Open Acoustic Devices

Measuring the Accuracy of AudioMoth GPS Synchronisation

theteam@openacousticdevices.info

5th September 2024

The AudioMoth-GPS-Sync firmware uses the pulse-per-second (PPS) output of a GPS module to generate recordings that are synchronised to GPS time. This application note describes experiments feeding the same 1kHz sine wave from a signal generator to four GPS-synchronised AudioMoth demonstrating synchronisation to better than 1us — much less than the interval between the samples and far exceeding that which is typically necessary for sound source localisation.

1 Introduction

The standard AudioMoth firmware supports the use of an external GPS module, powered up briefly before each recording period, to set the time and to measure the actual sample rate of the recordings that follow. This is sufficient for most applications where synchronisation of recordings to a few milliseconds is required.

By contrast, the AudioMoth-GPS-Sync firmware powers the GPS module continuously during recordings and uses the pulse-per-second (PPS) output of the module to count the number of samples that were taken within each second interval and to precisely determine the exact time at which each sample was acquired by capturing the value of the internal 48MHz clock that triggers sample acquisition. A desktop app can then be used to generate synchronised WAV files that have precise timestamps and are resampled to the precise sample rate requested.[1,2]

The accuracy of the resulting synchronised WAV files depends on the accuracy of the PPS output from the GPS module and the processing within the AudioMoth and desktop app that follows. Most GPS modules, including the CDTop PA1616S[3] and PA1616D[4] modules used in the AudioMoth GPS board and hat, provide a PPS output with an accuracy of better than 100ns. This time accuracy corresponds to a position accuracy of approximately 30m. Furthermore, the acquisition time of each sample is determined within the AudioMoth to a precision of 20ns.

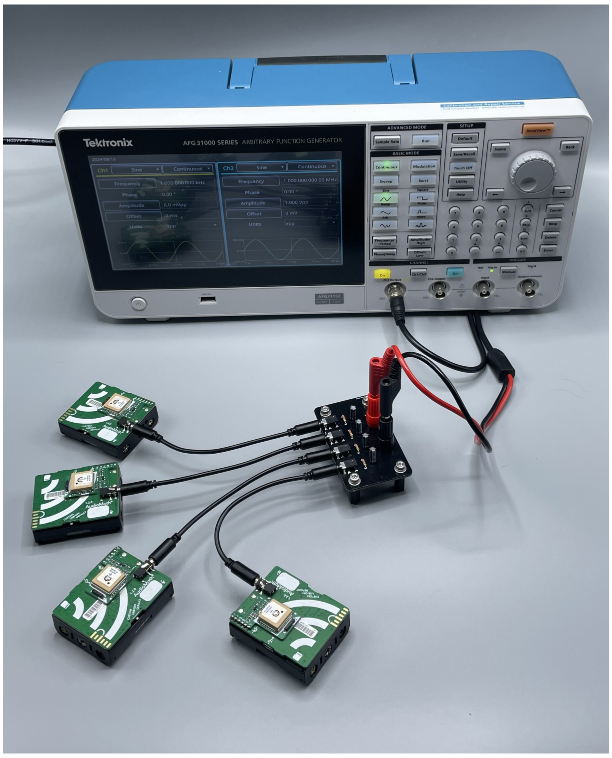

Figure 1: *Tektronix AFG31152 signal generator and four AudioMoth equipped with GPS hats.*

In this application note, we describe experiments to confirm the overall accuracy of the synchronisation process from acquisition to generation of the final synchronised WAV files. To do so, we feed the same 1kHz sine wave from a signal generator to four GPS-synchronised AudioMoth and measure the time offset in the resulting WAV files.

2 Methodology

To measure the time offset between AudioMoth devices we use a precise Tektronix AFG31152 signal generator to feed the same 1kHz sine wave to the external 3.5mm

[1] https://www.openacousticdevices.info/gps-sync

[2] https://github.com/OpenAcousticDevices/Application-Notes/blob/master/Using_AudioMoth_GPS_Sync_to_Make_Synchronised_Recordings/Using_AudioMoth_GPS_Sync_to_Make_Synchronised_Recordings.pdf

