Lab experience 8: QRS Detection with Discrete Wavelet Transform

Objectives

- ➢ Gain experience with single-lead electrocardiogram (ECG)
- ➢ Understand the principles behind wavelet analysis for ECG signal processing.
- ➢ Learn about the limitations in Discrete Wavelet Transform (DWT).

Lab Safety

If you have a history of cardiac disease, please inform the instructors before starting the lab. While you are not required to obtain the ECG signal from your own body, you are welcome to use ECG data from another student. If you experience any discomfort while obtaining the ECG signal, please immediately disconnect the electrode and inform the instructors. By attending the lab session, you agree to follow these safety rules.

Optional Readings

- Crone B. Mitigation Strategies for ECG Design Challenges. Analog Devices. 2011. Retrieved from: https://www.analog.com/en/technical-articles/mitigation-strategies-for-ecg-design-challenges.html
- AD8232 Datasheet: https://cdn.sparkfun.com/datasheets/Sensors/Biometric/AD8232.pdf
- Addison PS. Wavelet transforms and the ECG: a review. Physiol Meas. 2005 Oct;26(5):R155-99. doi: 10.1088/0967-3334/26/5/R01. 2005 Aug 8. PMID: 16088052.

Concepts

1. Right Leg Drive (RLD) in AD8232

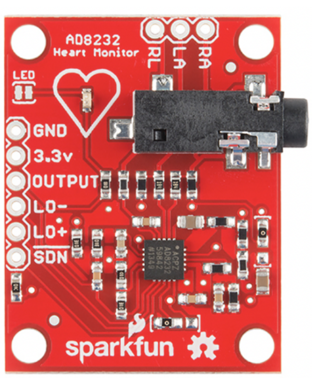


Figure 1. AD8232 Signal Conditioning board. The heart icon's LED will flicker and dim in response to the detected heartbeat.

The AD8232 is an affordable and user-friendly breakout board that is designed to measure the electrical activity of the heart. It has three input electrodes: the right arm (RA), left arm (LA), and right leg (RL). The AD8232 measures the voltage difference between the RA and LA electrodes to generate the ECG signal that is equivalent to Lead I in conventional ECG.

You may be wondering what the purpose of the RL electrode is. The answer lies in common-mode voltages, which are interfering voltages that can come from the mains or radio frequencies. These voltages are called "common mode" because they are present in both the RA and LA electrodes. Ideally, our (differential) amplifiers would be able to fully reject these voltages. However, in reality, they can manifest as baseline drifts and other noises in the ECG signal. We must take action to reduce their effects.

One technique for reducing the effects of common-mode voltages is to inject a small current that is 180 degrees out of phase with the common-mode voltage via the RL electrode. This technique is known as right leg drive (RLD), and it is what the red electrode is used for. By injecting a small current from the right leg, the AD8232 can help cancel out any common-mode voltage variations that may be present in the ECG signal.

2. Wavelet analysis

ECG signals are considered "non-stationary" because their statistical properties change over time. In contrast, signals like white noise and sine waves are "stationary" because their statistical properties remain constant over time. The Fourier transform isn't well-suited for analyzing non-stationary signals like ECG because it averages out the local temporal information and produces a global average that isn't very useful. In the acoustic lab, we learned about Short-Time Fourier Transform (STFT), which can be used to analyze non-stationary signals like breath sounds and produce a spectrogram. One downside of spectrograms is that the temporal window size is fixed. A smaller window size gives better temporal resolution, but increases the frequency bin width (as shown in the top row of Figure 1). Conversely, a larger temporal window achieves finer spectral resolution but makes temporal resolution coarser (as shown in the bottom row of Figure 1).

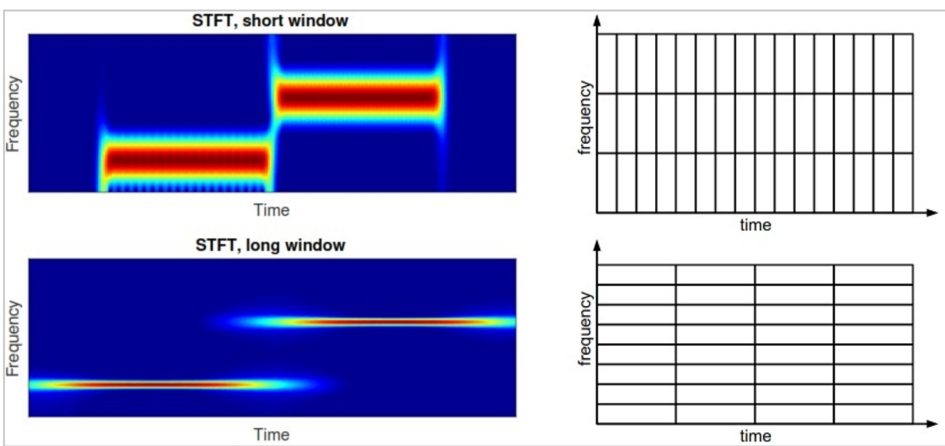


Figure 2. Tradeoffs between temporal and frequency resolutions in the Short-Time Fourier Transform (STFT). Because of the uncertainty principle, the temporal window length is inversely proportional to the frequency bin width. Figure source: [Panoradio SDR]

