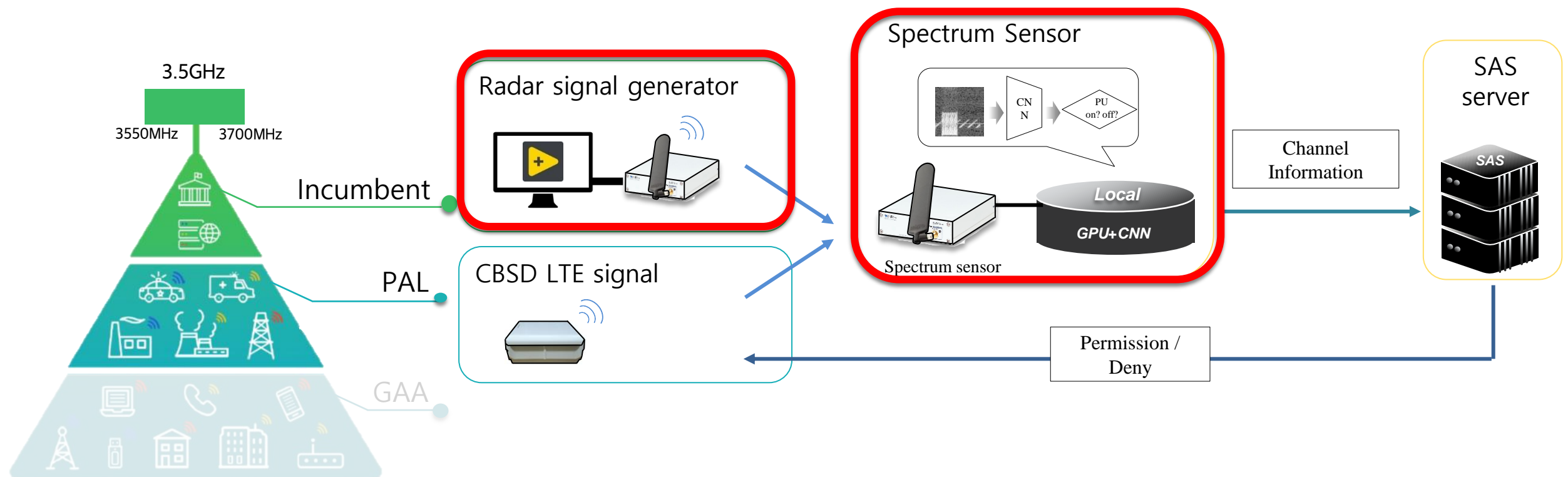
- 2차 사용자. LTE 신호를 송신

## ❖ Spectrum Sensor

- 3.5GHz 대역에 보내지고 있는 신호를 수신하여 1,2차 사용자의 사용 여부를 감지,
- SAS서버에 이러한 정보를 전송

## ❖ SAS server

- 2차 사용자에게 주파수 사용 (신호 송신) 권한을 실시간으로 부여
- Ex) 1차 사용자와 동시에 사용할 경우 주파수 사용 금지 권고



# Yonsei CBRS testbed

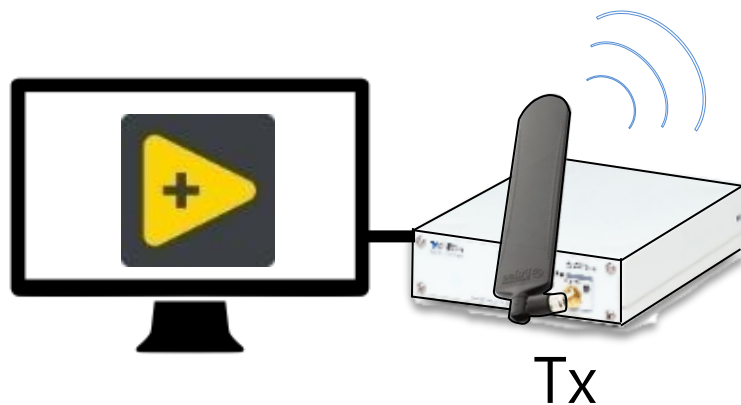
## ❖ Radar signal generator

- 1차 사용자. 연방, 위성 시스템 신호를 송출

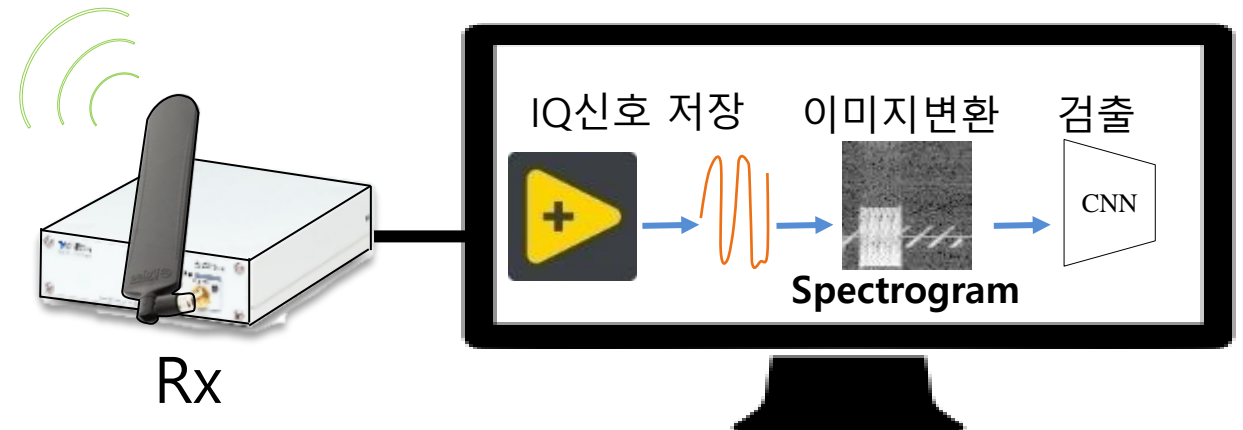
## ❖ Spectrum Sensor

- 3.5GHz 대역에 보내지고 있는 신호를 수신하여 1,2차 사용자의 사용 여부를 감지,
- SAS서버에 이러한 정보를 전송

Radar signal generator



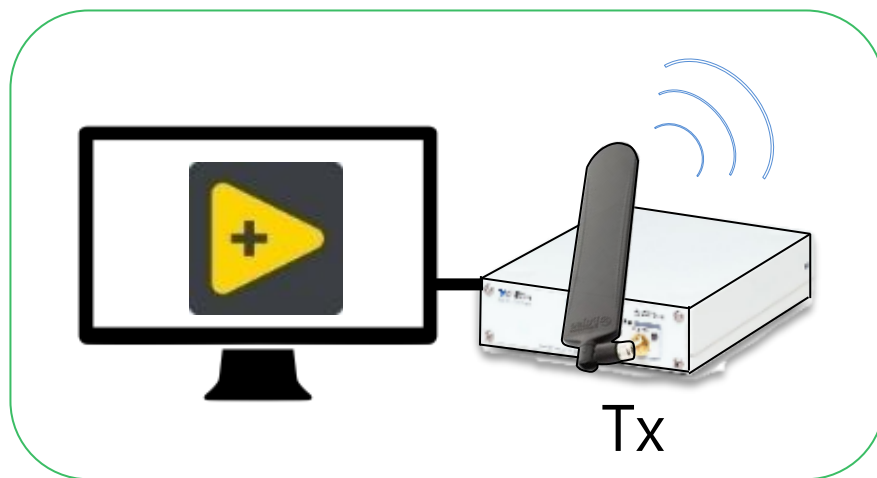
Spectrum Sensor



# Yonsei CBRS testbed

❖ 이제 직접 아래의 실험을 진행해보십시오.

Radar signal generator



Spectrum Sensor





# Radar generator 코드

- ❖ Week4 에서 자세히 다룰 코드입니다
  - 우선 Week3에서 실행 방법을 알아볼 것 입니다.
- ❖ 제공한 Radar\_TX 폴더에서 (압축풀기) 프로젝트 파일을 open

Radar_TX				
이름	수정된 날짜	유형	크기	
Documentation	2021-08-20 오전 11:20	파일 폴더		
PROJECTMEDIA	2021-08-20 오전 11:20	파일 폴더		
Radar_TX.lvcodedb	2021-08-20 오후 5:32	LVCODEDB 파일	8,633KB	
Radar_TX.lvprojectcache	2021-08-20 오전 11:20	LVPROJECTCACH...	633KB	
Radar_TX	2021-08-20 오전 11:20	Project for LabVIE...		←
Tx Continuous Async (CDB Cluster)	2021-08-20 오후 5:32	VI for LabVIEW N...	113KB	

