BATSY4-PRO: An Open-Source Multichannel Ultrasound Recorder for Field Bioacoustics

Ravi Umadi [1]

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

Abstract.

1. Behavioural studies of acoustic communication in animals—particularly echolocating bats—require lightweight, power-efficient recording systems robust under field conditions. High-frequency multichannel recordings typically use consumer-grade audio interfaces and laptops, limiting portability and reproducibility. Recent advances in embedded microcontrollers and MEMS microphones enable the development of compact, affordable, open-source alternatives, yet such platforms remain underutilised for multichannel ultrasonic research.

2. To address this gap, I developed Batsy4-Pro, a four-channel ultrasonic recorder based on the Teensy 4.1 microcontroller. The system uses WM8782 analogue-to-digital converters for synchronised 192 kHz audio recording to microSD storage. The firmware enables customisation of buffering, triggering, and recording modes and durations. The system weighs under 150 g and operates from a 5 V DC supply for reliable field deployment.

3. A key feature is real-time heterodyne monitoring via an integrated digital-to-analogue converter, providing audible down-conversion of ultrasonic calls through headphones. This allows researchers to assess activity during deployment and make informed decisions about when to record, thereby improving data relevance and experimental efficiency without additional bat detectors or spectrogram systems.

4. Performance was validated using synthetic bat calls and field recordings of free-flying bats. Analysis of 368 echolocation calls yielded a median maximum-channel SNR of 27.3 dB, matching that obtained with a professional-grade audio interface. The four-channel array enabled three-dimensional localisation via time-difference-of-arrival methods. Monte Carlo simulations were used to quantify localisation uncertainty as a function of source position and motion. Within the evaluated near-field region (<4 m), localisation accuracy was governed primarily by array geometry and range. Source velocity did not influence median localisation error; however, increasing flight speed systematically altered the shape of error distributions, increasing the occurrence of larger deviations beyond narrow accuracy thresholds. Together, these results provide practical guidance for selecting array geometry and defining usable operating volumes in experiments involving moving sound sources.

5. Batsy4-Pro provides an accessible platform for multichannel ultrasonic recording in behavioural and ecological research. Through open hardware, documented firmware, and performance characterisation, this system reduces technical barriers and promotes the adoption of spatial acoustic methods in field studies.

Keywords: bat echolocation, ultrasound monitoring, open hardware, multichannel recording, heterodyne audio, field bioacoustics, source localisation

Corresponding author: Ravi Umadi– Email: ravi.umadi@tum.de

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 [1, 2]. These community-supported platforms empower scientists to prototype, iterate, and deploy experimental hardware tailored to specific scientific questions, especially in resource-constrained or remote settings [3–7].

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 [8] and AudioMoth [9], and other open-source projects such as Raspberry Pi Bat Projekt [10], most remain limited by proprietary hardware, closed firmware, or single-channel architectures that constrain their adaptability for advanced experimental designs requiring spatial localisation and sound-source tracking. Readily customisable, fully open-source multichannel ultrasonic recording systems with integrated real-time monitoring capabilities remain notably absent from the available toolkit (see Table 1).

Behavioural field experiments on echolocating bats face particular bottlenecks due to the lack of accessible, flexible, and field-ready multichannel recording systems. Traditional laboratory setups typically require professional audio interfaces, external preamplifiers, laptop computers, and stable AC power—configurations that are cumbersome to deploy, expensive to replicate, and require substantial technical expertise to operate reliably in field conditions. Ultrasonic microphones with flat frequency responses beyond 100 kHz add further cost, and the cumulative expense of multichannel systems often exceeds several thousand euros. These barriers have concentrated experimental bioacoustics in well-funded laboratories in developed regions, while tropical and subtropical habitats—which harbour the most extraordinary bat diversity—remain critically underrepresented in behavioural research [11]. Of the more than 1300 known species of echolocating bats, fewer than 30 have been studied in experimental behavioural contexts, reflecting both the technical demands and economic constraints of the field. Beyond its immediate utility for behavioural field studies, Batsy4-Pro is well aligned with emerging needs in computational bioacoustics, where large, well-annotated, high-quality acoustic datasets underpin advances in automated detection, classification, and behavioural inference [12, 13].

To address these limitations, I present Batsy4-Pro, an open-source, microcontroller-based four-channel ultrasonic recorder designed explicitly for behavioural field experiments with echolocating animals. Built on the Teensy 4.1 platform [14], the system integrates all essential components—analogue-to-digital conversion, computational processing, data storage, and real-time heterodyne monitoring—into a compact device weighing less than 150 grams and operating from a standard USB power bank. The system records synchronised 16-bit audio at 192 kHz per channel using dual WM8782 ADCs, with data saved as timestamped WAV files to an on-board microSD card. External memory-based ring buffering enables dual-mode event-triggered recording: retrospective capture of the preceding 5 seconds and forward recording with prefill, which preserves behavioural context while continuing to record ongoing activity (terminology in

Table 2).

A key feature distinguishing Batsy4-Pro from passive acoustic monitoring devices is its integrated real-time heterodyne audio monitoring system. This firmware-based frequency down-conversion generates a continuous audible stream centred on a user-adjustable carrier frequency (10–85 kHz), allowing researchers to monitor ultrasonic bat activity through standard headphones during field deployment. The heterodyne output remains active during recording operations, providing immediate acoustic feedback that enables informed, manual triggering of recordings based on observed echolocation patterns—transforming passive monitoring into an active, researcher-guided experimental process. This eliminates the need for separate bat detectors or real-time spectrogram displays, reducing equipment complexity and power requirements in the field.

I validated Batsy4-Pro through both controlled playback experiments and field deployments with free-flying bats in natural foraging corridors. Recordings consistently demonstrated high signal-to-noise ratios (median 27.3 dB), comparable to professional-grade audio interfaces, and enabled accurate spatial localisation of echolocation calls in the near-field using time-difference-of-arrival methods. The total component cost, including recorder and four-channel microphone array, is approximately 200 euros when sourced locally. All hardware designs, circuit schematics, firmware source code, and assembly documentation are released under permissive open-source licensing[1] [15] that explicitly permits commercial production without royalty obligations, facilitating local manufacturing and institutional workshop projects.

Batsy4-Pro demonstrates that high-quality multichannel ultrasonic recording for experimental bioacoustics can be achieved using affordable, accessible, open-source hardware. By combining field-tested performance with modular extensibility and comprehensive documentation, this work aims to lower technical and economic barriers to behavioural research with echolocating animals, supporting more inclusive and geographically distributed participation in a field with implications extending from sensory neuroscience to conservation biology.

2 METHODS

This section describes the design, implementation, and validation of the Batsy4-Pro system, combining hardware architecture, firmware design, and analytical methods used to evaluate its performance in field-relevant conditions. I first detail the recorder's electronic design, signal acquisition pipeline, buffering strategy, and real-time heterodyne monitoring capabilities. I then describe the construction and deployment of the four-channel microphone array, followed by laboratory and field validation procedures. Finally, I outline the simulation-based framework used to quantify spatial localisation accuracy, providing a principled assessment of the system's usable operational volume for behavioural bioacoustics experiments.

2.1 Design and Implementation

The following section outlines the hardware architecture, signal-processing workflow, and firmware structure of the Batsy4-Pro ultrasonic recorder – provided to document the system's design and performance; familiarity with all technical aspects is not required for routine use or deployment. Definitions of technical terms, abbreviations, and the *Teensy Audio Library* components

[1] [OSHW] DE000165 | Certified open source hardware | oshwa.org

System	Ch.	Fs	Res.	Rec. Mode	RT-Mon.	Euro	Open	Primary use	Ref.
Batsy4-Pro	**4**	**192**	**16**	**Trig+ pre+post**	**Yes**	**125–200**	**Yes**	**Behavioural field experiments, spatial localisation**	**This work**
AudioMoth	1	8–384	16-bit	Continuous, sched	No	49–80	Yes	PAM, surveys	openacousticdevices, Hill *et al.* [9]
AURITA (dual)	1 (US) + 1 (aud)	500 (US)	16-bit	Trig (US), sched	No	300–400	Yes (design)	Combined aud/US mon.	Beason *et al.* [16]
Bat Pi 2	1	Up to 250	16-bit	Triggered	No	500–600	Partial	DIY, ext. Dodotronic mic	bat-pi
BattyBirdNET-Pi	1	256–384	16-bit	Cont. AI class.	Yes (web)	200–350	Yes (SW)	AI detection, ext. mic req	gh:rdz-oss
Pettersson D1000X	1	32–768	16-bit	Het, FD, TE	Yes (het, FD)	3,500–4,000	No	Professional handheld	batsound
Pettersson D500X Mk I & II	1	44.1, 300, 500	16-bit	Triggered, cont.	Yes (FD)	1,500–2,000	No	PAM, ext. mic.	batsound
Pettersson M500-384 (USB)	1	384	16-bit	USB (host req)	No	500–700	No	USB recording, portable	batsound
Pipistrelle (RPi Pico)	1	384	16-bit	Trig, passive	Yes (het, synth)	65–100	Yes (full)	Low-cost handheld, USB	pippyg
Ranger/Chorus /Swift etc	1/2	up-to 500	16-bit	Scheduled, continuous	No	5,00 - 1,500	No	PAM	Titley Sci.
Raspberry Pi WURB	1	192–500	16-bit	Trig, sched, TE	Yes (audio)	150–300	Yes (SW)	DIY, ext. USB mic req	gh:cloudedbats
Solo (RPi-based)	1–2	Up to 96	16-bit	Sched, cont.	No	150–250	Yes (full)	General bioacoustics	Whytock & Christie 2017
Ultramic Series	1	up-to 384	16-bit	USB (host req)	No/via app	200–500	No	USB rec, mobile devices	dodotronic
Walkabout /Scout	1	up-to 500	16-bit	Real-time active detection, rec.	Yes	1,200 - 1,950	No	Handheld.	Titley Sci.
Wildlife Acoustics SM4BAT-FS & Family	1	Up to 500	16-bit	Triggered, sched	No	1,000–1,500	No	PAM	wildlifeacoustics
Wisecorder	1-4	48 - 384	16-32 bit	Scheduled, triggered, profiles	No	600	No	Professional 4-ch recording, spatial loc., autonomous, ext. mic. req.	dodotronic
Custom soundcard+mic	1-12+	48-192/384	16-32 bit	Via Software, scripts etc	Possible	100's - 1000's	No	Professional multicahnnel, mixing, realtime audio. ext. mic.+computer req.	-

Table 1: Comparison of Ultrasonic Recording Systems for Bat Bioacoustics Research. The table compares Batsy4-Pro with commercially available and open-source alternatives across key performance and usability metrics. Batsy4-Pro offers unique combination of advantages with low-cost and fully open-source system that combines: (1) true multichannel (4-channel) synchronous recording at 192 kHz, enabling spatial localisation; (2) integrated real-time heterodyne monitoring for immediate acoustic feedback during field deployment; (3) dual recording mode architecture with retrospective and prospective capabilities for capturing spontaneous behavioural events; (4) modular, extensible firmware and hardware design; and (5) field-validated performance comparable to commercial systems. Unlike passive monitoring devices designed for autonomous ecological surveys, Batsy4-Pro is specifically engineered for active behavioural experiments requiring real-time decision-making, spatial resolution, and flexible triggering. This fills a critical gap in accessible tools for experimental bioacoustics research in resource-limited settings. *The costs are approximations based on publically available information at the time of writing. The set of features are subject manufacturer updates.*

are summarised in Table 2.

Batsy4-Pro is a four-channel ultrasonic audio recorder built around the Teensy 4.1 microcontroller unit (MCU) [14]. It is designed for field deployment in behavioural studies of

(a) (b) (c)

Figure 1: The Batsy4-Pro system and field deployment. (a) The recorder assembled on a perforated PCB, incorporating a Teensy 4.1 microcontroller, dual WM8782 analogue-to-digital converters (mounted on the reverse side), an OLED status display, a PCM5102A digital-to-analogue converter with headphone output (purple module), and a 6-pin array connector. A side-mounted rotary encoder is used to adjust the heterodyne carrier frequency or output level via a switchable control mode. At the same time, two push buttons enable *TAP* (retrospective buffer capture) and *HOLD* (buffered forward recording) modes. (b) Rear view of the array hub showing the 6-way connector, internal wiring, and an aluminium mounting rod with a 1/4" tripod thread, attached to a ball head for flexible orientation. (c) Fully assembled cross-type microphone array mounted on a tripod and deployed at a field site for bat recording. Upon power-up via a standard USB-C power bank and activation of the main switch, real-time heterodyne monitoring and continuous ring-buffer recording are initiated automatically. The compact, self-contained design enables rapid deployment, with the complete system assembled and operational in under five minutes.