Multiresolution or multiscale analysis is an intuitive solution that avoids the tradeoff between temporal and spectral resolutions by adjusting the window size based on the analysis frequency. Typically, the temporal window size is divided by two as the frequency increases. Figure 3, right panel, shows a dyadic grid. As the frequency increases, the temporal window is shortened to allow for finer spectral resolution. The resulting graph is called a scalogram, which is different from the spectrogram produced by STFT. Multiresolution analysis also allows us to go beyond the Fourier Transform by including more general basis functions that are not limited to sinusoids. Some of these functions are called wavelets (Figure 4), which are characterized by their small, wavy shapes and specific mathematical properties.

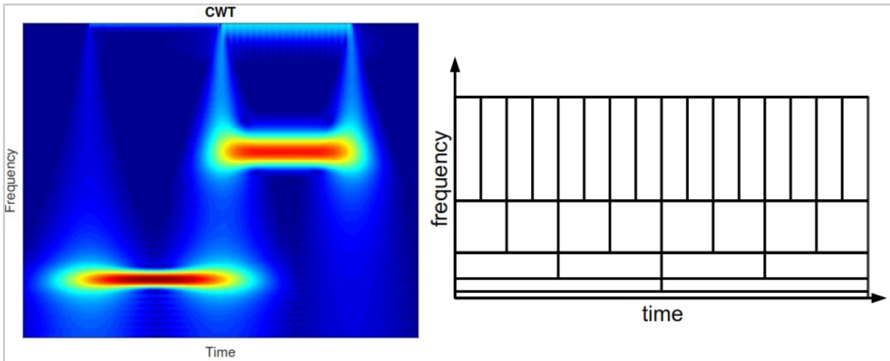


Figure 3. Multiresolution analysis. The basis function used here is a Morlet wavelet. The left panel shows the scalogram and the right panel shows the dyadic grid. Figure source: [Panoradio SDR]

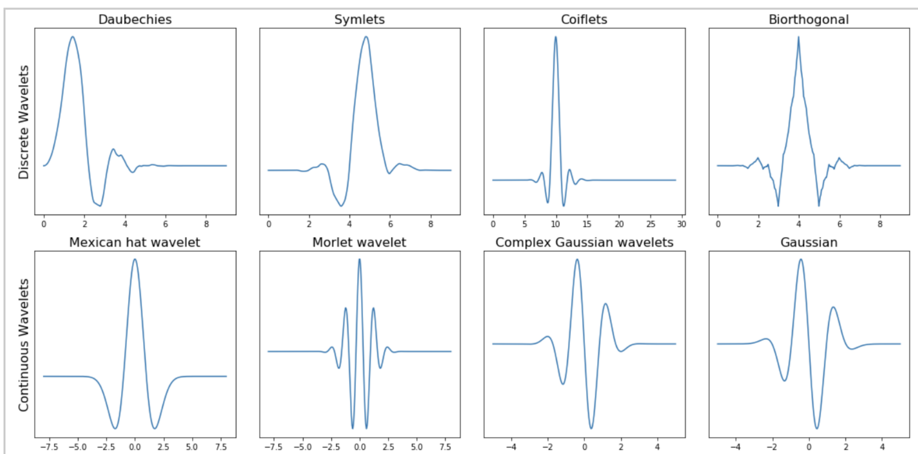


Figure 4. Some popular families of wavelets. Certain types of wavelets, such as the Symlets wavelet, resemble the characteristic QRS complex observed in ECG signals, making them useful tools in ECG signal analysis. Figure source: [ML Fundamentals].

3. Cardiac Rhythms

The heart typically beats in a regular rhythm called **sinus rhythm**, where natural heartbeats are initiated by the pacemaker of the sinoatrial node. This rhythm usually ranges from 60 to 100 beats per minute (bpm) with evenly spaced (< 10% variations) QRS complexes. Irregular heart rhythms are known as **fibrillation** and can be classified based on the location where the abnormal electrical activity begins. Two common types of fibrillation are atrial fibrillation (AF) and ventricular fibrillation (VF), with VF being more life-threatening than AF. **Bradycardia** and **tachycardia** occur when the heart rate is too slow (< 60 bpm) and too fast (> 100 bpm), respectively. **Asystole**, commonly known as flatline, is the absence of heart rhythm due to cardiac arrest.

Pre-Lab Exercise

There is no pre-lab exercise this week. However, please remember to fill out the final project topic ranking survey. If you are a graduate or honor student, you must also fill out the special project proposal survey.

In-Lab Exercise

Part 1. Heart Rate Measurement.

In Part 1 of this lab, we will collect ECG data and then use Discrete Wavelet Transform (DWT) to reduce noise. Next, we will apply a peak detector to extract the heart rate. Finally, we will compare the performance of heart rate measurement with and without DWT denoising to evaluate the effectiveness of the technique.

