SoundProcessor 3000

April 29, 2019

CSCI 250 - Building a Sensor System
Capstone Final Report
SoundProcessor 3000

1 Group

List all members and indicate specific project responsibilities:

Chandler Mitchell

`Hardware Design and Testing`

Evan Mele

`Software Design and Testing`

2 Description

The SoundProcessor3000 samples audio, either from a microphone or a 3.5 mm TRRS audio jack. This data can then be saved, displayed, processed, or analyzed with our program. Examples of analytics include smoothing, dominating signal percentages, and general statistics.

3 Hardware

3.0.1 Sensors

3.5 Audio Jack
Sound Detector
3.5mm TRRS Breakout Board
Used to gather sound data from a 3.5mm aux cord
Sound Detector
Used to gather sound data from the enviroment
Push Buttons
Used to signal when to start recording, stop recording and exit the program

3.0.2 Interface

- 2x ADC
  - Used to Conntect the analogue sensors to the digital input of the Raspberry Pi

4 Operation

The program is operated in the console window with a text based menu system by using the number keys. There is always a previous menu option available. A button is used to start/stop data sampling. Another button is used to close the program at any point. Data read from external files must be in the .npz format (the same format that the program saves data as). All graphs generated by the program are saved to .png files.

5 Data

As stated in the capstone guidelines notebook, your project must have 4 data components:

1. Collect data
2. Process data
3. Display data
4. Interpret data

Include here a description of the data you collected, what processing was done to it, graphs/plots/visuals created by your code, and your interpretation/analysis of the data. No need to include raw data or code (unless it helps your explanation).

6 Wiring

7 Code

Our code is partitioned into separate files. We have a file for graphing, analytics, displaying menus, file I/O, reading ADC values, and a main file. - graphing: graphs data from numpy arrays - analytics: processes data and/or computes statistics on data - menus: holds functions which display menu options - file I/O: saves and reads in data from .npz files to numoy arrays and visa versa - reading ADC: contains a class that is used to read data from both ADCs - main: imports all files and other libraries to execute main program function

8 Credits

Libraries: - Numpy - GPIOzero - MathPlotLib

```
In [ ]:
```