Speech Recognition

Kyle Limbaga, Nicholas Riggins, Dawson Tocarchick

TABLE OF CONTENTS

01 **About the Project**

Speech Recognition background and what we learned

02 **Major Requirements**

The task at hand, and DSP requirements

03 **Project Goals**

04 **Project Stages**

Project and team organization

05 **Quick Startup Guide**

Here you could describe the topic of the section

06 **References**

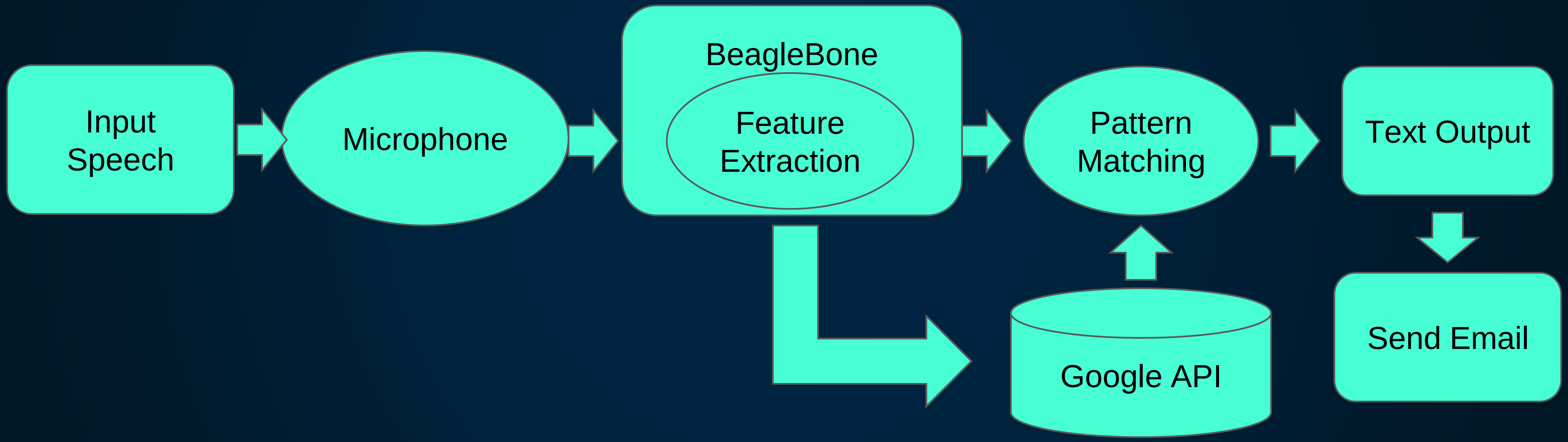
Here you could describe the topic of the section

Speech
Recognition

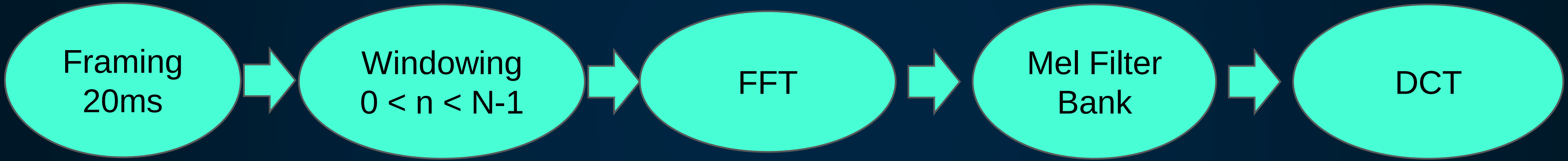
About Project

Our Speech Recognition project underlines the importance of Digital Signal Processing in real world products such as Speech Recognition. This project has not only taught us the importance, but the necessary aspects that Digital Signal Processing has to offer.