1. Download the files from the assignment page:

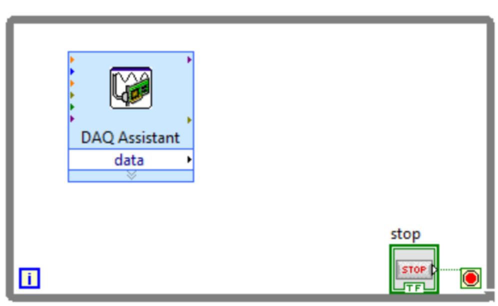
- ecg_sim.vi: This subVI simulates ECG as if they are recorded in real-time.
- noise_sim.vi: This subVI simulates some noises commonly found in real-world ECG recording.
- ecg_data.zip: This file contains the ECG recordings of a few patients from the MIT-BIH Arrhythmia Database. Remember to uncompress this file to get the contents inside.

2. Create a new VI. Name it “468_lab8_part1” or with some other name that makes sense. Create a **while-loop** and create a control for the stop condition.

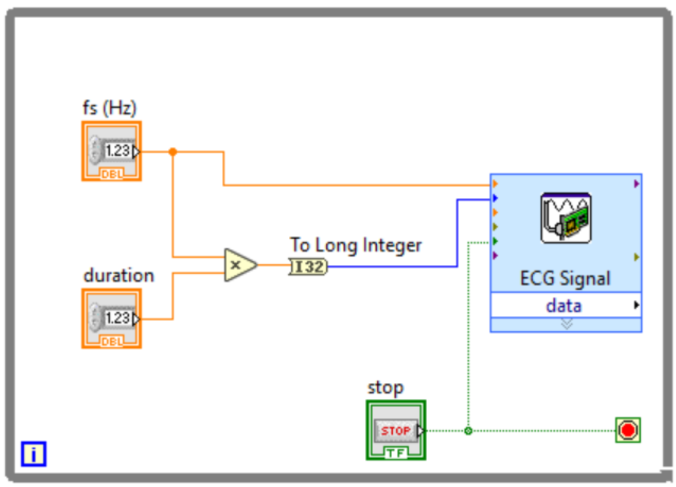
3. Drop a **DAQ Assistant** inside the while-loop. We will use this **DAQ Assistant** to collect ECG voltage. Configure the **DAQ Assistant** as follows:

- Acquisition Mode: “N Samples”
- Terminal Configuration: “RSE”
- Advanced Timing > Additional Timing Settings > Timeout (s): 12 seconds.

Note: We increased the timeout to more than 10 seconds because we need to collect 10 seconds of ECG data to calculate the heart rate. If we do not increase the timeout, the DAQ Assistant will generate an error.



4. We want to let the user control the sampling frequency and duration. Let's create two numeric controls and name them "fs" and "duration". Use the two numbers to calculate the number of samples. Then, wire "fs" and the number of samples you just calculated to the appropriate terminals of the **DAQ Assistant**.



5. To connect the AD8232 breakout board to the NI USB-600X DAQ module, refer to the example shown in Figure 5 (equivalent connections are also acceptable). Keep in mind that the AD8232 is not 5V-tolerant, so it's necessary to step down the 5V output from the DAQ module using either a voltage regulator or a voltage divider circuit.

