



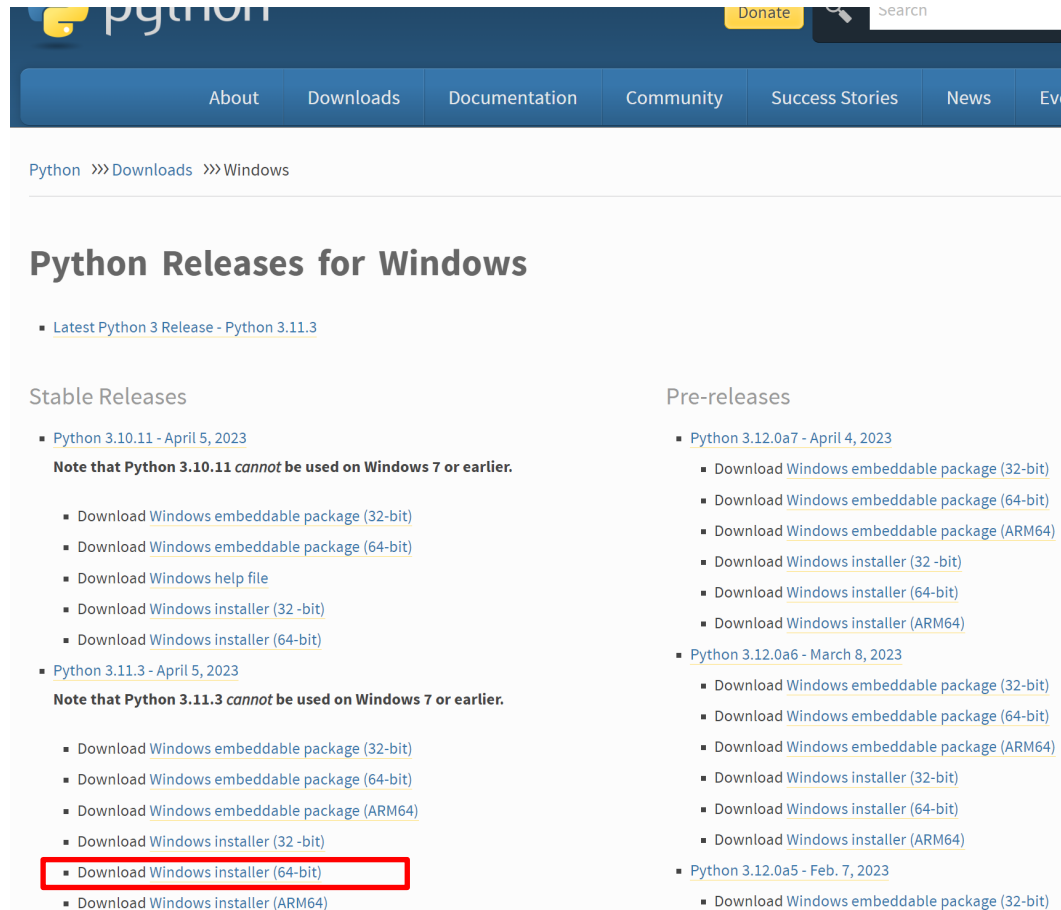
# Python을 이용한 EEboard 제어 및 실습

**Shinwoong Kim**

# Python

## • Python 설치

- ✓ <https://www.python.org/>
- ✓ Downloads → Windows (OS 및 32bit/64bit을 적절히 선택)



python Donate Search

About Downloads Documentation Community Success Stories News Events

Python >>> Downloads >>> Windows

### Python Releases for Windows

- Latest Python 3 Release - Python 3.11.3

#### Stable Releases

- Python 3.10.11 - April 5, 2023  
**Note that Python 3.10.11 cannot be used on Windows 7 or earlier.**
  - Download [Windows embeddable package \(32-bit\)](#)
  - Download [Windows embeddable package \(64-bit\)](#)
  - Download [Windows help file](#)
  - Download [Windows installer \(32-bit\)](#)
  - Download [Windows installer \(64-bit\)](#)
- Python 3.11.3 - April 5, 2023  
**Note that Python 3.11.3 cannot be used on Windows 7 or earlier.**
  - Download [Windows embeddable package \(32-bit\)](#)
  - Download [Windows embeddable package \(64-bit\)](#)
  - Download [Windows embeddable package \(ARM64\)](#)
  - Download [Windows installer \(32-bit\)](#)
  - Download [Windows installer \(64-bit\)](#)
  - Download [Windows installer \(ARM64\)](#)