Overall Process



Feature Extraction



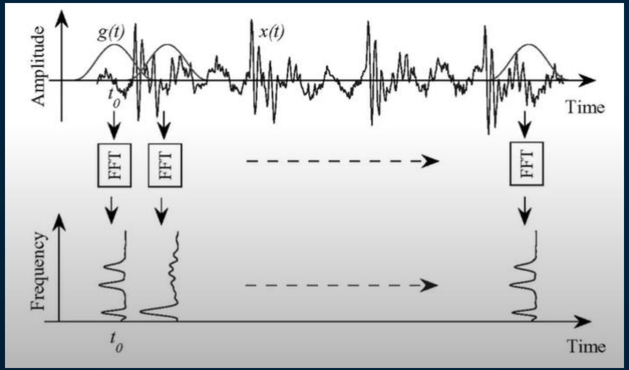
Framing 20ms

Windowing $0 < n < N-1$

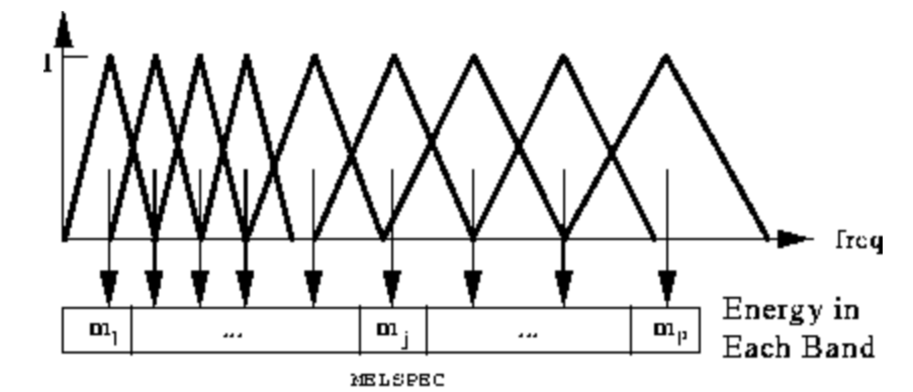
FFT

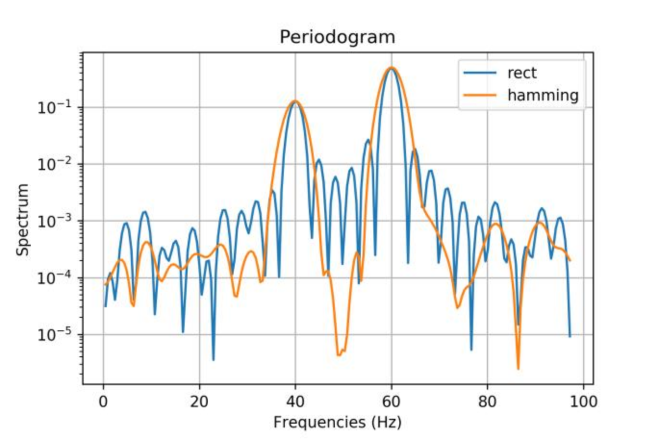
Mel Filter Bank

DCT



Examples


1
freq
m1
...
mj
...
mp
Energy in Each Band
MELSPEC


Periodogram
rect
hamming
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
Spectrum
0
20
40
60
80
100
Frequencies (Hz)

Machine Learning

Phonemes

A unit of sound that distinguishes one word from another in a particular language

Hidden Markov Model

HMM uses statistical probability to reconstruct the phrase that has just been said so by putting the right phonemes each other

Deep Neural Network

This is also known as deep learning which is an Artificial Neural Network (ANN) with multiple layers between input and output

Major Requirements

Fast Fourier Transforms

Performing FFT functions on our .wav file that is created

Speech to Text Output

We want to speak into a microphone and our Beagle Bone Black to output the transcript into the console

Quick Data Transfer

Using Email to send and display sample data

Project Goals

The goal of our project is to make use of the Beagle Bone Black and use our knowledge of Digital Signal Processing to further our knowledge on speech recognition and how it works.