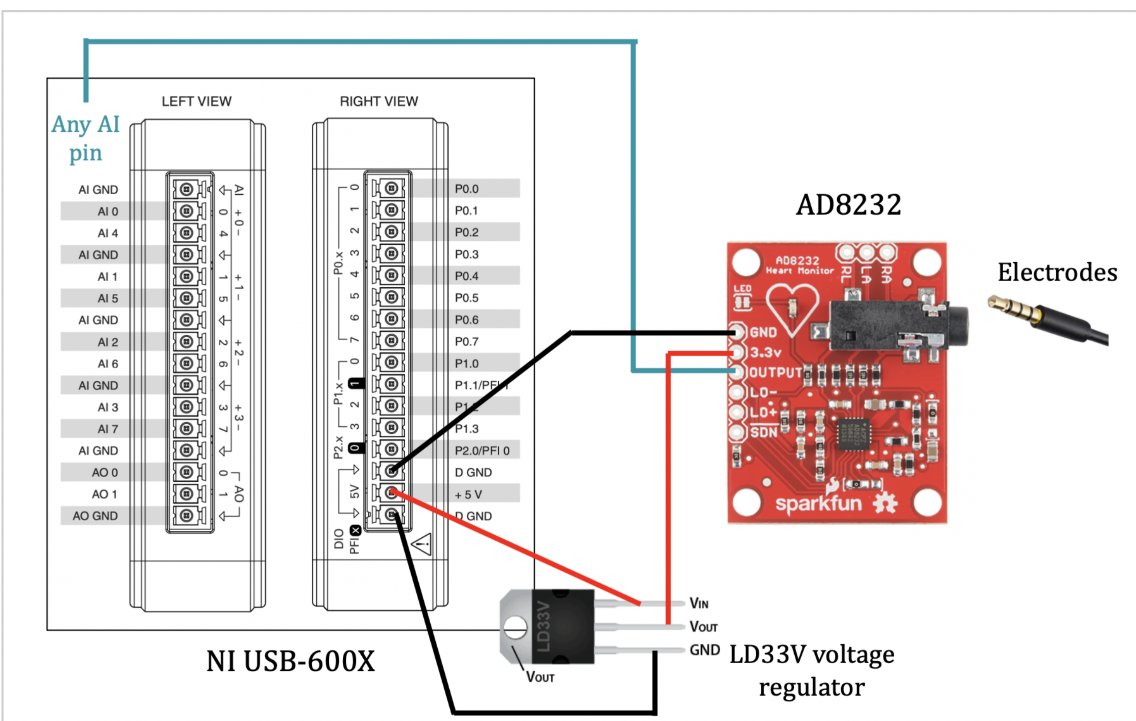


Figure 5. Connecting AD8232 to NI USB-600X. Use a voltage divider circuitIf if LD33V is not provided.

6. Place the electrodes according to Figure 6 (either is fine). Please note that although the AD8232 comes with a host of safety features to protect the test subject from electric shock, these features are not foolproof. To ensure safe operation, please avoid connecting the AD8232 to benchtop power supplies or other hazardous equipment. In case of emergency, pull the electrode cable from the headphone jack of AD8232.

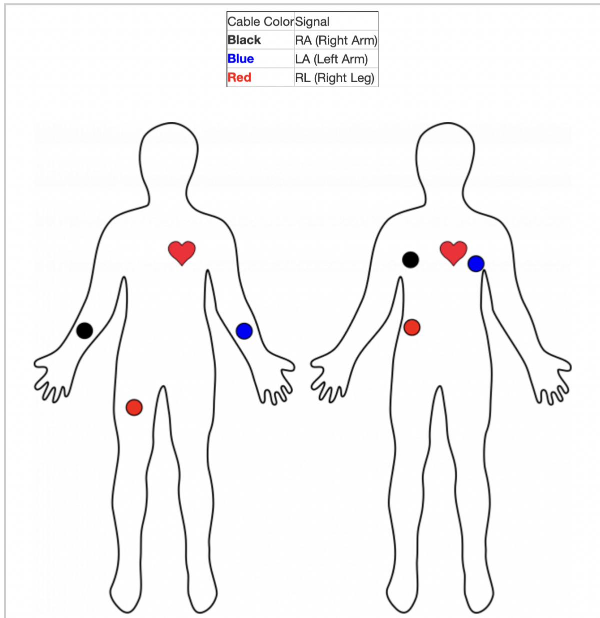
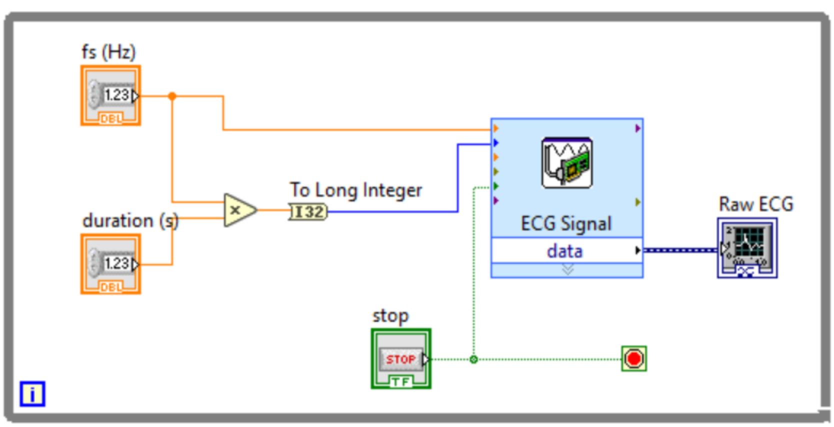


Figure 6. Typical sensor placement. Figure source: [Sparkfun AD8232 Hookup Guide](#).

7. Use a waveform graph to visualize the raw ECG signal.



8. Let's add some simulated noise to the ECG signal. The descriptions of the noises can be found in Table 1. In the function palette, go to "Select a VI…" and select the noise_sim.vi you just downloaded. For each of the terminals, do the followings:

- noisy_type: create a control.
- noise_amp: create a control.

- fs: connect it to the existing control.
- duration: connect it to the existing control.
- noise: add it to the raw ECG signal, then visualize noisy ECG signal on a **Waveform Graph**. Remember to convert dynamic data to a 1D array.

Note: In the figure below, I create local variables for "fs" and "duration" to keep the block diagram a little bit cleaner.

