Group 1

Physics 398DLP

Determining Wind Speed with Sound

Matt Massari, Nathan Smith, Mina Mazeikis, Riley Corrigan, Lele Wang

April 30, 2021

**Abstract**

The purpose of our experiment is to explore the feasibility of measuring wind speed and direction through sound. Our experiment is centered around our circuit board, which is composed of four microphones and a speaker. The board is constructed such that each microphone is equidistant from the speaker. The board includes an Adafruit DS3231 Real Time Clock (RTC) which, when turned on, causes the speaker to emit a periodic, but somewhat jagged, kHz tone. Our group will determine wind speed and direction by looking at the offsets in arrival time at each microphone. Offsets will be determined through a series of calculations that will be made using a series of code that lies in the brain of our board, an Adafruit Feather M4 Express. The code will digitize the sound each microphone hears and attempt to “slide” the digitized waves so that the digitized wave from each microphone lies on top of each other. The amount of “slide” per digitized wave will be calculated and will be used to determine the time offset of sound arrival between microphones.

**Introduction**

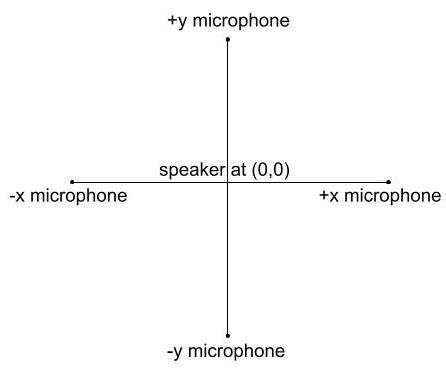
Purpose:

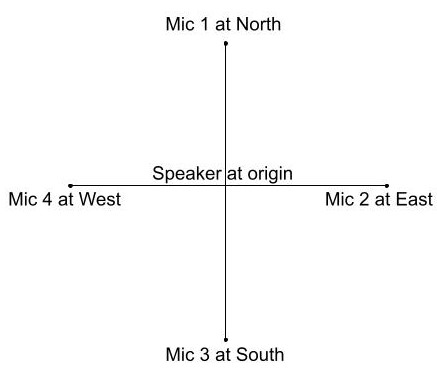
Our intent in exploring the effects of wind speed on sound is to build a machine that sound engineers at outdoor concert venues can use to determine wind speed and direction. Ideally, a setup of our final product would consist of multiple boards which would be placed at various points throughout the venue. Theoretically, local wind speed and direction could be determined at each “field station,” which would create a comprehensive wind vector map for a sound engineer. With this data set, engineers could adjust speaker phase accordingly and improve sound quality for listeners. When wind speeds vary at different sections of an outdoor venue, sound waves are shifted in time and phasing issues occur. Phasing is the timing difference caused by combining similar sound waves (*Nicholas*). Phasing issues can be caused by a variety of things. In our case, phasing issues are due to wind. Depending on the direction and speed of the wind, the timing of when a particular sound arrives at a destination can change. This leaves areas of a venue with uneven loudness and awkward arrival time of different sounds.

Sound in Air:

Sound is a wave that propagates through a medium. The source of a sound creates a sound wave by producing variations in air pressure, which travel outwards from the source in the form of a longitudinal wave, which is a wave that moves in the same direction it vibrates. Sound wavescan be described by their amplitude, frequency, and speed. The speed of a sound wave varies depending on the medium it is traveling through. The speed of sound in air is approximately , but this value changes depending on temperature. For this report, we will assume a standard temperature of degrees Celsius, and a corresponding speed of sound of (*Canadian Center*). A sound speed of means a pressure change maximum will move meters in the direction of propagation in one second. The amplitude of a sound wave is directly proportional to the loudness of the sound squared, meaning amplitude is proportional to the maximum pressure change (*Canadian Center*). Frequency of sound is measured in hertz and is the measure of how many waves will pass a single point in a second. Frequency corresponds to pitch of a sound. When listening to a wave with a high frequency, one would hear a high pitch sound. Typically, humans can hear sounds within the frequency range of Hz to Hz. Shown below in equation (1), frequency and speed are related to another quality of a wave, the wavelength, represented by the Greek letter (*Canadian Center*).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | (1) |

Effects of Wind on Sound Speed:

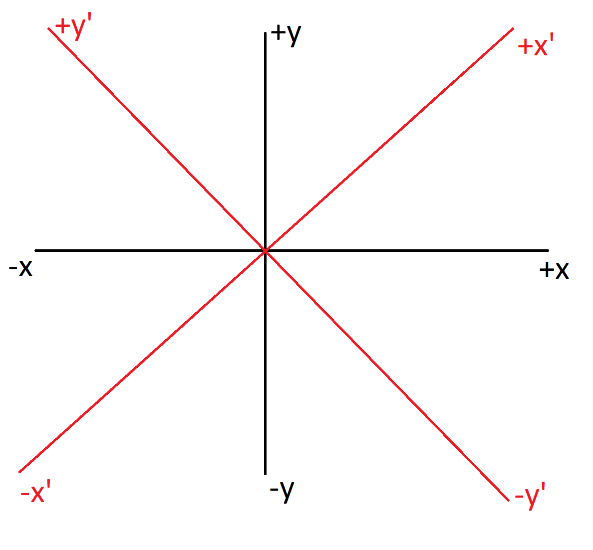
In still air of degrees Celsius, sound travels with respect to ground at . This means that a sound played at a speaker will arrive at a microphone based on the distance divided by the velocity, in this case . However, a gust of wind might blow during the recording. In that case, the velocity of sound would be added to the velocity of the wind. This would cause the sound to reach the microphone either faster or slower than the we expect.

**Figure 2:** This diagram shows the microphones with respect to their position on a graph.

**Figure 1:** Diagram of the microphone layout.

Consider **Figure 1**. One microphone is placed at and another placed at . If the velocity of the wind is in the direction, then the sound will travel slightly faster towards the microphone than the microphone. If a tone is played at the origin, it will reach the microphone slightly sooner than the microphone. This difference in arrival time creates a measurable time offset which we will use to determine wind speed.

Geometry and Microphone Pairings:

An important aspect to the calculation of wind speed and determination of precision of our board is our six microphone pairings. We have four microphones on our boards that we collect data from, and for our data analysis, we use a set of six microphone comparisons.

**Figure 3:** Diagram of unprimed versus primed coordinated. Coordinates are determined using microphone comparisons.

To find the vector for wind direction and magnitude the following equations were used:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | (2) |
|  |  |  | (3) |

Where is the bin offset magnitude in the prime frame shown in **Figure 3**. The original frame’s offset magnitude is represented by . Then the following represent the different bin offsets of each microphone comparison: and are in the direction, and are in the direction, is in the y-direction, and is in the x-direction. The average of these two vectors is calculated to solve for a more accurate overall offset magnitude. The angle from the positive y-axis is calculated with the following equations:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | (4) |
|  |  |  | (5) |
|  |  |  | (6) |

Where is the component of the vector and is the component. From the total bin offset magnitude and angle, the speed and direction are derived. The quadratic equation used to solve for wind speed is below:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | (7) |

Where is the overall time offset calculated from the overall bin offset magnitude. The distance from the speaker to a microphone is d and the speed of sound in air is .

Measuring Offset:

For our experiment, we play a jagged, periodic wave from our speaker and record it at four different microphones. Each microphone will record this same wave; however, all the values will be shifted along the x-axis based on the offset due to wind speed. In still air, the velocity of wind is zero. This means the wave should look the same at all 4 microphones, as the sound from the speaker will arrive at the same time. If wind is present in the +x direction, the two x-axis microphones will be affected as discussed above. This shift in the recorded wave can be compared between the other microphones, and an offset can be determined.

In order to measure the time offset, we record the values each microphone heard at multiple time intervals. The microphones record nearly simultaneously and record 80,000 times a second. By plotting the values read by the four microphones, we are able to generate plots of ADC count values vs. time. These plots demonstrate periodic waves and can be used to interpret the microphones reading of the 1 kHz tone. To compare the two waves, we run a chi-squared test based on the assumption of various offsets. For each comparison, we take each time interval and find the difference in the squared values of each microphone comparison. Next, we sum these numbers to find a goodness of fit over the entire interval. The lower the sum of the numbers, the closer the two waves are. In order to correct timing, we add in artificial time offsets to the second wave. Importantly, we are only measuring time intervals, or “bins”. In addition to a time offset of zero, we also try offsets between -10 and 10 bins. Finally, we can compare the goodness of fit values for each time offset and find the best number.

**Equation for the chi-squared goodness of fit value:**

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

sums over all the times in 10,000 bins

is the microphone’s recorded pressure value for that bin

is the first microphone’s wave given a bin number

is the second microphone’s wave, with added offset

It is possible that the true time offset falls between one of these intervals, and thus isn’t properly represented in our data. In order to correct for this, we create a parabola based on the time and values of the three lowest goodness of fit points. This curve is created in order to minimize our chi-square value. A parabola can be drawn between any three points on a plane. By using the best point and its two neighbors, we can come closer to what we think the true time offset is. This time offset lies at the minimum value on the goodness of fit parabola. By simply finding the minimum value, or apex, of this parabola, we can determine the true time offset between microphones. This whole comparison is done a total of times, in order to compare each microphone to each other. These comparisons are explained in the geometry section above.

One problem that exists is that microphones can only measure a limited number of times per second. If the wave the speaker emits has too high a frequency, the microphones will measure too few values per period. During analysis, our added offset could be too high for how few values we might have. This could cause the comparison to artificially move an entire period in time and return an incorrect time offset. To prevent this, we use a kHz wave, which measures around bins per wavelength. In addition, we limit the time offset to bins ( microseconds), which will prevent the offset function from moving an entire period.

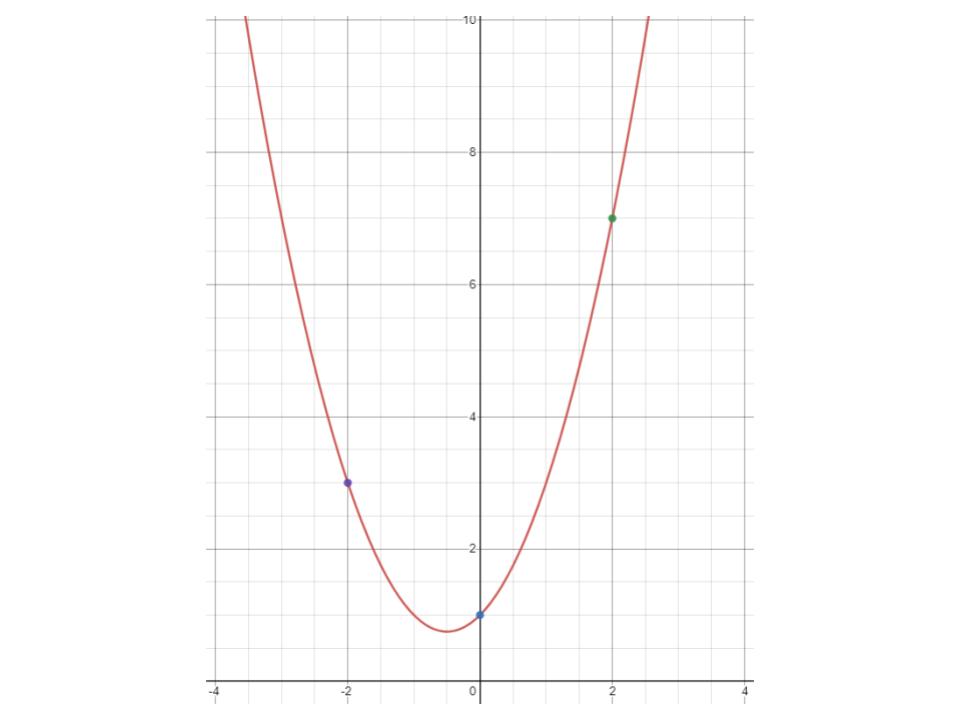
**Equation for Parabola Vertex:**

|  |  |  |
| --- | --- | --- |
|  |  | (9) |
|  |  | (10) |
|  |  | (11) |
|  |  | (12) |

Vertex is

Where , , and are the three points

This equation is a mathematical process to calculate the x and y coordinates of the vertex of any parabola. We make this parabola with three points: , , . We choose the points with the lowest chi-squared value. We can simply input these values into the equation above, calculate A, B, and C, and then calculate the x and y values of the vertex (*Z, David*). The x-value of the vertex will be the true time offset.