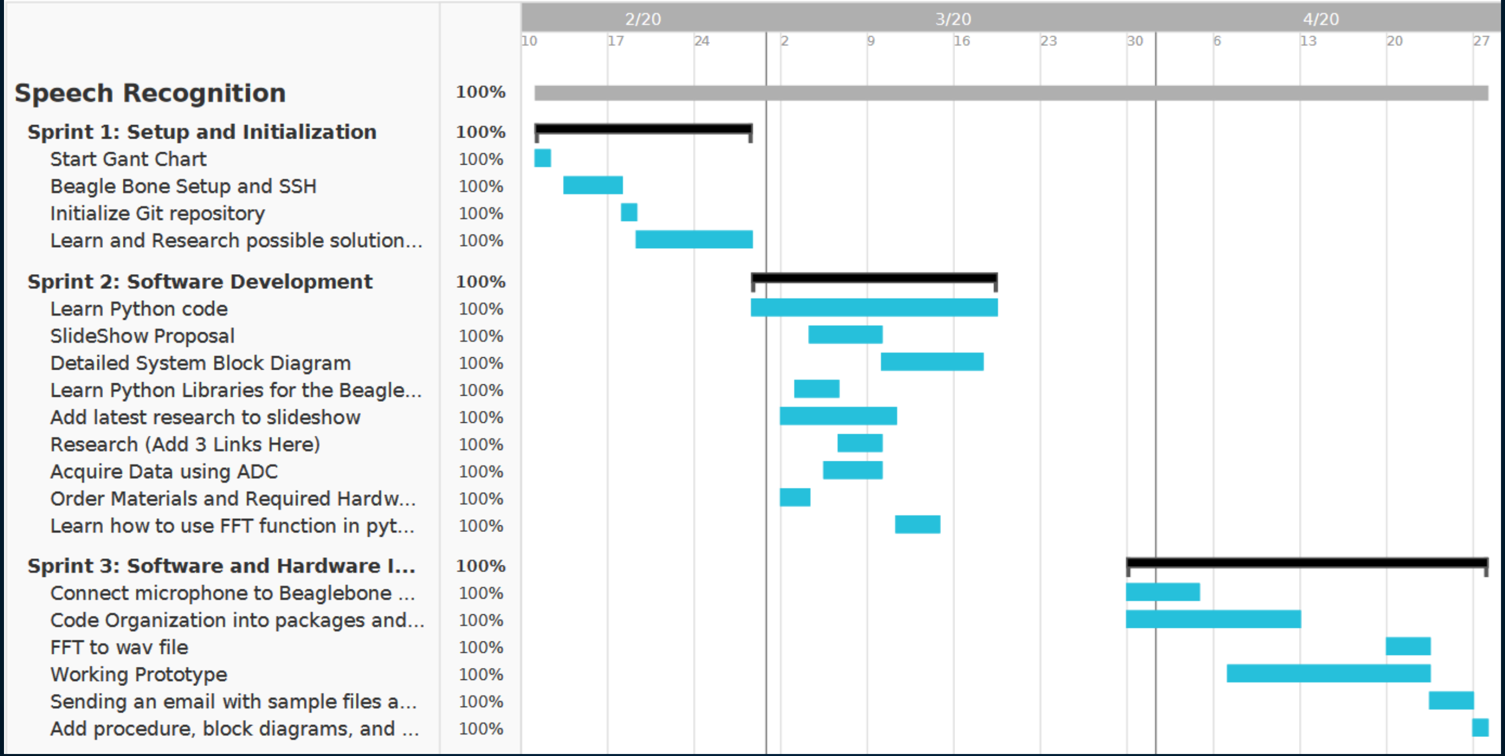
“Pretty darn good at Sudoku.”

