Names: Daniel Salazar and Thomas Martin
Word Count: 1,392
Title: Support Vector Classification of Musical Instruments

Abstract

The basic concept of the final project for CMPE 344 was learning and applying the concepts for software to interact with hardware. Group 5 chose to make a Musical Instrument Classification Device using the Beagle Bone Black (BBB) Micro Processor. Several modules were connected to the Beagle Bone Black including an LCD screen, a button, and a mic. Group 5 developed software to interact with the hardware for input and output in real time. The software was programmed in Python 3. Group 5 3D printed a case to house everything. The Machine Learning model had an accuracy of 92% to 97% and was overfit for the clarinet.

Introduction

CMPE 344 was designed to instruct students how to use software to interact with hardware. We learned about Python programming on the BBB using Adafruit_BBIO. The first assignment was to program all the hardware on a Bacon Cape. The second assignment was to take a selfie with the LCD Cape and Camera Cape. The third assignment was to design and develop a hardware and software solution to a problem.

For Group 5's final assignment they chose to do a project based on Machine Learning. Dr. Doering's passion for music inspired them to make a Musical Instrument classifier. Following a failed attempt at making a custom condenser mic and preamp circuit, Group 5 went through multiple hardware revisions.

Group 5's main goal was to record and classify an audio sample. They found multiple research projects, all of which hinted that Support Vector Classification was effective. One of those projects found a 99% accuracy using the 'rbf-kernel'. The music instruments they chose were based on one of the example projects they found: cello, clarinet, flute, guitar, saxophone, trumpet, and violin.

Group 5's stretch goal was to 3D print a case for the BBB that could hold the LCD display at a 90-degree angle.

Methods

The main program, startup.py, connects to the USB mic and prompts the user to press the button to start recording. When the button is pressed, an audio stream is opened up and the user is prompted to press the button to stop recording. The program will record audio samples into a buffer until the button is pressed. After the button is pressed a second time, the file is filtered, processed, and converted to a .WAV file. Then it is used as input into a Support Vector Classification model. The resulting instruments are displayed on the LCD screen.

Hardware:

- Beagle Bone Black.
  - Micro processor
- 1602A LCD display module
  - Pins used for LCD:
    - LCD Data = P8[12, 14, 16, 18]
    - LCD 5V = P9[8]
    - LCD GND = P9[2]

- Tactile Button
  - Pins used for Button:
    - Button Data = P8[17]
    - Button 3V = P9[3]
    - Button GND = P9[1]
- a USB Mic
  - Condenser Mic

Stretch Goal:

- 3D Printing

Software:

- Input
  - Adafruit_BBIO – LCD Cleanup and Button Event Handling
  - Librosa - Music and Audio Analysis
  - wave – Save Audio Data as .wav
  - datetime – Naming audio samples with current date
  - pyaudio – Recording Audio Samples
- Output
  - Adafruit_CharLCD – Displaying text
- Prediction
  - Scikit Learn - Support Vector Classification
  - numpy – calculating prediction accuracy for audio sample
  - pickle – saving SVM model into a binary file
  - time.sleep – pausing to display results

Results

Our model had a 92% to 97% accuracy on the test data sets. Image 9 shows the testing audio sample compared to our audio recordings of the testing audio sample. Using a low-end mic resulted in poor quality audio samples which reduced our model's capability to predict accurately. The model was overfit to the clarinet and resulted in a misclassification of clarinet for other instruments.

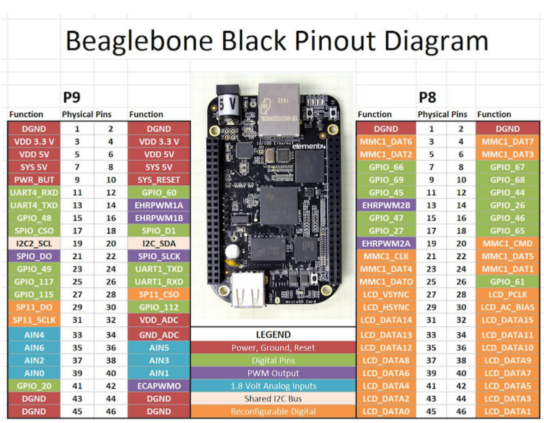

Image 0: The Beagle Bone Black and its GPIO Pin Input/Output.
This details the Beagle Bone Black's GPIO pins. I used this map to determine which pins to target in our software, in order to control the connected hardware. We reserve P9 pins for power and ground and P8 pins for data input and output.