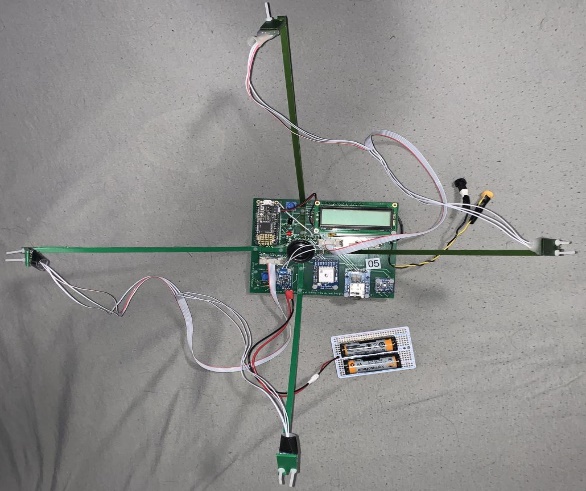


**Figure 4:** This is an example of a parabola constructed from 3 points. Measurements are arbitrary and used as demonstration.

Trials:

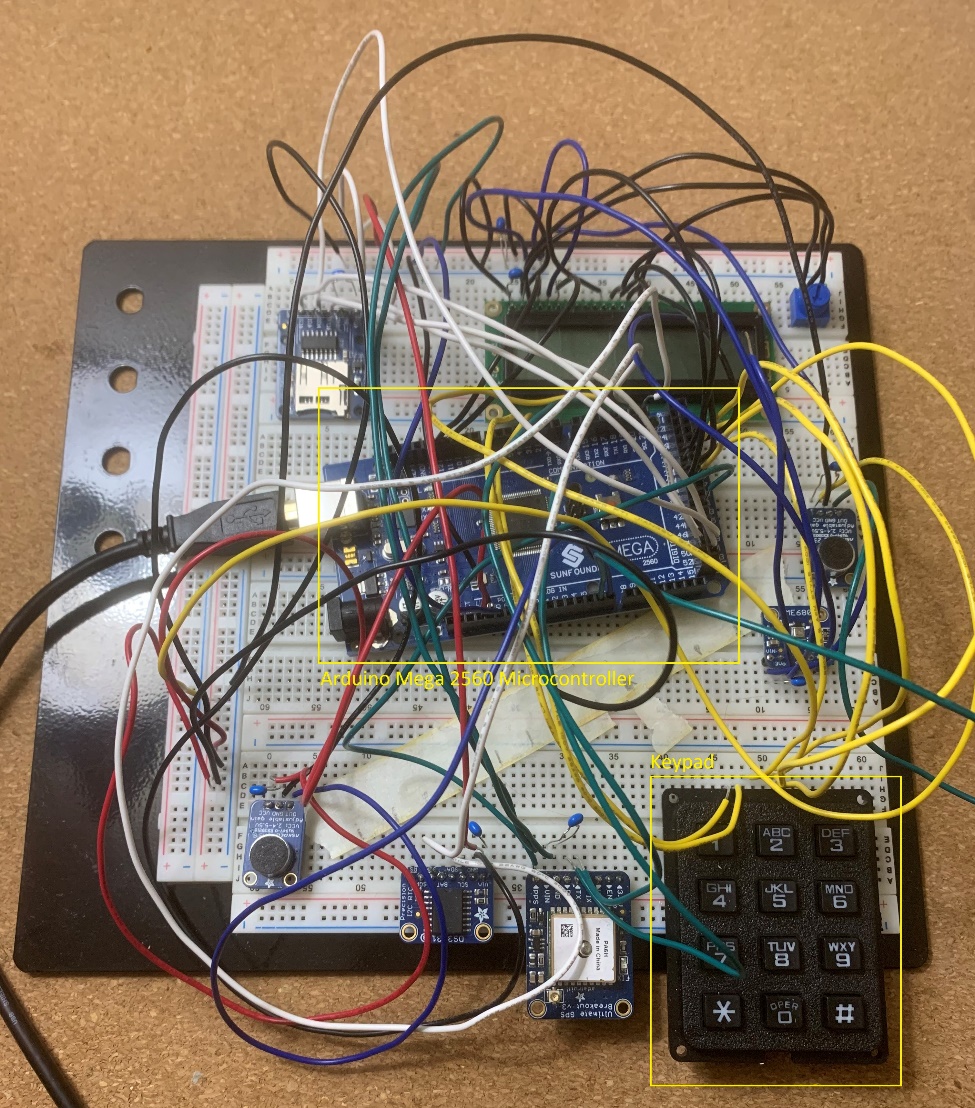
One trial using this device consists of one iteration of the code inside of the microprocessor. During a trial, each microphone is read 10,000 times, the speed per 4-ADC read is calculated, and our offsets between microphone pairings are calculated and analyzed. Each trial outputs a wind speed, direction, and precision to the device’s LCD screen, while also printing the offset calculations and statistical analysis in the device’s serial monitor. Our methods for measuring accuracy of the devices often require running multiple trials.

Statistical Analysis:

For each trial, we run several offset calculations. Initially, we analyzed the entire bin wave data we collected from each microphone. We did this for each of the six comparisons. Next, we sliced the bin data into “batches” of smaller bins, around or less bins per batch. We then performed the same calculations with these smaller batches. By finding a mean value for the comparison, we can finally calculate a standard deviation of the batched mean. The standard deviation will give us a time offset and a corresponding precision value, which are the mean and standard deviation of the bin sample, respectively. By finding these values, we can determine how accurate the smaller batches are in comparison to the total time offset. In doing so, we can analyze these smaller bin sizes to calculate time offsets faster than using the whole bins, within a degree of relative precision.

**Figure 5:** The most current version of our machine. Prominently featured are the four arms that hold microphones an equal distance away from the center speaker. See Figure 9 for labeled diagram.

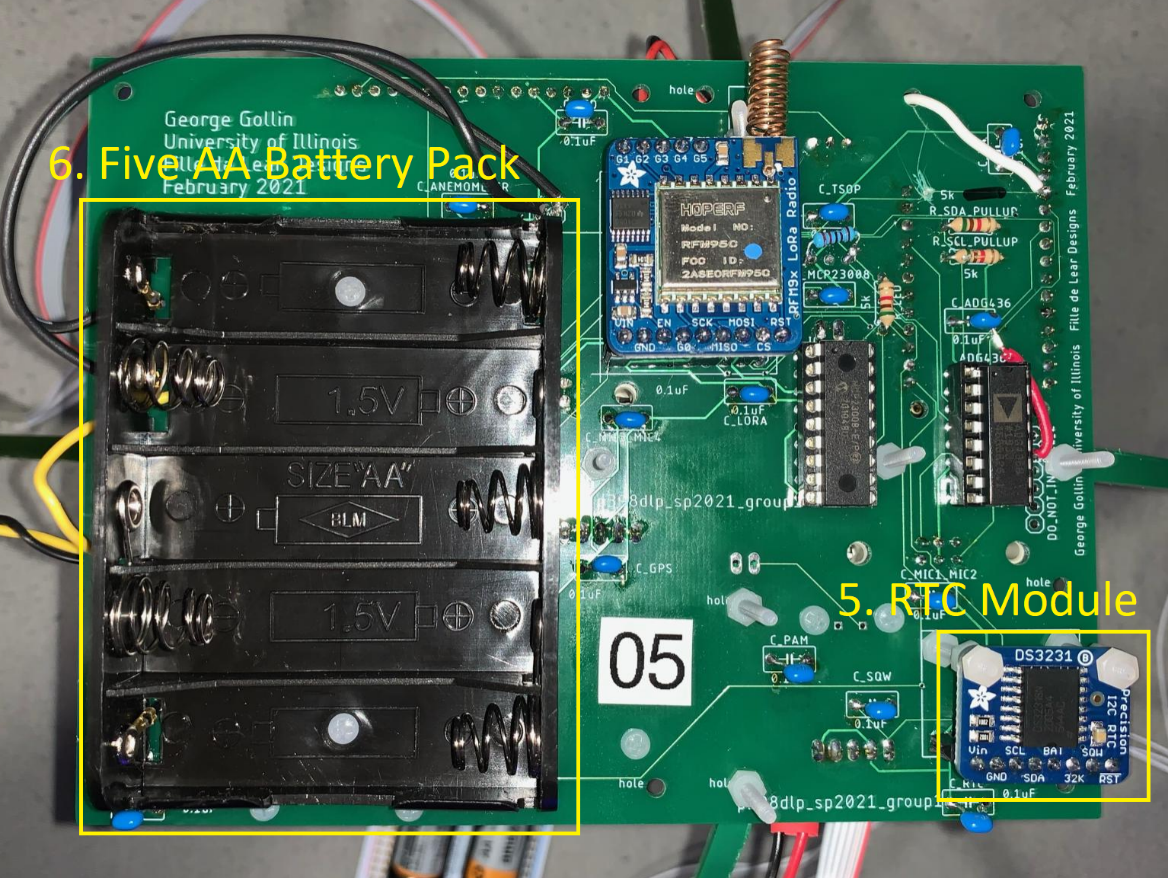
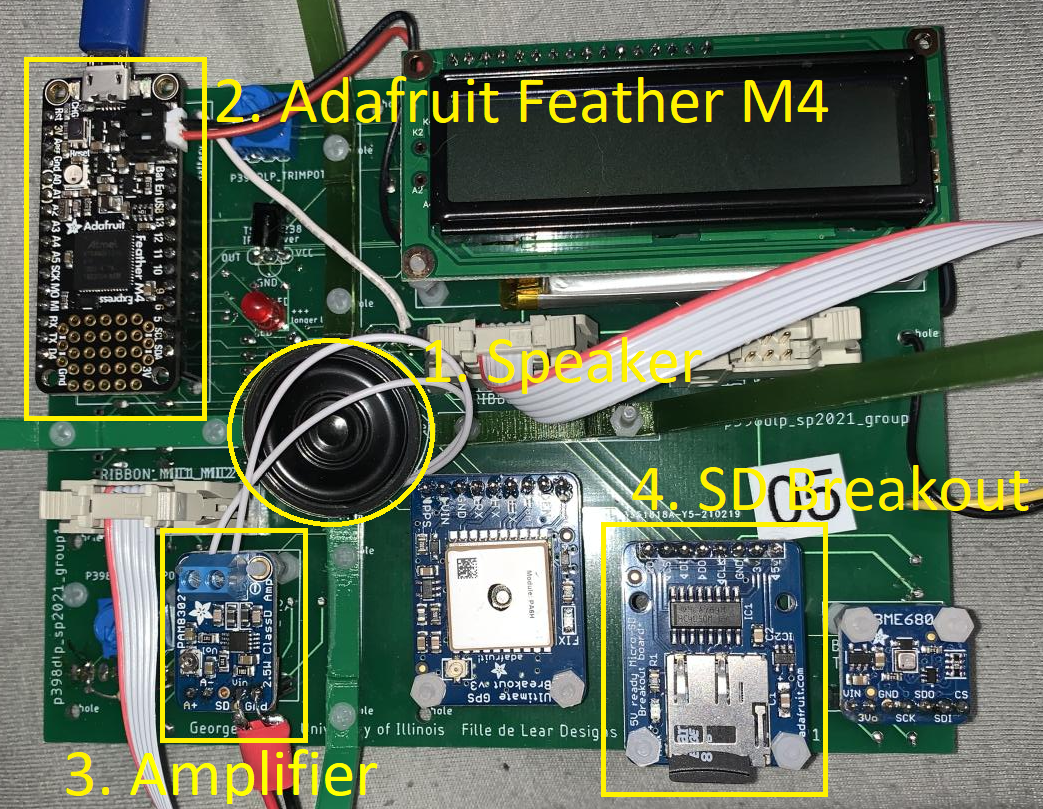
**Experiment**

