Aerodynamics Master Program Guide

For collecting and analyzing aerodynamic data.

By: Austin Scholp

Table of Contents

[Master Aero Program 4](#_Toc78530345)

[Front Panel Guide 4](#_Toc78530346)

[Main Menu 4](#_Toc78530347)

[Collect Data 4](#_Toc78530348)

[Calibrate Pneumotach 6](#_Toc78530349)

[Calibrate Helmet 6](#_Toc78530350)

[Randomize Tasks 7](#_Toc78530351)

[Calibrate SPL 7](#_Toc78530352)

[Extract Data from Text Files 7](#_Toc78530353)

[Extract Fundamental Frequency 8](#_Toc78530354)

[Analyze Data 8](#_Toc78530355)

[Data Clipper 9](#_Toc78530356)

[Fix LVMs 9](#_Toc78530357)

[Submit Error/Request 10](#_Toc78530358)

[Block Diagram 10](#_Toc78530359)

[Main Menu 10](#_Toc78530360)

[Aerodyanmics Collection 10](#_Toc78530361)

[Randomize Tasks 11](#_Toc78530362)

[Calibrate Pneumotach 11](#_Toc78530363)

[Calibrate Helmet 11](#_Toc78530364)

[Gain Estimator (SPL Calibration) 11](#_Toc78530365)

[Extract Data 11](#_Toc78530366)

[Extract F0 11](#_Toc78530367)

[Analysis Master 11](#_Toc78530368)

[Clipper 11](#_Toc78530369)

[LVM Fixer 11](#_Toc78530370)

[Timeout Check 11](#_Toc78530371)

[Master Analysis Program 11](#_Toc78530372)

[Front Panel Guide 11](#_Toc78530373)

[Main Menu 11](#_Toc78530374)

[Labial Interruption 13](#_Toc78530375)

[Mechanical Interruption 14](#_Toc78530376)

[Incomplete Interruption 15](#_Toc78530377)

[Airflow Redirector 16](#_Toc78530378)

[Block Diagram 16](#_Toc78530379)

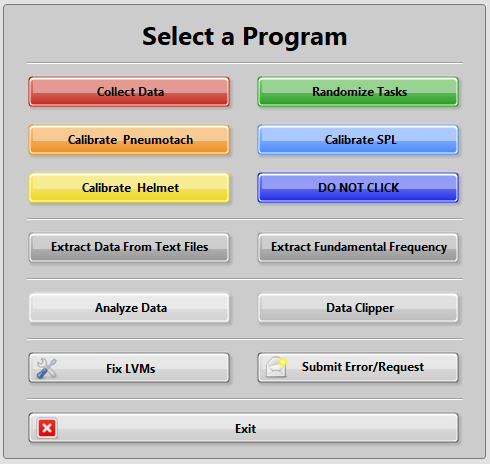
[Main Menu 16](#_Toc78530380)

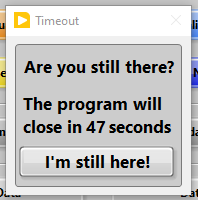
# Master Aero Program

The Master Aero program contains all of the collection and calibration programs used with our aerodynamic devices. This includes the complete/mechanical interrupter, the incomplete interrupter, the airflow redirector, and the singing helmet.

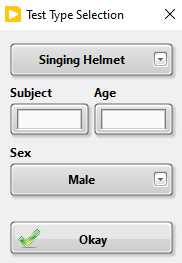
## Front Panel Guide

### Main Menu

The Master Aero program contains all of the collection, calibration, and analysis programs for our aerodynamic devices. The Front Panel is simply an array of different buttons. You can hover over each button for a short description of its function.

If the user spends more than two minutes without clicking any buttons, the timeout prompt will trigger. If there is no input after another minute, the program will close.

### Collect Data

The first thing that occurs when this VI is opened, the user will be prompted to select a folder where their data will be saved. This should default to the Saved Data folder on the Aero Drive.

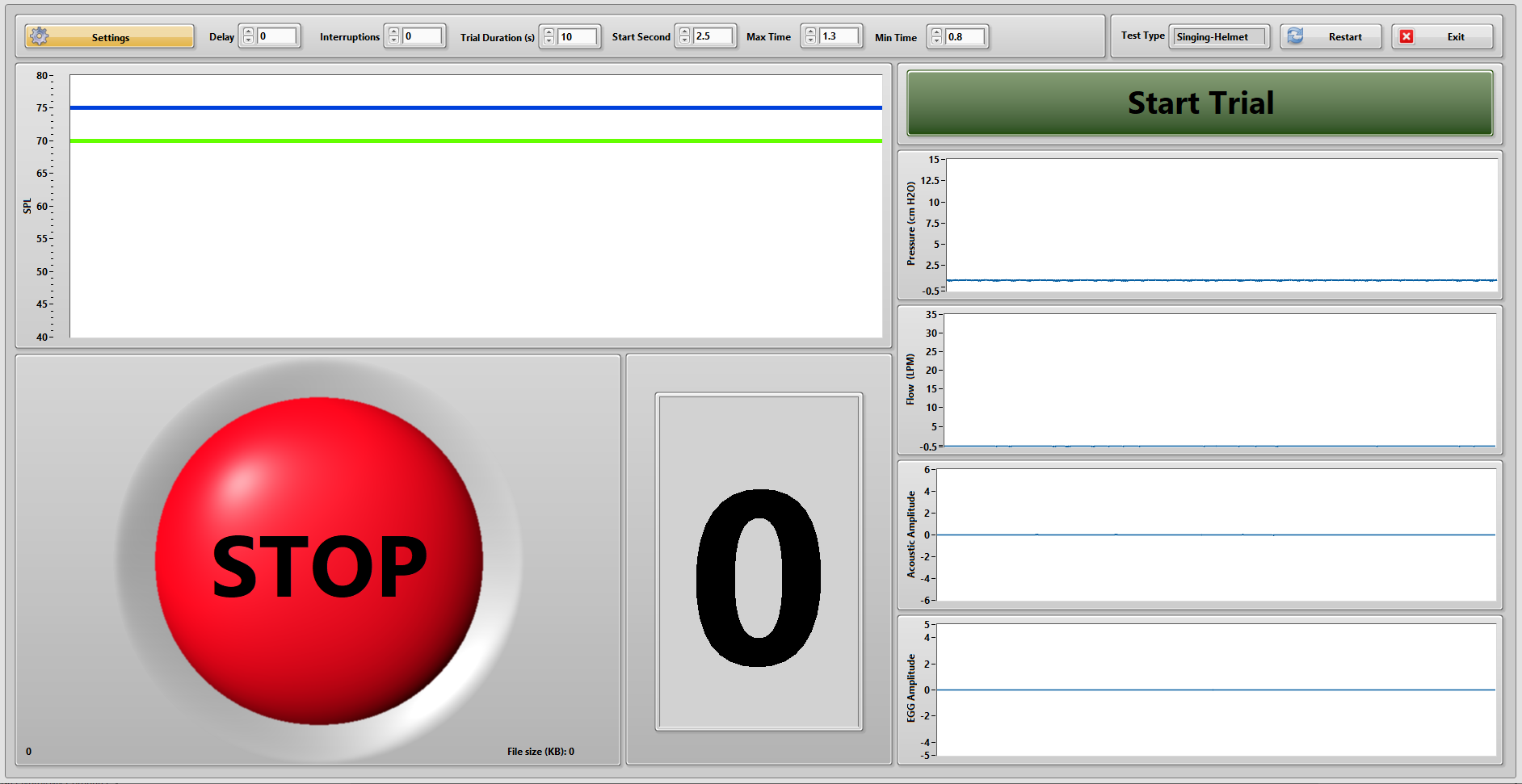
The user will then be prompted with the Test Type Selection VI. Here, they select what type of test you will be conducting. They should also enter the subject code, their age, select their sex, and then click **Okay**.

The settings and calibration data used during collection will depend on the type of test selected at this step.

The user will then be prompted to confirm the save location and file name.

Once the save file location is confirmed, signals will start being read from the NI DAQ board. Signals include:

* Sound Pressure Level (SPL) in dB – Upper Left
* Pressure in cm H2O – Upper Right
* Airflow in LPM – Below Pressure
* Acoustic Amplitude – Below Airflow
* EGG Amplitude – Bottom Right

All of these data will be saved whether or not you have sensors hooked up.

