Using AudioMoth with Filtering and Amplitude Threshold Recording

theteam@openacousticdevices.info

1st June 2020

The most recent versions of the AudioMoth firmware (version 1.4.0) and configuration app (version 1.3.0) introduce two new functionalities: the ability to apply low-pass, band-pass or high-pass filters on the device, and the ability to reduce recorded file size by applying an amplitude threshold.

Filters

The new AudioMoth firmware provides the option of applying first order Butterworth low-pass, band-pass and high-pass filters to acoustic data before it is written to the SD card. The type of filter and the cut-off frequency (or frequencies in the case of the band-pass filter) are set in the configuration app along with the other recording settings (see Figure 1). The filters are applied in conjunction with a DC blocking filter which removes any small DC offset from the microphone readings and which is itself a Butterworth high-pass filter with a cut-off frequency of 48 Hz. The filters are designed such that they exhibit unity gain at their peak frequency (see Figure 2).

The filters can be used to remove background noise and to improve the signal to noise ratio of recordings. For example, Figure 3 shows a typical band-pass filter which might be applied to improve the quality of bird song recordings. This is a band-pass filter with cut-off frequencies of 2 kHz and 12 kHz with a sample rate of 48 kHz. The lower cut-off frequency removes low-frequency noise, such as traffic, while the upper cut-off frequency improves the signal to noise ratio of frequencies below 12 kHz (where most bird song occurs) by effectively averaging over neighbouring samples. Test recordings of the dawn chorus in the New Forest, UK were made with these settings and the new AudioMoth

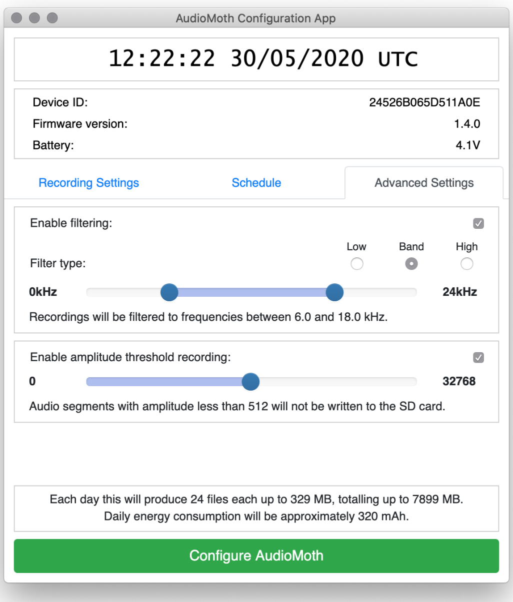

Figure 1: *Configuration app.*

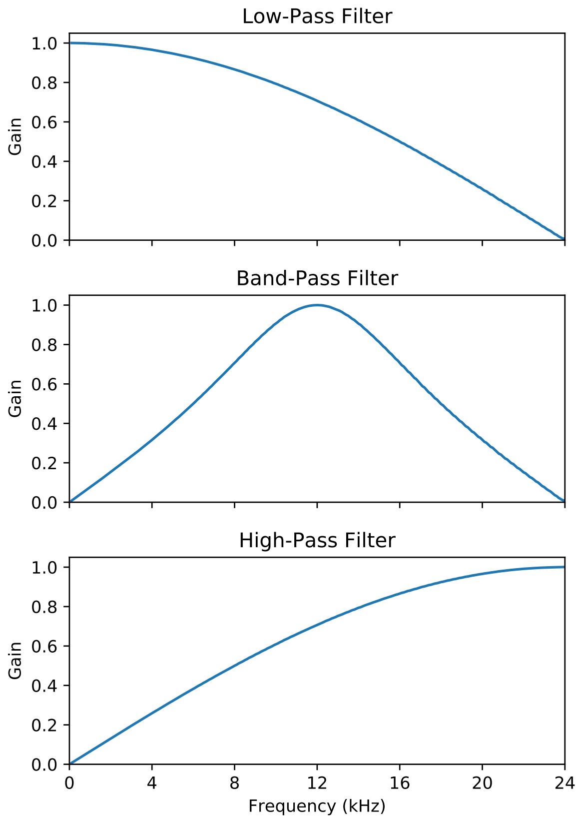

Figure 2: *Example 12 kHz low-pass, 8-16 kHz band-pass, and 12 kHz high-pass filters with 48 kHz sample rate.*