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* [17]. The live heterodyning can be tuned via a rotary encoder to scan for bat activity in the expected frequency band (Figure 1a).

2.1.1 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), 16MB allows extending forward record durations to 20 s.
- 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 D for the bill of materials.

2.1.2 Circuitry

The system architecture and signal flow are presented in figure 2. 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.

2.2 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 16/20/24-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 [18].

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 [18].

2.2.1 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. Although the WM8782 supports 24-bit resolution, the current implementation uses 16-bit to maintain compatibility with the *Teensy Audio Library.*
- `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.

Table 2: Abbreviations, Technical Terminology, and Audio Library Elements

Term	Description
Hardware Components	
ADC	Analogue-to-Digital Converter. Converts analogue audio signals from microphones into digital data. The WM8782 is a 24-bit stereo ADC operating at up to 192 kHz.
OSR	Oversampling Ratio. The factor by which the ADC's internal sigma-delta modulator samples faster than the target output rate. The WM8782 uses 32× OSR at 192 kHz, meaning it samples internally at 6.144 MHz.
Sigma-Delta ADC	Type of ADC that uses oversampling and noise shaping to achieve high resolution. Converts analogue signals to a high-speed 1-bit stream, then decimates and filters to produce multi-bit output at the target sample rate.
DAC	Digital-to-Analogue Converter. Converts digital audio data back to analogue signals. The PCM5102A DAC is used for heterodyne audio output to headphones.
MEMS	Micro-Electro-Mechanical Systems. Miniature microphones (e.g., SPU0410LR5H-QB).
MCU	Microcontroller Unit. The Teensy 4.1 (ARM Cortex-M7, 600 MHz) serves as the central processing unit.
PSRAM	Pseudo-Static Random Access Memory. External memory used for the ring buffer, accessed via EXTMEM directive in C++.
I^2S	Inter-IC Sound. Serial bus interface for digital audio transmission. Two I^2S buses (I^2S1, I^2S2) connect the ADCs to the Teensy.
Slave Mode	ADC operation mode, where clock signals (BCLK, LRCLK) are provided by the master device (Teensy). The WM8782 operates in slave mode, with the M/S pin tied to logic low (0).
I^2C	Inter-Integrated Circuit. Serial communication protocol used for the OLED display.
Signal Processing	
DDS	Direct Digital Synthesis. Method for generating waveforms digitally using a phase accumulator, employed for sine wave carrier generation.
SNR	Signal-to-Noise Ratio. Measure of signal quality expressed in decibels (dB).
THD	Total Harmonic Distortion. Measure of nonlinearity in audio systems, expressed as a percentage or in decibels. Quantifies the ratio of unwanted harmonic frequencies to the fundamental frequency.
PCM	Pulse-Code Modulation; a digital representation of an analogue audio signal in which the signal amplitude is sampled at fixed time intervals and quantised into discrete numeric values. Used for lossless audio storage formats such as WAV files.
Heterodyne	Frequency mixing technique that multiplies an ultrasonic signal with a carrier sine wave to produce audible frequencies.
Phase Accumulator	Digital oscillator that generates a sine wave by incrementing a phase variable per sample.
TDoA	Time-Difference of Arrival. Method for spatial localisation using arrival time differences between microphones.
Teensy Audio Library Elements	
`AudioInputI2S`	Audio input object for the first I^2S interface (I^2S1). Receives stereo data from ADC 1 (channels L1, R1).
`AudioInputI2S2`	Audio input object for the second I^2S interface (I^2S2). Receives stereo data from ADC 2 (channels L2, R2).
`AudioRecordQueue`	Buffer object that stores incoming audio blocks (128 samples each) for processing. Four instances used: `queueL1`, `queueR1`, `queueL2`, `queueR2`.
`AudioPlayQueue`	Buffer object for audio playback. Used to send heterodyned audio samples to the DAC output.
`AudioOutputI2S`	Audio output object that transmits digital audio to the DAC via I^2S.
`AudioConnection`	Defines signal routing between audio objects. Example: `patchCord1(i2sInput1, 0, queueL1, 0)` connects I^2S1 left channel to queue.
`AudioMemory()`	Allocates memory blocks for audio processing. `AudioMemory(120)` reserves 120 blocks for buffering.
`readBuffer()`, `getBuffer()`, `playBuffer()`	Audio queue buffer management methods used to retrieve input sample blocks, allocate output buffers, and submit filled buffers for digital-to-analogue conversion during real-time audio processing.
`available()`	Method to check if a queue has data ready for reading or if an output buffer is available.
`clear()` / `freeBuffer()`	Methods to release processed audio blocks back to the memory pool after use.
Recording Modes	
Retrospective (TAP)	Recording mode that saves the previous 5 seconds from the ring buffer when triggered by the tap button.
Forward with Prefill (HOLD)	Recording mode that first saves 5 seconds of buffered audio, then continues recording up to 10 seconds of new audio when the hold button is pressed.
Clock Configuration	
PLL	Phase-Locked Loop. Hardware oscillator that generates precise clock signals for I^2S interfaces.
SAI	Synchronous Audio Interface. Teensy 4.1 peripheral that implements I^2S communication (SAI1 = I^2S1, SAI2 = I^2S2).
BCLK	Bit Clock. Clock signal that times individual bit transmissions in I^2S protocol.
LRCLK	Left-Right Clock (Word Clock). Indicates whether left or right channel data is being transmitted.

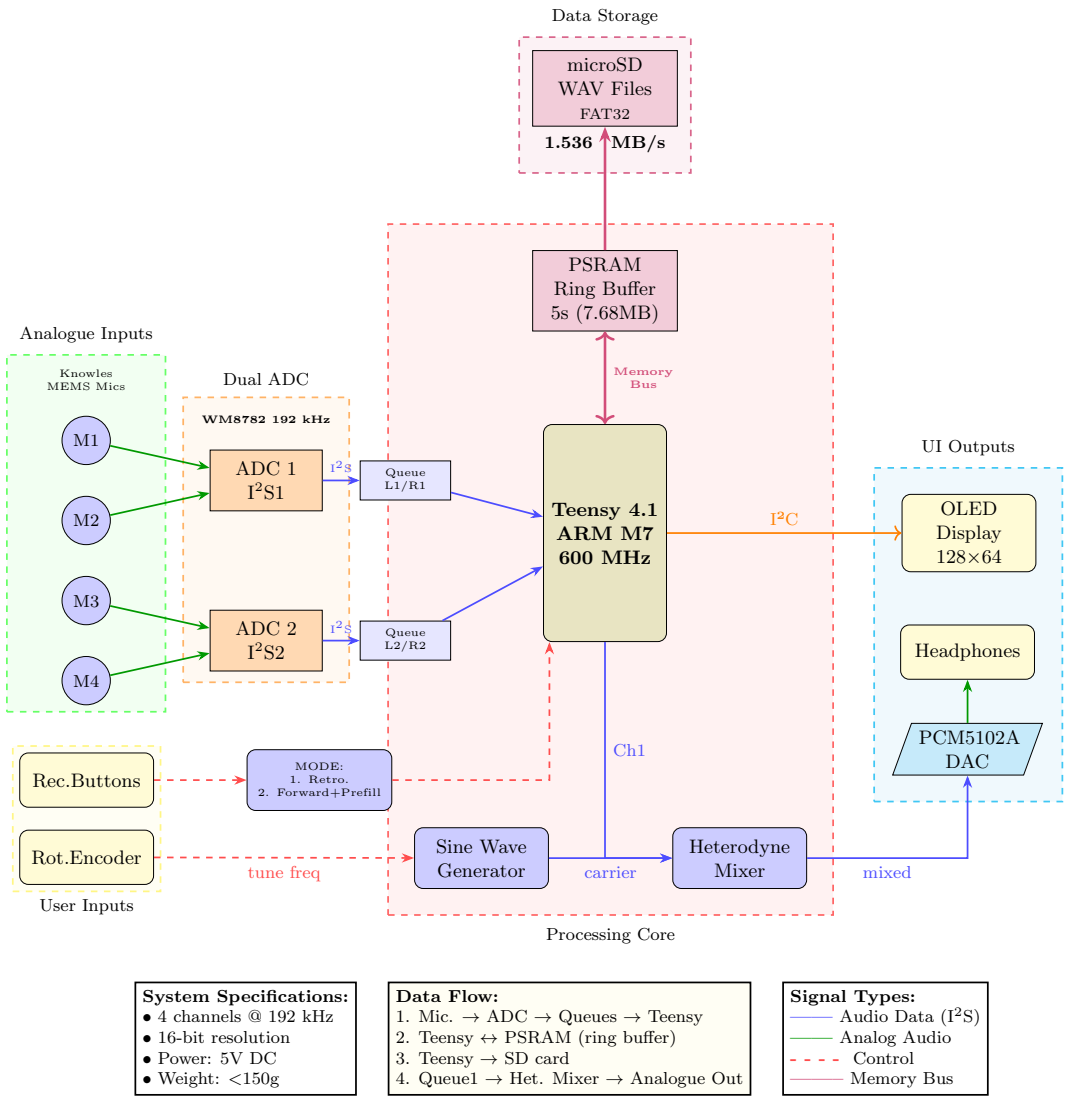

Figure 2: System architecture and data flow of BATSY4-PRO. Two WM8782 stereo ADCs digitise four Knowles SPU0410LR5H-QB MEMS microphones (M1–M4) at 192 kHz and stream via dual I^{2}S interfaces into a Teensy 4.1 microcontroller (ARM Cortex-M7, 600 MHz). Audio samples are buffered in channel-specific queues and written to an external PSRAM ring buffer, enabling both retrospective and forward-with-prefill recording modes. Upon user trigger, buffered data are interleaved and written as time-stamped, four-channel WAV files to a FAT32-formatted microSD card. In parallel, a selectable channel is routed through a heterodyne mixer via a sine-wave generator for real-time audible monitoring via a DAC and headphones, while system state and recording status are displayed on an OLED screen.

