# **Pre-lab 4 Assignment**

Components Required:

1. Personal Device
2. Ti Radar
3. USB Cable for radar

\*note: you will not use Raspberry Pi for interacting with radars

Goals:

The goal of this pre-lab is to set up the programming environment and software required to communicate with the radar we’ll be using in labs 4 and 5. If you are having any issues please ask your TAs.

## Part 1: Installing Necessary Programs

We recommend you use your personal laptops for interfacing with the radar.

Installing Anaconda:

We will be using Anaconda to manage the python environment and packages required for running the radar software.

1. Download the Anaconda from this link: [Anaconda Download](https://www.anaconda.com/download).
2. Run the install file and follow the directions for a default installation.

Installing Radar Toolbox:

Download all the folders from Canvas under pre-lab 4. This includes the files for the radar visualizer and the accompanying configurations we will be using:

* “boot”
* “Industrial\_Visualizer” folder
* Lab4.cfg

Installing the Required Python Packages:

**For Windows Machines:**

1. Open an Anaconda Prompt terminal (you may need to restart your machine for the Anaconda install to complete). You may do so by typing “anaconda prompt” in your menu’s search bar. To install all the required packages, run following command under Industrial Visualizer directory:
   1. cd /[**YOUR FOLDER PATH**]/Industrial\_Visualizer
   2. Start-Process SetUpEnvironment.bat
2. Follow the instructions and install all the packages. Run the following command to activate the virtual python environment just created:
   1. conda activate Cs437
3. To start the visualizer, run:
   1. python gui\_main.py

For MacOS or Linux Machines:

1. Open terminal and run the following commands:
   1. chmod +x setup.sh
   2. ./setup.sh
2. Follow the instructions and install all the packages. Run following command to activate the virtual python environment just created:
   1. conda activate Cs437
3. To start the visualizer, run:
   1. python gui\_main.py

If, when ‘running python gui\_main.py’, you receive a permission error, run the following command in terminal:

1. sudo /home/**[your directory]**/anaconda3/envs/CS437/bin/python arprog\_cmdline.py -p /dev/ttyACM0 -f motion\_and\_presence\_detection\_demo.release.appimage -s SFLASH -t META\_IMAGE1

## Part 2: Flashing the Radar

Configuring the Radar Pins:

1. **Make sure your radar is disconnected from power before following the next steps**
2. Configure the radar Switch S1 pins according to the following pin configuration:
   1. Pin 1: off
   2. Pin 2: off
   3. Pin 3: off
   4. Pin 4: off
   5. Pin 5: off
   6. Pin 6: off
3. Configure the radar Switch S4 pins according to the following pin configuration:
   1. Pin 1: off
   2. Pin 2: off
   3. Pin 3: off
   4. Pin 4: off
4. Connect the radar to your computer using the provided USB cable. Make sure the radar is powered up after connection (red LED light should turn on) and ready for communication.
5. To identify the COM port to which the radar device is connected, follow these instructions. Make note of the COM port number for future reference.
   1. For Windows devices:
      1. Open the “device manager” on your computer. You can do this by pressing ‘Win + X’ and selecting “Device Manager” from the menu
      2. Look for “Ports (COM & LPT)” in the list of device categories and expand it.
      3. Identify the COM port number associated with your radar device. It is typically labeled as something along the lines of “TI IWRL6432” or “USB Serial Device”.  
         
   2. For MacOS or Linux Machines:
      1. Open terminal and run following command:
         1. ls /dev/cu.\*
      2. Look for a device that represents your radar device.
         1. For Mac devices, it may appear as ‘/dev/cu.usbmodemRI321’
         2. For Linux devices, it may appear as ‘/dev/ttyACM0’
6. Open your terminal (MacOS) or command prompt (Windows).
7. Navigate to the “boot” folder inside the Radar Toolbox directory that you downloaded, and run the following command to flash the radar.
   1. python arprog\_cmdline.py -p **[PORT NUMBER]** -f motion\_and\_presence\_detection\_demo.release.appimage -s SFLASH -t META\_IMAGE1
8. Replace the ‘[PORT NUMBER]’ to the specific port number you noted down in step 7.
   1. If, when running the command, you receive a permission error, run the following command in terminal:  
      sudo /home/**[your directory]**/anaconda3/envs/CS437/bin/python arprog\_cmdline.py -p /dev/ttyACM0 -f motion\_and\_presence\_detection\_demo.release.appimage -s SFLASH -t META\_IMAGE1
9. To make using the radar easier, we will now be setting up a rudimentary stand so that the radar can stay upright without support:
   1. In your radar box, you should find a small bag with 4 metal standoffs, washers, and screws. We will only need the washers and standoffs.
   2. Assemble the standoffs and washers in the configuration shown in the image to the right.  
      NOTE: To avoid interference with any of the pins or power cables, screw your standoff into the bottom two screw holes on the radar (along the short side closest to the barrel jack)

