MIROSLAV: A modular open-source hardware and software platform for high-throughput circadian activity measurement and analysis in rodents

Davor Virag [1,2], Jan Homolak [1,2,3], Ivan Kodvanj [4], Ana-Marija Vargantolić [5], Ana Babić Perhoč [1,2], Patrik Meglić [1,2], Petra Šoštarić Mužić [1,2], Ana Knezović [1,2], Jelena Osmanović Barilar [1,2], Mario Cifrek [6], Vladimir Trkulja [1], Melita Šalković-Petrišić [1,2]

[1] Department of Pharmacology, University of Zagreb School of Medicine, Zagreb, Croatia

[2] Croatian Institute for Brain Research, University of Zagreb School of Medicine, Zagreb, Croatia

[3] M3 Research Center, Excellence Cluster 'Controlling Microbes to Fight Infections' (CMFI), Interfaculty Institute of Microbiology & Infection Medicine Tübingen (IMIT), Eberhard Karls University & University Hospital Tübingen, 72076 Tübingen, Germany

[4] Department of Gastroenterology and Hepatology, University Hospital Center Zagreb, Zagreb, Croatia

[5] Health Centre Zagreb – West, Zagreb, Croatia

[6] University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia

Corresponding author:

Davor Virag
Department of Pharmacology
University of Zagreb School of Medicine
Zagreb, Croatia
E-mail: davor.virag@gmail.com
Tel.: 00385 98 864 307

Abstract

While conventional behavioural tests offer valuable insight into rodent behaviour in specific paradigms, rodents spend most of their time in a safe, undisturbed environment – their home cage. However, home cage monitoring (HCM) is often impractical at scale over prolonged periods of time without significant loss of data.

To achieve this, we developed MIROSLAV, the Multicage InfraRed Open Source Locomotor Activity eValuator, a home cage activity and environmental monitoring device designed to be open, adaptable, and robust. Its transparent and modular design allows the device to be tailored to varying experimental requirements and environmental conditions, while its wireless operation and multiple redundancies minimise loss of data. Data quality is maintained by a modular software workflow for data preparation (Prepare-a-SLAV) and cleaning (TidySLAV), followed by exploratory (MIROSine – MIRO The Explorer) and statistical (MIROSine – StatistiSLAV) analysis of circadian periodicity.

Here, using MIROSLAV, we demonstrate circadian dysrhythmia and a disrupted response to regular stimuli (e.g. behavioural testing, bedding change) in a rat model of sporadic Alzheimer's disease, showcasing MIROSLAV's utility in a typical animal study of disease.

In accordance with the 3Rs, every rodent study and laboratory performing them can benefit from HCM, provided the existence of transparent, customisable, tools which allow robust deployment to tens or hundreds of cages in various conditions. MIROSLAV provides this opportunity to researchers in a cost-effective manner.

Introduction

Circadian rhythms are present in organisms from bacteria to humans, from the molecular to the behavioural level [1]. Disruption of the circadian rhythm of locomotor activity has been recognised as a feature, risk factor, and possible etiopathogenetic component in a number of pathological conditions prevalent in the human population (e.g. Alzheimer's disease [2]). As such, it is a subject of growing interest in biomedical research, particularly in animal models of disease.

Circadian measurement of locomotor activity in laboratory rodents, however, requires a different approach than conventional behavioural tests, and is commonly performed using continuously-operating, non-invasive home cage monitoring (HCM) devices. This type of monitoring greatly increases the amount of data obtained from the same set of animals and furthers adherence to the 3Rs principle [3]. In recent years, newly developed HCM systems have rapidly increased in number, particularly custom, open-source solutions [4].

Today, commercial tools for behavioural analysis are developed as proprietary, closed source hardware and software solutions. Proprietary assemblies and undocumented hardware communication protocols make it nearly impossible for researchers to extend a single device according to ever-changing experimental requirements. Before researchers are given the acquired data, it is automatically pre-processed and cleaned through closed source algorithms which differ across devices and manufacturers, in direct opposition to reproducibility as a fundamental scientific principle. Raw data can sometimes be obtained in an unstandardised format via explicit request to the manufacturer.

To avoid these limitations, researchers have started developing modular open-source solutions that can be customised to the requirements of novel experiments, and quickly adapted to handle unforeseen circumstances inherent to animal research,

without needing to rely on a lengthy process of external testing and support. The development and publication of methods extensively documented and fully replicable in accordance with academic standards invites wider collaboration towards the uptake and further development of these novel solutions [5].

Inspired by previous methods built in accordance with this movement (e.g. Rodent Activity Detector – RAD [6], Continuous Open Mouse Phenotyping of Activity and Sleep Status – COMPASS [7]), we propose MIROSLAV, the Multicage InfraRed Open Source Locomotor Activity eValuator, as a complete HCM hardware and software toolkit built with three main goals in mind:

1. Long-term high-throughput monitoring of hundreds of cages at a time,
2. Robustness and redundancy to prevent data loss,
3. Extensive modularity to support a wide array of experimental contexts.

This paper will first describe the MIROSLAV method, followed by a proof-of-concept study demonstrating MIROSLAV's utility.

Materials and Methods

The Multicage InfraRed Open Source Locomotor Activity eValuator: MIROSLAV

MIROSLAV is an Arduino-based platform for continuous, non-invasive monitoring of home-cage locomotor activity in laboratory rodents. It consists of multiple cage-mounted sensor arrays (Fig. 1A), which register home cage activity. Cage-mounted sensor arrays are connected to the main unit's input serialisation stack (Fig. 1B), which feeds the readings to the main unit's mainboard (Fig. 1C). The mainboard timestamps the data and transmits it to an off-site computer via Wi-Fi. Data acquisition is performed with the Record-a-SLAV Python tool.

The MIROSLAV hardware repository contains circuit board designs (produced in KiCad [8], with respective Gerber files, allowing customisation and quick production via online services) [9], whereas the MIROSLAV firmware repository contains MIROSLAVino Arduino firmware and Record-a-SLAV acquisition software [10]. The MIROSLAV analysis repository contains a software toolkit for data preparation, cleaning, and analysis [11].

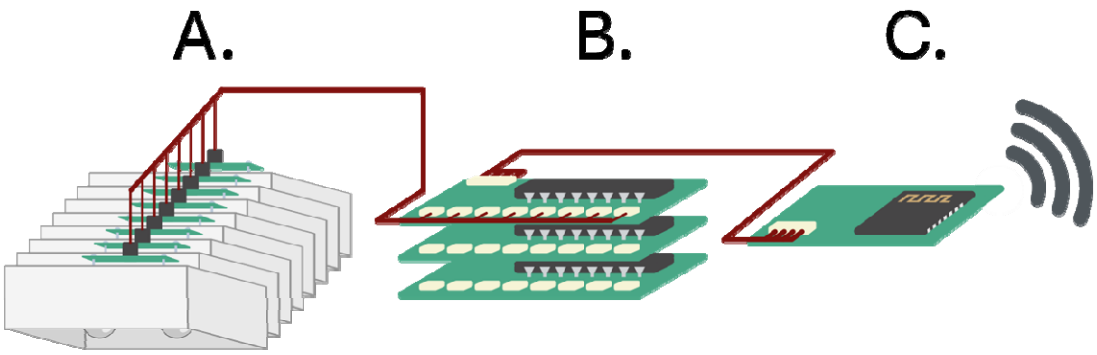


Figure 1. An illustration of the MIROSLAV device and its components. Cage sensor arrays (**A**) feed data into the main unit (**B**, **C**). The main unit consists of up to 64 input serialisation boards in a stack (**B**), with only the topmost board needing to be connected to the mainboard. The mainboard (**C**) timestamps the data and transmits it wirelessly to one or more recording devices.

Cage-mounted sensor array

The home cage-mounted sensor array (Fig. 2) consists of two passive infrared (PIR) modules which register the rodent's locomotor activity.

The array screws onto the cage lid and its sensors are partially encased with a modified PVC cable raceway to prevent IR interference from outside the cage. On top of the assembly lies a custom printed circuit board (PCB) which provides a single easily detachable connection for power and signal output to the main unit by way of an RJ9 socket.

The main sensor component is a pair of photovoltaic cells sensitive to infrared light, paired with a segmented Fresnel lens, which alternately focuses segments of the field of view to each of the cells. Movement of an IR-emitting body (like that of a warm-blooded animal) in the sensor's field of view therefore differentially excites the cells. The module's onboard BISS0001 integrated circuit [12] continuously calculates the difference between the signal of the two cells. When this difference crosses a threshold, the sensor's output flips from a logical 0 to a logical 1, signifying movement. This results in two binary values being transmitted from each cage, one per sensor module.

Power and data transmission to and from the sensor array is done with a single 4-wire cable affixed to the rack. On the main unit side, this cable splits into two JST connectors – one for power, and one for data transmission. On the cage side, the cable converges into an RJ9 plug (similar to Ethernet and telephone landline connectors) which maintains a reliable connection and enables zero pull-out and insertion force operation once the locking pin is pressed.