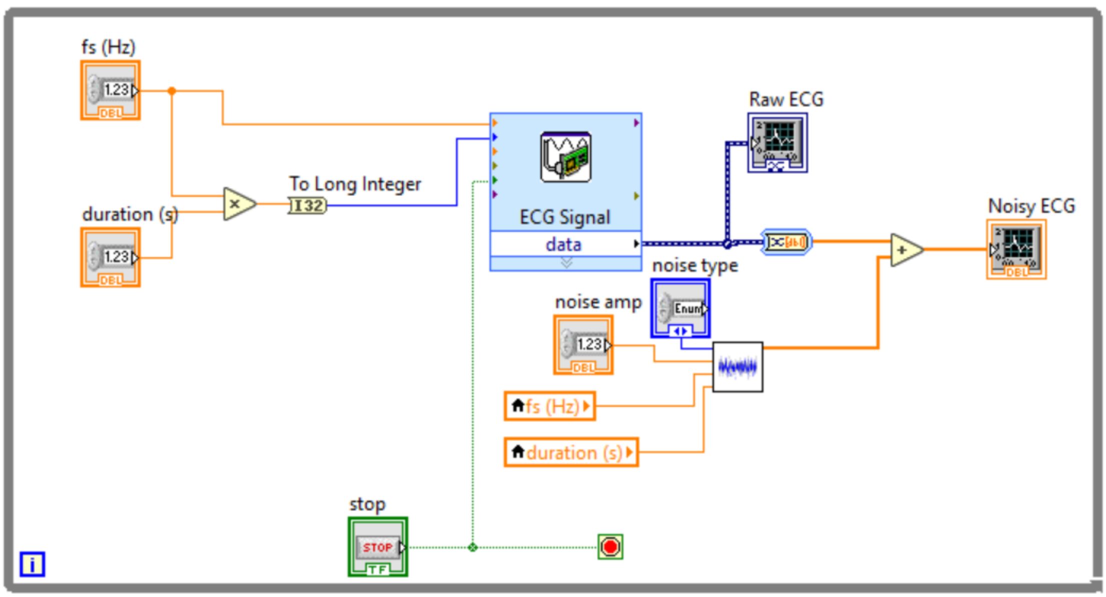
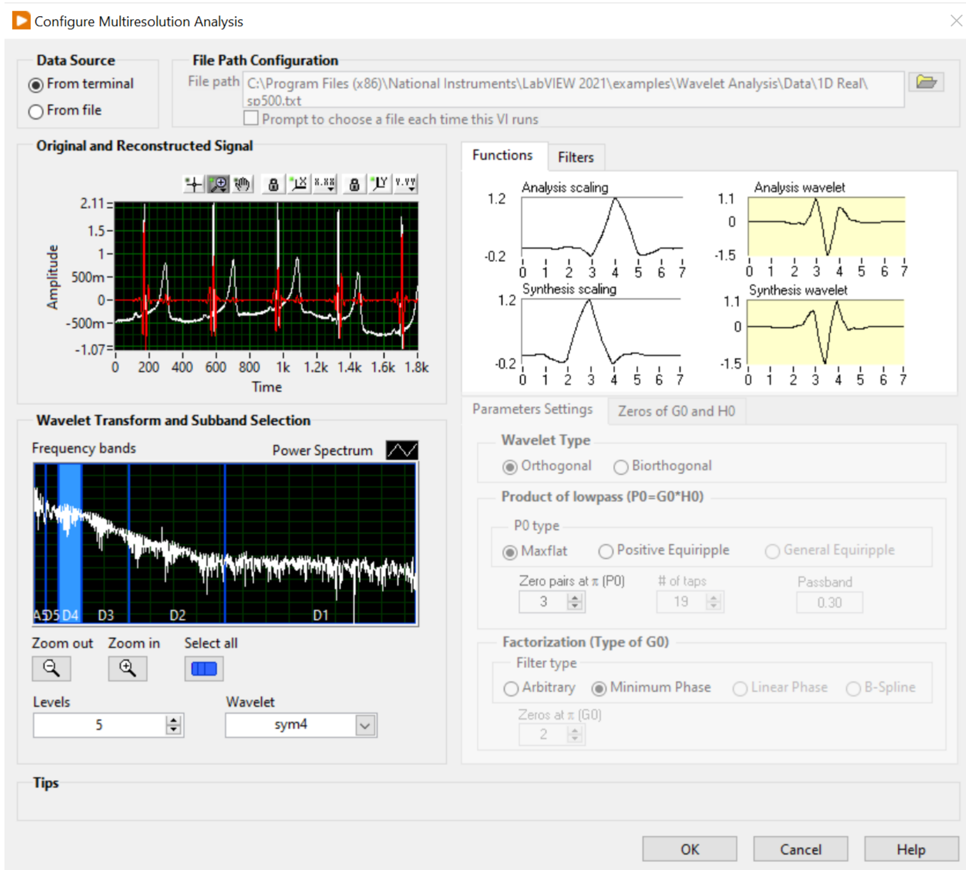


Table 1. Noise descriptions of noise_sim.vi.