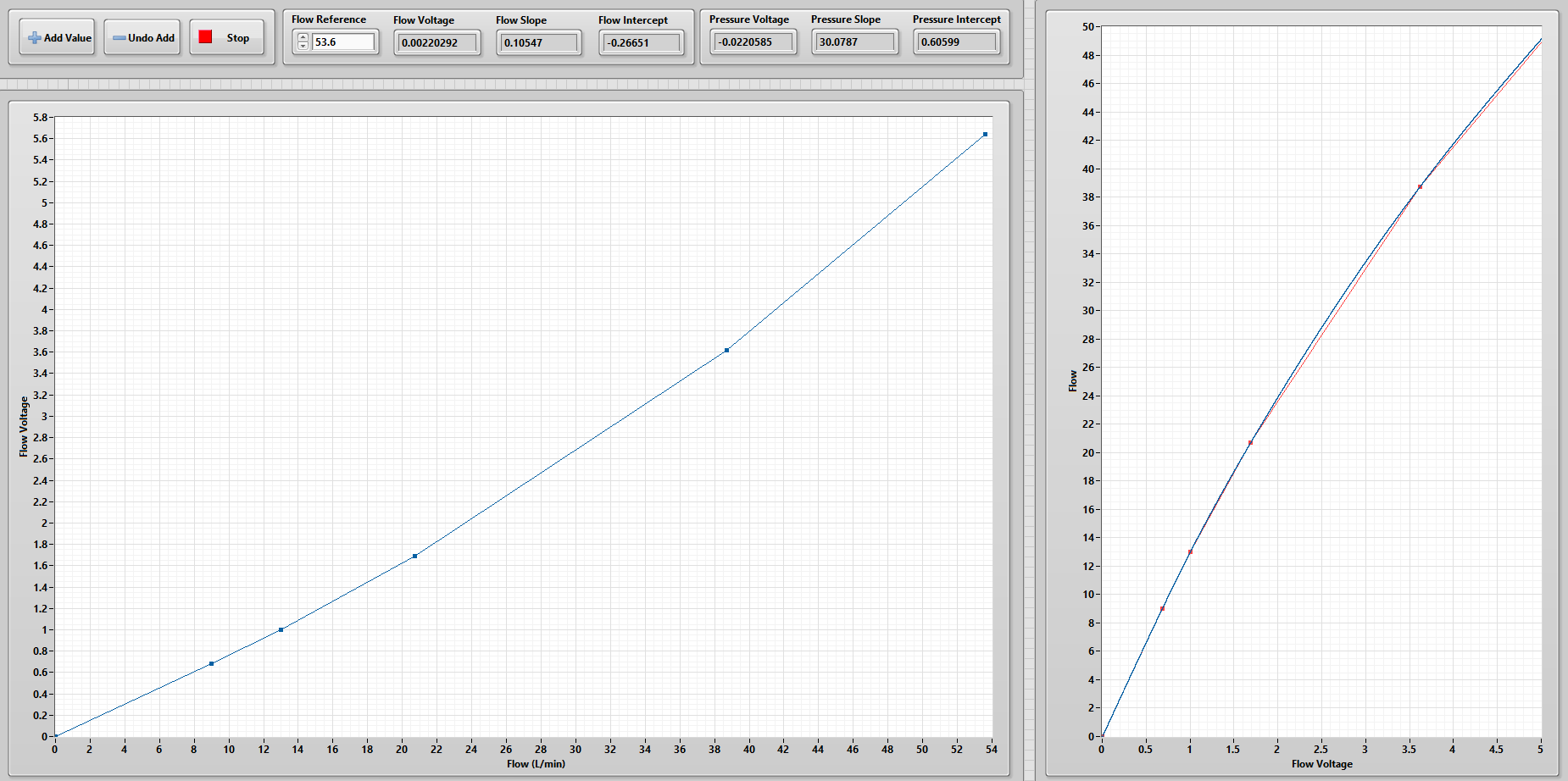
Clicking the **Settings** button will allow the user to alter the following:

* Delay – How long the balloon will be inflated
* Interruptions – How many interruptions there will be during a trial
* Trial Duration – How long until a trial will stop automatically
* Start Second – The earliest second of recording when the balloon can be triggered
* Max Time – The longest time between balloon triggers
* Min Time – The shortest time between balloon triggers

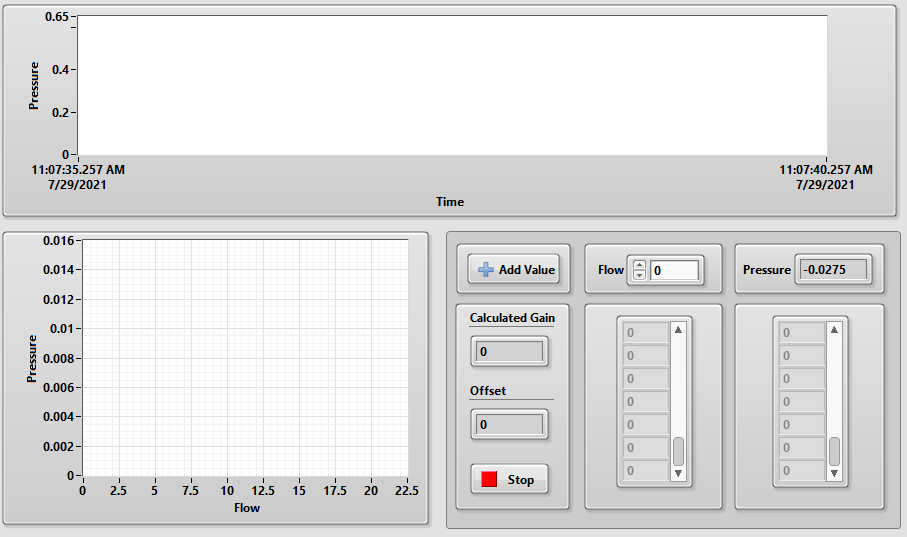
Other notes:

* The **Restart** button allows the user to change the test type without returning to the main menu. Default file names will adjust accordingly.
* The program is only recording when the stoplight shows a green **GO.**
* The small number in the bottom right indicates how many recordings the user has started.
* The size of the recording file is indicated to the bottom right of the stoplight. Use this to make sure that your data is being saved.
* When recording with the singing helmet, the flow data is saved as pressure.

### Calibrate Pneumotach

To calibrate the pneumotach, the user just needs to provide a reference flow value (i.e., tell the program the flow readout from the Omega flow meter) and add values. After adding at least 5 data points, you should end up with a plot similar to the one below. Do not worry, it is not supposed to be perfectly linear. Since only the flow portion of the pneumotach needs to be calibrated, **the P+ port should be disconnected from the interruption device during this process**.

### Calibrate Helmet

This program is used to define the relationship between the voltage read from a pressure transducer and the flow through the singing helmet.

The user just needs to provide the flow input while the transducer is connected to the pressure port on the helmet.

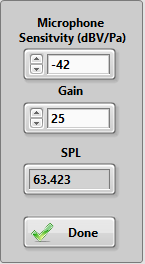
A *very* sensitive pressure transducer is needed to get a good calibration curve.

Note: when collecting data, the flow data is saved as pressure.

### Randomize Tasks

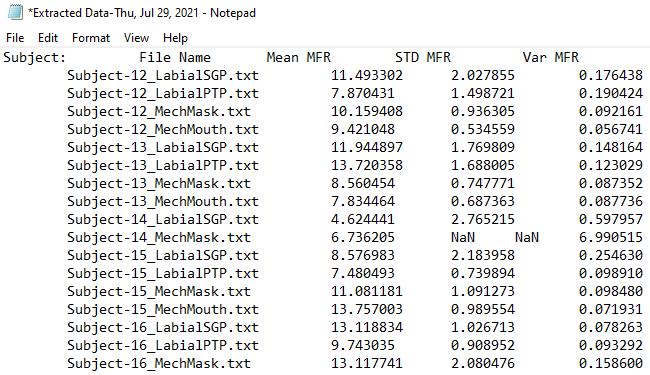
This simply displays a randomized order of tasks to be carried out while testing the complete interrupter. This is typically only used when you are comparing labial interruption to mechanical or complete interruptions.

### Calibrate SPL

To calibrate the SPL read from a microphone, you will need the microphone sensitivity in decibel-volts per Pascal (dBV/Pa). For the microphone currently in the complete interrupter, this value is -42. You will also need a calibrated SPL meter and a source of sound.

Set the sound source equidistant to the opening of the interrupter and the SPL meter. Then, adjust the gain on the program until the SPL readout matches the one seen on the SPL meter. This is not especially precise, but it gets the job done. Come up with a better way if you want.

### Extract Data from Text Files

**This program is used to get the means, standard deviations, and coefficients of variation from all the analysis text files in a folder. Note that the front panel for this program will not appear to the user.

