This application has been developed based on Python 3.10.12 and all the library dependencies can be found in the “requirements.txt”.

How to use the application

This example uses IWR6843ISK TI radar. Check the radar to confirm that it is the “Functional Mode” before connecting to the USB port. The application displays the following desktop interface. You can click the “Refresh Serial Ports” after connecting the USB to the computer.

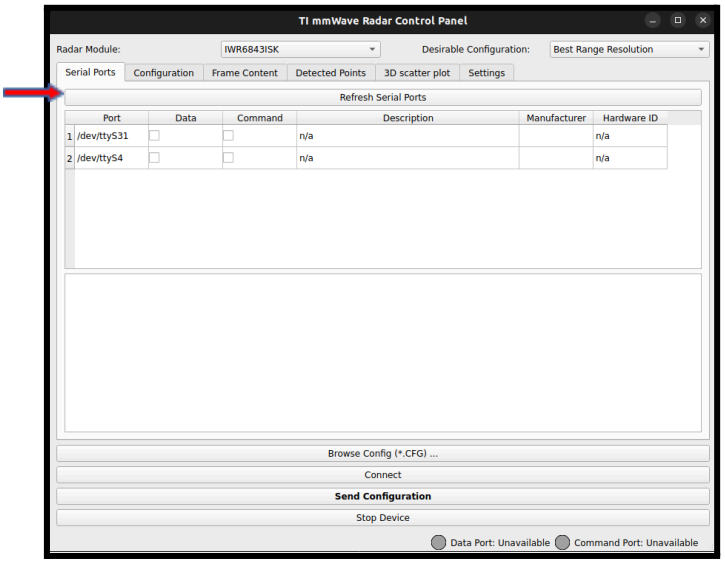

Figure (1): Initial user interface of the application

The two updated USB port names will be shown as follows if the application detects the respective ports. The application will automatically check the Command and Data Ports in Linux, whereas the user needs to select the associated ports manually in Window.

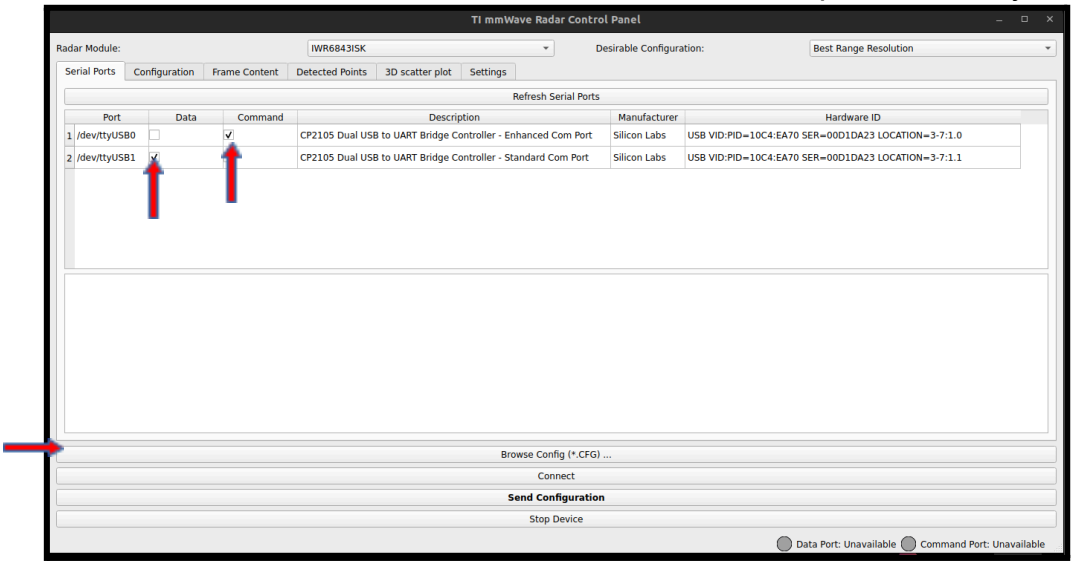

Figure (2): Result after refreshing

The configuration file has been required for this sensor. You can either generate from the “TI’s mmWave Demo Visualizer” or use the configuration file inside the folder “Sensor_cfg_file_example”.

