Lab experience 6: Lung Sound Analysis with LabVIEW

Objectives

- ➢ Gain experience with audio signal analysis in LabVIEW
- ➢ Learn to create a subVI that simulates real-time audio signal
- ➢ Get started with lung auscultation

Recommended Resources

- Spectral Measurements Express VI (Essick 4th Edition Ch 12.1)
- UW Medicine pulmonary skills module https://depts.washington.edu/physdx/pulmonary/terms.html
- Littmann™ Learning App.

New concepts for this week

1. Stethoscope

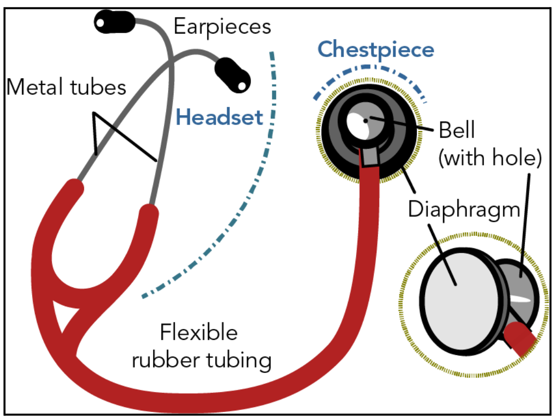


Figure 1. Anatomy of a stethoscope. Source: Wikimedia Commons

The chest piece (sometimes called the head) of the stethoscope is two-sided. The diaphragm side is better for listening to high-frequency sounds, and the bell is better for listening to lower frequencies (e.g., heart murmurs). You can switch the side by rotating its fitting to the rubber tubing.

2. Lung auscultation

Lung auscultation typically starts from tracheal/bronchial areas, and progresses downward in a stepladder fashion all the way down to the diaphragm. Figure 2 describes the characteristics of healthy bronchial, bronchovesicular, and vesicular (originating from the lobes) breath sounds. Physicians typically pay attention to three characteristics when auscultating:

1. Ratio of inhalation duration to exhalation duration.
2. Asymmetry in the intensity between left and right lungs.
3. Any *adventitious* (i.e., abnormal) lung sounds.

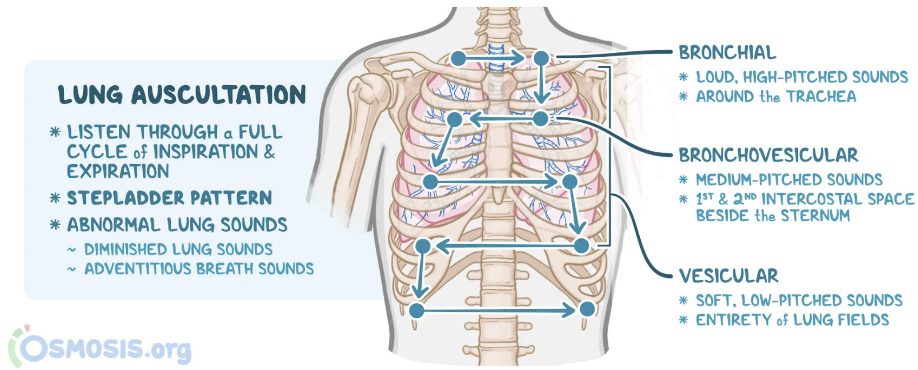


Figure 2. Sites of lung auscultation. Source: Osmosis.org

Some adventitious lung sounds to know:

Table 1. Adventitious lung sounds. Source: Littmann™ Learning

Adventitious lung sound	**Sound quality**	**Respiratory phase**	**Typical auscultation site**
Wheeze	Loud, high-pitched, musical	Usually expiratory	Trachea and chest
Rhonchi	Coarse, low-pitched, musical	Usually expiratory	chest
Fine crackle	Quite, high-pitched, popping	Inspiratory	chest
Coarse crackle	Loud, low-pitched, popping	Usually inspiratory	chest

EasyAuscultation.com (https://www.easyauscultation.com/lung-sounds) provides more in-depth explanations on adventitious lung sounds.

