TOOLS & TECHNIQUES IN BEHAVIOURAL BIOACOUSTICS

Embedded Ultrasonics: A Microcontroller-Based Multichannel Ultrasound Recorder for Behavioural Field Studies

Ravi Umadi

[1]Lehrstuhl für Zoologie, TUM School of Life Sciences, Technische Universität München, Liesel-Beckmann-Str. 4, 85354, Freising-Weihenstephan, Germany

Correspondence
Ravi Umadi,
Email: ravi.umadi@tum.de

Abstract

1. Behavioural studies investigating acoustic communication in animals—particularly echolocating bats—require transducers and recording systems that are lightweight, power-efficient, and easy to deploy. However, capturing high-frequency calls typically demands specialised, high-cost equipment and laptops, which limit portability and scalability in field settings. Advances in embedded microcontrollers and high-fidelity MEMS microphones offer an underexplored opportunity for building compact, affordable, and open-source recorders. These platforms provide design flexibility and a growing support community, but are seldom adapted for multi-channel ultrasonic recording in behavioural research.
2. To address this gap, I developed **Batsy4-Pro**, a 4-channel ultrasonic recorder based on the Teensy 4.1 microcontroller. The system uses WM8782 ADCs to acquire synchronised 192 kHz audio streams and records them to a microSD card. The firmware is written in C++ on Arduino IDE and allows flexible configuration of recording parameters, buffering schemes, and triggering logic. The system weighs under 150 g and runs on a 5 V DC power supply, enabling untethered field deployment.
3. The system also integrates real-time heterodyne monitoring using a PCM5102A DAC, allowing users to audibly monitor bat activity during field experiments, reducing the need for additional equipment and helping field researchers decide when to initiate recordings.
4. Using a custom-built microphone array, I validated its recording fidelity using both synthetic bat call playbacks under controlled conditions and free-flying bat vocalisations in a natural foraging corridor. Recordings consistently showed high signal-to-noise ratios (>40 dB), suitable for accurate call detection and spatial localisation.
5. Batsy4-Pro offers an accessible and extensible tool for ultrasonic recording in behavioural, ecological, and neuroethological research, supporting studies where portability, ease of use, and field-readiness are essential. The open-source code and modular design facilitate community-driven development and promote the system's adoption, enabling the evolution of experimental protocols and recording paradigms, as well as the exploration of novel approaches to behavioural experiments in natural settings.

KEYWORDS
Microphone Array, Ultrasound Field Recorder, Embedded System Audio, Multichannel Ultrasound, Behavioural Bioacoustics, Field Experiments

1 | INTRODUCTION

Scientific progress often hinges on the availability of cost-effective, portable, and easily customisable tools. In recent years, the growing ecosystem of open-source electronics has played a pivotal role in enabling non-commercial development of field instruments, acting as a bridge between engineering innovation and ecological research (Pearce, 2020; Oellermann et al., 2022). These community-supported platforms empower scientists to prototype, iterate, and deploy experimental

Abbreviations: MCU, Microcontroller Unit; I2S, Inter-IC Sound; ADC/DAC, Analogue to Digital, Digital to Analogue Converter

hardware tailored to specific scientific questions, especially in resource-constrained or remote settings (Ackermann, 2009; Pearce, 2012, 2015; Powell, 2015; GOSH, 2017).

Recent advances in embedded computing, particularly in high-performance microcontrollers, have opened new frontiers for field-based data acquisition. The domain of bioacoustics—especially studies involving ultrasonic signals in bats and insects—stands to benefit significantly from these developments. Despite the availability of compact recorders such as Solo (Whytock & Christie, 2017) and AudioMoth (Hill, Prince, Snaddon, Doncaster, & Rogers, 2019), and other open-source projects such as Raspberry Pi Bat Projekt (NABU, 2016), most remain limited by proprietary hardware and closed firmware, which constrain their adaptability for advanced experimental designs or integration with additional sensors. Even rarer are readily customisable designs and open-source acoustic device hardware that support multi-channel data recording of ultrasounds.

Furthermore, commercial recorders are often not amenable to in situ modification or rapid prototyping of new recording logic. This restricts the scope of field studies by limiting the kinds of experimental manipulations and real-time adaptations researchers can implement. There is thus a growing need for an open, programmable, and lightweight solution capable of high-fidelity ultrasonic recording under field conditions.

Behavioural field experiments—particularly those centred on bat bioacoustics and ultrasonic signal monitoring—face a persistent bottleneck: the lack of accessible, flexible, and scalable multichannel recording systems (Gaudette & Simmons, 2014). Ultrasonic microphones with sufficiently flat frequency responses beyond 100 kHz remain prohibitively expensive, and the associated data acquisition infrastructure—typically relying on external high-speed ADCs and tethered laptops—demands stable power supplies and cumbersome field setups. These limitations pose significant logistical challenges for conducting high-resolution, multichannel ultrasound recordings in remote or field-based environments, especially during biodiversity assessments or long-term autonomous deployments. Although recent developments have introduced portable and simplified recording systems, they often lack the scalability or configurability required for multi-channel behavioural studies. Consequently, the scope of advanced acoustic field research has remained largely confined to a limited number of species in economically developed regions. In contrast, the most biodiverse tropical habitats remain underrepresented in ultrasonic bioacoustic research, with cost and infrastructure barriers constituting a major impediment to widespread deployment of sophisticated recording systems (Van Helden, 2012).

To address these limitations, I present **Batsy4-Pro**, an open-source, microcontroller-based, four-channel ultrasonic recorder developed specifically for behavioural and ecological field studies. The system is built on the Teensy 4.1 (Stoffregen & Coon, 2005) platform and supports sustained multi-channel recording at 192 kHz per channel using dual WM8782 ADCs and high-speed SD card storage. It features external PSRAM-based ring buffering and firmware-level support for event-triggered recording, including retrospective (buffer-dump) and prospective modes. The compact form factor, low power requirements, and modular design make Batsy4-Pro suitable for portable and autonomous deployments, including drone-based surveys and multi-node acoustic networks. The firmware, written in C++ using the Arduino IDE, supports customisable recording parameters, trigger logic, and the integration of auxiliary sensors such as GPS, IMUs, or environmental probes. This paper details the system architecture, buffering and file-writing implementation, validation procedures, and potential ecological, ethological, and neuroacoustic research applications.