2.3 I^{2}S 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^{2}S interfaces.

2.3.1 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:

```
#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.

2.3.2 Master Clock Settings

The following clock configuration is a critical enabling step for high-frequency, multi-channel ultrasonic recording on the Teensy 4.1. To achieve a stable sampling rate of 192 kHz per channel, a custom clock configuration is implemented through the `setI2SFreqBoth()` function (Appendix A). 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 [19].

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 on Clock Control Module and SAI [19]).

2.4 Memory Buffer Architecture and Recording Modes

The firmware implements two complementary data capture modes that are directly referenced in the user interface and code as **TAP** and **HOLD** modes. These correspond to (i) a *retrospective buffer-triggered capture* (TAP), which saves a fixed-duration (default 5 s) window of recent audio from a continuously running ring buffer, and (ii) a *forward-triggered capture with buffer prefill* (HOLD), which appends ongoing audio recording to a preceding segment of buffered history. Together, these modes enable both rapid reaction to transient events and extended recording while preserving temporal context. The flowchart in figure 3 presents the implementation logic.

The choice of a 5 s retrospective buffer reflects typical bat flight dynamics and detection limits: a bat flying at 3–5 m s^{-1} traverses 15–25 m within this interval, beyond which echolocation calls are generally too weak for reliable recording or localisation. In contrast, behavioural events of

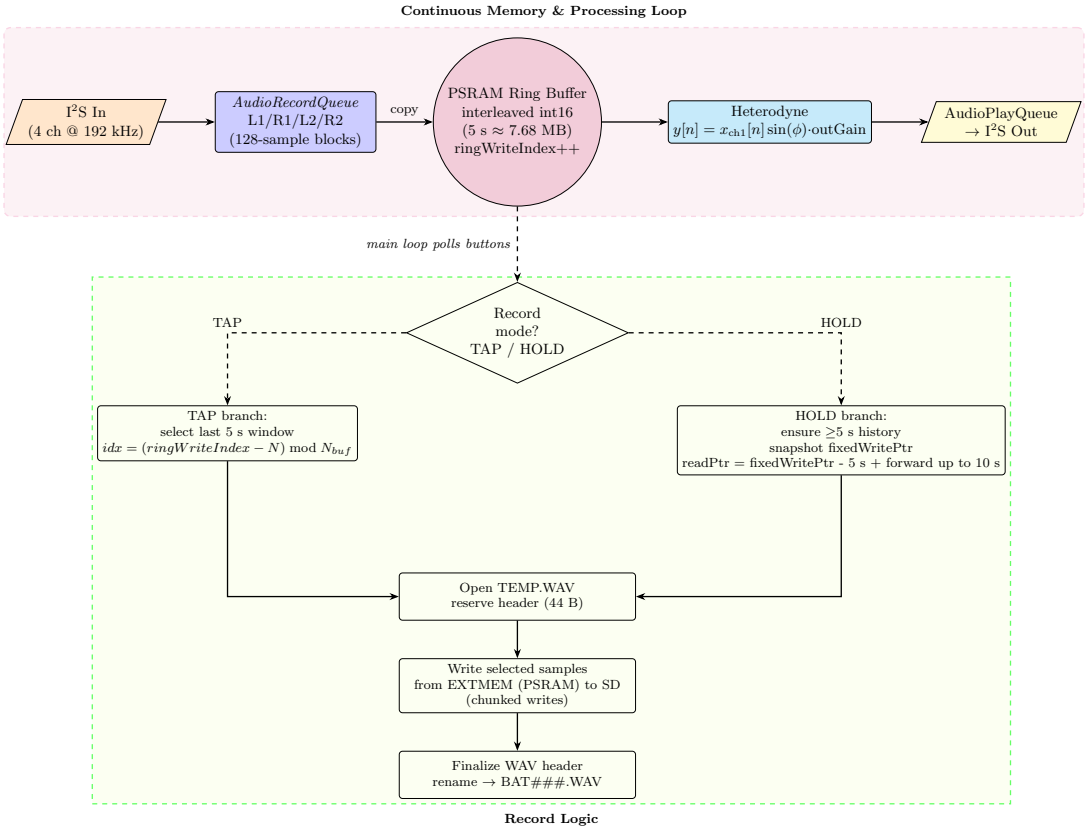

Figure 3: Ring-buffer recording logic Four synchronised 192 kHz channels are continuously acquired in 128-sample blocks via `AudioRecordQueue` objects and interleaved into a 5 s PSRAM (`EXTMEM`) ring buffer, while simultaneously producing a real-time heterodyne monitor stream. The main loop polls the user inputs to select the capture mode: **TAP** triggers *retrospective* capture by computing the start index for the last 5 s window idx, and copying that segment from PSRAM to a temporary WAV file using chunked writes; **HOLD** triggers *buffer-prefilled forward* capture by first ensuring $\geq$5 s history is available, snapshotting `fixedWritePtr`, writing 5 s of pre-buffer, then appending up to 10 s of forward audio before finalising. In both modes, audio is written to `/TEMP.WAV` with a reserved 44-byte header region, then the header is populated, and the file is renamed to a unique identifier.

interest—such as approaches to experimental prey items (e.g. mealworms)—often involve brief exploratory flight followed by a directed descent. Allowing an additional forward recording window of up to 10 s therefore provides a total capture duration of 15 s, which is sufficient to encompass the complete behavioural sequence from initial approach to prey interaction.

2.4.1 Ring Buffer

Incoming multi-channel audio is continuously written to a rolling ring buffer implemented in external PSRAM (`EXTMEM`) on the Teensy 4.1 microcontroller. Four synchronised audio channels are sampled at 192 kHz via two WM8782 ADCs interfaced over `I2S1` and `I2S2`. Samples are acquired in 128-sample blocks using `AudioRecordQueue` objects, interleaved across channels, and written sequentially into the ring buffer.

Each sample is stored as 16-bit PCM, yielding a sustained data rate of

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

A 5-second buffer (customisable in firmware code) is maintained, occupying 7.68 MB of PSRAM.

A circular write index advances continuously, overwriting the oldest samples once the buffer is full. A cumulative sample counter tracks total audio written, ensuring that a complete pre-trigger history is available before capture is initiated. (See implementation logic in Appendix B0

2.4.2 TAP Mode: Retrospective Buffer Capture

In **TAP mode**, a user tap triggers immediate saving of the most recent 5 seconds of audio from the ring buffer. Upon activation, the firmware computes a read pointer offset relative to the current write (to buffer) index and sequentially copies the buffered samples into a temporary WAV file (`/TEMP.WAV`). Data are written in small, non-blocking chunks (128–512 samples), interleaved with ongoing audio acquisition and processing, so that live recording and heterodyne monitoring continue uninterrupted.

Once the buffered window has been written, a standard 44-byte WAV header is inserted, and the file is finalised and renamed using an incrementing identifier (e.g., `/BAT001.WAV`). TAP mode, therefore, functions as a rolling "look-back" recorder, allowing capture of brief or unexpected acoustic events that occurred prior to user intervention. Since the buffer is not deleted after recording, subsequent TAP triggers within 5 s of the last recording include the newest data, with overlaps from the previous recording.

2.4.3 HOLD Mode: Forward Recording with Buffer Prefill

In **HOLD mode**, recording begins with a retrospective prefill followed by forward recording. When the HOLD input is activated, the firmware first verifies that the ring buffer contains at least 5 seconds of audio history. A snapshot of the current write index is then taken, establishing a fixed reference point for the pre-trigger segment.

The last 5 seconds of buffered audio are written to a new temporary WAV file, after which live audio is appended in real time for up to 10 seconds (user-configurable). During this forward phase, the firmware periodically compares the ring buffer write pointer and read pointer to determine how many new samples are available for flushing to disk, ensuring continuous recording without overruns or data loss. Recording terminates automatically upon timeout, after which any remaining buffered samples are written, and the WAV file is finalised.

2.4.4 Operational Characteristics

Both TAP and HOLD modes operate within a single cooperative processing loop without interrupts or blocking calls. The ring buffer continues to update at all times, independent of recording state, and user input determines how buffered data are extracted and committed to storage. This architecture provides reliable, low-latency, retrospective and prospective recording and continuous operation, making it well-suited for field deployments where acoustic events are brief and spontaneous.

2.5 Integrated Heterodyne Monitoring

The system incorporates a continuous heterodyne monitoring subsystem (continuous loop in Figure 3) 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.

2.5.1 Heterodyne Monitoring Implementation

Real-time heterodyne monitoring is implemented by digitally mixing the ultrasonic waveform from a user-selectable input channel (default: Channel 1) with a sinusoidal carrier of adjustable frequency. The carrier frequency defaults to 45 kHz and can be tuned dynamically between 10–85 kHz during operation using the rotary encoder, enabling rapid auditory exploration of different spectral regions.

The carrier is generated in real time using a phase-accumulator method [20], where the oscillator phase is incremented at each sample according to a precomputed phase increment derived from the selected carrier frequency and the fixed 192 kHz sampling rate (Appendix B). The input signal is heterodyned via sample-wise multiplication with the sine carrier, producing both sum and difference frequency components and effectively folding ultrasonic energy around the carrier frequency [21, 22]. For example, a 62 kHz component mixed with a 60 kHz carrier yields a 2 kHz audible tone. A related dual-band heterodyne implementation is described in Umadi (2025) [23].

No explicit digital low-pass filtering is applied after mixing. As in conventional heterodyne bat detectors, the high-frequency image band is naturally attenuated by the reconstruction filter of the PCM5102A digital-to-analogue converter [24] and by the downstream analogue playback chain. Because the heterodyne signal is used exclusively for real-time auditory monitoring and not for quantitative analysis, this minimal filtering approach is sufficient and avoids additional computational overhead.

The mixed signal is constrained to 16-bit signed integers and streamed via an `AudioPlayQueue` to the `AudioOutputI2S` interface for playback. Output amplitude is scaled by a user-adjustable gain parameter, effectively providing a digital volume control. All four input channels are simultaneously recorded in parallel as raw, full-bandwidth ultrasonic data to the ring buffer and SD card, ensuring that heterodyne monitoring does not alter or interfere with the stored recordings. Full implementation details are provided in Appendix C.

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.

2.6 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 a 1 cm^2 square-section aluminium rod, inserted into a central six-way connector hub (commercially available, see bill of materials in Appendix D). The microphones are wired internally along each arm, and all connections terminate at a single 6-pin connector at the hub, enabling fast, 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.

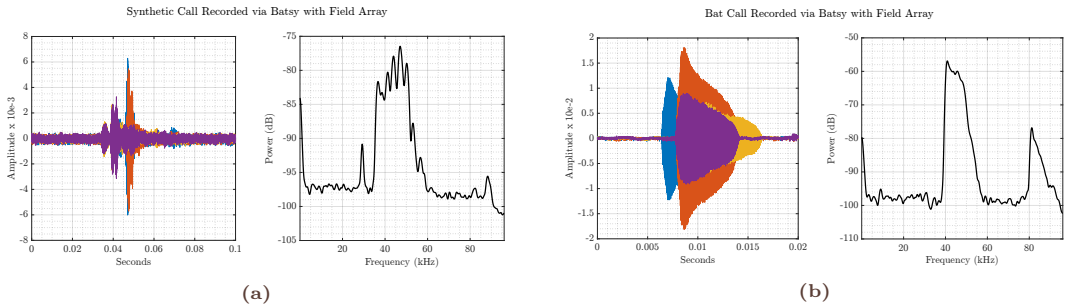

Figure 4: Left: A computer-generated quadratic FM chirp, recorded indoors without acoustic insulation on either the array or the room, resulting in spectral interference from multipath reflections. The sound source was positioned approximately 2 m from the microphones, with an unknown source level estimated to be comparable to normal speech (≈70 dB SPL at 10 cm). **Right:** 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.

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 acoustic 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.

2.7 Validation

Laboratory recording, testing, and validation were conducted using synthetic FM chirps played back through an RME Babyface (Audio AG, Haimhausen, Germany) audio interface, amplified via a single-channel power amplifier, and reproduced with a Scan-speak R2004 tweeter (Scan-Speak A/S, Denmark). The output level was indeterminate, but the playback level was approximately set to match normal speech (≈ 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 4a).

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. No special permit was necessary to record free-flying bats. The array was mounted on a tripod and deployed in a semi-open corridor clearing (Figure 1c).

Approximately 30 distinct foraging sequences characterised by rapid echolocation call bouts and frequency-modulated signals were recorded. Recordings were manually screened and sorted for high-quality multi-channel segments with high signal-to-noise ratios across all four microphones (Figure 4b) using the *Bat Reviewer* tool [25]. 20 ms regions around selected calls were extracted via custom-written MATLAB scripts, adopted from the `Biosonar Responsivity Analysis Toolkit V1.0` [26].

2.8 SNR characterisation of field data

To quantify relative signal-to-noise performance under field conditions, extracted echolocation calls were analysed using a consistent, peak-based signal-to-noise metric (Figure 5). Each call clip was approximately 20 ms long. Noise was estimated from the quietest 2 ms window within

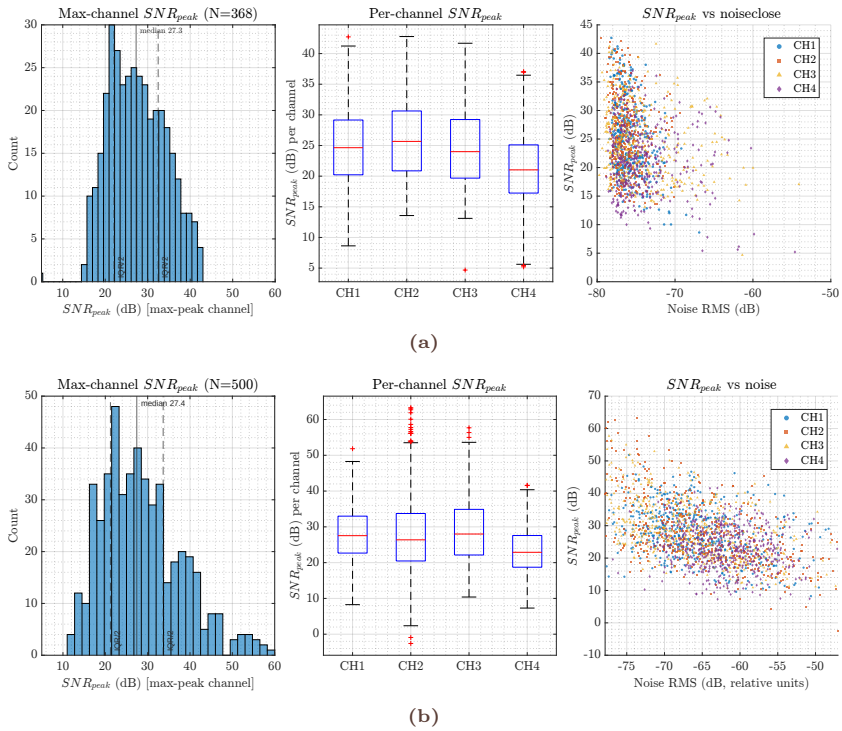

Figure 5: SNR Comparison against a professional audio interface. Peak-based signal-to-noise ratios ($\mathrm{SNR_{peak}}$) are shown for field recordings obtained with Batsy4-Pro (top) and for a randomly selected subset of recordings obtained using a professional audio interface (Behringer U-PHORIA UMC404HD; bottom). For each system, panels show (A) the distribution of maximum-channel $\mathrm{SNR_{peak}}$ across calls, with dashed lines indicating the median and interquartile range, (B) per-channel $\mathrm{SNR_{peak}}$ distributions, and (C) the relationship between $\mathrm{SNR_{peak}}$ and noise root-mean-square (RMS) level for all calls and channels. Median maximum-channel $\mathrm{SNR_{peak}}$ values were nearly identical between systems (27.3 dB for Batsy4-Pro; 27.4 dB for the Behringer soundcard), and both datasets show similarly tight noise distributions.

the final portion of each clip, identified as the minimum root-mean-square (RMS) amplitude using a sliding-window search to minimise contamination by call energy.

For each channel, call amplitude was defined as the maximum absolute sample value outside the noise-search region. A peak-based signal-to-noise ratio ($\mathrm{SNR_{peak}}$) was computed as

$$\mathrm{SNR_{peak}} = 20 \log_{10} \left(\frac{\max |x_{\mathrm{call}}|}{\mathrm{RMS}(x_{\mathrm{noise}})} \right).$$

For each call, the channel exhibiting the highest peak amplitude was used to summarise maximum-channel $\mathrm{SNR_{peak}}$, while per-channel SNR distributions were analysed to assess channel-to-channel consistency.

2.9 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.