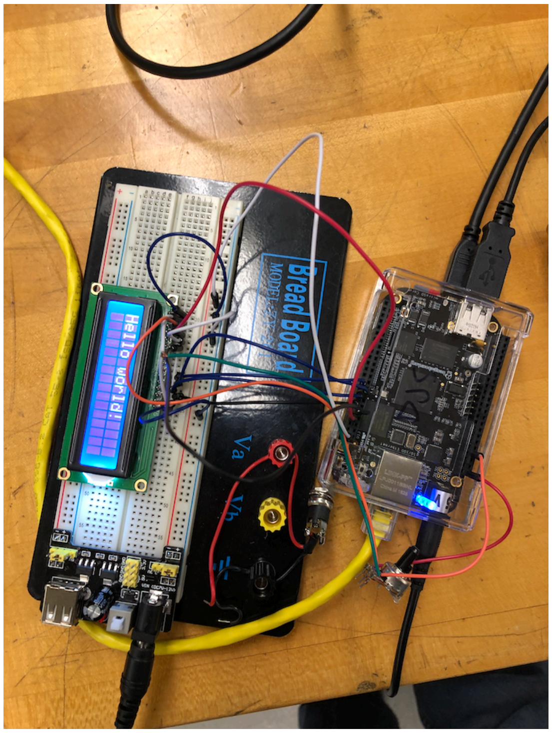

Image 1: LCD and Potentiometer
This is the Music Box’s LCD prototyping. I used a breadboard to test the LCD software.

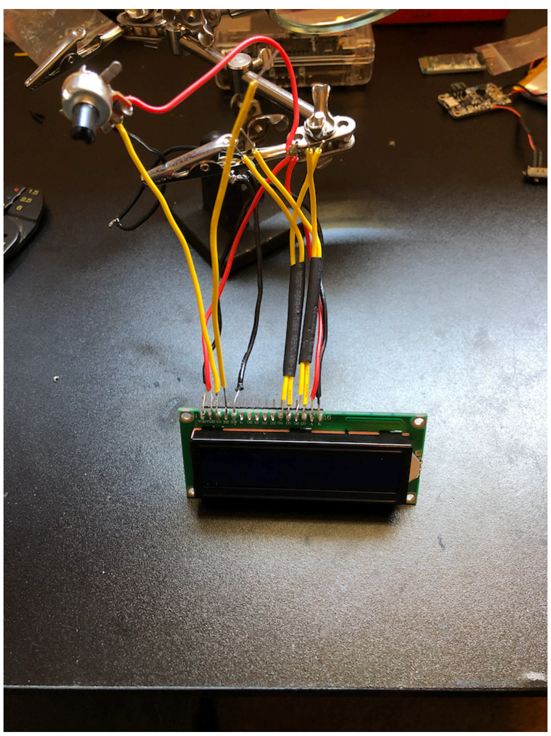

Image 2: LCD and potentiometer Wiring – 6 data pins, 2 5V pins, 3 GND pins
Once I had a successful test run, I soldered the potentiometer to the LCD

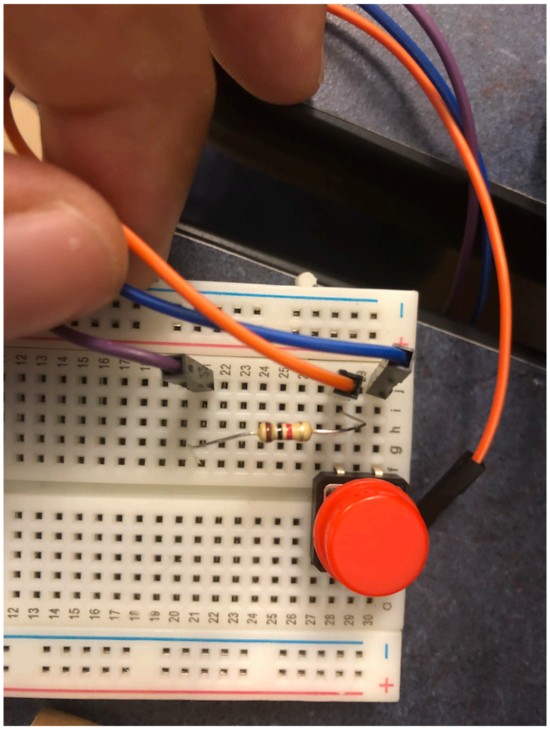

Image 3: Button Prototype - 1 data pin, 1 5V pin, 1 GND pin
This is the Music Box's Button prototyping. A breadboard was used again to test the Button software.