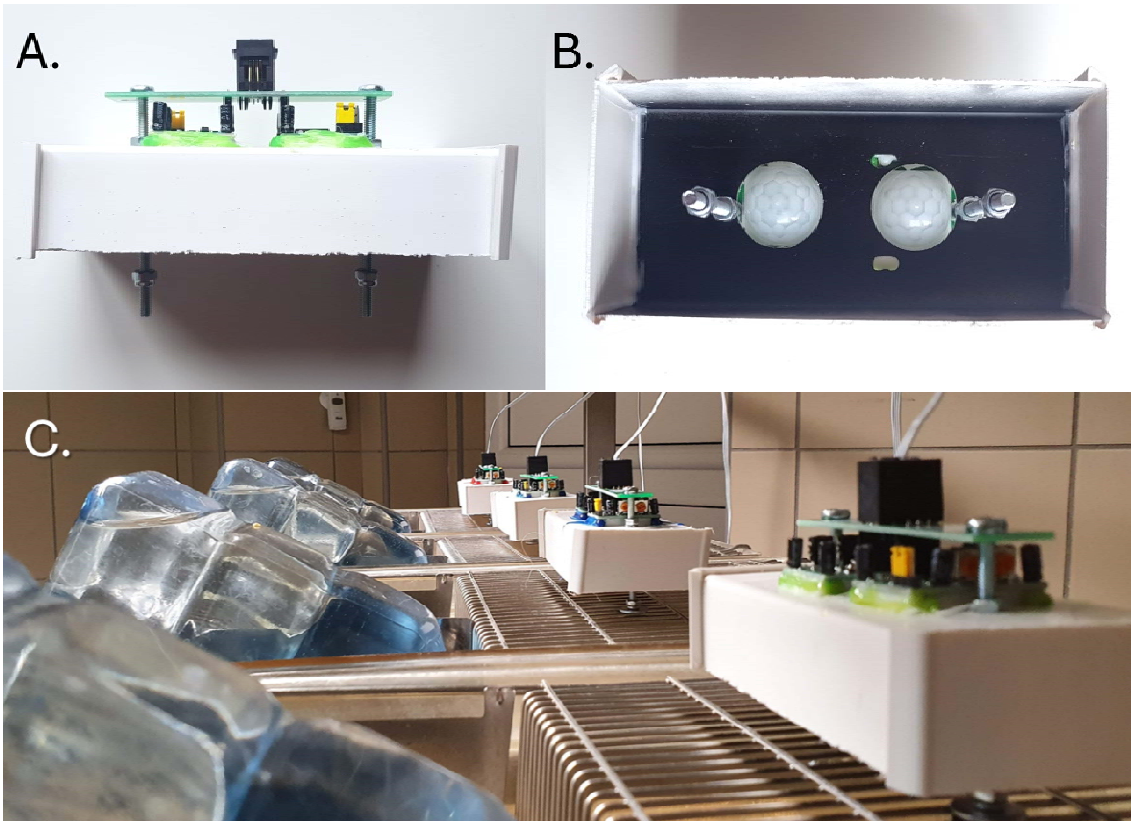


Figure 2. Cage-mounted sensor arrays, shown from **A.** the side, **B.** the bottom, revealing Fresnel lenses of the two sensor modules, and **C.** in operation as part of a typical MIROSLAV setup.

Main unit

Input serialisation stack

The input serialisation stack consists of one or multiple input boards. Up to 8 individual boards may be stacked together to expand a MIROSLAV's capacity for data input up to 64 cages (128 individual sensor modules).

Each input board consists of 8 cage input connectors, connectors for power and mainboard communication, and a stacking connector (Fig. 3A). When stacked, only the topmost board needs to be powered and connected to the mainboard. Other boards will receive power and have access to the data transmission bus towards the mainboard through the stacking connector. Thus, the entire stack can be extended to

the appropriate number of inputs while acting as one unit with a single power and data connection.

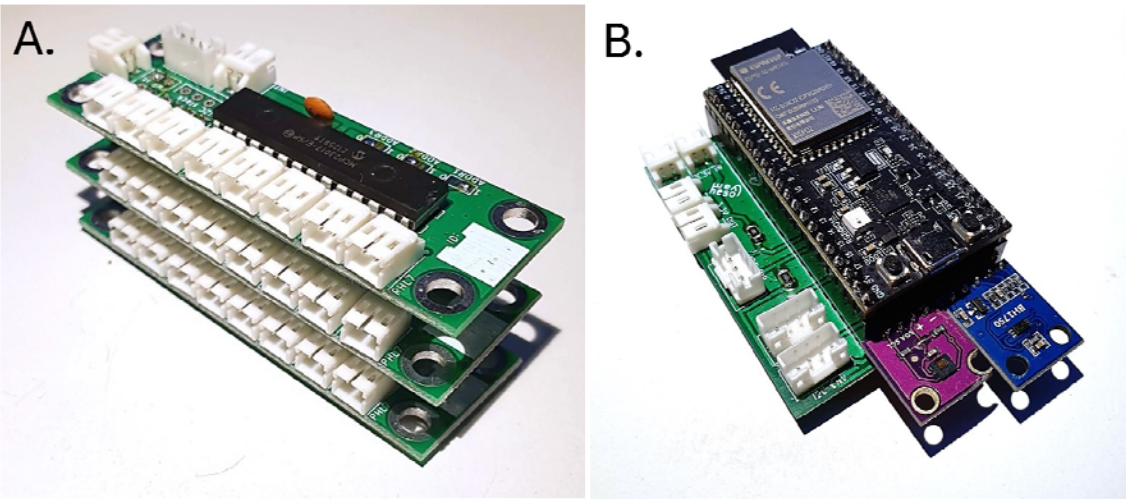


Figure 3. The MIROSLAV device main unit, consisting of **A.** the input serialisation stack, which consists of multiple stacked input boards collecting input from cage sensor arrays, serialising it, and sending it to **B.** the main unit for wireless transmission and recording.

Mainboard

The mainboard is based around the Wi-Fi-enabled ESP32-S2 microcontroller compatible with the Arduino software framework (Fig. 3B). The mainboard polls the input serialisation stack for sensor data and transmits it wirelessly over the MQTT protocol to one or more computers dedicated to data logging.

Aside from the cage sensor data received from the input serialisation stack, the mainboard also records environmental data. Onboard sensors measure ambient light level, temperature, and humidity, and a PIR sensor connected to a dedicated socket monitors the habitat for personnel entrances. An example of environmental data gathered with a MIROSLAV prototype, parsed, and plotted using EnviroSLAV is shown in Fig. 4.

The mainboard may be extended with a custom board (containing, for example, sensors for specific gases or a microphone for noise level monitoring) through the additional “I2C ENV” connector, which exposes the microcontroller’s I^2C bus.

MIROSLAV also supports operation as a standalone environmental monitor. This configuration only requires the MIROSLAV mainboard with an attached PIR sensor, and a power supply such as a smartphone charger.

Figure 4. An example of the data gathered from a MIROSLAV's environmental sensors. Pink bars denote missing data. Plot is produced with EnviroSLAV, the MIROSLAV environmental data monitoring and plotting tool, available in the MIROSLAV analysis toolkit repository [11].

Power boards

The cage sensor arrays and the mainboard may be powered by 5V supplies, commonly found as smartphone chargers. Consideration needs to be given to make sure the power supply's current capabilities meet a MIROSLAV's needs, particularly in regard to the number of sensors. However, a MIROSLAV's sensors may be distributed across multiple power boards, each with its own power supply. If necessary, the power supply boards may also be adapted based on the provided designs.

The input serialisation stack doesn't require additional power or connection to the power boards. It is always powered directly from the mainboard via the same connection responsible for data transmission.

Infrastructure

Wireless data acquisition

To utilise MIROSLAV's main mode of data acquisition, a Wi-Fi network is required. Data is sent using the MQTT protocol which is designed to support applications with thousands of messages per second exchanged between many senders and receivers [13,14], with a single device, e.g. a personal computer (PC) or a router, acting as a "broker". Multiple on- or off-site computers may be configured to save data from many MIROSLAVs simultaneously for data redundancy, with cloud recording also supported. Additionally, MIROSLAVs are configured by default to synchronise against a local or Internet NTP (Network Time Protocol) server for precise message timestamping across many concurrently running devices.

The default MIROSLAV configuration is set up for 10 messages per second, with each message containing data about the current state of all attached sensor arrays, theoretically allowing for upwards of 100 MIROSLAVs (each with up to 64 cage sensor arrays).

Direct data acquisition via USB connection

Alternatively, a MIROSLAV may also be attached directly to a computer via USB connection without requiring a wireless network. This mode is limited to one storage device which needs to be placed within the animal habitat in close proximity to the MIROSLAV, reducing robustness. As such, this mode is intended for testing and short-term recording.

Data acquisition

The accompanying software is used to: i) get current sensor states and transmit them wirelessly to one or more logging PCs; ii) save these readings as compressed

raw data; iii) preprocess the raw data for subsequent experiment-dependent analyses. All software is licensed under the GNU Public License version 3 (GPLv3) [15].

MIROSLAV mainboard

The MIROSLAV's onboard microcontroller is programmed in C++ atop the Arduino software framework. All user-configurable parameters (network settings, polling rates, enabling/disabling environmental monitoring) may be adjusted in the "config.h" file without the need to modify C++/Arduino code. A running MIROSLAV can be updated with a new configuration remotely through a web browser using the *HTTPUpdateServer* library included in the Arduino ESP32 core [16].

Data logging to a PC: Record-a-SLAV

A Python script, *Record-a-SLAV.py*, runs on logging PCs continuously for the duration of the experiment and saves MIROSLAV data to compressed files. The same script can be run on many PCs simultaneously for redundancy.