[3] https://www.cdtop-tech.com/products/pa1616s

[4] https://www.cdtop-tech.com/products/pa1616d

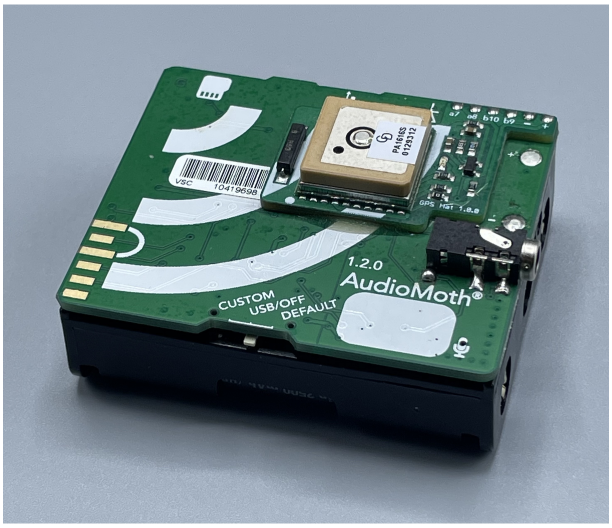

Figure 2: *AudioMoth with external 3.5mm jack and GPS hat.*

jack input of four AudioMoth (see Figure 1). The signal is distributed to each AudioMoth through a custom printed circuit board that uses a 2.7kΩ resistor and a 10uF capacitor to correctly bias and couple the input of each AudioMoth (see Appendix A).

Each AudioMoth runs the AudioMoth-GPS-Sync firmware and uses a GPS hat to generate PPS signals (see Figure 2). The output of the signal generator is recorded over one hour, generating the unsynchronised WAV and CSV files, which are then used by the AudioMoth GPS Sync App to generate the final synchronised WAV files.[5]

These files are then analysed to determine the time offset of the input signal. We consider a short interval of N samples within the one-hour recording, where each sample has a value of y_i and a corresponding acquisition time of t_i. The acquisition times can be calculated directly from the sample rate as the synchronised WAV file has been resampled to have exactly the requested sample rate (for example 48kHz). We assume that the recorded signal will be a sine wave with unknown DC offset, a_0, amplitude, a_1, and time offset, Δ_a, given by:

$$\hat{y}_i = a_0 + a_1 \sin\left(2\pi f \left(t_i + \Delta t\right)\right) \tag{1}$$

We can compare this expected signal against the actual sample values and determine the values of a_0, a_1, and Δt that minimise the squared error:

$$\underset{a_0, a_1, \Delta t}{\arg\min} \sum_{i=1}^{N} \Big[y_i - a_0 - a_1 \sin\left(2\pi f \left(t_i + \Delta t\right)\right) \Big]^2 \tag{2}$$

We perform this estimation process using an iterative routine where we sequentially optimise a single parameter while assuming that the others are constant. See Appendix B for the details of this process.

[5] `https://www.openacousticdevices.info/gps-sync`

Figure 3 shows an example of this process where the first 4ms of the synchronised WAV files, recorded at a sample rate of 48kHz, are shown. In this case, $N = 192$ and the interval covers four cycles of the 1kHz input signal. The estimated time offsets of each recording is -202.95, -202.75, -202.31 and -202.21us respectively. The absolute time offset will depend on the phase difference between the signal generator and the recordings, and is not of interest. The key aspect that interests us here is the relative time offset calculated by subtracting the mean from each of these four values to give -0.40, -0.20, 0.23, and 0.35us respectively.

Figure 4 shows the same process repeated every 100ms for the first 60 seconds of the recording. The upper plot shows the absolute time offset of each AudioMoth recording and shows the gradual drift that results from the actual frequency of the signal generator not being precisely 1kHz, and thus the phase of the input signal does not remain constant over each 4ms interval that is processed. The lower plot shows the relative time offset of AudioMoth after subtracting the mean value at each measurement point.

The plots show that the four AudioMoth maintain a consistent relative time offset across multiple intervals. The accuracy of the time offset determined is shown to be approximately 0.05us which results from the noise in the synchronised WAV files within the AudioMoth audio front-end and subsequent analogue-to-digital conversion of each sample.

3 Experiments

The process described above was repeated over four one-hour recordings separated by 5-minute intervals. All GPS modules acquired a fix from cold-start before the first one-hour recording period. The same 1kHz input signal was used in each experiment, but the sample rates of the AudioMoth devices were varied as shown below:

Experiment 1 - Mixed AudioMoth 1, 2, 3 and 4 record at sample rates of 16, 48, 96 and 192kHz respectively.

Experiment 2 - 192kHz All AudioMoth record at a sample rate of 192kHz.

Experiment 3 - 48kHz All AudioMoth record at a sample rate of 48kHz.

Experiment 4 - 16kHz All AudioMoth record at a sample rate of 16kHz.