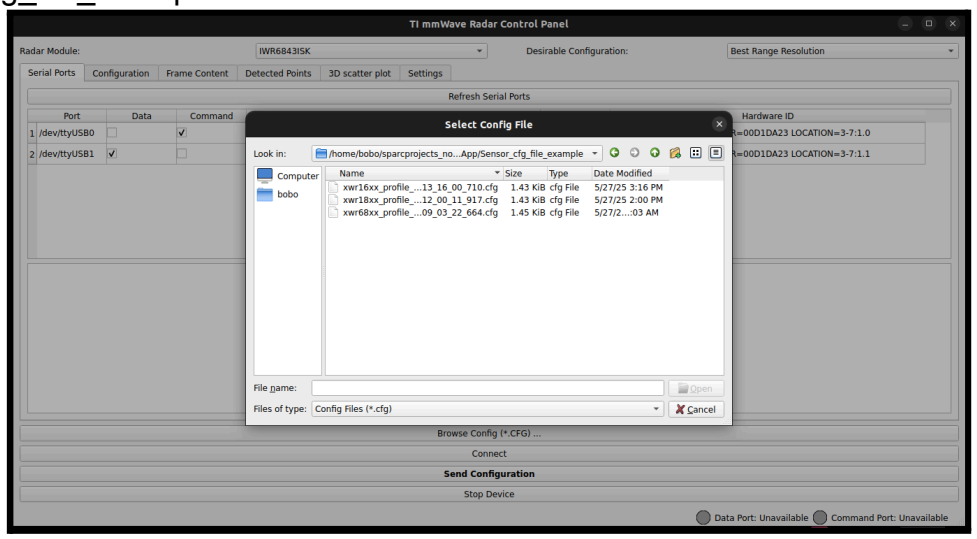

Figure (3): List of configuration files

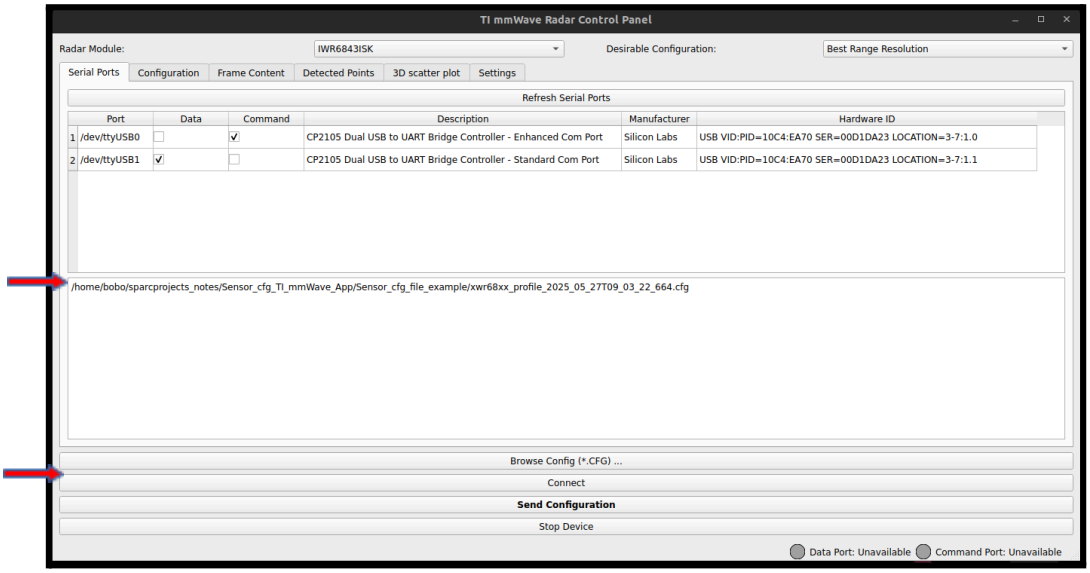

Figure (4): Output after successfully importing the configuration file

Firstly, click “Connect” for communicating between the computer and the sensor. If it is successfully connected, the application will display the “Green” color and it says “Data Port: Connected” and “Command Port: Connected”. Then, click “Send-Configuration” to the sensor.