Pre-lab calculations

There is no pre-lab calculation this week.

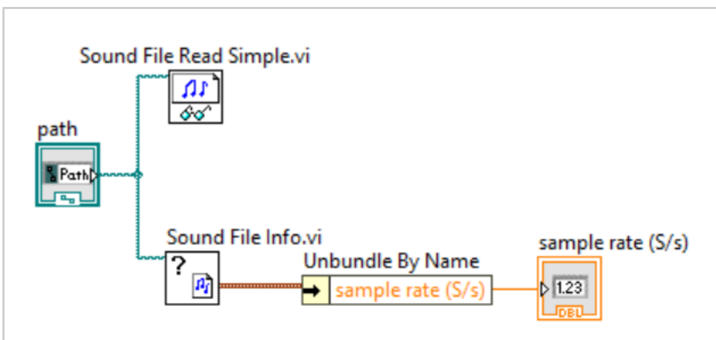
Pre-lab programming

Simulating real-time signals is an essential skill in real-time signal processing. In this pre-lab activity, we will create a subVI that crops the audio in a .wav file into one-second-long segments and outputs them in succession as if it was acquired in real time. Each step below comes with an accompanying figure showing the resulting block diagram.

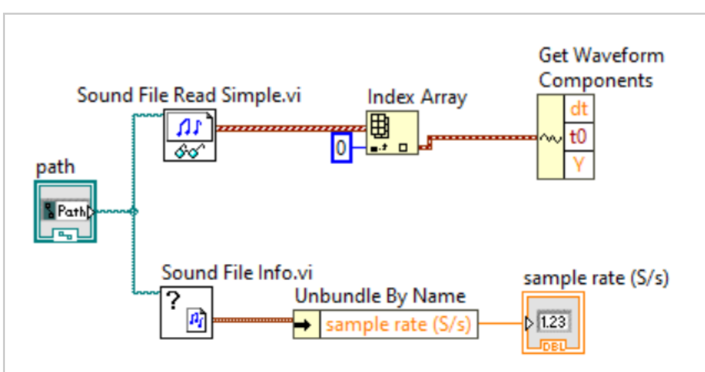
1. Download and extract `lung_sounds.zip` on the assignment page. The dataset is a subset of the Respiratory Sound Database collected by Rocha et al., 2018 (paper). The Kaggle website (link) provides an overview of the data set. Briefly, it contains .wav files of breath sound recordings, each of which comes with an expert-annotated text file (also provided on the assignment page for your reference) that contains four columns:

- Beginning time of respiratory cycle (s)
- End of respiratory cycle (s)
- Are crackles present? (yes=1, no=0)
- Are wheezes present? (yes=1, no=0)

2. Let's create the lung sound simulator subVI. Create a new VI. Open the function palette, go to Graphics & Sounds > Sound > Files. Drag and drop the **Simple Read** function to the block diagram. Repeat this step for the **Info** function. Create a control for the **Path** terminals of both functions. Drop an **Unbundle By Name** function next to the **Info** function, then wire the **sound format** terminal of **Info** to **Unbundle By Name.** This will allow you to extract the sampling rate information of the file.



3. Now we want to turn the waveform data type into a 1D array so that we can manipulate it. To do that, use the **Index Array** function to extract the first channel of the waveform. Then, in the function palette, go to Waveform, and drag and drop a **Get Wfm Comps** (get waveform components) function. Expand it vertically to get two more attributes. Click on the attributes and set them to be "dt" and "t0". Pass the output of **Index Array** to the input of **Get Wfm Comps**.



4. Let's extract a one-second segment from the waveform. Wire the **Y** channel of **Get Wfm Comps** to an **Array Subset** function. Here comes the question: what will the start **index** and **length** of the **Array Subset** be?