On completion of the data collection process, the raw WAV and CSV files were used to generate synchronised WAV files using the AudioMoth GPS Sync App and were analysed as described above with the absolute and relative time offset of each AudioMoth calculated

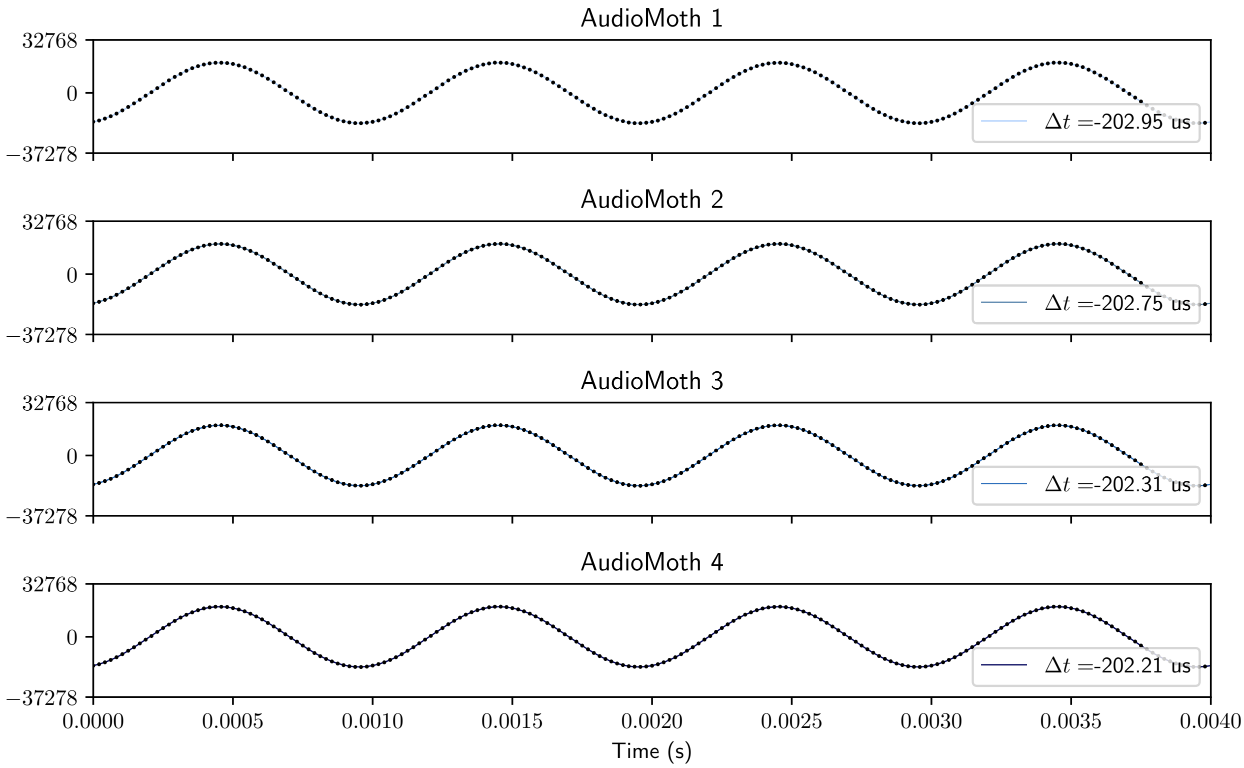

Figure 3: *Raw samples and estimated sine wave over the first 4ms of four synchronised AudioMoth recordings.*