— ”The Great” Richard Pederson

Project Stages

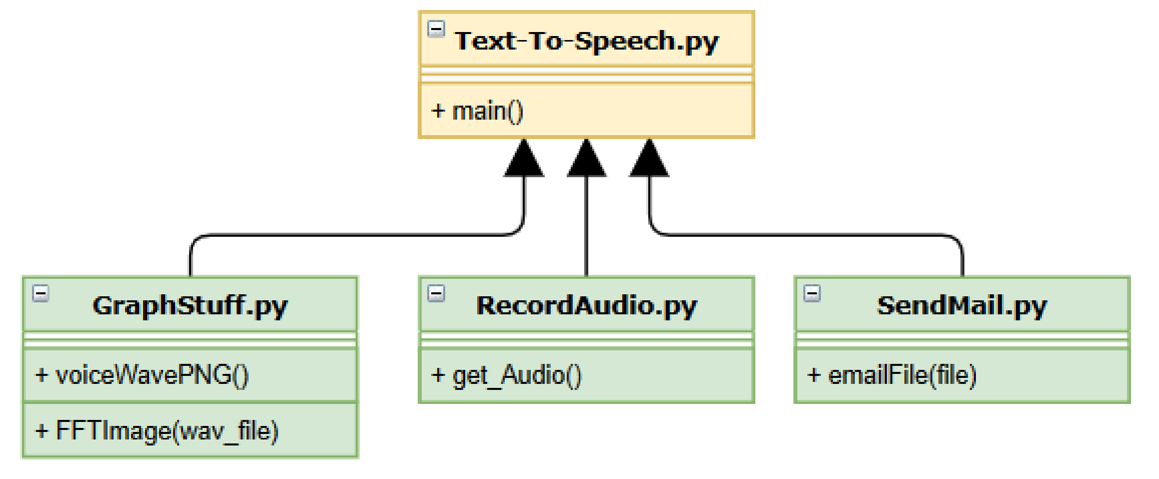
- Our Design Timeline
- Programming
- Testing

OUR TIMELINE



Speech Recognition	**100%**
Sprint 1: Setup and Initialization	**100%**
Start Gant Chart	100%
Beagle Bone Setup and SSH	100%
Initialize Git repository	100%
Learn and Research possible solution...	100%
Sprint 2: Software Development	**100%**
Learn Python code	100%
SlideShow Proposal	100%
Detailed System Block Diagram	100%
Learn Python Libraries for the Beagle...	100%
Add latest research to slideshow	100%
Research (Add 3 Links Here)	100%
Acquire Data using ADC	100%
Order Materials and Required Hardw...	100%
Learn how to use FFT function in pyt...	100%
Sprint 3: Software and Hardware I...	**100%**
Connect microphone to Beaglebone ...	100%
Code Organization into packages and...	100%
FFT to wav file	100%
Working Prototype	100%
Sending an email with sample files a...	100%
Add procedure, block diagrams, and ...	100%

Program Outline



Testing Process:

Each module in green is located in the "Setup" package. Therefore, the advantage of creating a package is that the developers can test code individually and later implement into the main program.

Text-To-Speech.py

Main Program File, which integrates the entire project into an orderly program

GraphStuff.py

- **voiceWavePNG():** Generates a png file of the .wav file
- **FFTImage(wav_file):** Generates visual graph of single and dbl FFT waveform.