2.9.1 Call spectral clustering

To demonstrate the suitability of Batsy4-Pro field recordings for downstream acoustic analyses, an unsupervised spectral clustering approach was applied to extracted echolocation calls. Such analyses are particularly relevant for field datasets, where recordings commonly contain calls from multiple species and a diversity of call types within species. Clustering based on spectral

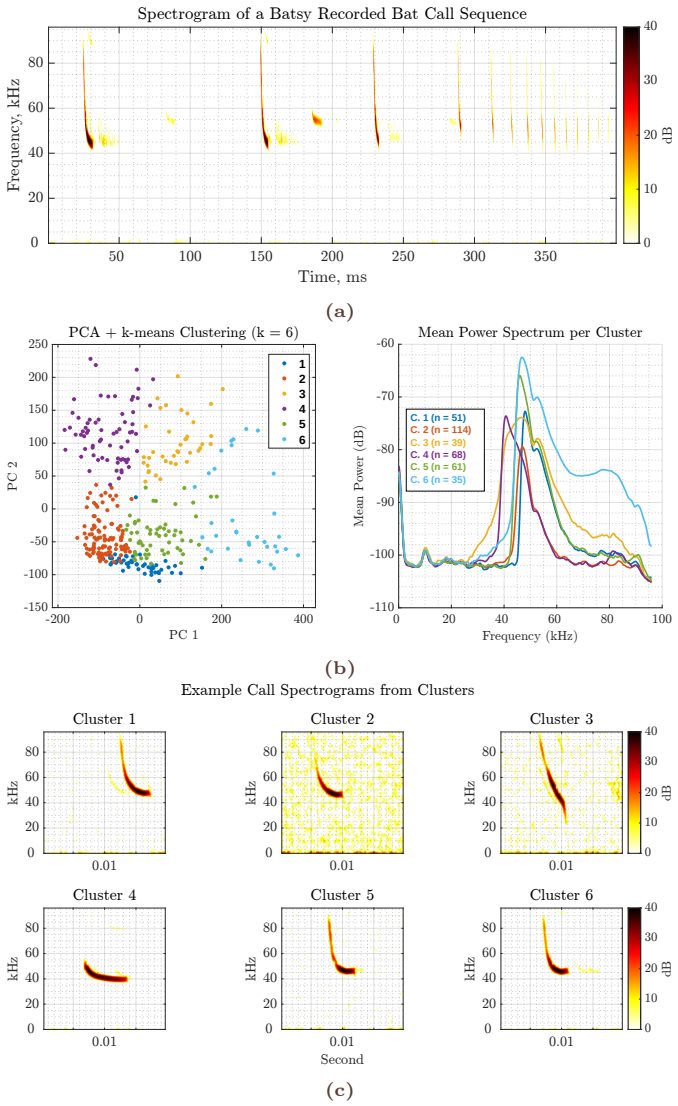

Figure 6: System Validation: (**a**) Spectrogram of a sequence of bat calls recorded on the Batsy4-Pro system at a foraging site along the Isar river, Freising, Germany (different day and site than the rest of the data). (**b**) The Clustering analysis of individual calls' power spectra using PCA and 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.

structure provides an objective means of segregating calls before species-focused or behavioural analyses.

For each call, the waveform from the first microphone channel was tapered using a Hann window, and its power spectral density was estimated using MATLAB's `pspectrum` function with a frequency resolution of 1 kHz. Power spectra were converted to decibel units and truncated to a fixed length of 1024 frequency bins to ensure consistent dimensionality across calls. The resulting spectra were assembled into a matrix representing spectral features for all calls.

Dimensionality was reduced using principal component analysis (PCA), retaining the first ten principal components for clustering. The optimal number of spectral clusters (tested for $k = 2$–6) was determined automatically using the Davies–Bouldin criterion [27] via the `evalclusters` function, providing an objective balance between within-cluster compactness and between-cluster

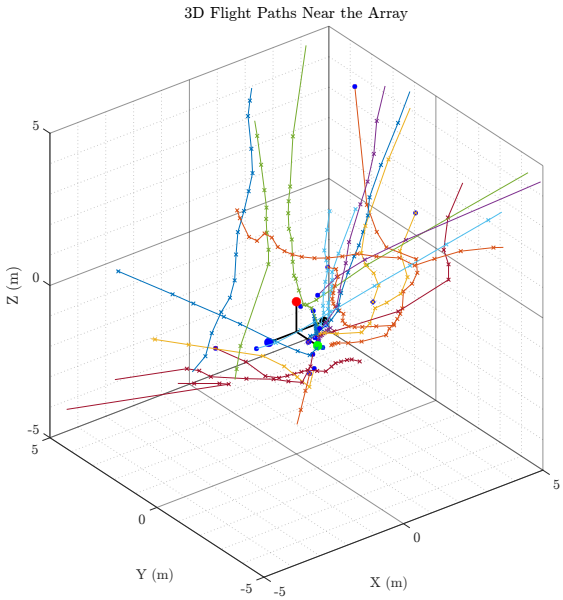

Figure 7: The array was deployed at a bat foraging site, and calls were recorded manually by monitoring the real-time heterodyne output of the Batsy4-Pro system. Using time-difference-of-arrival (TDoA) localisation, the 3D positions of calls were estimated. The flight paths are visualised within $\pm 5\,\mathrm{m}$ 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.

separation. Final clustering was performed using k-means on the reduced feature space.

Cluster validity was assessed qualitatively by visualising the mean power spectra for each cluster and by inspecting representative spectrograms randomly selected from each group. This approach illustrates how field-collected data from a compact recorder can be structured into acoustically coherent call groups, facilitating subsequent species identification, call-type classification, or behavioural state analyses without requiring prior labels. The results of the clustering analyses are presented in figure 6.

2.9.2 3D localisation via time-difference of arrival

To estimate the spatial position of the calling bat for each detected call, a time-difference-of-arrival (TDoA) approach was applied to the four-channel recordings. Inter-microphone arrival delays were extracted using a custom cross-correlation-based function (`estimateTDOA`) that computed pairwise time lags for all microphone pairs.

The resulting delay estimates were passed to a multilateration routine (`localiseTDOA`) that solved for the three-dimensional source position (Figure 7) based on the known microphone geometry and an assumed speed of sound of $c = 343\,\mathrm{m\,s^{-1}}$. These routines form part of the `Array WAH - Microphone Array Design Tool` [28, 29].

2.10 Localisation Accuracy Characterisation

To systematically characterise the spatial localisation accuracy of the Batsy4-Pro system, I implemented a Monte Carlo simulation framework that quantified positional uncertainty across the array's operational volume, derived from the `Array WAH - Microphone Array Design Tool` [28, 29]. The approach followed a grid-based sampling strategy designed to evaluate localisation performance under controlled, yet biologically realistic, conditions, explicitly accounting for

source motion during call emission (Figure 8).

All simulations employed synthetic frequency-modulated (FM) calls with a fixed duration of 5 ms (see Section 3.1), sweeping linearly from 80 kHz to 25 kHz. A systematic three-dimensional grid was constructed spanning ±10 m around the array centre, with a spatial resolution of 0.5 m between grid points. Two principal analysis planes were extracted from this volume: a horizontal XY plane at $z = 0.5$ m. Together, these planes comprised 3362 spatial positions and provided comprehensive coverage of the near to mid-field region relevant for field deployments.

To account for source motion during call emission, simulations included moving sources with speeds ranging from 0 to 10 $\mathrm{m\,s^{-1}}$ in increments of 1 $\mathrm{m\,s^{-1}}$. For the stationary condition ($v = 0\ \mathrm{m\,s^{-1}}$), a single trial was simulated per grid point. For all non-zero speeds, eight Monte Carlo trials were performed per grid point and speed, with flight direction randomised in three dimensions. This resulted in 272,322 simulation trials across all spatial and kinematic conditions.

During each trial, the source was displaced linearly over the 5 ms call duration according to the assigned velocity vector. Four-channel microphone signals were generated accordingly, and TDoA estimates were obtained using cross-correlation-based delay estimation. The source position was then recovered via multilateration. Localisation accuracy was quantified as the Euclidean distance between the estimated position and the true grid-point centre, expressed in centimetres [29].

Localisation performance was summarised at two levels. Per-grid-point statistics, including the median and 95th percentile deviation, were computed across all speeds and directional trials. At the per-trial level, deviation distributions were analysed as a function of speed and range. To assess the relative contributions of geometric configuration and source motion, variance explained (R^2) was calculated using linear regression models with range and speed as predictors of localisation error. Because error distributions were non-normal, Spearman rank correlations and Kruskal-Wallis tests were additionally employed. All quantitative analyses focused on the near-field region within 4 m of the array, based on the estimated localisation accuracy.

3 RESULTS

I deployed the Batsy4-Pro system and accompanying four-channel microphone array during the first week of August 2025 along the river Isar in Freising, Germany (Figure 1). I selected the recording site at the edge of a riparian forest along the riverbank for its consistent bat foraging activity and suitability for field validation of multichannel ultrasonic recording. I mounted the array on a standard photographic tripod and oriented it to face the primary approach corridor used by bats when moving between foraging areas.

Deployment involved three straightforward steps: I assembled the array arms onto the tripod-mounted hub, connected the microphone cables to the Batsy4-Pro recorder via the single 6-pin connector, and activated the system by connecting a USB power bank. The entire setup process typically requires less than five minutes. The system becomes immediately operational, with the integrated heterodyne monitoring providing real-time auditory feedback of bat activity through headphones.

3.1 Data Quality Assessment and Call Selection

I screened and validated recorded call sequences using the *Bat Reviewer* software tool [25]. The screening process evaluated each call sequence for signal quality, channel consistency, and freedom from contaminating echoes or environmental interference. Of the 30 recorded sequences, I selected 21 for detailed analysis based on the following quality criteria: clear separation between direct calls and ground echoes, consistent detection across all four microphone channels. I retained 368 of the 440 calls (83.6%) for quantitative analysis. This high validation rate indicates robust multichannel recording performance and minimal data loss from technical artefacts.

Call durations exhibited the short, frequency-modulated characteristics typical of aerial-hawking insectivorous bats foraging in semi-open environments. The median call duration was 4.51 ms (interquartile range: 3.64–5.53 ms, μ =4.91 ms), with the central 90% of calls spanning 3.14–9.16 ms. The full range extended from 2.78 ms to 13.26 ms, reflecting variation in call structure associated with different phases of foraging behaviour and target approach distances. Detailed statistics on call duration distributions are provided in Appendix E.

I stored all validated calls as individual 20 ms audio clips extracted from the continuous four-channel recordings, preserving synchronised waveforms across all microphone channels. These clips formed the basis for subsequent signal-to-noise characterisation, spectral clustering, and three-dimensional source localisation analyses presented in the following sections.

3.2 SNR Validations

Across $N = 368$ echolocation calls recorded with Batsy4-Pro, the maximum-channel $\mathrm{SNR}_{\mathrm{peak}}$ had a median value of 27.3 dB (interquartile range: 10.4 dB; mean ± SD: 27.6 ± 6.6 dB; Fig. 5a). Per-channel $\mathrm{SNR}_{\mathrm{peak}}$ distributions were broadly similar, with median values ranging from approximately 21 to 26 dB, indicating comparable noise performance across channels.

For comparison, the same analysis was applied to a randomly selected subset of $N = 500$ echolocation calls recorded during a separate experiment dataset using a professional-grade audio interface (Behringer U-PHORIA UMC404HD). The call extraction protocol was the same, with each call corresponding to a 20 ms duration. These data were collected in a different experiment [30] using a similarly sized microphone array constructed with the same MEMS microphones, and recorded via a custom MATLAB acquisition script on a MacBook Air (2015). The resulting maximum-channel $\mathrm{SNR}_{\mathrm{peak}}$ distribution closely matched that obtained with Batsy4-Pro, with a median of 27.4 dB (interquartile range: 12.5 dB; mean ± SD: 29.0 ± 9.9 dB; Fig. 5b). Together, these results indicate that variability in SNR is dominated by differences in call amplitude rather than fluctuations in recorder noise, and that the field performance of Batsy4-Pro approaches that of a professional soundcard-based recording system.

3.3 Characterising Localisation Accuracy

Within the near-field region (<4 m from the array centre), 386 grid points were evaluated. The spatial distribution of median localisation uncertainty exhibited pronounced geometric structure across both horizontal and vertical planes (Figure 8). The lowest uncertainties were concentrated in a compact region surrounding the array centre, with median deviations typically below 50 cm and confined to approximately 2 m. Beyond this zone, localisation error increased progressively, reaching median values of 300–350 cm near the boundary of the 4 m analysis volume.

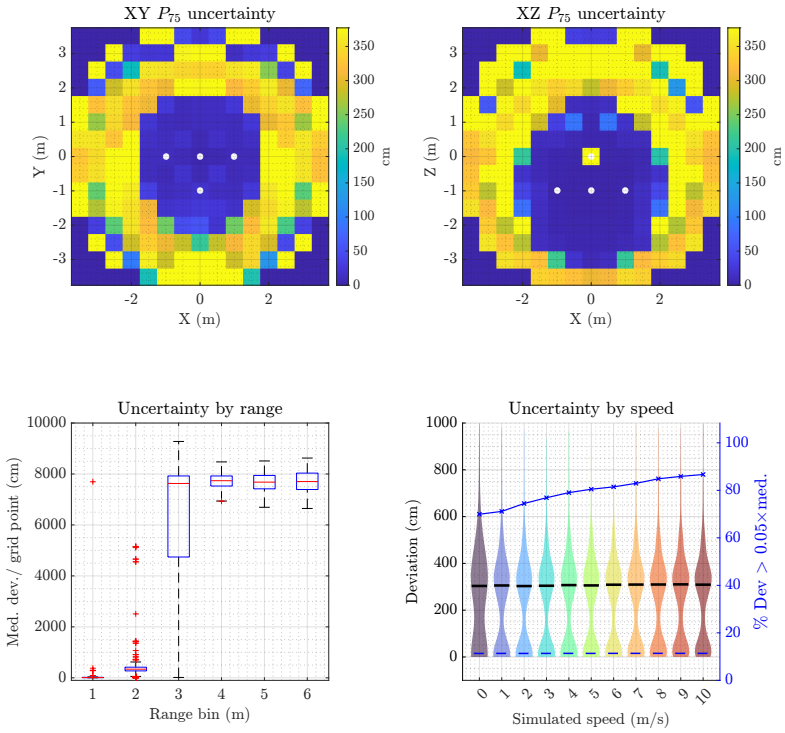

Figure 8: Grid-based localisation uncertainty analysis. **Top panels:** Spatial distribution of median localisation deviation across the horizontal XY plane (left) and vertical XZ plane (right), with values clipped at the 75th percentile for visualisation. White circles indicate microphone positions. Dark blue regions correspond to high-precision zones near the array centre. **Bottom left:** Median localisation deviation as a function of range from the array, showing a systematic increase with distance (boxes represent per-grid-point medians aggregated across all simulated source speeds; range bins span 0–12 m in 2 m increments). **Bottom right:** Effect of simulated source speed (0–10 m s^{-1}) on localisation uncertainty within the usable volume (<4 m). Violin plots show the full distribution of trial-wise deviations, with black bars indicating medians. The overlaid line (right axis) shows the fraction of trials exceeding a stringent accuracy criterion defined as 5% of the median deviation (approximately 17.5 cm), illustrating that while median localisation accuracy remains largely invariant with speed, the shape of the error distribution—and the proportion of larger deviations—changes systematically with increasing source velocity. Grid resolution: 0.5 m; call duration: 5 ms FM (80–25 kHz).

This spatial pattern was consistent across planes and was not well described by distance alone. Instead, uncertainty varied substantially among grid points at comparable ranges, reflecting anisotropic conditioning imposed by the array geometry. Positions that subtended favourable angular separations relative to multiple microphone baselines exhibited high localisation precision, whereas positions aligned with extended baselines or producing weak geometric dilution showed degraded performance, consistent with the results from Umadi [29].