Our Board:

Our setup is a custom printed circuit board built with our experiment in mind. As seen in **Figure 6**, the design process began with prototyping potential circuits on a breadboard and drafting functions in Python. The most current version of our board, drafted and built by our professor, George Gollin, can be seen in **Figure 5**. The main components of the board include four Adafruit Max 4466 microphones connected via ribbon cables and one tone emitting speaker connected to an Adafruit PAM8302A Class D amplifier and an Adafruit DS3231 RTC module.

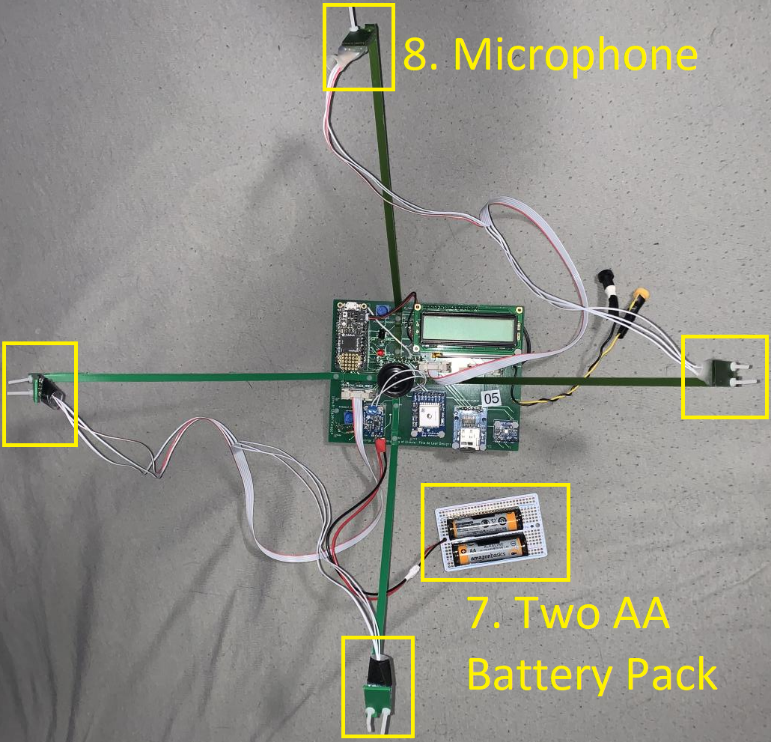
**Figure 6:** An early breadboard prototype of our machine. Note the keypad and the Arduino Mega 2560 Microcontroller. Both components were removed in the final version of the board.

A Class D amplifier takes an analog signal and converts it into a digital signal (*Infineon*). Class D amplifiers rapidly switch output devices, in our case a speaker, on and off to create this signal (*Munz*). Our PAM8302A Class D amplifier is connected to an external battery pack that gives the amplifier power separate from other components of the circuit. An RTC module is used to measure the passage of time (*Digikey*). On our board, one of the outputs of the RTC module is configured to make a kHz square wave which is then processed by the amplifier. The microprocessor our board uses is an Adafruit Feather M4 Express. The M4 was chosen because of its quick processing speed and its onboard file system. Other components of our board include a Liquid Crystal Display (LCD) with an external battery underneath to power the M4 when it is not connected to a computer, an RFM9x LoRa Radio transmitter, a Global Positioning System (GPS) module, a BME680, and a micro-SD breakout board. The LCD display is used to display the final calculated wind speed and direction. If the user did not have immediate access to the machine, future revisions of our machine could use the LoRa Radio transmitter to send the final wind speed and direction to a base station. In future revisions, the GPS module could be used to determine where in the venue boards are placed. The BME680, which measures atmospheric conditions, could be used to help determine a more accurate expected value for the speed of sound in still air. Since the speed of sound in still air is dependent on atmospheric conditions, the BME680 will be very useful in fine-tuning measurements in future revisions of our machine. **Figure 7** and **Figure 8** display labeled pictures of our most current board.

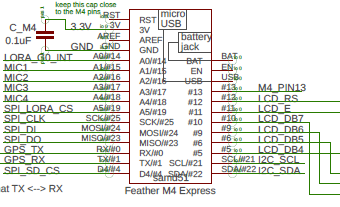


**Figure 8:** The bottom side of the most current version of our board. The labels in the image are as follows: 5. Real-time clock, 6. Battery pack for powering cup anemometer.

**Figure 7:** The top side of the most current version of our board. The labels in the image are as follows: 1. The central speaker, 2. The Feather M4 Processor, 3. The PAM Amplifier, 4. The Micro-SD breakout board.



**Figure 9**: Overhead shot of the most current version of our board. The labels in the image are as follows: 7. Battery pack for powering PAM Amplifier, 8. Microphone.

The loudspeaker labeled in **Figure 7** is used in conjunction with the amplifier and Real-Time Clock (labeled 5. in **Figure 8**) to produce a periodic wave output. **Figure 8** also shows a 5 AA external battery pack which is for powering an external cup anemometer to check our results when taking measurements in the field. The 2 AA external battery pack seen in **Figure 5** is the external power supply used to power the PAM8302A amplifier. Next to the amplifier is an attenuator which allows us to change the resistance between the amplifier and the RTC. The lower the resistance, the louder the sound wave. Most prominently, four arms, or struts, can be seen holding each of the microphones in **Figures 5 & 9**. Holding the microphones equally distant allows for greater accuracy. The distance from each microphone to the speaker is cm, which allows for a large enough distance for potential wind to be able to create a measurable offset.

**Figure 10:** The Feather M4 processor is faster and has more storage capacity than other potential processors, but also has fewer pins.

The M4 Feather processor is the brain of the board, it is where our code is uploaded and executed. Connected to the processor is the Micro-SD breakout board which stores our recorded data and has the ability to read data that has been previously stored on it. The Feather M4 has ADCs and is able to run of them at once. It has bits and a reference voltage of .

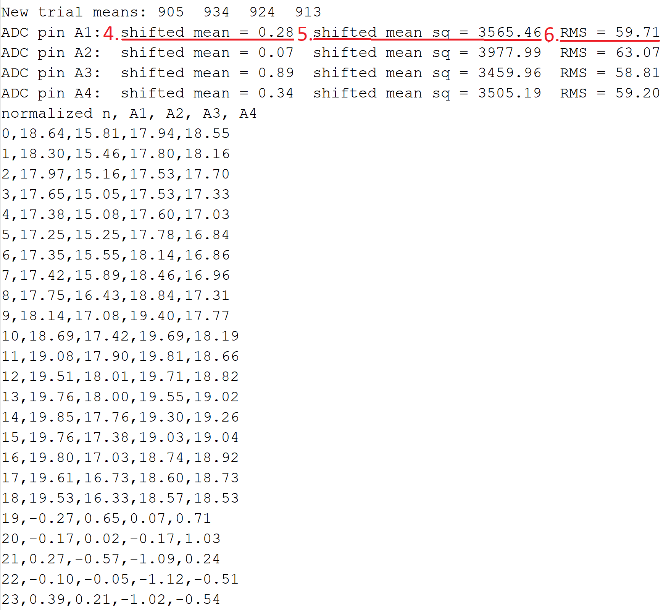
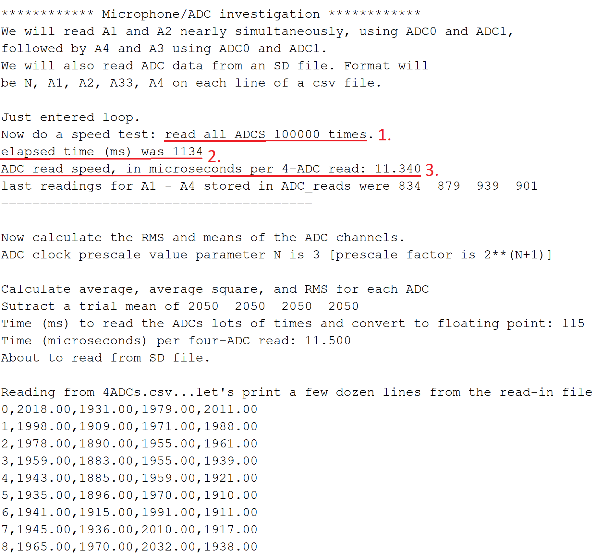
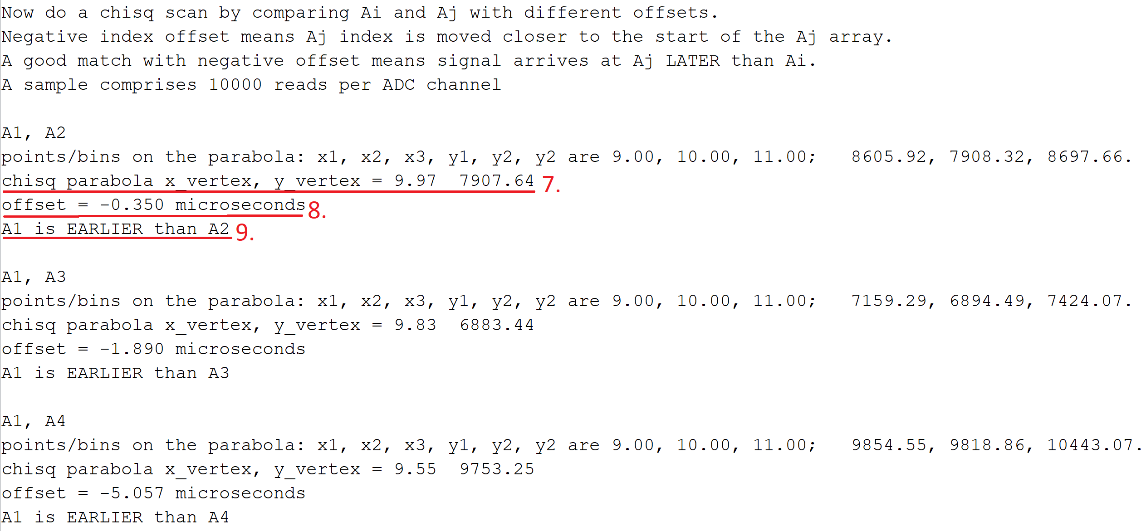
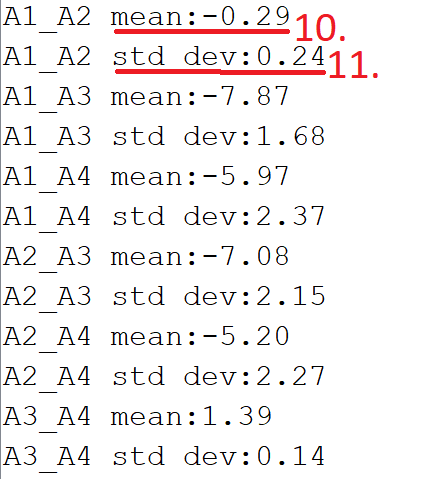
Our Code:

For our machine to be self-contained, the Feather M4 processor is programmed to communicate with each component of our circuit board and perform all the necessary calculations with no outside interference or directions. The code our board executes was originally written by our TA, Ivan Velkovsky, with further revisions being made by our professor, George Gollin, and members of our group. The code placed in our processor is designed to read the four microphones and store the measurements in four different arrays. It was written in C++ and features several key elements. First, it records from each of the four microphones in alternating pairs of two. It will make one measurement of microphone and microphone , and then it will make one measurement of microphone and microphone . Then it swaps back to microphone and microphone for another set of samples. In total, it will do this times, recording a total of values. Each set of are read at approximately the same time and will later be compared with each other. The code writes these values to the Micro-SD for future analysis. Then, the code runs the several comparisons described in the Statistical Analysis section. The values are printed out so they can be read by a human operator.

Tuning:

On each board, the loudspeaker is held in place with a patch of duct tape. Due to discrepancies in the loudspeaker positioning on our boards, the devices may measure offsets between microphones in still air. When taking a measurement with wind, we use a tuning method to compensate for the still air offsets. Three to five trials are performed in a quiet, wind-free environment. We then find the average offset between each of the six microphone pairings. These values are input into the code as 6 offset variables. When completing a trial with these values entered in the code, these offset variables are subtracted from the calculated offset values. In still air, the board should then give a reading of little to no wind. When tuning, our boards report an average value of mph in still air.

Our Measurements:

Measurements are taken via a series of code that the Feather M4 runs. The code attempts to “slide” the periodic waves received from each microphone so that each wave lies on top of each other and returns offset values in microseconds. As seen in **Figures 11-14**, our board gives us a detailed report from our recordings. The offset for each microphone comparison is given in microseconds, but measurements can easily be converted to bins or other units by using the measured ADC read speed. Shifted mean and RMS values are displayed in the serial monitor. They are used to identify any discrepancies between the sound being received from the microphone.  Later in our development process, we added a precision calculation for each pair of microphones. The precision calculation is performed with standard deviations.

**Figure 14:** Excerpt from serial monitor.

**Figure 13:** Excerpt from serial monitor.

**Figure 12:** Excerpt from serial monitor.

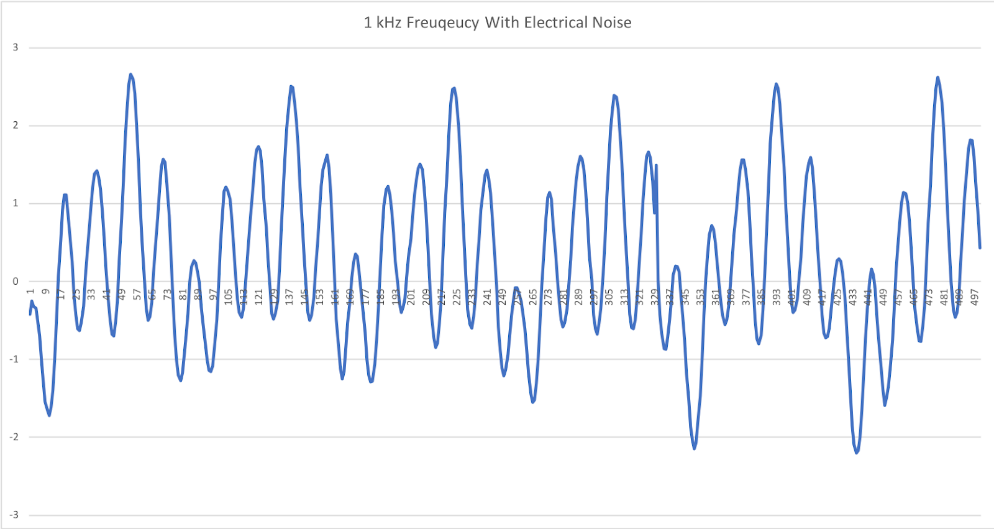
**Figure 11:** Excerpt from serial monitor.

Labels in **Figure 11** represent: 1. ADC counts, which represents the amount of data taken from each microphone, 2. Length of the overall recording, 3. Length of time between each recording from each microphone. In **Figures 12 & 13**, labels 4. through 6. show us how usable our data set is by giving us an average of the intensity recorded by each microphone. That then can be compared between microphones to ensure all microphones are working equivalently. This average is also used to normalize our data collected. Label 7. is the minimum location on the chi-squared parabola and 8. is the calculated offset in seconds. Finally, **Figure 14** displays the output from a batch of size 400. The comparisons for each mic are listed with a mean (label 10.) and standard deviation (label 11). These can be used to calculate the precision of smaller batches.

The average standard deviation of each microphone comparison from three recordings in still air was 3.544. This means that once the offset is over 3.544 microseconds, we are confident that an offset from the wind is being measured. The precision of the wind speed detected by our machine is calculated as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (13) |
|  |  | (14) |

Where equation is the quadratic equation to solve for velocity using the positive root, the speed of sound is , and the component of each microphone’s distance from the speaker is cm. Performing equations (13) and (14), we find the precision of wind speed to be which is mph. Having longer struts on our board would increase the sensitivity of the measurements and would allow our board to measure slower wind speeds. The longer the distance between microphones, the longer period of time wind has to speed up or slow down sound waves. For example, if the distance was twice the length, the speed necessary for confidence would be , exactly half of our current speed necessary for confidence.

When viewing the recorded signal from a single microphone, we received the graph pictured in **Figure 1****5**. Looking at **Figure 15** we see that the kHz signal that was supposed to be produced and received was being distorted by additional harmonic oscillation. This was due to the RTC driving higher frequency outputs. This caused a periodic wave of a much higher frequency to overlap the originally intended frequency. We feared this interference would affect our measurements. In order to check if our data was skewed because of this interference, we compared the data from two microphones. We performed tests on the data for two different situations. One where the microphones were the same distance from the speaker and a second where one of the microphones was closer to the speaker by . The bin offset due to difference in placement, labeled below as , can be calculated using the following equations:

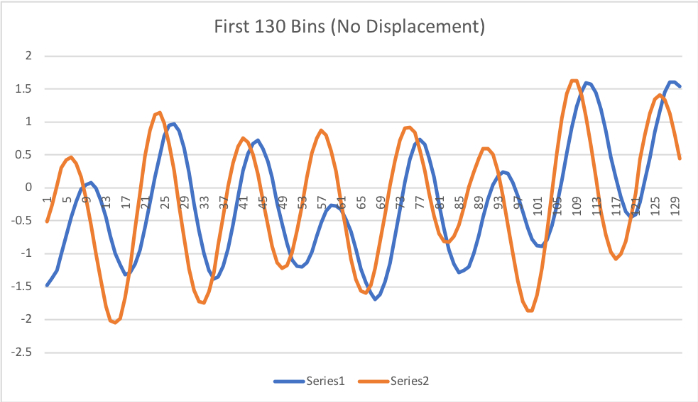
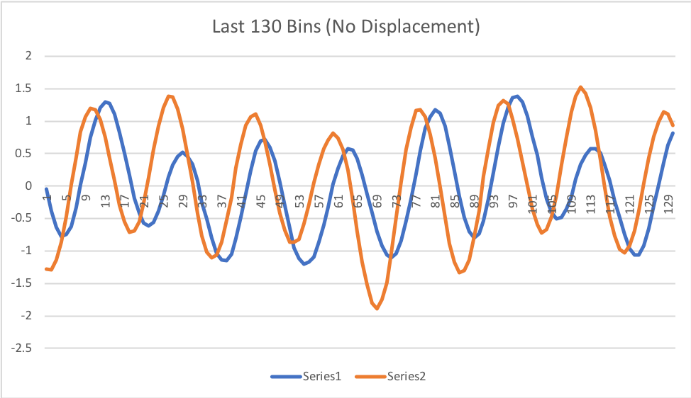
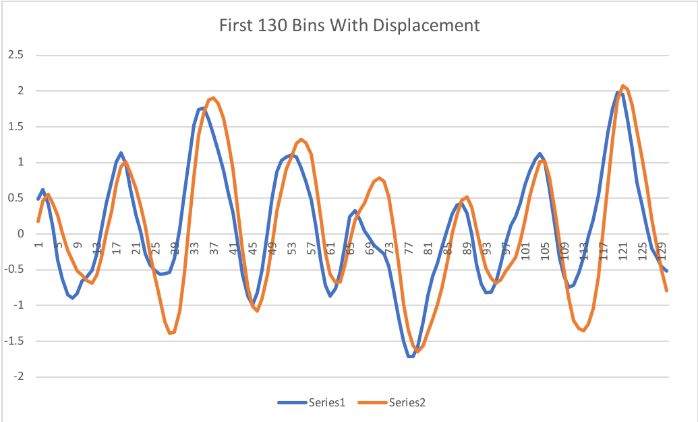
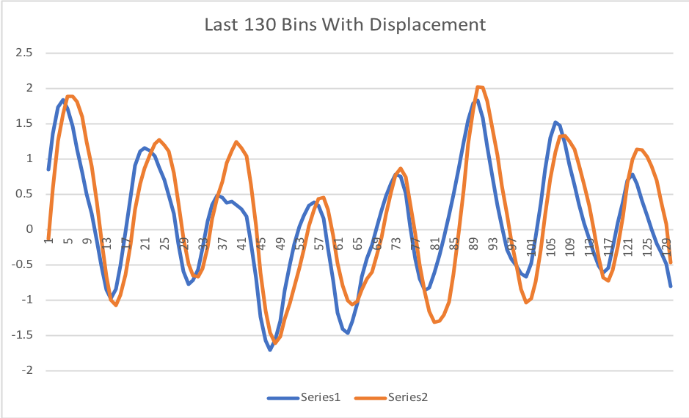
**Figure 15:** Recorded data from microphone 1.

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

Where is the time offset and is the speed of sound in air, which is .

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

Where is the time per bin and is the time offset. Multiplying these two together, we find the bin offset, which typically tends to be bins. By this logic, there should be about a 3 bin offset when comparing our data on a plot. Below, in **Figures 16-19**, one can clearly see the offset of about three each period of the wave.