RecordAudio.py

get_Audio() : Records the audio file using speech recognition library

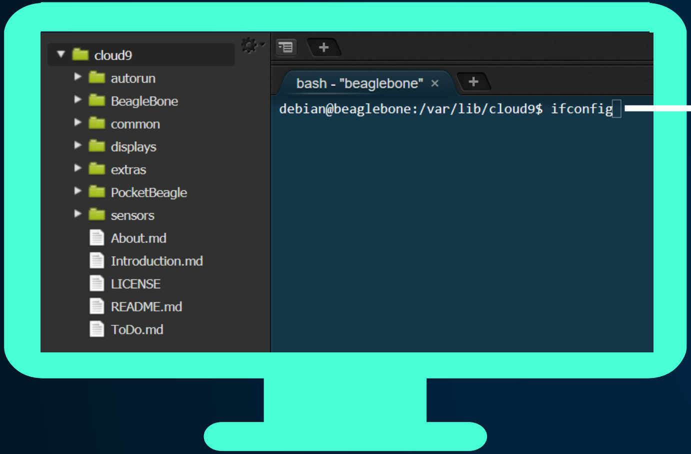
SendMail.py

emailFile(file) : Sends an email to the user with a selected file attachment

Quick Start Guide

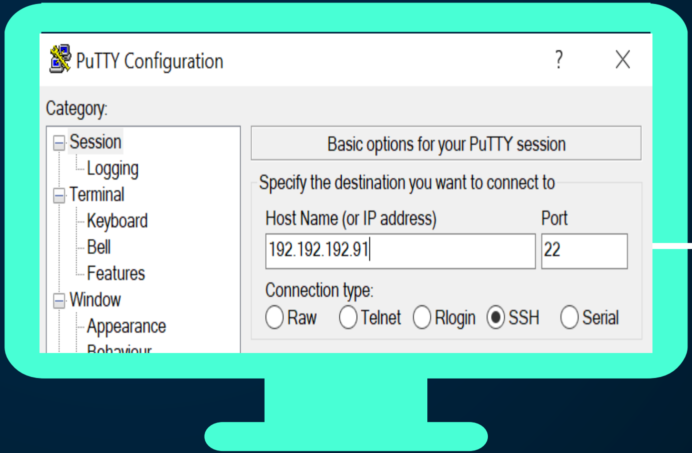
- Run Putty/SSH
- Creating and Navigating Directories
- Run Program

Running Putty: SSH tool



```
debian@beaglebone:/var/lib/cloud9$ ifconfig
```

Step One:
ifconfig



Specify the destination you want to connect to

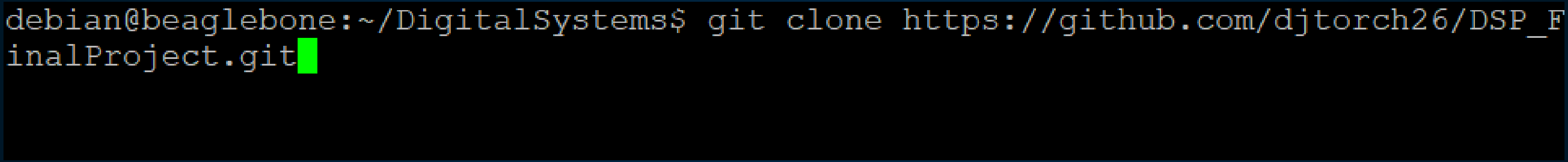
Host Name (or IP address): 192.192.192.91

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Step Two:
Start SSH Software and type ip address acquired from ifconfig command

Creating .git Directory



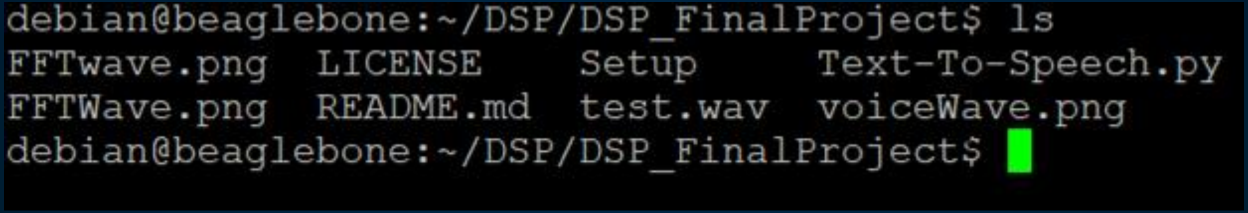
```
debian@beaglebone:~/DigitalSystems$ git clone https://github.com/djtorch26/DSP_F
inalProject.git
```

Step Three: Clone Repository command:

“git clone https://github.com/djtorch26/DSP_FinalProject.git”

Navigation commands:
ls: shows a list of files and folders inside a directory
cd: change directory ie. cd DigitalSystems