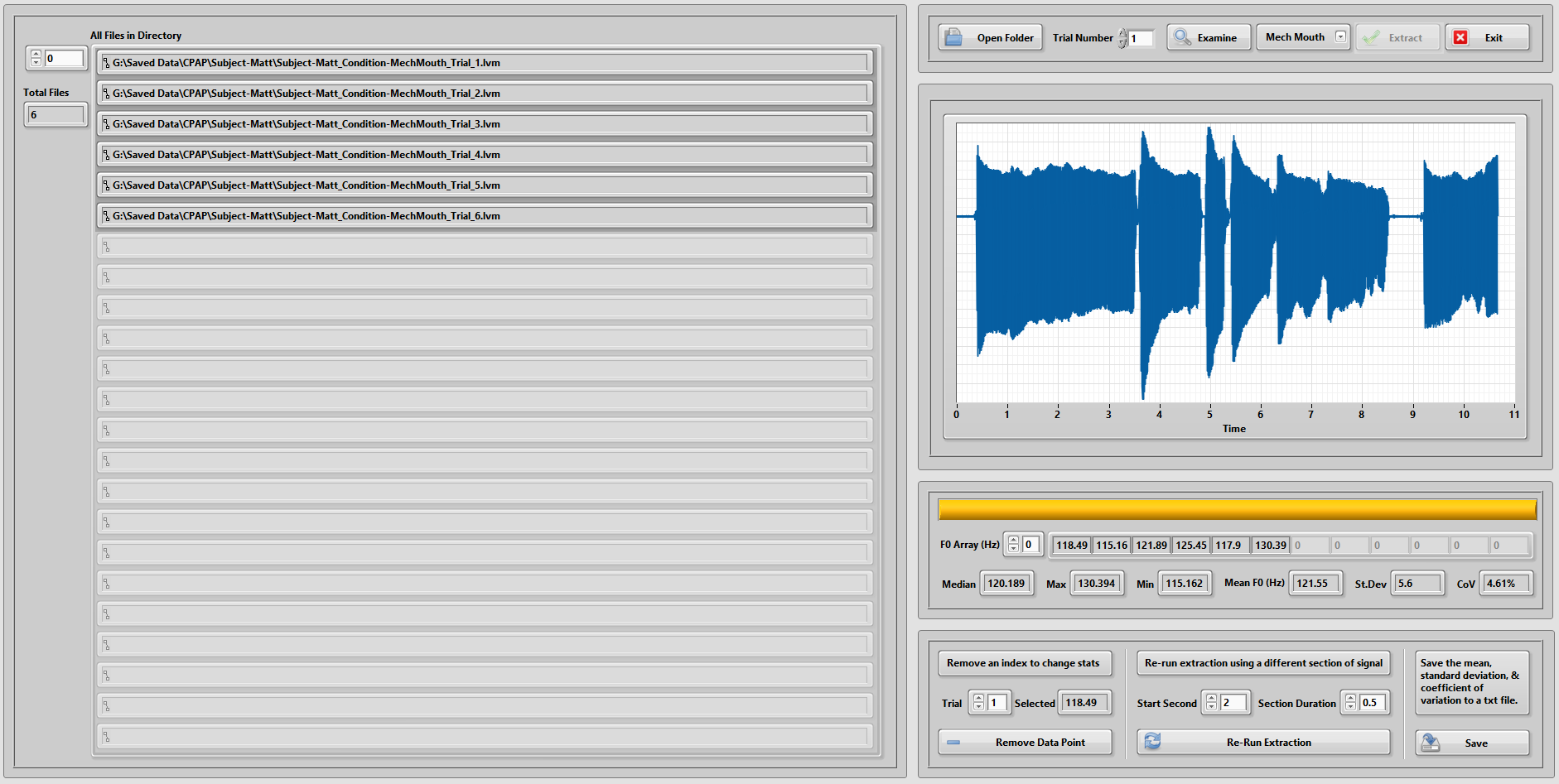
When opened from the main menu, the user will be prompted to select a folder of data. The program will then find all of the analysis results files in that folder and its subfolders and runs some basic statistics. The results of this are output in new text file with the default name of *Extracted Data-Day, Month, DD, YYYY.txt* For example: *Extracted Data-Thu, Jul 29, 2021*

Note: If and *NaN* appears in the extracted data, that means the program could not find any trials with that data. If the *NaN* is just in the standard deviation and variance columns, that means only one trial was found but stats could not be computed.

### Extract Fundamental Frequency

You should not need to use this program unless you are reanalyzing data collected before 2018. It is used to calculate the fundamental frequency on data files where F0 was not recorded, but acoustic data was recorded. Steps to complete this are below.

1. Click **Open Folder** and select a folder of LVM files.
2. Select the type of trial (e.g, mechanical mouthpiece) you need to extract data from using the dropdown menu.
3. View each trial by setting the **Trial Number** and clicking **Examine.**
4. In the bottom right, set the **Start Second** and **Section Duration** to be where there is phonation in all of the trials (2 second start and 0.5 second duration usually work).
5. Click **Extract** and wait for F0 data to be calculated for each trial.
6. If one of the trials shows an impossibly low or high F0, you will want to remove it. Do this by selecting that trial and clicking **Remove Data Point** in the section labeled “Remove an index to change stats”.
7. There is also an option to re-run the extraction.
8. Once you are satisfied with the results displayed in the **F0 Array**, click **Save.**

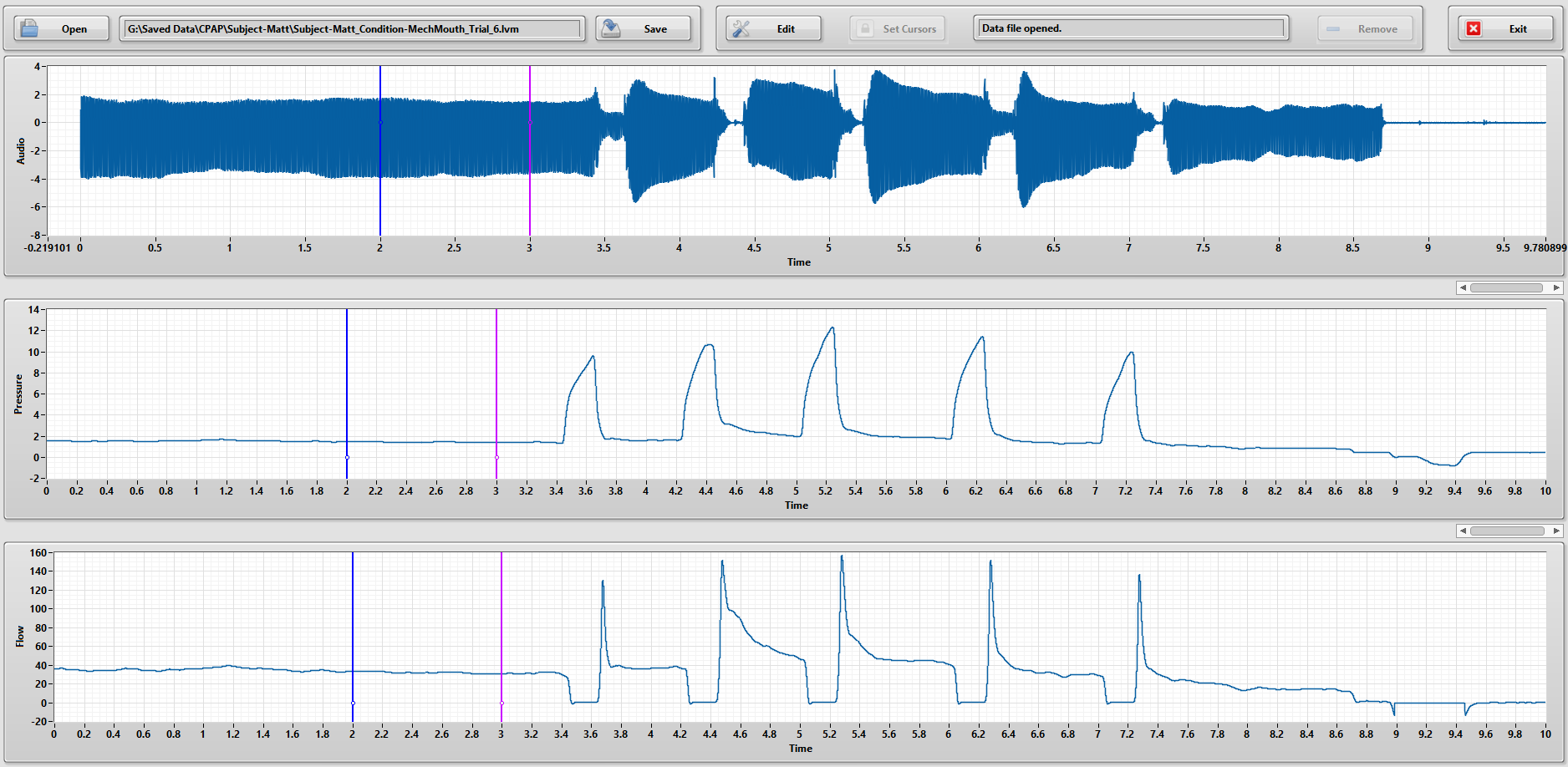


### Analyze Data

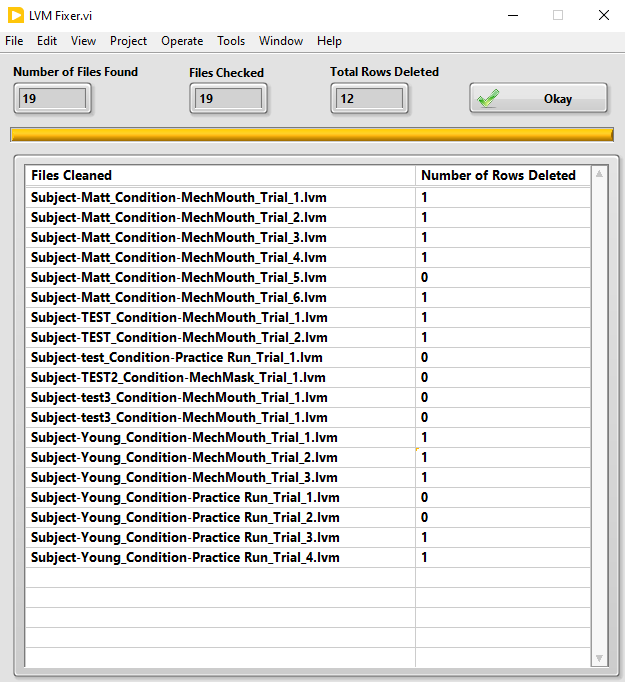
More information on the data analysis portion of this program can be found in a later section of this user guide.