Data analysis

Raw data from the MIROSLAV hardware setup is stored as a compressed textual file. Data analysis steps consist of: i) cleaning and preprocessing, to generate a tall, tidy [17] columnar file with appropriate labels and discarded faulty data; ii) exploratory data analysis, for initial data examination and validation; iii) statistical data analysis, to test for the existence of putative treatment/intervention effects.

For this process, we have prepared a versatile Python/R-based pipeline, written with the objective of giving the researcher full control over their data and all steps involved. The scripts are highly configurable and intermediate files are generated at multiple steps to allow for combined analysis using existing software libraries and packages such as *Cosinor* [18], *Kronos* [19], *GLMMcosinor* [20], and others.

Data preprocessing: Prepare-a-SLAV

The Python preprocessing script, *1_Prepare-a-SLAV.py*, utilises the *mirofile* library to load raw MIROSLAV data into a *pandas* data frame. Appropriate animal IDs are applied, and the data is downsampled from the default MIROSLAV sampling rate (10 sensor readings/binary values per second) to an arbitrary, user-defined time interval (bin). For example, resampling a sensor's readings to a 5-minute bin will turn 864.300 binary values sampled over one day into 288 decimal values by taking the mean of every 3.000 consecutive readings. Narrow bins (seconds, minutes) can give insight into short events, while wide bins (tens of minutes, hours) facilitate visual inspection of hourly, daily, or weekly trends. All parameters are set via a separate, human-readable TOML (*Tom's Obvious Minimal Language* [21]) configuration file, *1_Prepare-a-SLAV_config.toml*.

Data cleaning: TidySLAV

The Python cleaning script, *2_TidySLAV.py*, primarily turns the table into a tidy, tall format [17]. This enables easy addition of experimental metadata and later analysis in a statistical software package. Additionally, some columns are removed, such as disconnected and faulty sensors, as specified by the user. Brief periods when operational sensors were disconnected (e.g. due to behavioural testing or bedding change) are automatically recognised and set to *NA*, indicating an unknown activity state. Finally, the time series are standardised according to each sensor's respective baseline measurements taken before treatment application.

Circadian (24-hour) activity rhythm analysis: MIROSine

The objective of *MIROSine* (provided as the R script *3_MIROSine.R*) is to analyse the animals' circadian rhythm of locomotor activity over the duration of the experiment.

Simple periodicity with a defined period (24 hours in the case of a circadian rhythm) is well-described by the sinusoidal function:

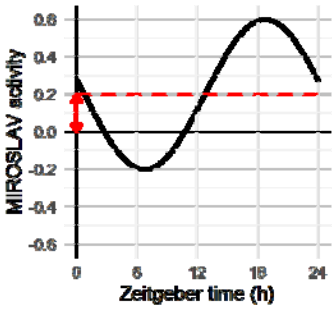
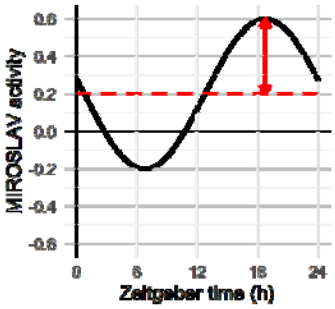
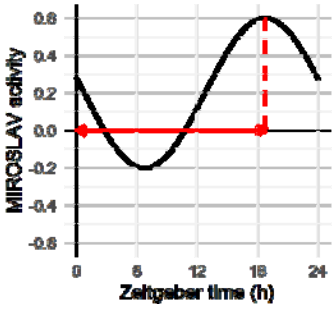
$$f(x) = M + A * \sin(x + \phi)$$

The parameters of the function are i) M – the midline, the sinusoid's equilibrium; ii) A – the amplitude, the sinusoid's maximum offset from its midline; iii) ϕ – the phase shift, or the horizontal offset of the sinusoid's peaks and troughs. Cornelissen defines these parameters as the MESOR (Midline Estimating Statistic of Rhythm (MESOR), amplitude, and acrophase, respectively, and proposes the *cosinor* statistical model to estimate them [22]. Note that while the terms "acrophase", "phase shift", and "peak time of day" refer to the same biological phenomenon, the exact definitions are somewhat different. A visualisation of these parameters is presented in Table 1.

In a manner similar to Cornelissen's approach, MIROSine estimates these parameters for every day of each sensor's readings across the duration of the experiment using R's *lm* function [23].

These parameters may then be estimated for an entire group across a specified time interval, as exemplified in the following exploratory and statistical analysis steps performed by MIRO The Explorer and StatistiSLAV, respectively. Care should be taken while interpreting these parameters for days when cage arrays have been disconnected for prolonged periods of time (e.g. due to behavioural testing), as the missing values could skew model fit. TidySLAV heatmaps can be used to find and assess intervals of missing values.

Table 1. A summary of the three sinusoid parameters.

	Definition	Value in the example	Visualisation (parameter in red)
Midline, MESOR	Midline Estimating Statistic of Rhythm, a rhythm-adjusted mean, the central value over which the sinusoid oscillates.	Mean activity over one full 24-hour period. Value in example plot: 0.2	
Amplitude	Difference between the midline and the sinusoid's peak value.	Value of peak activity relative to the day's mean (MESOR), difference between minimum and maximum activity (2*amplitude). Value in example plot: 0.4	
Phase shift, Acrophase, Peak time of day	<u>Phase shift</u>: Latency of the sinusoid along the axis (time). <u>Acrophase</u>: Zeitgeber time (ZT; time since lights-on) when peak activity occurs. <u>Peak time of day</u>: Clock time when peak activity occurs	Time of peak activity onset since lights on (Zeitgeber). If lights-on occurs at 6:00, time of day of peak activity occurs at 6:00 + [ZT of peak activity]. Acrophase in example plot: 18h 46min Peak time of day in example plot (lights on/ZT=0 at 6:00): 00:46	

Exploratory analysis of circadian rhythm parameters: MIRO The Explorer

Rhythm change is assessed by observing the temporal dynamics of these parameters within and between the experimental groups over the duration of the experiment. MIRO The Explorer offers visual assessment via two graphical approaches by utilising the *ggplot2* package [24]: *Parameter View* and *Rhythm Simulation*. Events such as behavioural testing and bedding changes are also labelled in these visualisations.

Parameter view shows the changes of sinusoid parameters (midline, amplitude, phase shift) per group across the days of the experiment. One time series plot is produced for each parameter, and all groups' daily means and standard deviations of the respective parameter are shown (PoC study example: Fig. 6A).

Rhythm simulation draws group circadian activity curves throughout the experiment. A group's daily medians of the three sinusoid parameters are used to reconstruct a group-representative sinusoid for that day. This is repeated for all experimental days, and a smoothed curve is drawn through the simulated values (PoC study example: Fig. 6B).

Statistical analysis of circadian rhythm parameters: StatistiSLAV

Statistical analysis is performed using the R script *4_StatistiSLAV.R*. Intervals of interest for comparison are selected. One interval may span multiple consecutive days, provided that MESOR, amplitude, and acrophase are stationary across the selected days within each treatment group based on the exploratory analyses.

A generalised linear mixed effects model with restricted maximum likelihood estimation is fitted for each of the three parameters using the *glmmTMB* package [25]. Residuals are simulated using the *DHARMa* package [26] and its diagnostic plots are inspected for model assumption violations. Estimated marginal means of the parameters are produced using the *emmeans* package [27]. The data and model specification were checked against the workflow for linear modelling in R by Santon et al. [28].

Two types of visualisations are produced for each time interval. The first are dot plots showing estimated parameter means and 95% confidence intervals for all treatment groups, drawn for each of the sinusoid parameters. The estimated group means of the three parameters are used to reconstruct each group's sinusoidal representation of circadian activity. The two sinusoids are overlayed in one plot

representing each group's circadian activity for easier interpretation on how the parameter differences affect circadian activity (PoC study example: Fig. 7). This intuitive visualisation is inspired by the *cosinoRmixedeffects* package developed by Hou et al. [29].

Finally, hypothesis testing of group differences per parameter is performed using the *emmeans* package's *contrasts* function. The Holm method is suggested to control the family-wise error rate.

A proof-of-concept study: Circadian dysrhythmia in a rat model of Alzheimer's disease

The experiment was performed to establish a comprehensive behavioural phenotype of the streptozotocin-induced rat model of sporadic Alzheimer's disease (manuscript in preparation). A subset of the data, presented in this work, was also used to demonstrate the validity of the MIROSLAV method, as per the Reduce principle of the 3Rs [30,31].

The rat model of sporadic Alzheimer's disease

All animal procedures were approved by the Croatian Ministry of Agriculture (EP 378/2022).

The sporadic Alzheimer's disease (AD) model was induced via intracerebroventricular (icv) injection of streptozotocin (STZ) following a standard procedure used and optimised by our research group [32–34] since the model's introduction by Lacković and Šalković [35], and Hoyer et al. [36]. Thirty 3-month-old male Wistar rats, bred at the Department of Pharmacology, University of Zagreb School of Medicine, were randomly divided into a treatment and a control group. The

control group (CTR, n = 15) was given 0.05 M citrate buffer (pH 4.5) intracerebroventricularly, and for the streptozotocin-treated group (STZ, n = 15), STZ (1.5 mg/kg) was freshly dissolved in the citrate buffer. Citrate buffer or STZ solution (2 μL) was given into each ventricle twice across 48 hours, at coordinates −1.5 mm posterior, ±1.5 mm lateral, and +4 mm ventral to the bregma [32,37–39]. For the procedure, the animals were anaesthetised with an intraperitoneal injection of ketamine/xylazine (70/7 mg/kg).