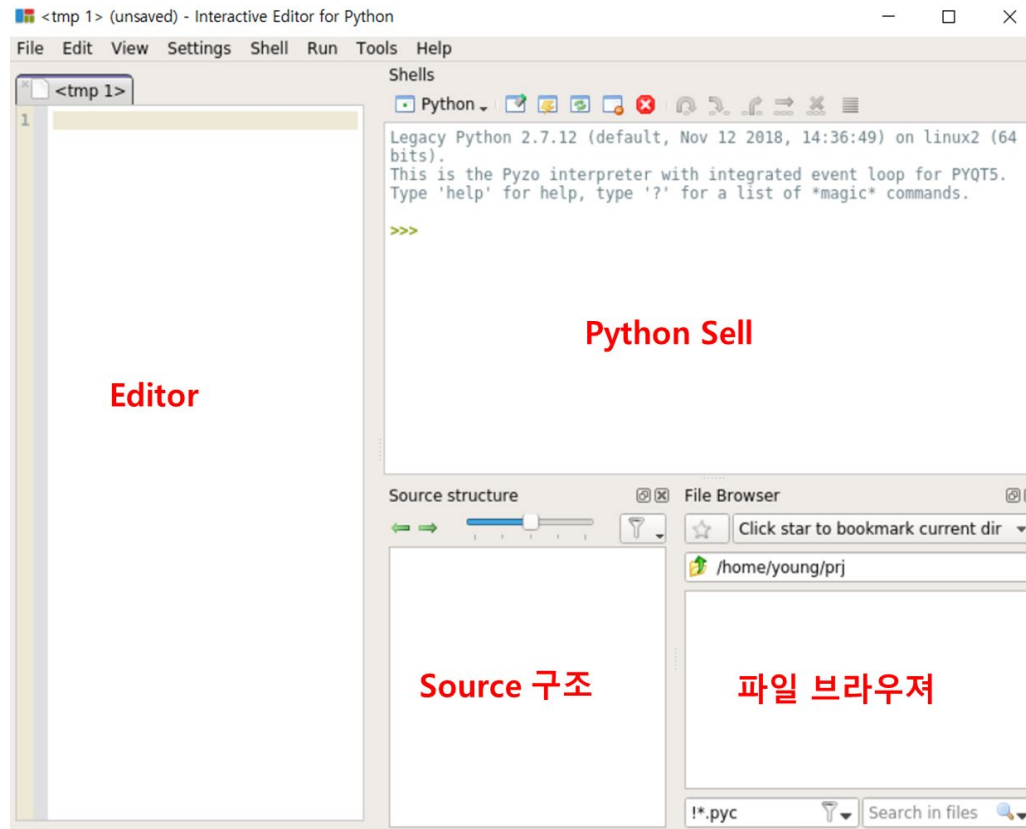
#### Pre-releases

- Python 3.12.0a7 - April 4, 2023
  - Download [Windows embeddable package \(32-bit\)](#)
  - Download [Windows embeddable package \(64-bit\)](#)
  - Download [Windows embeddable package \(ARM64\)](#)
  - Download [Windows installer \(32-bit\)](#)
  - Download [Windows installer \(64-bit\)](#)
  - Download [Windows installer \(ARM64\)](#)
- Python 3.12.0a6 - March 8, 2023
  - Download [Windows embeddable package \(32-bit\)](#)
  - Download [Windows embeddable package \(64-bit\)](#)
  - Download [Windows embeddable package \(ARM64\)](#)
  - Download [Windows installer \(32-bit\)](#)
  - Download [Windows installer \(64-bit\)](#)
  - Download [Windows installer \(ARM64\)](#)
- Python 3.12.0a5 - Feb. 7, 2023
  - Download [Windows embeddable package \(32-bit\)](#)