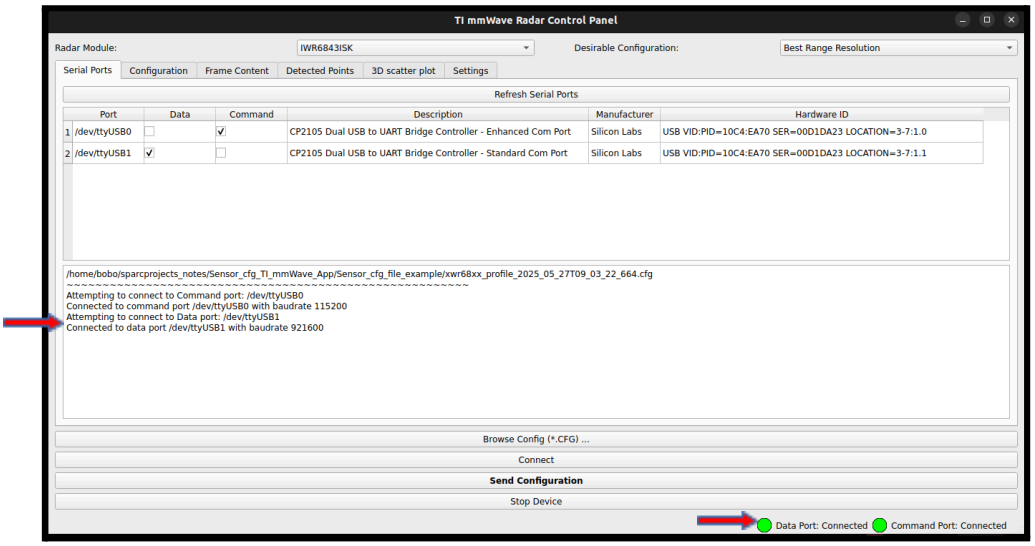

Figure (5): Log messages and the ports have been connected

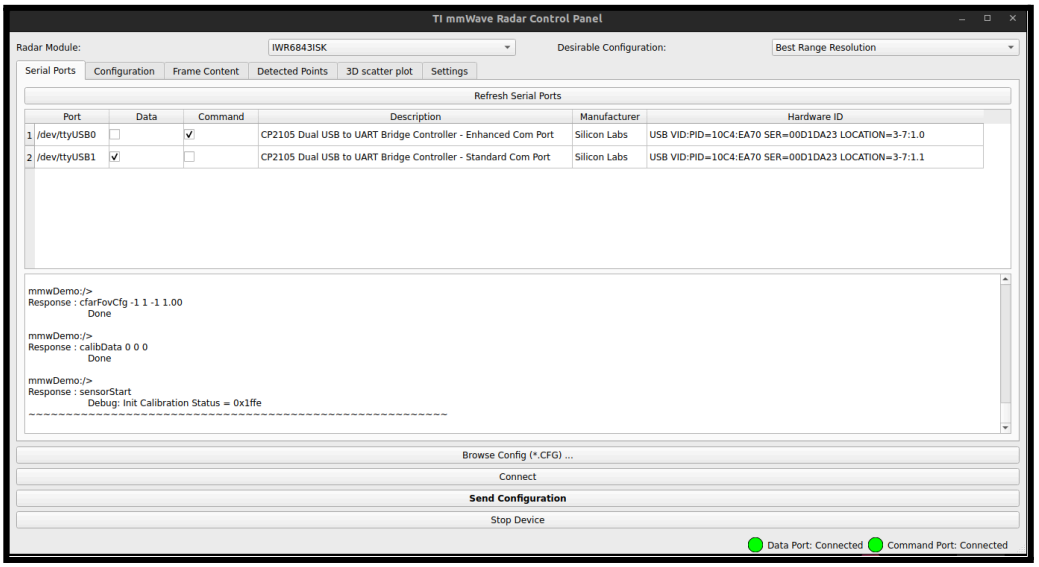

Figure (6): Output after clicking “Send Configuration”

Figure (7): Range Profile

After sending the configuration file, the above output will be shown as the log messages inside the red section and the “Range Profile” has been plotted as default.

