Journal of open hardware

Template of Hardware Metapaper

For submission to the *Journal of Open Hardware*

(1) Overview

Title

Open Sync Board – An open-source tool for hardware synchronization of multimodal physiological sensor setups

Authors

1. Pitzal, Mareike*;
2. Mücke, Annika*;
3. Bräunig, Johanna;
4. Brückner, Stefan;
5. Richer, Robert;
6. Schlechtweg, Niklas;
7. Albrecht, Nils C.;
8. Kölpin, Alexander;
9. Eskofier, Björn M.;
10. Vossiek, Martin

**Both authors contributed equally to this work.*

1. Institute of Microwaves and Photonics, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
2. Machine Learning and Data Analytics Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
3. Institute of Microwaves and Photonics, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
4. Institute of Microwaves and Photonics, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
5. Machine Learning and Data Analytics Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
6. Institute of Microwaves and Photonics, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
7. Institute of High-Frequency Technology, Hamburg University of Technology, Hamburg, Germany
8. Institute of Microwaves and Photonics, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
9. Machine Learning and Data Analytics Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany; Translational Digital Health Group, Institute of AI for Health, Helmholtz Zentrum München - German Research Center for Environmental Health, Neuherberg, Germany

10. Institute of Microwaves and Photonics, Friedrich-Alexander-Universität
Erlangen-Nürnberg (FAU), Erlangen, Germany

Abstract

The fusion of multimodal sensor data gains increasing importance in medical and psychological research. In particular, the analysis of multimodal data contributes to a broader understanding of human well-being. When experimenting with different sensing modalities, precise synchronization of recorded data is a crucial problem that needs to be addressed. We have developed a hardware-based solution capable of synchronizing up to seven arbitrary measurement modalities. These modalities can be mounted to a PCB with various connectors to be either triggered synchronously or receive simultaneous synchronization signals. In the latter case, data can be aligned after the measurement. All PCB outputs are galvanically decoupled to avoid interference and ensure safe usage in human-centered research. In this project, we put particular emphasis on flexibility, high usability, and accuracy. Thus, different synchronization signal options are provided to suit short- as well as long-term measurements. Settings can be configured easily with an accompanying graphical user interface. Additionally, we ensured a high precision of the output signals is ensured with an estimated maximum delay of 125 µs between the output ports. This enables sub-sample synchronization accuracy for sensors with up to 8 kHz sampling rate.

Metadata Overview

Main design files: https://github.com/empkins/open-sync-board
Target group: Biomedical engineers, physicians, and psychologists in interdisciplinary research projects
Skills required: PCB soldering, microcontroller flashing, basic Python programming for data postprocessing
Replication: The Open Sync Board is being used across multiple research projects of the authors and their collaboration partners. Thus, it has been rebuilt from the documentation more than ten times.

See section "Build Details" for more detail.

Keywords

synchronization; open source; multimodal sensing; vital sign monitoring

Introduction

The study of human behavior is a research interest in many fields, including medicine, psychology, neuroscience, and sports science. Many different sensing techniques, such as physiological sensors, motion capture systems, or video cameras, are utilized to assess this. Since the observation of human behavior is a challenging task, the use of a single sensing modality is seldom sufficient [1]. Consequently, such multimodal sensor setups are becoming increasingly popular as they allow for the collection of more comprehensive data. The opportunities provided by these multimodal datasets are

promising, ranging from improved disease prevention, detection, and understanding [2, 3] to validating new sensor technologies [4].

The evaluation of data simultaneously recorded from multiple sensors is a recurring problem in research practice. Especially for fast sensors that operate at higher sampling rates in the millisecond range and below, precise synchronization is crucial. However, it is often necessary to compare open-source systems or self-built prototypes with commercial systems that do not provide an integrated interface for synchronization. Therefore, it is necessary to find customized solutions for synchronization. To the best of our knowledge, there is no universal, manufacturer-independent and readily available commercial solution for synchronizing multimodal sensor setups in human-centered research.

The issue of synchronizing multimodal sensor setups has been addressed by previous work in multiple ways. For instance, a straightforward method to synchronize experimental devices is manual synchronization by identification of events in both signals [4, 5, 6]. However, this approach is very time-consuming, as it requires to (manually) match the events in the different data streams. Consequently, the synchronization precision heavily depends on the accuracy of the event matching.