# Python 개발 환경

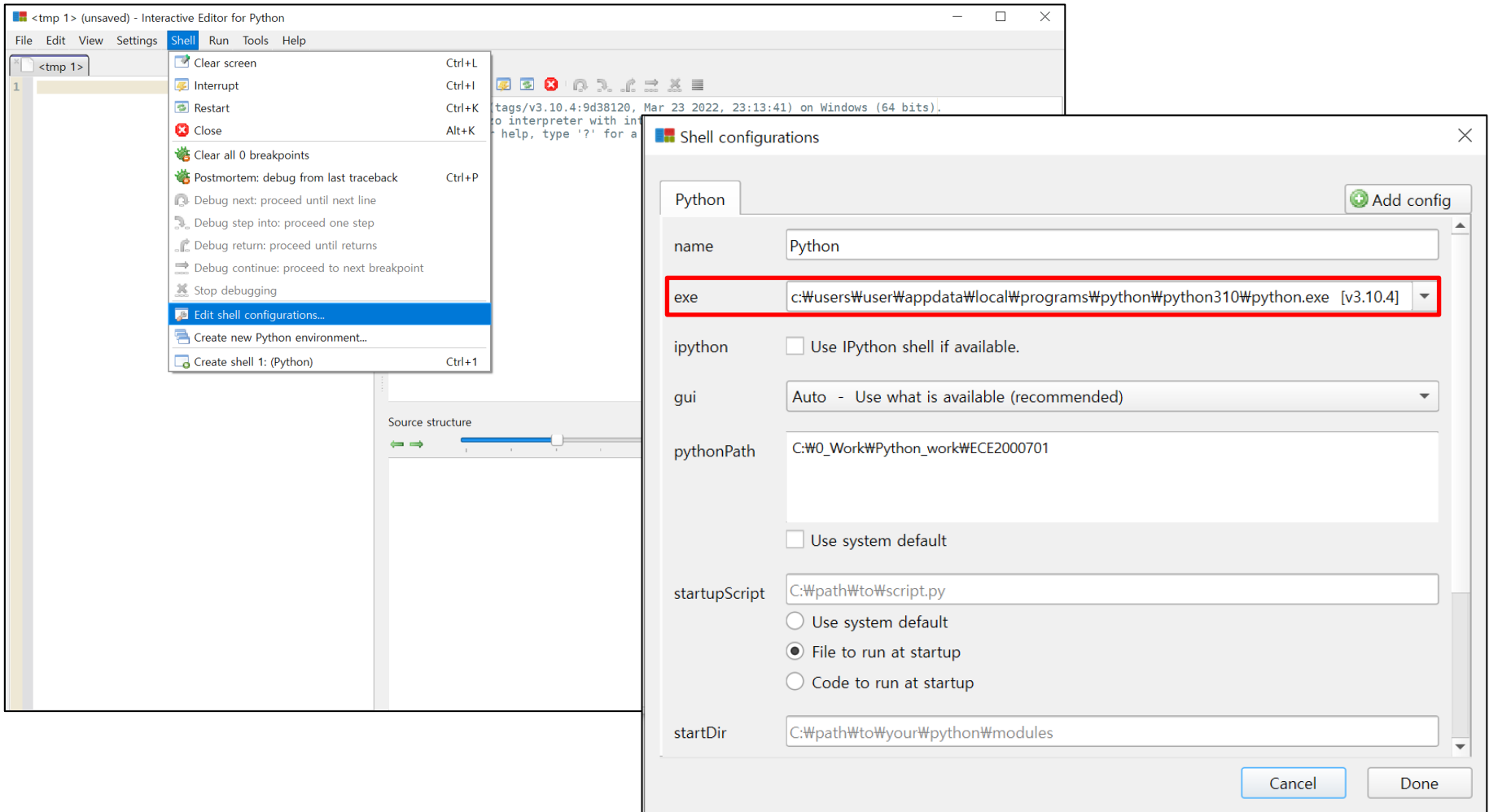
- Pyzo

- ✓ 개발 및 debugging 을 체계적으로 관리할 수 있는 통합 환경(IDE) tool이 다양하게 제공
- ✓ 본 실습에서는 Pyzo를 사용함 (<https://pyzo.org/>)



# Python 개발 환경

- 기본적으로 Python v3 이상 지원 가능
  - ✓ Shell -> Edit shell configurations...



