# First Time Setup

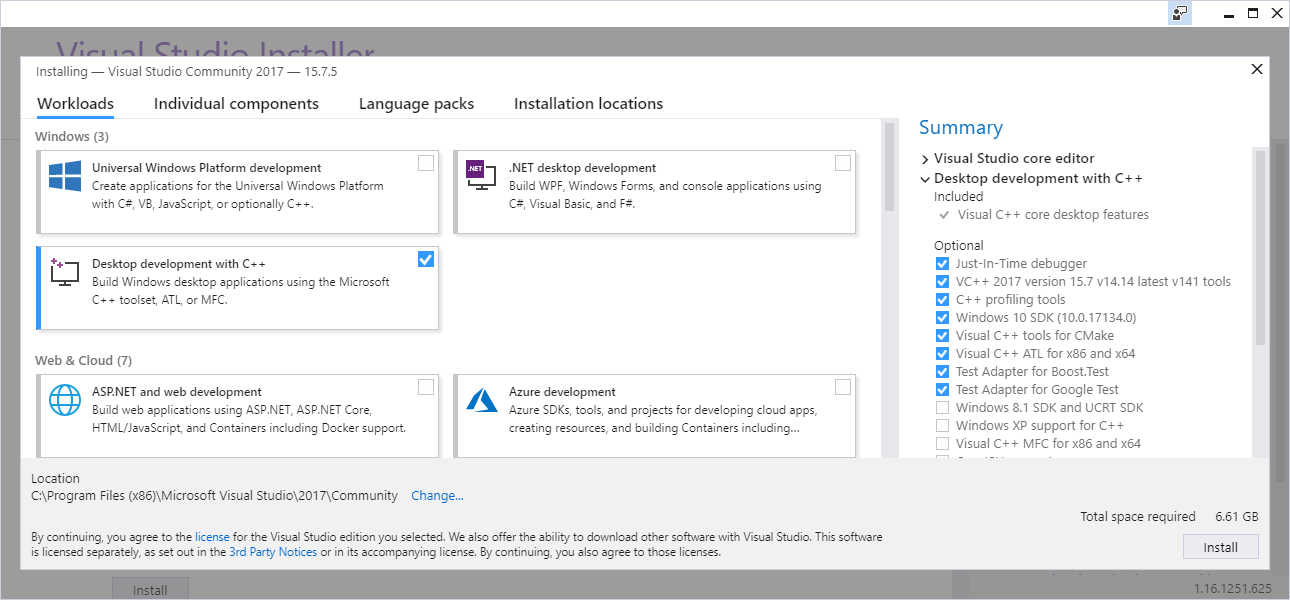
1. Download and install Python3.x

<https://www.python.org/downloads/>

1. Download and install Pycharm IDE Community Edition (choice is optional, but pycharm integrates well and downloads required modules   
   <https://www.jetbrains.com/pycharm/download/>
2. Download and install git for windows:

<https://gitforwindows.org/>

1. Download and install TortioseGIT  
   <https://tortoisegit.org/download/>
2. Clone the minivie project
3. Install Visual Studio Community with C++ Desktop Build tools  
   <https://visualstudio.microsoft.com/>
   1. Download for windows: “Community Edition”
   2. Select “Desktop Development with C++”