### Data Clipper

This program is used to remove sections of data from an LVM file. This is primarily used for the singing helmet project.

1. Click **Open** and select an LVM file you would like to edit.
2. Click **Edit** to unlock the cursors.
3. Move the cursors to a section of data you want to remove.
4. Click **Set Cursors** to lock them in place.
5. Click **Remove** to delete the select section of data.
6. Click **Save** to save the edited data file.

### Fix LVMs

*NaNs*can appear for a myriad of reasons, but they usually are not a problem. However, they can cause problems if there are a lot of them. If an LVM data file has a lot rows that show *NaN*, use this program to delete those rows.

The user will be prompted to select a folder of data files. The program will then search through all of the LVM files in that folder and delete the rows of data that contain one or more *NaNs.*

Once completed, it will display the files that were found in the selected folder and how many rows of data were deleted from each.

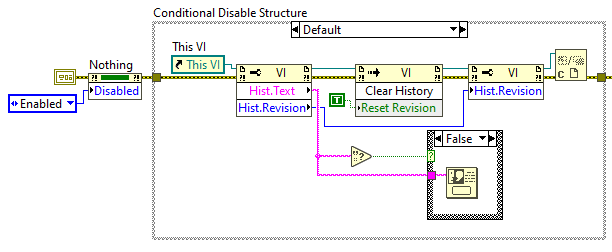
### Submit Error/Request

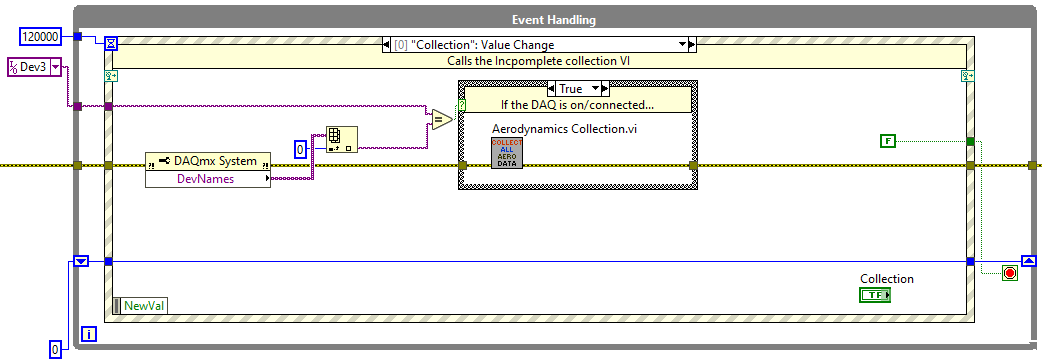
Clicking this button will open up a Google Form in your default internet browser. From there, you can report errors that you run into, request edits be made on a program, or request a completely new program. All submission on this form will notif y Austin and he will respond whenever he can.

## Block Diagram

In the following sections, we will go over the block diagrams for all of the important VIs and SubVIs found in the Master Aero Program. Details for the Master Analysis Program will be covered later in this user guide.

### Main Menu

The first thing that runs is Conditional Disable Structure. This only runs when the program is running from the LabVIEW project folder. The code in this structure displays the most recent revision notes and updates the revision number.

The primary part of the main menu code is an event structure. If you do not know what that is, you should review the first two sections of the LabVIEW binder. On the next pages, you will find descriptions of all the cases in this event structure.

#### Collection

* Checks to see if the DAQ board is connected by checking the device names found by the computer and comparing the primary name to what the default is set to (Currently Dev3).
  + **If you try running this on a different computer or with a different DAQ board, you will need to change the default device.**
* If the DAQ board is found, the Aerodynamics Collection VI will be called. If not, a warning message will be displayed to the user.

#### Randomize

* This only opens the Random Task Order VI.

#### Calibration

* First, this case checks if the user *really* wants to calibrate. This is to avoid accidentally running the calibration VI and setting everything to zero.
* Similar to the Collection case, this checks to see if the DAQ board is connected.
* If the DAQ board is connected, the Calibrate Pneumotach VI is called.

#### Calibrate Helmet

* There is the same check about *really* wanting to calibrate.
* Again, there is a check for the DAQ board connection.
* If the DAQ board is connected, the Calibrate Helmet VI is called.

#### SPL

* Opens the Gain Estimator VI if the user *really* wants to run calibration.

#### Extract Button

* The user is prompted to select a folder of data analysis files.
* Assuming file selection is not cancelled, the Extract Data VI is called.

#### Extract F0

* This just calls the Extract F0 VI.

#### Analysis

* This just calls the Analysis Master VI.

#### Clipper

* This just calls the Clipper VI.

#### Fix LVM Button

* The user is prompted to choose a folder of LVM files.
* Assuming file selection is not cancelled, the LVM Fixer VI is called.

#### Report Error or Request Change

* The Google form for reporting errors or requesting changes is opened in the default browser. This uses the Open URL in Default Browser VI. Very convenient.

#### Timeout

* This case is triggered when no user input is detected for the length of time wired to the event timeout node.
* This is currently set to two minutes (120,000 ms).

#### Application Instance Close

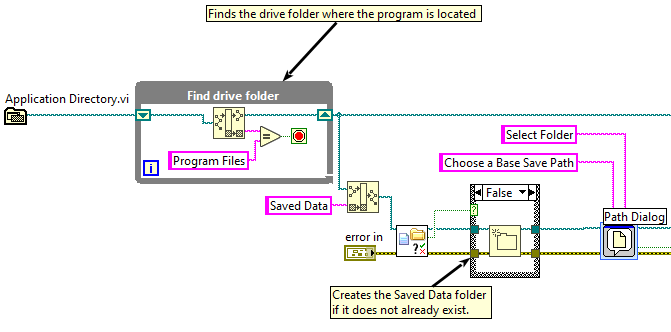
* This stops the loop when the user hits the X button and the VI is running as an executable or installed program.

#### Panel Close?

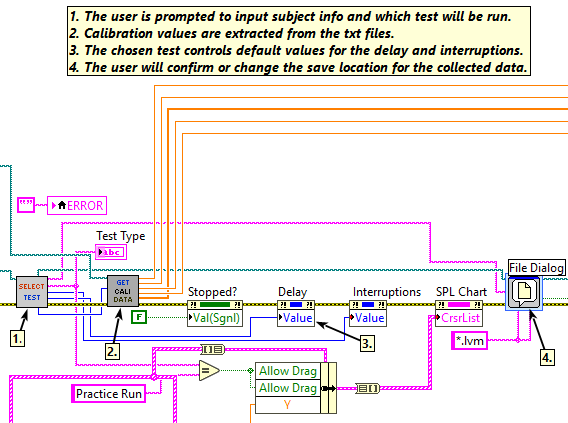
* This stops the loop if the user hits the X button and the VI is running from the LabVIEW project folder.

The final part of the main menu code is another Conditional Disable Structure. Here, LabVIEW is closed, but only when the program is running as an executable or installed program (i.e., when RUN\_TIME\_ENGINE==True).