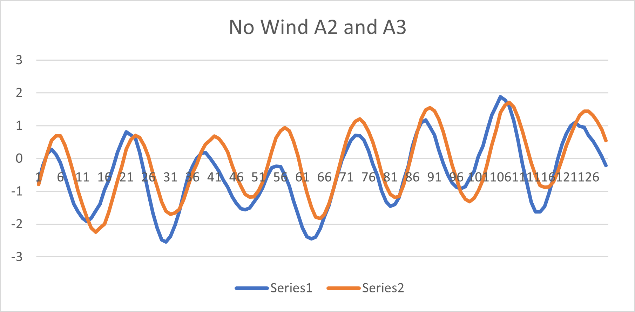
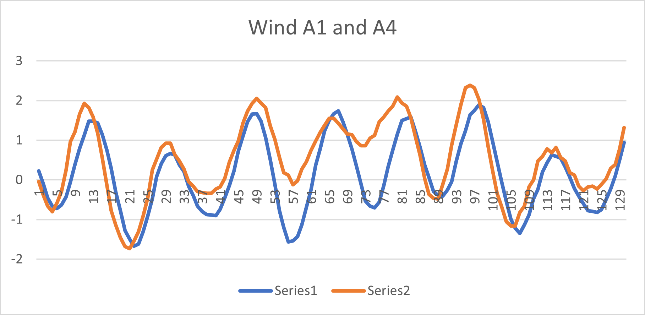
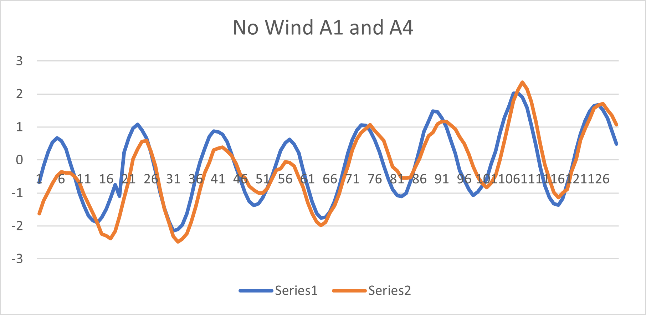
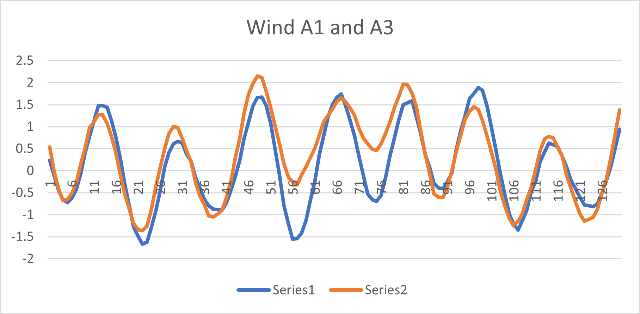
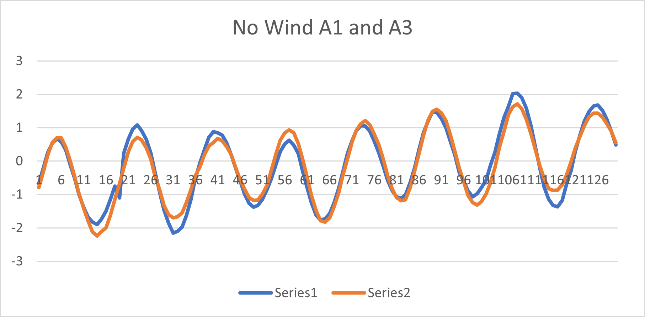
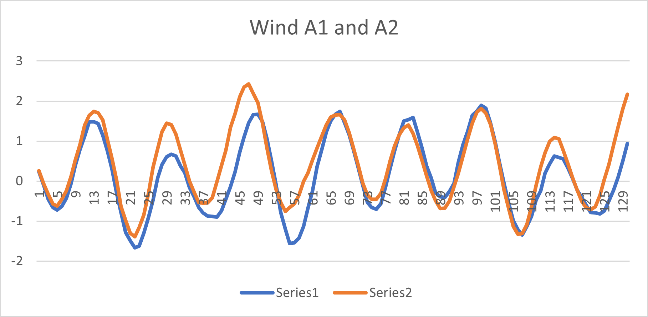
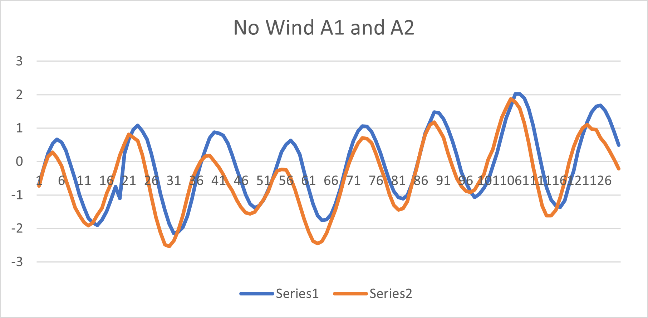
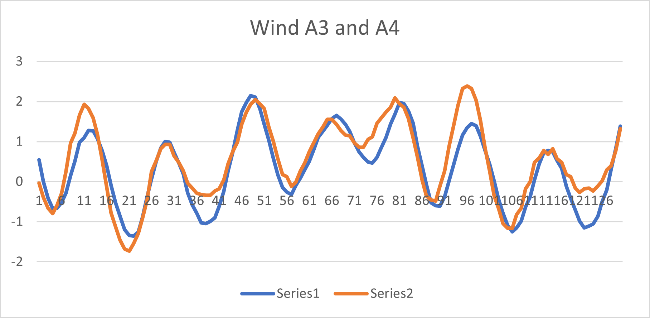
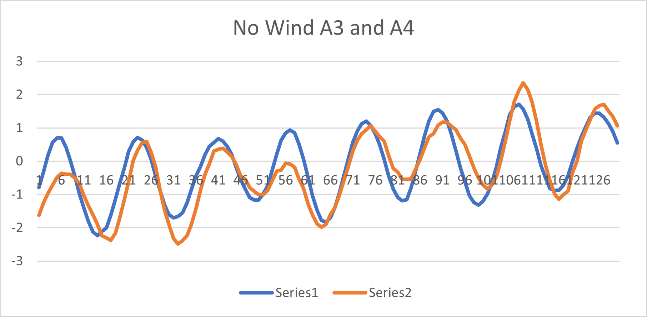
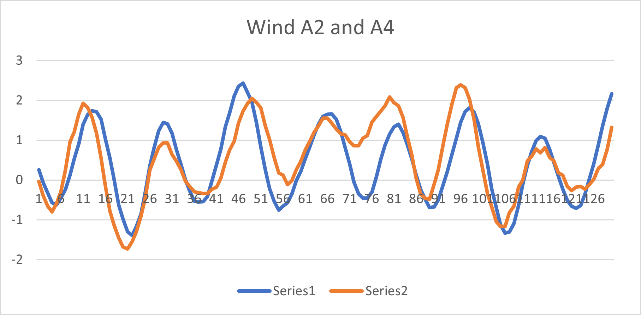
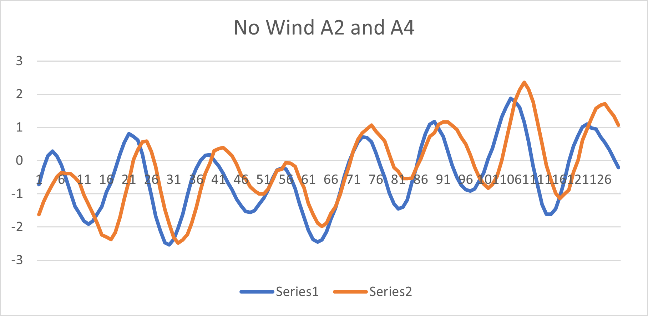
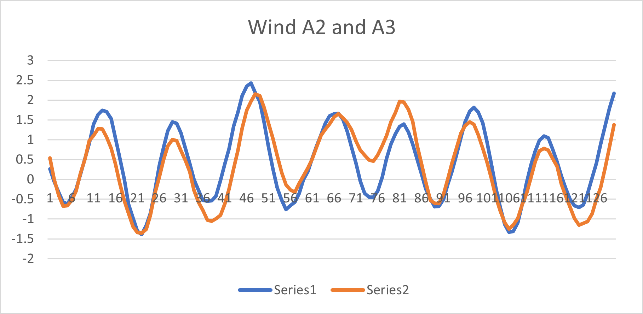
**Figure 18:** An analysis of the first 130 bins of a comparison between microphones 1 (series 1) and microphone 2 (series 2). In this graph microphone 2 has been moved closer to the sound emitting speaker.

**Figure 19:** An analysis of the final 130 bins of a comparison between microphones 1 (series 1) and microphone 2 (series 2). In this graph microphone 2 has been moved closer to the sound emitting speaker.

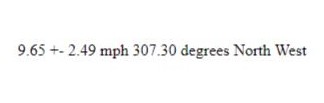
**Figure 17:** An analysis of the final 130 bins of a comparison between microphones 1 (series 1) and microphone 2 (series 2). In this graph both microphones are equally spaced from the sound emitting speaker.

**Figure 16:** An analysis of the first 130 bins of a comparison between microphones 1 (series 1) and microphone 2 (series 2). In this graph both microphones are equally spaced from the sound emitting speaker.

Analyzing **Figures 16 & 17**, our data shows a slight offset even when microphones are equidistant from the speaker. To compensate for this inaccuracy, the displaced offset is compared to the initial state, rather than to a zero condition. In **Figures 18 & 19**, we see that microphone 2 goes from receiving the sound signal before microphone 1 to receiving the sound signal after microphone 1. This data set is incredibly significant because it demonstrates the consistency of our machine. In **Figure 16**, one can see that all the peaks of the blue wave are to the right of the orange peaks. In **Figure 18**, the microphone represented by the orange curve has been moved further from the speaker. One can now see that all the peaks of the blue wave are to the left of the orange peaks. The same similarity can be seen between **Figures 17 & 19**. Our machine was easily able to detect that microphone 2 had moved and was able to show that difference consistently in the data it collected. This data set is also significant because it speaks to the accuracy and feasibility our project because it means our machine can pick up a displacement as small as . Looking at the graphs, specifically the -axis, it appears the offset between the equally spaced microphones and the moved microphones is approximately bins. This is incredibly close to our theoretical value of bins, meaning it is unlikely that the higher frequency seen in our data will affect our calculations. In further testing, we could study the interaction between the electrical signals and our microphones in more depth. Using a device such as an oscilloscope we could develop a circuit that has less electrical interference with the amplifier and RTC.



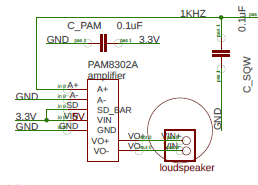
**Figure 20:** Comparison of each microphone pair where the left row of plots was taken with no wind present and the right row with a 4.5 mph wind.

The plots seen in **Figure 20** are representative of our measurements when the air is still and when a fan is blowing directly East at mph with respect to the board. The mph was measured using an anemometer. The measured value of wind speed using our microphones was mph. By comparing the 5th set of plots in reference to our mic placement shown in **Figure 20** it can be seen that there is an offset between mics 2 and 4 due to a Westward wind. The measured speed and from the anemometer and microphones have a difference that agrees with the approximate precision calculated. Below an example of our LCD print statement can be seen.

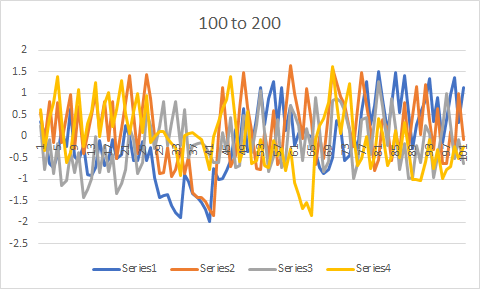
**Figure 21:** An example output of our wind measurement with speed, uncertainty, and direction.

