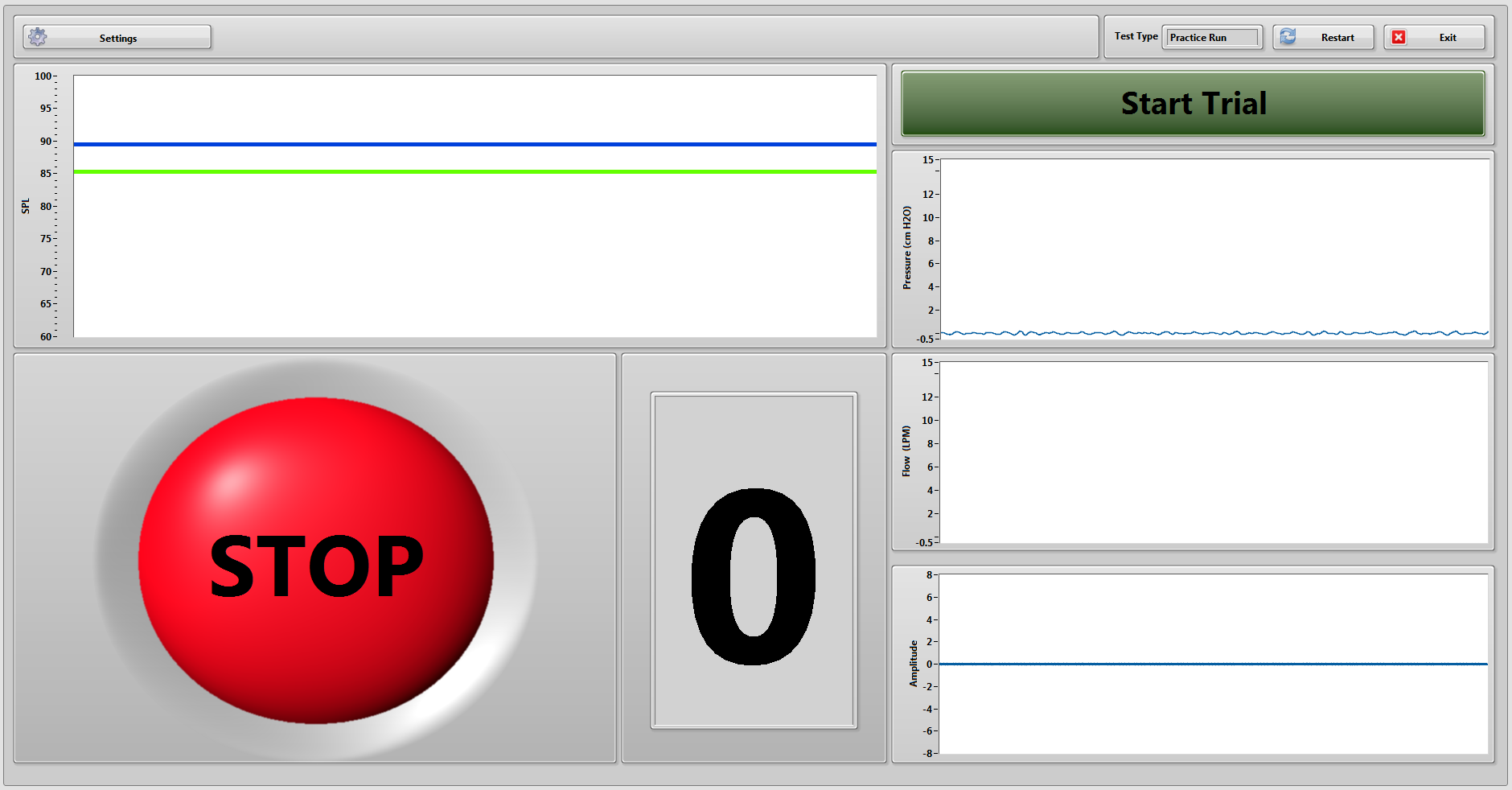
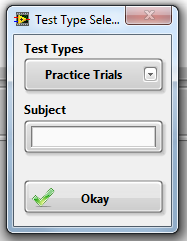
# Complete/Labial Intteruption Collection

