Lab2-1: Analyze Sensor Data with Python

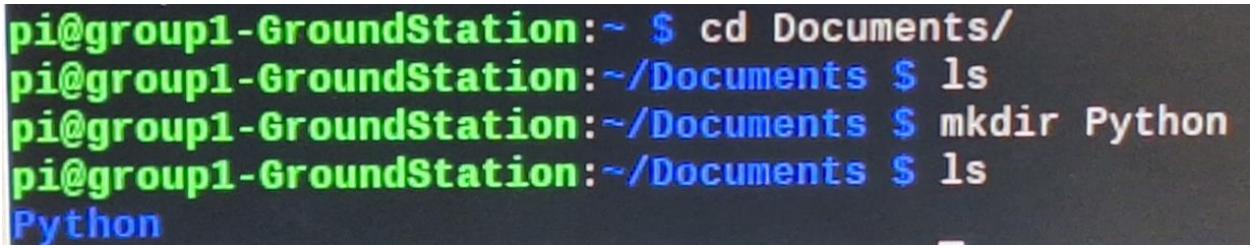
Lab Objective

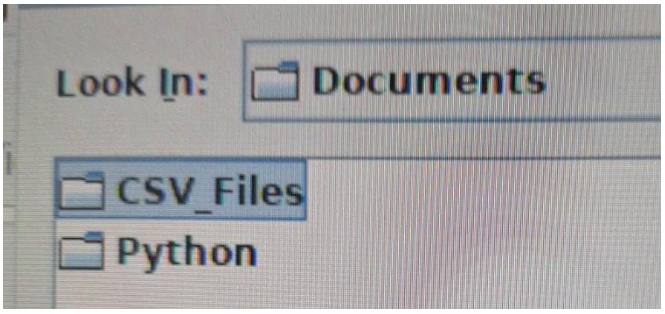
In this lab, we will explore how to visualize sensor data collected from a satellite. The data is stored in multiple CSV files, each containing time-series measurements from various sensors onboard the satellite. The goal is to plot these measurements to analyze trends, detect anomalies, and gain insights into the satellite's performance and the environmental conditions it monitors.

Lab instruction

Step1: Directory Setup

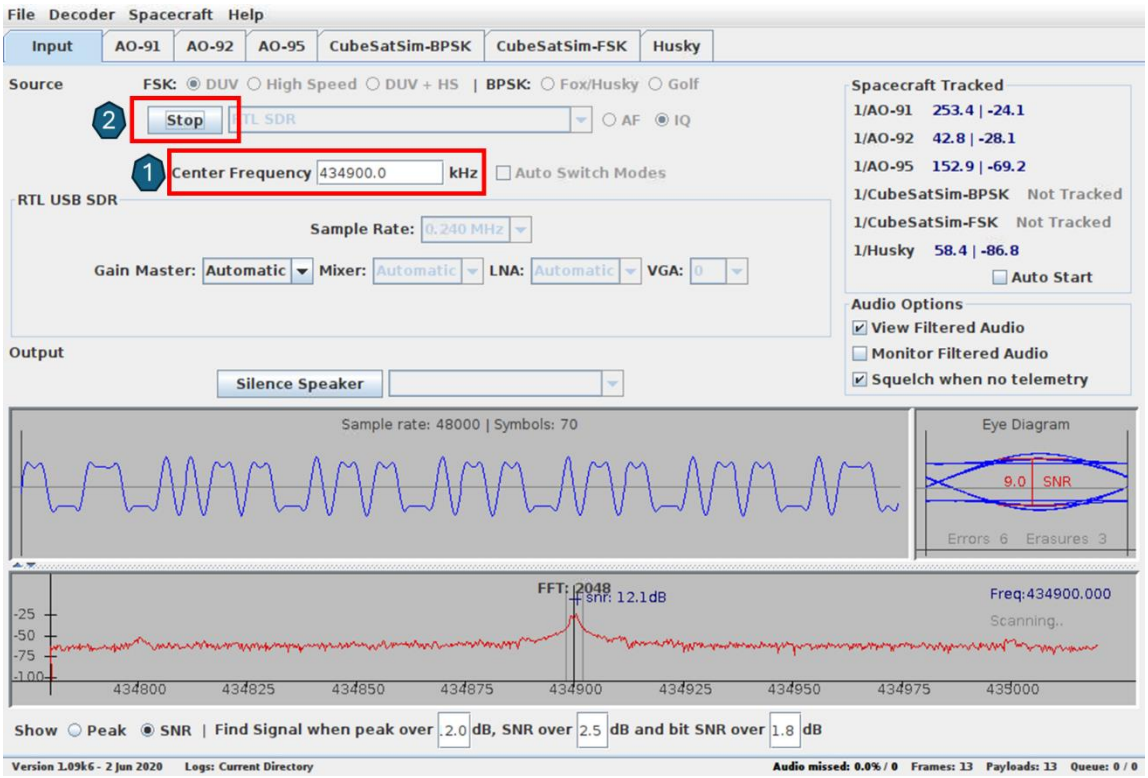
From the Pi terminal, create **'Python' and 'CSV_Files'** folders in the directory of 'Documents/'. Type 'mkdir Python' to create python file folder.





