2016

Rockin' Hat

Victoria Crank

Ivanna Mahabir

Tatjana Jemmott

Jasmine Anne Turla

Rockin' Hat

Victoria Crank, Ivanna Mahabir, Tatjana Jemmott, Jasmine Anne Turla

Faculty Advisors: Dr. Gibbs, Julius Chatterjee, Dept of ECE, Florida Institute of Technology

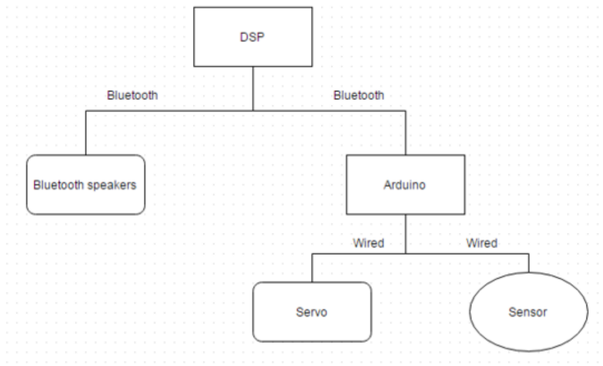
Objective

Build a talking hat that can analyze a given audio file and move the mouth accordingly.

Description

Physical:

The Hat's base and structure are 3-D printed into 4 interlocking pieces and a separate jaw for ease of assembly. The Arduino controls the sensors, the Bluetooth chip and the servo. These and the batteries are all housed within the base. The NI myRIO is a separate, external, real-time signal processor that uses a Bluetooth link to control and command the Arduino. It also uses a Bluetooth link to play the given audio file to a speaker.



System Block Diagram

Description

Software:

The Arduino stores the 'control file' for each song on an SD card. The control file is accessed and run when the myRIO sends commands. The myRIO synchronizes the hat movements to the audio playing on the speakers.

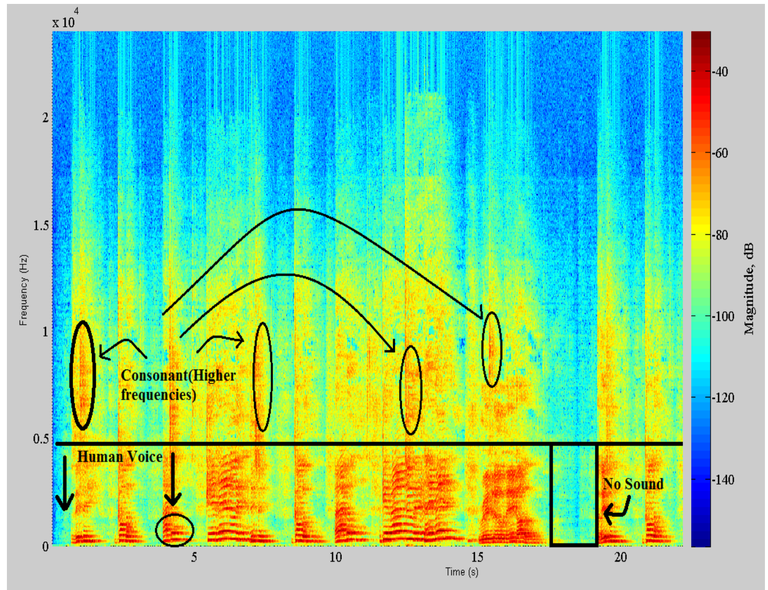
The myRIO is also responsible for the signal processing needed to decide on the mouth movements. This step is done before the user interacts with the hat, and it creates the 'control file' that the Arduino uses to control the mouth movements.

Signal Processing

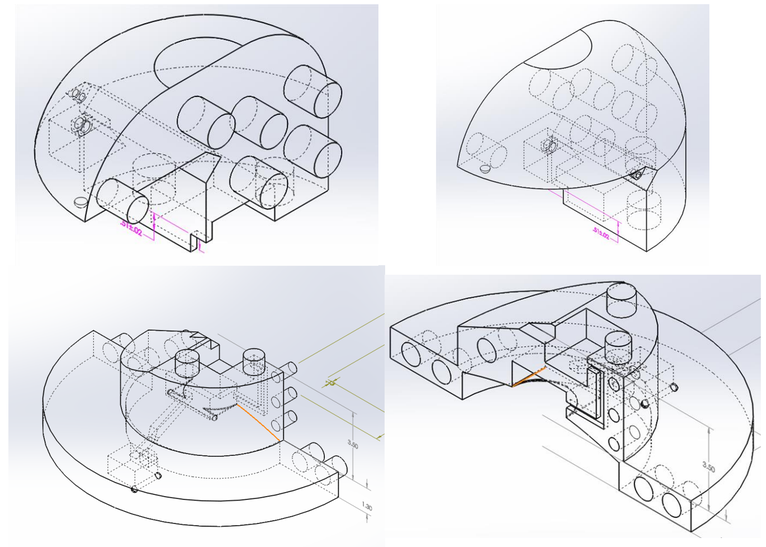
The information stored in an audio file is simply the amplitudes across time. Therefore by analyzing a vocals-only file, the areas where the amplitude is lower/near zero shows that there is no singing. This information alone is insufficient to decipher when the speaker is using separate words. Thus, the frequency information is needed.

Different sounds at the same loudness have different frequency contents. The audio waveform is transformed using a Fourier transform to extract the frequency content. The frequency explains more about the sound being played, as you can see in the example spectrogram.

Lower frequencies usually mean someone is talking, while higher frequencies mean sounds that aren't made with vocal chords, which can include consonant sounds, such as 's' or 'th'. These sounds are what the myRIO looks for when making a decision on when to close or open the mouth. The goal is to make a passable mouth movement, so there is no focus placed on identifying exact words or sounds.



Example Spectrogram of Audio Analysis



Conclusions

Over the course of our project, we established a Bluetooth link between the myRio DSP and the Arduino Mini situated within the Hat's base. With the addition of a Matlab analysis of a vocals-only file, we successfully passed this file through the link, instructing the Arduino to move the servo in time with given audio file. Using a control in LabView, users are able to select the mode at which they wish the Hat to be set at before execution.

Despite the Hat's singing illusion, there were several complications that occurred. The discovery that the myRio doesn't store .wav files easily led to an alternative route that would allow the DSP to play .wav files. The variance in the servo speeds also proved to be a concern given that the speed and frequency of the servo's rotation would determine how well the hat would mimic the vocal movements.

Extra Information

Acknowledgements to Mr. Beavers and the Electronic Support Lab for 3D printing services and the myRio.