Noise type	Noise description
None	no noise
Slow drift	0.2 Hz sinusoid with a random phase
White noise	Generated using the built-in white noise generator
Muscle artifact	Weighted combination of 4 high frequencies sinusoids with random phases
Motion artifact	Step function with a random transition time
60 Hz	60 Hz sinusoid with a random phase

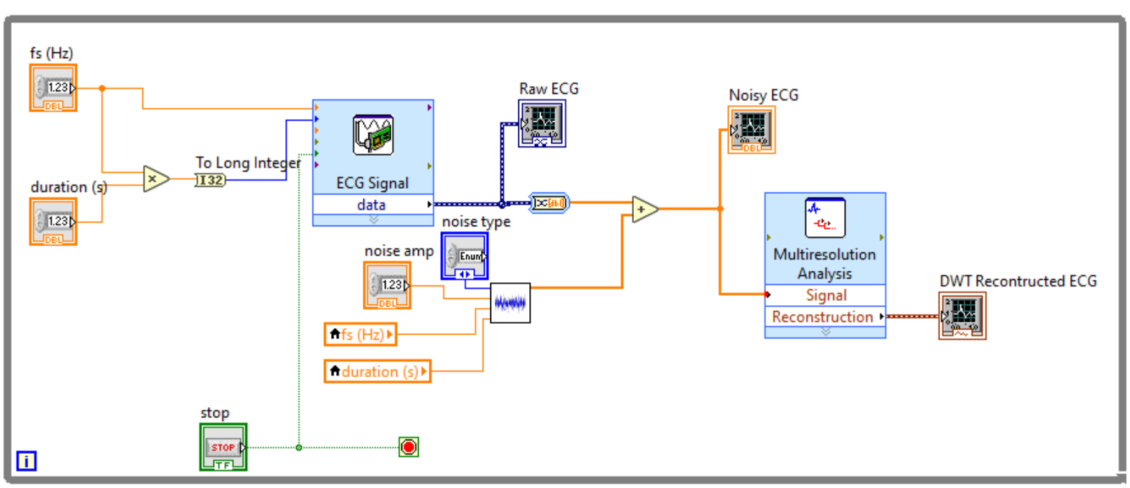
9. Now let's extract the R-peaks from the ECG signal. Go to Signal Processing > Wavelet Analysis > Discrete Wavelet and select **Multiresolution Analysis** function. You will see the following configuration window will pop up:



Below are the recommended configurations that will accentuate the R-peaks:

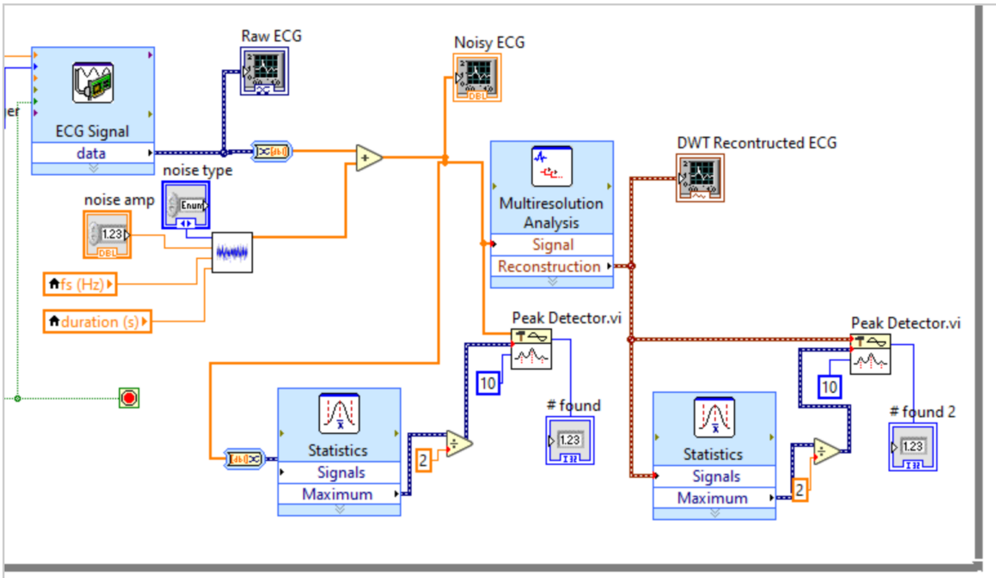
- <u>Levels</u>: set to 5.
- <u>Wavelets</u>: use sym4.
- <u>Frequency bands</u>: select D3 and D4.

Visualize the reconstructed signal with a waveform graph. Do you see the R-peaks? Feel free to try out different DWT configurations to improve R-peak detection.



Print date: Feb 23, 2023

10. To calculate the heart rate using ECG signals before and after DWT, let's use two **peak detectors** located in the Signal Processing > Sig Operation menu. Set the width to 10 samples. The peak threshold is adjustable, and I have set it to half of the maximum using the **Statistics** function found in Mathematics > Prob & Stat . However, you may choose a different threshold value as needed.



11. Obtain a pulse oximeter and measure the heart rate. Record the results in Table 2. A template is provided on the assignment page.

Table 2. Heart rate measurements with different simulated noise.

Noise type	Heart rate measured by pulse oximeter (bpm)	Heart rate measured by VI (bpm) without DWT	Heart rate measured by VI (bpm) with DWT
None			
Slow drift			
White noise			
Muscle artifact			
Motion artifact			
60 Hz			

12. After completing this part, there is no need to record any more live data. You can now remove the electrodes

Note: If you run out of time during the lab period, you can stop at this step. You can turn in what you have, and no points will be deducted for incomplete work.

