Aplicações Avançadas de Instrumentação MIEF/MIEB

1st Semester - 2023/2024

Universidade Nova de Lisboa - Faculdade de Ciências e Tecnologia

Lab 5 - Streamlit

Contents

Previous Requirements

Learn about Streamlit

Streamlit is an open source app framework in Python language. It helps to create web apps for data science and machine learning. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.

In https://streamlit.io/ you can find all the documentation and examples, necessary to learn how to use this framework. Particular attention to https://docs.streamlit.io/library/api-reference where you can find all the components that is possible to integrate in your web pages.

Install and Test Streamlit in Anaconda of PC

To install Streamlit on your PC (https://anaconda.org/conda-forge/streamlit):

- Open an QT console (or anaconda prompt):
- Do: *conda install -c conda-forge streamlit*

To test Streamlit (https://streamlit.io/):

- In the QT console (or anaconda prompt):
- Do: *streamlit hello*
    - Note1: This action should open a new page in your browser with a test page.
    - To end program, do Ctrl^C in anaconda prompt before close web page.

The First Program

1. Open Spyder in your PC and write the following python script:

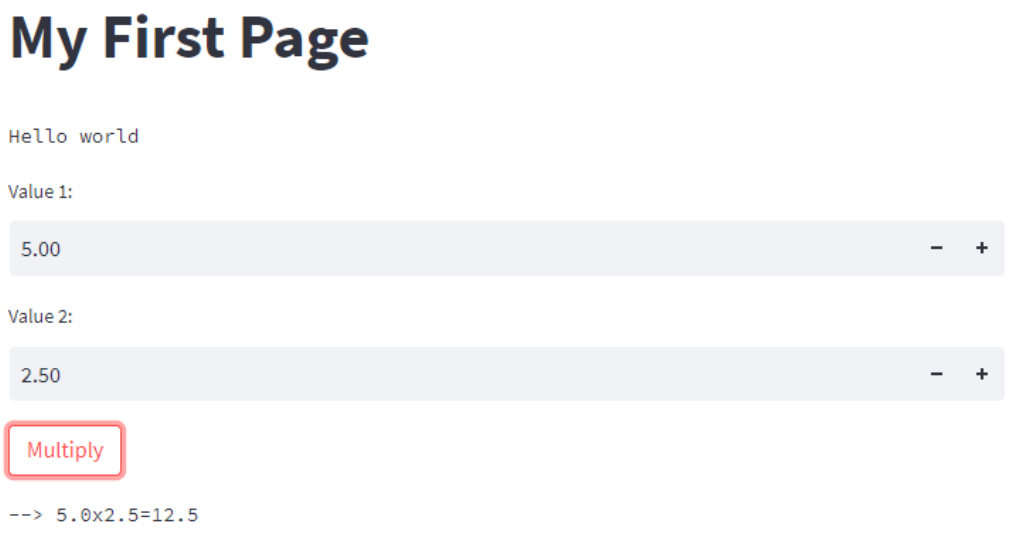
    ```
    import streamlit as st

    st.title("My First Page")

    st.text("Hello world")
    ```

2. Save the script in a folder of your choice.
3. From anaconda prompt do:
    a. change for the same folder where your program is
    b. streamlit run NameOfYourPogram.py

   i. Note: To end program, do Ctrl^C in anaconda prompt before close web page.
4. You should have now a new page showing the contents that you programmed.

The *st.title* and *st.text* are components that you have available to add in your pages. There is a collection of those elements that you can found in: https://docs.streamlit.io/library/api-reference. Explore those components and find the ones related with inputs.

1. Let's modify the previous program, into a program that multiplies two numbers.
2. Add 2 input number input boxes (*st.number_input*)
3. Add a button to do the calculation (*st.button*)
4. Print the result of the multiplication of the two numbers.



Plotting Data

Using the streamlit and its matplotlib components (*st.pyplots*):

1. Add a button to generate data, using the function given in exercise 3 on Lab 1

```
import numpy as np

# sampling rate
sr = 2000
# sampling interval
ts = 1.0/sr
t = np.arange(0,1,ts)

freq = 1.
x = 3*np.sin(2*np.pi*freq*t)
```

freq = 4
*x += np.sin(2*np.pi*freq*t)*

freq = 7
x += 0.5 np.sin(2*np.pi*freq*t)*

2. After the data is generated, plot it using the component: *st.pyplots.*
3. Make your interface more interesting and allow the user to specify: sr and frequencies 1 to 3

Data transfer from Streamlit to RPI

MQTT protocol allows machines to exchange data. So, we can use this protocol to get the data acquired by the acquisition machine and show it in streamlit. In the github of this course, there are two programs that allow you to test this concept:

- **Lab5_stream_mqtt.py** – run from the anaconda prompt and can publish and subscribe data. The default parameters are:
    - Broker: 192.168.1.98
    - Publication topic: AAI/PV/cmd
    - Subscription topic: AAI/PV/reply
    - Note: you should change the 'PV' for something that identifies your group
- **Lab5_RPI_mqtt.py** – run in the RPI and can also publish and subscribe data. The default parameters are:
    - Broker: 192.168.1.98
    - Publication topic: AAI/PV/reply
    - Subscription topic: AAI/PV/cmd
    - Note: you should change the 'PV' for something that identifies your group

Try these programs and studied them, for a better understanding how two machines can interact.

Challenge

In lab4 you developed a program that transfer ecg data from the RPI to the PC. Starting from the previous examples, modify them so you can plot the ecg data that you received in the streamlit interface.