

# Mini-Project 06: Simple Song Detection

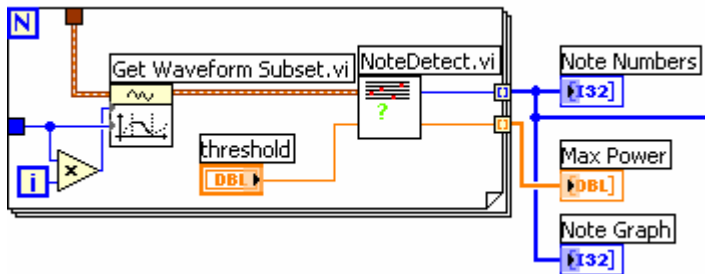
## Background

For this mini project you will write a simple LabVIEW function that listens to a wav file of a simple song and identifies the notes being played. The wav files<sup>1</sup> have some simple songs on which you can practice. The range of notes in these files is restricted to the range used in the previous mini project.

## Approach

Read the song wav file, and then break it into many smaller, equal length *segments*. Each segment should be around 50-100 ms. This is easily done with LabVIEW's **Get Waveform Subset VI**. Play with **Get Waveform Subset** until you are sure it works the way you think.

Next set up a `for` loop and loop for each of the input segments. For example:



Notice **NoteDetect** is called for each segment and it returns the note it heard for that segment. You need to figure out how to compute the **Start** and **Duration** inputs for the **Subset** block. Note also how I plot the **Note Numbers** via **Note Graph** to get an overview of how the system is performing. Add a silence **threshold** parameter to **NoteDetect**. If the power of the loudest note is below the threshold, assume the input is silence and return a **noteNum** and **max** value that means silence (for example **noteNum** = -1 and **max** = 0).

## Grading

If you are able to show me an output that lists the note for each segment your grade will be as much as 7/10. Here is a sample of such an output:

notes =

```
-1 -1 -1 -1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 -1 4 4 4 4 4
4 4 4 4 3 -1 -1 -1 -1 -1 -1 2 2 2 2 2 -1 -1 0 0 0 0 0 0 0 0 0
```

If you can make your output more compact and only show the notes that were played you will get more credit. For example:

notes=

```
4 4 3 2 0 0 2 1 2 2 4 2
```

In this example, note 4 was played, then note 4 again, then note 3 (which was wrong), and so on. You don't have to have perfect output, but it needs to be pretty close.

Better yet, return the note numbers and the time the notes were on. Be sure to explain what your note numbers mean and units on your time.

<sup>1</sup> The wave files can be found on the DSP First web site on the link below where this handout is.

Full credit will be given to those who can tie this lab to Lab 1 and resynthesize the input. That is, use the note information from this lab as input to the synthesizer for Lab 1. But when you do, play it back at twice the tempo. Your input will be a wav file as well as your output.

## What's Due

For this mini-project you are to work on your own. You may discuss your ideas with others, but you may not share LabVIEW files. The policy stated on the Homework Procedures handout under "Responsibility" applies to this and other mini-projects.

What is due:

1. One page memo describing what you did. Be sure to have a sentence or two intro and conclusion. Have a listing of the notes you detected for the song **AuntRhody.wav**. Mark any errors in the output. Highlight any **extras** you did.
2. Print out your LabVIEW code. Selecting File→Print will start a print wizard. When you reach the "Print Contents" window, select "Icon, description, panel, and diagram." This will print out the whole works in a compact format.
3. If you resynthesize the song, drop the .wav file in the drop box on Angel.

A hard copy of the memo and LabVIEW printout are to be handed in at the start of class on the due date.