## Purpose and Use

This program is used to collect data with the complete mechanical interrupter or through labial interruptions. It is one of the four main collection programs in the Master Aero controller. There are four charts displaying live data. SPL is shown with an upper and lower cursor to be used as a target range. Pressure, flow, and acoustics are also displayed.

|  |  |  |
| --- | --- | --- |
| Test Type | Delay | Interruptions |
| Practice | 0 | 0 |
| Mech Mask | 250 | 5 |
| Mech Mouth | 250 | 5 |
| Labial Normal | 0 | 0 |
| Labial Quiet | 0 | 0 |

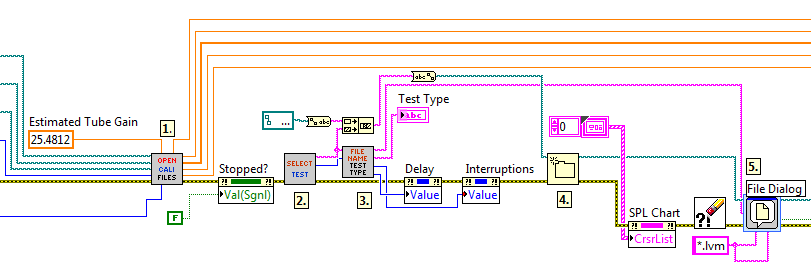
This program will be opened through the Master Controller VI. At first, the user will be prompted to select the test type and enter subject info. The different test types will control what the default configuration will be when data starts being collected. It also changes the default name and save path for the saved data. The settings themselves are hidden by default. They can be viewed by clicking the Settings button. Hover over each control to view a short description.

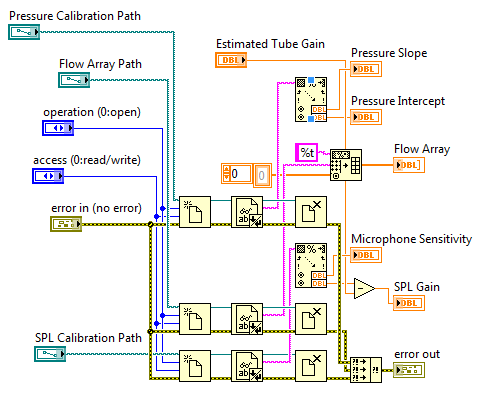
After the test type and subject info is entered, the program will start reading (but not recording) in data from the DAQ board. Flow, pressure, and acoustic data are read in directly from the DAQ board while the SPL is calculated from the acoustic data.

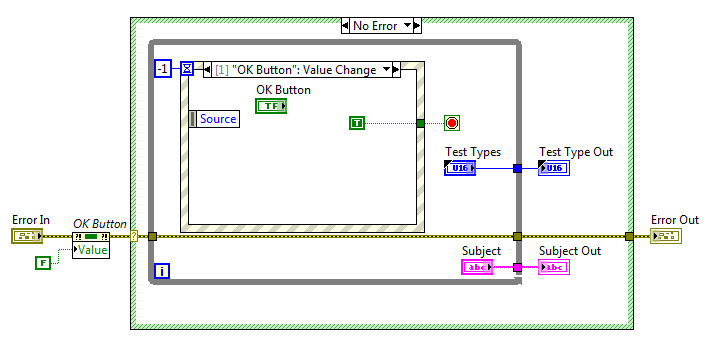
Data will only be saved to file during a trial, while the stoplight is green (GO). Once you click Start Trial, the button itself will turn red and now say stop trial. You can stop the recording before the trial is complete by pressing this button again. Assuming you want to run a complete trial, there will be a 3 second countdown before the stoplight turns green and starts writing collected data to a file.

Unless you click the stop trial button, the recording will stop automatically after however many seconds you set the trial length to be. If you wish to select a different trial type, you can click the Reset button and the test type select window will pop up again. If you are done collected, click exit.

