ASHA N – Daily Progress Report on 07/12/2020

**Updated Code of the GUI (Configurator)**

import tkinter as tk

import matplotlib

matplotlib.use('TkAgg')

import numpy as np

import matplotlib.pyplot as plt

from math import pi

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

from matplotlib.figure import Figure

def plot ():

    F=int(signal.get())

    input\_volt=int(amplitude.get())

    Fs=int(sampling.get())

    #F = 1.e2          # No. of cycles per second, F = 500 Hz

    T = 10.e-3         # Time period, T = 2 ms

    #Fs = 1.e3        # No. of samples per second, Fs = 50 kHz

    Ts = 1./Fs        # Sampling interval, Ts = 20 us

    N = int(T/Ts)     # No. of samples for 2 ms, N = 100

    #input\_volt=230    #input voltage.

                    #Vout=Asine(2\*pi\*f\*t)

    print(F)

    print(Fs)

    print(input\_volt)

    t = np.linspace(0, T, N)

    amp= 1.414\*input\_volt

    a=[0]\*(N)

    for n in range(N):

        a[n]= (amp \*np.sin(2\*pi\*F\*n/Fs))

        print(a[n])

    #signal = (amp \* np.sin(2\*np.pi\*F\*t))

    fig = Figure(figsize=(5,5))

    a1 = fig.add\_subplot(111)

    a1.plot(t,a,color='blue')

    a1.set\_title ("Estimation Grid", fontsize=16)

    a1.set\_ylabel("Y", fontsize=14)

    a1.set\_xlabel("X", fontsize=14)

    canvas = FigureCanvasTkAgg(fig, master=root)

    canvas.get\_tk\_widget().place(x=30, y=200)

    canvas.draw()

def select():

    option = measurement\_choices.get()

    if option=="Voltage - AC":

        dc\_range.place\_forget()

        ac\_range.place(x=230, y=35)

    else:

        ac\_range.place\_forget()

        dc\_range.place(x=230, y=35)

def take\_input():

    option = measurement\_choices.get()

    if option=="Voltage - AC":

        ac\_data=ac\_choices.get()

        print(option)

        print(ac\_data)

    else:

        dc\_data=dc\_choices.get()

        print(option)

        print(dc\_data)

root = tk.Tk()

root.title("Configurator")

root.geometry("650x700")

measurement\_choices = tk.StringVar()

ac\_choices = tk.IntVar()

dc\_choices = tk.IntVar()

measurement\_type = {"Voltage - AC", "Voltage - DC",}

AC\_range = {110,230,440,500}

DC\_range = {10,50,48}

measurement\_choices.set("Measurement Type")

ac\_choices.set("Select AC range")

dc\_choices.set("Select DC range")

label=tk.Label(root,text="Select the Type and range")

label.place(x=130, y=10)

measur\_type = tk.OptionMenu(root,measurement\_choices, \*measurement\_type)

measur\_type.configure(width=15)

measur\_type.place(x=50, y=35)

AC\_range=sorted(AC\_range)

ac\_range = tk.OptionMenu(root,ac\_choices, \*AC\_range)

#ac\_range.visible = False

ac\_range.configure(width=15)

#ac\_range.place(x=230, y=35)

#ac\_range.pi = ac\_range.place\_info()

DC\_range=sorted(DC\_range)

dc\_range = tk.OptionMenu(root,dc\_choices, \*DC\_range)

#ac\_range.visible = False

#dc\_range.place(x=230, y=75)

dc\_range.configure(width=15)

#dc\_range.pi = dc\_range.place\_info()

#Signal Frequency

signal\_freq\_label = tk.Label(text="Signal Frequency")

signal\_freq\_label.place(x=30, y=100)

signal = tk.DoubleVar()

signal\_freq\_entry = tk.Entry(root, width = 15, textvariable = signal)

signal\_freq\_entry.place(x=30, y=130)

#Amplitude

amplitude\_label = tk.Label(text="Amplitude")

amplitude\_label.place(x=160, y=100)

amplitude = tk.DoubleVar()

amplitude\_entry = tk.Entry(root, width = 15, textvariable = amplitude)

amplitude\_entry.place(x=160, y=130)

#Sampling Frequency

sampling\_freq\_label = tk.Label(text="Sampling Frequency")

sampling\_freq\_label.place(x=290, y=100)

sampling = tk.DoubleVar()

sampling\_freq\_entry = tk.Entry(root, width = 15, textvariable = sampling)

sampling\_freq\_entry.place(x=290, y=130)

next\_button = tk.Button(root,text="Next", bg="gray", command=select)

next\_button.place(x=80, y=150)

print\_button = tk.Button(root,text="Print", bg="gray", command=take\_input)

print\_button.place(x=170, y=150)

