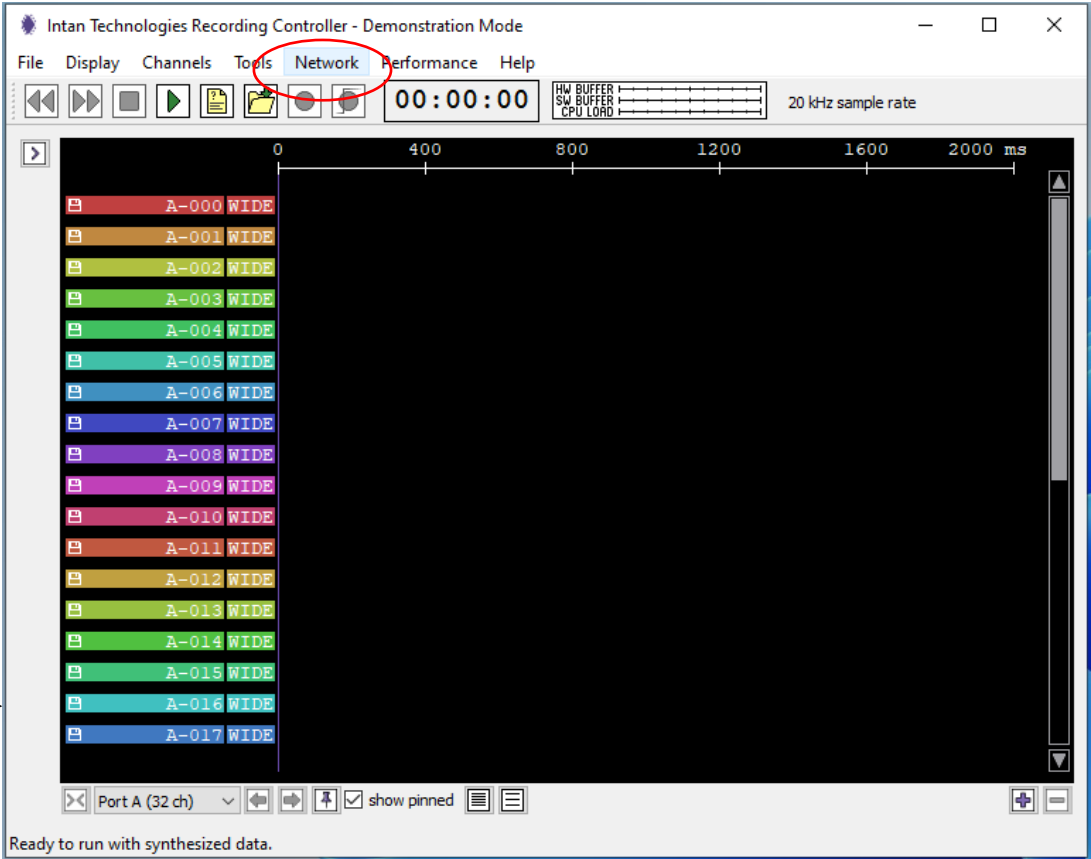


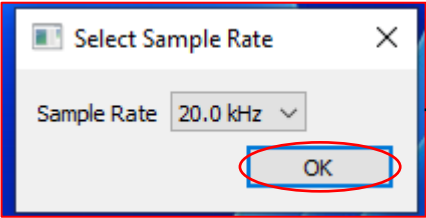
User Manual to run Plotter.py



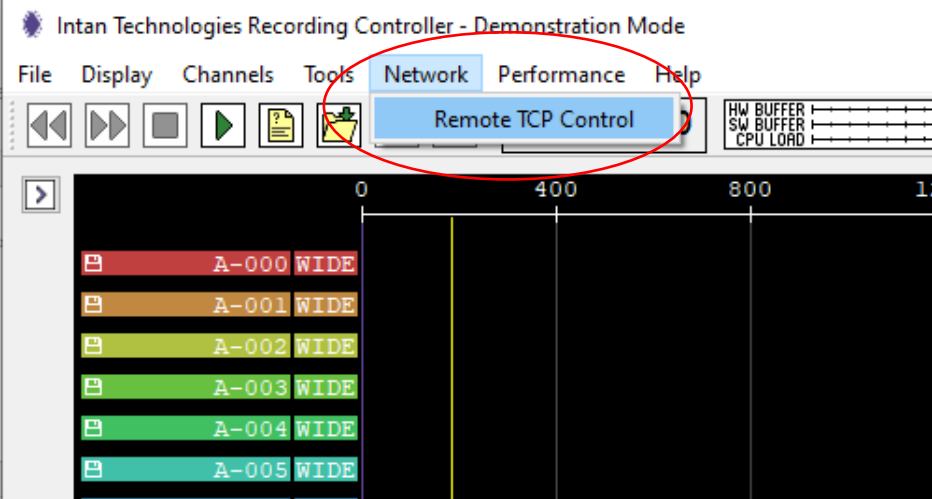
STEP 1



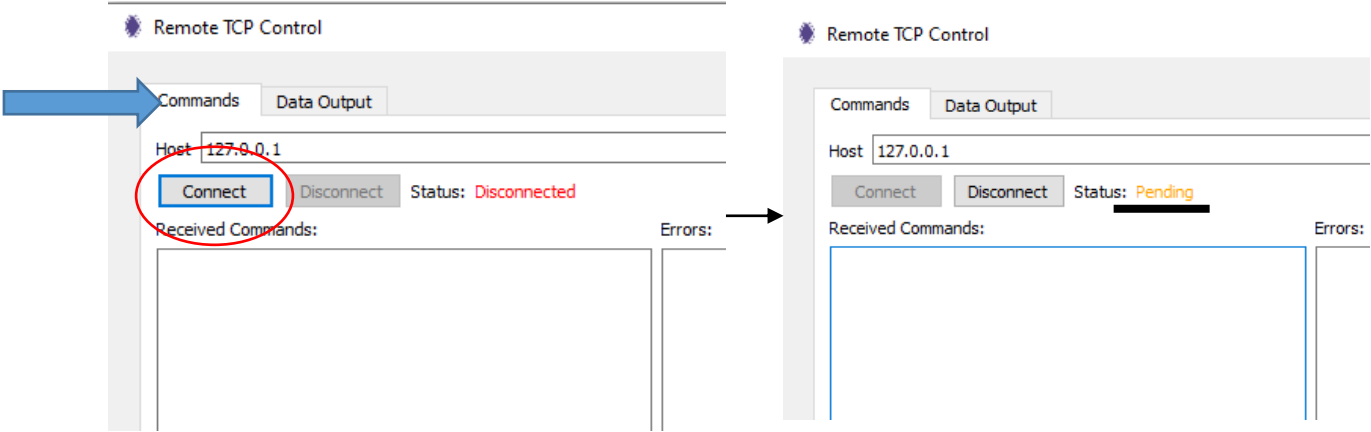
STEP 2



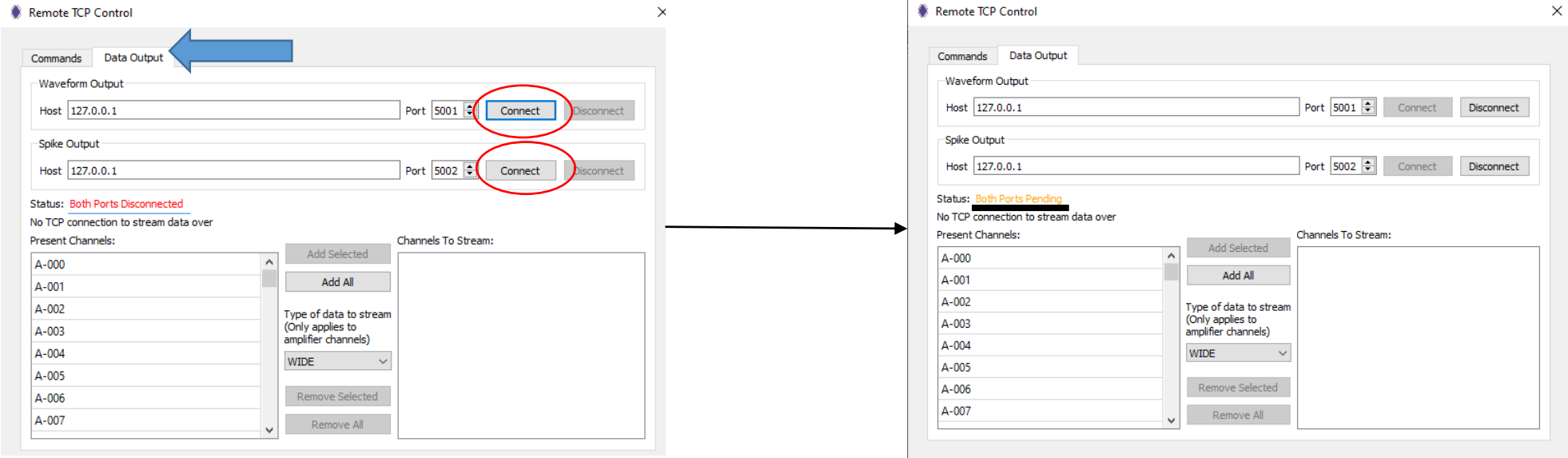
STEP 3



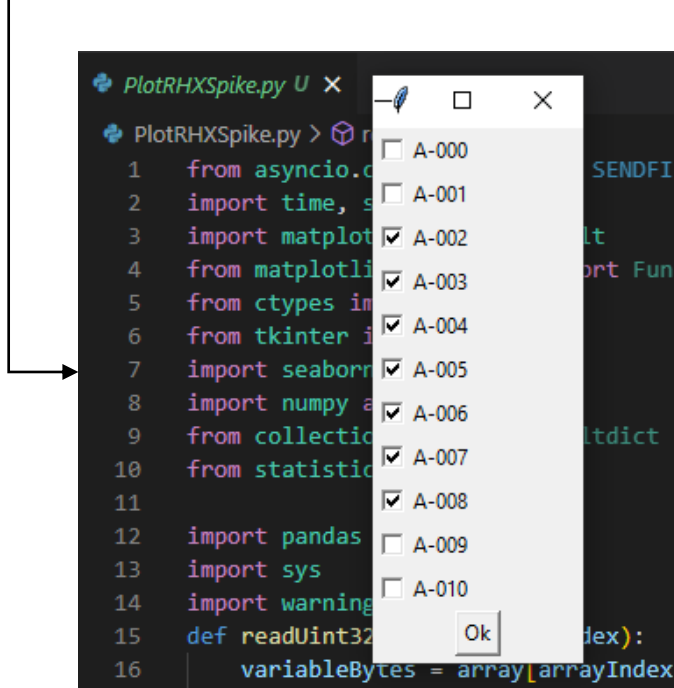
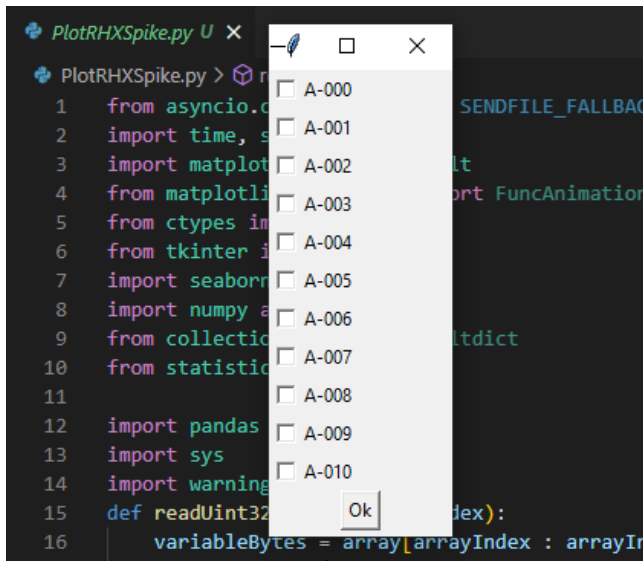
STEP 4



STEP 5



STEP 6



STEP 7

Once 'Ok' clicked , you will see

- two empty matplotlib plots (figure 1 & figure 2) open
- `Plotter.py` waiting for response from `Texture_task_wTCPIP.m`

```
H:\realtime-analysis>C:/Users/sammasi/AppData/Local/miniconda3/envs/RTA/python.exe h:/realtime-analysis/PlotRHXSpike.py