A key feature of the system is its built-in *heterodyne audio monitoring*. This module generates a real-time down-converted audio stream centred around a tunable carrier frequency, enabling researchers to monitor ultrasonic bat audibly calls through standard headphones. The monitoring output is active continuously, including during simultaneous SD card recording, and provides immediate acoustic feedback that helps field researchers determine optimal moments to trigger recordings without relying on external detection hardware. Field deployments confirmed that the system reliably achieves high signal-to-noise ratios (>40 dB), supports precise call localisation, and delivers performance on par with commercial systems at a fraction of the cost.

These design features position Batsy4-Pro as a practical and versatile solution for multichannel ultrasonic recording in behavioural field studies. By combining modular open-source hardware with field-tested performance, Batsy4-Pro lowers the barrier to high-resolution acoustic data recording and experimentation. Its extensibility and openly available firmware offer a flexible foundation for community-driven development and adaptation to evolving research paradigms with bats and beyond.

Summary

This study presents an open-source, portable, multichannel ultrasound recorder designed for behavioural field research with echolocating bats and other vocally active animals. The system is built using readily available electronic modules popular among DIY enthusiasts, combining the cost-effectiveness of off-the-shelf hardware with extensive community support. Batsy4-Pro addresses a critical gap in behavioural bioacoustics research—the lack of portable multichannel ultrasound recorders that can be rapidly customised to accommodate evolving experimental protocols. The integrated live bat detector allows researchers to monitor real-time activity and trigger recordings after the event, capturing call sequences from several seconds prior.

The system is tested and validated, with field recordings of foraging bats, and a demonstration of effective localisation and reconstruction of flight paths.

Such an open-source approach to scientific research extends access to advanced techniques in economically disadvantaged regions of the world, where biodiversity is often understudied.

2 | METHODS

Overview

The Batsy4-Pro is a four-channel ultrasonic audio recorder built around the Teensy 4.1 microcontroller unit (MCU) (Stoffregen & Coon, 2005). It is designed for field deployment in behavioural studies of echolocating animals. The system records synchronised 16-bit audio at 192 kHz per channel and saves data as WAV files to an onboard microSD card. Two tactile buttons enable user control: one for triggering a retrospective 10-second recording from a ring buffer, and the other for initiating ongoing recording while held. The firmware is written in C++ using the Arduino IDE environment and leverages the PJRC Teensy Audio library (Stoffregen & Coon, 2015). The live heterodyning can be tuned via input tactile buttons to scan for bat activity in the expected frequency band.

2.1 | Components

The core hardware components include:

- Teensy 4.1 microcontroller unit (600 MHz ARM Cortex-M7, 16-bit audio support)
- Two APS6404L-3SQR PSRAM - 8MB (Soldered on Teensy 4.1)
- Two WM8782 stereo ADCs interfaced via I^2S1 and I^2S2
- PCM5102A-based DAC Module with a headphone jack
- Four Knowles SPU0410LR5H-QB analogue omnidirectional MEMS microphones (direct differential input to ADCs)
- Adafruit SSD1306 128×64 pixel OLED display (I^2C interface)
- FAT32-formatted microSD card inserted into the built-in Teensy slot for data storage
- Two tactile push buttons: For retrospective and prospective recording.
- A rotary encoder with switch function: Adjusting carrier frequency and output level for heterodyne monitoring module – switchable mode.

Power is supplied via USB from a portable battery pack. Refer to Appendix B for the bill of materials.

2.2 | Circuitry

Each ADC interfaces with the Teensy over dedicated I^2S lines. Audio connections are managed via 'AudioInputI2S' and 'AudioInputI2S2' classes. A pair of 'AudioRecordQueue' objects buffers each input channel. Audio buffers are written as interleaved 16-bit samples into a WAV file stream on the SD card.

2.3 | I^2S ADC Configuration

The system uses two WM8782 ADC boards, each capable of digitising stereo analogue signals at up to 24-bit / 192 kHz. These are configured to operate in slave mode, streaming data via I^2S to the Teensy 4.1's `I2S1` and `I2S2` ports simultaneously, enabling synchronised four-channel audio recording. Each ADC contributes two 16-bit stereo channels per frame, with the PJRC Audio library managing the dual I^2S streams.

The WM8782 is a 24-bit sigma-delta stereo ADC designed for high-fidelity audio capture. It supports sample rates from 8 kHz to 192 kHz and offers digital audio output in I^2S, left-justified, or right-justified format. Its performance includes a signal-to-noise ratio (SNR) of up to 100 dB and total harmonic distortion (THD) better than -93 dB at 48 kHz (Cirrus Logic, 2020).

Each ADC operates in *slave mode*, meaning that bit clock (BCLK) and word clock (LRCLK) are provided by the Teensy. The WM8782 automatically detects the sampling rate by locking onto these clock signals, with a built-in mechanism to determine the oversampling ratio (OSR) via the `FSAMPEN` pin. To match the 192 kHz operation of the system, `FSAMPEN` is set to high-impedance (Z), resulting in 32× oversampling (Cirrus Logic, 2020).

2.3.1 | ADC Configuration Pins

The WM8782 is fully hardware-configured via digital input pins as follows:

- `FORMAT` (Pin 9): set to high-impedance (Z) for I^2S format
- `IWL` (Pin 7): set to low-impedance (Z) for 16-bit word length
- `M/S` (Pin 20): tied to logic low (0) for slave mode
- `FSAMPEN` (Pin 8): set to high-impedance (Z) to enable 192 kHz sampling (32× OSR)

This configuration results in each WM8782 transmitting 16-bit I^2S audio data synchronised to the Teensy's clocks, ensuring sample-accurate alignment across all four channels.

2.4 | I^2S Sampling Rate Configuration

Precise sampling rate configuration is critical in multichannel ultrasonic systems to ensure temporal synchrony across input channels and to preserve waveform fidelity at high frequencies. To achieve this, both the software sampling definitions and the Teensy's hardware phase-locked loop (PLL) clocks are explicitly configured to deliver a stable and exact 192 kHz sampling rate across both I^2S interfaces.

2.4.1 | Audio Library Sampling Rate

To ensure the Teensy Audio library internally aligns with the desired high-resolution sampling, the sampling rate is explicitly defined using:

(a) Batsy4 Pro, ultrasound recorder, assembled on a perforated PCB rack. (b) Connector hub and tripod mount (c) Assembled array on tripod.