The distribution of per-point median deviations across range bins illustrates the relationship between localisation error and range. Within 0–2 m of the array, most grid points exhibited median errors below 100 cm, with an interquartile range spanning approximately 20–150 cm. Beyond 3 m, localisation performance deteriorated sharply, with median deviations increasing to several metres, rendering these regions unsuitable for precise spatial tracking.

Within the operationally relevant near-field region (<4 m), localisation accuracy remained high for a substantial fraction of grid points. Overall, 32.1% of grid points (124 of 386) achieved median localisation deviations of 50 cm or less, while 35.2% (136 of 386) achieved deviations of 100 cm or less.

Performance was comparable across near-field range bins. In the 0–2 m bin, 38.8% of grid points achieved median deviations ≤50 cm and 40.6% achieved ≤100 cm. In the 2–4 m bin, the corresponding proportions were 27.1% and 31.2%, respectively. This gradual decline indi-

cates predictable scaling of uncertainty with the array's spatial configuration rather than abrupt performance boundaries in the near field.

Negligible influence of source velocity on central localisation accuracy Source velocity during call emission had no practically meaningful effect on the *central tendency* of localisation accuracy. Across the full tested velocity range (0–10 $\mathrm{m\,s^{-1}}$), median localisation error remained effectively constant, with differences of less than 5 cm between speed conditions. Interquartile ranges were similarly stable, indicating that typical localisation performance is dominated by array geometry rather than source motion.

Formal statistical analyses support this conclusion. Linear regression showed that source speed explained none of the variance in localisation error ($R^2 \approx 0$), and the Pearson correlation between speed and deviation was effectively zero ($r = -0.002$, $p = 0.682$). A weak but statistically detectable rank correlation was observed (Spearman $\rho = 0.053$, $p \to 0$); however, this accounted for less than 0.3% of rank variance and reflects the large sample size rather than a biologically meaningful effect.

While median accuracy is invariant, higher-order distributional properties reveal a subtle velocity dependence. Specifically, the fraction of trials exceeding a stringent error threshold—defined as deviations greater than 5% of the median localisation error (approximately 17.5 cm)—increased monotonically with source speed. This indicates that although typical localisation accuracy is unaffected by motion, higher velocities slightly increase the probability of rare, large-error events. These effects are visible in the changing shape of the error distributions with speed (Fig. 8, bottom right), despite unchanged medians.

The observed pattern is consistent with the expected displacement of the source during emission (0–50 mm at 10 $\mathrm{m\,s^{-1}}$ for 5 ms calls), which is small relative to inter-microphone spacing and therefore insufficient to bias central localisation estimates, but capable of modulating the tails of the error distribution. Taken together, these results indicate that for short FM calls, source velocity does not degrade typical localisation accuracy, but can influence the likelihood of extreme errors.

Comprehensive analyses of directional motion effects, array geometry optimisation, and angular localisation accuracy—particularly under controlled ground-truth conditions—are beyond the scope of the present study and will be addressed in a dedicated follow-up investigation.

4 DISCUSSION

The development of Batsy4-Pro addresses a persistent constraint in behavioural bioacoustics: the trade-off between recording quality, portability, and cost. Traditional multichannel setups for studying echolocating bats typically require a combination of professional audio interfaces, external preamplifiers, laptop computers, and reliable power infrastructure, and have been used in lab and field-based studies [30–38] (also see [39]). These systems, while capable of excellent signal fidelity, present significant operational challenges in field settings. Proper channel configuration, gain staging, weatherproofing sensitive components against condensation, and power management require both technical expertise and logistical support, which necessitate team-based deployments. The cumulative cost of such configurations typically ranges from several hundred to several thousand euros, restricting access to well-funded laboratories in developed regions.

Batsy4-Pro offers a fundamentally convenient approach by integrating all essential components – analogue-to-digital conversion, computational processing, data storage, and real-time monitoring – into a single compact device weighing less than 150 grams. The system requires no external laptop, no multichannel soundcard, and no AC power source, operating instead from a standard USB power bank. This architectural consolidation eliminates the multiplicity of failure points inherent in distributed recording chains, dramatically simplifying field deployment. A researcher can carry the complete system in a backpack, assemble the array in minutes, and begin recording without the need for calibration procedures or driver configuration that plague computer-based acquisition systems.

The baseline calibration of the system represents another significant advantage. Because all four channels share identical signal paths—from MEMS microphone through ADC to digital storage—inter-channel gain differences are minimised by design rather than requiring post-hoc correction. Unlike audio interfaces where each channel may have independent gain controls that drift or reset between sessions, Batsy4-Pro maintains consistent relative sensitivity across channels. Level calibration to an absolute acoustic reference pressure (20 μPa) needs to be performed only once, and the resulting compensation factors can be incorporated directly into the firmware or applied during post-processing. This inherent calibration stability is particularly valuable for studies requiring long-term projects or comparative measurements across multiple recording sessions.

For applications demanding higher precision, frequency-response calibration filters can be computed using established methods [40–44] and integrated into the processing pipeline. Given the computational capability of the Teensy 4.1 platform, such filtering can be applied in real time before data are written to storage, analogous to the heterodyne down-conversion already implemented in the current firmware. This flexibility to incorporate domain-specific signal conditioning at the firmware level—rather than relying on post-processing workflows—represents a significant advantage for researchers working with non-standard microphone arrays or specialised acoustic environments.

4.1 Cost Accessibility and Open Development Model

The total component cost for Batsy4-Pro, including the recorder unit and four-channel array, is approximately 200 euros when sourced in small quantities from local suppliers. This represents less than one-tenth the cost of comparable commercial multichannel ultrasonic recording systems, or using a professional audio interface with a computer (Table 1).

Importantly, all hardware designs, firmware source code, and assembly documentation are released as open source under the CERN Open Hardware Licence v2 – Strongly Reciprocal (CERN-OHL-S), GPLv3, and CC-BY-SA-4.0 licences, respectively [15]. These licences explicitly permit commercial manufacture, distribution, and sale without royalty obligations, provided that derivative works remain openly licensed and complete source documentation is made available to recipients. This deliberate choice encourages local manufacturing initiatives, institutional workshop projects, and community-driven development, thereby further reducing costs and increasing accessibility.

The economic barrier to entry for behavioural bat research has historically concentrated experimental work in a small number of laboratories in wealthy nations, despite the fact that the greatest diversity of chiropteran species is found in tropical and subtropical regions of the

developing world [11]. India alone hosts over 120 species of echolocating bats, yet significant field experiments on bat behaviour remain notably absent from the published literature originating from that region. Similar patterns exist throughout Central and South America, Southeast Asia, and sub-Saharan Africa, where ecological surveys and taxonomic work dominate over behavioural and neuroethological investigations. The prohibitive cost of multichannel ultrasonic recording equipment represents a non-trivial component of this geographic disparity.

Research on echolocation in bats and other animals is inherently interdisciplinary, drawing on concepts and tools from electronics, signal processing, and communication engineering, as well as biological theory. Since the earliest demonstrations of ultrasonic biosonar in bats, progress in this field has depended on close collaboration between biologists and engineers. Notably, Donald Griffin and Robert Galambos's foundational work on bat echolocation was enabled by access to specialised ultrasonic detection equipment developed by the physicist G. W. Pierce at Harvard [45–48]. This pattern has persisted: advances in understanding biosonar behaviour, sensory-motor coordination, and ecological function have consistently required custom instrumentation and bespoke analytical methods.

While scientifically productive, this reliance on engineering expertise can pose substantial barriers for biologists and ecologists, particularly when specialised hardware is expensive, difficult to deploy, or available only in a small number of laboratories. Recent developments in lightweight acoustic tags and on-animal recorders have demonstrated how accessible instrumentation can transform the study of natural behaviour, enabling unprecedented insights into free-flight echolocation, foraging strategies, and sensorimotor control [49–53]. Similarly, neuroethological studies employing custom-developed wireless neural telemetry systems have revolutionised our understanding of spatial navigation by enabling single-neuron recordings from bats flying in naturalistic environments—from large indoor flight rooms to 200-meter tunnels and even on remote oceanic islands [54–56]. However, such technologies often remain closely coupled to the groups that develop them, limiting broader adoption and replication.

By providing a complete, tested, and documented design that can be assembled from globally available components, Batsy4-Pro lowers the threshold for entry into experimental bioacoustics. The choice to base the system on the widely adopted Teensy platform, supported by extensive community documentation and a large user base among hobbyists and engineers, ensures that technical support and troubleshooting resources are readily available outside traditional academic networks. This accessibility extends beyond professional researchers to educators, citizen scientists, and conservation practitioners who may wish to incorporate behavioural monitoring into field programmes but lack access to expensive commercial systems.

4.2 Integrated Real-Time Monitoring and Adaptive Recording

The inclusion of continuous heterodyne monitoring as an integrated system feature addresses a practical limitation that has long complicated ultrasonic fieldwork: the inability to perceive bat activity directly. Researchers deploying traditional recording systems must either rely on visual observation of bat movement, which is unreliable at night and provides no information about call production, or employ separate bat detectors that add weight, power consumption, and complexity to field setups. The heterodyne output in Batsy4-Pro, generated in real time through firmware-based frequency down-conversion and delivered via a high-fidelity DAC to standard headphones, allows researchers to hear echolocation calls as they occur.

This auditory feedback is particularly valuable for manual triggering of recordings in behavioural studies where experimental context matters. Rather than recording continuously and sorting through hours of data post hoc, researchers can listen for specific acoustic signatures—approach sequences, social calls, feeding buzzes—and selectively capture relevant events. The retrospective buffer architecture complements this capability by preserving the five seconds preceding any trigger event, ensuring that the behavioural context leading up to a decision to record is not lost.

The ability to dynamically tune the heterodyne carrier frequency during deployment (10–85 kHz) enables researchers to scan across frequency bands and optimise monitoring for species with different call characteristics. This is particularly relevant in regions with high bat diversity, where multiple species may be active simultaneously but vocalising in different frequency ranges. By adjusting the carrier frequency in real time, a researcher can selectively attend to the acoustic activity of interest without requiring prior knowledge of the exact spectral composition of calls.