Navigating the .git directory



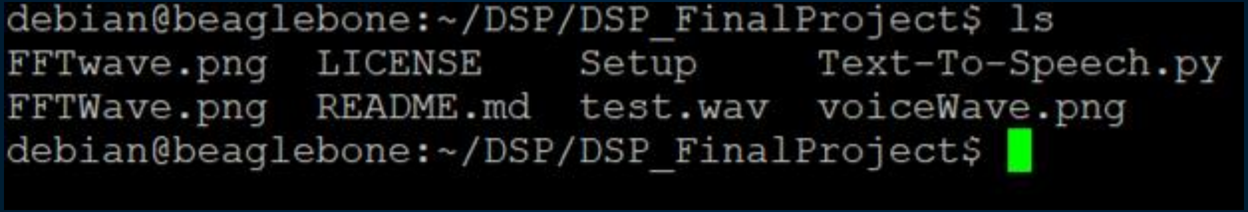
```
debian@beaglebone:~/DSP/DSP_FinalProject$ ls
FFTwave.png  LICENSE    Setup     Text-To-Speech.py
FFTWave.png  README.md  test.wav  voiceWave.png
debian@beaglebone:~/DSP/DSP_FinalProject$
```

Step Four:

“cd DSP_FinalProject”

Navigation commands:
ls: shows a list of files and folders inside a directory
cd: change directory ie. cd DigitalSystems

Running the Program



```
debian@beaglebone:~/DSP/DSP_FinalProject$ ls
FFTwave.png  LICENSE    Setup     Text-To-Speech.py
FFTWave.png  README.md  test.wav  voiceWave.png
debian@beaglebone:~/DSP/DSP_FinalProject$
```

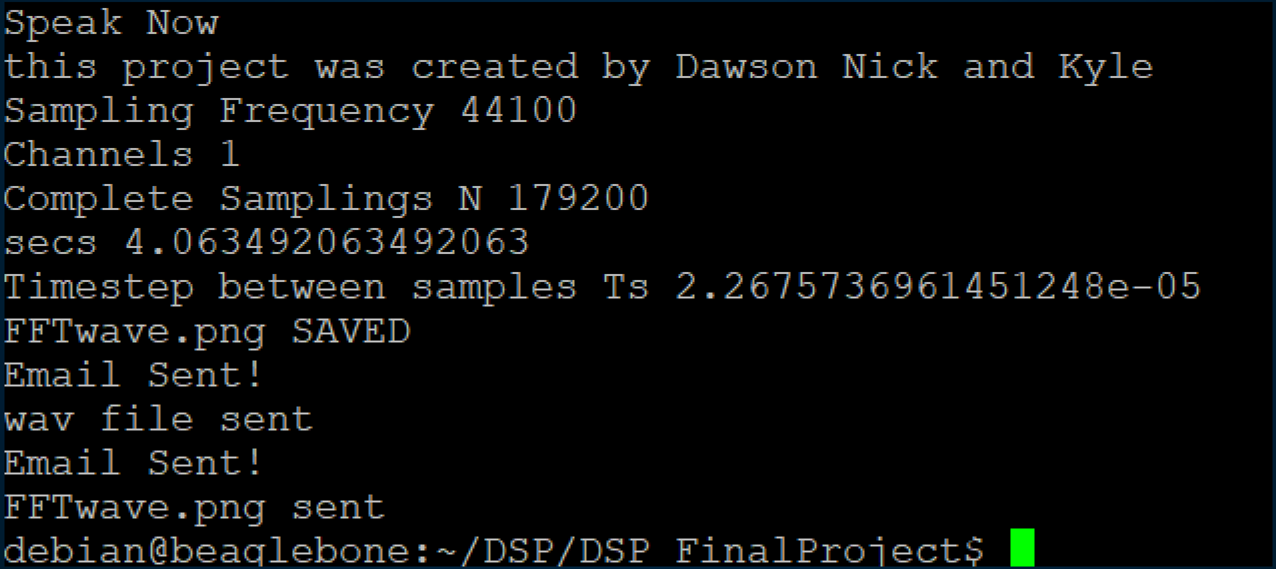
Step Five:

“sudo python3 Text-To-Speech.py”

This command will run the python program and will send back data based off of the .wav file created.

