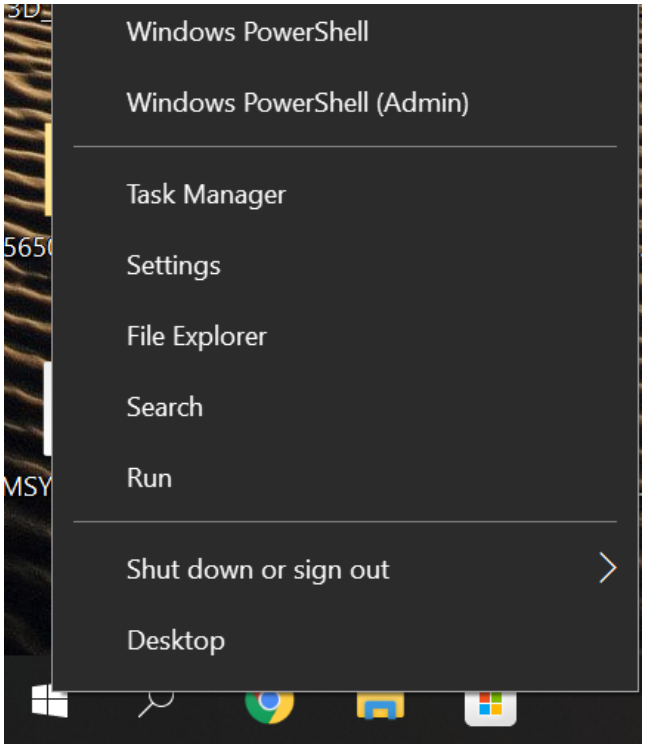
Using and Creating a New Python 3.9 Virtual Environment Under *Miniconda*

This documents explains how to use and create new Python virtual environments using `conda` and at the end of the document it explains how to install `pyaudio_helper` in Win10 by manually downloading and installing a Python `*.whl` file. The virtual environment works the same regardless if you are using `Anaconda` or `Miniconda`. The emphasis of this document is on using `Miniconda`.

On the Lab PCs you should be able to get by using virtual Python environment `dsp-comm39`. The Lab PC usage instructions are inside the vertical *Quote* bar immediately below. Instructions for installing Miniconda and setting up own `dsp-comm39` virtual environment on your own PC follow the Quote bar.

On the Lab PCs

Here you will be able to use an already created virtual Python environment using the Windows PowerShell (Admin) terminal window. To get started go to lower left windows icon, right–click, and launch `Windows PowerShell (Admin)` as shown in the screenshot below:



- Change directory to the current user:

```
PS C:\WINDOWS\system32> cd ~
PS C:\Users\mwickert>
```

- Enter and run the line `C:\ProgramData\Miniconda3\shell\condabin\conda-hook.ps1` to get the `conda` command line tool on your path
- Now list the available Python environments:

```
PS C:\Users\mwickert> conda env list
# conda environments:
#
base                  *  C:\ProgramData\Miniconda3
dsp-comm39               C:\ProgramData\Miniconda3\envs\dsp-comm39
```

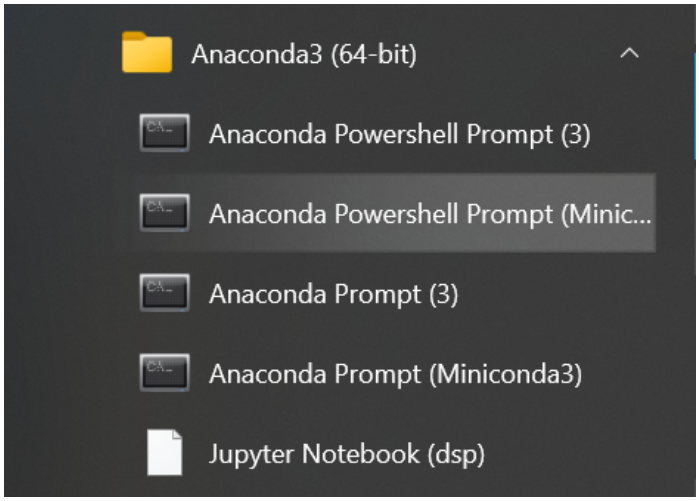
- Now activate the `dsp-comm39` environment so that you can launch `Jupyter lab` and then begin editing and running a `Jupyter notebook`:

```
PS C:\Users\mwickert> conda activate dsp-comm39
(dsp-comm39) PS C:\Users\mwickert> jupyter lab
```