- **Length**: is the number of samples of the waveform segment, and it can be calculated as follows:

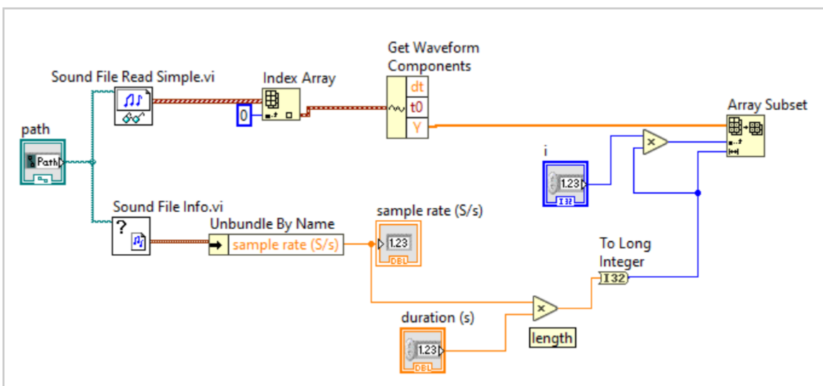
  $$length = f_s \times T$$

  where f_s is the sampling rate, and T is the sampling duration, which is 1 second by default. That is, you also need to make a numeric control named "duration" and set its default value to 1 second. Remember to convert *length* into a 32-bit integer (in Numeric > Conversion) before passing it to **Array Subset**.

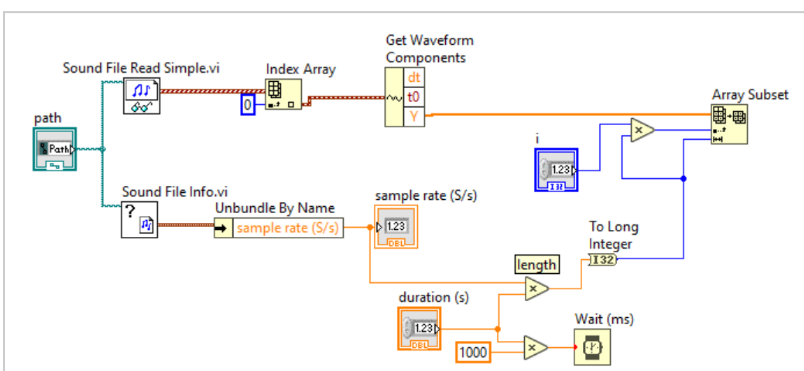
- (Start) **index**: Let's allow the user to pass an external counter **i** to this subVI, and this subVI should return the **i**-th waveform segment. This means that the start **index** can be calculated as follows:

  $$index = i \times f_s \times T = i \times length$$

  Pass the resulting *index* to **Array Subset**.

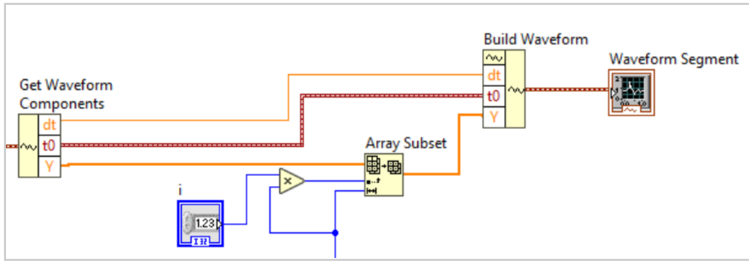


5. Add a **Wait (ms)** timer, and set the wait time to the duration (multiplied by 1000 because it counts in ms). This pauses the subVI to simulate data acquisition.



6. Now we will convert the waveform segment back to Waveform data type. In the function palette, go to Waveform, and select **Build Waveform**. Expand it vertically to