Navigation commands:
ls: shows a list of files and folders inside a directory
cd: change directory ie. cd DigitalSystems

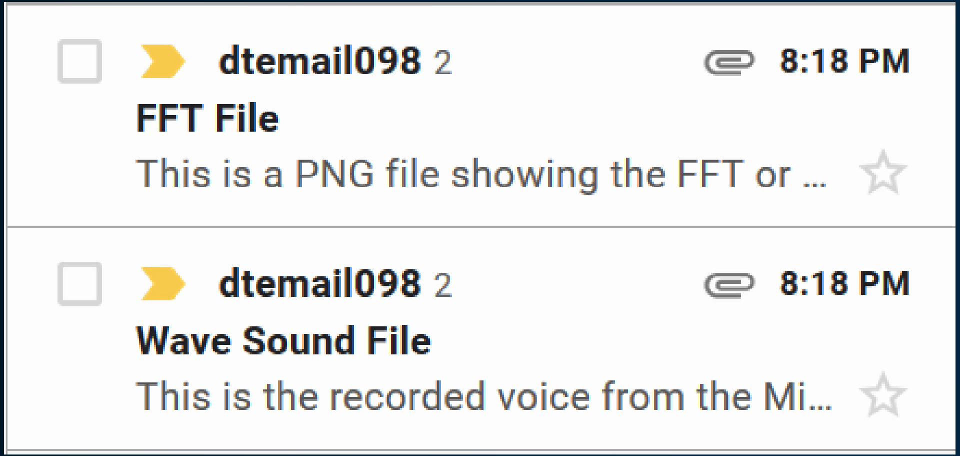
Program Return



```
Speak Now
this project was created by Dawson Nick and Kyle
Sampling Frequency 44100
Channels 1
Complete Samplings N 179200
secs 4.063492063492063
Timestep between samples Ts 2.2675736961451248e-05
FFTwave.png SAVED
Email Sent!
wav file sent
Email Sent!
FFTwave.png sent
debian@beaglebone:~/DSP/DSP FinalProject$
```

This is the return when the user says “This project was created by Dawson, Nick, and Kyle.”