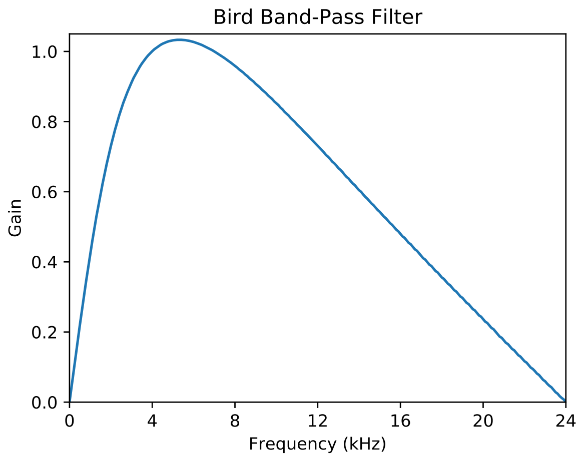

Figure 3: *Band-pass filter (2 kHz to 12 kHz) used for bird song recordings with a sample rate of 48 kHz.*

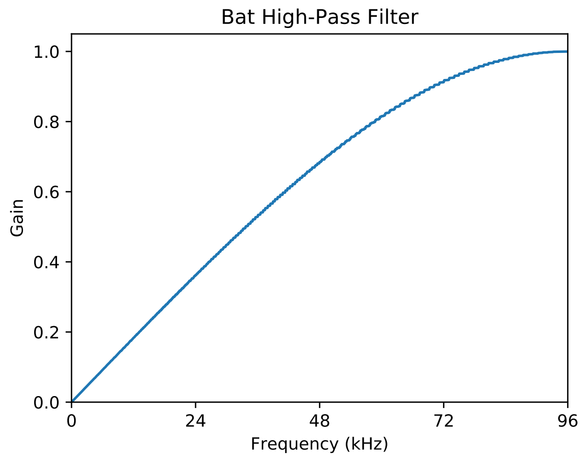

Figure 5: *High-pass filter (50 kHz) used for bat recordings with a sample rate of 192 kHz.*

Figure 4: *Deployed AudioMoth in its waterproof case.*

waterproof case (see Figure 4) in May 2020 (see `https://soundcloud.com/openacousticdevices/example_case_recordingwav`).

Amplitude Threshold Recording

The second new feature allows an amplitude threshold to be set such that only those audio segments (each 32 KB in length) which contain a sample value greater or equal to the threshold will be written to the SD card. This results in much smaller WAV files as silent periods are automatically removed. The lengths and positions of these silent periods are also encoded into the resulting WAV file so that its original length, and the relative timings of events within it, can be restored using an additional feature of the configuration app.

A typical use for this feature is to reduce the resulting file size when recording bats which make short passes over the deployed AudioMoth, leaving most of the recordings empty. In this case, we can use a filter to first reduce the background noise in the recording. A 50 kHz high-pass filter works well for this. A band pass filter could also be used if the echolocation frequency of the target bat is known in advance. Figure 5 shows the characteristics of the 50 kHz high-pass filter and Figure 6 shows an example of its use. The background noise in this case is bird song and speech, and it is almost completely removed by the filter, leaving just the clearly distinguished bat calls. Since AudioMoth WAV files use 16-bit signed values, the possible values of samples are in the range from -32,768 to 32,767.

Having removed background sound, a threshold value can be set to ensure that only those 32 KB segments which contain a sample that exceeds this value will be written to the SD card. The files generated in this way are valid WAV files and can be viewed or listened to by any WAV viewer. The filenames of these recordings end with a 'T' to indicate that a threshold has been set (e.g. `20200530_210400T.WAV`) and the value is also reported in the WAV header comments.