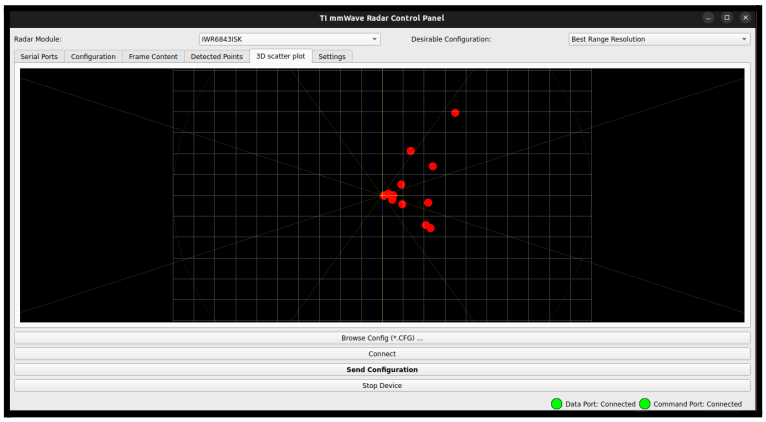

Figure (8): Point clouds in 3D scatter plot

In the “3D scatter plot” section, the point clouds have been displayed, which are coming from the radar.

With this application, “Noise Profile”, “Range Azimuth Heat Map”, and “Range Doppler Heat Map” can also be plotted in the “Configuration” section. Go to “Display” and click on the boxes. Afterwards, click the “Send Configuration” button.

Whenever you change the parameters, you must click the “Send Configuration” button.

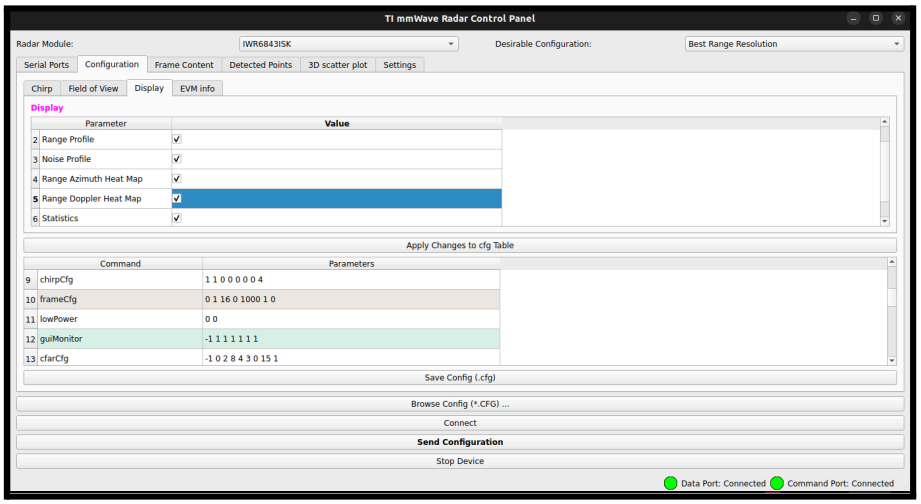

Figure (9): Display section from Configuration

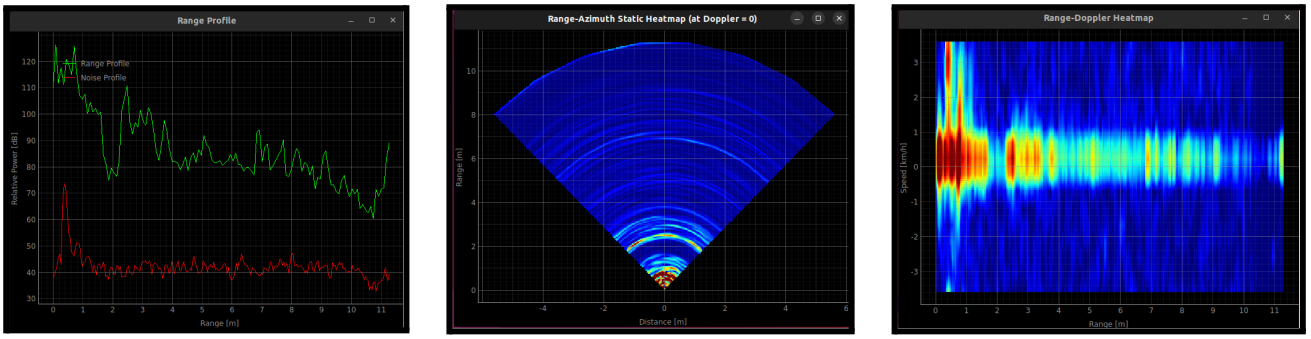

Figure (10): Plots of Range Profile, Range-Azimuth Static Heap map and Range-Doppler Heat map

Instead of importing configuration files, the app has its three default configuration parameters, which are “Best Range Resolution”, “Best Velocity Resolution”, and “Best Range” as follows. You can find the parameters values in “Chirp” at the “Configuration” tab.