### Aerodyanmics Collection

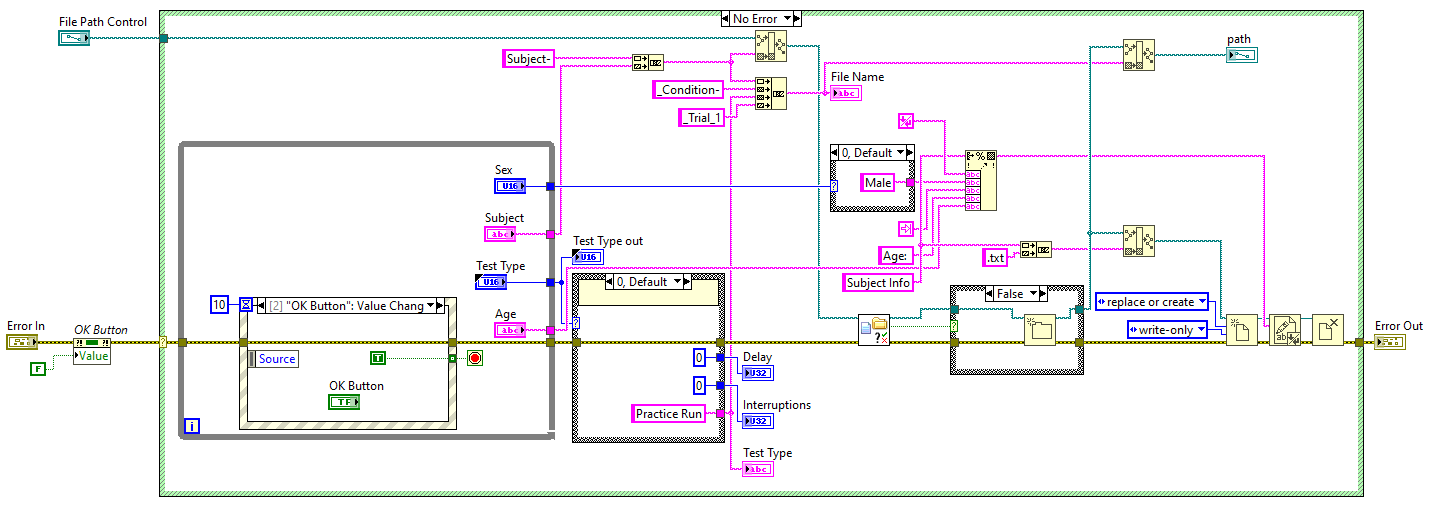
First, the program finds the drive folder where the program is located. **It must be *somewhere* in a folder named “Program Files”. If it is not, this will fail.** You could also change the folder name within the Find drive folder loop.

Next, a folder named “Saved Data” is created if it does not already exist and the user is prompted to confirm the save path using the File Dialog express VI. If “Cancel” is clicked in the file dialog, nothing else will happen and the user will be sent back to the main menu. Once the save path is confirmed, the program enters the Master Control Loop.

Once in the loop, the Test Type Selection SubVI is called. Here, the user selections determine what the default delay and number of interruptions. It also controls which pressure calibration data is output by the next SubVI, Get Calibration Data. More details on what happens in each of these SubVIs can be found in the following sections.

Again, the user is prompted to confirm the save file name and location. If this is not cancelled, the program moves on to the parallel loops portion (described after the details of the Test Type Selection and Get Calibration Data SubVIs).

#### Test Type Selection

First the error and file path are wired in from the calling VI and the OK button is set to False. Assuming there are no error, the program starts the while loop. Using the while loop with an event structure, this SubVI waits for the user to click the OK button. Once that happens, whatever the user has set in the front panel will be used to determine the default file name and default number of interruption and interruption length (delay).

Default file names are built as outlined below. Items in parenthesis indicate user input.

*Subject-(Subject)\_Condition-(Test Type from case structure)\_Trial\_1.txt*

A folder for the subject is created if one does not already exist. Subject Info text files are also saved here. They include Age and Sex of the subject.

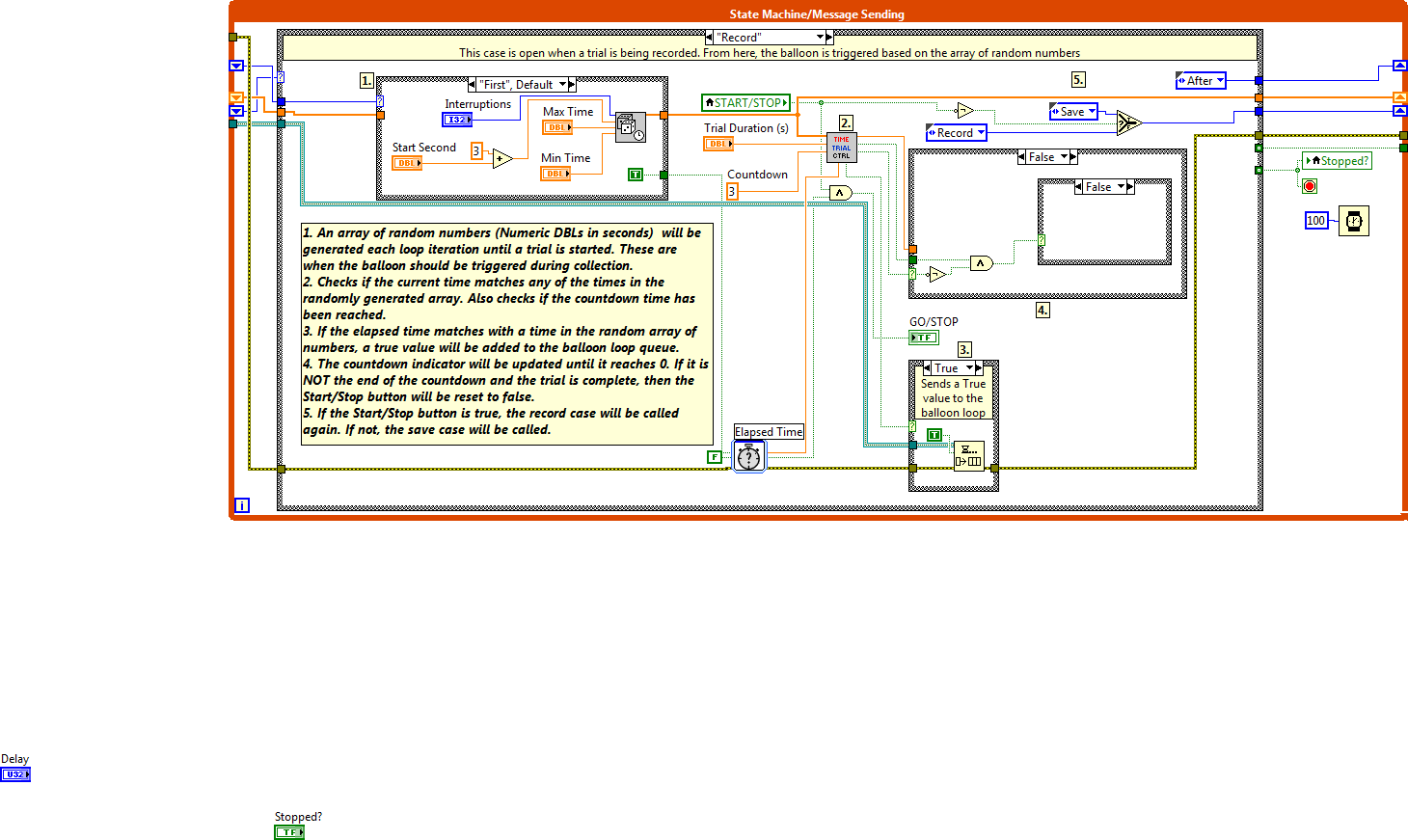
#### Get Calibration Data

This SubVI opens all of the calibration text files using the sequence of functions shown to the right. This opens a file reference for a given file path, reads the text file as a single string, then closes the reference.

#### Parallel Loops Portion

This program is organized into three parallel loops within the Master Control Loop. The design pattern used here is called a queued message handler. One loop contains a state machine that sends instructions to the other two loops based on user input.