FIGURE 1 Batsy4-Pro system and field array configuration. (a) The Batsy4-Pro recorder, assembled on a perforated PCB board, features a Teensy 4.1 MCU, dual WM8782 ADCs (bottom layer, invisible), OLED display, PCM5102A DAC with headphones output, and a 6-pin connector. The rotary encoder on the side, via switchable mode, tunes the heterodyning carrier frequency and adjusts output level through the DAC. (b) A rear view of the array hub shows the 6-way connector, internal wiring, and custom-machined aluminium mounting rod with a 1/4" tripod thread, attached to a hot-shoe ball head for angle adjustment. (c) Fully assembled cross-type microphone array mounted on a tripod, deployed at a field site for recording bat activity.

```
#undef AUDIO_SAMPLE_RATE_EXACT
#define AUDIO_SAMPLE_RATE_EXACT 192000.0f
```

This directive ensures that all internal buffers, sample timing, and I^{2}S objects are configured precisely for 192 kHz sampling before any audio object initialisation.

2.4.2 | Master Clock Settings

To achieve a stable sampling rate of 192 kHz per channel, a custom clock configuration is implemented through the `setI2SFreqBoth()` function (Appendix A.1). This function configures the internal phase-locked loop (PLL) on the Teensy 4.1 to generate the precise master clock required for high-rate I^{2}S communication on both SAI1 and SAI2 interfaces (NXP Semiconductors, 2021).

The function first calculates the necessary PLL multiplier and divider settings for a given sampling frequency. The core calculation derives from the formula:

$$C = \frac{f_{\text{sample}} \times 256 \times n_1 \times n_2}{24{,}000{,}000}$$

Where f_{sample} is the target sampling frequency (192 kHz), n_1 is the SAI prescaler (set to 4), and n_2 is the PLL divider computed to approximate the required master clock. The fractional components of C are then passed to the `set_audioClock()` function, which configures the audio clock source with the appropriate numerator and denominator.

After configuring the PLL, the function sets the I^{2}S peripheral clock dividers (`CCM_CS1CDR` and `CCM_CS2CDR`) for both I^{2}S1 and I^{2}S2. This ensures that both stereo input channels operate synchronously at the same sample rate.

This configuration guarantees a sampling rate of exactly 192000 Hz for all four input channels (two per I^{2}S interface), maintaining synchrony and timing accuracy required for multichannel ultrasonic recordings. See Chapters 14 and 38 of the i.MX RT1060 Corssover Processor reference manual for a detailed system description (NXP Semiconductors, 2021).

2.5 | Memory Buffer Architecture and Dual-Mode File Recording

The recording system supports two complementary data capture modes: (i) *Retrospective buffer-triggered recording*, which saves a fixed-duration window of past audio using a real-time ring buffer, and (ii) *Forward-triggered recording with buffer prefill*, which records ongoing audio while seamlessly appending a user-defined segment of recent history. These modes enable flexible behavioural monitoring with minimal data loss, even during spontaneous or unpredictable acoustic events.

2.5.1 | Ring Buffer

A continuously running ring buffer is implemented in the external PSRAM (`EXTMEM`) attached to the Teensy 4.1 microcontroller. Incoming multi-channel audio data—sampled at 192 kHz from four synchronised input channels via two WM8782 ADCs interfaced over `I2S1` and `I2S2`—is acquired in 128-sample blocks via `AudioRecordQueue` objects. Each sample is 16 bits, interleaved per frame to yield an effective throughput of:

$$192{,}000\,\text{samples/s} \times 4\,\text{channels} \times 2\,\text{bytes/sample} = 1.536\,\text{MB/s}.$$

A ring buffer of 5 seconds is maintained, requiring 7.68 MB of memory—nearly half of the available PSRAM. Samples are continuously written to this buffer in real time using a write index that tracks the buffer length. A volatile counter tracks the cumulative samples to ensure that pre-trigger segments can be reliably retrieved after startup or under rapid-trigger conditions.

2.5.2 | Retrospective Recording

Upon a tap-button trigger, the last 5 seconds of buffered audio are immediately written to a temporary WAV file (`/TEMP.WAV`). The ring buffer contents are copied in small non-blocking chunks (128–512 samples) interleaved with real-time audio processing to prevent loss of data continuity (Appendix A.2). After all buffered samples are saved, a proper 44-byte WAV header is inserted at the start of the file, and the file is renamed to a unique identifier (e.g., `/BAT001.WAV`).

2.5.3 | Forward Recording with Prefill

A second input initiates a buffered-forward recording mode. To preserve the context leading up to the trigger, the first segment of the file contains the last 5 seconds of buffered audio. This is written incrementally using an interleaved streaming approach: buffered data is flushed in small, non-blocking steps while audio capture and ring buffer updates continue uninterrupted.

Following this, up to 10 (user-defined) seconds of ongoing audio is continuously recorded into the same file in real time, with periodic checks to the ring buffer write pointer to determine available data for flushing. This allows real-time capture without overflows or gaps. Recording ends automatically after a timeout or on user input, at which point remaining buffered samples are written and the file is finalised.

This dual-mode architecture provides retrospective surveillance and prospective capture capabilities with full multi-channel fidelity. All audio processing and file writing are handled within a single cooperative main loop without interrupts or blocking calls. The design is optimised for low-latency response in field deployments, such as capturing bat calls, animal vocalisations, or short-duration acoustic cues in natural environments.

2.6 | Integrated Heterodyne Monitoring

The system incorporates a continuous heterodyne monitoring subsystem to assist researchers in identifying bat activity in real time during field recordings. This feature enables auditory monitoring of ultrasonic bat calls via live frequency down-conversion, eliminating the need for separate detection devices in the field.

2.6.1 | Implementation

The heterodyne output is produced by multiplying the incoming ultrasonic waveform from any user-defined channel with a continuous sine wave of user-adjustable frequency. The carrier frequency, which defaults to 45 kHz, can be dynamically adjusted between 10–85 kHz using a rotary encoder during runtime. This allows flexible auditory exploration of frequency bands of interest, particularly useful when working with species that produce calls outside a fixed spectral window.