A more automated approach is timestamp-based synchronization. On the one hand, this can be implemented by connecting all sensors to the same reference device physically or via wireless networks [7]. In this case, the timestamps of the reference device serve as ground truth for all recordings. On the other hand, some studies assume that the clocks of various devices used for their recordings are sufficiently synchronous [8, 9]. Thus, the data is aligned based on the underlying timestamps from each recording device under the assumption that their differences are negligible. However, this approach can be imprecise due to clock delay [10].

Another open-source approach to synchronizing data is directly at recording time via lab streaming layer (LSL)[1]. LSL allows synchronous capturing and storing of data streams during data collection via standardized protocols and has been widely used in various multimodal human-centered studies [11, 12, 13]. However, LSL requires all devices to have accessible APIs and to provide reliable, consistent streaming of data packages.

An alternative approach is sensor synchronization on hardware level. This approach avoids the above-mentioned disadvantages, as it works automatically and minimizes delays to be in the millisecond range and below. The underlying idea is that all sensors record the same synchronization signal as an additional data stream. In a post-processing step, all data streams and their time axes are then stretched and shifted to obtain aligned synchronization signals, and thus also synchronized measurement data. For instance, this method can utilize a micro-controller to send a (periodic) synchronization signal to all connected devices to align their recordings during postprocessing [11]. Previous studies have also utilized pseudo-random sequences as synchronization signals [14, 15]. The delay between two recorded data streams is then calculated by cross-correlating the captured pseudo-random signals and extracting the peak time of the resulting signal. This synchronization method yields pleasing results also for longer measurement durations. However, to employ hardware-based synchronization, depending on the requirements of a specific study, such as measurement duration, utilized sensors and their connectors. Currently, researchers must build custom solutions for every specific study setup, which involves a considerable resource effort often at the expense of usability.

We attempt to solve these issues by developing a multi-functional, customizable, and easily adaptable synchronization solution. Our Open Sync Board (OSB) is based on a printed circuit board (PCB), powered by a microcontroller that can generate different hardware synchronization signals that are distributed to each device attached to the board while assuring galvanic isolation. Thus, it allows synchronizing multiple sensors with an accuracy of less than 1ms, which is precise enough for typical physiological measurement scenarios while assuring galvanic isolation. Designed with modularity in

[1] https://labstreaminglayer.org

mind, the OSB provides different connectors that can be removed and added according to the sensor type. For optimal usability, it can be configured with a graphical user interface (GUI) provided as a standalone executable. Thus, it can be used without extensive technical knowledge and employed in a wide range of medical or psychological studies. Two exemplary use cases of the OSB are displayed in Figure 1.