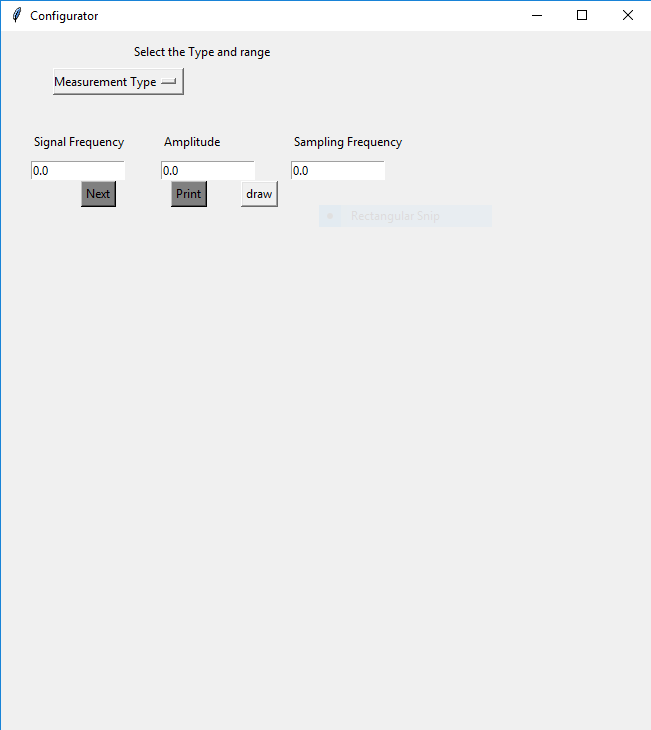
graph\_button = tk.Button(root, text="draw", command=plot)

graph\_button.place(x=240, y=150)

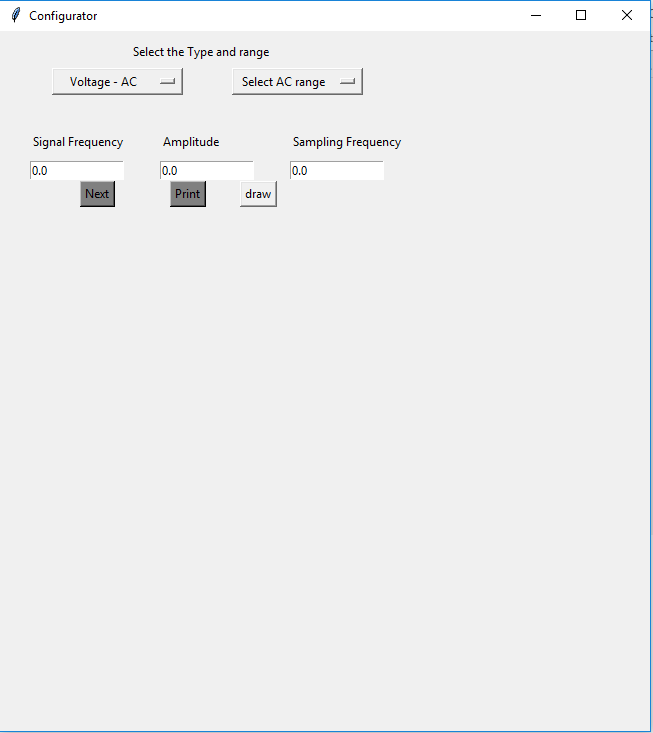
root.mainloop()

This is the GUI for Data Acquisition System, in this GUI we have

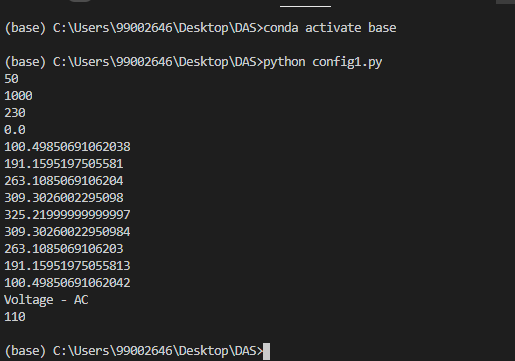
* Dropdown menu for selecting the Measurement Type.
* Label and Entry box for
* Signal Frequency
* Amplitude
* Sampling Frequency
* Then we have three buttons namely Next, Print and Draw
* If we click the Next button, then the control goes to source method.



* In Measurement Type Option Menu, if we select one of the option like Voltage -AC or Voltage -DC, then the range option menu will be enabled depends on the Measurement Type input.
* For example, in the below screenshot Voltage – AC is selected
* Range option menu will have enabled for AC range.



* If the print button is clicked, then the given input and samples will print in terminal.



* When the Draw button is clicked, the graph will be drawn in the GUI.