The sine wave is generated internally in real time using a phase accumulator approach, where a phase variable is incremented each cycle according to a precomputed *phase increment*, determined by the desired carrier frequency and the fixed 192 kHz sampling rate (see Appendix A.2). The signal from the selected channel is then heterodyned by point-wise multiplication with the sine carrier, effectively folding spectral content symmetrically around the carrier frequency. This shifts ultrasonic components into the human-audible range. For example, a 62 kHz component with a 60 kHz carrier produces a 2 kHz tone at the output.

The heterodyned signal is constrained to 16-bit signed integers and transferred via a `AudioPlayQueue` to the `AudioOutputI2S` interface for playback. A ring buffer continues to record all raw multi-channel data in parallel, ensuring that heterodyne monitoring does not interfere with data acquisition.

2.6.2 | Output

Heterodyne audio is delivered via a high-fidelity PCM5102A (Texas Instruments, 2015) digital-to-analogue converter (DAC), connected to the `I2S1` interface of the Teensy 4.1. This DAC supports the full 192 kHz output rate and ensures low-distortion, high-dynamic-range conversion suitable for real-time headphone or speaker monitoring. The output level rotary switch encodes a gain variable that amplifies the audio output buffer before playing, thus effectively acting as a volume control.

2.6.3 | Operational Continuity

Heterodyne output always remains active, including during ongoing recording operations in retrospective and forward-triggered modes. This allows the researcher to monitor bat activity acoustically without interruption, enabling rapid and informed decision-making in the field.

Audible monitoring of call presence, rhythm, and approximate pitch provides a reliable and intuitive means to assess when echolocation activity begins or ends, guiding the timing of manual trigger events.

The need for additional monitoring equipment—such as handheld bat detectors or real-time spectrogram interfaces—is eliminated by embedding heterodyne functionality into the device. This reduces the logistical and power burden during field deployments, while ensuring that critical acoustic events remain observable.

3 | ARRAY ASSEMBLY AND DEPLOYMENT

The microphone array consists of four Knowles SPU0410LR5H-QB analogue MEMS microphones with omnidirectional sensitivity, mounted at the ends of hollow aluminium arms arranged in a 3D cross configuration (Figure 1c). Each arm is made of 1 cm^2 square-section aluminium rod, inserted into a central six-way connector hub. The microphones are wired internally along each arm, and all connections terminate at a single 6-pin connector located at the hub, enabling fast and secure attachment to the recorder.

The central hub features a rear-mounted 15 cm solid aluminium rod (1 cm^2 cross-section) with a custom-machined 1/4-inch tripod thread. This allows attachment to a standard ball-head adapter with a hot shoe plate mounted on a conventional tripod (Figure 1b). The assembled array, including the connector and mounting rod, weighs approximately 800 grams and is easily transportable.

During deployment, the arms are pulled out of the hub and inserted as needed, enabling rapid assembly and disassembly in field conditions. Each arm is fitted with tailored sleeves made of acoustically transparent insulation material made of polyester fur.

The array geometry enables spatial localisation of ultrasonic calls when combined with synchronised four-channel recording. The precise configuration and rigid construction ensure repeatable microphone placement and stable performance during field sessions.

4 | VALIDATION

Laboratory recording, testing, and validation were conducted using synthetic FM chirps played back through an RME Babyface audio interface, amplified via a single-channel power amplifier, and reproduced with a Scanspeak tweeter. The output was uncalibrated, but the level was estimated to be comparable to normal speech, at approximately 70 dB SPL. As expected, the recordings exhibited a complex pattern of reflections from surrounding objects. Nevertheless, these conditions confirmed proper system functionality and readiness for deployment (Figure 2a).

To validate a use case and demonstrate the functional deployment of the Batsy4-Pro system, a field experiment was conducted at a location known for frequent bat foraging activity. The array was mounted on a tripod and deployed in a semi-open corridor clearing (Figure 1c).

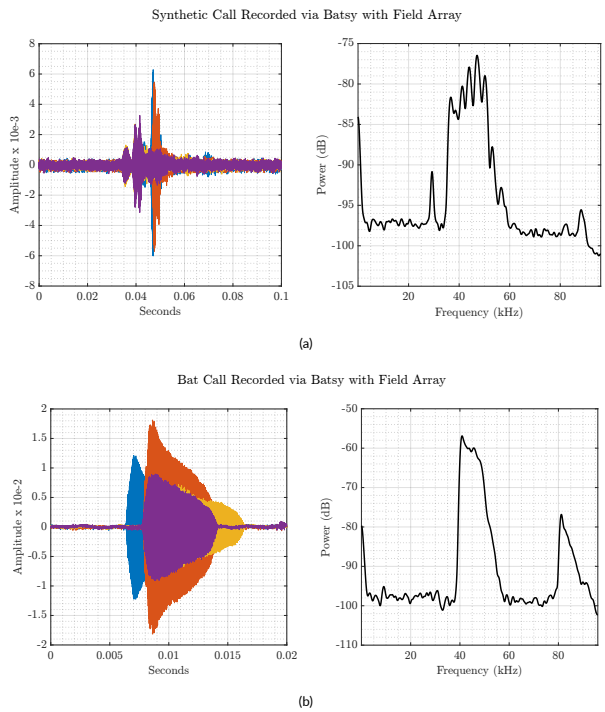

FIGURE 2 Top: A computer-generated quadratic FM chirp, recorded indoors without acoustic insulation on either the array or the room, resulting in spectral interference from multipath reflections. The sound source was positioned approximately 2 m from the microphones, with an unknown source level estimated to be comparable to normal speech (≈ 70 dB SPL at 10 cm). **Bottom:** A representative bat echolocation call recorded with the Batsy4-Pro system using the 4-channel field array. The waveform on the left shows the time-domain signal from 4 channels, and the corresponding power spectrum on the right is calculated from the first channel. The spectrum reveals a prominent third harmonic of the echolocation call.

Approximately 20 distinct foraging sequences characterised by rapid echolocation call bouts and frequency-modulated signals were recorded. Recordings were manually screened for high-quality multi-channel segments with clean signal-to-noise ratios and overlapping detections across all four microphones (Figure 2b). 20 ms regions around selected calls were extracted via custom-written MATLAB scripts, adopted from the `Biosonar Responsivity Analysis Toolkit V1.0` (Umadi, 2025b).