# Example labview code

## ❖ 코드 열기 (바로 안나타나는 경우에만)



# Basic tutorial with Example labview code

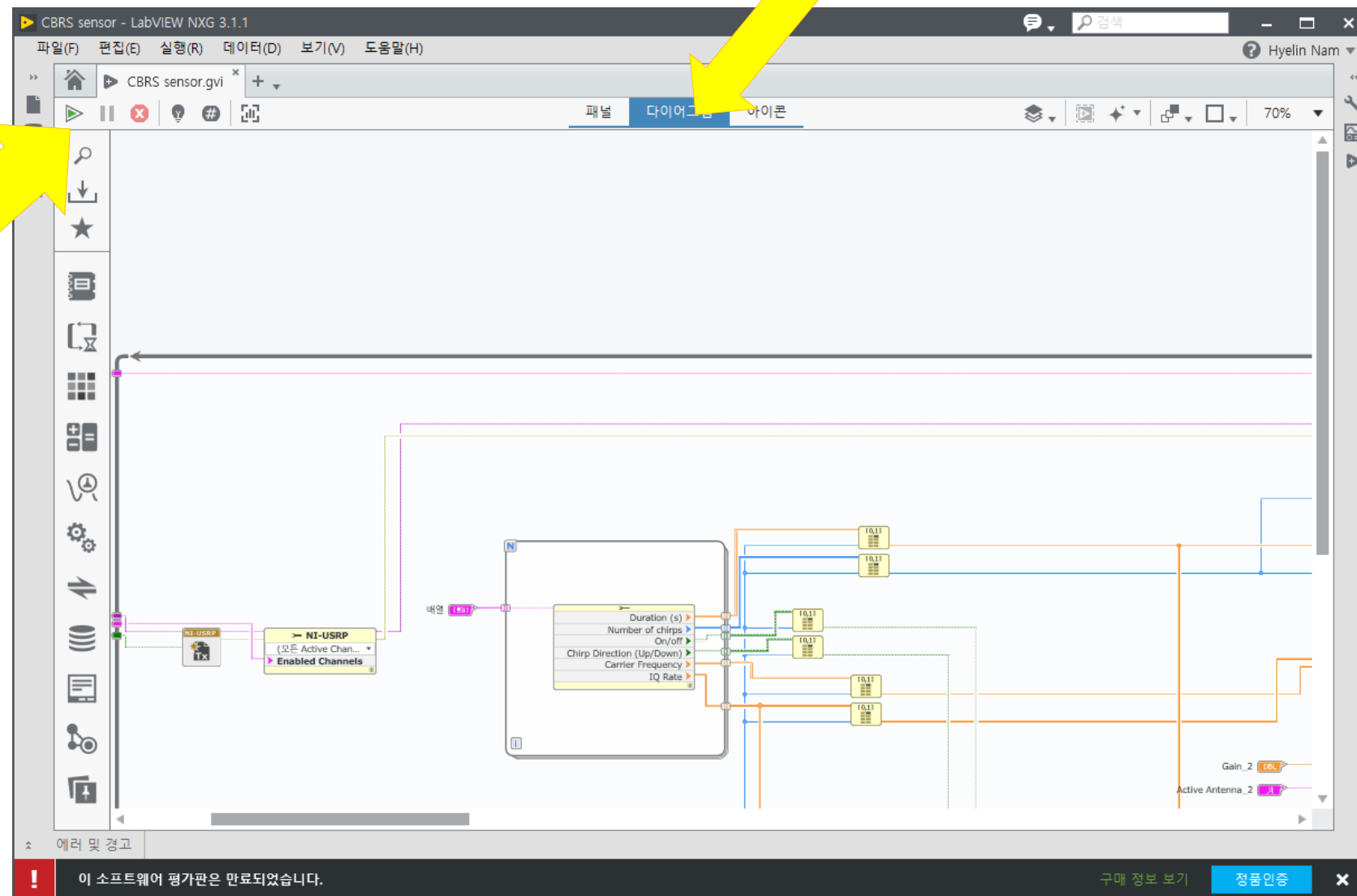
## ❖ 다이어그램

- 코드를 수정할 수 있는 창
- 왼쪽에서 오른쪽 방향으로 실행됨

즉, 변수 정의 (왼쪽) → 변수를 이용하여 연산 → 결과 출력 (오른쪽) 등의 순서로 블록을 배치해야함

- Ctrl+scroll 로 창 확대/축소 가능

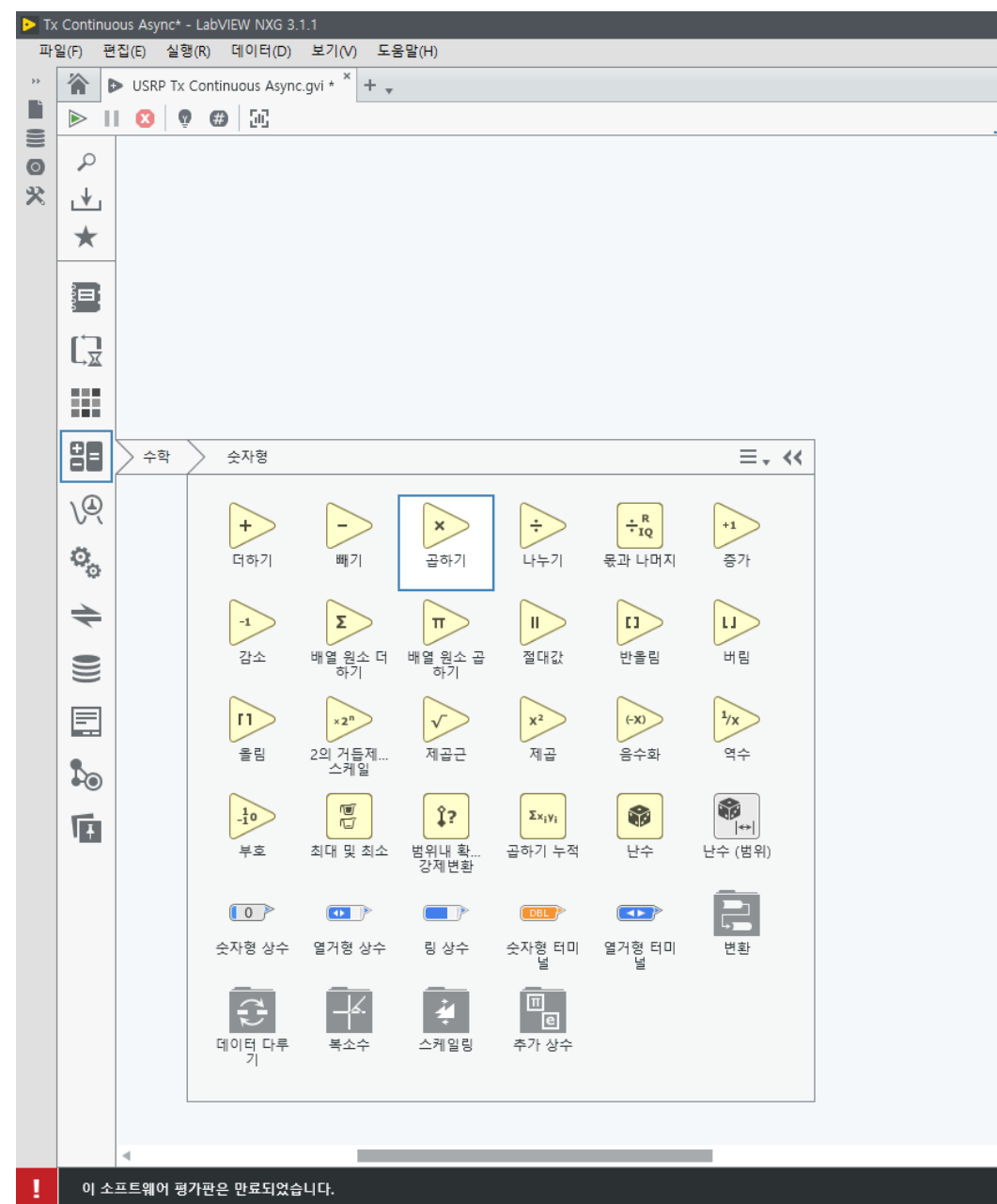
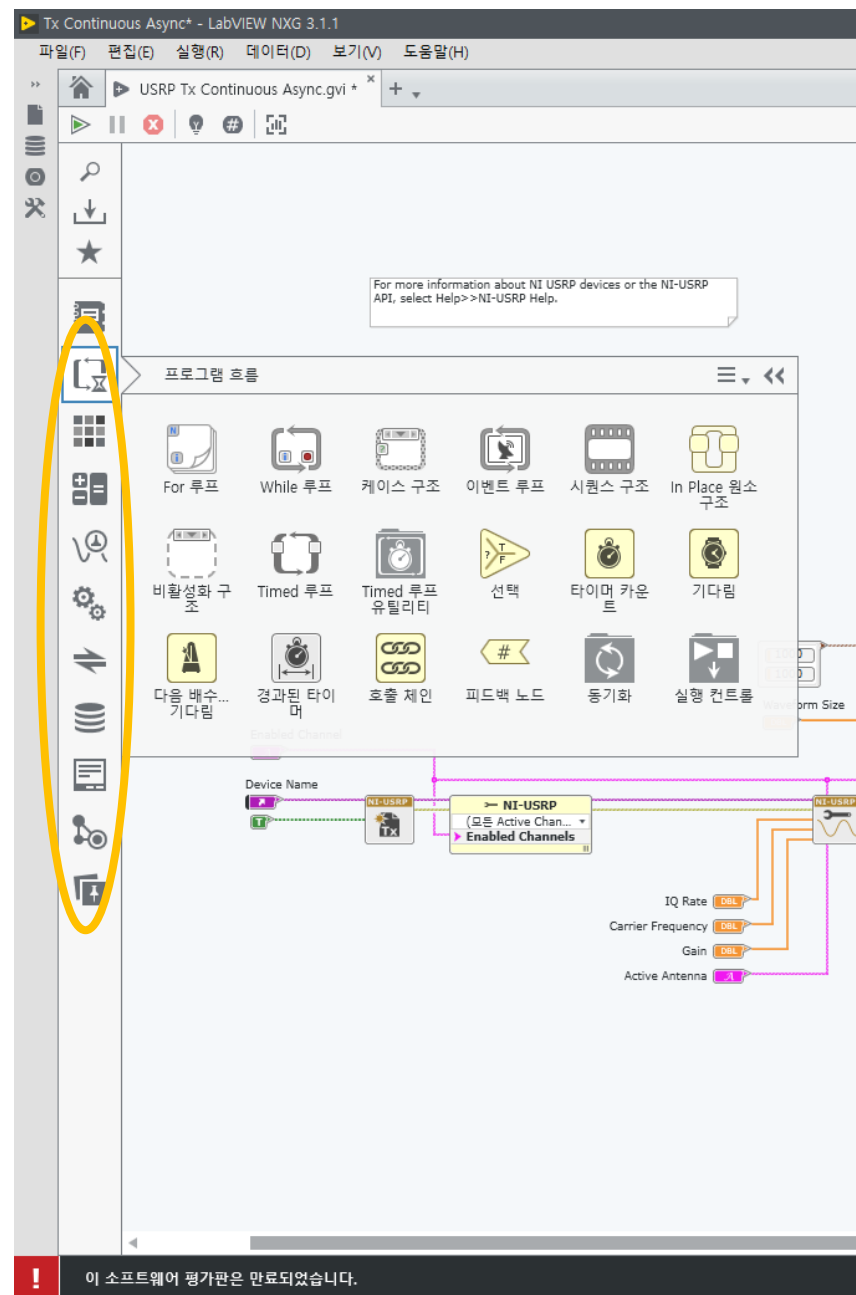
코드 실행,  
중지 버튼