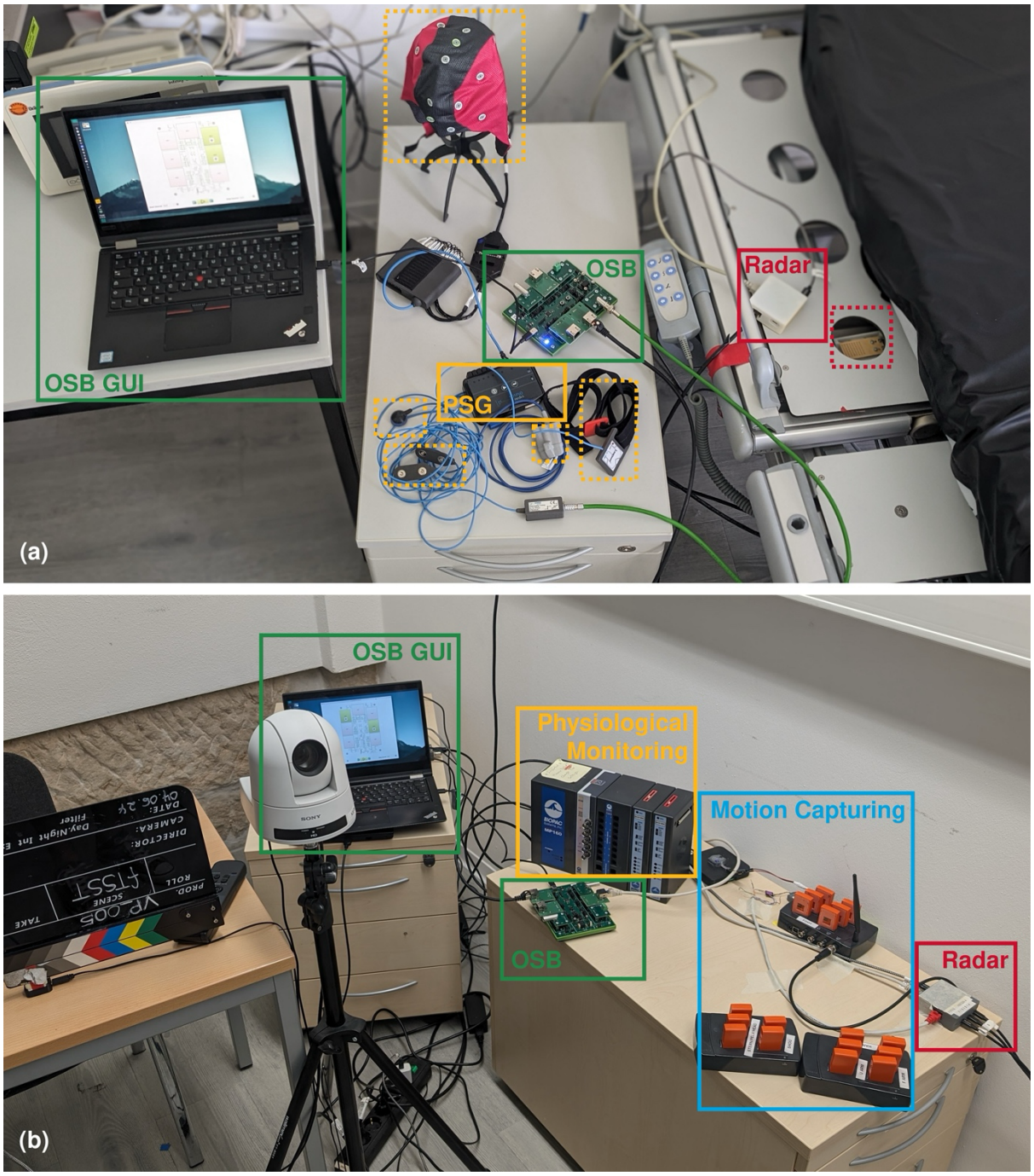


Figure 1: OSB in use.
(a) Use Case 1: The OSB (outlined in green) is used to synchronize recordings from a polysomnography (PSG) system with custom continuous-wave (CW) radar nodes to investigate contactless sleep monitoring. The PSG (solid yellow box) has sensors for monitoring various physiological signals (dotted yellow boxes). The radar node (red box) receives data from antennas mounted underneath the bed (dotted red box).
(b) Use Case 2: The OSB (outlined in green) is used to analyze the influence of acute psychosocial stress on human physiology and behavior. Thus, various physiological signals are measured simultaneously with a gold-standard device (yellow box) and a custom CW radar node (red box). To assess human behavior, individuals are

concurrently equipped with a full-body motion capture system consisting of several inertial measurement unit (IMU) sensor nodes (blue box). All sensor systems are synchronized with the OSB. In both scenarios, the OSB is operated via the OSB GUI on the connected laptop.

Overall Implementation and Design

System Overview

The OSB offers synchronization capabilities for a multi-modal sensor setup with up to seven measurement devices (MDs), as schematically illustrated in Figure 2.
We designed the OSB to ensure maximum flexibility, usability, and accuracy. Thus, the only requirement for a measurement device to be used with our OSB is that it offers an available input channel that can be used to record the synchronization signal generated by the OSB. As outlined in the Section *Application*, this requirement is provided by a multitude of sensors commonly used in physiological measurements. To allow as many different MDs with different synchronization inputs as possible to be connected to the OSB, we provide different daughterboards with standard connectors that serve as adapter inter- faces between the MDs and the OSB. Furthermore, the OSB incorporates ports for receiving input signals to, for example, provide users the ability to record timestamps that log certain events during the measurements.