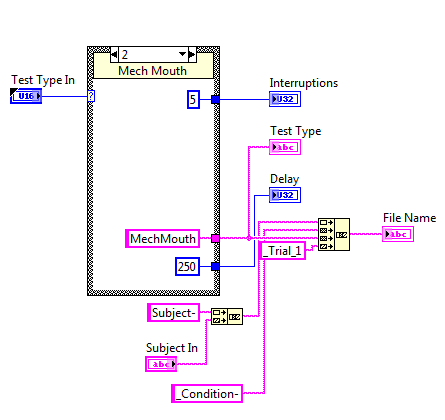
## The block diagram

This program is organized into three parallel loops within a master control loop. Before the parallel loops, there are a couple of SubVIs that get calibrations values, and set the default interruption values.

First, there is the Open Calibration Files VI. This takes in the paths for the pressure, flow array, and SPL calibration text files then opens them to extract the data as spreadsheet strings. These are then read to obtain their values as doubles (an array of doubles in the case of the flow array).

For the SPL gain, the microphone sensitivity is read along with the estimated gain from the … gain estimator. The estimated tube gain was calculated experimentally and is a constant that is then subtracted from the gain estimation to give us our actual gain. This value is then used in the SPL calculator later in the program.

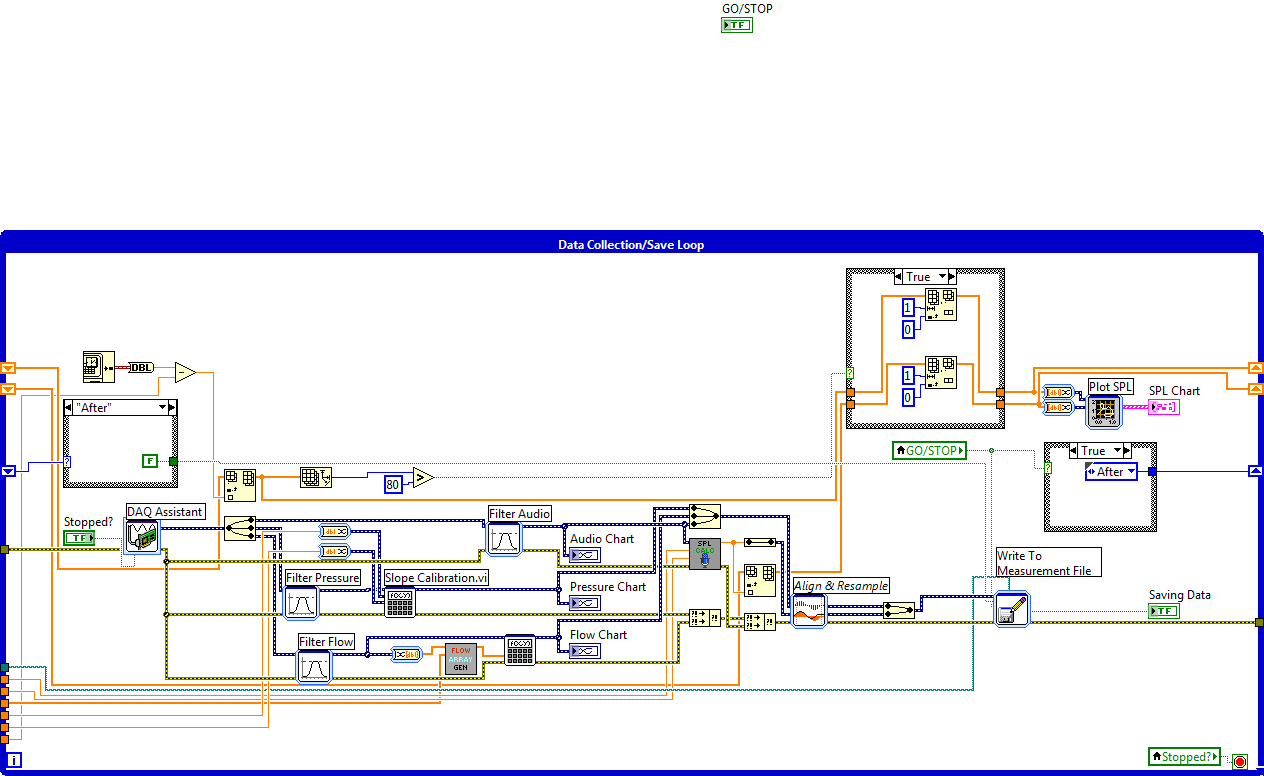
Second is the test type selector. It simply outputs the selected test type and subject info string when the user click the okay button or hits enter.