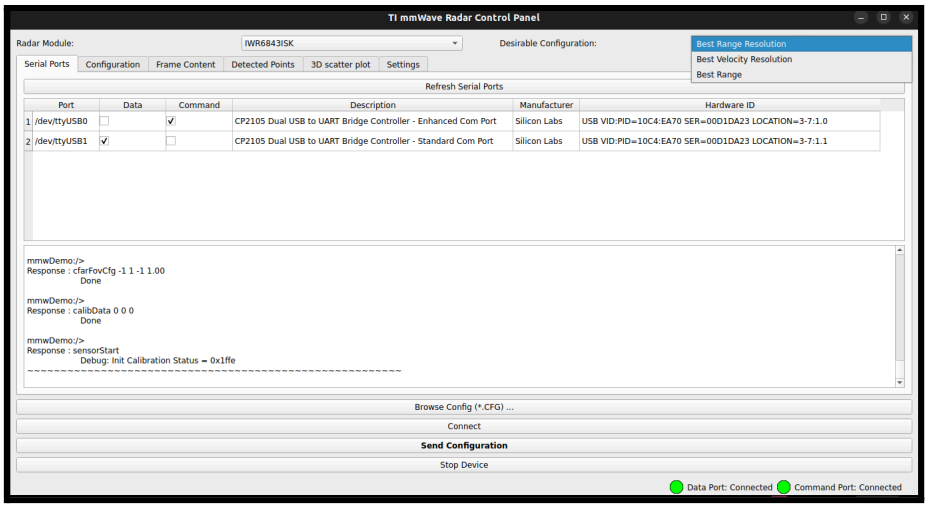

Figure (11): Three default settings

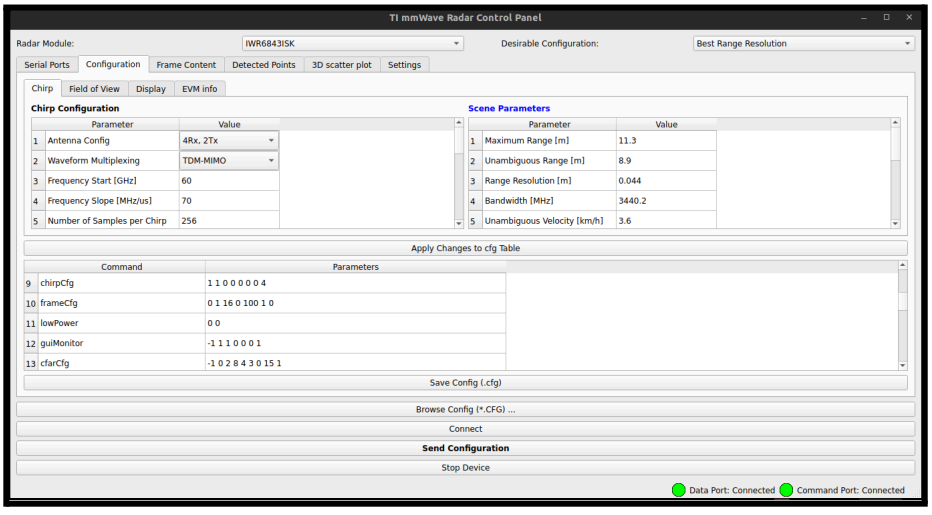

Figure (12): Configuration of “Best Range Resolution”