# Basic tutorial with Example labview code

## ❖ 다이어그램 블록 배치

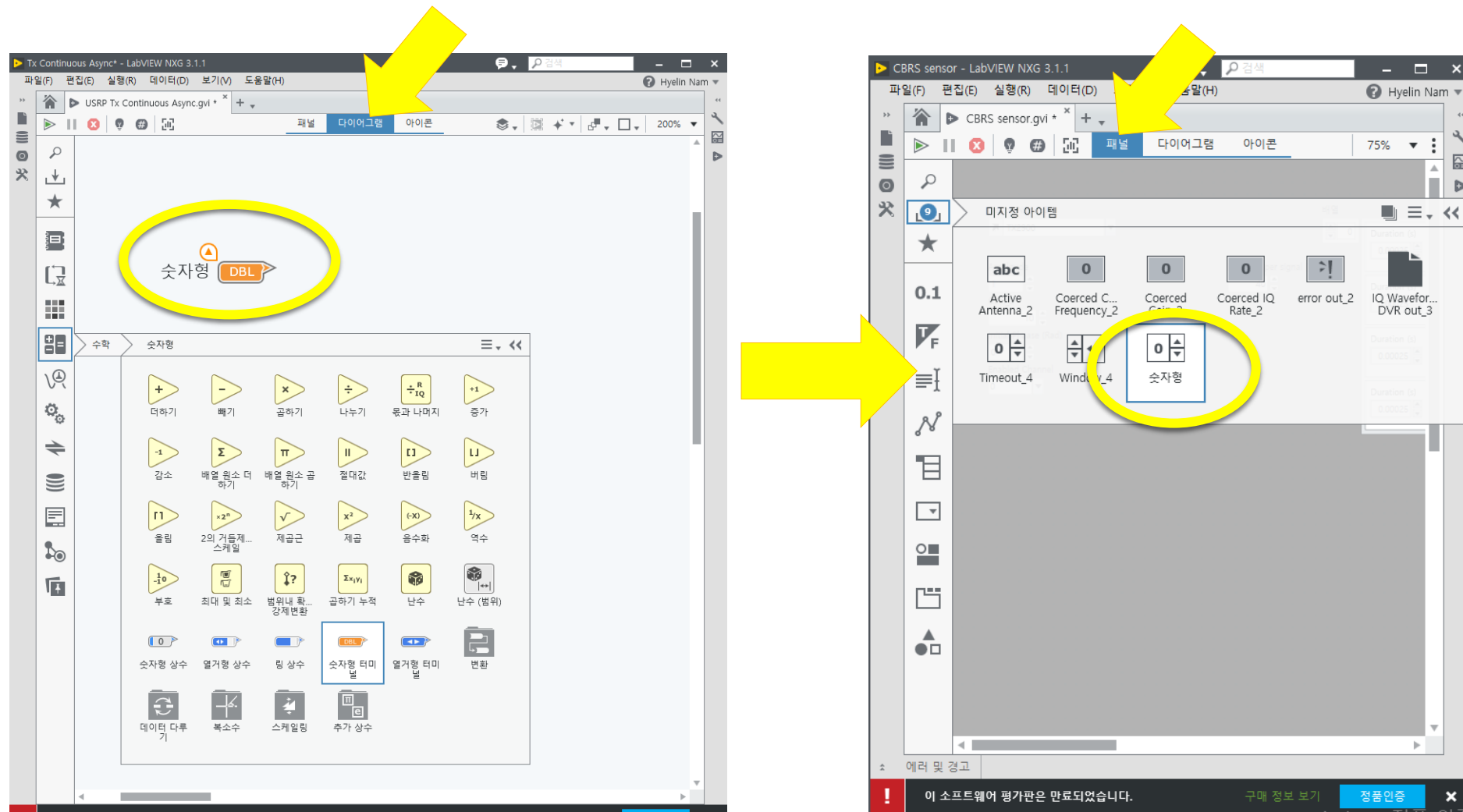
- 루프, 연산기호, 변수 정의 등 다양한 기능의 블록이 있음
- 선택하여 다이어그램 창에 배치



# Basic tutorial with Example labview code

## ❖ 패널

- 다이어그램 창에 배치한 변수 값을 입력 / 출력값을 확인 할 수 있는 창
- 다이어그램창에 입력 혹은 출력 기능의 블록을 배치하면, 이를 확인할 수 있는 블록이 패널창에 연동되어 저절로 나타남
- 패널의 블록과 연동된 다이어그램창의 블록을 알고 싶으면, 블록 더블클릭을 하면 창이 넘어가면서 해당 블록이 표시됨

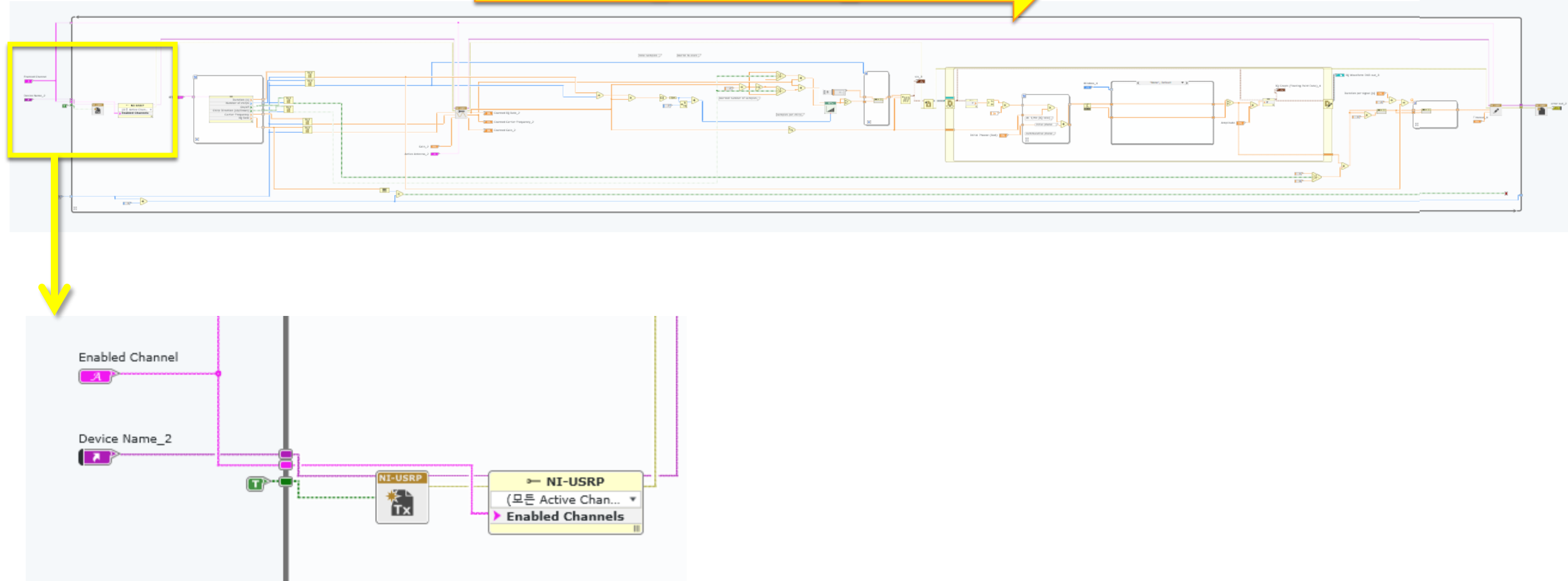


# Basic tutorial with Example labview code

## ❖ USRP 관련 블록

- 자세한 기능은 중요하지 않으나, 코드의 흐름을 해석하기 위해 기본적인 부분은 알면 좋습니다

전체 코드의 흐름



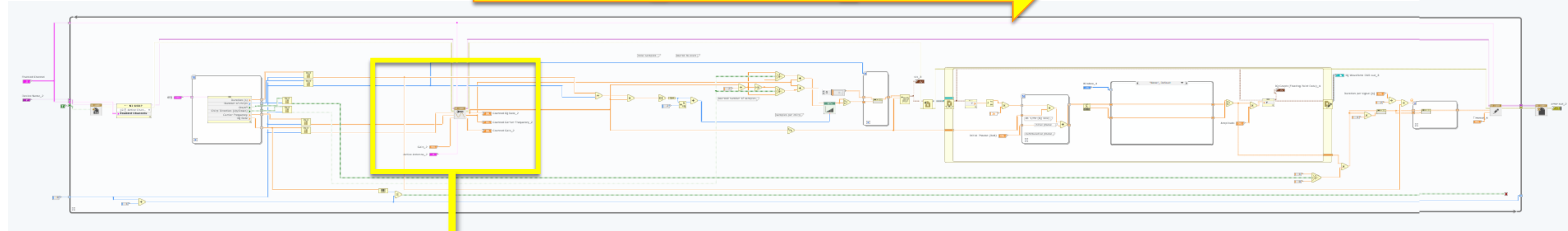
- 본체와 연결된 USRP를 코드와 연동시키고 USRP를 작동on 시키는 부분