The size of the resulting recordings can be used as an indicator of their content. For example, Figure 7 shows the result of using a 55 second record and 5 second sleep cycle for one hour with a sampling rate of 192 kHz, a 50 kHz high-pass filter and a threshold of 512. These recordings would each normally be 21 MB in size, resulting in over 1.2 GB of data in one hour. However, in this case, the total size of all 60 files is just 13 MB, reducing the total file size by a factor of close to 100, and enabling a quick scan of the file sizes to indicate periods of bat activity.

The original length of the recordings, and the correct relative timings of events within them, can be restored using the 'Expand AudioMoth Recordings' option in the

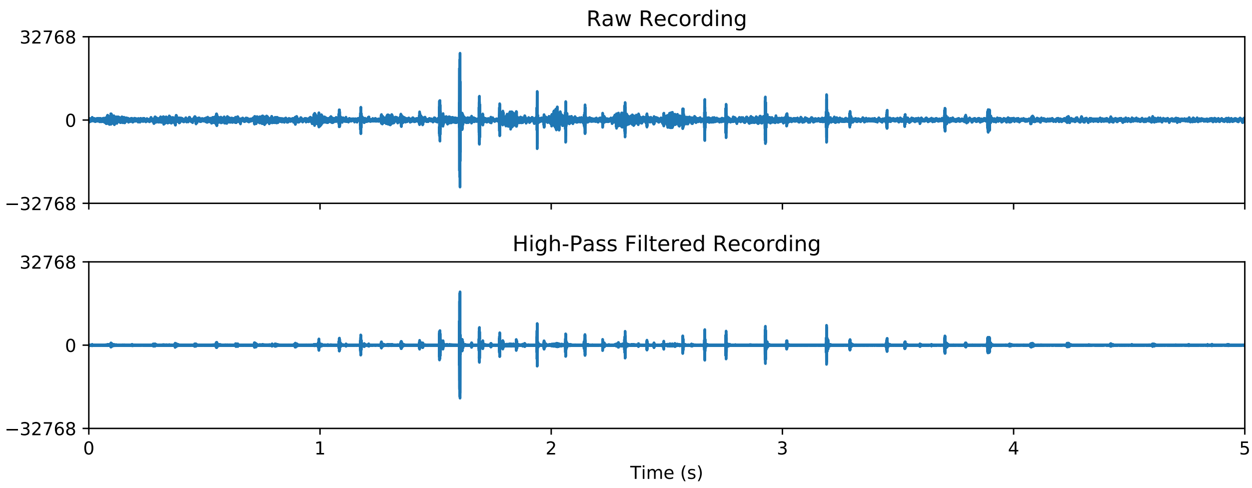

Figure 6: *Example of raw and 50 kHz high-pass filtered recordings showing the reduction in background noise.*

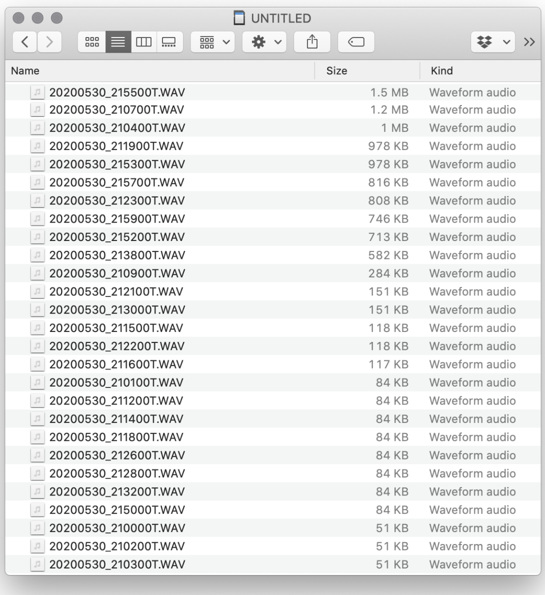

Figure 7: *File sizes during 1 hour of data collection using filtering and amplitude thresholds whilst recording.*

configuration app (see Figure 8). Any silent periods in the file are restored to their original length with a default sample value of zero.

Figures 9, 10 and 11 show an example of this process. Figures 9 shows the original `20200530_210400T.WAV` file recorded by the AudioMoth. This file is 1 MB in size and on examination contains just over 2 seconds of audio. The sonogram clearly shows a bat passing over the AudioMoth with a call around 50 kHz.