The third SubVI takes in the test type and subject info from the selector and creates a file time based on this information. It also outputs the default balloon delay and number of interruptions, depending on which test was selected. These two values are set through the two property nodes following this SubVI.

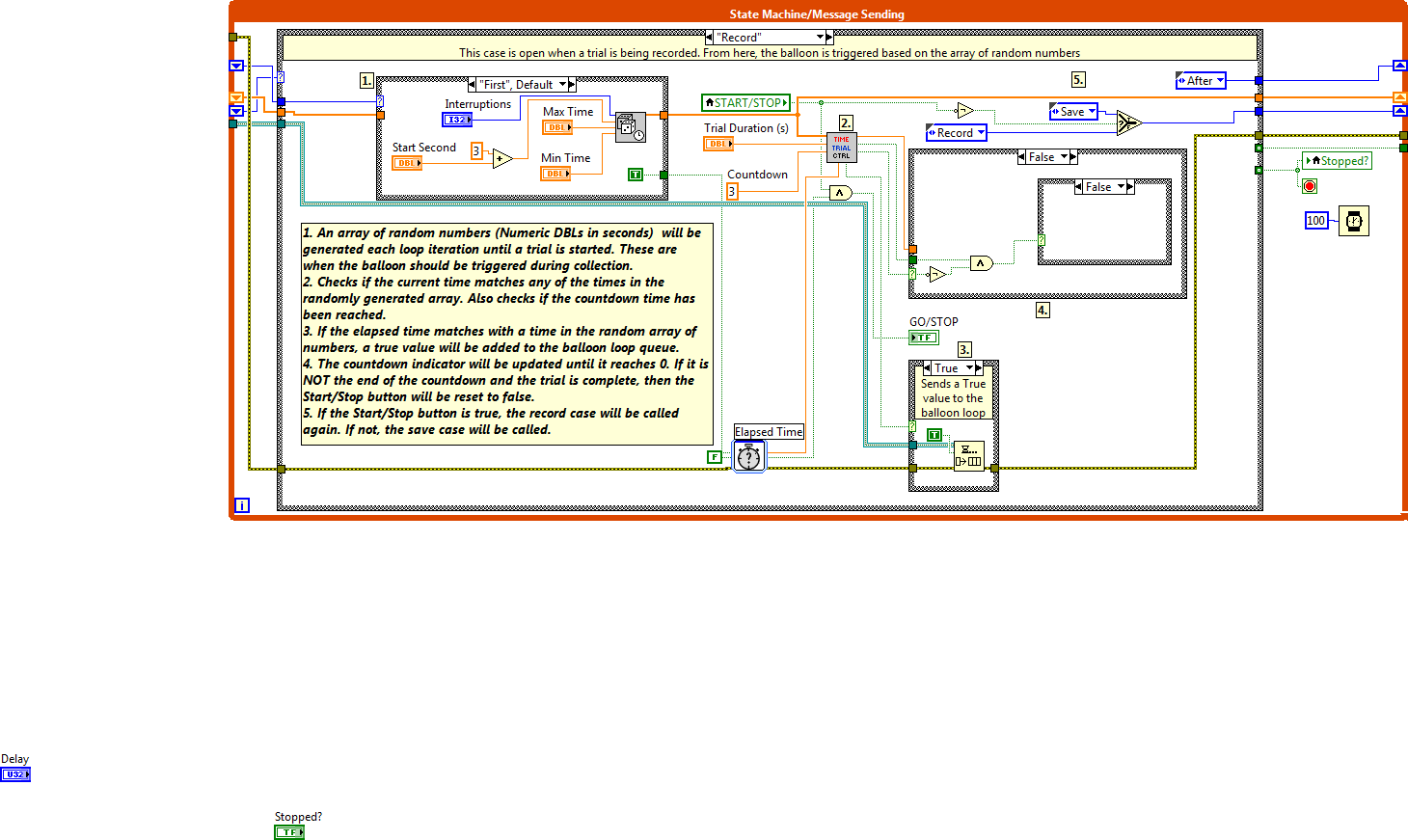
A folder is then created for the specific subject name/number. Currently, he base save path is to the peds\_data folder on my desktop. This will need to change once I am gone and my user account is no longer usable. The path itself is <C:\Users\scholp\Desktop\peds\_data\subject-> The new folder will simply be the same name/number that is input into the Test Selection SubVI. If the folder exists, an error is thrown by the create folder function. This error is then cleared by the Clear Errors VI that follows.