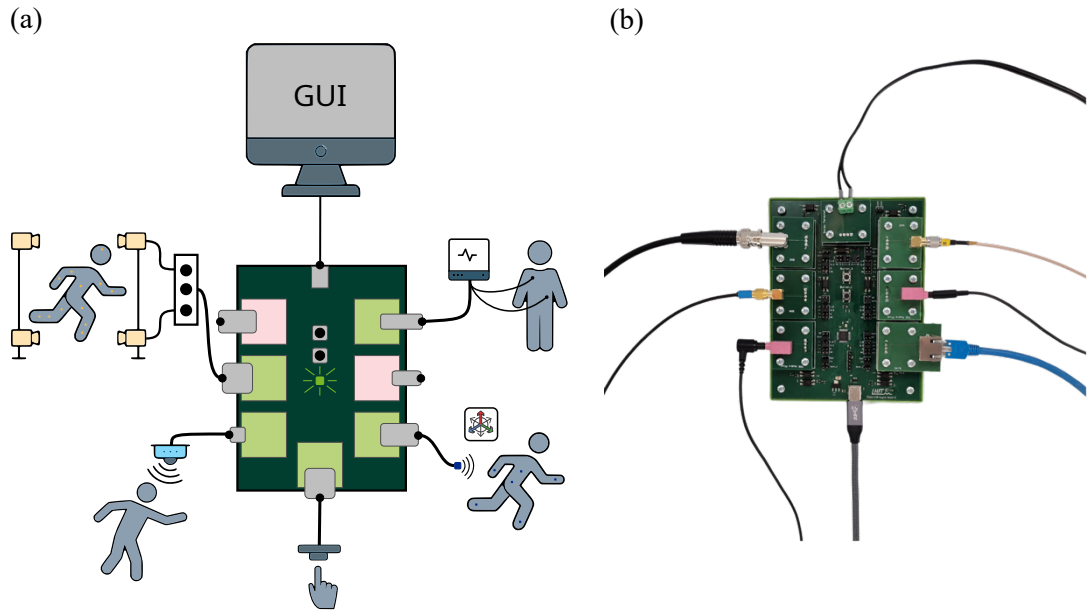


Figure 2: (a) Schematic overview of multimodal measuring possibilities using the OSB and (b) photographic illustration of the assembled OSB with all supported connectors: clockwise starting from the top: Screw terminal, SMA, TRS Audio Plug, RJ45, USB-C (Power supply), TRS Audio Plug, SMA, BNC.

A synchronized recording session can be started and stopped in the GUI, by pressing a hardware button on the OSB, or by detecting a toggled edge in an input signal transmitted to the OSB. A central light emitting diode (LED) on the OSB provides real-time feedback on the measurement status. Once a measurement is started, the LED transitions from an inactive to a blue state, while a red light indicates the occurrence of an error. The OSB and the daughterboards, which are plugged into the motherboard and fastened with screws, are galvanically separated using optocouplers to ensure that the board and the MDs are decoupled to fulfill the IEC60601-1 norm for safe usage in medical research [16]. As different sensor systems may require different signal levels for synchronization, the OSB provides adjustable values for the out-

put voltage. Additionally, the waveform of the synchronization signals can be adjusted for each output individually to generate trigger or clock signals, which can be level- or edge-controlled. Furthermore, users have the option to generate pseudo-random M-sequences, which are particularly beneficial for long-term measurements since M-sequences can be used to determine and compensate sensor clock drifts [17].

To increase usability, the OSB can be controlled with a GUI to assist researchers in operating the OSB efficiently. For data postprocessing after a measurement, we provide a Python module to facilitate the alignment of OSB-recorded data.

PCB Design

A comprehensive overview of the hardware implementation is shown in Figure 3. Additionally, Figure 4 shows the placement of all PCB components onto the OSB.