selected channels ['A-001', 'A-002', 'A-003', 'A-004', 'A-005', 'A-007', 'A-007', 'A-008', 'A-007']
opening plot.....
Connecting to TCP command server...
Connecting to TCP SPIKE output server...
```

STEP 8

Once the connection is established ,
plotter.py will be ready to receive
stim_cond & SPK_outut ON OFF signal

```
H:\realtime-analysis>C:/Users/sammasi/AppData/Local/miniconda3/envs/RTA/python.exe h:/realtime-analysis
selected channels ['A-002', 'A-003', 'A-004', 'A-005', 'A-006']
opening plot.....
Connecting to TCP command server...
Connecting to TCP SPIKE output server...
connected from ('172.27.85.105', 39080) to receive stim conditions
connected to receive t time to collect SPK
stim condition recieved :
```

When the texture_taskwTCPIP.m is
terminated, the plotter.py terminal
would look something like this

```
stop SPK output
stim condition recieved : 2
start SPK output
.
.
.
.
.
.
.
.
stop SPK output
stim condition recieved :
Enter 'q' to quit: 
```

And to quit plotter.py, enter 'q'

```
.
.
stop SPK output
stim condition recieved :
Enter 'q' to quit: q
H:\realtime-analysis>
```

The spike data from the trials will be
stored in “CH_stim_SPK_data.csv”

To run the code again ,follow steps 3-7

Note :

- If the port number/s is/are changed in step 5 , update the changes in the code simultaneously
- Run step 1 through 5 before executing the code.