4.1 | Clustering and Localisation Methods

I implemented a combination of spectral clustering and time-difference-of-arrival (TDoA) based localisation to analyse the diversity and spatial structure of the recorded echolocation calls.

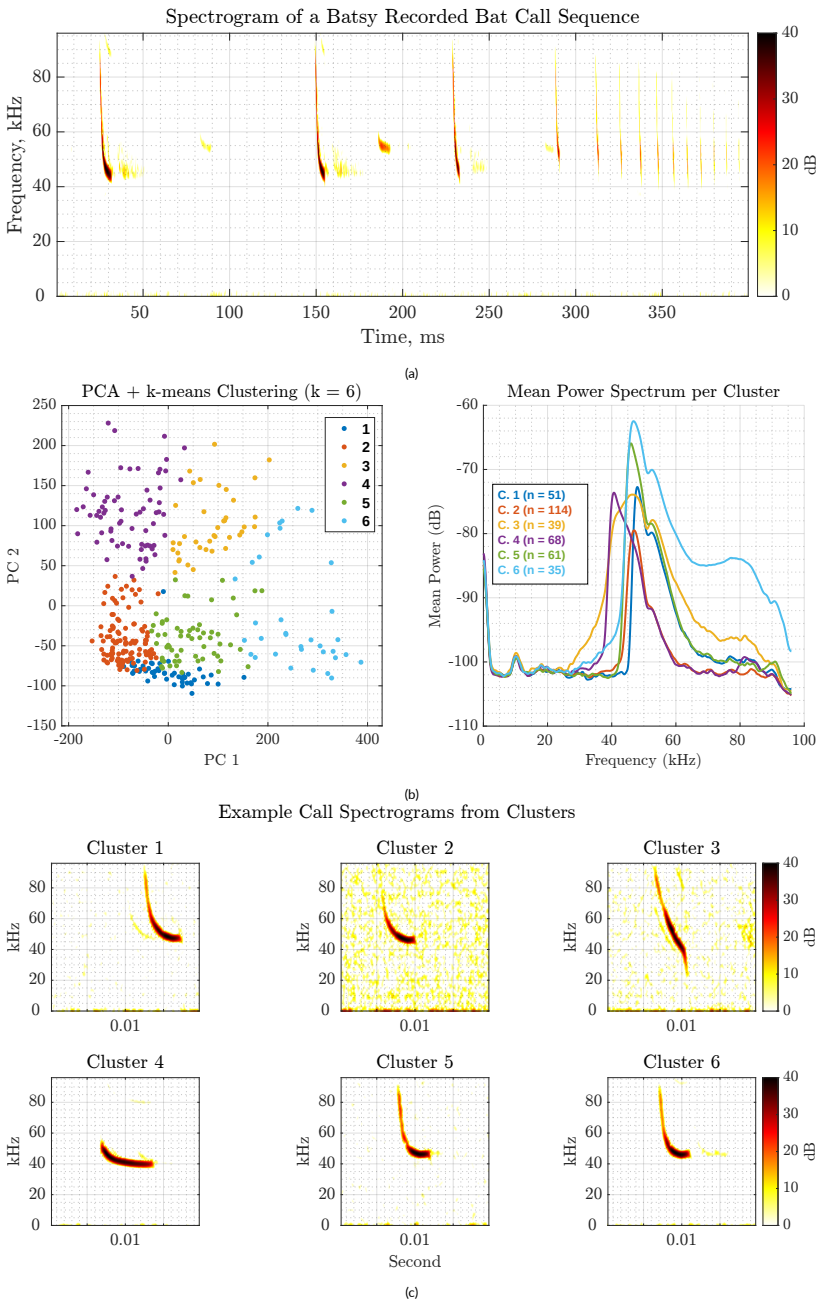

FIGURE 3 **System Validation:** (**a**) Spectrogram of a sequence of bat calls recorded on the Batsy4-Pro system at a foraging site along the Isar river, Freising, Germany (different day and site than the rest of the data). (**b**) The Clustering analysis of individual calls' power spectra using PCA and k-means clustering, with the optimal number of clusters ($k = 6$) selected via the Davies-Bouldin criterion. The right panel shows the mean power spectrum of each cluster, revealing spectral profiles. (**c**) Example spectrograms of individual calls sampled from each cluster, demonstrating the diversity in spectral shape and frequency content.

4.1.1 | Call Spectral Clustering.

For each recorded call, the first channel was windowed using a Hann function and its power spectral density was estimated using MATLAB's `pspectrum` function with 1 kHz resolution. The resulting power spectra (in dB) were truncated to a fixed length of 1024 bins and stored in a matrix. Principal Component Analysis (PCA) was used to reduce dimensionality before clustering. The optimal number of spectral clusters (between 2 and 6) was automatically determined using the Davies-Bouldin criterion (Davies & Bouldin, 1979) via the `evalclusters` function. Final

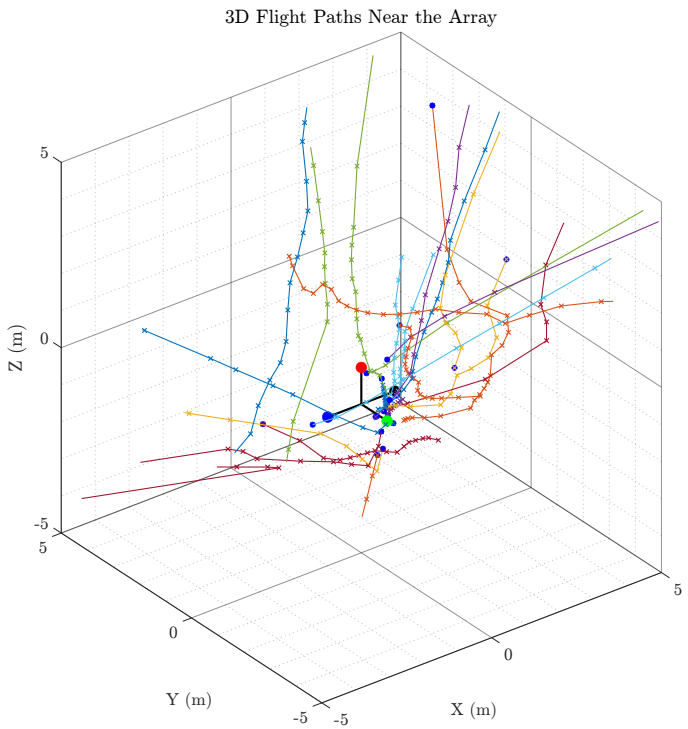

FIGURE 4 The array was deployed at a bat foraging site, and calls were recorded manually by monitoring the real-time heterodyne output of the Batsy4-Pro system. Using time-difference-of-arrival (TDoA) localisation, the 3D positions of calls were estimated. The flight paths are visualised within ± 5 m in all directions for clarity. Microphone positions are shown as large coloured dots (RGBK). The blue dots mark the final call detected in each sequence, corresponding to the last reliably localisable event. The trajectories suggest that bats were actively approaching the array.

clustering was performed using *k*-means on the first ten principal components. Each group's cluster means and example spectrograms were visualised to assess call diversity qualitatively.