# Python Package for EEBoard

- Python class package

- ✓ Digilent 사의 Waveforms를 설치하면 EEboard를 제어하기 위한 SDK는 자동으로 설치가 되고, EEboard를 제어하려면 SDK를 이용하여 필요한 Python Class를 설계 해야함

```
|-- Readme.txt : this file                                     https://github.com/young1329/EEboard
|-- eeboard
|   |-- __init__.py : package init file
|   |-- awg.py : AWG object
|   |-- device.py : Electronic Explorer board base class
|   |-- digitalio.py : digital static io object
|   |-- dwfconstants.py : constants provided from Digilentinc
|   |-- logic.py : logic analyzer object
|   |-- pattern.py : digital pattern generation object
|   |-- power.py : Power supply and voltagemeter boject
|   |-- scope.py : analog scope object
|-- examples
    |-- 01_Device.py : check out the device connection.
    |-- 02_Power.py : Vref1, Vref2, VP+ and Voltage meters are tested
    |-- 03_AWG_Scope.py: AWG and Scope test
    |-- 04_DigitalIO.py : Static digital IO test
    |-- 05_Pattern_Logic03.py : 4bit counter and meausre with 8-bit 1000 samples with Single Acquisition Mode
    |-- 06_Pattern04.py D22 and D21 as output terminal, 1khz clock 50% duty output
    |-- 07_Pattern
```

- ✓ eeboard: EEboard제어 클래스 정의 패키지
- ✓ Example: 패키지를 시험하기 위한 test-bench 파일
- ✓ 본 실습에서는 device.py와 power.py class만을 사용할 예정

# Python Package for EEBoard

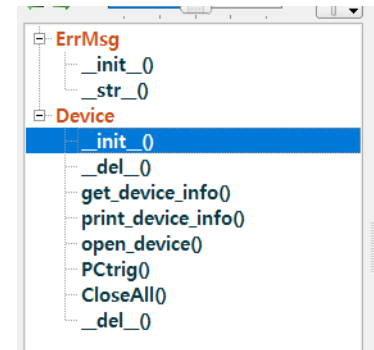
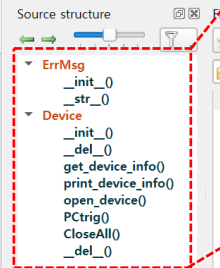
- Device class

✓ EEboard 제어를 위한 모든 클래스의 base class가 됨

```
device.py (C:\WO_Work\Python_work\ECE2000701W\device.py) - Interactive Editor for Python
File Edit View Settings Shell Run Tools Help

1  ...
2  EEBoard control based on Python
3  Coded by Youngsik Kim @Handong University
4  Updated to V06 @2018.09.12
5  ... adding destruction
6  ... The object should be del at the end to clear out the device handler
7  ...
8
9  from dwfconstants import *
10 from ctypes import *
11 import time
12 dwf = cdll.dwf
13
14 class ErrMsg(Exception):
15     def __init__(self, msg):
16         self.msg = msg
17     def __str__(self):
18         return self.msg
19
20 # Device class : System, Device group : detection, get info open / close
21 class Device():
22     idDev=[];
23     hdwf=[]; cDevice=[]; szDevName=[]; szSerialNum=[]; szLabel=[]; szName=[]; szUnits=[];
24     szVersion = create_string_buffer(16)
25     def __init__(self, idx=0):
26         if ( not(idx in Device.idDev) ):
27             Device.idDev.append(idx)
28             Device.hdwf.append(c_int(idx))
29             Device.cDevice.append(c_int(1))
30             Device.szDevName.append(create_string_buffer(64))
31             Device.szSerialNum.append(create_string_buffer(16))
32             Device.szName.append(create_string_buffer(32))
33             Device.szLabel.append(create_string_buffer(16))
34             Device.szUnits.append(create_string_buffer(16))
35         else :
36             pass
37
38     def __del__(self):
39         for idx in Device.idDev:
40             dwf.FDwfDeviceClose(Device.hdwf[idx])
41             print("Device %d is closed\n"%(idx))
42
43     def get_device_info(self, idx=0):
44         if ( idx in Device.idDev ):
45             dwf.FDwfEnum(c_int(idx), byref(Device.cDevice[idx])) # Get the first(0) EEboard handler
46             dwf.FDwfEnumDeviceName(c_int(idx), Device.szDevName[idx]) # Get the EEboard Name
47             dwf.FDwfEnumSN(c_int(idx), Device.szSerialNum[idx]) # Get the seiral number
48             dwf.FDwfGetVersion(Device.szVersion) # Get the version info
49
50     def print_device_info(self):
51         for idx in Device.idDev:
52             print('.....')
53             #print('Device %d'%(idx)+' : '+Device.szDevName[idx].value+'\t'+Device.szSerialNum[idx].value)
```

