ASHA N – Daily Progress Report on 05/12/2020

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

import tkinter as tk

root = tk.Tk()

'''

def take\_input():

    ac\_data=ac\_choices.get()

    dc\_data=dc\_choices.get()

    print("numbers = {}, {}".format(ac\_data, dc\_data))

'''

def decide(option):

    if option=="Voltage - AC":

        dc\_range.configure(state="disabled")

    else:

        ac\_range.configure(state="disabled")

root.title("Configurator")

root.geometry("400x200")

measurement\_choices = tk.StringVar()

ac\_choices = tk.IntVar()

dc\_choices = tk.IntVar()

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

AC\_range = {1, 50, 100, 200,300,400,500}

DC\_range = {1, 6, 12, 18,24,30,36,42,48}

measurement\_choices.set("Measurement Type")

ac\_choices.set("AC range")

dc\_choices.set("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.configure(width=10)

ac\_range.place(x=230, y=35)

DC\_range=sorted(DC\_range)

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

dc\_range.place(x=230, y=75)

dc\_range.configure(width=10)

option1=measur\_type.cget()

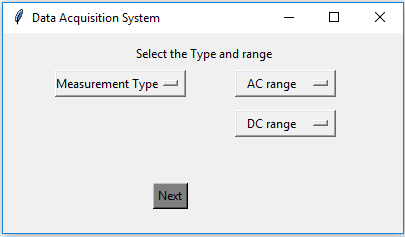
next\_button = tk.Button(root,text="Next", bg="gray",command = decide(option1))

next\_button.place(x=150, y=150)

root.mainloop()

This is the GUI for Data Acquisition System, in this GUI we have three dropdown menus namely measurement type, AC and DC ranges.

* In Measurement Type Option Menu, if we select one of the option then option menu of the other option will be disabled.
* If we click the Next button, then the control goes to source method.



Unable to take the screenshot of dropdown button in snippet tool.

**Code of GUI (Display)**

import tkinter as tk

root = tk.Tk()

def clear\_widget\_text(type\_label,unit\_label):

    type\_label['text'] = ""

    unit\_label['text'] = ""

def close\_window():

    root.destroy()

root.title("Calculation display")

root.geometry("270x150")

label=tk.Label(root,text="Selected measurement type and range")

label.place(x=25, y=10)

type\_label=tk.Label(root,text="Measurement type",borderwidth=2, relief="groove")

type\_label.place(x=40, y=50)

type\_label.configure(width=15)

unit\_label=tk.Label(root,text="Range",borderwidth=2, relief="groove")

unit\_label.place(x=160, y=50)

unit\_label.configure(width=8)

clear\_button = tk.Button(root,text="Clear", bg="gray",command=lambda : clear\_widget\_text(type\_label,unit\_label))

clear\_button.place(x=50, y=90)

clear\_button.configure(width=8)

exit\_button = tk.Button(root,text="Exit", bg="gray",command = close\_window)

exit\_button.place(x=150, y=90)

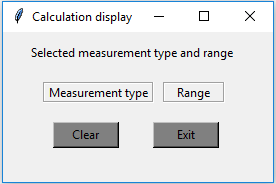
exit\_button.configure(width=8)

root.mainloop()

This is the GUI for Display to display the result of measurement calculations.

In this GUI we have following,

* One Label
* Two Labels for displaying the selected Measurement type and Range in the Configurator.
* Two buttons namely Clear and Exit
* Clear button to empty the label.
* Exit button to exit the user interface.



When the Clear button is clicked, label will be cleared.