Step 2: Get the Signal

From the ground station software, set your listening frequency and press the start button above to receive the signal from the satellite.

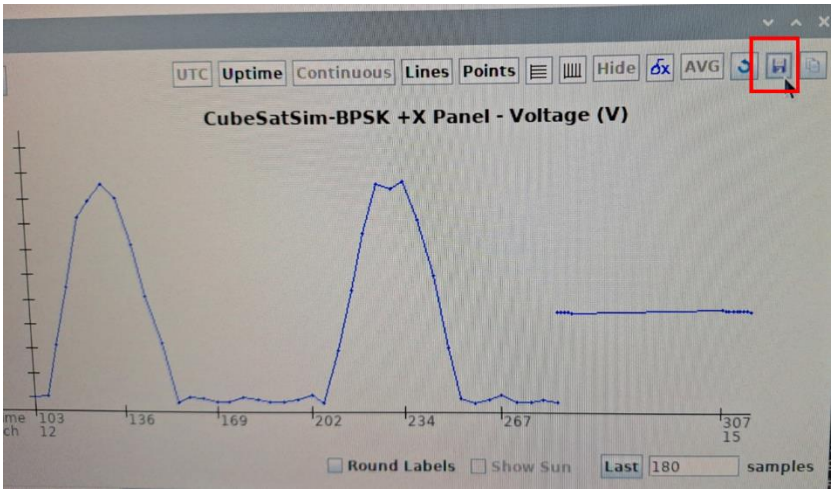


Step 3: Get CSV Files for sensors you want to check

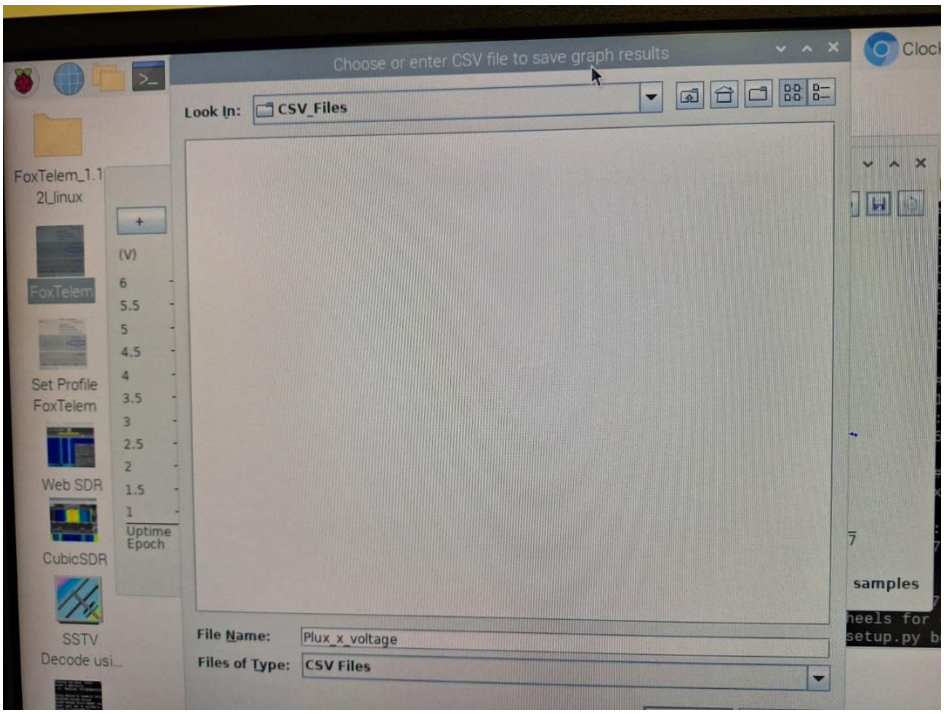
You can easily think of a .csv file as a kind of excel file that saves data to easily deal with code.

Choose one of ‘CubeSatSim-BPSK’ or ‘CubeSatSim-FSK’. The modulation should be the one you transmitted from your satellite.

Click the save button to save the data as a .CSV file.



File name is better to include the sensor name to differentiate various sensors.



Step 4: Install Python3 Libraries

Python3 is already installed what we need more is libraries to be installed.

Jump to 6. if:

```
python3 --version
pip3 --version
```

Above command returns the version of python3 and pip3.