##### State Machine/Message Senging Loop

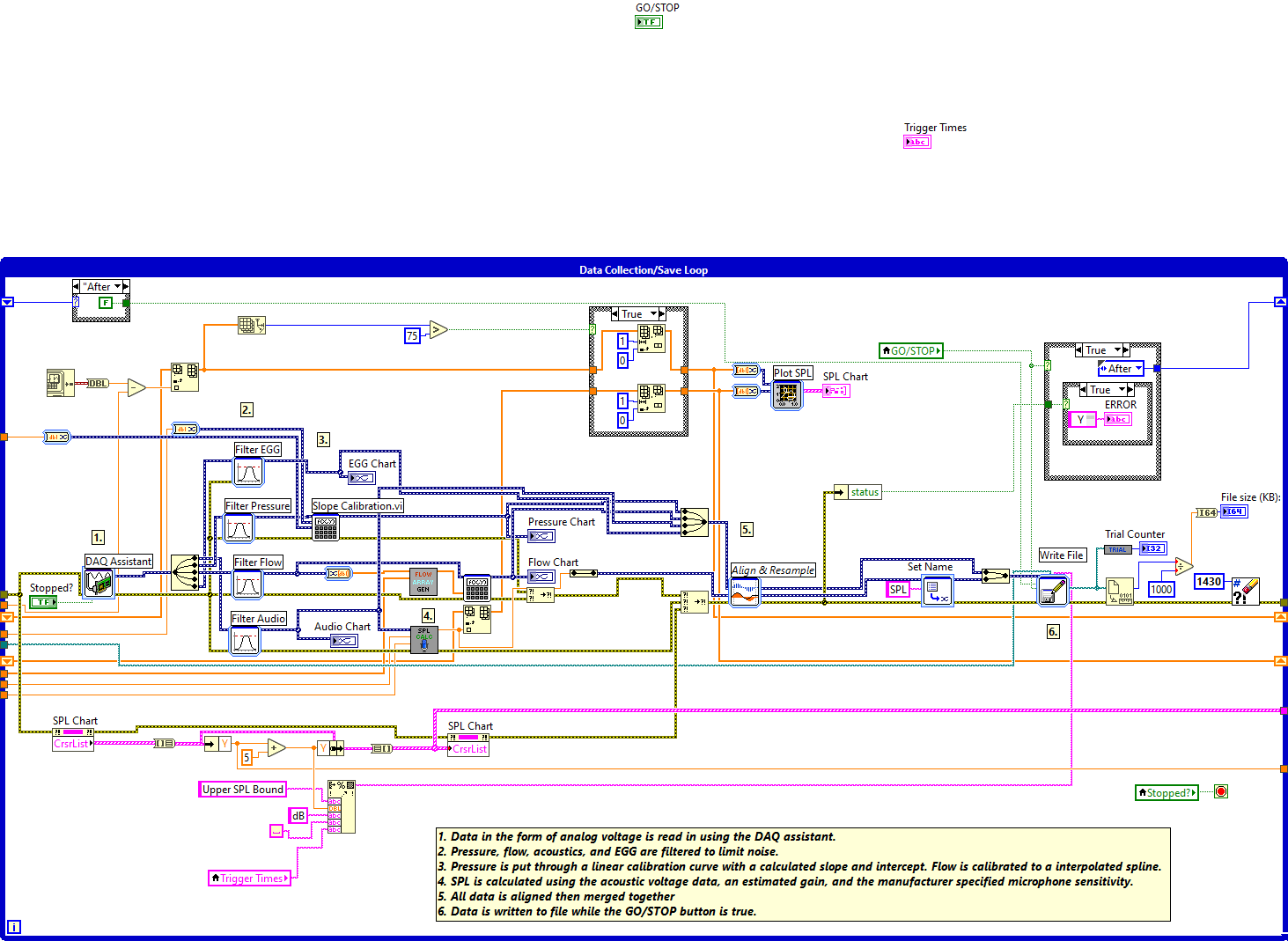
This loop is an event-controlled state machine. This controls the GO/STOP light, as well as when the balloon interruptions are triggered. A short description of each case is below.

* **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, going back to the Test Type Selection 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.

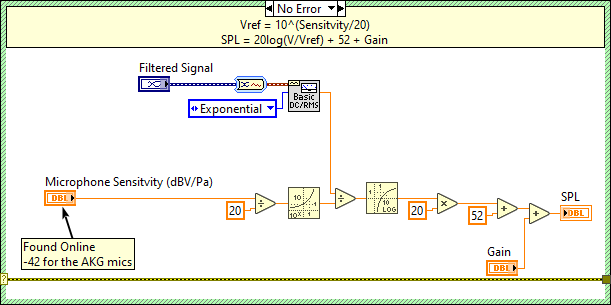
##### Data Collection/Save Loop

Here is the data collection/save loop. It reads from four channels on the DAQ board using the DAQ Assistant. It takes in acoustic, pressure, EGG, and flow data from their respective devices. Data is then filtered, and run through calibration curves to obtain actual values.

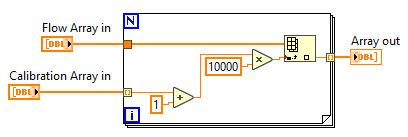
Other things to note are the pair of “First-After” case structures. These set the reset value on the Write Measurement to true only on the first iteration of the loop so a new data file is created whenever a new trial is started. Also, the x-axis of the SPL graph is controlled by building an array of values that is limited to 80 points.

On the next page, you will find details on the flow array and SPL SubVIs found in this loop.

###### SPL Calculator (2) SubVI

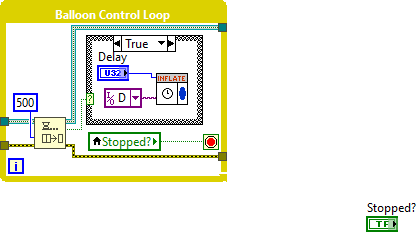
This is where SPL is calculated from the filtered acoustic signal. The equations used in this calculation are below. The gain is taken from the SPL Gain.txt file.

###### Flow Array Compare SubVI

When the pneumotach is calibrated, an array of flow values is created instead of using a slope-intercept formula. This is because the voltage-flow relationship for the sensor is not completely linear.

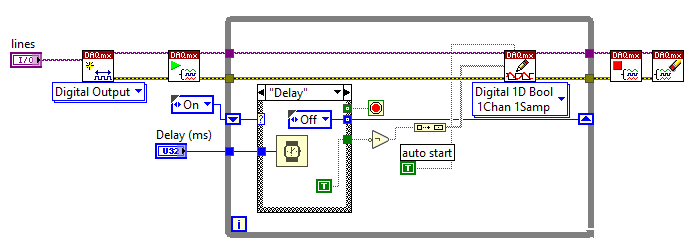
This SubVI takes in the flow voltage collected through the pneumotach applies it to the calibrated array to get calibrated flow data.

##### Balloon Control Loop

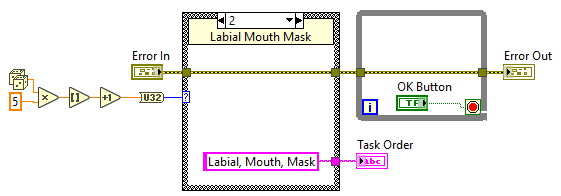
Within the **Record** state of the trial control loop, if the current trial time matches one of the times in the randomly generated array of numbers, a true value will be added to the Boolean queue. When this happens, the case structure within the balloon control loop changes to true and the balloon 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. This previously used the DAQ assistant. It now uses the formal DAQmx control structure, which is what you are supposed to use if you are not lazy.

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

### Randomize Tasks

Each of the six possible options is preloaded in the case structure. A random number from 1 to 6 is generated to pick a case and display it to the user. The while loop waits until the user clicks OK then exits the SubVI.

