

Task 2: 555 Timer

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt

from natsort import natsorted

from matplotlib import cm
from glob import glob

cmap = cm.get_cmap('viridis')
plt.rcParams.update({'font.size': 14})
```

/tmp/ipykernel_810156/1263976303.py:9: MatplotlibDeprecationWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed two minor releases later. Use ``matplotlib.colormaps[name]`` or ``matplotlib.colormaps.get_cmap(obj)`` instead.

```
cmap = cm.get_cmap('viridis')
```

```
In [ ]: dataPaths = glob("../data/potentiometer/frequencyVariation/*")
dataPaths = natsorted(dataPaths)
print(dataPaths)
```

```
['../data/potentiometer/frequencyVariation/100nF.txt', '../data/potentiometer/frequencyVariation/300nF.txt', '../data/potentiometer/frequencyVariation/500nF.txt']
```

```
In [ ]: fig, axes = plt.subplots(3, 1, figsize=(6,6), sharex=True)

# Extract labels from file paths
labels = [f"C={el.split('/')[0:5]}" for el in dataPaths]

for i, path in enumerate(dataPaths):

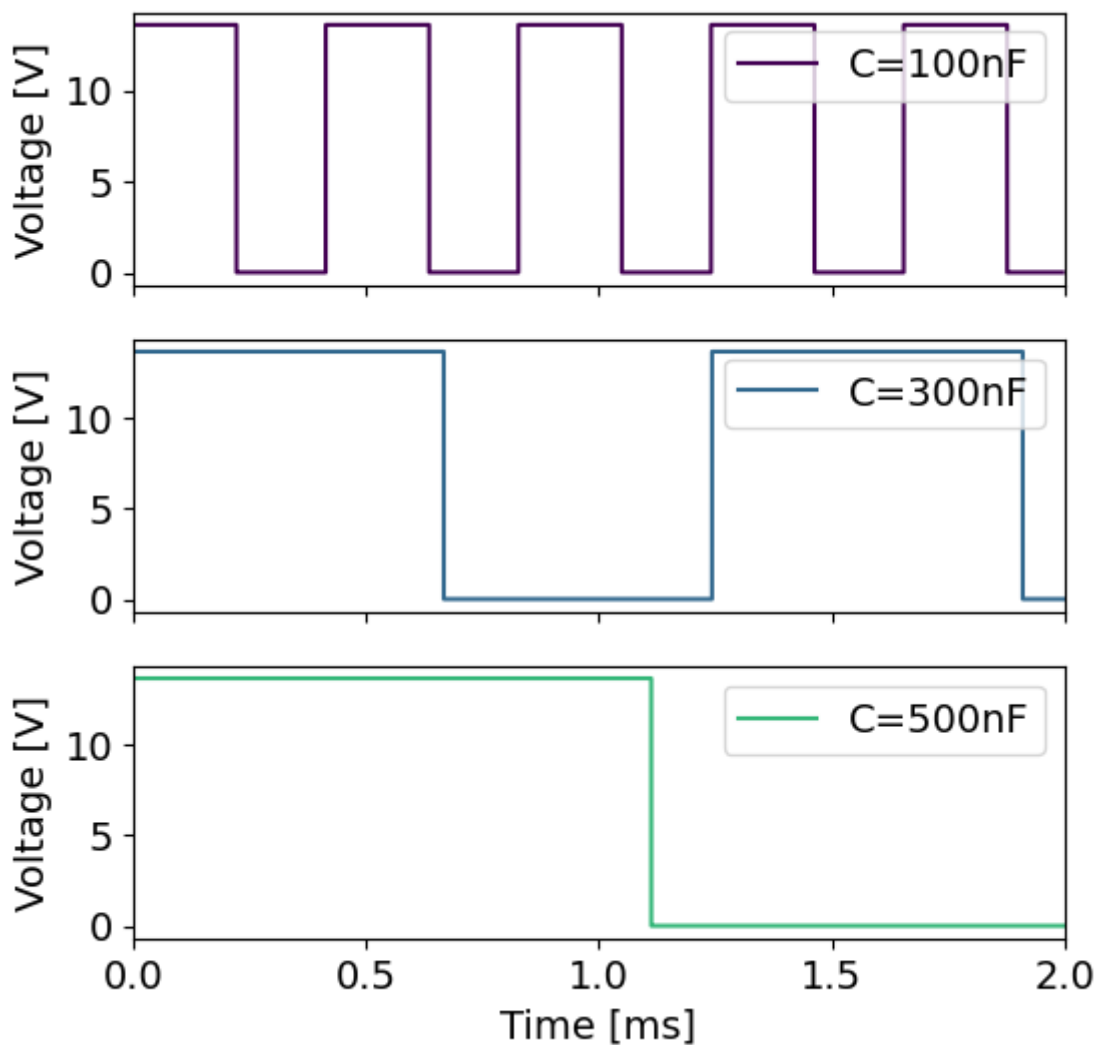
    data = np.loadtxt(path, skiprows=1)

    axes[i].plot(data[:,0]*1000, data[:,1], color=cmap(i/len(axes)), label=labels[i])

    axes[i].set_ylabel("Voltage [V]")
    axes[i].legend(loc="upper right")

plt.xlim(0, 2)
plt.xlabel("Time [ms]")
```

```
Out[ ]: Text(0.5, 0, 'Time [ms]')
```



```
In [ ]: dataPaths = glob("../data/potentiometer/potentiometerVariation/*")
dataPaths = natsorted(dataPaths)
print(dataPaths)

['../data/potentiometer/potentiometerVariation/p0.txt', '../data/potentiometer/potentiometerVariation/
p25.txt', '../data/potentiometer/potentiometerVariation/p50.txt', '../data/potentiometer/potentiometer
Variation/p75.txt', '../data/potentiometer/potentiometerVariation/p100.txt']
```

```
In [ ]: fig, axes = plt.subplots(5, 1, figsize=(6,6), sharex=True)

labels = ["0%", "25%", "50%", "75%", "100%"]

for i, path in enumerate(dataPaths):

    data = np.loadtxt(path, skiprows=1)

    axes[i].plot(data[:,0]*1000, data[:,1], color=cmap(i/len(axes)), label=labels[i])

    axes[i].set_ylabel("Voltage [V]")
    axes[i].legend(loc="upper right")

plt.xlim(0, 5)
plt.xlabel("Time [ms]")
```

```
Out[ ]: Text(0.5, 0, 'Time [ms]')
```