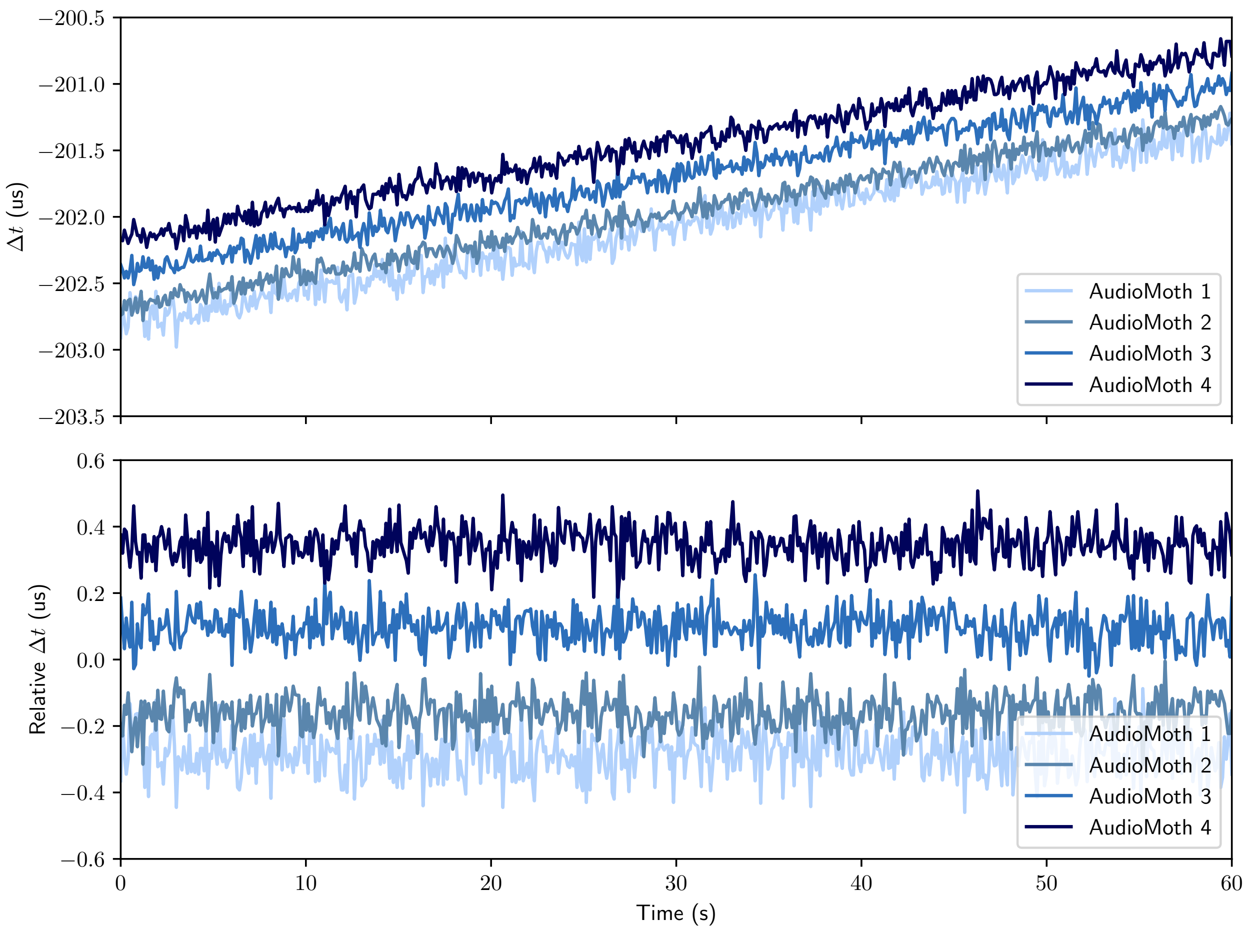

Figure 4: *Estimated absolute and relative time offsets over the first 60 seconds of four synchronised AudioMoth recordings.*

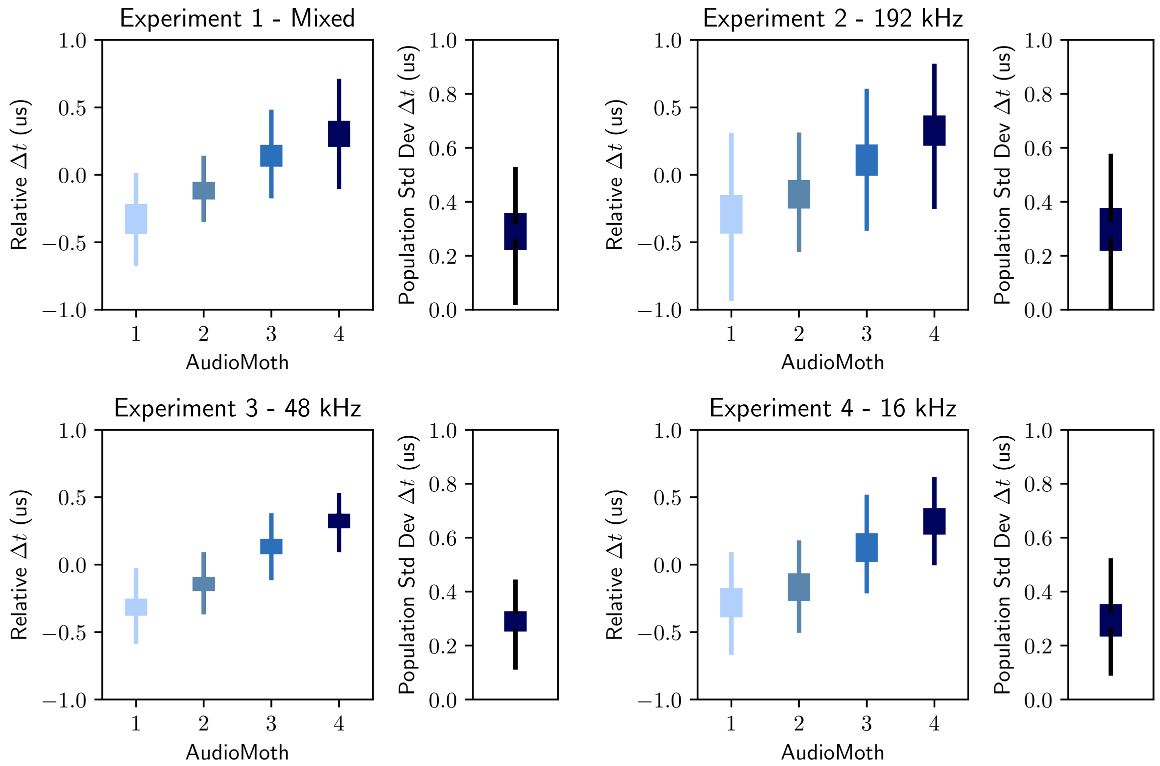

Figure 5: *Maximum and minimum values (thin lines) and standard deviation (thick lines) of individual relative time offsets and the population standard deviation of relative time offset over all 36,000 4ms intervals in each experiment.*

every 100ms across each one-hour recording, resulting in 36,000 estimates of the time offset. In all cases, we consider $N = 192$ samples, corresponding to 1, 2, 4 and 12ms at sample rates of 192, 96, 48 and 16kHz.