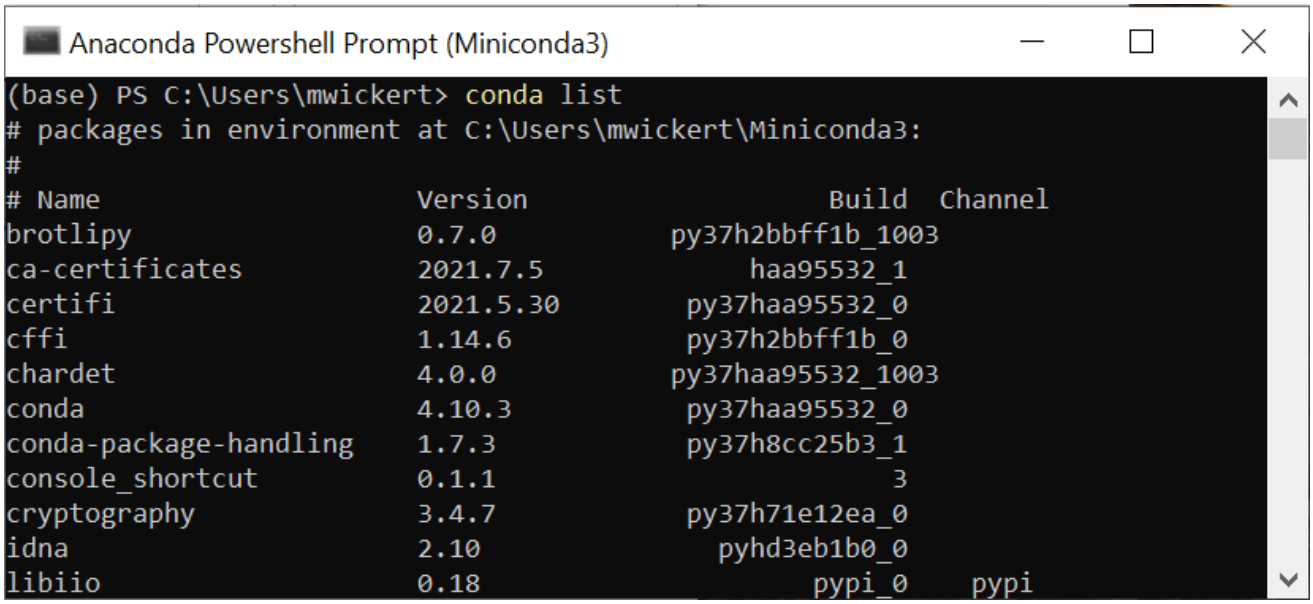
On Your Own PC Start by downloading and installing `miniconda`

This first major step has already been done for the lab PCs. If you are building up a Python environment on your own system start here.

- Download from https://docs.conda.io/en/latest/miniconda.html
- The User Guide is here https://conda.io/projects/conda/en/latest/user-guide/index.html
- The Conda *cheat sheet* is here: https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf (learn more about managing virtual environments, including deleting an environment if you want to start over)
- Launch using a teminal/shell window using the Windows `Start` menu for Anaconda and choose `Anaconda Powershell Prompt (Miniconda)`



- A `Powershell` prompt appears and you see `(base) PS C: ...` as the leading text. The `(base)` indicates that the base Python environment that comes when miniconda is activated
- Running `conda list` will show the packages installed in the base environment



Create a new virtual environment

The new virtual environment will be created using the base environment of the just installed `Miniconda` as the spring board.

> Note: Virtual environments are where you do your work with Python. Creating multiple virtual environments is a safe way of trying new Python packages without the risk of messing up a known good virtual environment. If you do mess up a new virtual environment you can always delete and start over. Leaving the minimalistic base environment of `Miniconda` intact gives you an environment you can always *activate* to safely mange one or more virtual environments.