Difficulties and Revisions:

We encountered many difficulties while prototyping and testing our board and code.

One difficulty we encountered was the large amount of internal, electrical noise that was in our original circuit. Because of this “random” electrical noise, it was hard for us to see the periodic waves in our data. After additional testing, we determined that the noise was coming from the PAM8302A amplifier. The shared power source allowed the amplifier to introduce noise to the PCB. To remedy the issue, we removed the amplifier from the power source the rest of our board uses and attached it to an independent voltage source, which ended up being an external battery pack.

**Figure 22:** Diagram of the original wiring of the PAM8320A. Originally, it was connected to the 3.3V source powered by the Feather M4.

After changing the power source of the amplifier there was still extraneous harmonic signals interacting with the kHz signal. As discussed in the Our Measurements section, this extraneous signal did not handicap the capabilities of our board. The interfering signal was still a periodic wave, but was a higher frequency than the signal we intended to produce. An additional way to remedy this situation besides reassembling the board would be to add a low-pass filter. A low-pass filter is a device that only allows frequencies below a decided value to pass through. Though this was an option, we found this extraneous signal didn’t affect our data, so we simply chose to leave it alone.

**Figure 23:** An example of the remaining electrical noise. Note that the data is normalized and is less than half the electrical amplitude of the microphone recording.

Another difficulty we encountered involved our microphones. When performing tests, we noticed that values received from our microphones did not change when we held one of the four microphones in our hand. We were holding the microphone so it did not hear any noise from the speaker, but we still found that the “deaf” microphone was receiving a signal. This error was corrected by resoldering connections between the microphone and our board.

Our code also went through a series of developments and difficulties to arrive at our current design. We began with a simple script to record data from two microphones and arrange the data into one array. This program worked well for early prototyping on an Arduino Mega2560 wired on a breadboard, but was not easily adaptable to our current, four microphone device. After switching to our printed circuit boards with the Feather M4 processor, we needed to develop a new program that could quickly read all four microphones and store the values in separate arrays inside of the M4’s SRAM, where the values could be used to calculate offset values between arrays.