Figure 10 shows the resulting `20200530_210400.WAV` produced using the 'Expand AudioMoth Recordings' option of the configuration app. This file is 21 MB in size and has been expanded out to the original 55

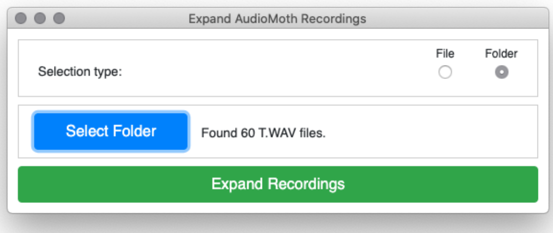

Figure 8: *Configuration app allowing the expansion of recordings back to their original size.*

seconds. The sonogram shows that there was a single bat pass that occurred 35 seconds into the recording.

Figure 11 shows the same expanded `20200530_210400.WAV` file, but this time zoomed in to the period around the 35 second marker. Over this period the correct timings of the individual echo location calls are restored. In this case, each 32 KB segment is approximately 85 milliseconds in length (calculated from 32 $\times$ 1024 bytes / 2 bytes per sample / 192,000 samples per second = 85 milliseconds), and there are several segments within the call, as well as those on either side of it, which did not contain a sample whose value exceeded the threshold.

Choosing the threshold value requires some care and can be done on a trial basis or through analysis of previously collected recordings. If the threshold is set too low the resulting recordings will likely contain too much background noise. If it is too high the recordings will likely miss some quiet bat calls. The AudioMoth red LED will only flash when it is actually writing data to the SD card and can be used to judge whether or not background noise is triggering the recording. A value of 512 works well in many cases. We plan to release a web-based tool to assist in choosing the threshold value in the future.