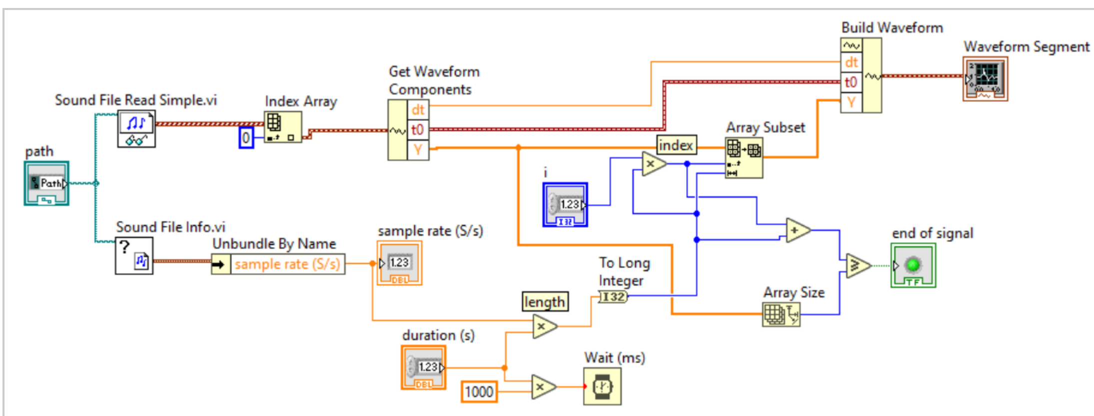
add two more attributes: “dt” and “t0”. Then, wire the “dt” and “t0” from **Get Wfm Comps** to Build Waveforms. For “Y”, wire the output of the **Array Subset**. Plot the result on a Waveform Graph. Rename the Waveform Graph as **Waveform Segment**.



7. To prevent the user from indexing out of range, we can provide a boolean indicator that becomes true whenever **i** is greater or equal to the last segment. In order words, write code to check if the following inequality is true:

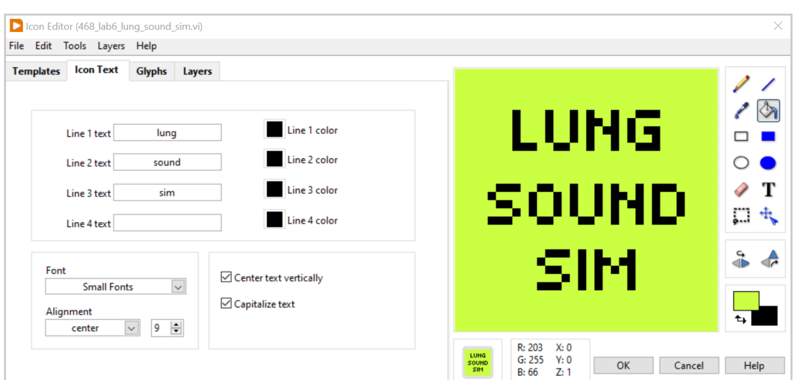
$$index + length \geq N$$

where N is the total length of the waveform. Name the indicator “**end of signal**”.

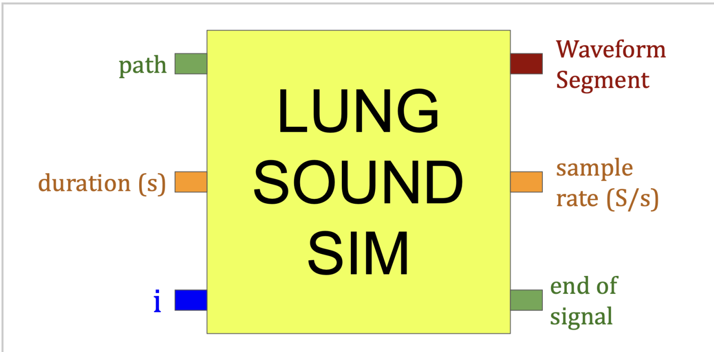


Note: You could instead reset the index back to 0 when the index is out of range.

8. Design the icon of the subVI. Double-click on the VI’s icon on the top right. This will open the Icon Editor. Go to Layers and delete the current pattern. Create a pattern of your choice. An example icon is shown below:



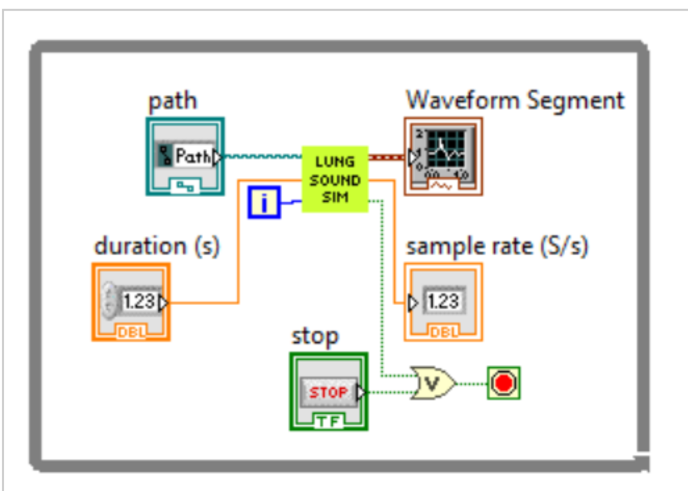
9. Set up the terminals of the subVI. Right-click on the terminal diagram on the top right corner of the front panel. Select Patterns. Then choose a pattern with at least 3 terminals on each side. Wire the terminals by clicking on the terminal first, and click on the items in the front panel. On the left, the subVI should have 3 terminals: **path**, **duration**, and **i**. On the right, it should also have 3 terminals: **Waveform Segment**, **sample rate**, and **end of signal**. If you are unsure how to do this, check this tutorial on how to create a subVI: https://youtu.be/zr2qqv1F4kk?t=140.



10. Save the subVI and give it a descriptive name.

11. Now let's use this simulator subVI to do some "real-time" audio signal analysis. Open a new VI. Create a **While loop**. In the function palette, go to Select a VI…, then select the simulator SubVI you just created, and put it inside the **While loop**.

12. It is time to do some wiring for our simulator subVI! Wire the loop index **i** to the simulator's **i** terminal. Create controls for **path** and **duration**. Create indicators for **Waveform Segment** and **sample rate**. Lastly, write code such that the loop will terminate either if the button is true or if the **end of signal** boolean is true.

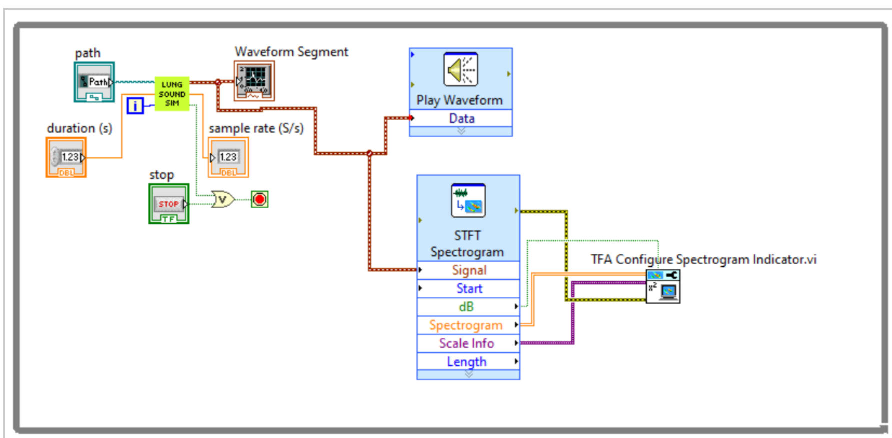


13. Select a .wav file and run the VI to confirm that the VI is working properly. You will see that for every one second, the graph is updated to the next waveform segment, and this continues until the signal ends.

14. (Optional) Let's play the waveform with the speaker. In the function palette, go to Graphics and Sound > Sound > Output, and select **Play Waveform**. Wire the waveform segment to **Play Waveform** and now you should hear the one-second segments.

15. Let's plot the spectrogram of the waveform segment. In the function palette, go to Signal Processing > Time Frequency Analysis > Spectrogram, and select **Time Frequency Spectrogram**. When configuring the spectrogram, make sure to select "dB" and "Auto Z scale". Also, consider decreasing the time steps as this will make the spectrogram look smoother.

16. Plotting the spectrogram requires another configuration function. In the function palette, go to Signal Processing > Time Frequency Analysis > Utilities, and select **Configure Spectrogram Indicator**. You should make four connections between this function and the Spectrogram: (1) dB, (2) Spectrogram, (3) Scale Info, and (4) Error in/out.