# Basic tutorial with Example labview code

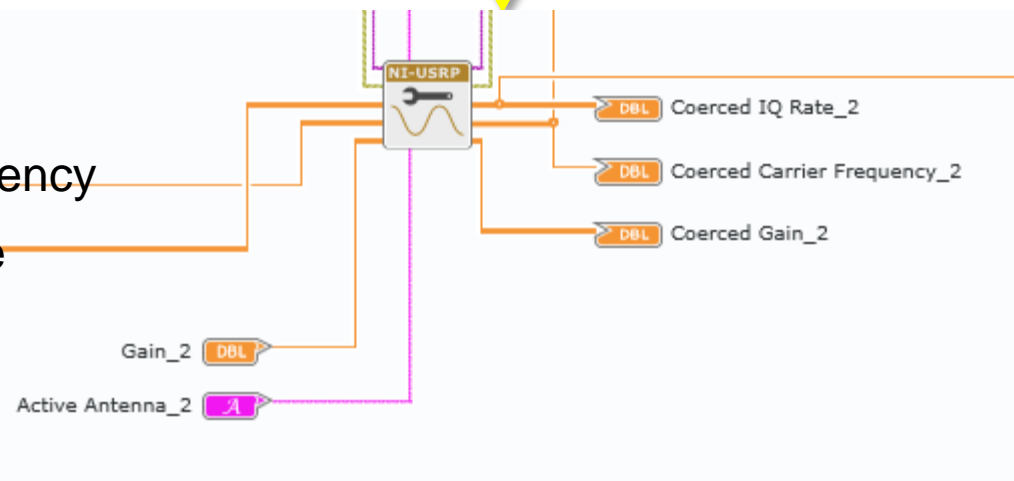
## ❖ USRP 관련 블록

- 자세한 기능은 중요하지 않으나, 코드의 흐름을 해석하기 위해 기본적인 부분은 알면 좋습니다

전체 코드의 흐름

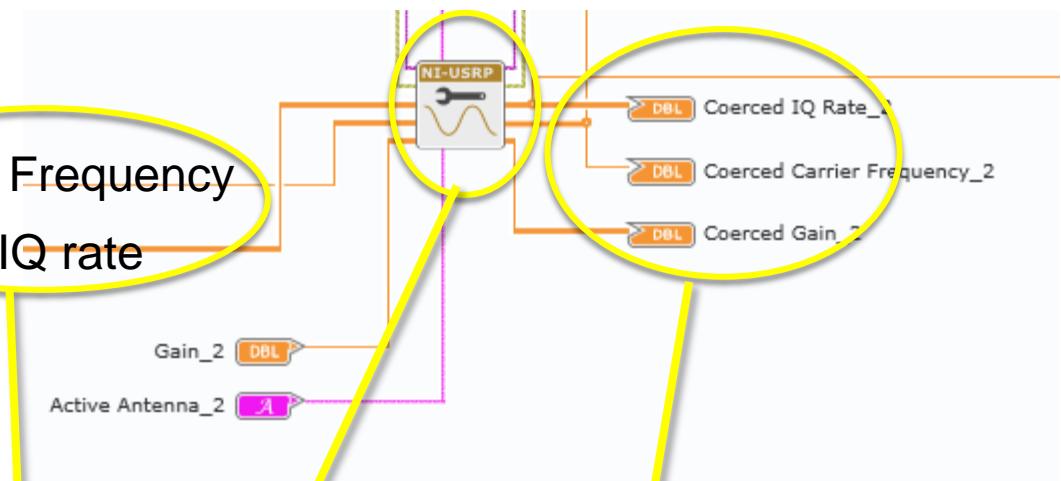


Carrier Frequency  
IQ rate



- USRP 안테나가 작동하기 위한 변수를 입력하는 부분 (패널 창에서 입력)

Carrier Frequency  
IQ rate



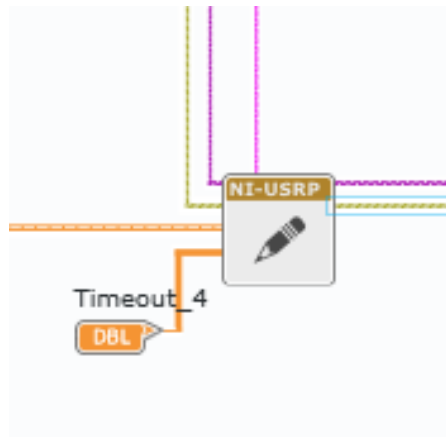
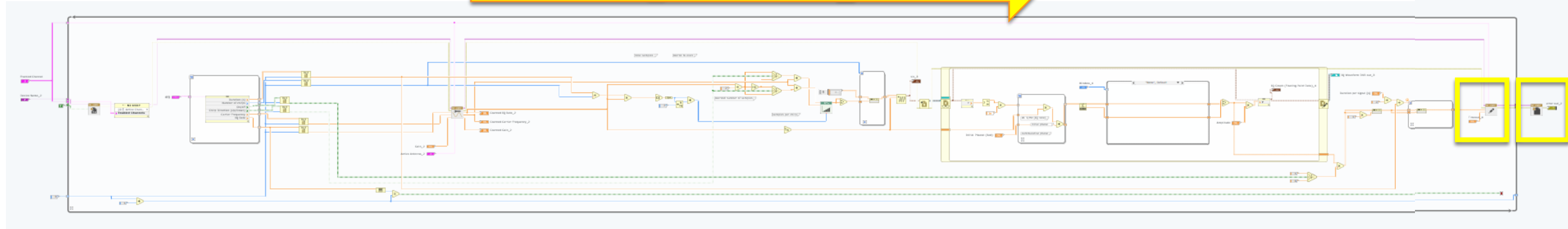
- 여러 입력값 중 USRP에 쓰이는 입력값은 바로 사용하는 것이 아닌
- NI-USRP 블록을 거친 값을 사용한다

