**Continuous Signal Builder GUI Application**

This project is a Python-based GUI application that allows users to build, modify, and visualize continuous signals by adding different signal segments.

**The Features of the Project:**

**Add Signal Segments:** Users can add different types of signal segments — either linear (Y) or step (S) — by specifying start and end points in time and amplitude.

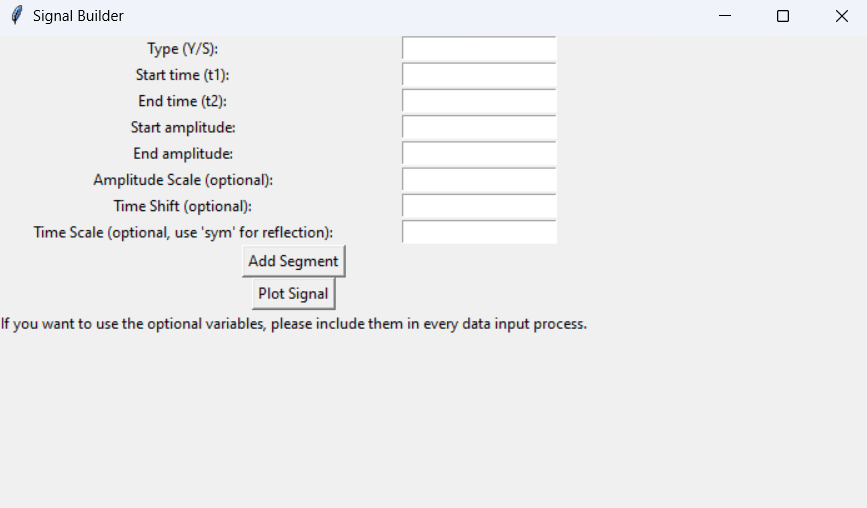
**Amplitude Scaling:** Users can optionally apply a scaling factor to the amplitude of each segment.

**Time Shifting and Time Scaling:** Time shifting allows moving the signal along the time axis, and time scaling allows stretching or compressing the signal in time.  
 Users can also apply a reflection across the time axis by entering "sym" in the time scaling field.

**Signal Visualization:** The complete constructed signal can be visualized through a plotted graph with time and amplitude axes, gridlines, and a legend.

**User-Friendly Interface with Error Handling:** The GUI provides clear error messages and success notifications, making it easy to identify and correct input mistakes.

**The Inputs in the User Interface:**

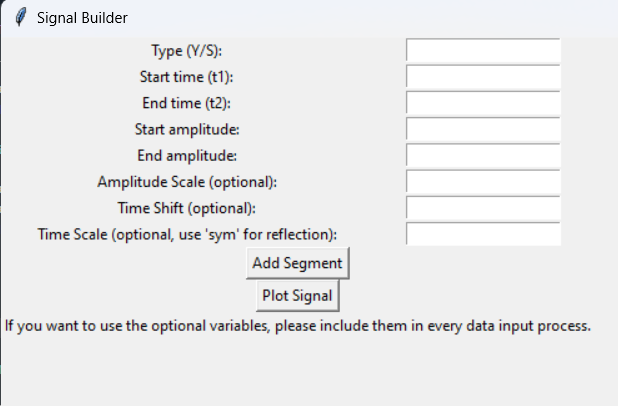
****

**The Requirements of the Project:**

To run this program, the following Python libraries must be installed:

* **NumPy** (numpy)  
   Used for numerical operations, such as creating and manipulating the time and signal arrays.
* **Matplotlib** (matplotlib)  
   Used for plotting and visualizing the constructed signal.
* **Tkinter** (tkinter)  
   Used for building the graphical user interface (GUI).  
   (*Note: Tkinter is included by default with standard Python installations.*)

**How to Use:**

****

* You can define each segment of the continuous signal by specifying its start time, end time, start amplitude, and end amplitude. By using the first five input fields in the interface, you can build and customize a continuous signal piece by piece.
* If you want to apply amplitude scaling, time shifting, or time scaling, it is recommended to use only one of these operations at a time for more accurate results.
* Additionally, if you want to apply any of these operations, you must enter the optional input values each time you add an amplitude segment. If you wish to create a highly processed (complex) signal, it is better to separate the operations into stages and apply them individually at each stage.

**Developer Information:**

**Developer Name:** Ahmed Dillo

**Email:** ahmeddillo4444@gmail.com

**GitHub:** <https://github.com/Ahmeddillo>

**Date:** 26/04/2025