4.3 Signal Quality and Localisation Performance

The signal-to-noise characterisation presented in this study demonstrates that Batsy4-Pro achieves recording fidelity comparable to professional-grade audio interfaces under field conditions. Median maximum-channel SNR values of 27.3 dB across 368 echolocation calls, with consistent performance across all four channels, indicate that the system noise floor is sufficiently low to support high-quality acoustic analysis. The close correspondence between SNR distributions obtained with Batsy4-Pro and those from a commercial soundcard-based system (median 27.4 dB, $N = 500$ calls) confirms that observed SNR variability is dominated by natural variation in call amplitude and source distance rather than by recorder noise characteristics. This performance validates the use of compact MEMS microphones and integrated ADC architectures for demanding bioacoustic applications that previously required more complex recording chains.

The spatial localisation accuracy simulations provide a principled basis for interpreting positional estimates derived from the four-channel cross-type array configuration employed here. Within the near-field region extending to approximately 4 m from the array centre, more than one-third of grid points achieved median localisation uncertainties below 50 cm, with accuracy degrading predictably as a function of range and geometric configuration. These results delineate a well-defined operational volume suitable for behavioural studies focused on close-range interactions, such as prey capture, obstacle avoidance, or social encounters.

Source velocity during call emission had no practically meaningful effect on the *central tendency* of localisation accuracy. Across the tested velocity range of 0–10 $\mathrm{m\,s^{-1}}$, median localisation error and interquartile ranges remained effectively constant. This result can be explained by the short duration of typical echolocation calls: a 5 ms FM sweep emitted by a bat flying at 10 $\mathrm{m\,s^{-1}}$ results in a displacement of only 50 mm during emission, which is small relative to the inter-microphone spacing (100–200 cm) of the array. Under these conditions, motion-induced time-of-arrival distortions and Doppler-related effects do not bias typical localisation estimates.

However, inspection of the full error distributions reveals a more nuanced velocity dependence. While medians remain invariant, the *shape* of the localisation error distributions changes systematically with increasing speed, reflected in a growing fraction of trials exceeding a stringent accuracy threshold (defined here as deviations greater than 5% of the median error, approximately 17.5 cm). This increase in extreme-error probability indicates that source motion does

not degrade typical localisation performance, but does elevate the likelihood of rare, large-error events. Such effects are not captured by summary statistics alone and highlight the importance of considering distributional properties—rather than central tendency alone—when evaluating localisation performance for moving sound sources.

These findings underscore an essential methodological implication: although velocity correction is unnecessary for most short-call localisation tasks, accounting for source motion becomes relevant when robust tracking or outlier suppression is required, particularly at higher speeds. Velocity-aware localisation models or post hoc filtering strategies may therefore improve reliability in studies targeting fast-moving animals or rare behavioural events.

Finally, it is essential to acknowledge that the present array geometry, while optimised for portability and rapid deployment, is not designed for far-field accuracy or extended tracking ranges. As demonstrated by the Array WAH simulations, localisation uncertainty increases rapidly beyond 4 m, reaching several metres at the boundary of the analysed volume. This reflects the geometric dilution of precision inherent in small-baseline arrays and represents a fundamental trade-off between portability and spatial coverage. Researchers requiring extended tracking ranges or far-field precision may benefit from larger or alternative array geometries, which can be explored systematically using the Array WAH design framework developed alongside this work [29].

4.4 Validation Against Established Methods and Ground Truth Considerations

The localisation performance presented here must be interpreted in the context of broader methodological challenges in bat bioacoustics. Many published studies employing microphone arrays with 8–12 channels report localisation results primarily for the final approach to terminal phase (<2 s) of echolocation sequences, where signal-to-noise ratios are highest and geometric configurations most favourable [36, 37]. Screening procedures that exclude spatial jumps, far-field data points, and low-SNR calls are routinely applied, yet the underlying ground-truth accuracy of these systems is rarely quantified or reported in detail. Instead, reliability is inferred indirectly from signal quality metrics and trajectory smoothness, with the implicit assumption that high-SNR calls localise accurately. (see Umadi [29] for a related discussion)

This lack of standardised validation against known source positions represents a significant gap in the field. While simulation-based approaches, such as those implemented in Array WAH [29], provide valuable insights into expected performance under idealised conditions, empirical validation using calibrated acoustic sources at pre-calculated positions remains essential for establishing absolute accuracy. The discrepancy between simulated and empirical accuracy is likely to be non-negligible, particularly in field environments where multipath propagation, atmospheric absorption, and environmental noise degrade signal quality in ways that are difficult to model comprehensively. However, the results are consistent with empirical validation studies such as de Framond et al. [33] – a rare example of explicit ground-truth validation – who assessed localisation accuracy using calibrated loudspeaker playbacks at 45 precisely surveyed positions around a fixed array and reported strong spatial dependence of error.

The Batsy4-Pro system, combined with the Array WAH simulation framework, represents an intentional step toward addressing this methodological gap. By providing both accessible hardware for data collection and open-source tools for performance characterisation, this work aims to encourage more rigorous reporting of localisation accuracy in future studies. The mod-

ular design of the system facilitates controlled validation experiments, such as playback studies using loudspeakers positioned at known locations, which can directly compare estimated positions against ground truth. Such empirical validation studies remain a priority for ongoing development and will be incorporated into future releases of both the hardware and analysis toolkit.

4.5 Extensibility and Future Development Directions

The architecture of Batsy4-Pro is inherently extensible in both hardware and software dimensions. The Teensy 4.1 platform supports dual I^2S interfaces capable of handling up to eight synchronous audio channels with appropriate ADC configurations, and time-division multiplexing (TDM) protocols can extend this to sixteen channels for applications requiring higher spatial resolution or increased angular coverage. While TDM implementation is beyond the scope of the current work, the underlying microcontroller capabilities and firmware structure are compatible with such expansion, requiring modifications primarily to the I^2S configuration and buffer management routines.

Sampling rate extensibility represents another avenue for development. Although the current implementation operates at 192 kHz—adequate for most bat species and ultrasonic bioacoustic applications—certain taxa produce calls with substantial energy above 120 kHz, including many rhinolophid and hipposiderid species. Higher sampling rates (384 kHz or 768 kHz) can be achieved using alternative ADC hardware, such as the AK5572 from Kaamos Tech [57], which offers dual-channel 32-bit conversion at up to 768 kHz. Similarly, while the current system employs 16-bit resolution to maintain compatibility with the *Teensy Audio Library* and minimise memory requirements, the WM8782 ADCs support 24-bit conversion. Implementing 24-bit recording would require custom firmware that bypasses the Audio Library's 16-bit constraints, but such increased bit depth is unlikely to yield substantial practical improvements for most field applications, where environmental noise rather than quantisation noise determines the effective signal-to-noise ratio.

Beyond audio acquisition, the Teensy 4.1 provides extensive support for auxiliary sensors via I^2C, and serial interfaces. Integration of GPS modules for geolocation and precision timestamping, environmental sensors (temperature, humidity, barometric pressure) for atmospheric correction, passive infrared or photogate sensors for motion-based triggering, and anemometers for wind monitoring all represent straightforward extensions that can enhance the contextual richness of acoustic datasets. Such multimodal data streams are particularly valuable for behavioural ecology, where understanding the environmental conditions and physical constraints under which animals operate is essential for interpreting acoustic behaviour.

Channel count expansion, although not yet implemented in the publicly released firmware, requires only incremental modifications to the existing codebase. The underlying technical implementation—I^2S clock configuration, buffer management, and file-writing routines—scales naturally to additional channels, with the primary constraint being the finite buffer memory available on the Teensy 4.1. With careful memory management and the use of external PSRAM modules currently employed, eight-channel or even sixteen-channel configurations are technically feasible within the existing hardware platform.

4.6 Broader Implications for Behavioural Research and Conservation

The availability of low-cost, field-ready multichannel ultrasonic recording systems has implications that extend beyond individual research projects. Behavioural experiments with echolocating bats have historically been concentrated in a small number of laboratories with the resources and expertise to develop custom acquisition systems. This concentration has resulted in a narrow taxonomic focus, with fewer than 30 species of echolocating bats (personal approximation) having been studied in experimental behavioural contexts—a small fraction of the more than 1300 known species [58]. Expanding the taxonomic breadth of experimental bioacoustics requires democratising access to the tools and methods that enable such work.

Tropical and subtropical regions, which harbour the greatest diversity of bat species and represent critical areas for conservation, are notably underrepresented in the behavioural bioacoustics literature. This geographic bias reflects not only the concentration of research infrastructure in temperate developed nations but also the practical challenges of conducting fieldwork in remote, resource-limited settings. Portable, affordable recording systems that require minimal technical infrastructure and can be operated by researchers with basic training lower the barrier to field experimentation in these regions. By releasing Batsy4-Pro as open-source hardware with permissive licensing, this work aims to facilitate local production and distribution in countries where commercial ultrasonic recording equipment is prohibitively expensive or unavailable.

Recent work using Batsy4-Pro to investigate spectral ripple structure in bat echolocation sequences [59] demonstrates how such portable systems enable direct empirical testing of ecological and sensory theories under natural field conditions. By allowing hypotheses derived from theoretical and computational frameworks [30] to be evaluated against spatially and temporally resolved acoustic data, field-ready multichannel recorders serve as essential instruments for validation/falsification in behavioural ecology.

Educational and citizen science applications represent another domain where accessible ultrasonic recording technology can have a significant impact. Current single-channel bat detectors and recorders have proven popular among amateur naturalists, educators, and conservation volunteers, fostering engagement with chiropteran ecology and building public support for bat conservation initiatives. Enabling behavioural experiments—rather than solely passive monitoring—in educational contexts can deepen this engagement and contribute to scientific literacy. Students and citizen scientists equipped with multichannel recording systems can participate meaningfully in research projects, collecting high-quality data that complements professional investigations and expands the geographic and temporal scope of acoustic monitoring efforts.