### Calibrate Pneumotach

### Calibrate Helmet

### Gain Estimator (SPL Calibration)

### Extract Data

### Extract F0

### Analysis Master

### Clipper

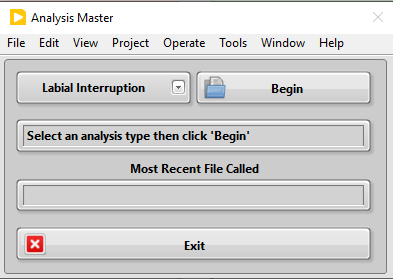
### LVM Fixer

### Timeout Check

# Master Analysis Program

## Front Panel Guide

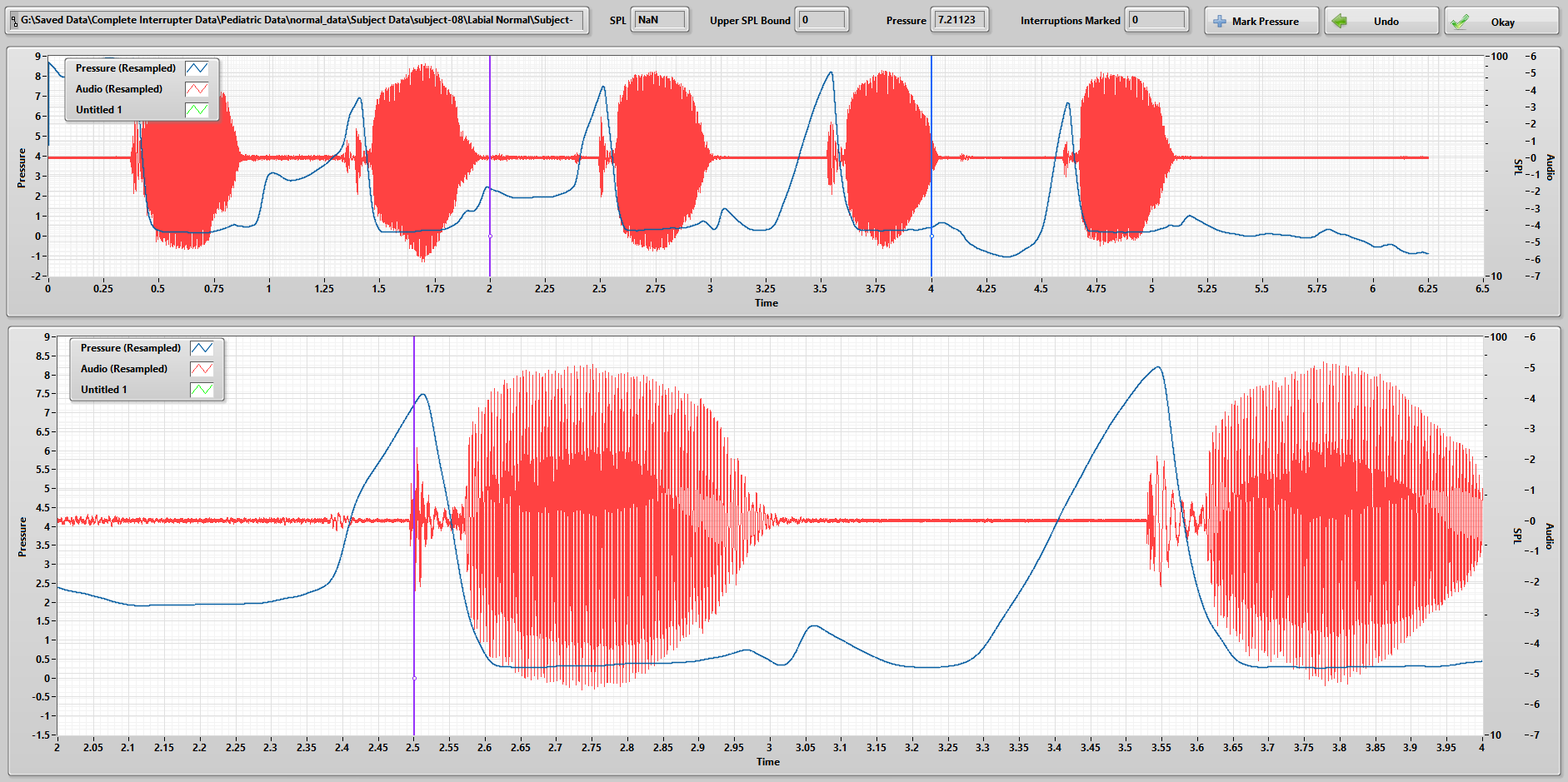
### Main Menu

The main menu is fairly simple. The user just needs to select the type of data they want to analyze and click **Begin**. No matter which test type is selected, the user will always be prompted to select an LVM file before anything else occurs.

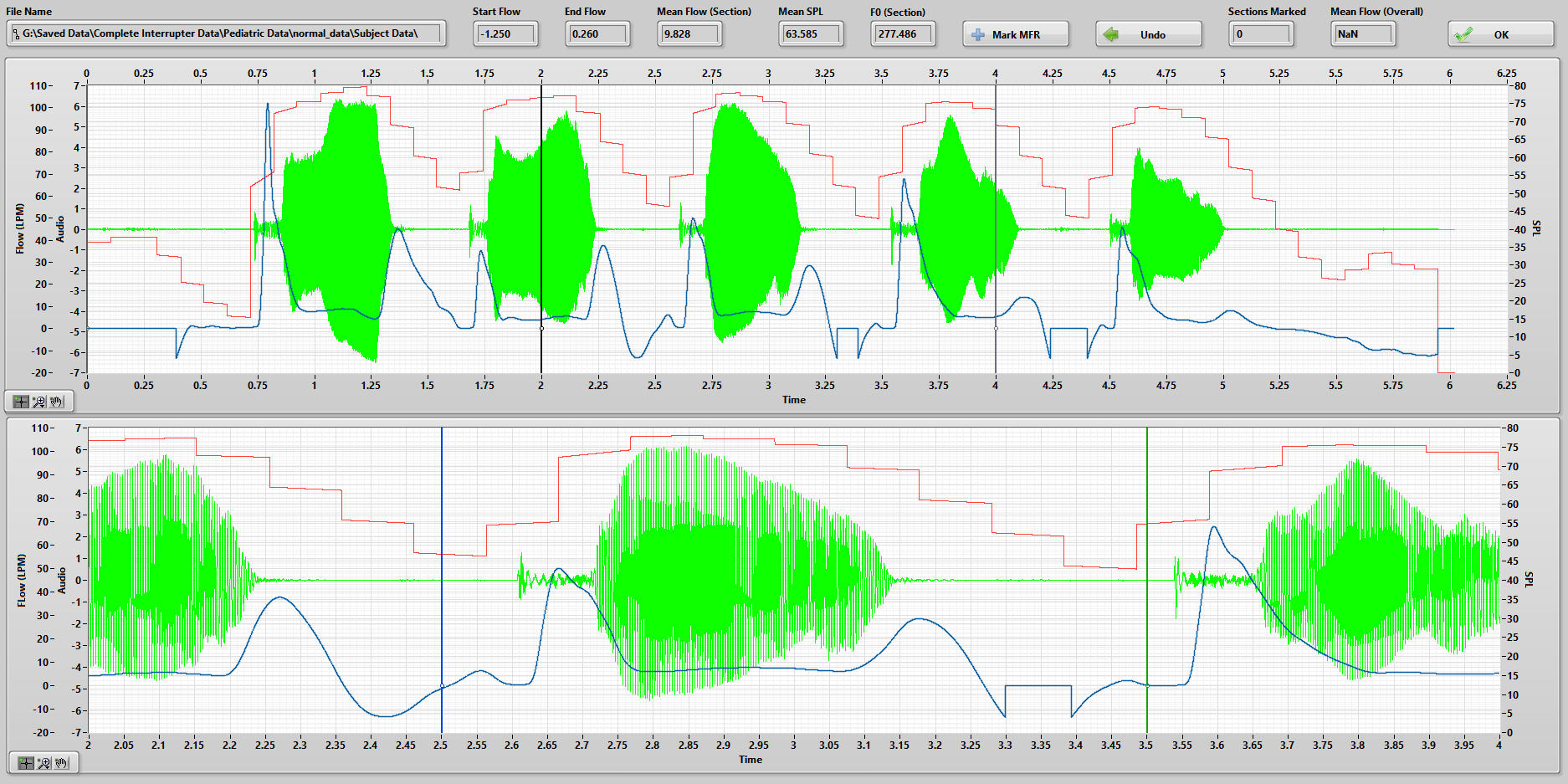
The display for most recent file called is to help the user remember where they are in analysis and avoid reanalyzing the same file multiple times.

Note: there is no analysis option for the signing helmet data. Analysis for this was done with a MATLAB program.

### Labial Interruption

The first interface that appears when analyzing labial interruptions is call Labial Pressure Marker. Here, the user moves the cursors on the upper graph to control zoom in on the each interruption, seen in the lower graph.

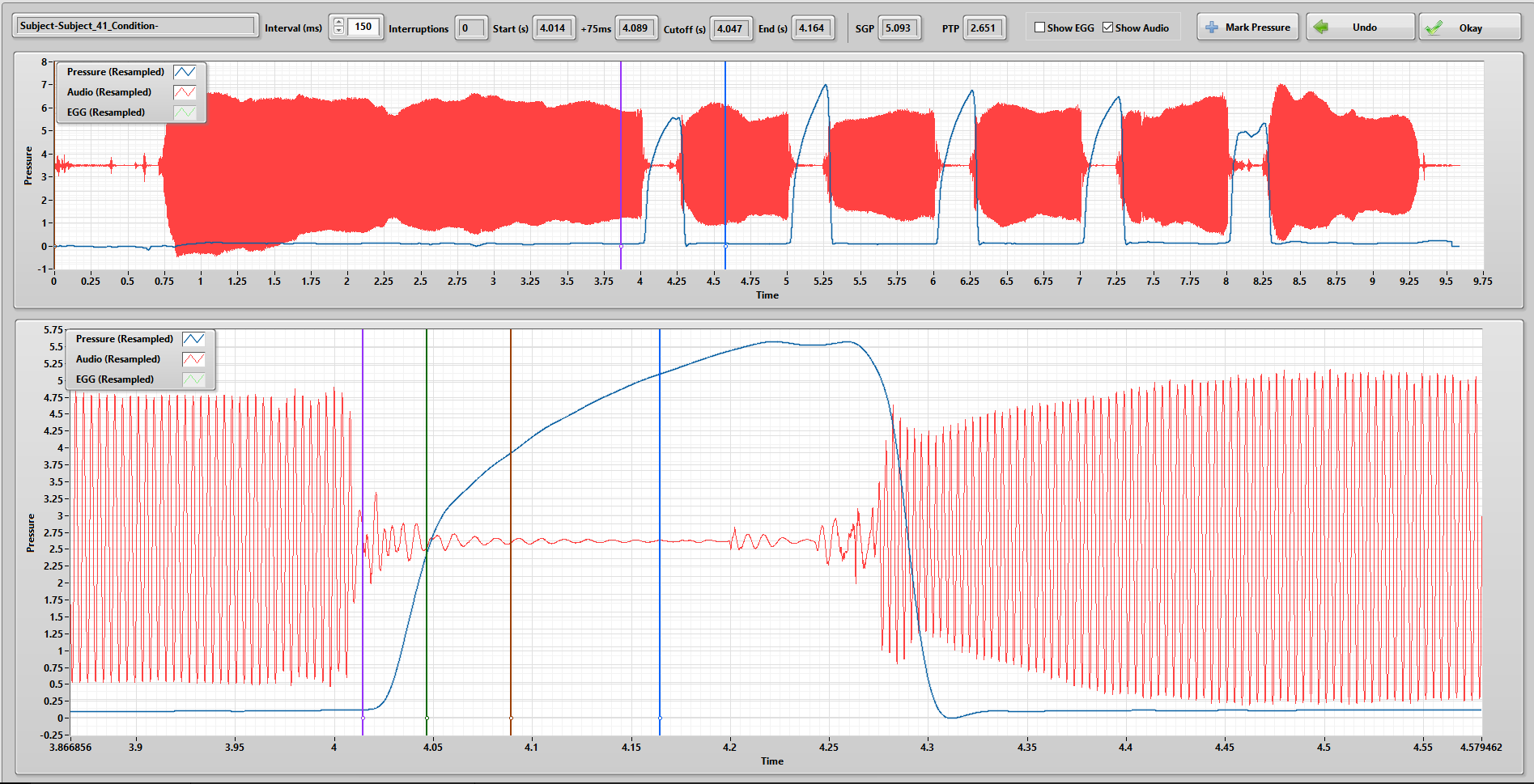
1. Move the cursor in the lower graph to the peak pressure that occurs immediately before phonation.
2. Once the peak is located, click **Mark Pressure.**
3. If you accidentally mark the wrong spot, you can click **Undo.**
4. Repeat steps 1 and 2 for all of the peaks found in the data file.
5. Click **Okay.**

The next interface is simply called MFR. Similar to the Labial Pressure Marker, the user moves the cursors on the upper graph to control zoom in on the each interruption, seen in the lower graph.