Note that Experiment 1 involves AudioMoth recording at different sample rates. The AudioMoth GPS Sync App already corrects the acquisition time of each sample to accommodate the different period over which it is acquired at different sample rates (due to the different oversampling rates). However, due to the processing limits of the AudioMoth, the software DC blocking filter that is normally applied within the firmware is omitted when sampling at 192kHz. This filter is a first-order high-pass Butterworth filter, and like all filters, will generate a small phase shift, and resulting time offset, in the filtered output signal. The size of this time offset is given by:

$$\delta t = \frac{\arctan\left(f_c/f\right)}{2\pi f} \tag{3}$$

where f_c is the cutoff frequency of the filter and f is the signal frequency. For values of $f_c = 48$Hz and $f =$ 1kHz the expected time offset, δt, is 7.6us. Since all AudioMoth in Experiments 2 to 4, and all AudioMoth other than AudioMoth 4 in Experiment 1 are subject to this small additional absolute offset, we apply a 7.6us correction to the time offset estimated for AudioMoth 4 in Experiment 1 to compensate.

4 Results

Figure 5 shows the overall results across all four experiments, each plot line shows the maximum and minimum values (thin lines), and the standard deviation (thick lines) over all 36,000 4ms intervals. Note that the relative time offset is consistent across each one-hour recording period regardless of the sample rate. The figure also shows the population standard deviation of relative time offsets, again evaluated over all 36,000 4ms intervals. In all cases, the population standard deviation is less than 1us with a mean value across all experiments of approximately 0.3us.

Figure 6 shows the recorded GPS positions every second over each one-hour recording. The GPS positions are plotted relative to a ground truth position measured using an EMLID Reach RS+ GPS receiver over a 12-hour period and processed using the Canadian Spatial Reference System Precise Point Positioning (CSRS-PPP) service to give an estimated position accurate to $\pm$1m.

The GPS tracks recorded are typical of the results from a stationary GPS receiver, where individual noisy GPS fixes generate a random walk in the output of the GPS module's internal Kalman filter. Note that the AudioMoth recording at 192kHz exhibits significantly greater noise in their GPS positions. This is most likely due to the higher current consumption generating more power supply noise that ultimately has an impact on