Program Return



□ dtemail098 2 8:18 PM

FFT File

This is a PNG file showing the FFT or ... ☆

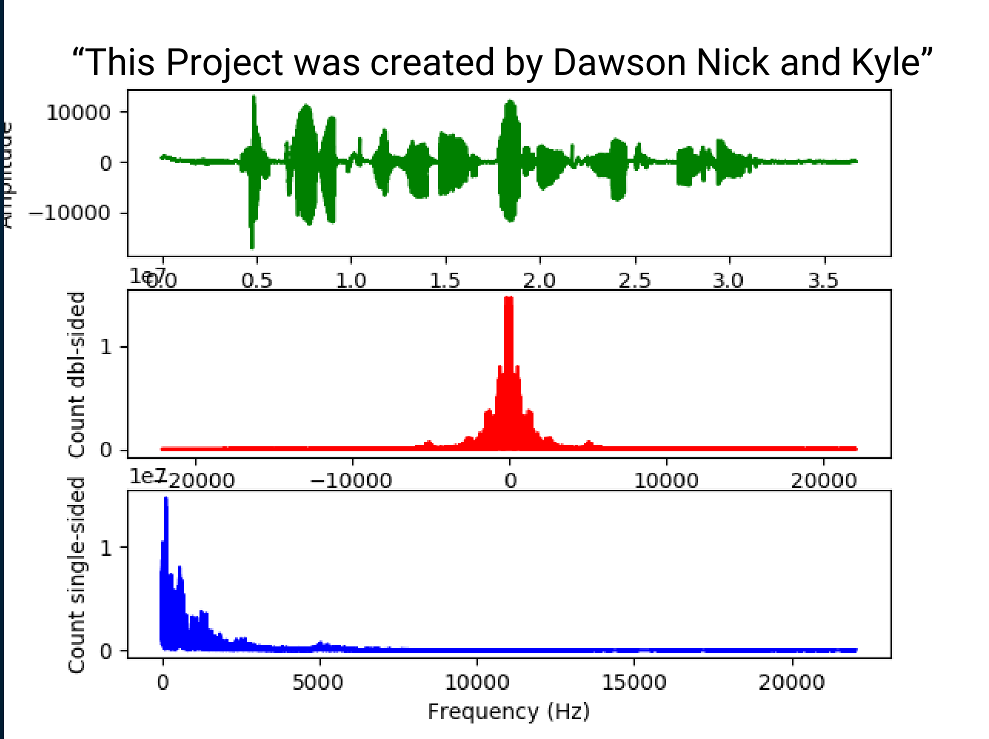
□ dtemail098 2 8:18 PM

Wave Sound File

This is the recorded voice from the Mi... ☆

Audio File sent by email

Program Return



PNG File sent by email

Future Improvements

USABILITY

Creating a UI, user interface, to make it easier for the user to operate this program. This would mean the user would not have to SSH into the console. The user would only need to push a button

Data Transfer

Adding a change email feature, so when the program is running, you can send the data to any selected email, including multiple emails.

Fixing bugs

When recording your voice, there is an awkward period of time where you don’t know when to speak. This is in regards to the ALSA library printing to console. This would also include improving code and removing unnecessary code and packages.

RESOURCES

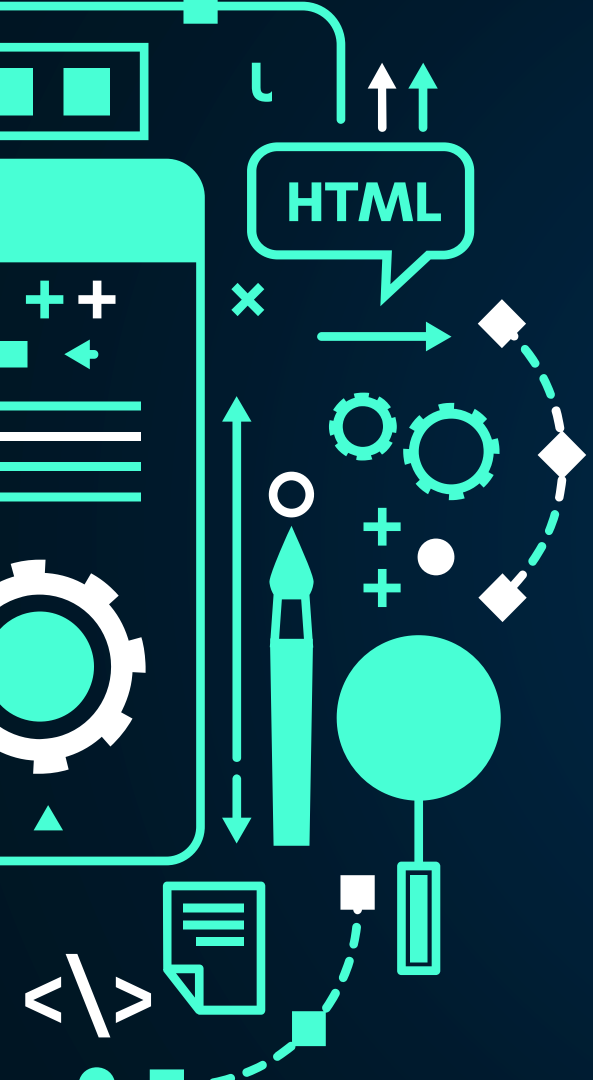
These are some of the sources that helped us with our project

Websites

- Github
- The Fundamentals of FFT-Based Signal Analysis and Measurement
- Audio Processing: Speech Synthesis and Recognition
- Speech Recognition Library
- Python For Scientists and Engineers
- How to use Email in Python Project
- Deep Learning Material
- Speech Recognition and Deep Learning

Libraries

- Numpy
- Pyaudio
- Scipy
- time
- Matplotlib



THANKS!

Does anyone have any questions?