Part 2. Arrhythmia Detection.

As instructed by the Department of Bioengineering, we will not be attempting any medical diagnosis on the students during the lab activities. Instead, we will be using simulated ECG signals obtained from the MIT-BIH Arrhythmia Database, which is publicly available at https://physionet.org/content/mitdb/1.0.0/. The **ecg_data.zip** file contains a subset of this database, and Table 3 provides a summary of the patient information. Each signal in the dataset has a duration of 10 seconds and was sampled at a frequency of 360 Hz."

Table 3. Patient information. Some terminologies/abbreviations: Sinus = normal; PVC = premature ventricular contraction, a type of arrhythmia; AFIB = atrial fibrillation, a type of arrhythmia; TA = tachycardia; BR = bradycardia; AV block = atrioventricular block.

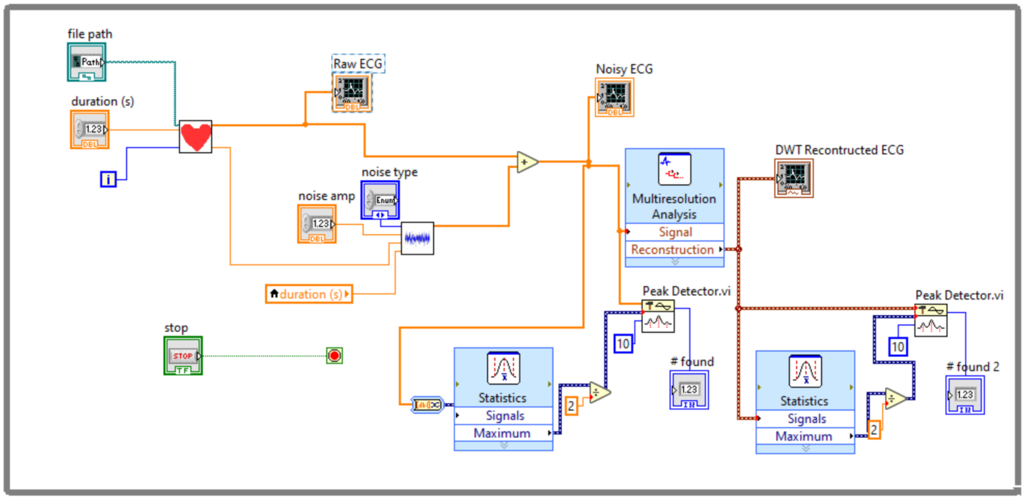
Patient number	fs (Hz)	Age	Gender	Diagnosis	Any noise present?
113	360	24	F	Sinus	Baseline drifts
117		36	M	Sinus	no
201		68	M	PVC and AFIB	no
203		43	M	AFIB and TA	Baseline drifts and muscle artifact
208		23	F	PVC	no
232		76	F	BR caused by 1st degree AV block	no

1. Make a copy of the VI you created in Part 1, name this file “468_lab8_part2” or something that makes sense.

2. Replace the DAQ Assistant with **ECG simulator subVI** (i.e., ecg_sim.vi). You need to make the following changes for the following terminals:

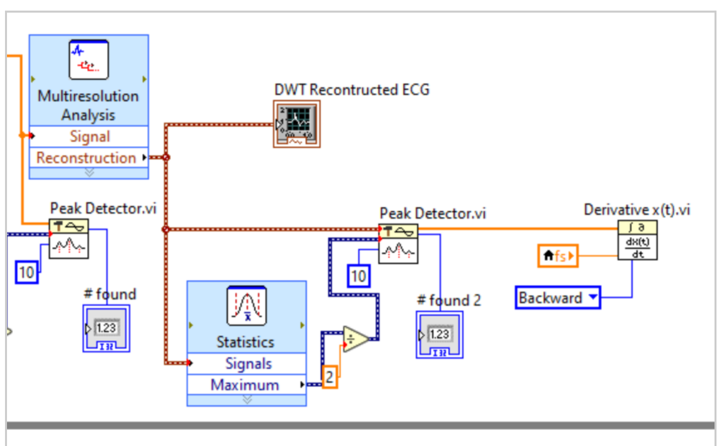
- file path: create a control.
- duration: use the existing control.
- i: connect it to the loop index of the while-loop.
- signal: connect it to the wire that was connected to the waveform graphs.
- fs: connect it to the “fs” terminal of the noise simulator subVI.

Delete any unused wires, etc.