Animal housing

Animals were kept at the Department of Pharmacology's licensed animal facility (HR-POK-007). The facility was kept at 21-23°C and 40-70% relative humidity. The lighting cycle was 12:12 h, with lights on from 05:46 (Zeitgeber time [ZT] 0) to 17:46 (ZT 12). The animals were single-housed starting 7 days prior to STZ administration. Animals had access to standard food pellets and water *ad libitum*. Bedding change was performed on a weekly basis.

MIROSLAV recording and other in vivo procedures

MIROSLAV sensor arrays were fitted to the cage 4 days prior to STZ administration (day 0) to record baseline activity measurements. Activity was recorded for the following 25 days. Notable events of interest for MIROSLAV recording were weighing, bedding change, and behavioural testing. Animals were weighed on a weekly basis during bedding change. Behavioural testing was performed from day 16 to day 20 in the following order: i) Open field; ii) Novel object recognition; iii) Social interaction; iv) Morris water maze; v) PASTA startle and prepulse inhibition [40]. A schematic illustrating the timeline is shown in Fig. 5.

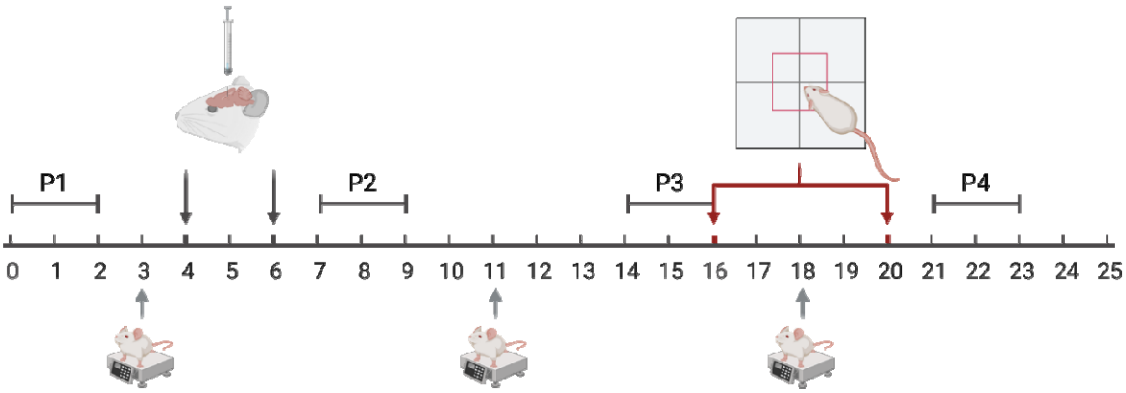


Figure 5. A schematic of the experiment timeline. MIROSLAV recording was performed throughout the 25 days of the experiment. P1, P2, P3, and P4 denote the four periods taken for statistical analysis with StatistiSLAV. Black arrows signify days when intracerebroventricular application of STZ or citrate buffer was performed. Red arrows signify the period when behaviour testing was performed. Gray arrows signify days of weighing and bedding change.

Data analysis

Data analysis was performed using the MIROSLAV software suite. Data preparation was performed with Python using Prepare-a-SLAV and TidySLAV. Raw logs were resampled to 1-minute bins using Prepare-a-SLAV. TidySLAV was used to examine the data for technical faults and export it into a tidy, long format.

Circadian activity was analysed using MIROSine to examine the dynamics of 24-hour rhythm amplitude, midline, and phase shift over the course of the experiment. Exploratory data analysis was performed in R using MIRO The Explorer to gain an insight into overall trends and select time periods of particular interest for statistical analysis.

Statistical models were fitted using the R script StatistiSLAV to estimate group MESORs, amplitudes, and phase shifts, as well as respective contrasts between control and STZ treatment groups. Family-wise error rate was controlled using the Holm correction. Alpha was set to 5%.

Full MIRO The Explorer and StatistiSLAV analyses are available in the MIROSLAV analysis repository, which also contains links to live online notebooks [11].

Results

Based on MIRO The Explorer analyses (Fig. 6), periods of interests for further analysis were selected: i) baseline measurements (days 0 and 1); ii) acute rhythm changes within days of streptozotocin application (days 7 and 8); iii) stabilised rhythms one week after STZ application (days 14 and 15); iv) rhythms after behavioural testing (days 21 and 22) (Fig. 5).

MIROSLAV measurements before streptozotocin administration show no difference between animals at baseline (Fig. 7A). Within the two days following model induction, a lower nocturnal activity peak (zenith) is seen in the STZ-icv group (Fig. 7B). One week later, the zenith is reestablished, but the diurnal activity trough (nadir) is higher in the STZ-icv group. STZ-icv-treated animals also exhibit a latency in the zenith onset of approximately 50 minutes at this timepoint (Fig. 7C). During behavioural testing, MIRO The Explorer indicates a higher diurnal activity in the STZ-icv group, as well as a lingering circadian disruption in the following days (Fig. 6). StatistiSLAV analysis on the second and third day after behavioural testing shows a reduction in amplitude, indicating both a lower nocturnal zenith, and a higher diurnal nadir in the STZ-icv group (Fig. 7D).

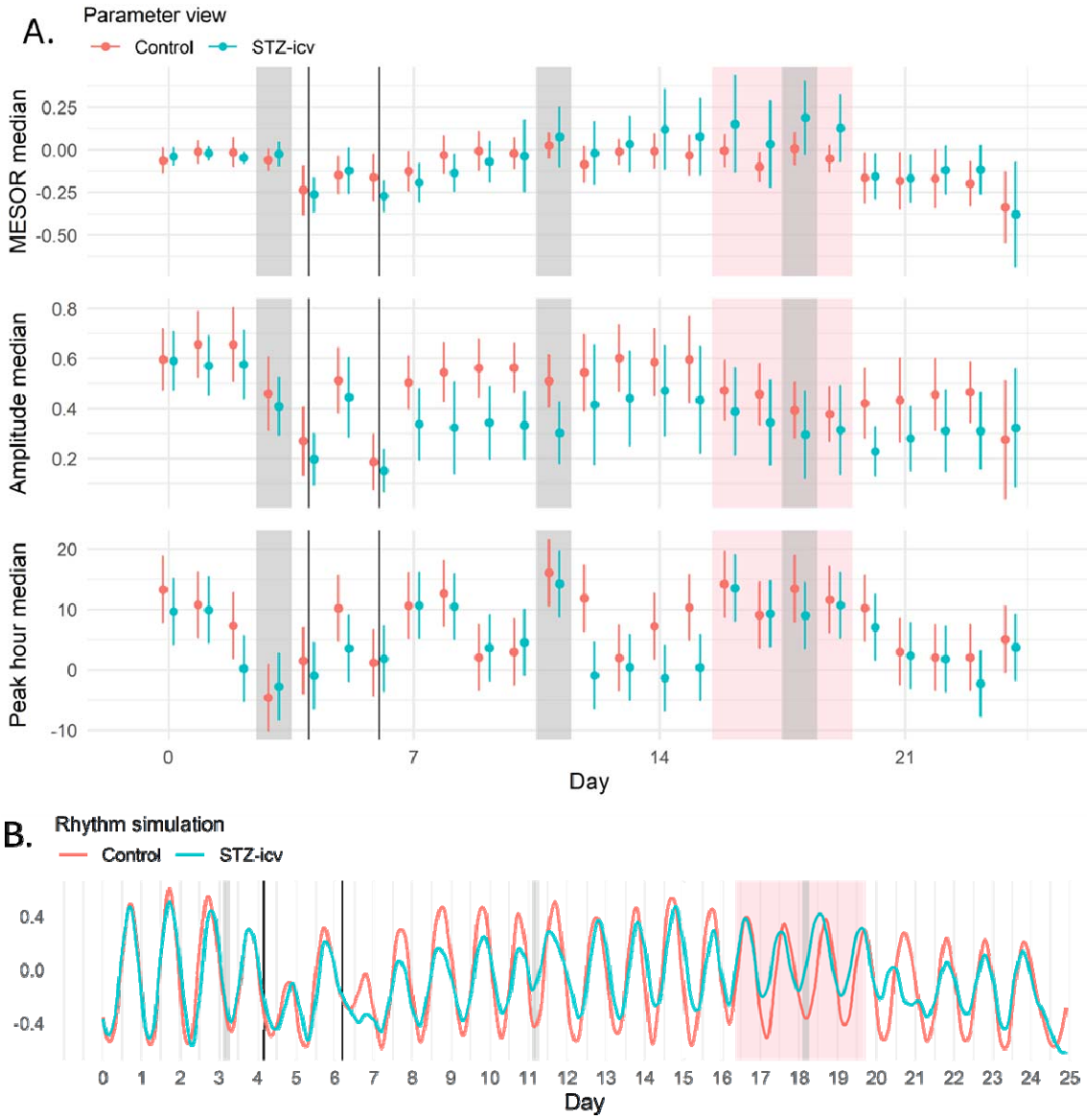


Figure 6. MIRO The Explorer's output plots. Gray bars signify the days/times of animal weighing and cage bedding change. The black vertical lines at days 4 and 6 signify intracerebroventricular STZ or buffer injections. Behavioural testing was performed during the period highlighted in red. **A.** *Parameter view* shows temporal dynamics of the three sinusoid parameters over the course of the experiment. **B.** *Rhythm simulation* reconstructs group-representative activity over the course of the experiment, calculated from group median amplitude, midline, and phase shift parameters for each day. In *Rhythm view*, each day label signifies the start of the corresponding 24-hour period.