# Basic tutorial with Example labview code

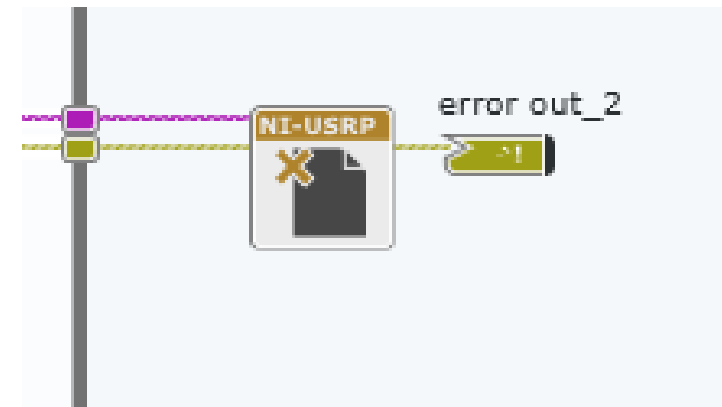
## ❖ USRP 관련 블록

- 자세한 기능은 중요하지 않으나, 코드의 흐름을 해석하기 위해 기본적인 부분은 알면 좋습니다

전체 코드의 흐름



- USRP가 들어오는 데이터를 안테나로 송출



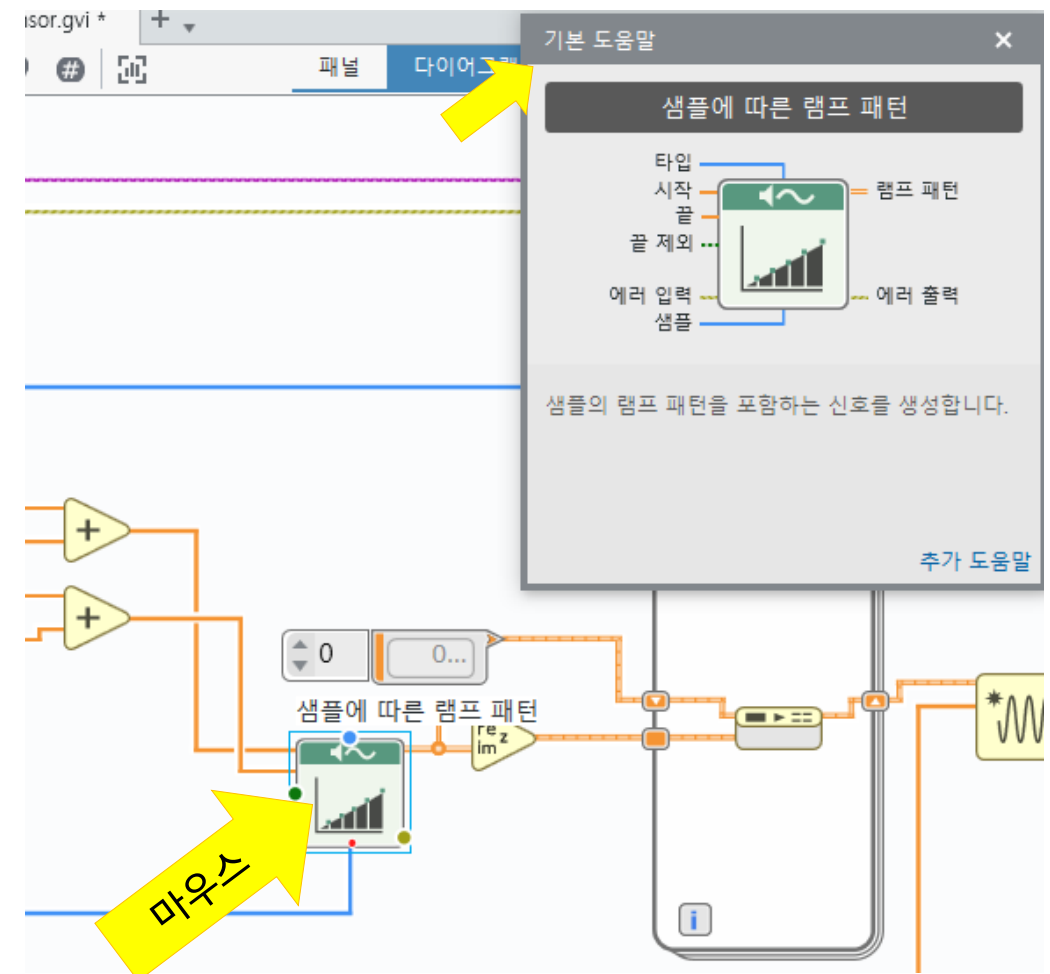
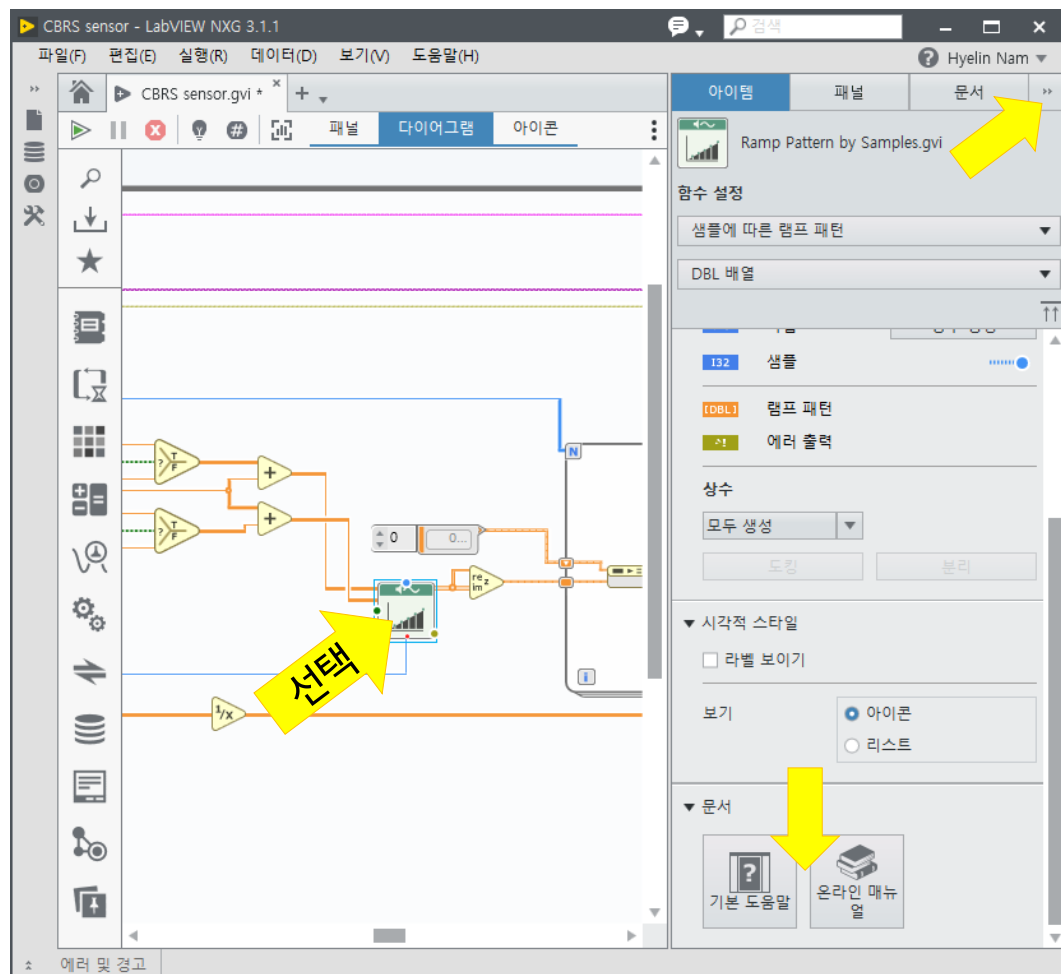
- USRP 작동 off



# Basic tutorial with Example labview code

## ❖ 코드 해석 팁: 도움말

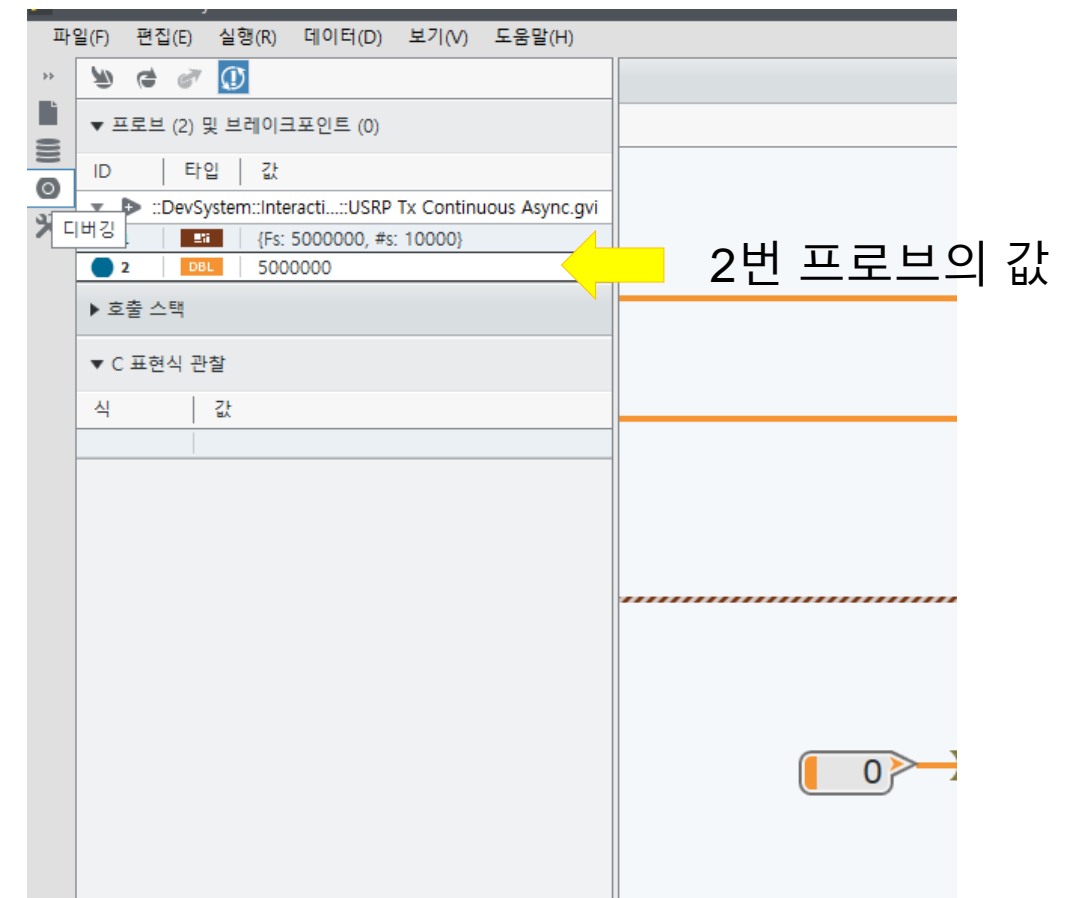
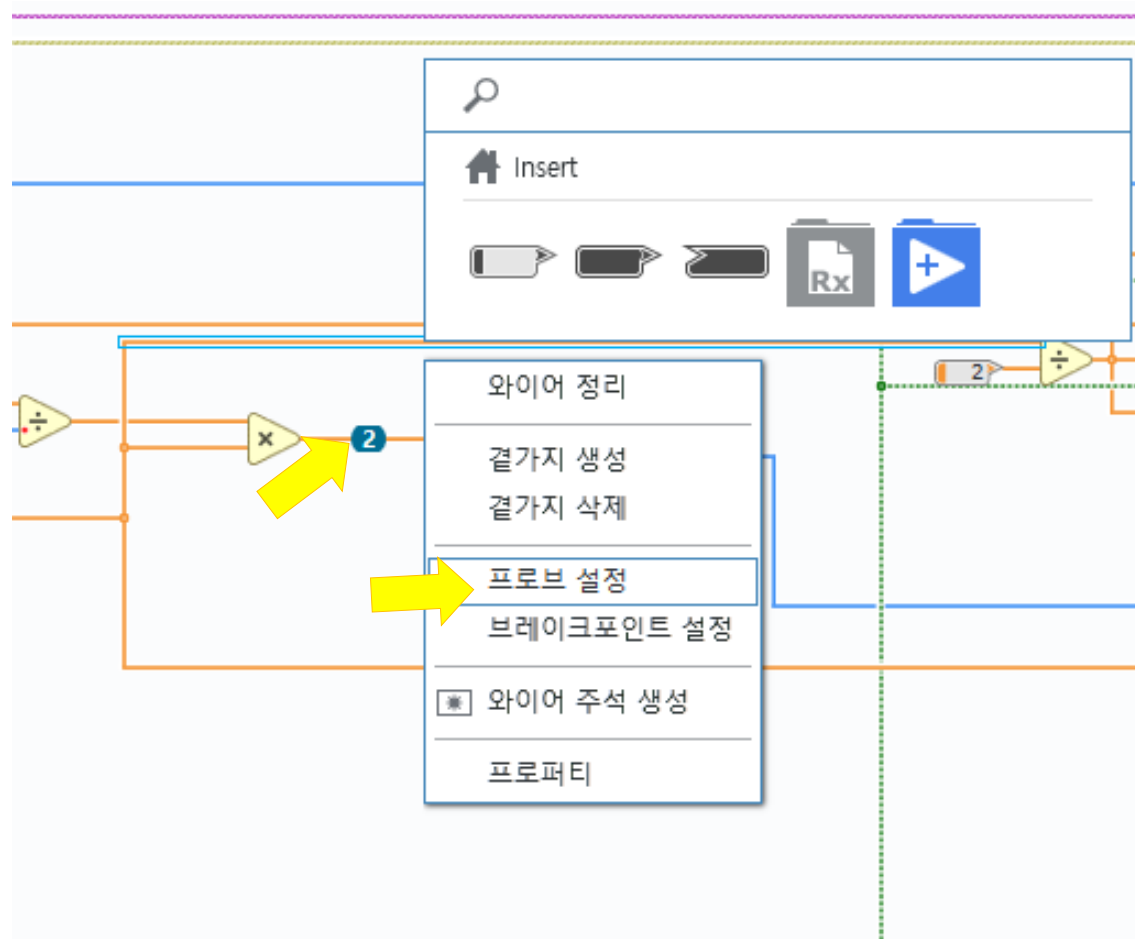
- 블록의 기능, 각 입력과 출력에 대한 설명
- 해당 블록을 선택한 후 화면의 오른쪽위 화살표를 누른뒤 '기본 도움말', 혹은 '온라인 매뉴얼' 을 참고한다
- '기본 도움말' 은 Ctrl+H 를 누르고 마우스를 해당 블록으로 옮기면 간단하게 볼 수 있다