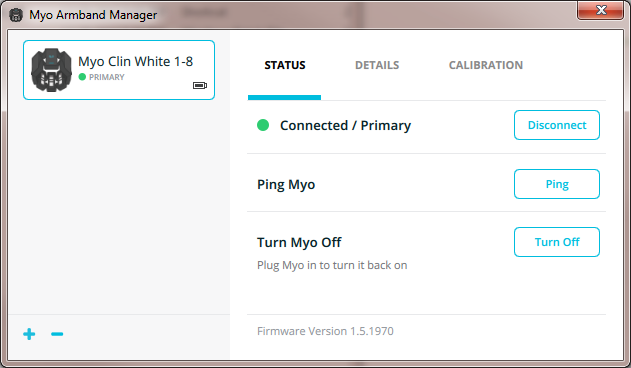


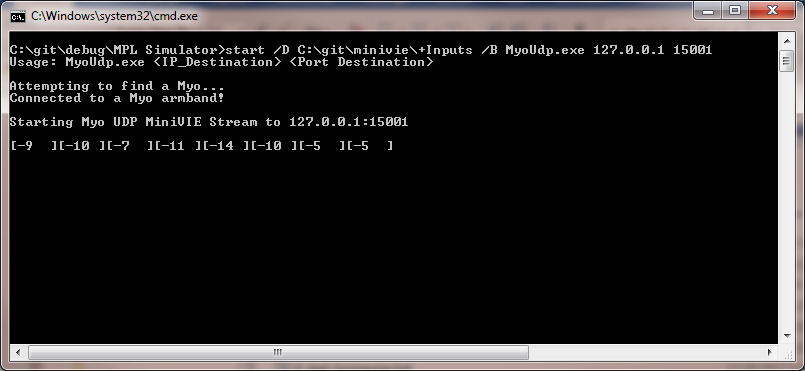
1. Open Pycharm, setup minivie project, and install requirements

# Running the Python VIE

1. MyoConnect

Use this tool from Thalmic labs to ensure your myo is paired / connected.  
Note: I usually disable all the pop-up and menus.  You can find those in settings/preferences

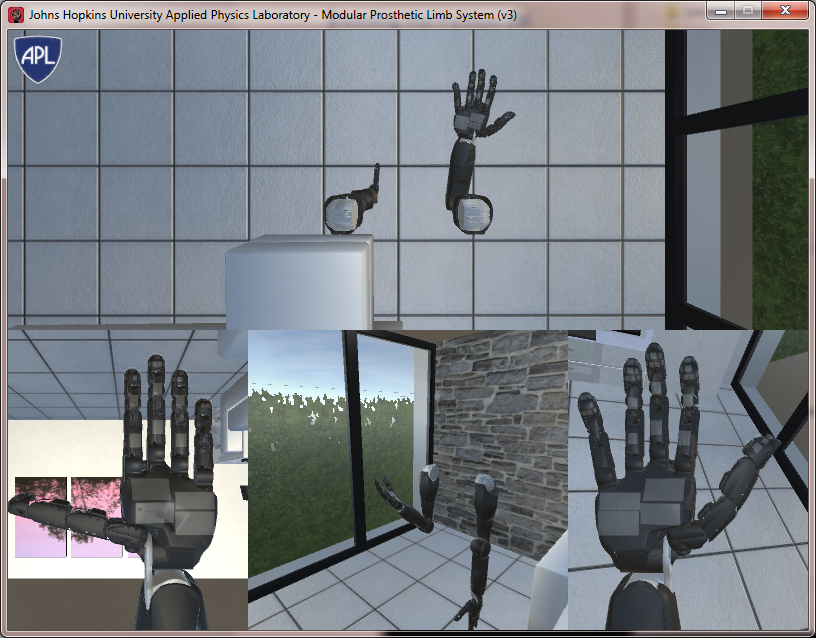
  
\minivie\+Inputs\MyoUdp.exe  
This is the helper program that connects the myo to python (via UDP).  This also takes IP and Port parameters.