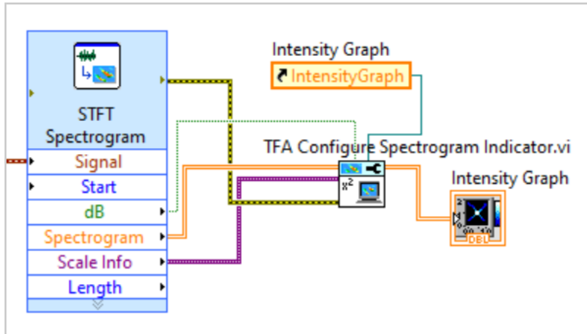


17. Create an **Intensity Graph** in the front panel. In the block diagram, wire the Spectrogram from the **Configure Spectrogram Indicator** to the **Intensity Graph**. Right-click on the **Intensity Graph**, and right-click on the Intensity Graph > Create > Reference. Then connect this reference to the top of the **Configure Spectrogram Indicator**. Passing the reference of the graph gives the configuration function access to the graph's properties without the need of explicitly exchanging those setting information over wires. Connect the **Intensity Graph** to the **Spectrogram** output terminal of the **Configure Spectrogram Indicator**. The amplitude of the spectrogram is going to be pretty small. Remember to turn ON the Autoscale of Amplitude (Z-axis) of the **Intensity Graph**.

Tip1: You can increase the acquisition duration and adjust the Y-axis range for better visualization.

Tip2: If the **Intensity Graph** shows nothing, turn ON auto-scaling for all three axes.

Tip3: Use a tone generator app, verify your spectrogram is working properly.



The first few steps of the in-lab activities are better done with a teammate, but they can actually be done at home. So please feel free to give yourself a head start.

In-lab activities

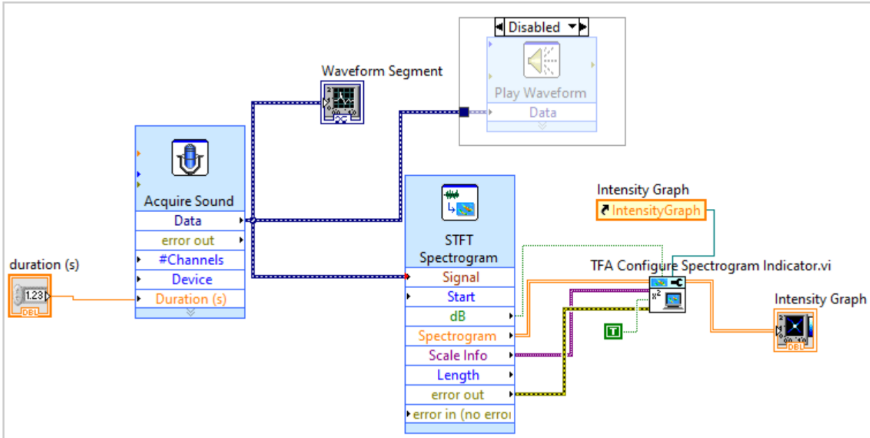
When auscultating, be aware of frictional noises caused by clothing. A single layer of clothing should be fine.

1. Use your pre-lab VIs to look at the spectrograms of different patients. Take a screenshot of the spectrograms of the last three patients (i.e., 166, 104, and 206), and note where there is evidence of the adventitious sounds, e.g., "*over the 3 second window, crackles can be seen as the high frequency components appearing at around 0.8-1.2 seconds.*"

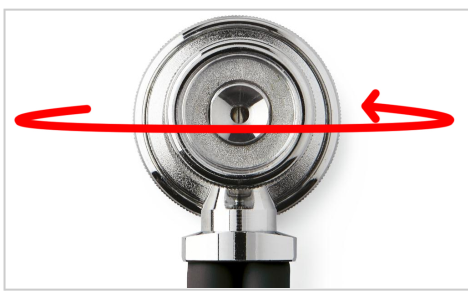
Table 2. Patient information from Respiratory Sound Database.