ACKNOWLEDGEMENTS

I thank Uwe Firzlaff for valuable discussions and constructive feedback during the development of the recording system, and Christian Fink for his assistance with the machining and assembly of the microphone array. I also thank the two anonymous reviewers, whose diligent engagement and constructive comments substantially improved the clarity, scope, and overall quality of this study.

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.

ETHICS STATEMENT

All data were obtained using passive, non-invasive acoustic recordings of free-flying bats. No animals were handled or subjected to experimental manipulation. The study did not require ethical approval or permits, in accordance with local and institutional guidelines.

FUNDING

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

CONFLICT OF INTEREST

The author declares no conflict of interest.

REVISION SUMMARY

The manuscript has been substantially revised and expanded in response to the editor and reviewers' comments, with the overarching goal of improving accessibility for a broader bioacoustics readership while strengthening performance validation and contextualisation. Key revisions include:

(i) Major reorganisation of the system description, now led by a hardware/firmware architecture and signal-flow overview.

(ii) Addition of a terminology table defining acronyms and embedded audio concepts in accessible language, and a table with comparison to prominent commercially available/open source recorders for contrasting placement of novelty.

(iii) Inclusion of two dedicated flowcharts describing the system architecture and full recording sequence, explicitly linking the two firmware-defined capture modes (*TAP*: retrospective buffer dump; *HOLD*: buffered-forward recording with prefill) to the ring-buffer and WAV writing steps.

(iv) Expansion of Methods/Results to quantitatively support core performance claims, including SNR characterisation and a grid-based Monte-Carlo localisation uncertainty analysis using the Array WAH toolkit and Widefield Acoustics Heuristics.

(v) A substantially expanded Discussion and comparative table situating present system against representative commercial and open-source systems, clarifying novelty, intended use-cases, and extensibility.

Further, the manuscript now clarifies that Batsy4-Pro has obtained OSHWA certification (DE165), providing externally validated openness standards. Hardware design files and firmware are released under permissive licenses (CERN-OHL-S, GPLv3).

Manuscript Information

Version: V2
Last updated: January 22, 2026
Words in Text: ca. 8400
Tables and Figures: 9

REFERENCES

1. Pearce, J. M. Economic Savings for Scientific Free and Open Source Technology: A Review. *HardwareX* **8,** e00139. doi:10.1016/j.ohx.2020.e00139 (Oct. 1, 2020).
2. Oellermann, M. *et al.* Open Hardware in Science: The Benefits of Open Electronics. *Integrative and Comparative Biology* **62,** 1061–1075. doi:10.1093/icb/icac043. PMID: 35595471 (May 20, 2022).
3. Ackermann, J. R. Toward Open Source Hardware. *University of Dayton Law Review* **34** (2009).
4. Pearce, J. M. Building Research Equipment with Free, Open-Source Hardware. *Science* **337,** 1303–1304. doi:10.1126/science.1228183 (Sept. 14, 2012).
5. Pearce, J. M. Open-Source Lab How to Build Your Own Hardware and Reduce Research Costs. Elsevier, New York. - References - Scientific Research Publishing. *Modern Economy* **6** (Jan. 2015).
6. Powell, A. B. Open Culture and Innovation: Integrating Knowledge across Boundaries. *Media, Culture & Society* **37,** 376–393. doi:10.1177/0163443714567169 (Apr. 1, 2015).
7. GOSH. *Global Open Science Hardware Roadmap - Gathering for Open Science Hardware* https://openhardware.science/global-open-science-hardware-roadmap/ (2025).
8. Whytock, R. C. & Christie, J. Solo: An Open Source, Customizable and Inexpensive Audio Recorder for Bioacoustic Research. *Methods in Ecology and Evolution* **8,** 308–312. doi:10.1111/2041-210X.12678 (2017).
9. Hill, A. P., Prince, P., Snaddon, J. L., Doncaster, C. P. & Rogers, A. AudioMoth: A Low-Cost Acoustic Device for Monitoring Biodiversity and the Environment. *HardwareX* **6,** e00073. doi:10.1016/j.ohx.2019.e00073 (Oct. 1, 2019).
10. NABU, e. *Raspberry Pi Bat Projekt* Raspberry Pi Bat Projekt. http://bat-pi.eu/ (2025).
11. Van Helden, P. The Cost of Research in Developing Countries. *EMBO reports* **13,** 395–395. doi:10.1038/embor.2012.43 (May 2012).
12. Stowell, D. Computational Bioacoustics with Deep Learning: A Review and Roadmap. *PeerJ* **10,** e13152. doi:10.7717/peerj.13152 (Mar. 21, 2022).
13. Aodha, O. M. *et al. Towards a General Approach for Bat Echolocation Detection and Classification* https://www.biorxiv.org/content/10.1101/2022.12.14.520490v1 (2026). Pre-published.
14. Stoffregen, P. & Coon, R. *Teensy® 4.1* PJRC. https://www.pjrc.com/store/teensy41.html (2025).
15. Umadi, R. *BATSY4-PRO: OSWHA Certification* version 1.0. Germany, Jan. 15, 2026.
16. Beason, R. D., Riesch, R. & Koricheva, J. AURITA: An Affordable, Autonomous Recording Device for Acoustic Monitoring of Audible and Ultrasonic Frequencies. *Bioacoustics* **28,** 381–396. doi:10.1080/09524622.2018.1463293 (July 4, 2019).
17. Stoffregen, P. & Coon, R. *Teensy Audio Library* Sherwood, OR. USA: PJRC.COM, LLC., 2015.
18. Cirrus Logic, I. N. C. WM8782 - 24-Bit 192 kHz Stereo ADC Datasheet. *Cirrus Logic* (2020).
19. NXP Semiconductors, N. V. *I.MX RT1060 Processor Reference Manual R3* (2021).
20. Tierney, J., Rader, C. & Gold, B. A Digital Frequency Synthesizer. *IEEE Transactions on Audio and Electroacoustics* **19,** 48–57. doi:10.1109/TAU.1971.1162151 (Mar. 1971).
21. Leis, J. W. *Digital Signal Processing Using MATLAB for Students and Researchers* 396 pp. (John Wiley & Sons, Inc, 2011).
22. Tan, L. & Jiang, J. *Digital Signal Processing - Fundamentals and Applications* (2014).

23. Umadi, R. ESPERDYNE: A Dual-Band Heterodyne Monitor and Ultrasound Recorder for Bioacoustic Field Surveys. *Methods in Ecology and Evolution* **n/a.** doi:10.1111/2041-210x.70241 (2026).

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

25. Umadi, R. *Bat Reviewer: Version 1.0 - With Installation Packages* Zenodo, Dec. 9, 2025. doi:10.5281/zenodo.17870453.

26. Umadi, R. *Biosonar Responsivity Analysis Toolkit V1.0.0* version V1.0.0. Zenodo, July 3, 2025. doi:10.5281/zenodo.15801316.

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

28. Umadi, R. *Array WAH: Microphone Array Design Tool* version V1.1.0. Zenodo, Aug. 2025. doi:10.5281/zenodo.15691372.

29. Umadi, R. Widefield Acoustics Heuristic: Advancing Microphone Array Design for Accurate Spatial Tracking of Echolocating Bats. *BMC Ecology and Evolution* **25,** 92. doi:10.1186/s12862-025-02441-4 (Sept. 8, 2025).

30. Umadi, R. & Firzlaff, U. *Biosonar Responsivity Sets the Stage for the Terminal Buzz* https://www.biorxiv.org/content/10.1101/2025.06.16.659925v2 (2025). Pre-published.

31. Brinkløv, S., Jakobsen, L., Ratcliffe, J. M., Kalko, E. K. V. & Surlykke, A. Echolocation Call Intensity and Directionality in Flying Short-Tailed Fruit Bats, Carollia Perspicillata (Phyllostomidae). *The Journal of the Acoustical Society of America* **129,** 427. doi:10.1121/1.3519396 (Feb. 2, 2011).

32. Surlykke, A., Pedersen, S. B. & Jakobsen, L. Echolocating Bats Emit a Highly Directional Sonar Sound Beam in the Field. *Proceedings of the Royal Society B: Biological Sciences* **276,** 853–860. doi:10.1098/rspb.2008.1505. PMID: 19129126 (2009).

33. De Framond, L., Beleyur, T., Lewanzik, D. & Goerlitz, H. R. Calibrated Microphone Array Recordings Reveal That a Gleaning Bat Emits Low-Intensity Echolocation Calls Even in Open-Space Habitat. *Journal of Experimental Biology* **226,** jeb245801. doi:10.1242/jeb.245801 (Sept. 27, 2023).

34. Eitan, O. *et al.* Echolocating Bats Rapidly Adjust Their Mouth Gape to Control Spatial Acquisition When Scanning a Target. *BMC Biology* **20,** 282. doi:10.1186/s12915-022-01487-w (Dec. 17, 2022).

35. Kounitsky, P. *et al.* Bats Adjust Their Mouth Gape to Zoom Their Biosonar Field of View. *Proceedings of the National Academy of Sciences* **112,** 6724–6729. doi:10.1073/pnas.1422843112. PMID: 25941395 (2015).

36. Jakobsen, L. *et al.* Velocity as an Overlooked Driver in the Echolocation Behavior of Aerial Hawking Vespertilionid Bats. *Current Biology,* S096098222401710X. doi:10.1016/j.cub.2024.12.042 (Jan. 2025).

37. Xia, H., Ma, N., Li, A. & Luo, J. Call Production and Wingbeat Coupling Is Flexible and Species-Specific in Echolocating Bats. *Annals of the New York Academy of Sciences* **1547,** 105–115. doi:10.1111/nyas.15325 (Mar. 30, 2025).

38. Gaudette, J. E. & Simmons, J. A. *Observing the Invisible Using Microphone Arrays to Study Bat Echolocation* in (Acoustics Today, Sept. 2014).

39. Jens, K. Arrayvolution: Using Microphone Arrays to Study Bats in the Field (2018).

40. Wathen-Dunn, W. On the Reciprocity Free-Field Calibration of Microphones. *The Journal of the Acoustical Society of America* **21,** 542–546. doi:10.1121/1.1906545 (Sept. 1, 1949).

41. *IEC 61094-1: Measurement Microphones Part 1: Specifications for Laboratory Standard Microphones* version 2.0. Geneva, Switzerland, July 21, 2000.

42. *IEC 61094-3: Measurement Microphone Part 3: Primary Method for Free-Field Calibration of Laboratory Standard Microphones by the Reciprocity Technique* version 2.0. Geneva, Switzerland, June 14, 2016.

43. MacDonald, J. A. & Tran, P. K. Loudspeaker Equalization for Auditory Research. *Behavior Research Methods* **39,** 133–136. doi:10.3758/BF03192851 (Feb. 1, 2007).

44. Barrera-Figueroa, S. Free-Field Reciprocity Calibration of Measurement Microphones at Frequencies up to 150 kHz. *The Journal of the Acoustical Society of America* **144,** 2575–2583. doi:10.1121/1.5063815 (Oct. 31, 2018).

45. Pierce, G. W. & Griffin, D. R. Experimental Determination of Supersonic Notes Emitted by Bats. *Journal of Mammalogy* **19,** 454–455. doi:10.2307/1374231 (Nov. 1, 1938).

46. Griffin, D. R. & Galambos, R. The Sensory Basis of Obstacle Avoidance by Flying Bats. *Journal of Experimental Zoology* **86,** 481–506. doi:10.1002/jez.1400860310 (1941).

47. Griffin, D. R. *Listening in the Dark: The Acoustic Orientation of Bats and Men* xviii, 413. xviii, 413 (Yale Univer. Press, Oxford, England, 1958).

48. Griffin, D. R. Return to the Magic Well: Echolocation Behavior of Bats and Responses of Insect Prey. *BioScience* **51,** 555–556. doi:10.1641/0006-3568(2001)051[0555:RTTMWE]2.0.CO;2 (July 1, 2001).

49. Stidsholt, L. *et al.* A 2.6g Sound and Movement Tag for Studying the Acoustic Scene and Kinematics of Echolocating Bats. *Methods in Ecology and Evolution* **10,** 48–58. doi:10.1111/2041-210x.13108 (2019).

50. Stidsholt, L., Johnson, M., Goerlitz, H. R. & Madsen, P. T. Wild Bats Briefly Decouple Sound Production from Wingbeats to Increase Sensory Flow during Prey Captures. *iScience* **24,** 102896. doi:10.1016/j.isci.2021.102896 (Aug. 2021).

51. Stidsholt, L. Unlocking the Mysteries of Biosonar Use and Foraging Behaviour in Wild Bats with On-Board Acoustic Recording Tags. *Journal of Experimental Biology* **228,** jeb250777. doi:10.1242/jeb.250777 (Aug. 15, 2025).

52. Stidsholt, L. *et al.* Greater Noctule Bats Prey on and Consume Passerines in Flight. *Science* **390,** 178–181. doi:10.1126/science.adr2475 (Oct. 9, 2025).

53. Stidsholt, L., Zebele, M., Scholz, C. & Voigt, C. C. Wild Bats Hunt Insects Faster under Lit Conditions by Integrating Acoustic and Visual Information. *Proceedings of the National Academy of Sciences* **122,** e2515087122. doi:10.1073/pnas.2515087122 (Sept. 16, 2025).

54. Yartsev, M. M. & Ulanovsky, N. Representation of Three-Dimensional Space in the Hippocampus of Flying Bats. *Science* **340,** 367–372. doi:10.1126/science.1235338 (Apr. 19, 2013).

55. Eliav, T. *et al.* Multiscale Representation of Very Large Environments in the Hippocampus of Flying Bats. *Science* **372,** eabg4020. doi:10.1126/science.abg4020 (May 28, 2021).

56. Palgi, S. *et al.* Head-Direction Cells as a Neural Compass in Bats Navigating Outdoors on a Remote Oceanic Island. *Science* **390,** eadw6202. doi:10.1126/science.adw6202 (Oct. 16, 2025).

57. Nihtila, T. *ADC AK5572 v2 - Very High Performance AD-converter in a Compact Form Factor.* Kaamos Tech Ltd. https://kaamostech.com/product/adc-ak5572-v2/ (2025).

58. *Handbook of the Mammals of the World : Vol. 9 : Bats* (eds Wilson, D. E. & Mittermeier, R. A.) (Lynx Edicions, 2019).

59. Umadi, R. *Rates, Ripples, and Responsivity: The Geometry of Echolocation in Water-Foraging Bats* https://www.biorxiv.org/content/10.64898/2026.01.10.698675v1 (2026). Pre-published.

SUPPLEMENTARY INFORMATION

A I²S CLOCK SETTING