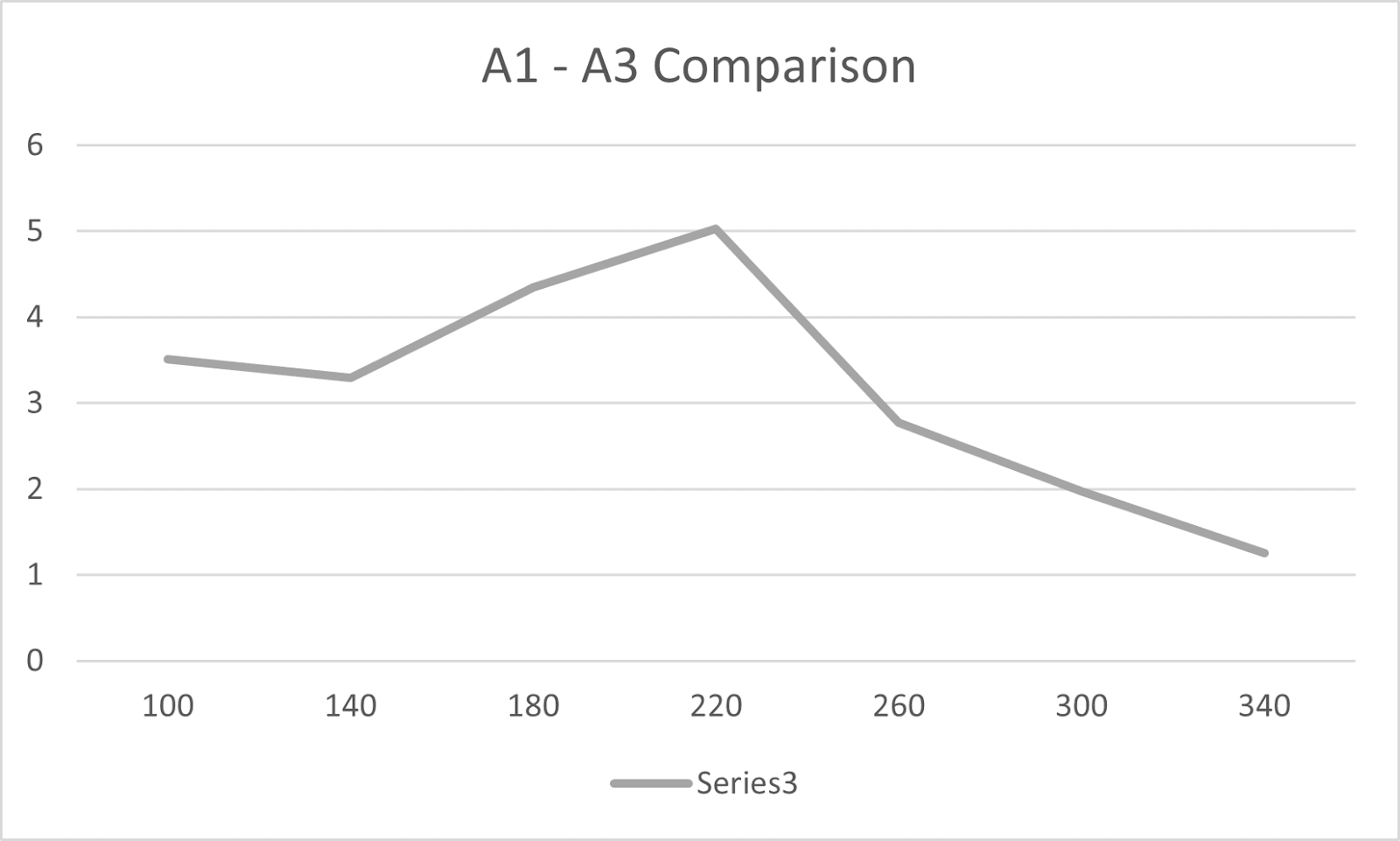
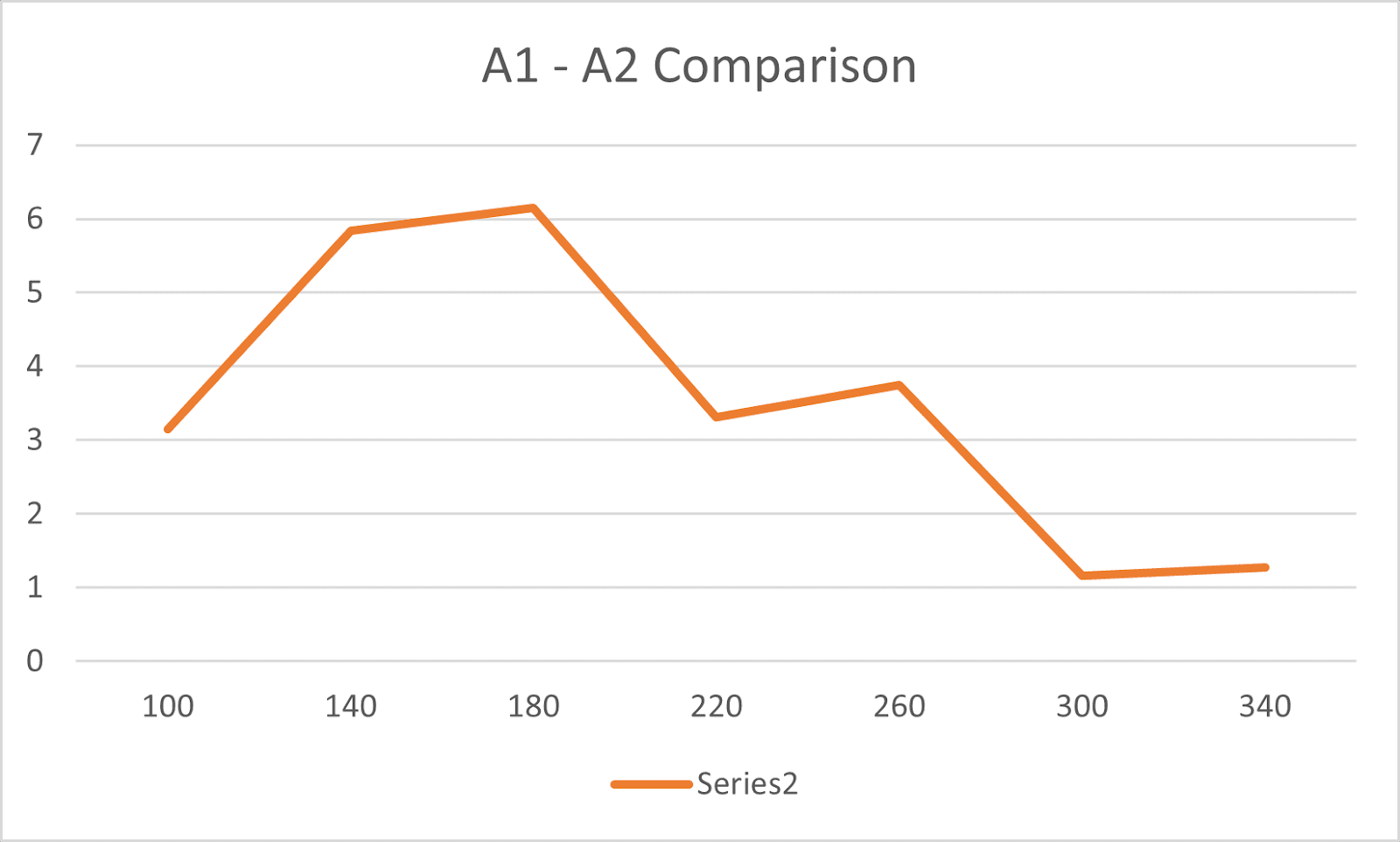
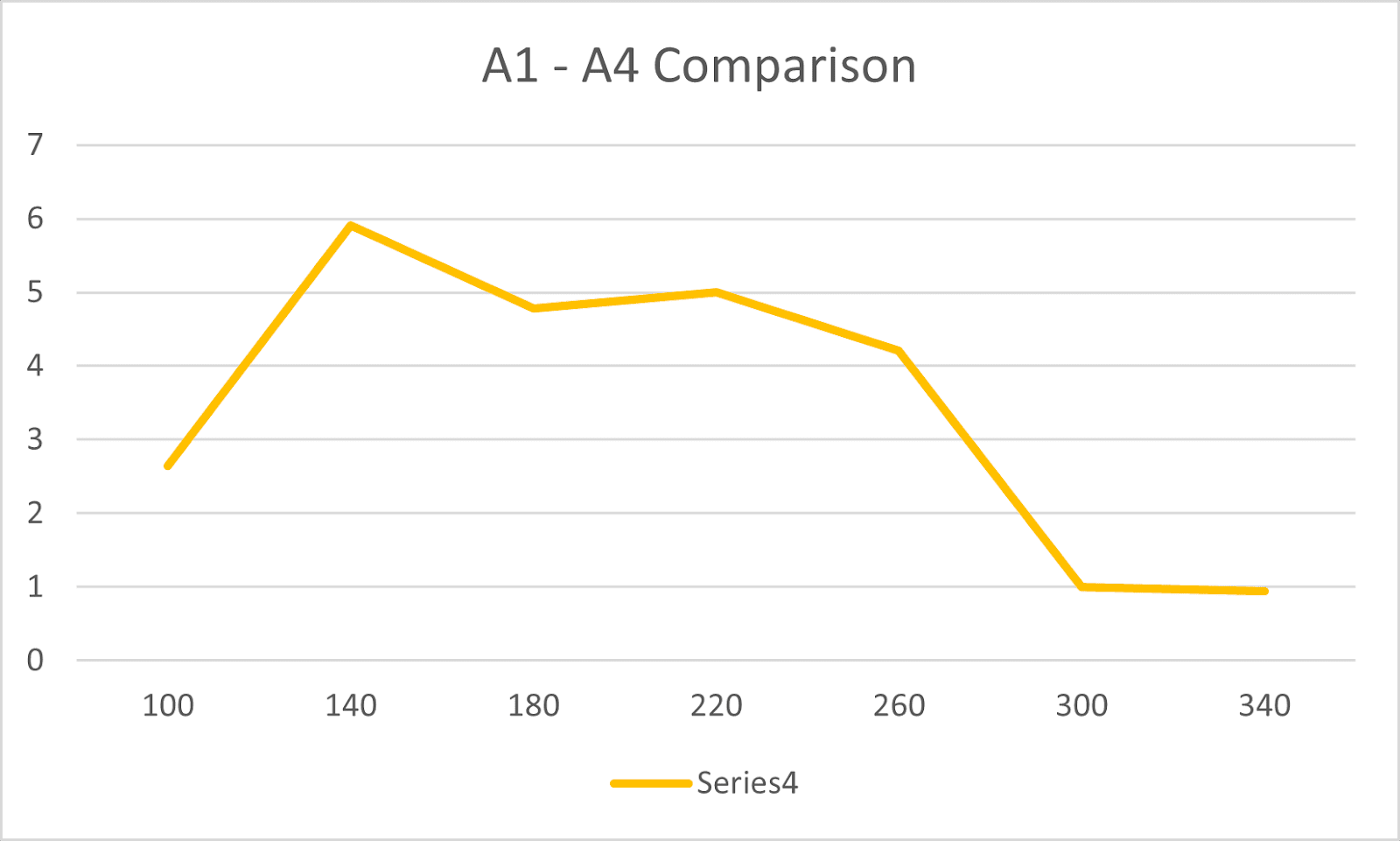
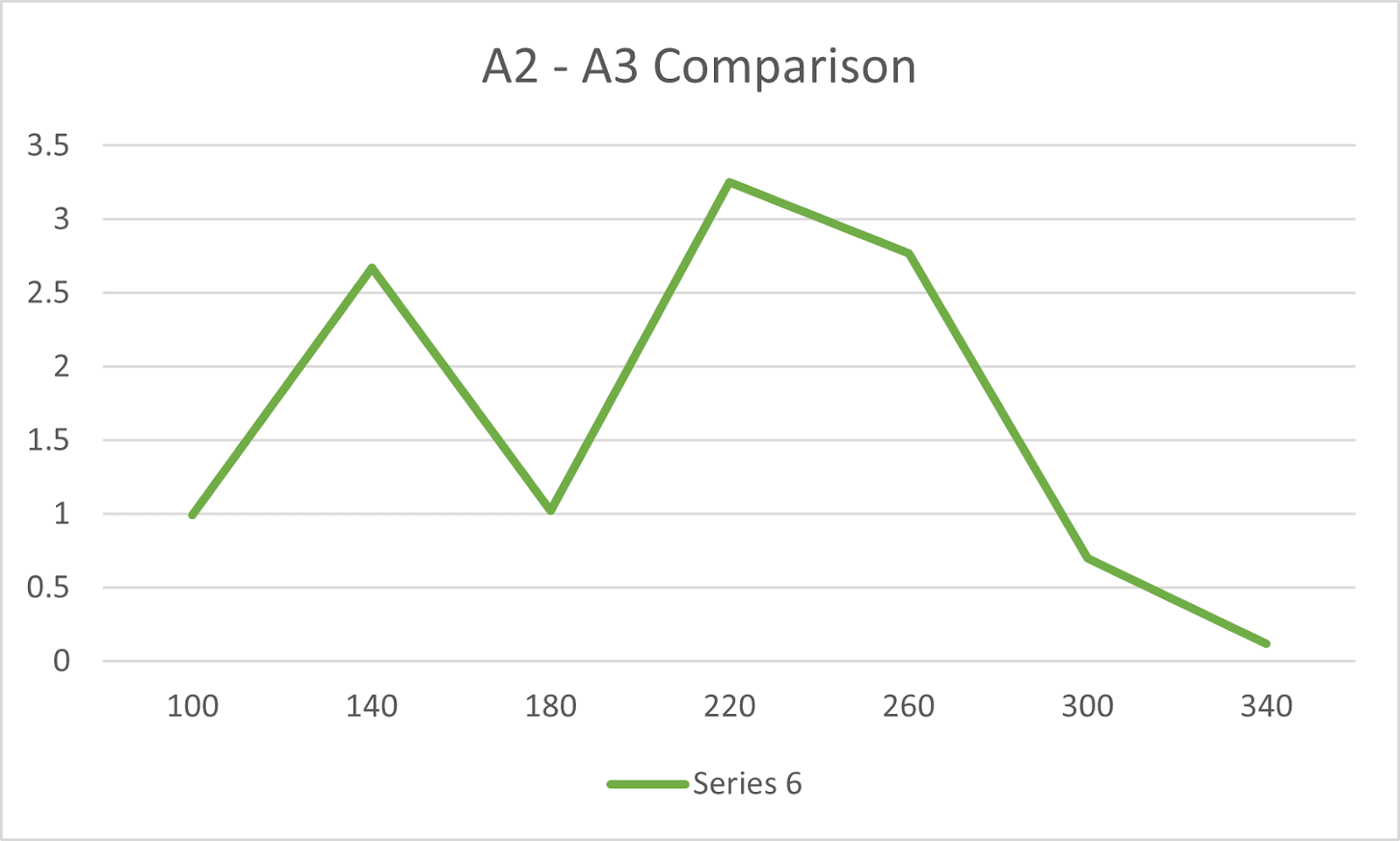
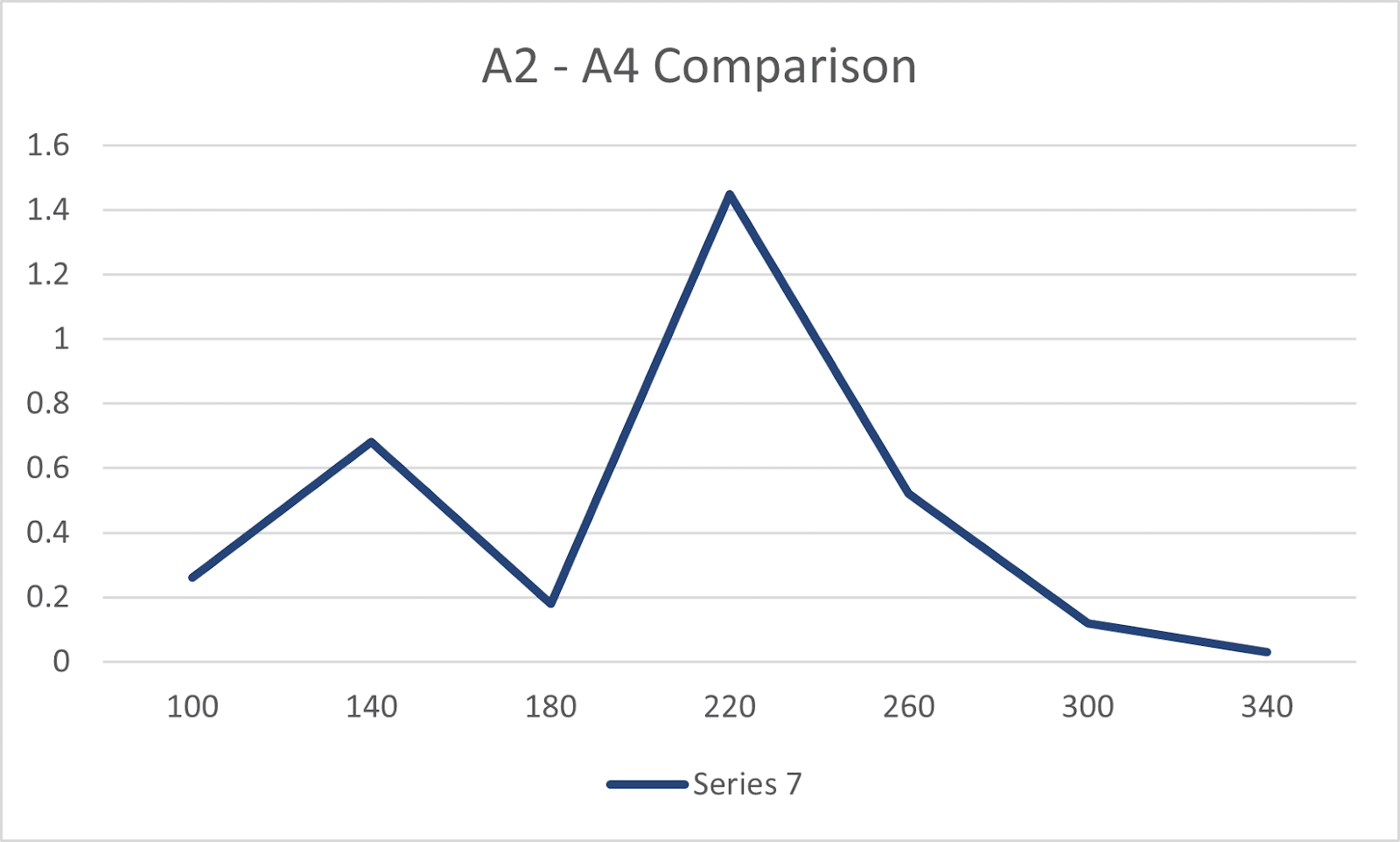
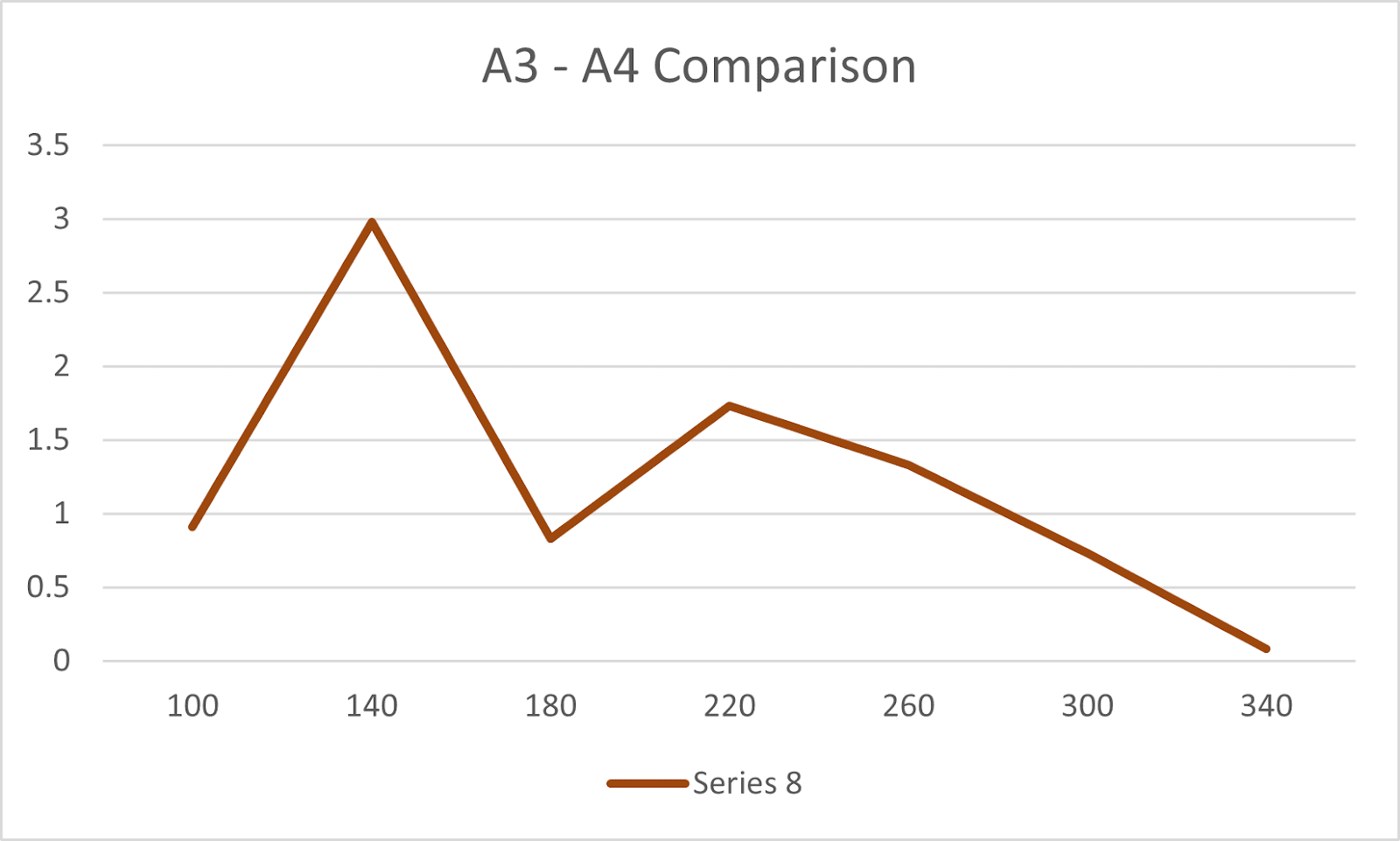
We had to further edit our code to run on boards with no microphones. Due to the COVID-19 pandemic, some group members did not have access to all of the necessary equipment needed to be able to take their own data from fully functioning boards. This problem was addressed by adding the ability to write and read from a Micro-SD card. This function allows us to easily transfer data between group members, who can then perform their own analysis on that data.