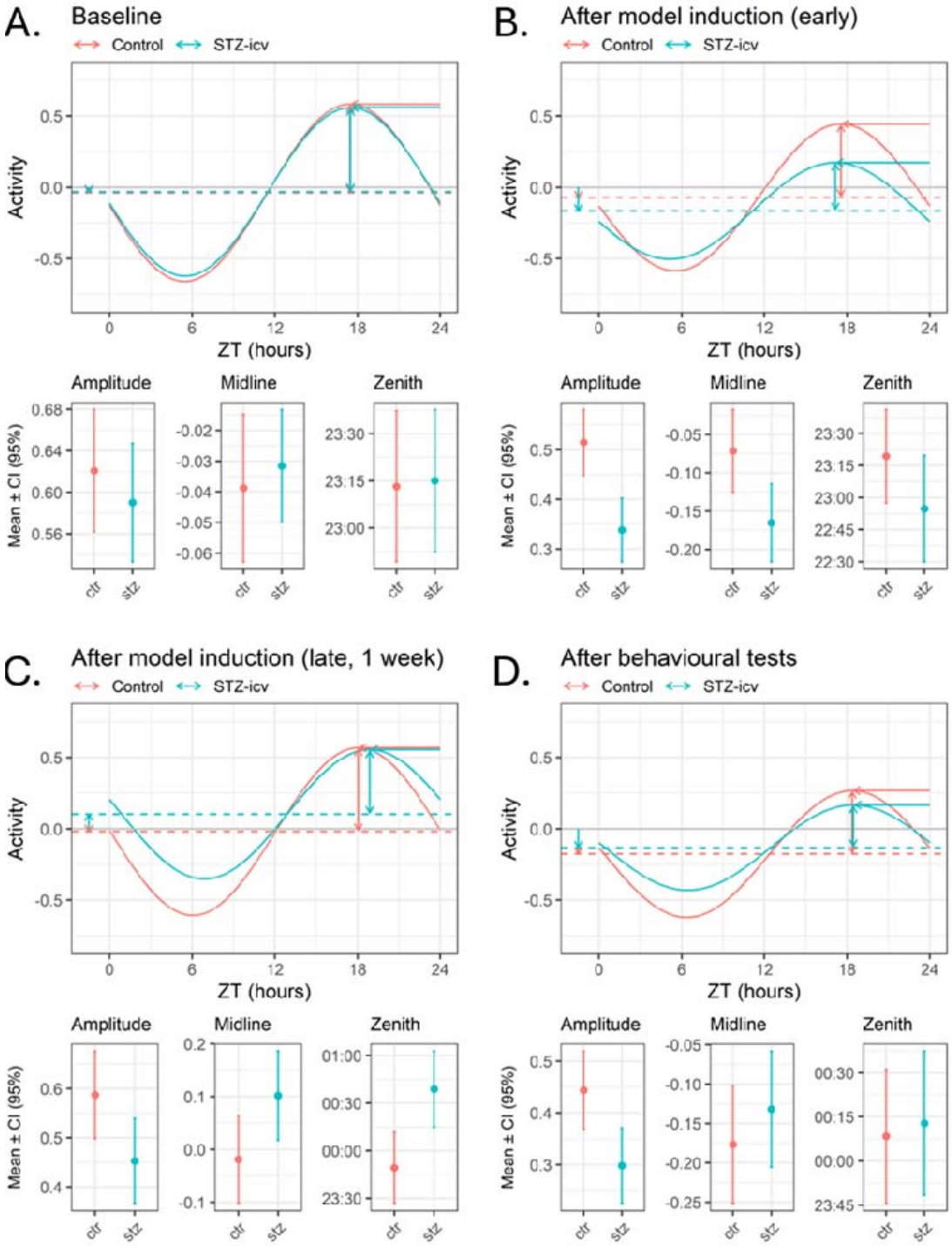


Figure 7. *StatistiSLAV* output. Estimated circadian rhythm parameters (amplitude, midline, and time of day of zenith [peak activity] onset) and reconstructed sinusoidal representations for each treatment group at four timepoints of interest, as described in the text. **A.** At baseline, no differences are observed between groups. **B.** The second and third day after STZ administration, the amplitude and midline are lowered in the STZ-icv group. **C.** One week after model induction, STZ-icv amplitude is reduced, with a trend towards higher midline activity, and a latency to zenith of approximately 50 minutes. **D.** After behavioural tests, only the amplitude seems to be changed, with

a lower value in the STZ-icv group.

Discussion

Due to the general limitations of proprietary, commercial home cage monitoring systems, and by building on pioneering ideas of open source tools such as RAD [6] and COMPASS [7], we have decided to develop MIROSLAV as a modular platform with the goal of supporting as many experimental settings as possible. To achieve the necessary robustness, over the years of prototype development and testing, we have identified common points of failure and implemented this experience in our design to support high-throughput recording of rodent locomotor activity in large studies.

MIROSLAV has shown that its design considerations resulted in a platform with real-word utility by allowing us to non-invasively assess and describe the circadian dysrhythmia in the rat sporadic Alzheimer's disease model induced by intracerebroventricular streptozotocin. We present these results thoroughly processed as part of the proof-of-concept study here, but MIROSLAV's earlier prototypes have shown preliminary findings of circadian dysrhythmia, to the best of our knowledge, for the first time in this model [41], where locomotor disruption has been described in conventional behavioural tests ever since the model's first behavioural phenotyping by Mayer et al. [36]. Following these MIROSLAV results, an extensive behavioural characterisation of the STZ-icv sporadic AD model yielded interesting findings in the context of a link between AD and attention-deficit hyperactivity disorder [42], prompting a new avenue of research. This brief example illustrates further benefits that widely available HCM tools could carry.

What makes MIROSLAV a friend indeed?

Hardware and infrastructure design

Addressed hardware and infrastructural considerations making MIROSLAV appropriate for long-term monitoring at scale are listed in Table 2.

Our experience has shown the system to be robust used as four parallelised MIROSLAV units simultaneously recording activity from 87 cages to two recording computers over the same wireless network, and a third one off-site via a secure Internet connection. Having multiple wirelessly connected recording devices outside of the animal habitats was of great practical value in our experiments – it enabled remote animal and equipment checks (in addition to regular, in-person daily checks), without increasing risk of contamination as no habitat entry was required. Considering the lack of issues in a large-scale deployment and high theoretical limits of operation (described above in the *Infrastructure* section of the MIROSLAV device description), we are confident the MIROSLAV platform could be used in deployments of hundreds of cage units. Finally, of note is the low cost of the hardware system, amounting to under 1,500 EUR for a full package needed to build five MIROSLAVs with a total of 120 cage sensor arrays and a large quantity of spare parts in case of damage. A full bill of materials available in the MIROSLAV hardware repository [9].

Table 2. MIROSLAV hardware design. Common failures discovered in empirical testing, and MIROSLAV design choices made to mitigate or eliminate points of failure.

Point of failure	Design choices	Result
Cage-mounted sensor array: • Disconnecting the sensor array to remove a cage from the rack • Removing the sensor array from the cage lid	• Sensor array connects to the main unit with a zero-force ethernet/telephone-like connector (RJ9), similar to COMPASS [7] • No tools required to fully remove the sensor array from the cage lid	The connector requires grabbing the head and pressing a pin to release the connection, eliminating the need for applying force to the connector and cable-tugging. This approach maintains and prolongs the integrity of the cable over many connect-disconnect cycles.
Partial (spurious readings) or complete failure (data loss) of a sensor unit	Two sensor units in each array, sharing one connection to the main unit.	In case of complete failure, data is still recorded by the second sensor. In case of partial failure, TidySLAV delta heatmaps show a high discrepancy between the two sensors' readings, alerting the user.
Power outage in the habitat	Main data acquisition unit is based around a microcontroller (as opposed to PC, Raspberry Pi)	Lacking a filesystem and operating system, microcontrollers do not suffer from program corruption on power loss, long start-up times, or require reconfiguration. After power is reestablished, data acquisition is resumed within seconds.
Disruptive noise in the habitat, physical damage to the device from habitat conditions	The WiFi-enabled microcontroller stores no data in the habitat, but transmits it via WiFi. Requires WiFi coverage of the habitat area (but not Internet access).	A separate device, placed outside of the habitat, but connected to the same network, accepts MIROSLAV data and records it permanently. In case of physical damage to the MIROSLAV device itself, all components can be immediately replaced without additional configuration, ensuring minimal loss of data.
Failure of a recording device	Wireless data transmission via the MQTT protocol, which allows dynamic connection of recording devices	An MQTT "broker" forwards data to as many on- or off-site recording devices as needed, all of which retain copies of the data. Devices may connect ("subscribe") or disconnect dynamically without disrupting the data acquisition process.
Time synchronisation of sensor readings from multiple operational MIROSLAVs	Network Time Protocol (NTP) synchronisation of all devices, timestamping on message send and receive.	All MIROSLAV and recording devices synchronise their clocks against the same NTP server (which can be deployed on a local network without requiring Internet access), and record the exact date and time when the message was both sent (MIROSLAV) and received (recording device).