```
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);
}
```

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

B RING BUFFER IN PSRAM

```


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
}
```

Listing 2: Ring Buffer Implementation

C HETERODYNE IMPLEMENTATION CODE

This appendix provides the complete firmware implementation of the real-time heterodyne monitoring subsystem described in Section 2.5.

C.1 Global Variables and Constants

The heterodyne system relies on a phase accumulator approach to generate a continuous sine wave carrier. The following global variables define the carrier frequency, phase state, and audio parameters:

```
// Heterodyne carrier parameters
const float FREQ_MIN   = 10000.0f;     // Minimum carrier frequency (Hz)
const float FREQ_MAX   = 85000.0f;     // Maximum carrier frequency (Hz)
const float FREQ_STEP  = 5000.0f;      // Frequency adjustment step (Hz)

volatile float carrierFreq = 45000.0f;  // Default carrier: 45 kHz
volatile float outGain     = 0.50f;     // Output gain: 0.0-1.0 (50% default)
const float VOL_STEP   = 0.05f;         // Volume adjustment step (5%)

// Phase accumulator variables
const float twoPi = 6.28318530718f;     // 2*pi constant
float phase = 0;                        // Current phase position [0, 2*pi]
float phaseIncrement = 0;               // Phase increment per sample

// Audio system parameters
const int sampleRate = 192000;          // Sampling rate: 192 kHz
const int numChannels = 4;              // Number of input channels
```

Listing 3: Global variables for heterodyne monitoring and audio processing.

The phase increment is calculated during initialisation based on the carrier frequency and sampling rate:

```
void setup() {
    // ... other initialisation code ...

    // Calculate phase increment for carrier generation
    phaseIncrement = twoPi * carrierFreq / sampleRate;

    // For 45 kHz carrier at 192 kHz sampling:
    // phaseIncrement = 6.283 * 45000 / 192000 = 1.472 radians/sample

    // ... rest of setup code ...
}
```

Listing 4: Phase increment initialization in `setup()`.

C.2 Real-Time Audio Processing with Heterodyne

The `readAndProcessAudio()` function is called continuously in the main loop and performs three operations simultaneously: (1) fills the PSRAM ring buffer with raw 4-channel data, (2) generates the heterodyne output from Channel 1, and (3) sends the heterodyned audio to the DAC for real-time monitoring.

```
void readAndProcessAudio() {
    // Check if all four input queues have data available
    if (queueL1.available() && queueR1.available() &&
```

```
        queueL2.available() && queueR2.available()) {

            // Get pointers to 128-sample blocks from each channel
            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();

            // Temporary buffer for heterodyned output
            static int16_t tempOut[128];

            // Process 128 samples (one audio block)
            for (int i = 0; i < 128; i++) {
                // 1. Write raw samples to ring buffer (all 4 channels, interleaved)
                ringBuffer[ringWriteIndex++] = ch1[i];
                ringBuffer[ringWriteIndex++] = ch2[i];
                ringBuffer[ringWriteIndex++] = ch3[i];
                ringBuffer[ringWriteIndex++] = ch4[i];
                if (ringWriteIndex >= ringBufferSamples) ringWriteIndex = 0;

                // 2. Generate a carrier sine wave using phase accumulator
                float carrier = sinf(phase);           // sin(phase)
                phase += phaseIncrement;               // Advance phase
                if (phase >= twoPi) phase -= twoPi;    // Wrap to [0, 2*pi]

                // 3. Heterodyne: multiply input by carrier and apply gain
                //    output[n] = input[n] * sin(2*pi*f_carrier*n/f_s) * gain
                float mixed = (float)ch1[i] * carrier * outGain;

                // 4. Hard limit to 16-bit signed integer range
                if (mixed > 32767.f)  mixed = 32767.f;
                if (mixed < -32768.f) mixed = -32768.f;
                tempOut[i] = (int16_t)mixed;
            }

            // Update total sample counter (for ring buffer readiness check)
            totalSamplesWritten += 128UL * numChannels;

            // Release input buffers
            queueL1.clear();
            queueR1.clear();
            queueL2.clear();
            queueR2.clear();

            // 5. Send heterodyned audio to DAC output (PCM5102A)
            if (outputQueue.available()) {
                int16_t *out = outputQueue.getBuffer();
                memcpy(out, tempOut, sizeof(tempOut));  // Copy 128 samples
                outputQueue.playBuffer();                // Stream to I2S output
            }
        }
    }
```

Listing 5: Real-time heterodyne processing with parallel ring buffer recording.

C.3 Dynamic Carrier Frequency Adjustment

The carrier frequency can be adjusted in real time using a rotary encoder. When the user rotates the encoder in `EDIT_FREQ` mode, the carrier frequency is updated, and the phase increment is recalculated:

```
    void applyEncoderSteps(int detents) {
```

```
        if (detents == 0) return;

        if (editMode == EDIT_FREQ) {
            // Adjust carrier frequency by step size
            carrierFreq += detents * FREQ_STEP;  // +/- 5 kHz per detent

            // Clamp to valid range [10 kHz, 85 kHz]
            if (carrierFreq < FREQ_MIN) carrierFreq = FREQ_MIN;
            if (carrierFreq > FREQ_MAX) carrierFreq = FREQ_MAX;

            // Recalculate phase increment for new frequency
            phaseIncrement = twoPi * carrierFreq / sampleRate;

            Serial.print("Carrier Frequency: ");
            Serial.print(carrierFreq);
            Serial.print(" Hz, phaseInc: ");
            Serial.println(phaseIncrement, 8);

        } else {  // EDIT_VOL mode
            // Adjust output gain (volume)
            outGain += detents * VOL_STEP;  // +/- 5% per detent

            // Clamp to valid range [0.0, 1.0]
            if (outGain < VOL_MIN) outGain = VOL_MIN;
            if (outGain > VOL_MAX) outGain = VOL_MAX;

            Serial.print("Output Volume: ");
            Serial.print(outGain * 100.0f, 0);
            Serial.println("%");
        }

        drawUI();  // Update OLED display
    }
```

Listing 6: Dynamic carrier frequency tuning via rotary encoder.

C.4 Mathematical Principle

The heterodyne process performs multiplication in the time domain, which corresponds to convolution in the frequency domain. Given an input signal $x[n]$ containing a spectral component at frequency f_{bat}, and a carrier sine wave at frequency f_{carrier}, the multiplication produces:

$$y[n] = x[n] \cdot \sin(2\pi f_{\text{carrier}} n / f_s) \cdot g \tag{1}$$

where $f_s = 192\,\text{kHz}$ is the sampling rate and g is the output gain. This operation generates sum and difference frequencies:

$$f_{\text{sum}} = f_{\text{bat}} + f_{\text{carrier}} \tag{2}$$

$$f_{\text{diff}} = |f_{\text{bat}} - f_{\text{carrier}}| \tag{3}$$

The sum frequency is typically above the Nyquist limit (96 kHz) and is naturally filtered by the DAC reconstruction filter. The difference frequency falls into the audible range, enabling real-time monitoring through standard headphones.

Example: A bat echolocation call at 62 kHz mixed with a 60 kHz carrier produces:

- Sum: $62 + 60 = 122\,\text{kHz}$ (inaudible, filtered)
- Difference: $|62 - 60| = 2\,\text{kHz}$ (audible tone)

This 2 kHz tone is heard in real time through the headphones, allowing the researcher to detect bat activity without viewing a spectrogram.

C.5 Performance Characteristics

- **Computational cost:** Per 128-sample block (0.67 ms at 192 kHz):
  - 128 calls to `sinf()` (hardware-accelerated)
  - 128 floating-point multiplications
  - 128 floating-point to integer conversions
  - Negligible impact on Teensy 4.1 @ 600 MHz
- **Latency:** <1 ms (one audio block buffering delay)
- **Frequency response:** Flat across tunable range (10–85 kHz)
- **Dynamic range:** Limited by 16-bit integer output (∼96 dB theoretical)
- **Harmonic distortion:** Minimal (depends on `sinf()` precision)
- **Interference:** None—operates in parallel with ring buffer recording

D BILL OF MATERIAL

Table 3: Bill of Materials for the Batsy4 Pro Recording System. The listed prices reflect procurement costs in Germany during development, from January to 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
APS6404L-3SQR PSRAM - 8MB -	External Memory	2	2.50	tinytronics.nl
PC5102A	I2S DAC	1	7.99	Amazon.de
MP1584EN	DC-DC 3A 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			**125.28**	

Table 4: 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	Tripod mount (or use 1/4” 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 rod	1	4.99	Hagebaumarkt
Polyester Fur	Acoustic insulation	1 m^2	8.99	Amazon (generic)
Cables and Miscellaneous	Extension cables, adhesives, etc.	N/A	20.00	Local purchases
Total			**81.42**	

E FIELD DATA - CALL DURATION STATS

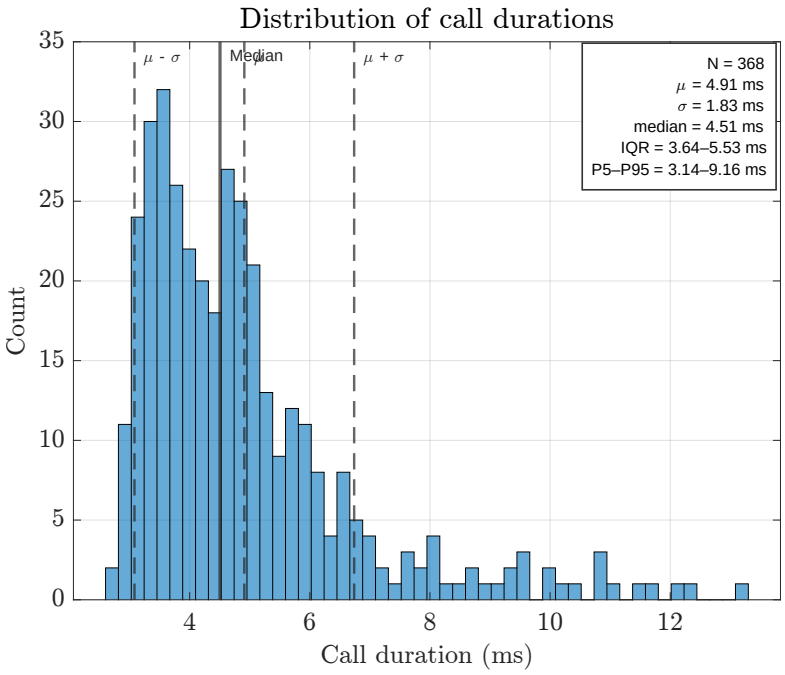

Figure 9: Distribution of echolocation call durations extracted from validated detections ($N = 368$ calls). Call duration was defined using an energy-based percentile criterion: for each call waveform segment, the cumulative sum of squared amplitude was normalised, smoothed, and the call duration was taken as the interval between the time points at which the cumulative energy crossed the lower and upper fractional thresholds (5–95% of total energy). The histogram shows a unimodal distribution centred on short-duration frequency-modulated calls. The vertical solid line indicates the median call duration (4.51 ms), while the shaded region denotes the interquartile range (IQR: 3.64–5.53 ms). Dashed lines mark the 5th and 95th percentiles (3.14–9.16 ms), illustrating the limited spread and absence of long-duration outliers.