The command line below will create a Python 3.9 virtual environment names `dsp-comm39`:

```
(base) PS C:\Users\mwickert> conda create --name dsp-comm39 python=3.9
```

- Change to/Activate the new environment:

```
(base) PS C:\Users\mwickert> conda activate dsp-comm39
```

- Now install the scipy stack

```
(dsp-comm39) PS C:\Users\mwickert> conda install numpy matplotlib scipy
```

- Now install jupyter related packages

```
(dsp-comm39) PS C:\Users\mwickert> conda install -c conda-forge jupyterlab
```

- Now install scikit-dsp-comm

```
(dsp-comm39) PS C:\Users\mwickert> conda install -c conda-forge scikit-dsp-comm
```

- More to Install? Let's see . . .

  - Suppose we needed the symbolic Python package `sympy`

Note this is already included with the full version of Anaconda. Using `conda` of `pip` should both work:

```
(dsp-comm39) PS C:\Users\mwickert> conda install -c conda-forge sympy
```

or

```
(dsp-comm39) PS C:\Users\mwickert> pip install sympy
```

Deleting a Virtual Environment

Suppose your virtual environment gets corrupt in some way. You can delete the environment and build a new one. Delete an environment and everything in it; Note your Python code and notebooks are not kept in the virtual environment:

```
# Begin by activating the (base) environment
(dsp-comm39) PS C:\Users\mwickert\somewhere_folder> conda activate base
# Now that you are in (base) you can delete the unwanted environment
(base) PS C:\Users\mwickert\somewhere_folder> conda env remove --name dsp-comm39
```

Installing `pyaudio_helper` Under Python 3.9

The formal instruction can be found at https://github.com/scikit-dsp-comm/pyaudio_helper . Unfortunately for this version of Python we have to manually locate a Python binary `*.whl` (wheel) file for the Win10 64bit OS. The site below maintains a huge collection of `.whl` files for various versions of Python.

https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyaudio

- The files on this site as of September 3, 2021 are shown in Figure 1 below:

PyAudio: bindings for the PortAudio library.
Includes ASIO, DS, WMME, WASAPI, WDMKS support.

PyAudio-0.2.11-cp310-cp310-win_amd64.whl
PyAudio-0.2.11-cp310-cp310-win32.whl
PyAudio-0.2.11-cp39-cp39-win_amd64.whl
PyAudio-0.2.11-cp39-cp39-win32.whl
PyAudio-0.2.11-cp38-cp38-win_amd64.whl
PyAudio-0.2.11-cp38-cp38-win32.whl
PyAudio-0.2.11-cp37-cp37m-win_amd64.whl
PyAudio-0.2.11-cp37-cp37m-win32.whl
PyAudio-0.2.11-cp36-cp36m-win_amd64.whl
PyAudio-0.2.11-cp36-cp36m-win32.whl
PyAudio-0.2.11-cp35-cp35m-win_amd64.whl
PyAudio-0.2.11-cp35-cp35m-win32.whl
PyAudio-0.2.11-cp34-cp34m-win_amd64.whl
PyAudio-0.2.11-cp34-cp34m-win32.whl
PyAudio-0.2.11-cp27-cp27m-win_amd64.whl
PyAudio-0.2.11-cp27-cp27m-win32.whl

Figure 1: Note support for up to Python 3.10 are now available and in particular we want `0.2.11-cp39-cp39-win_amd64.whl`.