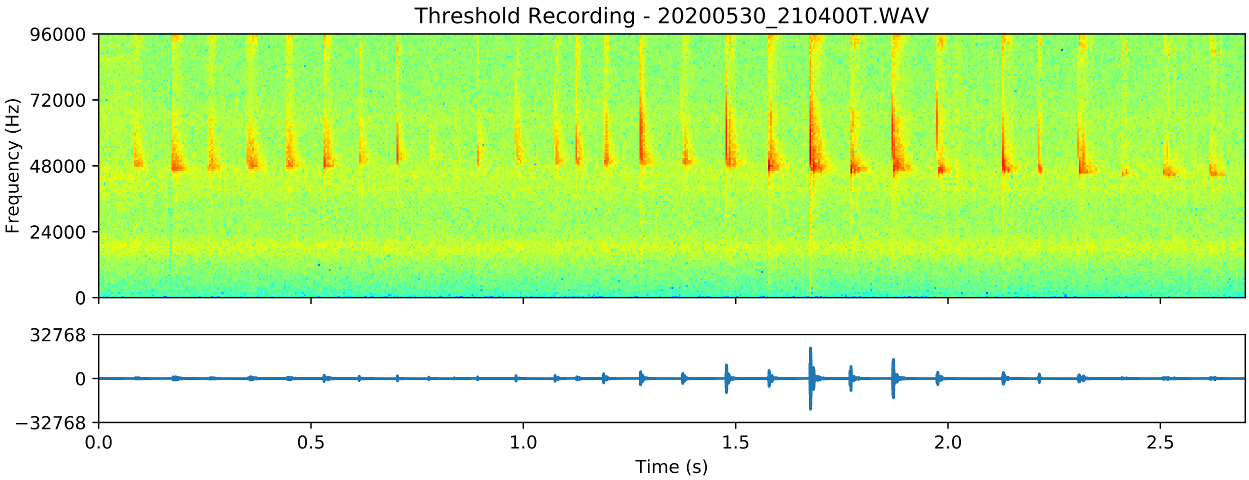

Figure 9: *Original* 20200530_210400T.WAV *file.*

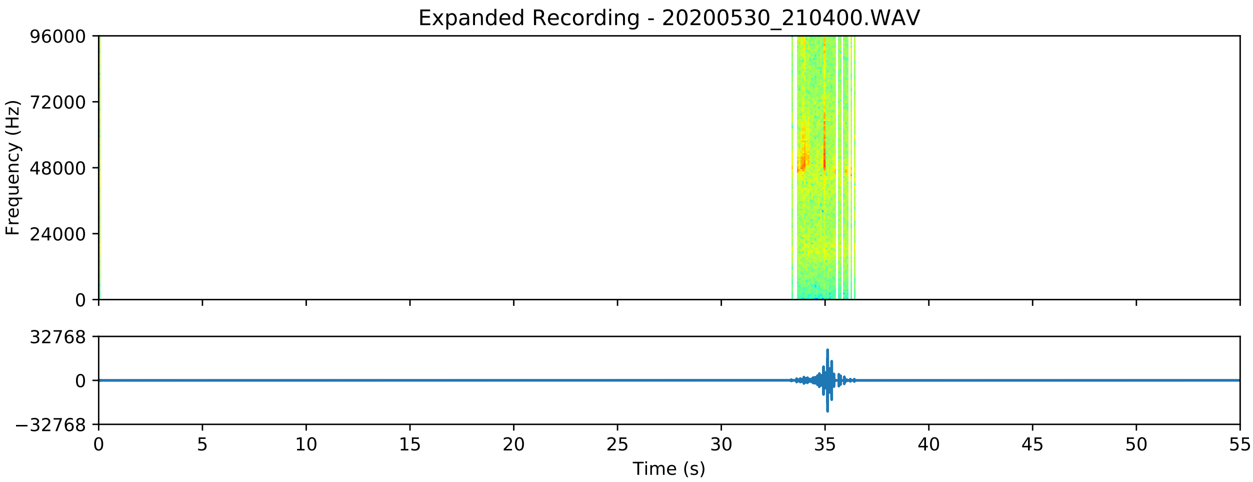

Figure 10: *Expanded* 20200530_210400.WAV *file.*

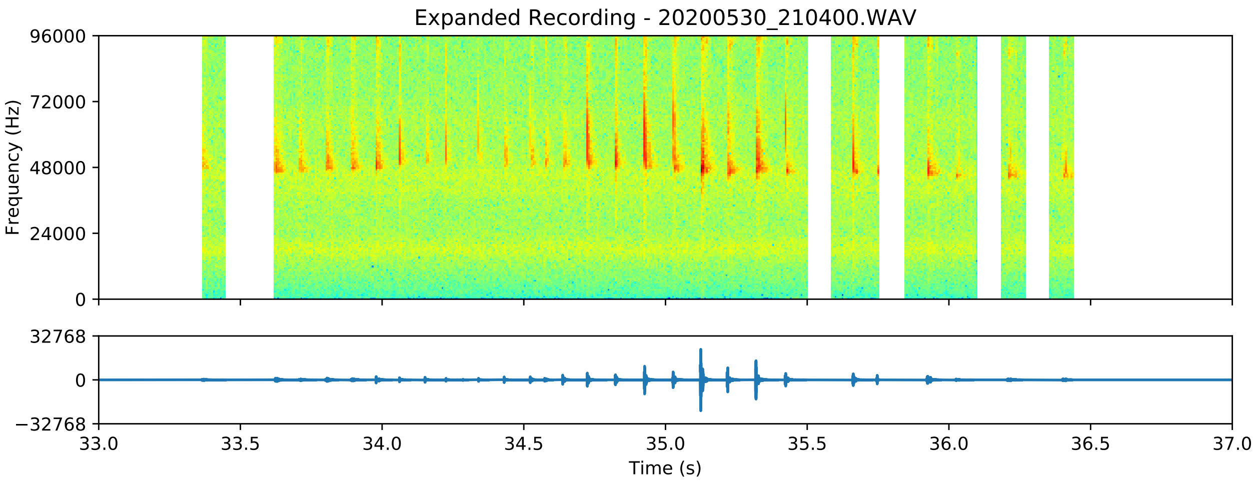

Figure 11: *Expanded* 20200530_210400.WAV *file zoomed in over the bat pass.*