# Basic tutorial with Example labview code

## ❖ 코드 해석 팁: 프로브설정

- 코드 실행 시 각 선(와이어)에 어떠한 값이 흐르는지를 확인하고 싶을 때 사용하는 기능
- 해당 와이어 마우스 우클릭/ 프로브설정 클릭, 와이어에 번호가 나타난다
- 코드 실행 후 창의 왼쪽 위 '디버깅'을 클릭하면 해당 프로브의 값을 확인할 수 있다



❖ 코드에 대한 자세한 해석과 설명은 Week2 수업에서 진행될 것입니다

# Basic tutorial with Example labview code

## ❖ 코드 실행

- 패널창에 인풋값을 입력한 뒤 실행 (입력값 다음장)

② 실행 버튼 클릭  
(실행이 끝나면 다시 초록으로 변한다)

① 적당한 입력값을 입력한다

④ 생성한 신호가 잘 plot 됨을 확인  
(어떤 것을 그래프화(plot) 한것인지 확인하고  
싶으면 더블클릭하여 다이어그램창에서 어느  
블록인지 확인한다)

③ 에러가 없고 에러 팝업이  
뜨지않으면  
잘 작동한것이다

# Basic tutorial with Example labview code

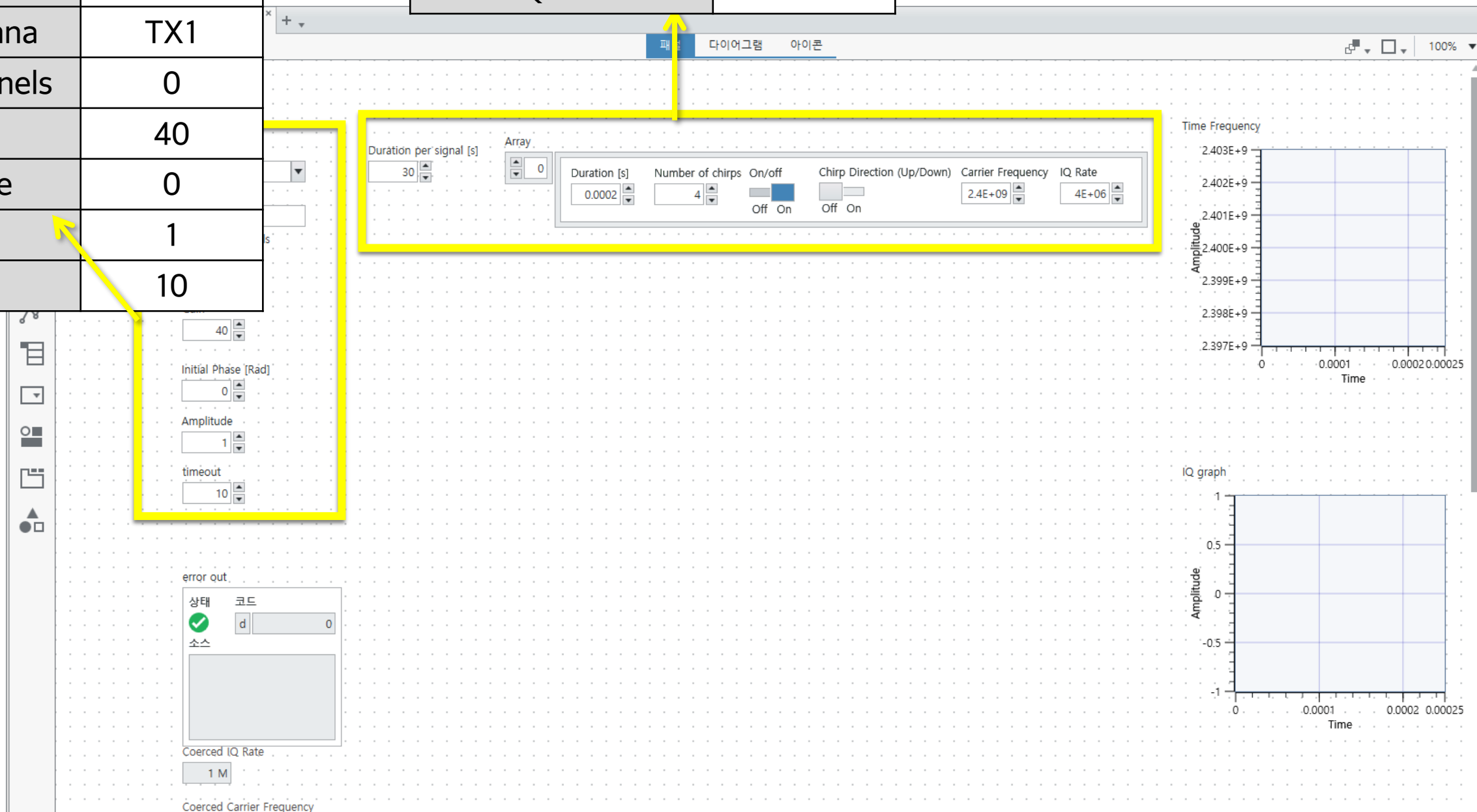
## ❖ 코드 입력값

Device ID	NI2901 (이름이 다를 수는 있으나, 상관없이 목록에 뜨는것 선택)
Active Antenna	TX1
Enabled Channels	0
Gain	40
Initial Phase	0
Amplitude	1
timeout	10

Duration per signal	30
Duration	200u
Num of chirps	4
On/off	On
Chirp Direction	Off(Up)
Carrier Frequency	2.4G
IQ rate	4M

→ 한 Array 입력값에 대해 실행되는 시간 (수정가능)