## Part 3: Running a demo using the Visualizer

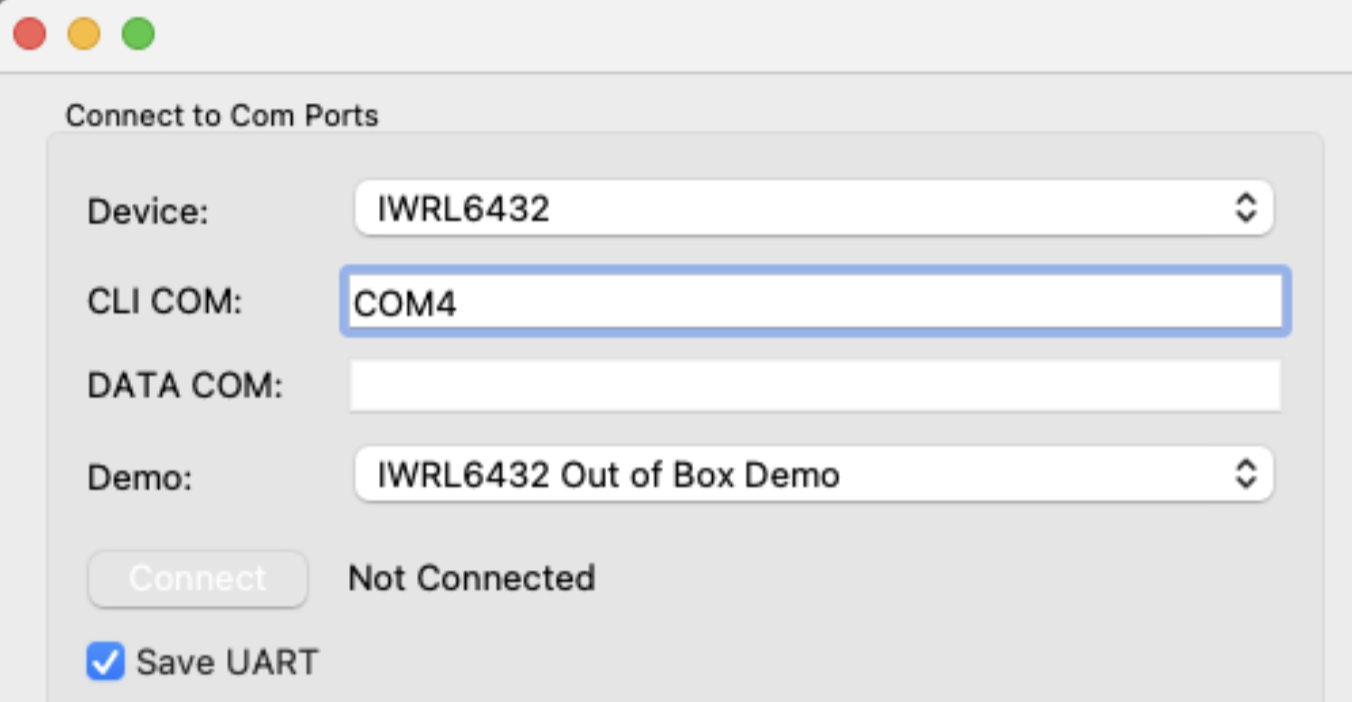
Introduction:

This document will take you through the general steps to run the motion and presence detection demo with Visualizer and TI IWRL6432Boost board, visualizing the objects in front of the radar as a point cloud.

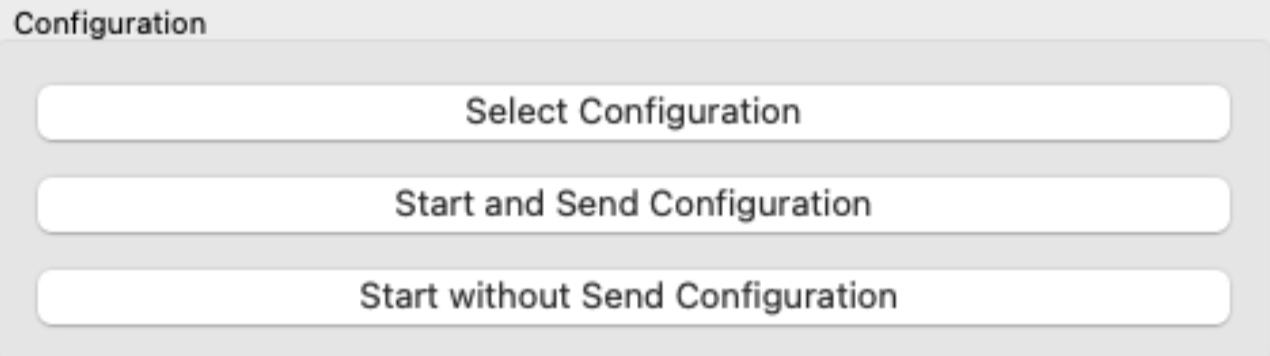
1. **Disconnect the radar from power before following the next steps**
2. Configure the radar Switch S1 pins according to the following pin configuration:
   1. Pin 1: on
   2. Pin 2: off
   3. Pin 3: off
   4. Pin 4: off
   5. Pin 5: off
   6. Pin 6: off
3. Now, start the visualizer with following by running following code:

Python gui\_main.py

1. In the application that opens up, enter the port number you found earlier into “CLI COM” and check “Save UART”:

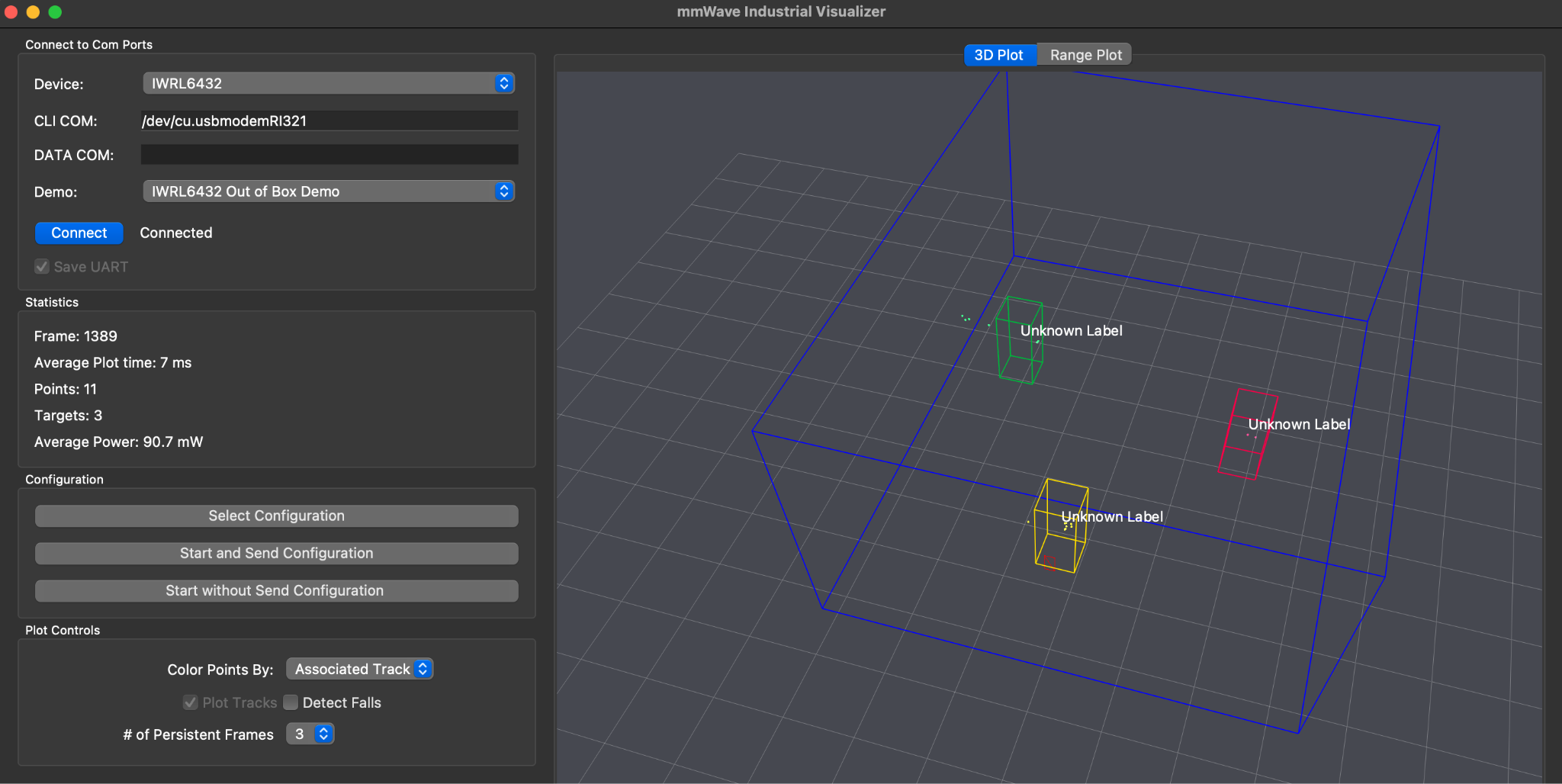


1. Click ‘Select Configuration’ and navigate to: profiles > xwrL64xx-evm from the Profiles folder downloaded in Part 1, and select “TrackingClassification\_MidBw.cfg”.



1. Click “Start and Send Configuration”

You should now see a series of white dots on the main tab of your visualizer similar to below – this is the point cloud data gathered by the radar.



At the top of the visualizer, you can switch tabs to see the radar’s range profile. If you move around in front of the radar’s line of sight, you will see that the point cloud follows your movement. In some cases, the visualizer may even recognize you and other objects in the radar’s line of sight.

1. Play around with movement and see how it affects the range profile of the radar. Based on your results, **what does the range plot of the visualizer represent?**

# Pre-lab Deliverable:

Now that you have a working visualizer, take a short video of your visualizer’s output after starting and sending the configuration to the radar. In your video, show the visualizer’s range profile while you’re moving an object (i.e. your hand) away from the radar in its line of sight. Your video should see that the range detected by the radar increases as your object moves away from the radar. Be sure to show your hand moving in the video alongside the visualizer output.