Anomalies in Tuning:

In order to use a device in an environment with wind, we need to tune it first to a still air environment. However, the devices are susceptible to detuning in between trials. After testing the board in a windy environment and then returning to still air, the average offset for a microphone pairing may change by up to . Then, trials in still air need to be repeated, the average offsets between microphone pairings must be recalculated, and those values must be reentered into the code. After multiple trials alternating between tuning and testing the board in front of a fan, we were able to see a trend in which some microphones were more likely to have average offset changes than others. It is possible that after the board has been in use in many different environments, some of the microphones experience wear and are not as reliable. It is also possible that the tape securing the loudspeaker to the board has a tendency to shift between certain orientations, which could be why we see some microphone pairings alternating between higher and lower offsets. The variance in average offset in still air has a negative effect on both our speed and direction measurements, and it is not possible currently to receive a highly accurate wind speed measurement without first tuning the device. Since a still air environment is necessary for our current tuning method, further research and development is needed before the board can be reliably used in an outdoor setting, where there is no option for quickly re-calibrating the board.

Accuracy and Further Experimentation:

Further experimentation is needed to determine the efficacy and feasibility of our machine to accurately determine wind speed and direction. Current experimentation has shown our machine to be incredibly accurate and efficient, only needing milliseconds to determine wind speed and direction, but further data replicating our findings would strengthen our hypothesis. Below are some graphs comparing precision data from each pair of microphones.



**Figure 28:** Plot of standard deviation from different sizes of recorded arrays. Comparison between microphone 2 & 4.

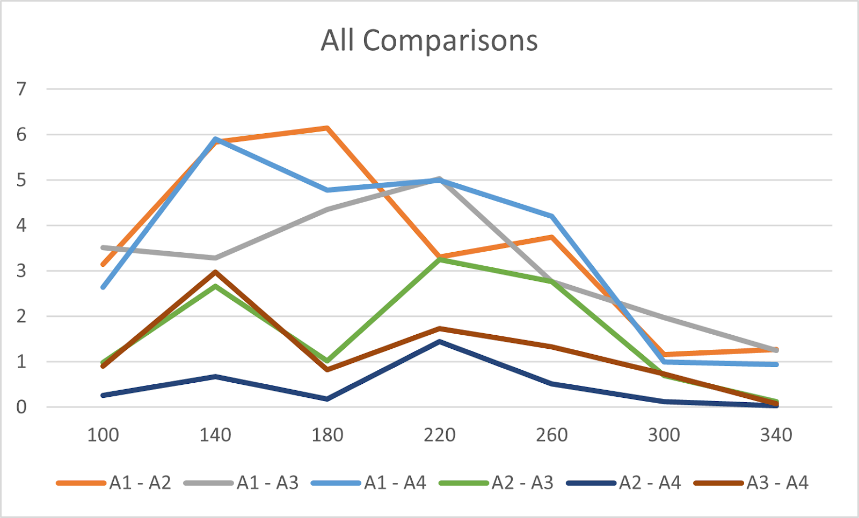
**Figure 29:** Plot of standard deviation from different sizes of recorded arrays. Comparison between microphone 3 & 4.

**Figure 27:** Plot of standard deviation from different sizes of recorded arrays. Comparison between microphone 2 & 3.

**Figure 26:** Plot of standard deviation from different sizes of recorded arrays. Comparison between microphone 1 & 4.

**Figure 25:** Plot of standard deviation from different sizes of recorded arrays. Comparison between microphone 1 & 3.

**Figure 24:** Plot of standard deviation from different sizes of recorded arrays. Comparison between microphone 1 & 2.



**Figure 30:** Plot of standard deviation from different sizes of recorded arrays. All previous graphs combined for analysis.

Based on the plots, the standard deviation decreases as the batch size increases. This occurs because the value of each individual batch is being compared to the mean of all their values. The larger the batches are, the less each individual one will vary from the mean of all of them. At bins we found an acceptable average value for standard deviation, which is . There are a total of ADC counts for each microphone, so when we have a batch size of , there are slices.  From our recordings, a batch size of bins is viable for measuring wind speed. Importantly, this is compared to the “perfect” sample of bins. The goal of batching is to reduce the time needed for calculations, so that a theoretical engineer could get samples quicker than a full bins.

Further experiments could look at possible interference caused by background noise. Our testing was done primarily in quiet environments with little outside sound. However, additional noise could render the device significantly less accurate. The exact scope and magnitude will need to be determined to assess the feasibility of our device in other situations. It might be possible to implement code or a physical element that could remove unwanted audio interference. This could be done by removing outlying microphone values or physically altering the microphones in some way. However, it is likely that the device is less accurate in very noisy environments. Extreme modification will be needed to eliminate interference.

Additional experiments could look at the effects of different speaker tones. We chose to use the jagged, periodic kHz wave because of its continuous variance over time to maximize the difference between two recorded waves. This way a parabola function could be used for specific calculation of the offset. Future experiments analyzing our device might find it beneficial to look into using square waves, triangle waves, or sawtooth waves. Such an implementation could use the various peaks of these waves to compare the offset more accurately between two microphones, at the cost of offset precision.

**Conclusion**

Possible Applications:

 Originally, our intent was to build a machine that sound engineers at outdoor concert venues could use to determine wind speed and direction in order to adjust speaker phase for better sound quality. When wind speeds vary at different sections of an outdoor venue, sound waves are shifted with respect to each other and phasing issues occur, leaving areas of a venue with uneven loudness. Our idea was for multiple boards of our machine to be placed at various points throughout the venue and local wind speed and direction could be determined at each of those points, creating a comprehensive wind vector map for the sound engineer.  Frequencies that may typically be found in music range from Hz to kHz (*PSB Speakers*). These frequencies correlate to wavelengths of meters through centimeters. With our current device, we can confidently measure wind speeds above . In an outdoor setting with sound waves between Hz and kHz, a wind speed of can create phasing issues throughout the venue. This device would be accurate enough to provide use to a sound engineer controlling the phasing of outdoor venue speakers.

**Figure 31:** Red Rocks Amphitheater is located in the mountains just outside the city of Denver, CO. Though the mountains are very good at blocking out wind, an outdoor venue like this may benefit from using our device. Credit: <https://cmhof.org/red-rocks-amphitheatre-kicks-off-another-season/>

Summation of Findings:

Our goal for this project was to create a highly accurate way to measure wind speed and direction using four microphones and a tone generator. We were able to successfully create a device that could record data from four microphones and then compare the arrival time of a kHz wave at each microphone. Our device takes readings at a rate of approximately kHz and is capable of measuring wind speeds of mph and above. By comparing offsets between different microphones, our device also determines wind direction. Our device can be further modified for accuracy by extending the length of the microphone struts. A future experimenter could design new struts with changeable lengths so future experiments have the ability to quickly change the distance between the microphones and speaker. Each measurement from our device consists of samples total from the four microphones. However, we found that we could lower the sample size to without significant loss in precision. A sample size of corresponds to a recording of milliseconds. Our device provides a highly accurate measurement of wind speed in a very short amount of time.

Works Cited

“Class D Audio Amplifier Basics.” *Infineon*, www.infineon.com/dgdl/an-1071.pdf?fileId=5546d462533600a40153559538eb0ff1.

“Enabling Timekeeping Function and Prolonging Battery Life in Low Power Systems.” *DigiKey*, Digi-Key Electronics, 14 Dec. 2011, www.digikey.com/en/articles/enabling-timekeeping-function-and-prolonging-battery-life-in-low-power-systems.

“The Frequencies of Music.” *PSB Speakers*, 1 Apr. 2021, www.psbspeakers.com/the-frequencies-of-music/.

Munz, Steve. “What Are the Different Types of Audio Amplifier Classes?” *Audioholics Home Theater, HDTV, Receivers, Speakers, Blu-Ray Reviews and News*, Audioholics Home Theater, HDTV, Receivers, Speakers, Blu-Ray Reviews and News, 11 Dec. 2019, www.audioholics.com/audio-amplifier/amplifier-classes.

Nichols, Phillip. “5 Ways to Address Phase Issues in Audio.” *IZotope*, 22 Apr. 2021, www.izotope.com/en/learn/5-ways-to-adjust-phase-after-recording.html.

“Noise - Basic Information.” *Canadian Centre for Occupational Health and Safety*, 29 Apr. 2021, www.ccohs.ca/oshanswers/phys\_agents/noise\_basic.html?=undefined&wbdisable=true.

Salomons, Erik. *Sound and Wind*, home.kpn.nl/msalomons/sound.html.

Z, David. “How to Calculate the Vertex of a Parabola Given Three Points.” *Stack Overflow*, 1 Jan. 1958, stackoverflow.com/questions/717762/how-to-calculate-the-vertex-of-a-parabola-given-three-points.