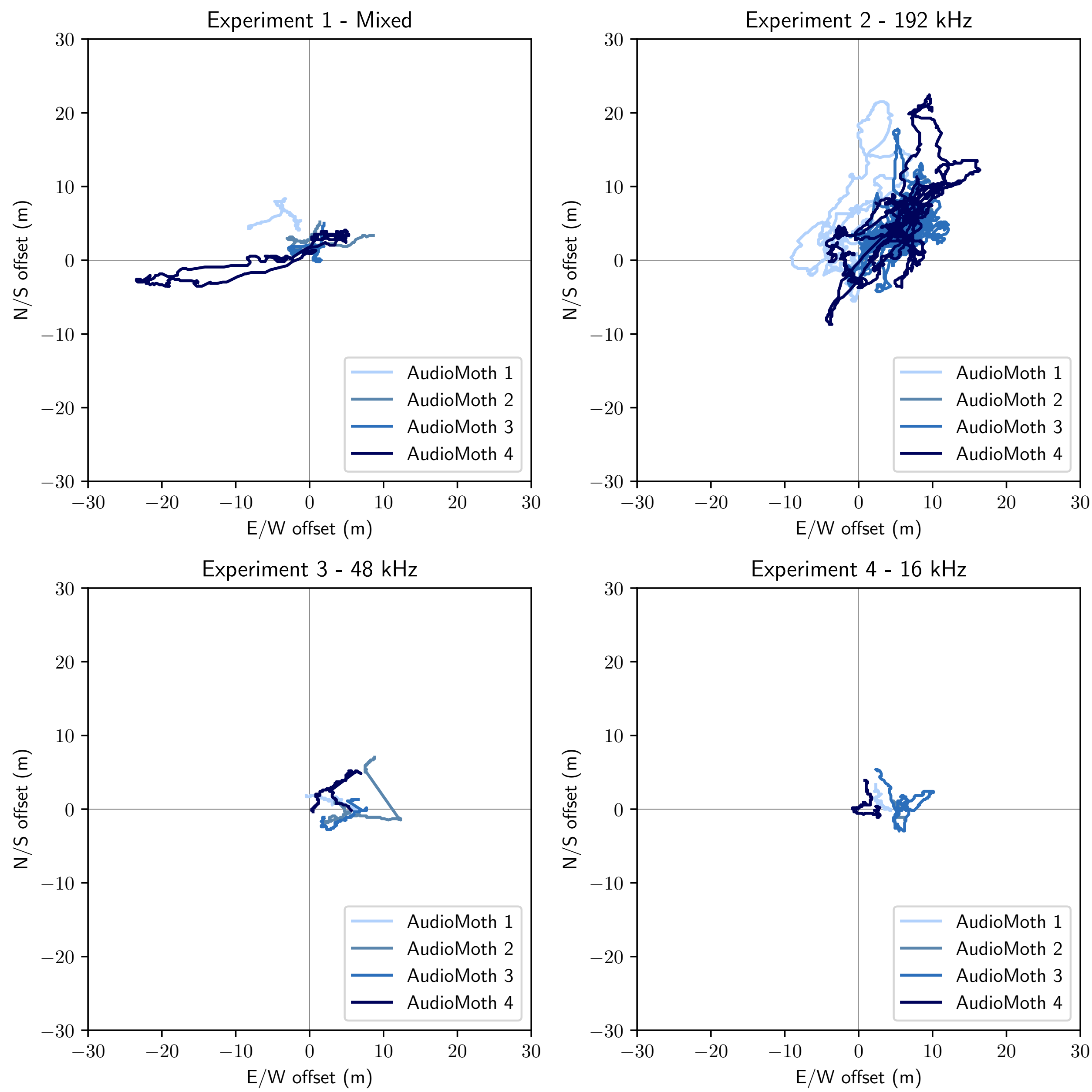

Figure 6: *Relative GPS position every second over each one-hour recording in each experiment.*

the reception quality of the GPS module. However, in all cases, the position estimate is typically less than 30m from the ground truth position, corresponding to a time error of less than 0.1us.

5 Discussion

The experiments performed here demonstrate synchronisation between individual AudioMoth to better than 1us. This level of accuracy is much less than the interval between the samples (which ranges from 5.2 to 62.5us at sample rates of 192 and 16kHz respectively). It also far exceeds that which is typically necessary for sound source localisation where a time offset of 1us corresponds to a distance offset of 0.34mm.

However, it is interesting to explore the source of the time offset that is observed, given that the offset of each individual AudioMoth is constant across experiments, and the time offset follows the individual AudioMoth device rather than the GPS module itself. As described earlier, the software DC blocking filter implemented by the AudioMoth firmware introduces a small phase shift and resulting time offset. The same is true of the analogue filters in the AudioMoth amplifier stages. AudioMoth uses a three-stage amplifier.[6] The first stage

[6] `https://www.hardware-x.com/article/S2468-0672(19)30030-6/fulltext`

Frequency (Hz)	Measured Relative Δt (us)				Measured Mean Population Std Dev Relative Δt (us)	Calculated Nominal Filter Time Offset (us)	Calculated Std Dev Filter Time Offset (us)
	AM 1	AM 2	AM 3	AM 4			
1000	0.28	-0.15	0.13	0.29	0.26	2.53	0.15
500	-0.91	-0.05	0.21	0.75	0.70	10.13	0.60
200	-5.19	0.37	0.99	3.83	3.79	63.19	3.70
100	-19.00	1.19	3.97	13.83	13.81	251.20	14.30

Table 1: *Measured relative Δt over 5-minute intervals, and calculated filter time offsets (assuming a 10kΩ 1% resistor and a 1uF 10% capacitor), at four different input signal frequencies.*

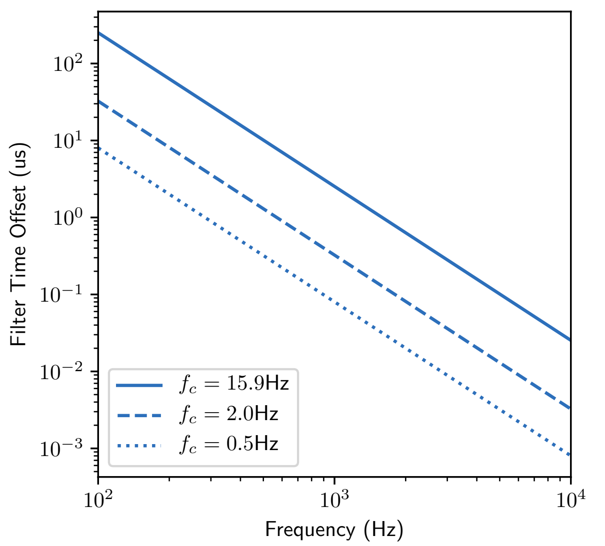

Figure 7: *Calculated filter time offset at different input signal frequencies and filter cutoff frequencies.*

uses a 1uF capacitor and 10kΩ resistor to form a high-pass filter with a cutoff frequency of 15.9Hz. Later stages use 10uF capacitors, with resistors whose values vary with the selected gain. The minimum values of these resistors are 7.8kΩ and 31.8kΩ, over all gain settings, corresponding to maximum cutoff frequencies of 2.0 and 0.5Hz in the second and third stages respectively.