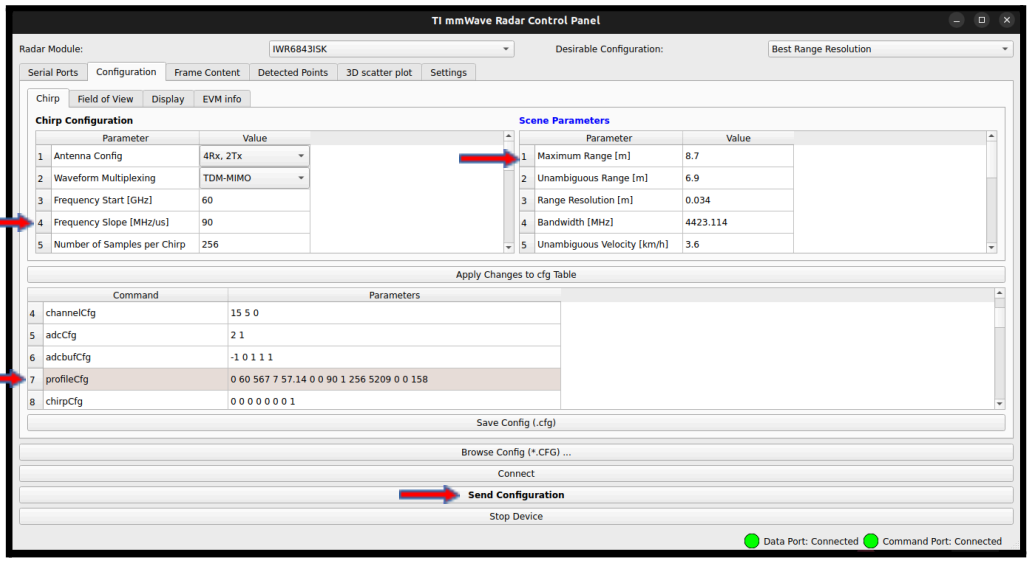

Figure (13): Customize the parameters

You can also customize the configuration's values. For example, changing the frequency slope will dynamically update the parameters of both "Maximum Range [m]" and "profileCfg" sections. Afterward, you must click on "Send Configuration" to update these parameters inside the sensor. Be careful when changing the parameters, as some value combinations may be incompatible with the sensors and could cause the program to crash and stop working.

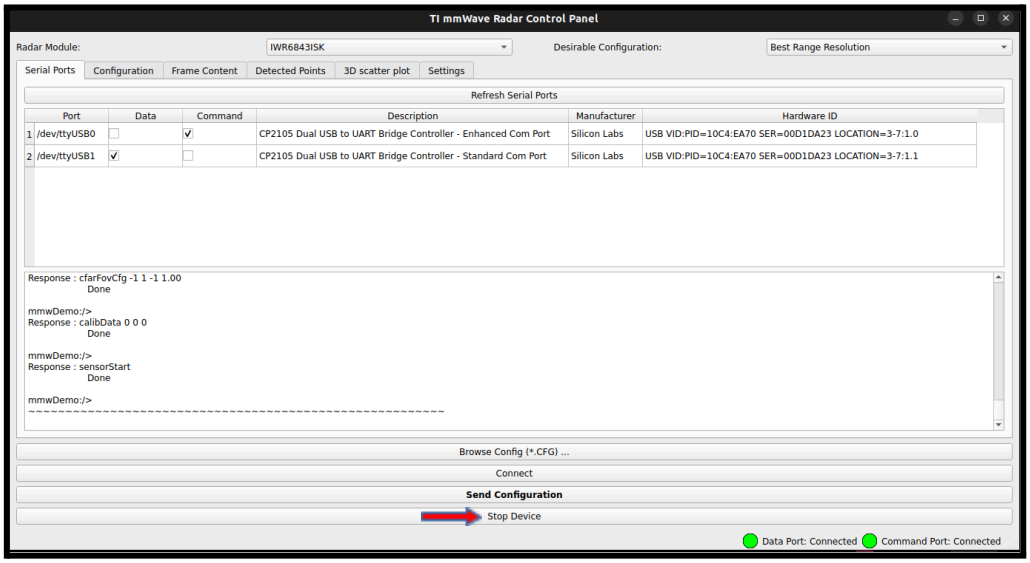

Figure (14): "Stop Device" button

Whenever you encounter the errors, you may need to click on the "Stop Device" button. This button is equivalent to the "NRST" button on the sensor's hardware. Its functionality is to restart the sensor from the beginning.