Image 4: AmazonBasics USB Microphone – mini-USB to USB I/O
This is the USB Mic that we used to record audio samples. It is connected to the Beagle Bone Black with a mini-USB to USB.

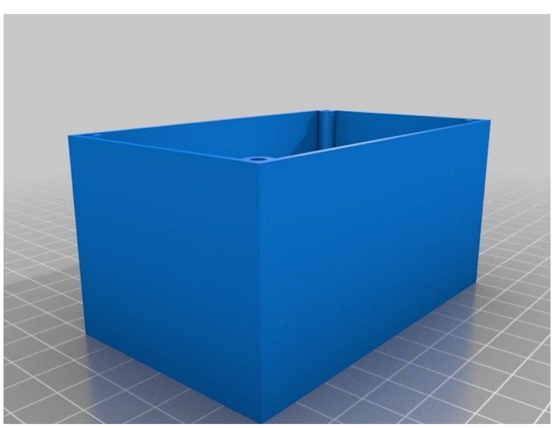

Image 5: 3D Printed Case
This is the 3D Printed Case I got from Thingiverse. I printed one normal and another one with the

LCD window sliced off. LCD was secured to the front part with 4 screws.

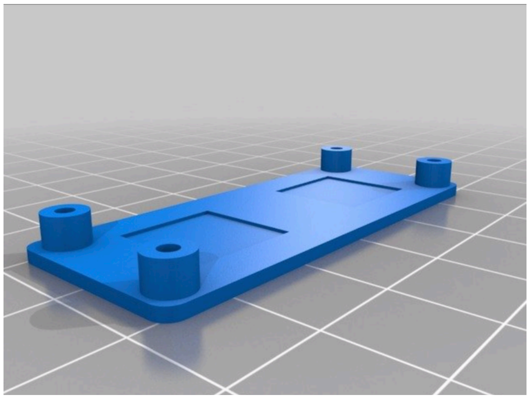

Image 6: 3D Printed Button Bridge
This is the 3D Printed Button Bridge I got from Thingiverse. The button was glued onto this. This was secured to the top of the front part with glue.

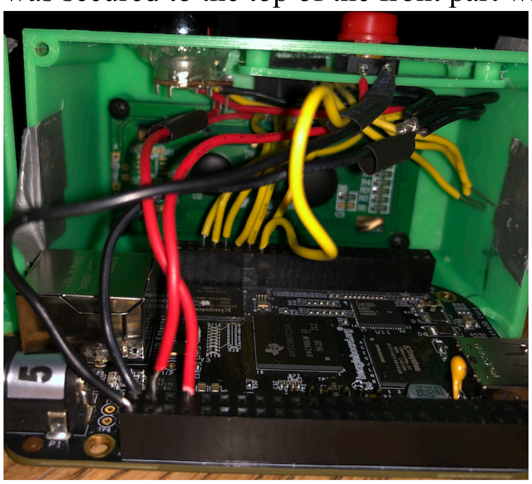

Image 7: Version 1.0 – The Music Box (inside)
This is the final wiring of the hardware

Image 8: Version 1.0 – The Music Box (outside)
This is the Music Box with the case closed and the program classifying an audio sample