1. In your terminal, type 'which **python3**' to get your python3 directory.
2. Type 'nano ~/.bashrc'
3. Add the following lines at the end of the file:

```
export PATH="/usr/local/bin:$PATH"  # Adjus
export PATH="/usr/local/bin/python3:$PATH"
export PATH="/usr/local/bin/pip3:$PATH"
```

4. Save the file by 'ctrl+x --> y --> enter' and reload it with command:

```
source ~/.bashrc
```

5. Verify the path by typing below command in terminal:

```
python3 --version
pip3 --version
```

6. Once the path is verified (could check path) install libraries. (This might take longer than you think.)

```
pip install pandas matplotlib
```

7. Download a python file from the link below or create python file and run.

[Python Github]

How to run:

```
pi@group2-GroundStation:~/Documents/Python $ python3 draw_graph.py
Traceback (most recent call last):
```

8. You might see the error:

```
AttributeError: 'NoneType' object has no attribute 'lower'
TypeError: Couldn't find foreign struct converter for 'cairo.Context'
TypeError: Couldn't find foreign struct converter for 'cairo.Context'
TypeError: Couldn't find foreign struct converter for 'cairo.Context'
TypeError: Couldn't find foreign struct converter for 'cairo.Context'
```

To solve, type this in the terminal:

```
pi@group2-GroundStation:~/Documents/Python $ sudo apt-get install python3-gi-cairo
```

Lab2-2: Calculate Solar Power Generated by Python

Lab Objective

The objective of this lab is to learn how to read, process, and analyze solar power generation data using Python. Students will read CSV files containing voltage and current measurements, compute the instantaneous and cumulative power generated, and visualize the results using appropriate plots. The lab will enhance skills in data manipulation with `pandas` and data visualization with `matplotlib`. Additionally, it will provide an understanding of solar power generation by analyzing data from different spatial orientations.

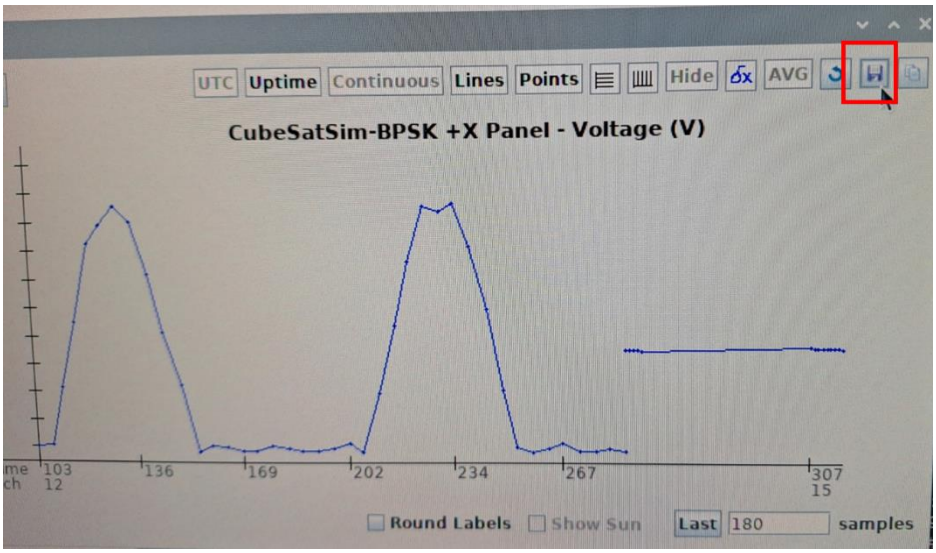
Lab instruction

Step1: CSV Files Download and Directory Setup

Power of the solar panel is calculated with below equation:

Power (Watts) = I (Current) * V (Volt)

Choose one of 'CubeSatSim-BPSK' or 'CubeSatSim-FSK' from the ground station depending on your modulation choice.



Download CSV files for the 'Voltage' and 'Current'.

The directory setup is the same as the previous lab. Save the CSV files from the data UI as mentioned in Lab 2-1.

```
pi@group2-GroundStation:~/Documents/CSV_Files $ ls
minus_x_solar_curr.csv  minus_x_solar_volt.csv  plus_x_solar_curr.csv  plus_x_solar_volt.csv
```

```
pi@group2-GroundStation:~/Documents/Python $ ls
draw_graph.py  solar_power.py
```

You will probably set up all the required things from the previous lab. Create or get the code from the Solar Power Code.

Step2: Running python file

You may write a Python code that generates two graphs:

one for **power over time** and another for **cumulative power** to visualize the power generated by a solar panel.

You can simply execute python file by running following command:

```
pi@group2-GroundStation:~/Documents/Python $ python3 solar_power.py
```

Then the graph will show up, you can modify the code as you like.