Figure 7 shows the resulting time offset that results from these filters as the input signal frequency is varied (calculated using Equation 3). Note that at low frequencies, the offset due to the first filter is significant. It has a value of 251.2us at 100Hz. The second and third filters have less significant impact due to their lower cutoff frequencies. At 10kHz, all filters have a time offset below 0.1us.

The time offset seen between different AudioMoth devices is largely due to the tolerances of the resistor and capacitor pairs that form these filters. Table 1 shows the results of experiments measuring the relative time offsets over five-minute intervals with test signals of frequencies of 1000, 500, 250, and 100Hz. Note that the observed time offsets increase as the signal frequency decreases and that the observed population standard deviation of relative time offsets is in line with the expected standard deviation assuming a resistor of tolerance $\pm 1\%$ and a capacitor of tolerance $\pm 10\%$ in the first stage of the amplifier (modelled as a uniform distribution about the nominal value). The tolerance of the resistors in the latter stages is expected to be higher (possibly up to $\pm 30\%$) as these are poly resistors within the micro-controller itself [7] However, the lower cutoff frequency of these stages mean that their effect will only be seen at very low frequencies.

If very high accuracy sound source localisation is required with low-frequency sources, the time offset of individual AudioMoth can be measured, and a correction applied. However, in most applications, this will not be necessary.

Appendix

A. Connecting Multiple AudioMoth

The external 3.5mm jack on AudioMoth devices is intended for connection to external electret capsule microphones. However, other signal sources can be connected with the appropriate load resistor to correctly bias the AudioMoth amplifier. The circuit in Figure 8 uses a 2.7kΩ resistor on each AudioMoth input to provide this load, and uses a 10uF capacitor to decouple the resulting DC offset from the input signal source. The same signal is directed to four AudioMoth devices.

B. Estimating Waveform Parameters

As described earlier, we assume that we have N samples, each with a value of y_i and a corresponding acquisition time of t_i. We assume that the input signal is a sine wave of fixed frequency, f, such that each sample value should have the form:

$$\hat{y}_i = a_0 + a_1 \sin\left(2\pi f \left(t_i + \Delta t\right)\right) \tag{4}$$

We can compare this against the actual sample values to derive an error term representing the sum of the

[7] https://www.silabs.com/documents/public/application-notes/an0038.1-efm32-series-1-operational-amplifiers.pdf