- Download the `*.whl` file and place the file on the path where your terminal is currently pointed (see the below code in the *code fence*)
    - Note For the Lab PCs and most personal Win10 systems, the needed file is `PyAudio-0.2.11-cp39-cp39-win_amd64.whl`; I have placed this `whl` in the ZIP package described in the Appendix and linked here: http://ece.uccs.edu/~mwickert/ece5650/notes/Python_Projects/Project1_f2021.zip

```
# Assuming your path has the .whl file, run
(dsp-comm39) PS C:\Users\mwickert\Documents\pyaudio_setup> pip install PyAudio-0.2.11-cp39-cp39-win_amd64.whl
# Now do the official install of pyaudio
(dsp-comm39) PS C:\Users\mwickert\Documents\pyaudio_setup> pip install pyaudio
# Next install the ipywidgets
(dsp-comm39) PS C:\Users\mwickert\Documents\pyaudio_setup> pip install ipywidgets
# Next install of pyaudio_helper
(dsp-comm39) PS C:\Users\mwickert\Documents\pyaudio_setup> pip install pyaudio-helper
```

- When complete `pip list` or `conda list` should show packages similar to:

```
...
pip                       21.2.4           py38haa95532_0
prometheus_client         0.11.0             pyhd8ed1ab_0    conda-forge
prompt-toolkit            3.0.19             pyha770c72_0    conda-forge
pyaudio                   0.2.11                   pypi_0    pypi ###### <====
pyaudio-helper            1.0.4                    pypi_0    pypi ###### <====
pycparser                 2.20               pyh9f0ad1d_2    conda-forge
pygments                  2.10.0             pyhd8ed1ab_0    conda-forge
pyopenssl                 20.0.1             pyhd8ed1ab_0    conda-forge
...
```

A remaining issue is that the newest version of `pyaudio_helper.py` has a bug that needs to be fixed. A temporary solution is to place the file `pyaudio_helper_old.py` into the folder where you are running a notebook that uses `pyaudio_helper`, then swap importing the official package with the old file, e.g., from the first code cell of the sample notebook (see link below) I made the needed change as shown below:

```
%pylab inline
import sk_dsp_comm.sigsys as ss
# import sk_dsp_comm.pyaudio_helper as pah
import pyaudio_helper_old as pah # <=== import the old version with the same alias
import sk_dsp_comm.fir_design_helper as fir_d
import scipy.signal as signal
import ipywidgets as widgets
from IPython.display import Audio, display
from IPython.display import Image, SVG
```

This worked for me! A good `pyaudio_helper` example notebook can be found at http://ece.uccs.edu/~mwickert/ece5650/notes/Python_Projects/Project1_f2021.zip. The notebook file is `5650_Project_1_pyaudio_helper_Sample_2021.ipynb` and the above mentioned `pyaudio_helper_old.py` is included in the ZIP. I run the *Stereo Looping Example* near the top of the notebook and listen to a stereo music track I created in Apple's *GarageBand*. You will have to carefully pick the audio input and output devices found on your system.

Appendix

The ZIP package `Project1_f2021.zip` was used as a sample Jupyter notebook for the fifth problem of Project 1. The intimate problem details are not are not the concern, but the design of a `pyaudio_helper` real–time DSP application is relevant to other courses such as ECE 4655/5655 Real–Time DSP and ECE 4670 the Communication Lab. The file contents of `Project1_f2021.zip`:

5650_Project_1_pyaudio_helper_Sample_2021.ipynb	10/20/2021 2:19 PM	IPYNB File
5650_Project_1_Sample_2021.ipynb	10/20/2021 1:47 PM	IPYNB File
Music_Test.wav	11/19/2016 9:42 AM	WAV File
Problem5_title.png	11/7/2018 8:22 PM	PNG File
Problem5a.png	11/6/2018 10:52 PM	PNG File
Problem5b.png	11/7/2018 9:49 PM	PNG File
Problem5c.png	11/6/2018 10:55 PM	PNG File
proj1_prob2_fig.png	10/25/2017 4:36 PM	PNG File
Project1_fig2.PNG	11/18/2017 4:03 PM	PNG File
Project1_fig3.PNG	11/18/2017 4:04 PM	PNG File
pyaudio_dsp_IO.png	6/15/2018 5:48 AM	PNG File
pyaudio_helper_old.py	10/20/2021 1:48 PM	PY File
PyAudio-0.2.11-cp39-cp39-win_amd64.whl	9/2/2021 8:19 PM	WHL File
sounddevice_helper.py	10/20/2021 1:59 PM	PY File
speech_8k.wav	10/28/2018 7:41 PM	WAV File
speech_jam_8k.wav	10/28/2018 7:41 PM	WAV File