Software design and data quality

While the utility of continuously operating HCM devices in virtually all types of experimental designs is indisputable, there are many practical considerations when working with the acquired data. Today, the industry and the scientific community are hard at work developing new techniques handling the immense quantity of data these types of devices produce, in terms of storage, processing, and ensuring data quality [3,43,44].

These concerns apply to MIROSLAV as well – after obtaining raw data, it needs to be analysed to garner biological insight. The MIROSLAV software package was developed to transparently guide the researcher through three principal steps of the data analysis process, as described in Table 3.

Table 3. MIROSLAV software design. A description of the MIROSLAV software toolkit workflow.

Processing step	What does it do?
Prepare-a-SLAV	Processes raw textual logs into a data structure, labels the data, resamples the data to reduce size and facilitate subsequent analyses. Output: A tabular dataframe containing labelled sensor readings.
⇩ **TidySLAV**	Melts Prepare-a-SLAV's table into a tall format, performs standardisation against baseline values, detects periods when the sensors were disconnected, and adds metadata. Output: A tall dataframe with experimental metadata, suitable for plotting and statistical analysis.
⇩ **MIROSine**	Operates on TidySLAV's dataframe, reduces each sensor's readings to daily 24-hour rhythm amplitudes, mean activities (MESORs), and times of peak activity (acrophases). Output: A dataframe containing three parameters for every day of each sensor's recordings, describing the 24-hour rhythm of circadian activity. Additionally, this dataframe is used by **MIRO The Explorer** to generate exploratory plots, and by **StatistiSLAV** to perform statistical comparisons of treated groups.

In the MIROSLAV software suite, we would like to highlight MIROSine as a novel approach to analysing locomotor activity data with a focus on the animals'

circadian rhythm. Without assuming static circadian rhythm parameters over the course of an experiment, the approach allows for: i) intuitive visualisations of circadian rhythm dynamics over time; ii) the application of mixed regression models to the data for statistical analysis, with the inclusion of appropriate covariates, interactions, and random effects. Unlike bootstrapping approaches, it is not computationally intensive, with MIROSine analyses with our example data being successfully completed within minutes on a personal laptop.

MIROSLAV as an open, modular platform

The above considerations led to a reliable method which has so far found extensive use in our laboratory. However, to achieve high configurability as a prerequisite for supporting possible future experimental designs and wider use outside of our own context, MIROSLAV needed to be designed as an open and modular platform [45].

For example, a MIROSLAV unit doesn't need to be used as a home cage monitoring tool. A standalone mainboard can function as a small, Wi-Fi-connected monitor of habitat environmental parameters. Even though one unit supports up to 64 cages, the modular input stack can be reduced to a single board, e.g. to spread out many cages across multiple MIROSLAV units as a means of reducing the number of failure points. This option is also feasible due to MIROSLAV's low cost. Furthermore, to change the scale from a smaller, preliminary experiment to a large-scale one, or vice versa, the power supply is perhaps a less obvious consideration, but one of critical importance. As large 5V power supplies are somewhat difficult to source due to availability and price, the load of a larger number of sensors can be distributed across a larger number of standard, smaller power supplies which are trivial to add. This architecture can be used as a foundation for other novel tools, and we are currently

developing a home cage conditioning apparatus based on the MIROSLAV architecture in our laboratory [46].

From the software side, we recognise that freely available code in a repository is insufficient for replicability and reuse. MIROSLAV analysis code is written as notebooks [47–50] – documents with human-readable, formatted text serving as documentation, with interspersed code cells where configuration parameters can be modified and code can be executed to produce the desired results within the document itself. Unlike a graphical user interface, this approach allows researchers to quickly become intimately familiar with how their data is handled throughout the entire process. As the notebook is mainly composed of formatted text describing the code cells' processes and configuration, a researcher with no knowledge of programming can simply adjust the available parameters to adapt the analysis for their needs. Similarly, the extensive code descriptions required by this approach facilitate contributions by researchers adept at programming, who can submit their code changes to the repository and continue to build up the MIROSLAV platform through a collaborative effort seen in many open source projects, from DeepLabCut [51,52] to the Linux kernel itself [53]. Finally, the analysis code is also modularised according to the steps outlined previously (Table 3). Prepare-a-SLAV, TidySLAV, and the MIROSine tools are completely independent and store a dataframe with a defined format, giving researchers the freedom to seamlessly intercalate other tools at various points in the process.

By designing MIROSLAV to be transparent and highly adaptable at every level, we hope every researcher in need will find a friend indeed in MIROSLAV.

Limitations

The MIROSLAV system has some inherent limitations.

While general benefits of passive infrared sensors include non-invasiveness and low cost, they aren't able to discern individual animals in group-housed experimental designs – any movement is registered, regardless of which animal produced it. Furthermore, the HC-SR501 sensor modules used in the construction of MIROSLAV have individual variability, requiring baseline measurements and subsequent standardisation (performed by TidySLAV) to detect deviations from baseline measurements taken with one sensor after an event of interest has occurred (e.g. application of pharmacotherapy). The latter limitation can be mitigated, allowing direct and immediate comparison of readings from different sensor arrays by fitting the MIROSLAV sensor arrays with factory-calibrated sensor modules (e.g. Panasonic EW AMN32111 used by the COMPASS system [7]). Their unit price is, however, approximately 10x higher than that of the default HC-SR501 modules.

While PIR sensors are suitable for tracking patterns of overall locomotor activity, due to the nature of the technology, only information on the existence and duration of any movement within a cage can be recorded. To accurately assess the nature of the movement, combining the MIROSLAV system with other methods (e.g. video recording) is advised. The main unit is fitted with unused connectors which can be used to precisely synchronise MIROSLAV devices with other recording devices using hardware synchronisation/triggering platforms such as the open-source Behaviour-recording Device Synchroniser (BeDSy) [54].

Furthermore, while all parts of a MIROSLAV device can withstand 70% ethanol or isopropanol and gentle scrubbing, it was not designed to be sterilised with more aggressive solvents or high heat. Although not yet attempted, we expect vaporised hydrogen peroxide sterilisation can be employed without damaging the device [55].

Future development

As part of further research and development, to increase the utility of MIROSLAV, the feasibility of overcoming the aforementioned limitations should be assessed, and improvements implemented. The software package will be updated to utilise the onboard environmental sensors capable of monitoring habitat humidity, temperature, illumination, and personnel movement. Work on the documentation will persist as a continuous effort to make MIROSLAV even more approachable by potential users.

Conclusion

MIROSLAV is a highly modular open-source hardware and software platform for monitoring home cage activity and habitat conditions, capable of non-invasive, high-throughput operation over extended periods in many experimental settings. The comprehensive, extensively documented software toolkit gives the researcher full insight into every step of the data's journey, from acquisition to statistical analysis, as an advance from proprietary, commercial tools. It is our hope that this adaptable, transparent, and low-cost solution will help many laboratories maximise the data output of their ongoing research projects and entice inter-group collaboration for further development of the open-source ecosystem of tools made by researchers, for researchers. In this spirit, readers are encouraged to contact the authors should they want to build, adapt, or contribute to the MIROSLAV platform.

Data and code availability statement

All MIROSLAV data and code are placed in three GitHub repositories, *MIROSLAV-hardware*, *MIROSLAV-firmware*, and *MIROSLAV-analysis*. With each new release, a snapshot of the entire repository is taken and stored by Zenodo, so these repositories are referenced by their Zenodo DOIs throughout the main text.

MIROSLAV hardware

Repository URL: https://github.com/davorvr/MIROSLAV-hardware

Repository contents: Contains KiCad 8.0 printed circuit board (PCB) schematics and Gerber files.

MIROSLAV firmware

Repository URL: https://github.com/davorvr/MIROSLAV-firmware

Repository contents: Arduino firmware for the MIROSLAV device (MIROSLAVino), as well as data acquisition software (Record-a-SLAV) for PC recording devices.

- MIROSLAV Arduino firmware (MIROSLAVino): https://github.com/davorvr/MIROSLAV-firmware/tree/main/miroslavino
- Data acquisition tool (Record-a-SLAV): https://github.com/davorvr/MIROSLAV-firmware/tree/main/Record-a-SLAV

MIROSLAV analysis

Repository contents: Contains raw data for the PoC study, sample environmental data, and the MIROSLAV analysis toolkit. MIROSLAV home cage data analysis tools are provided in the main directory of the repository as:

i) Pure Python or R scripts (file extension: *.py* or *.R*)

ii) Interactive Jupyter Notebooks (file extension: *.ipynb*)

iii) Live online Google Colab notebooks (URLs in the repository README file: https://github.com/davorvr/MIROSLAV-analysis/blob/main/README.md).

Environmental data parsing and plotting tools are included in a subdirectory.

The repository also contains intermediate and final analysis outputs of the PoC study data from all of the listed tools.