MyoUdp.exe 127.0.0.1 15001  
Run it and you should see streaming numbers  
  


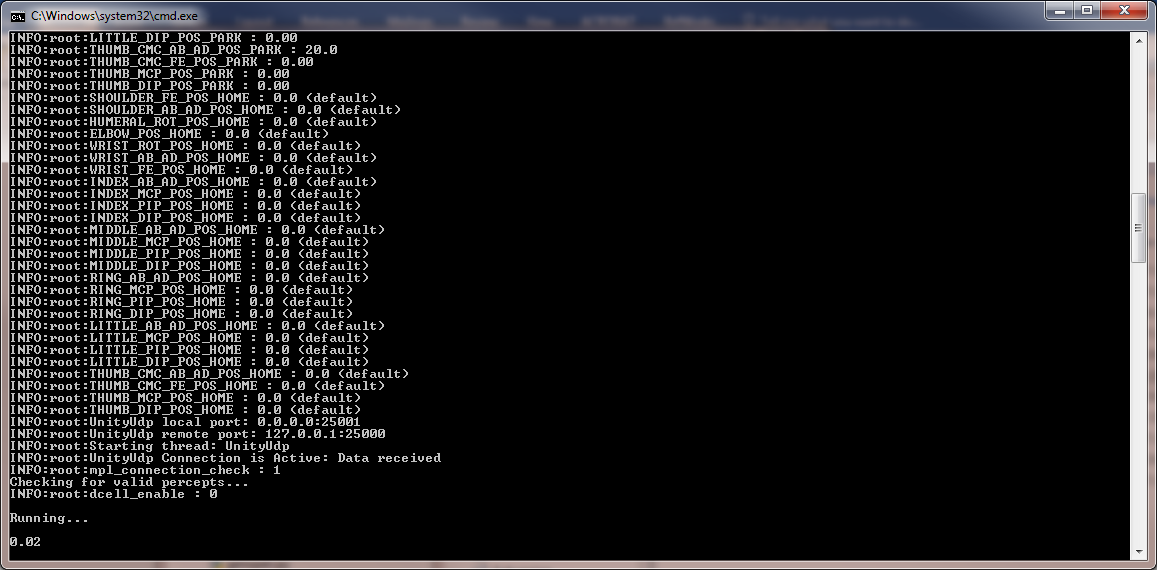
1. /minivie/python/minivie  
   This is the home directory and you should ‘cd’ there when using python
2. vmpl\_user\_config.xml

This is the parameter file for lots of systems settings.  I’ve attached one configured for your virtual setup.  Note though that you can ignore the myo settings in there now (those are for linux)

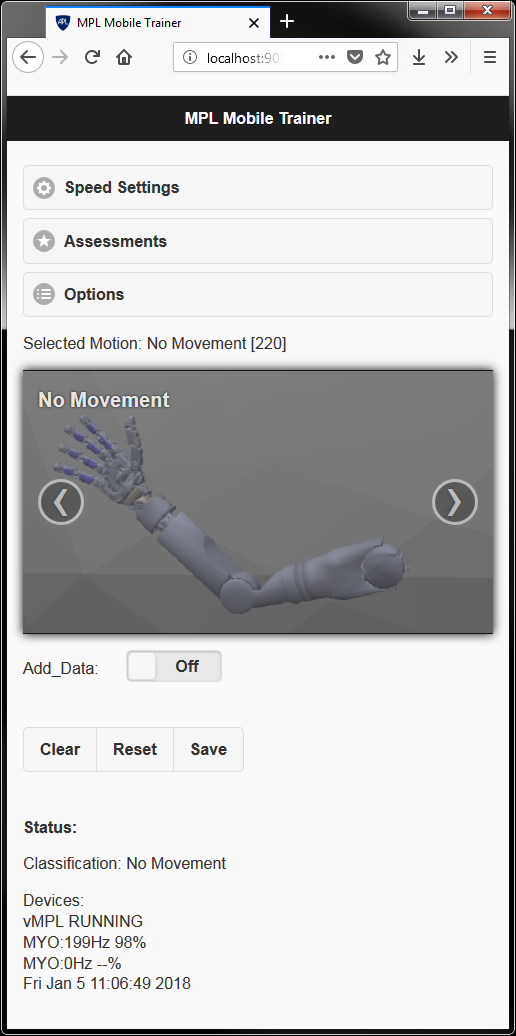
1. Virtual Arm   
   Contact JHUAPL about receiving this virtual environment =.

Run: Virtual\_Modular\_Prosthetic\_Limb\_v3.exe  


1. run\_www.py  
   This is the ‘main’ file that handles all the signal acquisition, pattern recognition, training, etc.  it runs in the background and will display some status info  
     
   Launch the program like this:  
   py -3 run\_www.py -x vmpl\_user\_config.xml



1. localhost:9090  
   When the main file is running, it hosts a local web server.  Open this interface to the arm.



Use the add data button to train a class, swipe images, etc. and these should control the arm in real-time.