Patient number	Gender	Age	Diagnosis	Auscultation site	Adventitious sounds
224	F	10	Healthy	Anterior left	None
121	F	13	Healthy	Trachea	None
206	M	3	Bronchiolitis	Lateral right	Wheezes
166	M	71	COPD	Lateral left	Crackles
104	F	70	COPD	Anterior right	Dry coughs

3. Make a copy of the current VI. Replace the lung sound simulator with an actual microphone input. In the function palette, go to Graphics & Sound > Sound > Input, and select **Acquire Sound**, which only has to make two connections: **Duration** and **Waveform Segment**. Delete the rest of the broken wires.



4. Obtain a stethoscope that is connected to a microphone (via 3/16”, 1/4” and 3/8” I.D. clear vinyl tubing). Plug it into the computer, and configure the input device of **Acquire Sound** to be from the stethoscope (usually shows up as “headphone …” in the dropdown menu). Use alcohol wipes to clean the chest piece and ear tips before use. The stethoscope is dual-headed, so remember to rotate the tube fitting such that the head is listening from the diaphragm side, that is, the hole should not be exposed on the bell side.

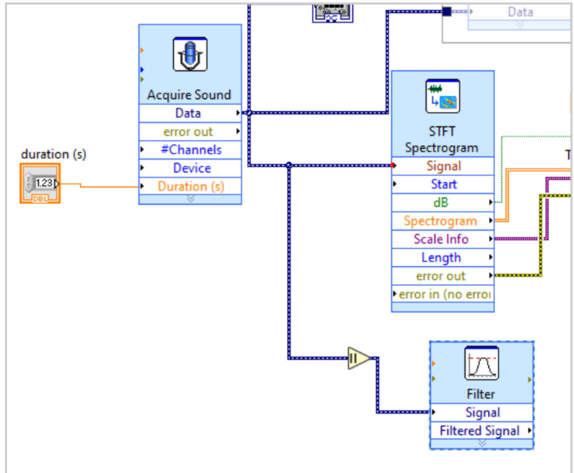


(Figure source: ScrubSmart)

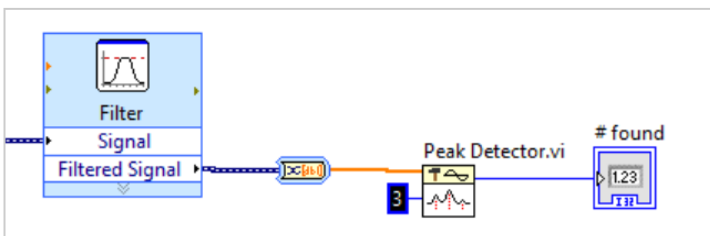
Tip: If the stethoscope is missing from the dropdown menu, quit and reopen the VI.

5. The student who is being auscultated, either by another classmate or self, should sit still and take deep breaths with an open mouth. Place the stethoscope near the windpipe until the VI detects the breath sound.

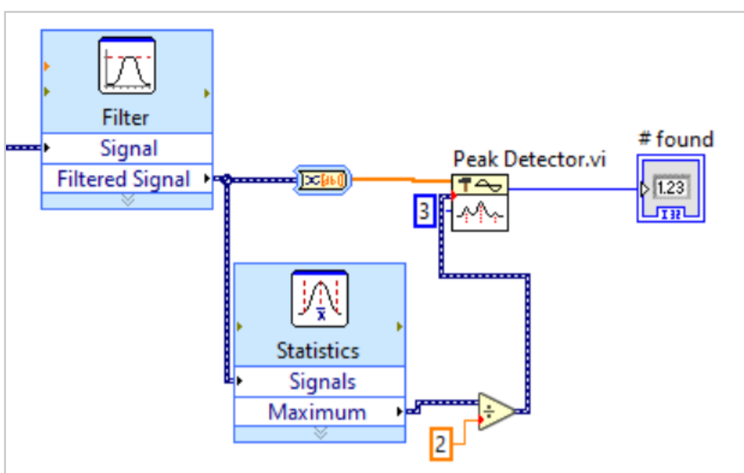
6. Now let’s write code to measure the respiratory rate. The first thing we notice is that the louder the sound, the larger the envelope of the signal. This feature is particularly evident in tracheal breath sounds. We can take advantage of this feature to detect inhalation and exhalation. To extract the magnitude of the envelope, first take the absolute value of the signal, then pass the signal to a filter (go to Express > Signal Analysis > **Filter**). Make it a low-pass Butterworth filter with a cutoff frequency of 1 Hz and an order of 3.