Shells  
Python 3.10.4 (tags/v3.10.4  
This is the Pyzo interpreter  
Type 'help' for help, type  
>>>

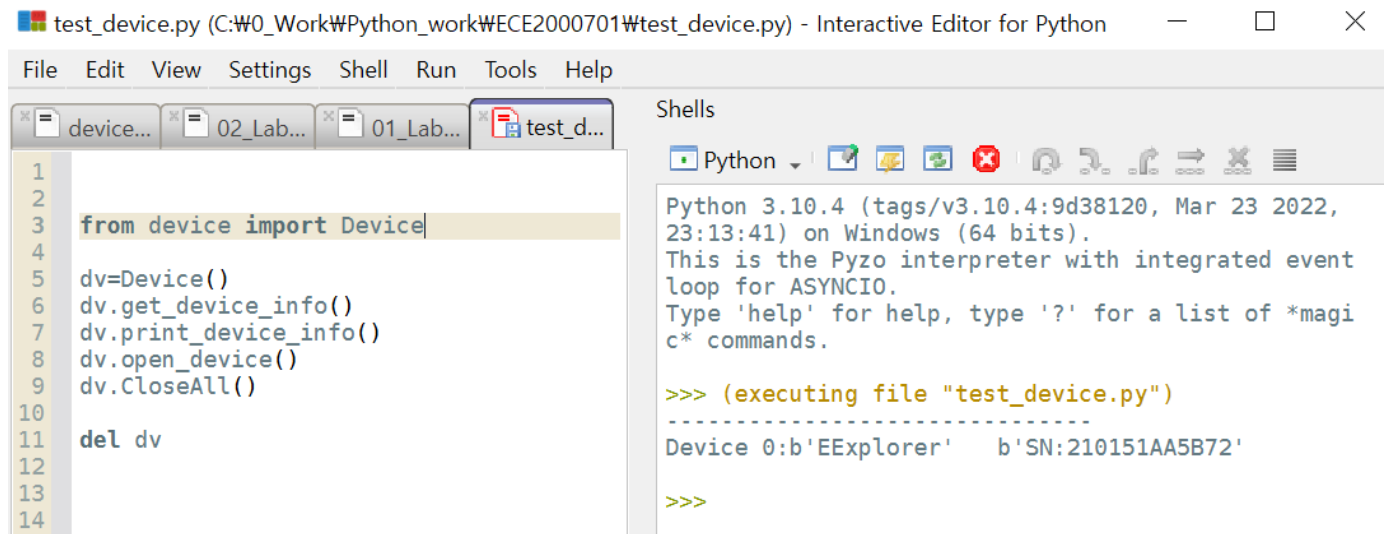


Device class의 구조

# Python Package for EEBoard

- **Example: Device class**

- ✓ 해당 코드는 EEboard에서 Device 객체를 import한 후 dv 객체를 생성
- ✓ 그리고 dv.get\_device\_info()로 EEboard의 정보를 가져와 dv.print\_device\_info() 메서드로 EEboard의 정보를 출력함



The screenshot shows the 'Interactive Editor for Python' window. The left pane displays a Python script in 'test\_device.py' with the following code:

```
1
2
3 from device import Device
4
5 dv=Device()
6 dv.get_device_info()
7 dv.print_device_info()
8 dv.open_device()
9 dv.CloseAll()
10
11 del dv
12
13
14
```

The right pane, titled 'Shells', shows the output of running the script. It displays the Python version and environment information, followed by the execution of the script and the resulting device information:

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) on Windows (64 bits).
This is the Pyzo interpreter with integrated event loop for ASYNCIO.
Type 'help' for help, type '?' for a list of *magi c* commands.