Through the file dialog express VI, the user is prompted to confirm the save location of the data that will be saved in this session. The user can change the location here if desired.

### Data collection and Saving loop

Here is the data collection/save loop. It reads from three channels on the DAQ board using the DAQ Assistant. It takes in acoustic, pressure, and flow data from their respective devices. Data is then filtered, and run through calibration curves to obtain actual values. Other things to note are the pair of “First-After” case structures. This this sets the reset value on the Write Measurement to file VI unless the GO/STOP light is on. Also, the x-axis of the SPL graph is controlled by building an array of values that is limited to 80 points.

### State Machine/Trial control loop

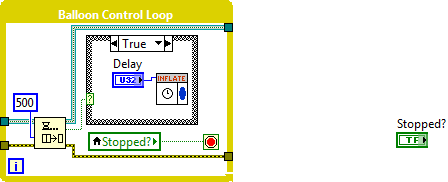
This loop is an event-controlled state machine. This controls the GO/STOP light, as well as when the balloon interruptions are triggered. A short description of each case is below.

* **Initialize** Sets the GO/STOP button to false (Red/Stop) and then calls the **wait** state.
* **Wait** Contains the event structure. The next state called depends on the event.
* **Record** Has the controls for the random array of numbers that will be used as interruptions. It also controls when to stop the trial and sends triggers to the Balloon Control Loop. Finally, it controls the stoplight and the START/STOP button.
* **Save** A previous version of the program utilized this state, but it is no longer called.
* **Exit** Stops the State Machine loop, which in turn stops the other two loops. This will also stop the master control loop.
* **Reset** Stops the State Machine loop, stopping the other two internal loops, but does not stop the master control loop, which then goes through another iteration of the loop, starting back with the calibration files SubVI.

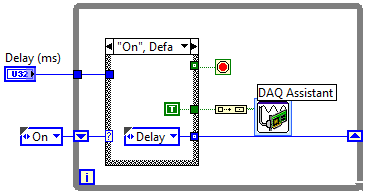
I mentioned an event structure. Here are the events:

* **START/STOP** Whenever this button is pressed, the record case is called. Whether or not data is recorded is controlled through the **Record** state.
* **Exit Button** calls the **Exit** case when the exit button is pressed.
* **Restart Button** calls the **Restart** case
* **Settings** Depending on if the settings button is true or false, the trial settings will become visible.

### Balloon Control Loop

Within the **Record** state of the trial control loop, if the current trial time matches one of the times in the randomly generated array of number, a true value will be added to the Boolean queue. When this happens, the case structure within the balloon control loop changes to true and the balloon is inflated for however many milliseconds the delay control is set to.

#### Inflate Subvi

The balloon interruption length is controlled by delaying the loop iteration by the input number of milliseconds.

First the “On” case is called which sets the DAQ digital output to True. In the “Delay” case, the loop waits for a given number of milliseconds before stopping the loop.