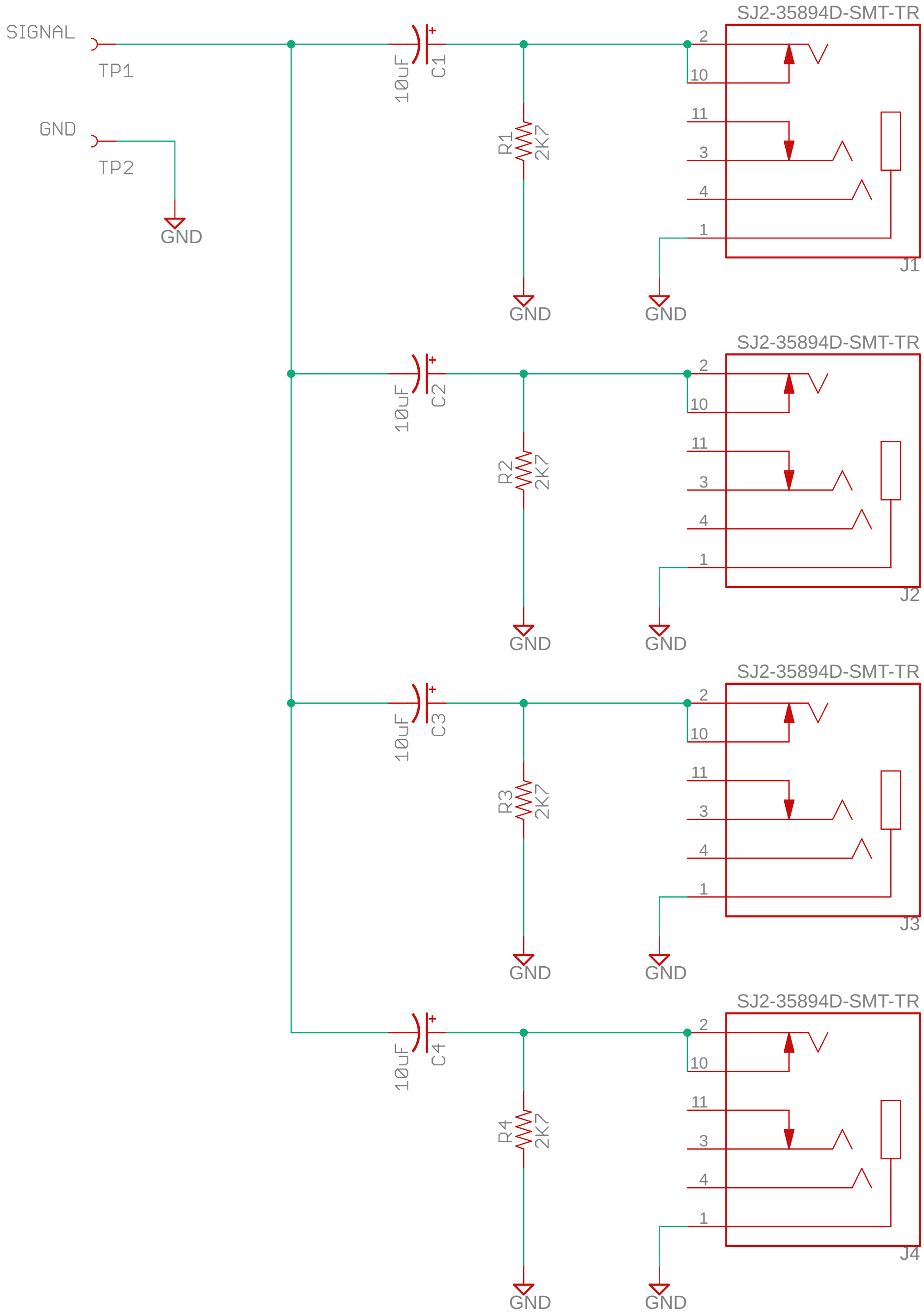

Figure 8: *Circuit to correctly bias the input of four AudioMoth devices and to feed an identical signal to each of them.*

squared difference between the two:

$$E = \sum_{i=1}^{N} \Big[y_i - a_0 - a_1 \sin \left(2\pi f \left(t_i + \Delta t \right) \right) \Big]^2 \quad (5)$$

We then seek to find the values of a_0, a_1 and Δt that minimise this squared error term.

Given an initial guess for a_0, a_1 and Δt, we can generate improved estimates for a_0 and a_1 by calculating the appropriate partial differential, $\partial E / \partial a_i$, and finding the value, $\hat{a_i}$, where the gradient is zero (corresponding to a minimum in the error term). Doing so generates the following two expressions:

$$\hat{a}_0 = \frac{1}{N} \sum_{i=1}^{N} \big[y_i - a_1 \sin \left(2\pi f \left(t_i + \Delta t \right) \right) \big] \quad (6)$$

$$\hat{a}_1 = \frac{\displaystyle\sum_{i=1}^{N} \left(y_i - a_0 \right) \sin \left(2\pi f \left(t_i + \Delta t \right) \right)}{\displaystyle\sum_{i=1}^{N} \sin^2 \left(2\pi f \left(t_i + \Delta t \right) \right)} \quad (7)$$

Doing the same for the time offset, Δt, is a bit more challenging. While it is possible to derive $\partial E / \partial \Delta t$, it is not possible to derive a closed-form expression to solve for the value of $\hat{\Delta} t$ where the gradient is zero.

However, we can calculate the second-order partial differential, $\partial^2 E / \partial \Delta t^2$, and use Newton's method to derive an updated estimate of Δt such that:

$$\hat{\Delta} t = \Delta t + \frac{\displaystyle\sum_{i=1}^{N} \big[y_i - a_0 \big] \cos \theta_i - \frac{a_1}{2} \sin 2\theta_i}{\displaystyle 2\pi f \sum_{i=1}^{N} \big[y_i - a_0 \big] \sin \theta_i + a_1 \cos 2\theta_i} \quad (8)$$

where $\theta_i = 2\pi f \left(t_i + \Delta t \right)$.

Starting with initial guesses of $a_0 = 0$, $a_1 = 32768$ and $\Delta t = 0$, we apply each of the three expressions above, in turn, to derive improved estimates of a_0, a_1 and Δt until the change in Δt is smaller than our required precision. In practice, with a precision of 0.01us, this typically occurs in 2 to 4 iterations.

Note that we do not apply the same estimation to the frequency of the input signal, f, since the Tektronix AFG31152 signal generator has an absolute accuracy of better than 1ppm and small deviations in frequency have minimal effect over the short time intervals (between 1ms and 12ms) considered.