>>> (executing file "test_device.py")
-----
Device 0:b'EEExplorer'    b'SN:210151AA5B72'

>>>
```

# Python Package for EEBoard

- Power class

- ✓ Device 클래스를 상속 하여 다음을 제어
  - 전원공급 Vcc, VP+, VP- 포트
  - 기준 전압을 출력하는 Vref1, Vref2 포트
  - 전압을 측정하는  $V_{MTR1} \sim V_{MTR4}$  포트

```
from device import *

channels = {'Vcc':0, 'VP+':1, 'VP-':2, 'Vref1':3, 'Vref2':4,
           'Vmtr1':5, 'Vmtr2':6, 'Vmtr3':7, 'Vmtr4':8}
nodes = {'Enable':0, 'Voltage':1, 'Current':2, 'Power':3}

class Power(Device):
    def __init__(self, idx=0):
        self.idx = idx;
        self.nChannel = c_int()
        self.szChannel = create_string_buffer(32)
        self.szLabel = create_string_buffer(16)
        self.szNode = create_string_buffer(32)
        self.szUnits = create_string_buffer(16)
        #Current for power supply
        self.crntVP = c_double()
        self.crntVN = c_double()
        self.crntVCC = c_double()

        Device.__init__(self, idx)
        #Vmtr Variables
        self.Vmtr = [c_double(), c_double(), c_double(), c_double()]
        #print ('pwr idx is %d'%self.idx)
```

```
Power
  __init__
  #####
  #####
  reset_analogIO()
  get_number_of_channels()
  get_nodes_of_channels()
  what_is_channel_node()
  enable_channel()
  disable_channel()
  set_channel_voltage()
  set_channel_current()
  analogIO_configure()
  analogIO_ON()
  analogIO_OFF()
  measure_vmtr()
  get_vmtr()
  get_crntVP()
  get_crntVN()
  get_crntVCC()
```

Power class의 구조



# Python Package for EEBoard

## • Example: Power class

✓ 해당 코드는 Vref1, Vref2, VP+, Vcc 포트를 제어하여 전압을 출력함