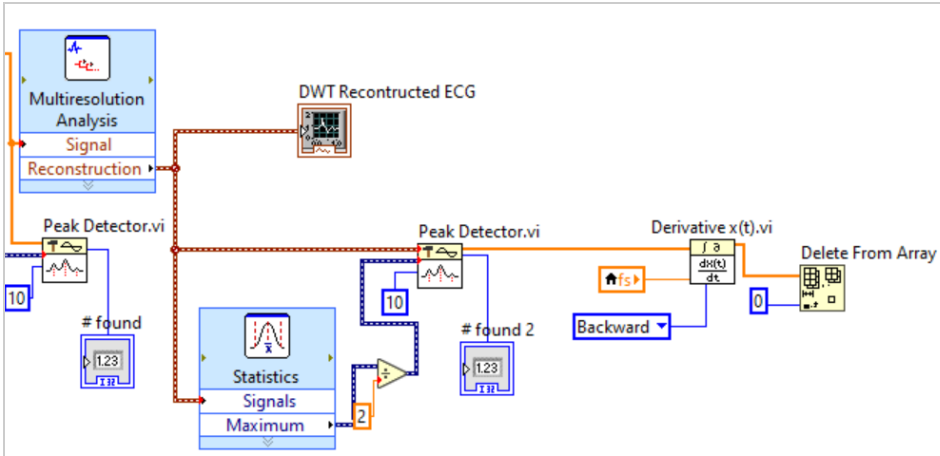


3. Let's measure the R-R intervals. Go to Mathematics > Integ & Diff menu and select **Derivative x(t)**. Wire the terminals according to the followings:

- X: connect it to the "Locations" output of the second peak detector (i.e., the one that detects peaks for the reconstructed ECG).
- dt: connect it to the sampling frequency.
- method: create a constant and set it to "Backward".
- dX/dt: this outputs an array of the R-R intervals. We will explain this in the next step.



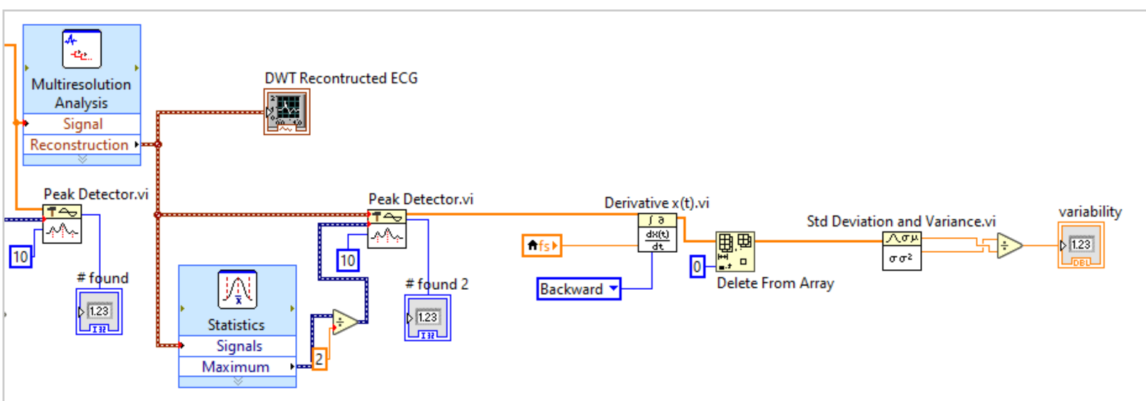
4. The "dX/dt" output terminal of the **Derivative x(t)** function should return an array that contains the R-R intervals. However, since the first element of the array is actually just the time at which the first R peak occurs. We need to remove it using the Delete from Array function found in the Array palette.



5. Next, we will measure the variability in the R-R intervals, which can be a useful, albeit crude, metric of arrhythmia. For this purpose, we will define the variability as follows:

$$variability = \frac{STDDEV(R-R\ intervals)}{AVG(R-R\ intervals)}$$

To calculate the variability, you can use the **S.D. & Variance** functions located in the Mathematics > Prob & Stat menu. Once the calculation is complete, you can create an indicator for the variability.



6. Turn off the noise and set the duration to 10 seconds. Record your observations in the following table.

Table 4. Variability in R-R intervals without noise.

Patient number	Actual diagnosis	variability (std/mean) in R-R intervals (After wavelet transform)
113	Sinus	
117	Sinus	
201	Arrhythmia	

203	Arrhythmia and Tachycardia	
208	Arrhythmia	
232	Bradycardia	

7. (Optional) Repeat the previous step but with a noise that you choose.

Table 5. (Optional) Variability in R-R intervals with ____________ noise.

Patient number	Actual diagnosis	variability (std/mean) in R-R intervals (After wavelet transform)
113	Sinus	
117	Sinus	
201	Arrhythmia	
203	Arrhythmia and Tachycardia	
208	Arrhythmia	
232	Bradycardia	

8. Summarize your findings in Table 6. Please keep your responses short. For example, Question: "What sampling frequency did you use?", answer: "360 Hz".

Table 6. Lab summary. Answer the following questions based on your observations. Again, there is no wrong answer. Your answers can be just a few words.

Question	Answer
Which wavelet and number of levels did you end up using in multiresolution analysis?	
Which kind of noise distorts ECG signals the most?	
For Table 4, is 0.1 a good cutoff for the variability metric to detect arrhythmia? If not, what cutoff would you use?	
Will you recommend wavelet analysis to your friends and colleagues?	
(Optional) Any suggestions for improving the performance of the VI?	

Deliverables

- To receive full credit, please submit whatever work you have completed to demonstrate your participation in the lab activity. A template is provided on the assignment page.

Note: All deliverables may be co-authored by up to three students.