4.1.2 | 3D Localisation via TDoA.

To estimate the 3D position of the bat at each call instance, I used a time-difference-of-arrival method. A custom function (`estimateTDOA`) extracted arrival delays between microphones using cross-correlation between all microphone pairs. These delays were then passed to `localiseTDOA`, which performed multilateration based on known microphone positions and the assumed speed of sound ($c = 343$ m/s). The functions are part of the `Array WAH - Microphone Array Design Tool` (Umadi, 2025a, 2025c). Only localisation results with complete four-channel data and valid arrival times were retained.

4.1.3 | Trajectory Visualisation.

Localised positions were grouped by recording segment (i.e., continuous sequences of calls within a foraging bout), and smoothed trajectories are represented in a 3D plot in Figure 4. The last detected call in each sequence is marked with a blue dot to indicate the endpoint. For clarity, axes are limited to ± 5 metres. This visualisation allowed direct interpretation of bat approach trajectories relative to the array.

5 | EXTENSIBILITY

5.1 | Channel Extensibility

The Batsy4-Pro architecture is inherently extensible for multichannel audio acquisition. The Teensy 4.1 microcontroller supports two hardware I^2S interfaces (I^2S1 and I^2S2), each capable of handling stereo input and outputs, with I^2S1 supporting up to 3 stereo in-out configurations.. With the maximum configuration, up to 8 channels may be employed via compatible ADCs. Further, the Teensy 4.1 also supports Time Division Multiplexing (TDM), which can expand inputs up to 16 channels with TDM-compatible ADCs (Stoffregen & Coon, 2005).

5.2 | Sampling Rate Expansion

While the current implementation supports 192 kHz across four channels, the design is compatible with higher sampling rates through other ADCs. The AK5572 v2 ADC from Kaamos Tech is one such option, offering dual-channel 32-bit recording at up to 768 kHz (Nihtila, 2023).

A prototype of Batsy supporting 384 kHz sampling is under active development. This opens the system to ultrasonic recording applications that demand higher temporal resolution, such as particular species of bat calls that go well over 120kHz, or close-range recording of higher harmonics.

5.3 | Adjunct Sensors and Environmental Data Logging

The Teensy 4.1 offers broad support for serial, I^2C, and SPI peripherals, making it ideal for integrating adjunct sensors. These include:

- GPS modules (for geolocation and time stamping)
- Temperature, humidity, and barometric sensors
- Ambient light sensors (for diurnal/nocturnal activity correlation, etc.)
- Anemometers (to monitor wind and atmospheric noise)
- Passive IR or photogates (for motion-based triggering)

Incorporating such sensors can enhance acoustic data by providing contextual metadata, correcting environmental bias (e.g., atmospheric attenuation) and improving the ecological precision of behavioural interpretations.

6 | DISCUSSION

Most tools used in bioacoustic fieldwork—handheld recorders, AD/DA converters, and microphones—originate from consumer audio technology. While such equipment is often sufficient for studying audible sounds (e.g., birds, insects, primates), it falls short for recording high-frequency ultrasonic vocalisations. Despite their necessity in behavioural ecology, there is little commercial incentive to develop multichannel ultrasound recorders for research. Even specialised systems (e.g., Wildlife Acoustics, Dodotronics) are typically designed for passive ecological monitoring and are not readily adaptable to experimental paradigms.

Behavioural studies on echolocating bats have used high-end recording chains to reconstruct flight paths, estimate beam patterns, and investigate call dynamics (Gaudette & Simmons, 2014; Jens, 2018). However, such deployments remain constrained by cost, portability, and technical setup. Batsy4-Pro was developed to address these limitations. Built from widely available components and open-source firmware, it offers a modular, affordable, and lightweight alternative for multi-channel ultrasonic recording in the field.

The inclusion of real-time heterodyne monitoring allows users to detect bat activity aurally during deployment without requiring separate monitoring equipment. This facilitates informed, manual triggering of recordings and reduces redundant data collection. The hybrid ring-buffer architecture ensures that the 5-second context before triggering is also saved, improving the relevance of captured data in behavioural contexts.

Batsy4-Pro is extensible in both hardware and software. Channel count, microphones, sampling rates, sensor integrations, and trigger protocols can all be adapted. The entire firmware is provided as a single-file Arduino sketch to minimise technical barriers, avoiding the complexity of advanced development environments such as PlatformIO.

6.1 | Limitations

While both the hardware design and firmware for Batsy4-Pro are openly available for non-commercial use, a notable barrier to widespread adoption is the requirement of technical familiarity with embedded electronics and programming. Although detailed assembly and setup instructions are provided in the online documentation and GitHub repository, successful implementation benefits from prior experience with Arduino-based modules and a basic understanding of I^2S audio interfaces.

For students and researchers in biology or ecology—many of whom may not have formal training in electronics, signal processing, or soldering—an approachable entry point lies in beginner-level Arduino projects. The abundance of freely available tutorials and a large and active user community make self-guided learning accessible and rewarding.

Customising the firmware to suit project-specific requirements can still pose a challenge for field researchers. However, interdisciplinary collaboration can address this hurdle by engaging participants with programming expertise, such as students or citizen scientists. While these requirements may initially appear demanding, they represent an opportunity to foster cross-disciplinary learning—an ethos that aligns well with the inherently interdisciplinary nature of bioacoustics research.

Moreover, compared to commercial systems, the cost-effectiveness and flexibility offered by Batsy4-Pro provide strong justification for the learning investment. The skills gained through system assembly and modification will likely benefit future research endeavours and experimental innovation.