```
01_Lab09_Power_example.py device.py power.py
11
12     Vmtr1, Vmtr2, Vmtr3, Vmtr4 : measure and print the v
13     ... You can check it by connecting Vmtr1 --> Vref1, Vmtr2
14     ...
15
16
17 from power import Power
18
19 import time
20
21 pwr=Power()
22
23 pwr.get_device_info()
24 pwr.print_device_info()
25
26 pwr.open_device()
27 pwr.reset_analogIO()
28 print('=====\\n')
29
30
31 #get the number of channels for AnalogIO
32 pwr.get_number_of_channels()
33 # get the number of node for channel 1
34 pwr.get_nodes_of_channels(1)
35 # figure out channel 1, and node 2
36 pwr.what_is_channel_node(1,2)
37 print('=====\\n')
38
39
40 # Configure VP+=5.0V with 50mA current
41 pwr.set_channel_voltage('Vref1',4.0)
42 pwr.set_channel_voltage('Vref2',-3.0)
43 pwr.set_channel_voltage('VP+',5.0)
44 pwr.set_channel_current('VP+',50e-3)
45 pwr.set_channel_voltage('Vcc',3.3)
46
47 pwr.enable_channel('Vref1')
48 pwr.enable_channel('Vref2')
49 pwr.enable_channel('VP+')
50 pwr.enable_channel('Vcc')
51
52 pwr.analogIO_ON()
53
54 time.sleep(1)
55
56 pwr.measure_vmtr()
57
58 pwr.analogIO_OFF()
59
60
61 print('Vmtr1=%.2f \\n'%(pwr.get_vmtr(0)))
62 print('Vmtr2=%.2f \\n'%(pwr.get_vmtr(1)))
63 print('Vmtr3=%.2f \\n'%(pwr.get_vmtr(2)))
64 print('Vmtr4=%.2f V'%(pwr.get_vmtr(3)))
65
66 del pwr
```

✓ EEboard에서 아래 같이 연결 하는 것이 필요

- Vref1 →  $V_{MTR1}$
- Vref2 →  $V_{MTR2}$
- VP+ →  $V_{MTR3}$
- Vcc →  $V_{MTR4}$

>>> (executing file "01\_Lab09\_Power\_example.py")

-----  
Device 0:b'EEExplorer' b'SN:210151AA5B72'

-----  
Numberof Channel is 9

Number of nodes of the channel(1, b'Positive Supply', b'VP+') is 4

Channel(1): b'Positive Supply', b'VP+'

Node(2) : b'Current', b'A'

-----  
Vmtr1=4.01 V

Vmtr2=-3.09 V

Vmtr3=4.99 V

Vmtr4=3.33 V

실행 결과

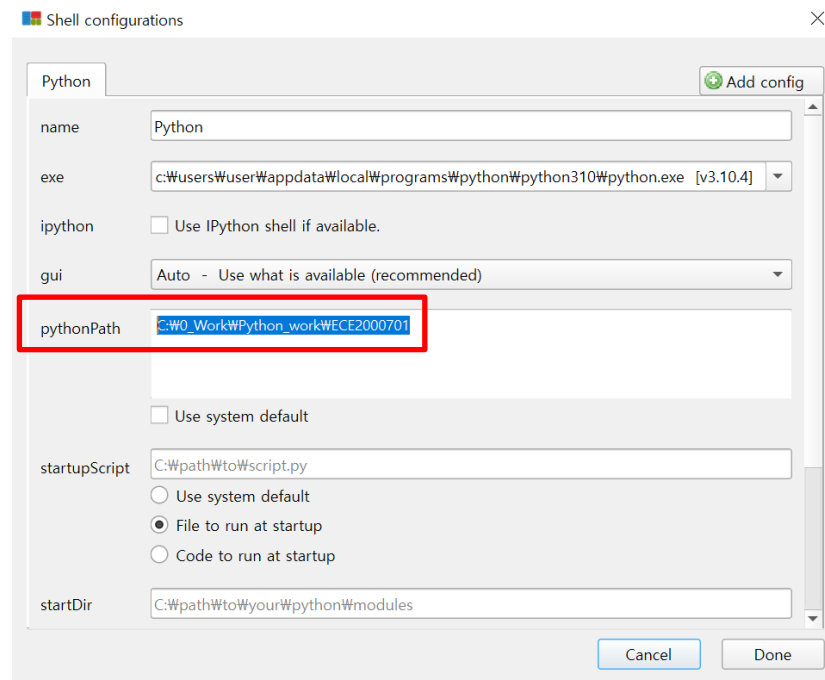
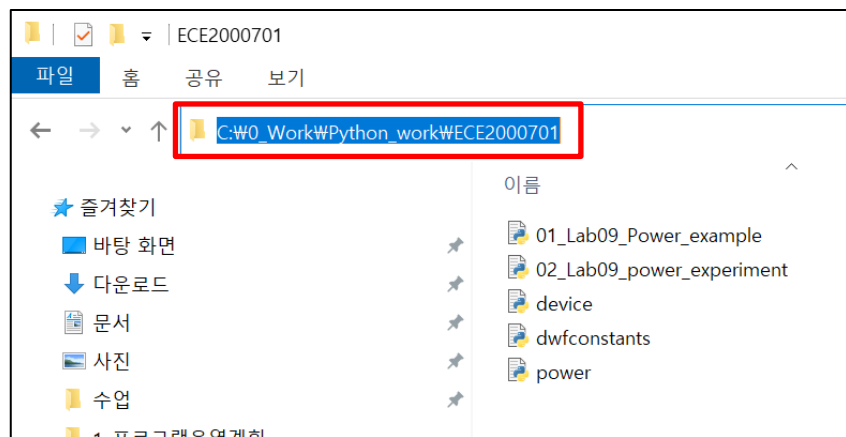
# 실습 1

- 01\_Lab09\_Power\_example.py

- ✓ 해당 python 코드를 그대로 실행하여 EEboard가 제어되는지 확인

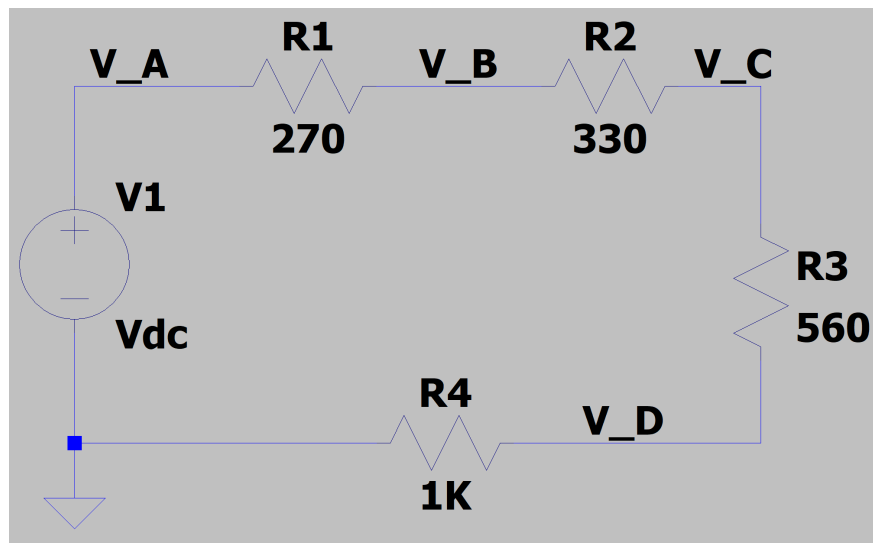
- [참고]

- ✓ Python 코딩을 수행하는 작업 폴더의 경로를 'pythonPath'에 설정 필요
    - Pyzo tool에서 Shell -> Edit shell configurations에서 확인



## 실습 2

- 다음과 같이 회로를 꾸미고 Python으로 EEboard를 제어
  - ✓ 전압 소스로는 VP+를 사용하고, 4개의 저항으로 직렬 연결 회로 구성
  - ✓ 각 node의 이름을 V\_A, V\_B, V\_C, V\_D로 정의하고 Volt meter에 연결
    - $V_A \rightarrow V_{MTR1}$ ,  $V_B \rightarrow V_{MTR2}$
    - $V_C \rightarrow V_{MTR3}$ ,  $V_D \rightarrow V_{MTR4}$
  - ✓ VP+의 전압이 1.0V, 3.0V, 5.0V 각각 세가지 경우에서
    - 모든 node의 전압을 측정 및 이를 그래프로 표현하기



# 실습 2