Repository URL: https://github.com/davorvr/MIROSLAV-analysis

Components list:

- Raw data: https://github.com/davorvr/MIROSLAV-analysis/tree/main/0_raw_logs
- Data analysis tools (https://github.com/davorvr/MIROSLAV-analysis):
    - Prepare-a-SLAV (data preparation)
    - TidySLAV (data cleaning)
    - MIROSine (24-hour rhythm analysis)
    - MIROSine-based analyses:
        - MIRO The Explorer (exploratory data analysis)
        - StatistiSLAV (statistical data analysis)
- EnviroSLAV (environmental monitoring parsing and plotting tools) https://github.com/davorvr/MIROSLAV-analysis/tree/main/E_EnviroSLAV

Authors' contributions

DV, **JH**, and **IK** conceptualised the MIROSLAV platform. **DV** developed prototype and final iterations of MIROSLAV electronics, firmware, and infrastructure. **JH** designed and built three iterations of MIROSLAV prototype units. **AMV** designed the final MIROSLAV device, and built it with **DV**, **JH**, **IK**, and **ABP**.

All prototype analysis code was written by **IK**, upon which **DV** wrote the *mirofile* Python library, Prepare-a-SLAV, and TidySLAV. Exploratory visualisations were conceptualised by **IK** and developed into TidySLAV diagnostic plots and MIRO The

Explorer by **DV**, **JH**, **PM**, and **MŠP**. **DV** conceptualised the MIROSine analysis approach and developed it with **JH** and **PŠM**, with critical input from **MC** and **VT**. **DV**, **JH**, and **VT** conceptualised the StatistiSLAV analysis with critical input from **MC**. The MIROSLAV logo used in the repositories was drawn by **PŠM**.

In the PoC experiment, **DV**, **JH**, **ABP**, **AK**, and **JOB** carried out the drug treatments, and **JH** performed the behavioural testing.

DV wrote the manuscript. **JH**, **IK**, **AMV**, **ABP**, **PM**, **PŠM**, **AK**, **JOB**, **MC**, **VT**, and **MŠP** commented on the manuscript and provided critical feedback. **MŠP**, mentor of DV and the laboratory's PI, provided funding and supervised the project.

Funding source

This work was funded by the Croatian Science Foundation research project IP-2018-01-8938 ("Mechanisms of nutrient-mediated effects of endogenous glucagon-like peptide-1 on cognitive and metabolic alterations in experimental models of neurodegenerative disorders"), and Young Researchers' Career Development Projects DOK-2018-09-6526 (PhD student: Jan Homolak, graduated) and DOK-2021-02-6419 (PhD student: Davor Virag). The research was co-financed by the Scientific Centre of Excellence for Basic, Clinical and Translational Neuroscience (project "Experimental and clinical research of hypoxic-ischemic damage in perinatal and adult brain"; GA KK01.1.1.01.0007 funded by the European Union through the European Regional Development Fund).

Acknowledgements

The authors would like to thank Pia Kahnau and Lars Lewejohann for their valuable feedback on the idea that was subsequently developed into the MIROSine analysis approach during Davor Virag's visit to their laboratory (Short-Term Scientific Mission funded by the COST-TEATIME [CA20135] project).

The authors would also like to thank laboratory technician Božica Hržan for supervising MIROSLAV devices and ensuring their continuous operation throughout all MIROSLAV experiments, as well as animal caretakers Miroslava Jerković, Tomislav Petanjek, and Anita Šantak for their patience while handling MIROSLAV-equipped cages.

Figures 1 and 4 were created with BioRender.com.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] M. Merrow, Circadian clocks: It's time for chronobiology, PLoS Biol. 21 (2023) e3002426. https://doi.org/10.1371/journal.pbio.3002426.

[2] J. Homolak, M. Mudrovčić, B. Vukić, K. Toljan, Circadian Rhythm and Alzheimer's Disease, Med. Sci. Basel Switz. 6 (2018) 52. https://doi.org/10.3390/medsci6030052.

[3] S.W. Baran, N. Bratcher, J. Dennis, S. Gaburro, E.M. Karlsson, S. Maguire, P. Makidon, L.P.J.J. Noldus, Y. Potier, G. Rosati, M. Ruiter, L. Schaevitz, P. Sweeney, M.R. LaFollette, Emerging Role of Translational Digital Biomarkers Within Home Cage Monitoring Technologies in Preclinical Drug Discovery and Development, Front. Behav. Neurosci. 15 (2021) 758274. https://doi.org/10.3389/fnbeh.2021.758274.

[4] P. Kahnau, P. Mieske, J. Wilzopolski, O. Kalliokoski, S. Mandillo, S.M. Hölter, V. Voikar, A. Amfim, S. Badurek, A. Bartelik, A. Caruso, M. Čater, E. Ey, E. Golini, A. Jaap, D. Hrncic, A. Kiryk, B. Lang, N. Loncarevic-Vasiljkovic, H. Meziane, A. Radzevičienė, M. Rivalan, M.L. Scattoni, N. Torquet, J. Trifkovic, B. Ulfhake, C. Thöne-Reineke, K. Diederich, L. Lewejohann, K. Hohlbaum, A systematic review of the

development and application of home cage monitoring in laboratory mice and rats, BMC Biol. 21 (2023) 256. https://doi.org/10.1186/s12915-023-01751-7.
[5] M. Oellermann, J.W. Jolles, D. Ortiz, R. Seabra, T. Wenzel, H. Wilson, R.L. Tanner, Open Hardware in Science: The Benefits of Open Electronics, Integr. Comp. Biol. 62 (2022) 1061–1075. https://doi.org/10.1093/icb/icac043.
[6] B.A. Matikainen-Ankney, M. Garmendia-Cedillos, M. Ali, J. Krynitsky, G. Salem, N.L. Miyazaki, T. Pohida, A.V. Kravitz, Rodent Activity Detector (RAD), an Open Source Device for Measuring Activity in Rodent Home Cages, eNeuro 6 (2019) ENEURO.0160-19.2019. https://doi.org/10.1523/ENEURO.0160-19.2019.
[7] L.A. Brown, S. Hasan, R.G. Foster, S.N. Peirson, COMPASS: Continuous Open Mouse Phenotyping of Activity and Sleep Status, Wellcome Open Res. 1 (2016) 2. https://doi.org/10.12688/wellcomeopenres.9892.2.
[8] KiCad Development Team, KiCad: An open source software suite for Electronic Design Automation (EDA), n.d. https://www.kicad.org.
[9] D. Virag, MIROSLAV hardware, (2024). https://doi.org/10.5281/ZENODO.11659494.
[10] D. Virag, MIROSLAV firmware, (2024). https://doi.org/10.5281/ZENODO.12191593.
[11] D. Virag, MIROSLAV analysis, (2024). https://doi.org/10.5281/ZENODO.12191589.
[12] Adafruit Industries, BISS0001 Micro Power PIR Motion Detector IC Technical Description, (n.d.). http://www.ladyada.net/media/sensors/BISS0001.pdf (accessed May 29, 2024).
[13] A. Banks, E. Briggs, K. Borgendale, R. Gupta, MQTT Version 5.0, (2019). https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.
[14] B. Mishra, B. Mishra, A. Kertesz, Stress-Testing MQTT Brokers: A Comparative Analysis of Performance Measurements, Energies 14 (2021) 5817. https://doi.org/10.3390/en14185817.
[15] GNU General Public License, (2007). http://www.gnu.org/licenses/gpl.html.
[16] Arduino core for the ESP32, ESP32-S2, ESP32-S3, ESP32-C3, ESP32-C6 and ESP32-H2, (2024). https://github.com/espressif/arduino-esp32 (accessed May 29, 2024).
[17] H. Wickham, Tidy Data, J. Stat. Softw. 59 (2014). https://doi.org/10.18637/jss.v059.i10.
[18] M. Sachs, cosinor: Tools for Estimating and Predicting the Cosinor Model, 2023. https://CRAN.R-project.org/package=cosinor.
[19] T.F.S. Bastiaanssen, S.-J. Leigh, G.S.S. Tofani, C.E. Gheorghe, G. Clarke, J.F. Cryan, Kronos: A computational tool to facilitate biological rhythmicity analysis, bioRxiv [preprint] (2023). https://doi.org/10.1101/2023.04.21.537503.
[20] R. Parsons, O. Jayasinghe, N. White, O. Rawashdeh, GLMMcosinor: Fit a Cosinor Model Using a Generalised Mixed Modelling Framework, 2024. https://CRAN.R-project.org/package=GLMMcosinor.
[21] T. Preston-Werner, P. Gedam, et al., TOML v1.0.0: Tom's Obvious, Minimal Language, (2021). https://toml.io/en/v1.0.0 (accessed May 30, 2024).
[22] G. Cornelissen, Cosinor-based rhythmometry, Theor. Biol. Med. Model. 11 (2014) 16. https://doi.org/10.1186/1742-4682-11-16.
[23] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2024. https://www.R-project.org/.
[24] H. Wickham, ggplot2: Elegant Graphics for Data Analysis, Springer-Verlag New York, 2016. https://ggplot2.tidyverse.org.
[25] M.E. Brooks, K. Kristensen, K.J. van Benthem, A. Magnusson, C.W. Berg, A. Nielsen, H.J. Skaug, M. Maechler, B.M. Bolker, glmmTMB Balances Speed and Flexibility Among Packages for Zero-inflated Generalized Linear Mixed Modeling, R J. 9 (2017) 378–400. https://doi.org/10.32614/RJ-2017-066.
[26] F. Hartig, DHARMa: Residual Diagnostics for Hierarchical (Multi-Level / Mixed) Regression Models, 2022. https://CRAN.R-project.org/package=DHARMa.
[27] R.V. Lenth, emmeans: Estimated Marginal Means, aka Least-Squares Means, 2024. https://CRAN.R-project.org/package=emmeans.