1. Move the cursors in the lower graph to a section where there is phonation.
2. Click **Mark MFR.**
3. If you accidentally mark the wrong spot, you can click **Undo**.
4. Repeat steps 1 and 2 for at least 3 phonation segments.
5. Click **OK**.

You can now save these data (MFR, SGP, SPL, and F0) to a new or existing text file.

### Mechanical Interruption

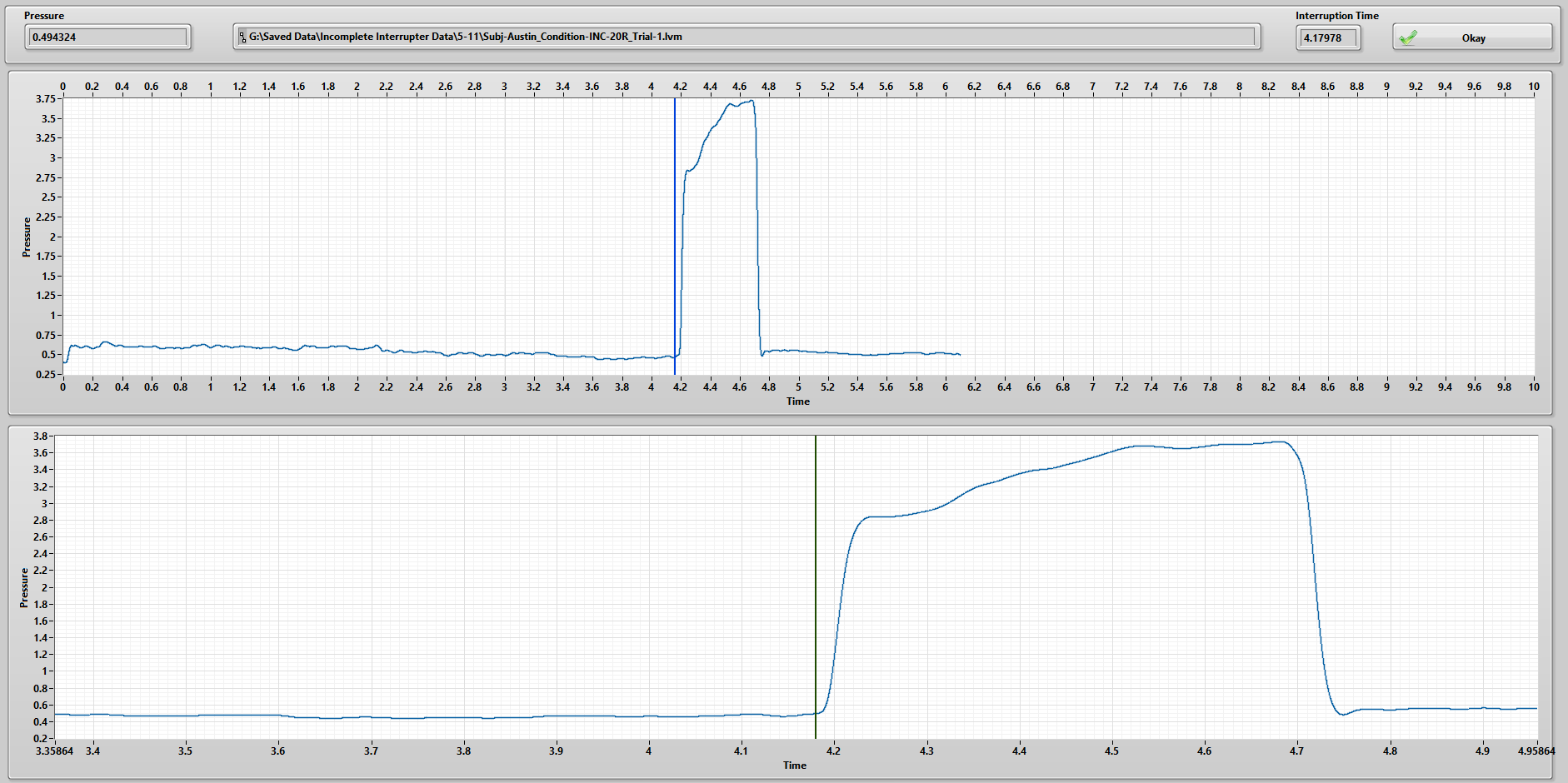
The first interface for analyzing mechanical interruptions is called Find SGP and PTP.

1. Move the cursors on the upper graph to zoom in on an interruption.
2. In the bottom graph, move the farthest left cursor (purple) to the spot right before where the pressure trace starts increasing
   1. The brown and blue cursors move with the purple one.
   2. The brown cursor shows is 75 ms after the purple one.
   3. The blue cursor is after the purple at whatever the interval is set to (this is usually 150 ms).
3. Use the check boxes in the top bar to show acoustic and/or EGG data.
4. Using either the acoustic or EGG data, move the green cursor to where phonation is cut off (this can be tricky, refer to JJ’s analysis guide in the Aero Protocols binder)
5. Click **Mark Pressure**.
6. Repeat steps 2 through 4 for all the interruptions in the data file.

The next interface is the MFR analysis program. This is the same one used when analyzing labial interruptions. Please refer to that section.

You can now save these data (MFR, SGP, PTP, SPL, and F0) to a new or existing text file.

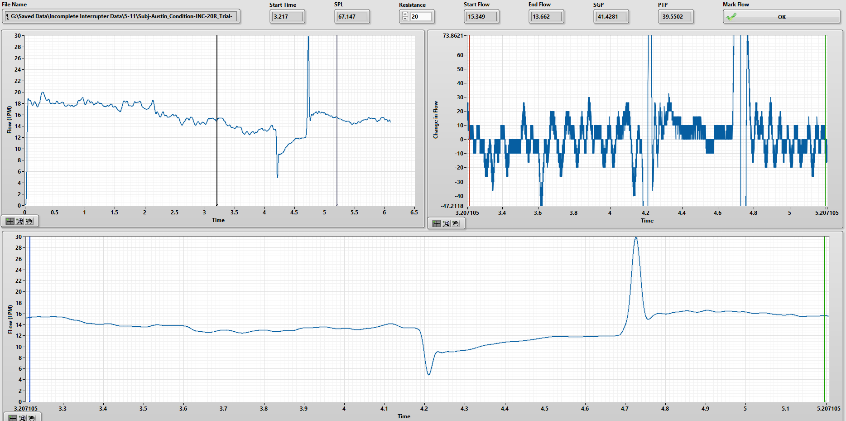
### Incomplete Interruption

This program was originally written back in 2016 and has not seen many updates since. Its purpose is to analyze a single long (500 ms) interruption. During the interruption, the subject will be phonating through a set resistance instead of completely blocking airflow. The user will be prompted to choose PTP or Not PTP.

If Not PTP is chosen, the above will appear. Here the user should

1. Move the cursor on the upper graph to change the bounds on the lower graph.
2. Use the lower cursor to mark where the interruption begins.
3. Click **Okay** when ready to move on.

In the following interface,

1. Set the resistance to be whatever was used during collection.
2. Use the cursors in the upper left graph to zoom in on the interruption.
3. Use the cursors in the upper right graph to mark a section of minimal changes in airflow.
4. Use the blue cursor in the bottom graph to mark a spot of steady flow just before the interruption and click **OK.**

If PTP is chosen at the beginning, the first interface will be similar to marking cutoffs for mechanical interruptions.

### Airflow Redirector

In this program, the user simply needs to move the cursors to mark where the max pressure of the trial is. This will be where the pressure in the tank equilibrates with subglottal pressure (read the redirector paper for more information).

After the pressure is marked, the MFR program will come up again. See the labial interruption analysis section for more details.

## Block Diagram

In the following sections, we will go over the block diagrams for all of the important VIs and SubVIs found in the Master Analysis Program.

### Main Menu

#### Initialize

#### Wait to Start

#### Analyze

#### Save