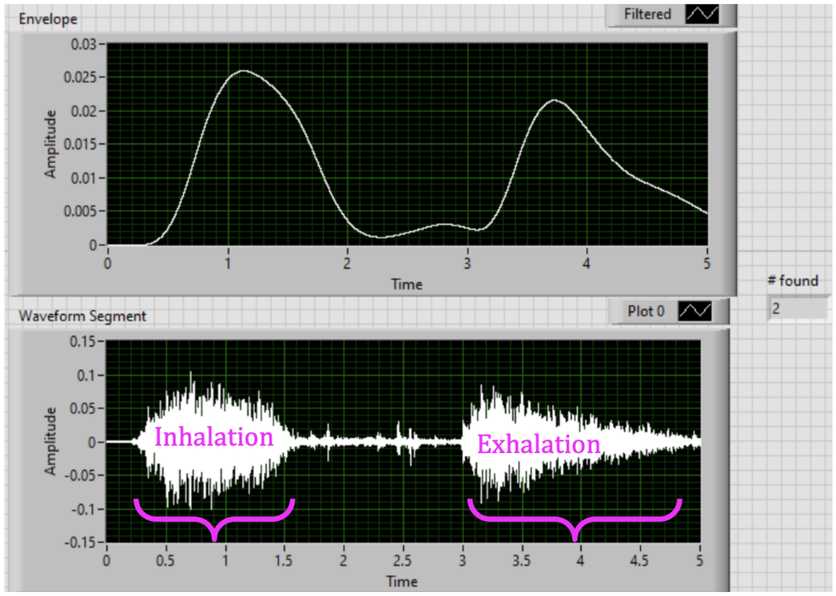


7. Pass the filtered signal into a **Peak Detector** (go to Signal Processing > Sig Operation). A datatype convertor will pop up to convert the dynamic data to an array. Double-click on the convertor and select 'Single Channel'. Also, set the width of the **Peak Detector** to 3 seconds, and create an indicator for **#found**, which tells us how many peaks there are.



8. But what threshold should we set for the **Peak Detector**? How about ½ of the current maximum? Go to Mathematics > Prob & Stat, and get the **Statistics** function. Configure it to output the maximum, and divide it by 2. Let this be our threshold.





9. Let's take some respiration rate measurements. Use a metronome app (or just google "metronome") to time your breathing and record the observations in the table below:

Table 3. Respiration rate measurements.

Respiration rate timed by a metronome (bpm)	**Respiration rate measured by VI (bpm)**	**% Error** $\frac{\lvert metronome - VI \rvert}{metronome} \times 100\%$
12		
24		

Tip: Set the metronome to be twice as fast as the respiration rate. Then, inhale at one tick and exhale at another.

Note: If the above breathing rates are making you feel uncomfortable, feel free to just record a few respiration rates that you are used to. If you are experiencing any discomfort, stop the experiment immediately and let an instructor know.

10. (Optional) Look for any adventitious sounds in your own breath sounds.

11. (Optional) Download the Respiratory Sound Database (link) to analyze more patients.

12. (Final Project Option) Build a cough detector.

Deliverables

- Your lung sound simulation subVI.
- Your in-lab lung sound analysis VI.
- A short write-up containing the following elements:
    - Your name(s)
    - The screenshots of the spectrograms of Patients 166, 104, and 206. Include a caption for each screenshot noting where there is evidence of adventitious lung sounds. Feel free to annotate the screenshots if that makes captioning easier.
    - Table 3 (Respiration rate measurements)

Note: VIs and write-ups may be co-authored by up to three students. Documentation of the VIs is optional. You will be graded on completeness.