Folder - Code

fluid_handler_gui.py

This is the primary control script for the automated fluid handler shown in SolidWorks_Fluid_Handler.PNG. The purpose of this system is to act as a full service dynamic cell culture platform. The user can choose between solutions, flowrates, and collection modes. The script depends on pyserial and tkinter for its primary functions. It initializes the Tk() object which provides us with the gui backend. Each subsequent page class contains an instance of tkinter's frame object which is instantiated as a child of the main App class. Each page has a set of functions tailored specifically to its needs. I'll turn your attention to a few functions of interest.

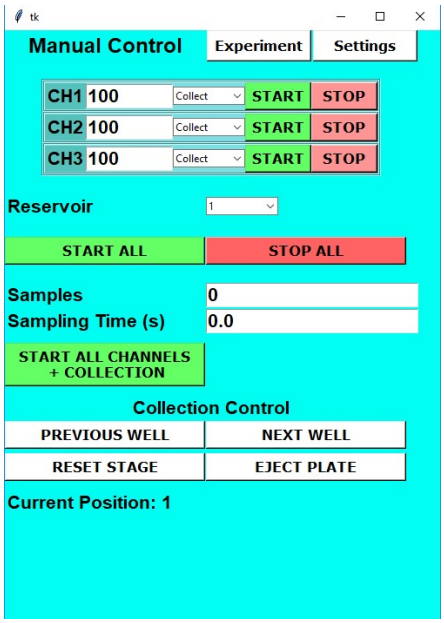

Figure 1: Manual Page

The **start_and_collect** function under the **ManualPage** class checks that all entries of the manual page (fig.1) are of the correct ranges and types. If so, we open a tk.TopLevel object which functions as our frame and stores our variables. Then, we push all of the parameters to each device and flow fluid through the system, moving the collection plate as we go keeping track of sampling time using python's **time** module.

Next, the **add_step** and **delete_step** functions under the **AutomaticPage** (fig2.) class which allow the user to create an experimental protocol. These functions were organizationally challenging because each time the user makes a change the steps have to be renumbered, the graphical numerical representations updated, and the parameters recounted so that the actual device doesn't overshoot its collection area or volume. The graphics were the trickiest part because of how the labels needed to be gridded on the page. I enjoyed writing this functionality the most.

Figure 2: Automatic Page

Lastly, I'll just point to the **file_save** and **file_load** functions under the **AutomaticPage** class, which are very simple, but hopefully demonstrate that I'm comfortable importing and exporting other file types within python.

*A note about authorship of this code. During the Spring 2019 semester an undergraduate contributed to this work by adding an extra component called a two-switch. His contributions can be found by searching "my2Switch" within the code.