Batsy4-Pro exemplifies the broader movement towards open and decentralised ecological instrumentation. Its affordability, flexibility, and accessibility make it well-suited for decentralised research, particularly in biodiversity-rich regions underserved by commercial tools. As ultrasonic monitoring expands across behavioural, ecological, and neuroethological disciplines, systems like Batsy4-Pro will be key in democratising high-resolution field data collection.

7 | CONCLUSION

Batsy4-Pro provides a flexible, low-cost, and field-ready solution for high-fidelity multichannel ultrasonic recording, addressing critical limitations in existing commercial and custom-built systems. Its open-source design, compact form factor, and real-time heterodyne monitoring make

it a versatile tool for behavioural, ecological, and neuroethological research. By combining modular hardware with accessible firmware, the system enables scalable deployment in diverse environments while promoting a community-driven approach to bioacoustic instrumentation. The system thus represents a practical and extensible platform for advancing acoustic research in previously under-resourced contexts.

CODE AND DATA AVAILABILITY

The source code, its derivative versions with other MCU platforms such as ESP32, and derived applications can be found on the GitHub Repository https://github.com/raviumadi/Embedded_Ultrasonics.

Ongoing development of the embedded ultrasound system, including the extensions described in this work, will be documented and published via the repository.

FINANCIAL DISCLOSURE

This project was conducted without external funding, with the author covering all expenses.

CONFLICT OF INTEREST

The author declares no conflict of interest.

ACKNOWLEDGEMENTS

I want to thank Uwe Firzlaff for his valuable comments during the development of the recording system, and Christian Fink for his assistance with the machining work involved in building the array.

References

Ackermann, J. R. (2009). Toward Open Source Hardware. *University of Dayton Law Review*, *34*(2).

Cirrus Logic, INC. (2020). WM8782 - 24-bit 192 kHz Stereo ADC Datasheet. *Cirrus Logic*.

Davies, D. L., & Bouldin, D. W. (1979, April). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. doi: 10.1109/TPAMI.1979.4766909

Gaudette, J. E., & Simmons, J. A. (2014, September). Observing the Invisible Using Microphone Arrays to Study Bat Echolocation..

GOSH. (2017, December). *Global Open Science Hardware Roadmap - Gathering for Open Science Hardware.*

Hill, A. P., Prince, P., Snaddon, J. L., Doncaster, C. P., & Rogers, A. (2019, October). AudioMoth: A low-cost acoustic device for monitoring biodiversity and the environment. *HardwareX*, *6*, e00073. doi: 10.1016/j.ohx.2019.e00073

Jens, K. (2018). Arrayvolution: Using microphone arrays to study bats in the field.

NABU, e. (2016). *Raspberry Pi Bat Projekt* [Hardware-Software Product]. http://bat-pi.eu/.

Nihtila, T. (2023, July). *ADC AK5572 v2 - Very high performance AD-converter in a compact form factor. https://kaamostech.com.*

NXP Semiconductors, N. V. (2021). *I.MX RT1060 Processor Reference Manual R3*.

Oellermann, M., Jolles, J. W., Ortiz, D., Seabra, R., Wenzel, T., Wilson, H., & Tanner, R. L. (2022, May). Open Hardware in Science: The Benefits of Open Electronics. *Integrative and Comparative Biology*, *62*(4), 1061–1075. doi: 10.1093/icb/icac043

Pearce, J. M. (2012, September). Building Research Equipment with Free, Open-Source Hardware. *Science*, *337*(6100), 1303–1304. doi: 10.1126/science.1228183

Pearce, J. M. (2015, January). Open-Source Lab How to Build Your Own Hardware and Reduce Research Costs. Elsevier, New York. - References - Scientific Research Publishing. *Modern Economy*, *6*(1).

Pearce, J. M. (2020, October). Economic savings for scientific free and open source technology: A review. *HardwareX*, *8*, e00139. doi: 10.1016/j.ohx.2020.e00139

Powell, A. B. (2015, April). Open culture and innovation: Integrating knowledge across boundaries. *Media, Culture & Society*, *37*(3), 376–393. doi: 10.1177/0163443714567169

Stoffregen, P., & Coon, R. (2005). *Teensy® 4.1* [Open Source Electronics Products]. https://www.pjrc.com/store/teensy41.html.

Stoffregen, P., & Coon, R. (2015). *Teensy Audio Library.* PJRC.COM, LLC. Sherwood, OR. USA.

Texas Instruments, INC. (2015). PCM510xA 2.1 VRMS, 112/106/100 dB Audio Stereo DAC with PLL and 32-bit, 384 kHz PCM Interface.

Umadi, R. (2025a, June). *Array WAH: Microphone Array Design Tool.* Zenodo. doi: 10.5281/zenodo.15691372

Umadi, R. (2025b, July). *Biosonar Responsivity Analysis Toolkit V1.0.0.* Zenodo. doi: 10.5281/zenodo.15801316

Umadi, R. (2025c, June). *Widefield Acoustics Heuristic: Advancing Microphone Array Design for Accurate Spatial Tracking of Echolocating Bats.* bioRxiv. doi: 10.1101/2025.06.03.657701

Van Helden, P. (2012, May). The cost of research in developing countries. *EMBO reports*, *13*(5), 395–395. doi: 10.1038/embor.2012.43

Whytock, R. C., & Christie, J. (2017). Solo: An open source, customizable and inexpensive audio recorder for bioacoustic research. *Methods in Ecology and Evolution*, *8*(3), 308–312. doi: 10.1111/2041-210X.12678

How to cite this article: Umadi, R, Embedded Ultrasonics: A Microcontroller-Based Multichannel Ultrasound Recorder for Behavioural Field Studies. (August 2025). *BioRxiv Pre-published.* doi:

APPENDIX

A CODE: I²S CONFIGURATIONS

A.1 I²S Clock Setting

LISTING

1 ²S clock configuration for 192 kHz sampling on Teensy 4.1.

```
void setI2SFreqBoth(int freq) {
        // SAI prescaler (must be 1-8)
        int n1 = 4;

        // PLL divider
        int n2 = 1 + (24000000 * 27) / (freq * 256
              * n1);
        double C = ((double)freq * 256 * n1 * n2)
             / 24000000;
        int c0 = (int)C;
        int c2 = 10000;
        int c1 = (int)(C * c2) - (c0 * c2);

        set_audioClock(c0, c1, c2, true);

        // Set SAI1 (I2S1) and SAI2 (I2S2) clock
             dividers
        CCM_CS1CDR = (CCM_CS1CDR &
        ~(CCM_CS1CDR_SAI1_CLK_PRED_MASK |
        CCM_CS1CDR_SAI1_CLK_PODF_MASK)) |
        CCM_CS1CDR_SAI1_CLK_PRED(n1 - 1) |
        CCM_CS1CDR_SAI1_CLK_PODF(n2 - 1);

        CCM_CS2CDR = (CCM_CS2CDR &
        ~(CCM_CS2CDR_SAI2_CLK_PRED_MASK |
        CCM_CS2CDR_SAI2_CLK_PODF_MASK)) |
        CCM_CS2CDR_SAI2_CLK_PRED(n1 - 1) |
        CCM_CS2CDR_SAI2_CLK_PODF(n2 - 1);
}
```

A.2 Ring Buffer in PSRAM

LISTING 2 Ring Buffer Implementation

```
const int sampleRate = 192000;
const int bytesPerSample = 2;
const int numChannels = 4;
const int ringBufferSeconds = 5;
const uint32_t samplesPerChannel = sampleRate *
     ringBufferSeconds;
const uint32_t ringBufferSamples =
     samplesPerChannel * numChannels;

EXTMEM int16_t ringBuffer[ringBufferSamples];
volatile uint32_t ringWriteIndex = 0;

// For heterodyning
const float carrierFreq = 45000.0f;
const float twoPi = 6.28318530718f;
float phase = 0.0f;
float phaseIncrement = twoPi * carrierFreq /
     sampleRate;

// In the main loop, fill the ring buffer with
     samples, along with real-time heterodyne on a
     channel
void readAndProcessAudio() {
        if (queueL1.available() && queueR1.
             available() && queueL2.available() &&
             queueR2.available()) {
                int16_t *ch1 = (int16_t *)queueL1.
                     readBuffer();
                int16_t *ch2 = (int16_t *)queueR1.
                     readBuffer();
                int16_t *ch3 = (int16_t *)queueL2.
                     readBuffer();
                int16_t *ch4 = (int16_t *)queueR2.
                     readBuffer();
                static int16_t tempOut[128];

                for (int i = 0; i < 128; i++) {
                        ringBuffer[ringWriteIndex
                             ++] = ch1[i];
                        ringBuffer[ringWriteIndex
                             ++] = ch2[i];
                        ringBuffer[ringWriteIndex
                             ++] = ch3[i];
                        ringBuffer[ringWriteIndex
                             ++] = ch4[i];
                        if (ringWriteIndex >=
                             ringBufferSamples)
                             ringWriteIndex = 0;
                        totalSamplesWritten += 128
                              * numChannels;

                        // Heterodyne channel 1
                        float carrier = sinf(phase
                             );
                        phase += phaseIncrement;
                        if (phase >= twoPi) phase
                             -= twoPi;
                        float mixed = ch1[i] *
                             carrier * gain;
                        tempOut[i] = constrain((
                             int16_t)mixed, -32768,
                              32767);
                }

                queueL1.freeBuffer();
                queueR1.freeBuffer();
                queueL2.freeBuffer();
                queueR2.freeBuffer();

                if (outputQueue.available()) {
                        int16_t *out = outputQueue
                             .getBuffer();
                        memcpy(out, tempOut,
                             sizeof(tempOut));
                        outputQueue.playBuffer();
                }
        }
}


// On trigger, dump ring buffer to SD card
void dumpRingBufferToSD() {
        startNewRecording();  // open temp file
        uint32_t idx = ringWriteIndex;
        for (uint32_t i = 0; i < ringBufferSamples
             ; i++) {
                int16_t sample = ringBuffer[idx
                     ++];
                file.write((uint8_t*)&sample, 2);
                sampleCount++;
                if (idx >= ringBufferSamples) idx
                     = 0;
        }
        closeWavFile();  // insert header and
             rename file
}
```

B BILL OF MATERIALS

TABLE B1 Bill of Materials for the Batsy4 Pro Recording System. The listed prices correspond to procurement costs in Germany during development between January and July 2025.

Item	Description	Count	Unit Cost (€)	Source
Teensy 4.1	Microcontroller	1	39.50	reichelt.de
WM8782	I2S ADC	2	14.90	Audio-phonics.fr
Knowles SPU0410LR5H-QB	MEMS Mics	4	0.50	Digikey.de
PC5102A	I2S DAC	1	7.99	Amazon.de
MP1584EN	DC-DC 3A Power Step-Down Adjustable Module Buck Converter	1	2.00	Amazon.de
SSD1306	OLED Display	1	5.25	AZDelivery.de
Push Buttons	User Input	2	0.50	Amazon.de
Rotary Encoder	User Input	1	1.50	Amazon.de
Circuit Switch	Power Supply ON/OFF	1	0.25	Amazon.de
Perforated PCB	System Base	3	1.50	Amazon.de
Hexnut Screws	PCB Stacking	Assorted	5.00	Amazon.de
Wiring	18/22 gauge coloured wires	Assorted	5.00	Amazon.de
6-Pin Connector	Mic power and signal connection	1	14.99	Amazon.de
CR2032	RTC Battery	1	1.50	Generic
Total			**120.28**	

TABLE B2 Bill of Materials for the Field Array. The material list is only suggestive; equivalent components may be used depending on array design and local availability.

Item	Description	Count	Unit Cost (€)	Source
Aluminium Square Hollow Profiles (1 cm^2)	Arms of the array	4	4.99	Hagebaumarkt
Aluminium Square Solid Profile (1 cm^2)	Holder, mounting rod	1	7.99	Hagebaumarkt
Ball Joint	Mounting on tripod (or attach to 1/4" screw mount)	1	13.50	Generic
1/4" Tripod Mount Hot Shoe Adapter	Between holder and ball joint	1	5.99	Amazon (generic)
6-way Plastic Connector	Holds the array arms and mounting rod	1	4.99	Hagebaumarkt
Polyester Fur	Acoustic insulation, echo suppression from arms	1 m^2	8.99	Amazon (generic)
Cables and Miscellaneous	Extension cables, adhesives, etc.	N/A	20.00	Local purchases
Total			**81.42**	