영어로 단위값을 입력하면(u, G,M)  
저절로 소수점으로  
단위가 바뀜

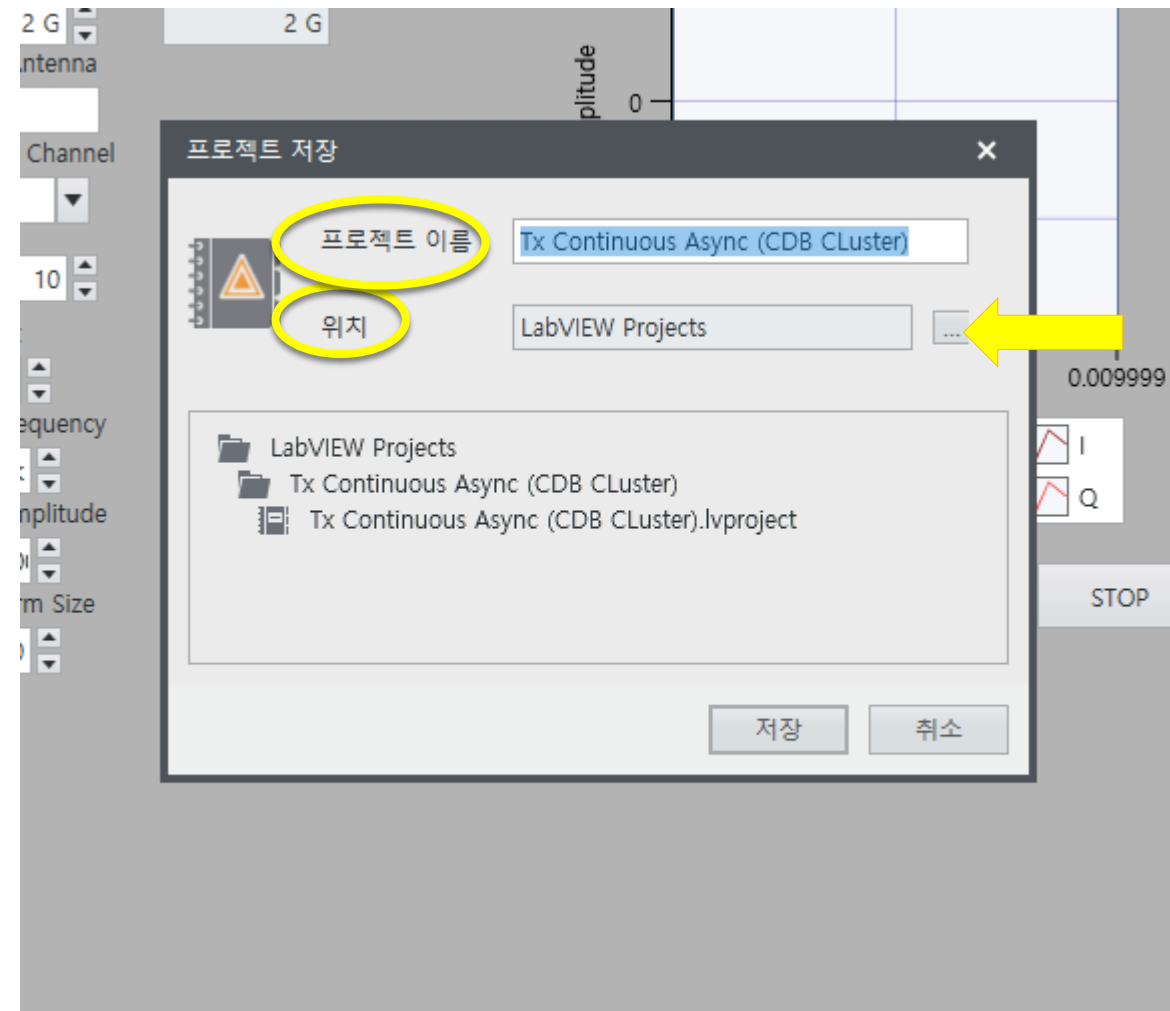
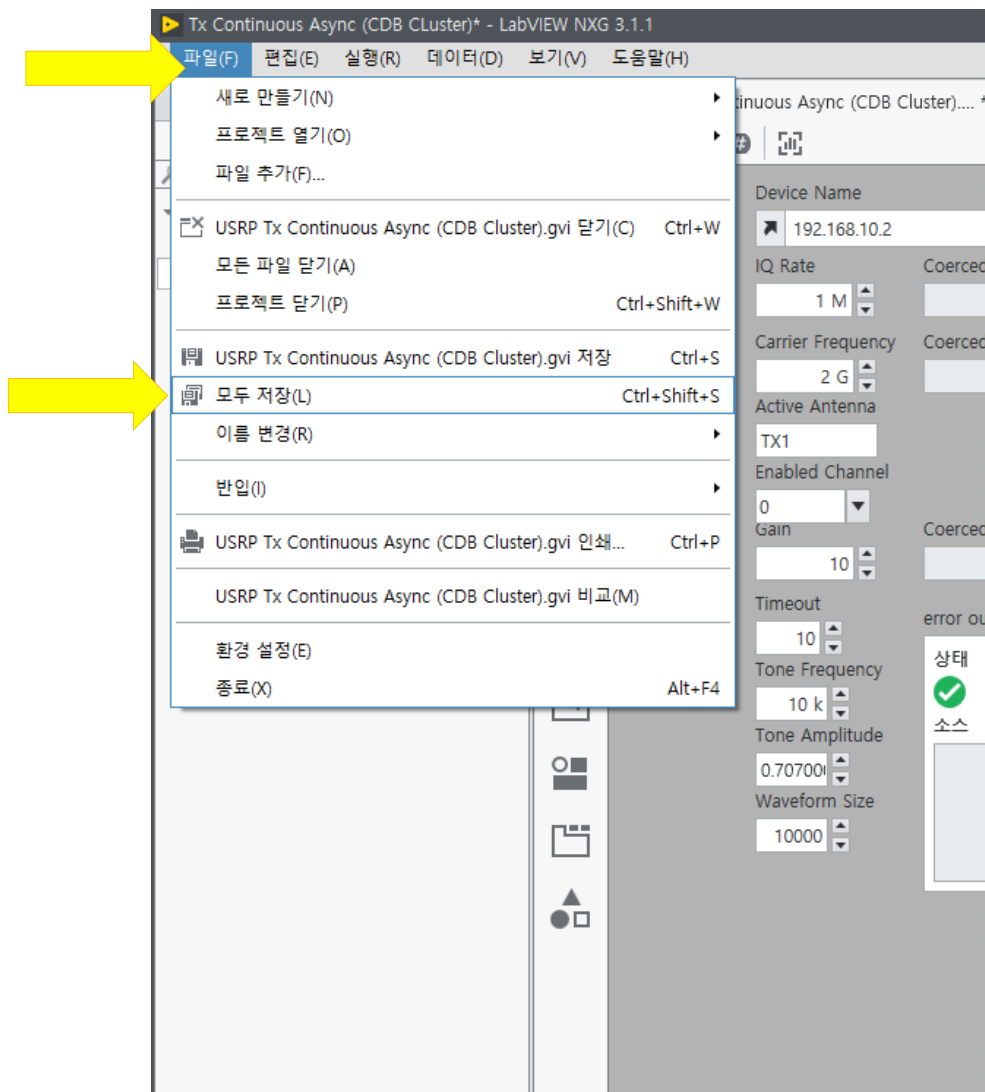


# 프로젝트 저장 방법

## ❖ 파일 > 모두저장

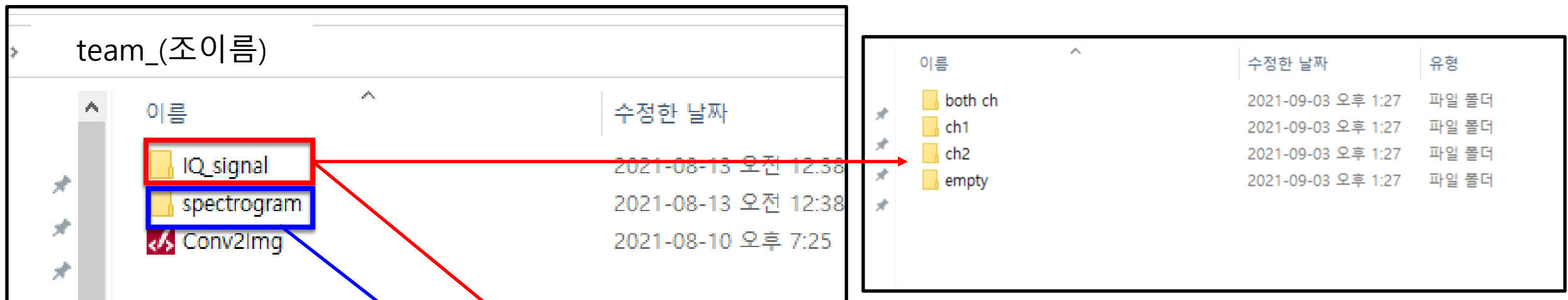
- 프로젝트를 이름을 자유로 바꾸고 (변경 안해도 됨),
- 위치를 오른쪽 버튼을 눌러 team\_(조이름) 폴더로 지정한다

❖ 모두 저장을 눌렀을 때 다음 팝업이 뜨지 않으면 원본 파일에 잘 저장된 것이다



# Python Visualization

- ❖ 우선 Visualization 코드가 동작을 잘하는지 테스트해봅시다.
  - team\_(조이름) (이름 상관없음) 폴더를 생성하여 그 안에 배포한 Visualization code를 저장합니다.
  - 생성한 폴더에 다음과 같이 2개의 하위 폴더를 생성합니다.
  - 배포한 IQ\_signal 데이터를 IQ\_signal 폴더에 저장합니다.