```
02_Lab09_power_experiment.py
1  """
2  Lab is based on the Code by Youngsik Kim @ CSEE . HGU
3  Test the Power supply and Voltage meter function
4  """
5
6
7  from power import Power
8  import time
9  import matplotlib.pyplot as plt
10
11  pwr=Power()
12
13  pwr.get_device_info()
14  pwr.print_device_info()
15
16  pwr.open_device()
17  pwr.reset_analogIO()
18  print('=====\\n')
19
20
21  # Configure VP+ with 50mA current limit
22  pwr.set_channel_current('VP+',50e-3)
23  pwr.enable_channel('VP+')
24  pwr.analogIO_ON()
25
26  # Vraible declaration
27  Vdc_list = [1.0, 3.0, 5.0]
28  V_A_list = []
29  V_B_list = []
30  V_C_list = []
31  V_D_list = []
32
33  # Experiment
34  print('*** Experiment****')
35  for vdc in Vdc_list:
36      print('Vdc=%.2f (V)'%(vdc))
37      pwr.set_channel_voltage('VP+',vdc)
38      time.sleep(0.5)
39      pwr.measure_vmtr()
40
41      V_A_list.append(pwr.get_vmtr(0))
42      V_B_list.append(pwr.get_vmtr(1))
43      V_C_list.append(pwr.get_vmtr(2))
44      V_D_list.append(pwr.get_vmtr(3))
45
46  pwr.analogIO_OFF()
47  print('=====\\n')
48  print('*** Finished****')
49  del pwr
50
51  # Data representation
52  plt.plot(Vdc_list, V_A_list, 'o--', Vdc_list, V_B_list, 'o--', Vdc_list, V_C_list, 'o--', Vdc_list, V_D_list, 'o--')
53  plt.xlim([1,5])
54  plt.ylim([0,5])
55  plt.xlabel('Vdc(V)')
56  plt.ylabel('Node voltage(V)')
57  plt.legend(('V_A', 'V_B', 'V_C', 'V_D'))
58  plt.grid()
59  plt.show()
```

## <실행 결과>

```
>>> (executing file "02_Lab09_power_experiment.py")
-----
Device 0:b'EExplorer'   b'SN:210151AA5B72'
=====

*** Experiment****
Vdc=1.00 (V)
Vdc=3.00 (V)
Vdc=5.00 (V)
=====

*** Finished****
```