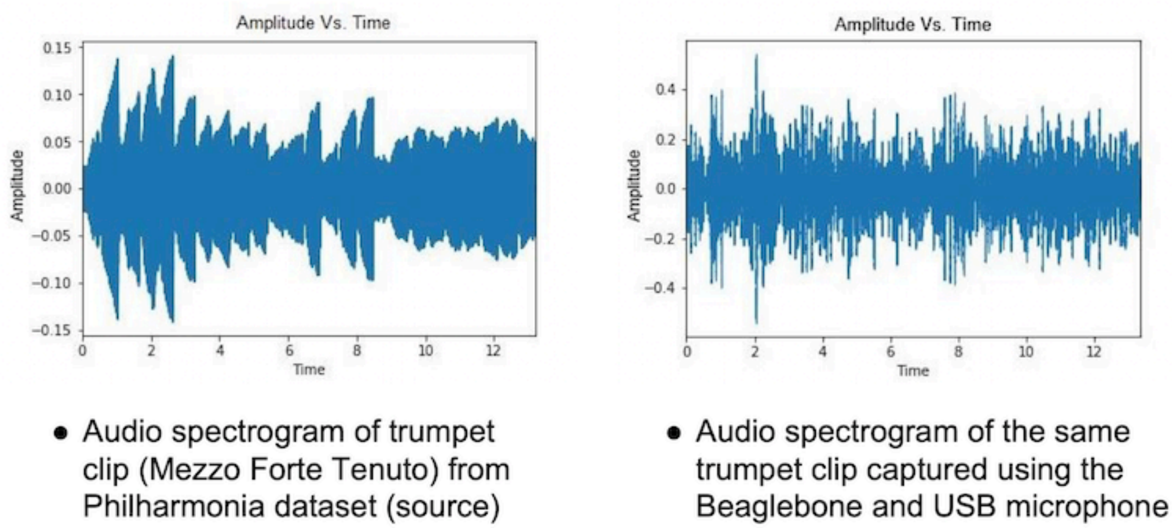

Image 9: Comparison between the test data and the recorded audio sample

Category	Part Number	Part Name	Qty	Unit Cost	Total Cost	Notes
Mic	B078GYH477	AmazonBasics Portable USB Condenser Microphone	1	$15.86	$15.86	
Button	1009	Round Tactile Button Switch	1	$0.39	$0.39	Adafruit sells a 15 pack for $5.95
Microprocessor	1996	Beaglebone Black	1	$55.00	$55.00	
Screen	LCD1602	LCD1602 Module	1	$2.95	$2.95	
Case	Custom	3D Printed Case	1	$0	$0	I made this at home
Case	Custom	3D Printed Button Bridge	1	$0	$0	I made this at home
				Total Cost	$74.21	

Image 10: Bill of Materials
This is how much each part costs, including the total price to build this yourself.

Discussion

Group 5 was successful at predicting multiple instruments using their custom device. They approached the design process using an Agile Methodology. At the beginning of the semester thier goal was to construct a custom condenser microphone that would be able to record live music for classification. In addition, their initial proposal included the development and testing of a variety of machine learning models created with both supervised and deep learning, but during the course of the semester a number of their designs changed. The custom condenser microphone was removed and replaced with a pre-made USB condenser microphone. Also, the development and testing of multiple machine learning models based on different algorithms was scaled down to only a single machine learning model implemented using support vector machines. Additionally, a TensorFlow SVM could have been used on a computer with a GPU and a larger dataset, and then save the model on the BBB.

Citations

Software Libraries

- Adafruit_BBIO – https://github.com/adafruit/adafruit-beaglebone-io-python
- Librosa - https://librosa.github.io/librosa/index.html
- wave – https://docs.python.org/3/library/wave.html
- datetime – https://docs.python.org/3/library/datetime.html
- pyaudio – https://pypi.org/project/PyAudio/
- Adafruit_CharLCD – https://github.com/adafruit/Adafruit_Python_CharLCD
- Scikit Learn - https://scikit-learn.org/stable/modules/svm.html
- numpy – https://www.numpy.org/
- pickle – https://docs.python.org/3/library/pickle.html#module-pickle

- time.sleep – https://docs.python.org/3/library/time.html

Example Projects

- MatLab and sklearn - https://github.com/ArushiSinghal/Automatic-Instrument-Identification
- Librosa and sklearn - https://github.com/IvyZX/music-instrument-classifier
- Data Set - http://www.philharmonia.co.uk/explore/sound_samples

3D Printed Devices

- 1602A LCD Case - https://www.thingiverse.com/thing:1873666
- Button Bridge - https://www.thingiverse.com/thing:1277483

Appendix

23-gauge wire was used to interface the hardware with the GPIO pins.