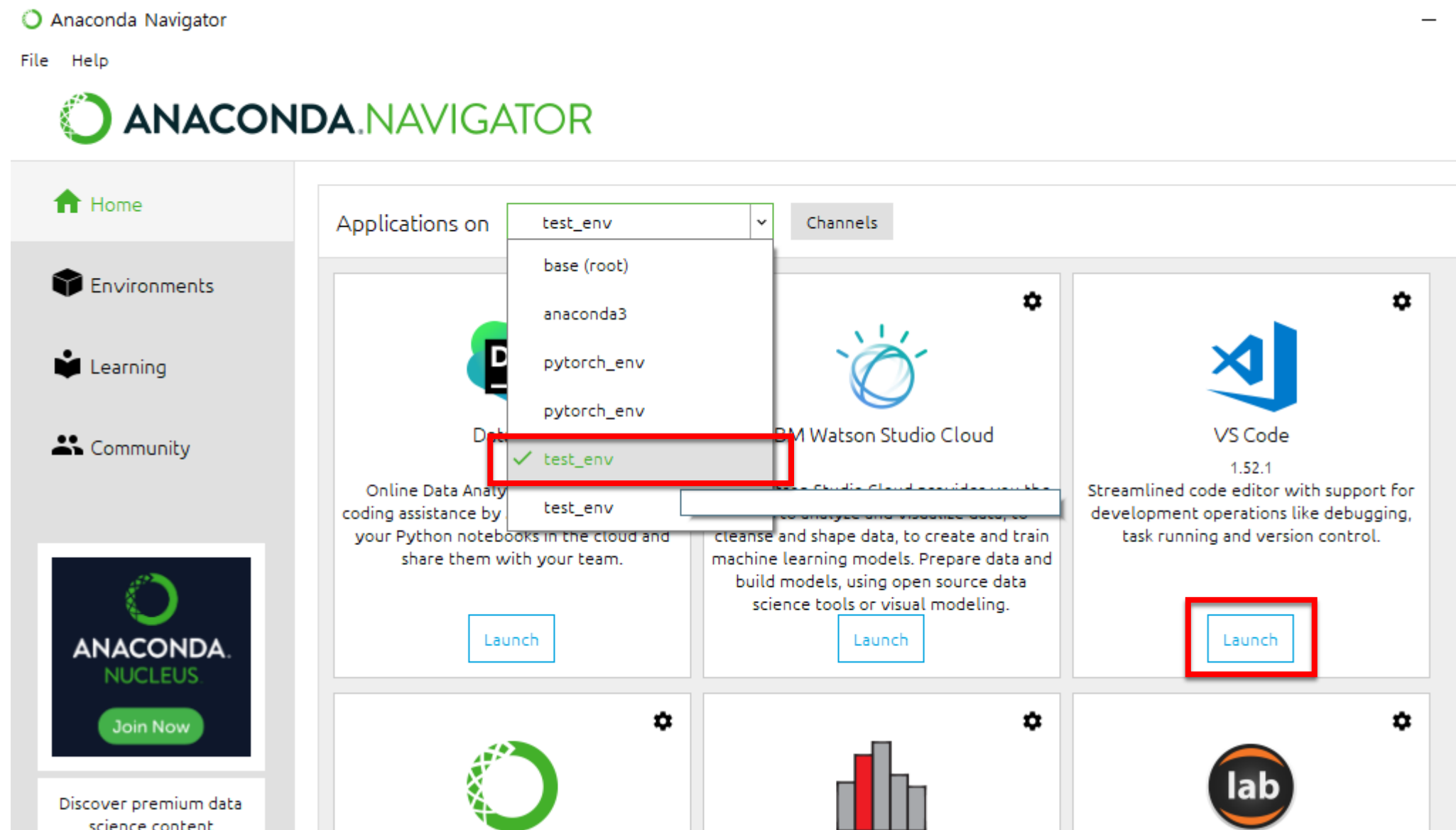


수집한 IQ 데이터  
이미지 데이터를  
저장할 위치

# Python Visualization

## ❖ Anaconda Navigator 실행

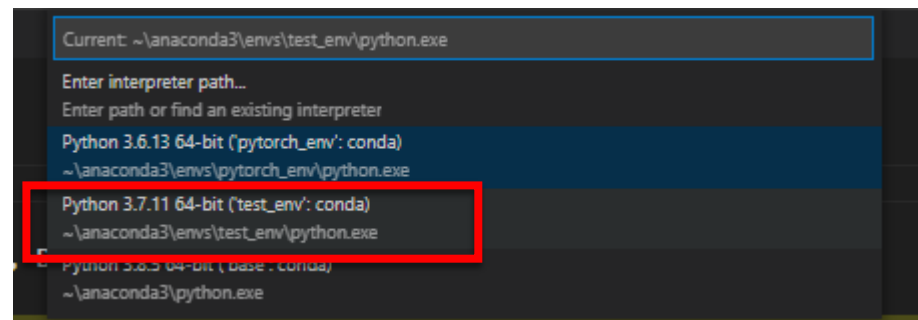
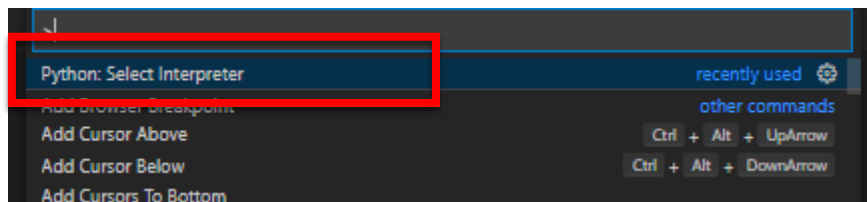
- 본인의 virtual environment로 변경 후 VS code 실행합니다.



# Python Visualization

## ❖ 코드 실행

- Ctrl + Shift + p 입력 후 인터프리터 변경.
- 폴더 이름 확인.



```
1 import torch
2 import numpy as np
3 import torch.nn as nn
4 import torch.nn.functional as F
5 from torch.utils.data import Dataset, DataLoader, random_split
6 import numpy as np
7 import pandas as pd
8 import os
9 from datetime import datetime
10 import matplotlib.pyplot as plt
11 import sys
12 import cv2
13
14 def main():
15     data_root = "./IQ_signal"
16     write_root = "./spectrogram"
17
18     MyDataSet = CustomDataSet(data_root)
19     data_length = MyDataSet.__len__()
20     for i in range(data_length):
21         [datum, target] = MyDataSet.__getitem__(i)
22         name = target[0]
23         cv2.imwrite(write_root + '/' + name[0:-4] + '.png', datum[0])
24
25 class CustomDataSet(torch.utils.data.Dataset):
```

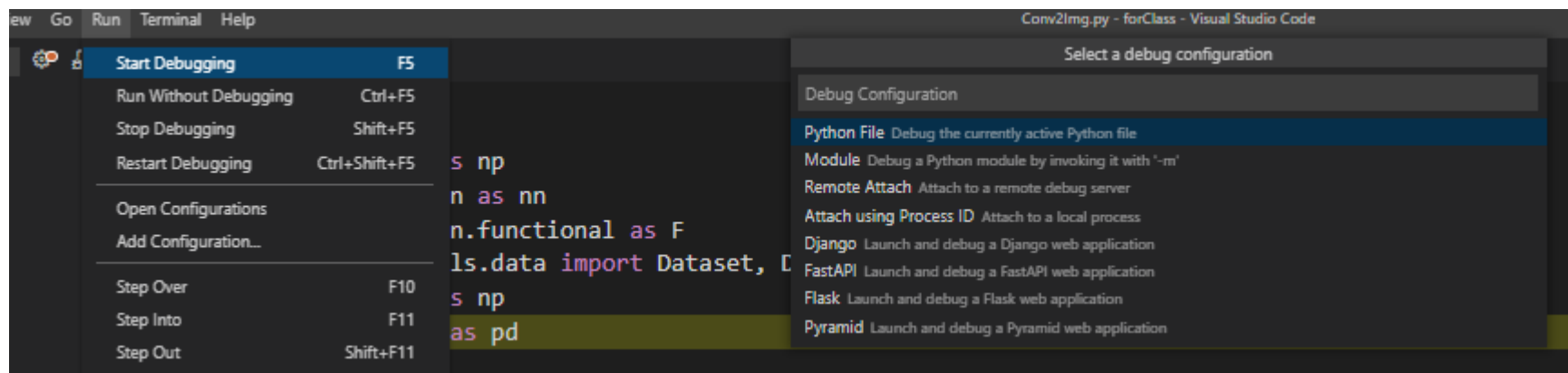


# Python Visualization

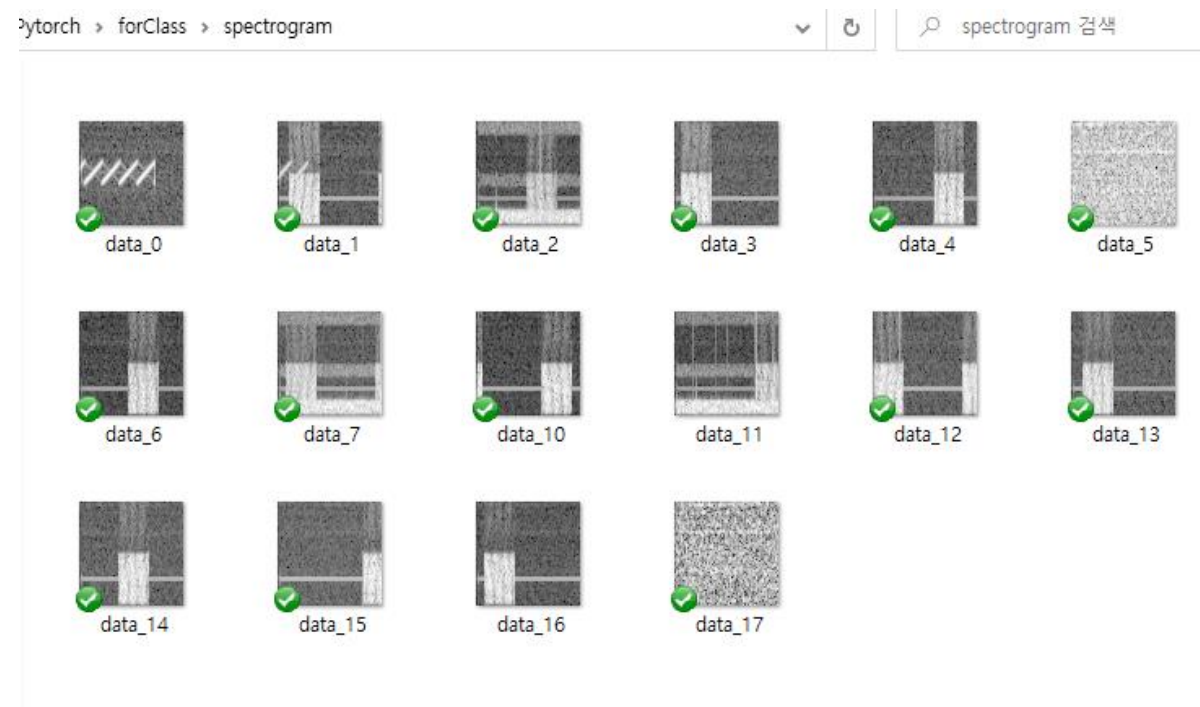
## ❖ 코드 실행

### ■ F5 입력 (Run - Start Debugging)

- Python File 클릭하여 실행.



- Spectrogram 폴더에 다음과 같이 나타나면 성공.



# 과제 (결과 report)

- ❖ **결과리포트 1:** Python visualization 코드 실행
  - 주어진 IQ data를 Visualization하여 제출하십시오.
  - spectrogram 폴더를 압축하여 제출.
  
- ❖ **결과리포트 2:** Labview 코드 실행
  - 예제 코드와 같은 인풋값을 입력하여 실행
  - 입력값과 출력 그래프가 보이도록 패널창 캡처하여 제출하십시오
  
- ❖ **결과리포트 3:** 다른 인풋값으로 Labview 코드 실행
  - **Number of chirps / Chirp Direction / Carrier Frequency/ IQ rate**
    - 각각 하나씩 다른값으로 바꿔보고 각각에 대한 출력이 어떻게 변하는지 간단히 설명하십시오
    - 시각적으로 큰 차이는 없을 수 있으니, 그래프의 **눈금 값**을 관찰해야 합니다.
    - 예시의 인풋값과 크게 다르지 않은 값을 입력해야 동작합니다  
(Mega, Giga등의 단위가 변하면 안됨)
    - Carrier Frequency  $\in (2.39\text{G}, 2.41\text{G})$  , IQ rate  $\in (1\text{M}, 15\text{M})$
  - **각각에 대한 입력값, 출력 그래프, 설명을 제출하십시오**

# 예비 보고서

---

## ❖ 2주차 예비보고서

1. Radar로 많이 쓰이는 Chirp 신호에 대해 조사하시오.
2. 한국의 5G 특화망 정책에 대해 조사하시오.