[28] M. Santon, F. Korner-Nievergelt, N.K. Michiels, N. Anthes, A versatile workflow for linear modelling in R, Front. Ecol. Evol. 11 (2023). https://doi.org/10.3389/fevo.2023.1065273.
[29] R. Hou, L.E. Tomalin, M. Suárez-Fariñas, cosinoRmixedeffects: an R package for mixed-effects cosinor models, BMC Bioinformatics 22 (2021) 553. https://doi.org/10.1186/s12859-021-04463-3.
[30] N. Percie du Sert, V. Hurst, A. Ahluwalia, S. Alam, M.T. Avey, M. Baker, W.J. Browne, A. Clark, I.C. Cuthill, U. Dirnagl, M. Emerson, P. Garner, S.T. Holgate, D.W. Howells, N.A. Karp, S.E. Lazic, K. Lidster, C.J. MacCallum, M. Macleod, E.J. Pearl, O.H. Petersen, F. Rawle, P. Reynolds, K. Rooney, E.S. Sena, S.D. Silberberg, T. Steckler, H. Würbel, The ARRIVE guidelines 2.0: Updated guidelines for reporting animal research, PLOS Biol. 18 (2020) e3000410. https://doi.org/10.1371/journal.pbio.3000410.
[31] W.M.S. Russell, R.L. Burch, The principles of humane experimental technique, Special ed., [Nachdr. der Ausg.] London 1959, Universities Federation for Animal Welfare, Potters Bar, Herts, 1992.
[32] J. Homolak, J. De Busscher, M. Zambrano-Lucio, M. Joja, D. Virag, A. Babic Perhoc, A. Knezovic, J. Osmanovic Barilar, M. Salkovic-Petrisic, Altered Secretion, Constitution, and Functional Properties of the Gastrointestinal Mucus in a Rat Model of Sporadic Alzheimer's Disease, ACS Chem. Neurosci. 14 (2023) 2667–2682. https://doi.org/10.1021/acschemneuro.3c00223.
[33] M. Salkovic-Petrisic, A.B. Perhoc, J. Homolak, A. Knezovic, J. Osmanovic Barilar, P. Riederer, Experimental Approach to Alzheimer's Disease with Emphasis on Insulin Resistance in the Brain, in: R.M. Kostrzewa (Ed.), Handb. Neurotox., Springer International Publishing, Cham, 2021: pp. 1–52. https://doi.org/10.1007/978-3-030-71519-9_98-1.
[34] J. Homolak, A.B. Perhoc, A. Knezovic, J. Osmanovic Barilar, M. Salkovic-Petrisic, Additional methodological considerations regarding optimization of the dose of intracerebroventricular streptozotocin A response to: "Optimization of intracerebroventricular streptozotocin dose for the induction of neuroinflammation and memory impairments in rats" by Ghosh et al., Metab Brain Dis 2020 July 21, Metab. Brain Dis. 36 (2021) 97–102. https://doi.org/10.1007/s11011-020-00637-9.
[35] Z. Lacković, M. Šalković, Streptozotocin and alloxan produce alterations in rat brain monoamines independently of pancreatic beta cells destruction, Life Sci. 46 (1990) 49–54. https://doi.org/10.1016/0024-3205(90)90056-w.
[36] G. Mayer, R. Nitsch, S. Hoyer, Effects of changes in peripheral and cerebral glucose metabolism on locomotor activity, learning and memory in adult male rats, Brain Res. 532 (1990) 95–100. https://doi.org/10.1016/0006-8993(90)91747-5.
[37] E.P. Noble, R.J. Wurtman, J. Axelrod, A simple and rapid method for injecting H3-norepinephrine into the lateral ventricle of the rat brain, Life Sci. 6 (1967) 281–291. https://doi.org/10.1016/0024-3205(67)90157-9.
[38] A. Knezovic, A. Loncar, J. Homolak, U. Smailovic, J. Osmanovic Barilar, L. Ganoci, N. Bozina, P. Riederer, M. Salkovic-Petrisic, Rat brain glucose transporter-2, insulin receptor and glial expression are acute targets of intracerebroventricular streptozotocin: risk factors for sporadic Alzheimer's disease?, J. Neural Transm. Vienna Austria 1996 124 (2017) 695–708. https://doi.org/10.1007/s00702-017-1727-6.
[39] A. Knezovic, M. Piknjac, J. Osmanovic Barilar, A. Babic Perhoc, D. Virag, J. Homolak, M. Salkovic-Petrisic, Association of Cognitive Deficit with Glutamate and Insulin Signaling in a Rat Model of Parkinson's Disease, Biomedicines 11 (2023) 683. https://doi.org/10.3390/biomedicines11030683.
[40] D. Virag, J. Homolak, I. Kodvanj, A. Babic Perhoc, A. Knezovic, J. Osmanovic Barilar, M. Salkovic-Petrisic, Repurposing a digital kitchen scale for neuroscience research: a complete hardware and software cookbook for PASTA, Sci. Rep. 11 (2021) 2963. https://doi.org/10.1038/s41598-021-82710-6.

[41] D. Virag, Promjene cirkadijanog ritma u štakorskom modelu sporadične Alzheimerove bolesti, MD Thesis, University of Zagreb, School of Medicine, Department of Pharmacology, 2021. https://urn.nsk.hr/urn:nbn:hr:105:442725 (accessed November 7, 2022).
[42] E. Grünblatt, J. Homolak, A. Babic Perhoc, V. Davor, A. Knezovic, J. Osmanovic Barilar, P. Riederer, S. Walitza, C. Tackenberg, M. Salkovic-Petrisic, From attention-deficit hyperactivity disorder to sporadic Alzheimer's disease-Wnt/mTOR pathways hypothesis, Front. Neurosci. 17 (2023) 1104985. https://doi.org/10.3389/fnins.2023.1104985.
[43] S. Fuochi, M. Rigamonti, E.C. O'Connor, P. De Girolamo, L. D'Angelo, Big data and its impact on the 3Rs: a home cage monitoring oriented review, Front. Big Data 7 (2024). https://doi.org/10.3389/fdata.2024.1390467.
[44] V. D'Argenio, The High-Throughput Analyses Era: Are We Ready for the Data Struggle?, High-Throughput 7 (2018) 8. https://doi.org/10.3390/ht7010008.
[45] K. Gavras, V. Kostakis, Mapping the types of modularity in open-source hardware, Des. Sci. 7 (2021) e13. https://doi.org/10.1017/dsj.2021.11.
[46] D. Virag, J. Homolak, A.B. Perhoč, A. Knezovic, J. Osmanovic, M. Salkovic-Petrisic, Assessing the cognitive effects of methylphenidate in a rat model of sporadic Alzheimer's disease using a home-cage-based approach, in: Unpublished, 2024. https://doi.org/10.13140/RG.2.2.20651.66082.
[47] A. Paxton, How to use Jupyter notebooks and R markdown to create living documents and reproducible reports., (2023). https://github.com/a-paxton/living-documents (accessed May 29, 2024).
[48] E.R. Ivimey-Cook, J.L. Pick, K.R. Bairos-Novak, A. Culina, E. Gould, M. Grainger, B.M. Marshall, D. Moreau, M. Paquet, R. Royauté, A. Sánchez-Tójar, I. Silva, S.M. Windecker, Implementing code review in the scientific workflow: Insights from ecology and evolutionary biology, J. Evol. Biol. 36 (2023) 1347–1356. https://doi.org/10.1111/jeb.14230.
[49] J.F. Pimentel, L. Murta, V. Braganholo, J. Freire, Understanding and improving the quality and reproducibility of Jupyter notebooks, Empir. Softw. Eng. 26 (2021) 65. https://doi.org/10.1007/s10664-021-09961-9.
[50] K.L. Grayson, A.K. Hilliker, J.R. Wares, R Markdown as a dynamic interface for teaching: Modules from math and biology classrooms, Math. Biosci. 349 (2022) 108844. https://doi.org/10.1016/j.mbs.2022.108844.
[51] A. Mathis, P. Mamidanna, K.M. Cury, T. Abe, V.N. Murthy, M.W. Mathis, M. Bethge, DeepLabCut: markerless pose estimation of user-defined body parts with deep learning, Nat. Neurosci. 21 (2018) 1281–1289. https://doi.org/10.1038/s41593-018-0209-y.
[52] DeepLabCut, (2024). https://github.com/DeepLabCut/DeepLabCut (accessed May 29, 2024).
[53] J.Y. Moon, L. Sproull, Essence of distributed work: The case of the Linux kernel, First Monday (2000). https://doi.org/10.5210/fm.v5i11.801.
[54] D. Virag, Behaviour-recording Device Synchroniser, (2024). https://github.com/davorvr/bedsy (accessed May 29, 2024).
[55] S.M. Loo, J. Kiepert, D. Klein, M. Pook, Evaluation of the Effects of Hydrogen Peroxide on Common Aircraft Electrical Materials, 2010. https://apps.dtic.mil/sti/pdfs/ADA518975.pdf (accessed June 11, 2024).