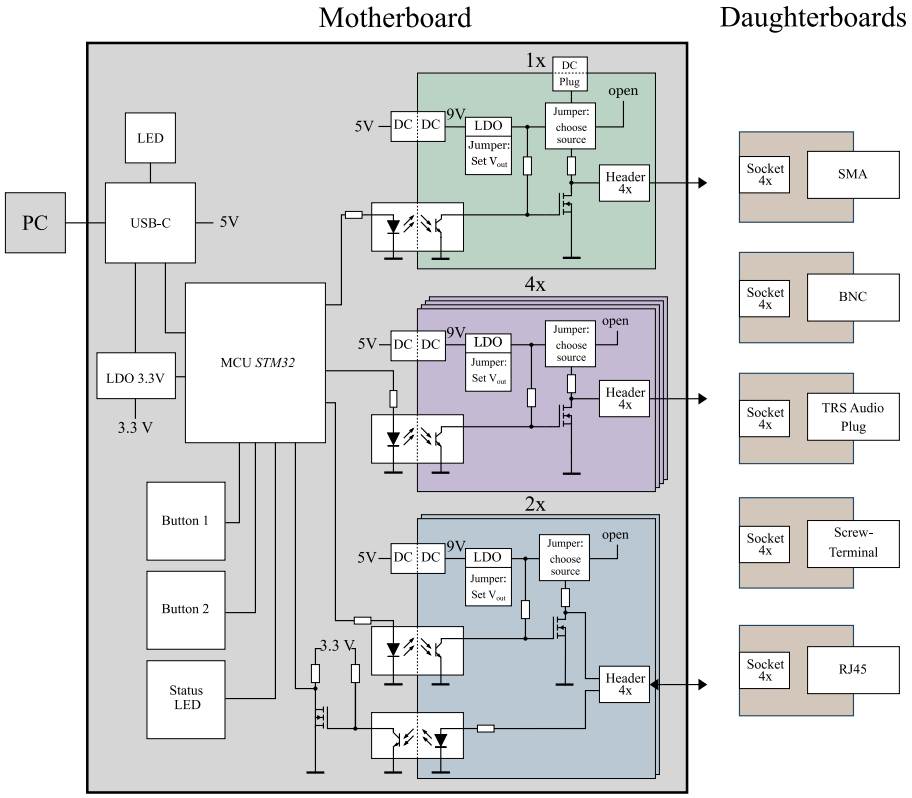


Figure 3: Block diagram OSB hardware: USB-C connector for Power supply (5 V) and as serial communication interface, 4 output ports, 1 output port with external DC Plug, 2 ports adjustable as in- or output, 5 daughterboards with different connectors (SMA, BNC, TRS Audio Plug, Screw Terminal, RJ45).

The board is supplied with power from the computer via a USB-C connection. Upon power connection, the LED adjacent to the connector illuminates, indicating that the OSB is ready for operation. The *STM32L151C8T6A* microcontroller (MCU) serves as the central control element of the board, operating on a 3.3 V input voltage. Therefore, the initial 5 V voltage provided by the USB-C connector is regulated down to 3.3 V using a low dropout regulator (LDO). The MCU establishes two-way communication between the board and the computer via a serial port (USB 2.0) over a USB-C connector.

If an input signal to start or stop the measurement or to indicate an event occurs, an interrupt signal is transmitted to the MCU. After the measurement has been started, the synchronization signals are sent from the MCU via general-purpose input-output (GPIO) pins to the respective MDs.

Unwanted galvanic couplings can disturb the signal integrity of a system as well as the reliability of safety functions [18]. To prevent ground loops, even among the connected devices, each connection requires separate components to ensure galvanic isolation. Thus, each signal transmitted from the MCU is fed through an optocoupler.

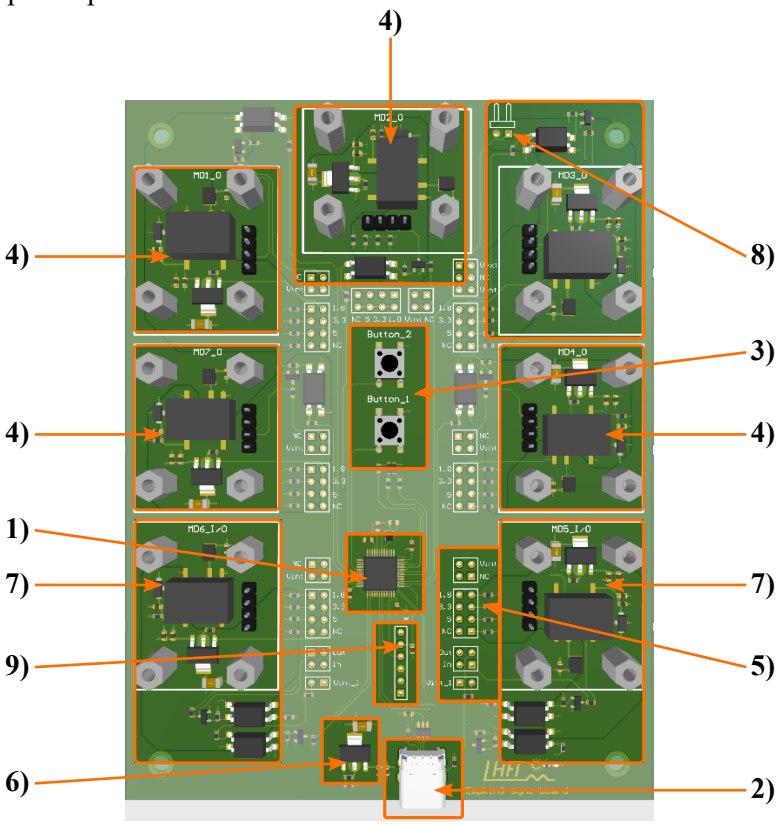


Figure 4: 3D illustration of the PCB of the OSB. 1) MCU, 2) Power supply and serial port over USB-C connector 3) Buttons, 4) Assembly MD Output, 5) Jumper to adjust the required settings, such as pull-up voltage, source of supply and transmitting direction, for each MD, 6) LDO, 7) Assembly MD In- and Output, 8) DC-Plug for MD3, 9) serial interface for flashing the MCU.

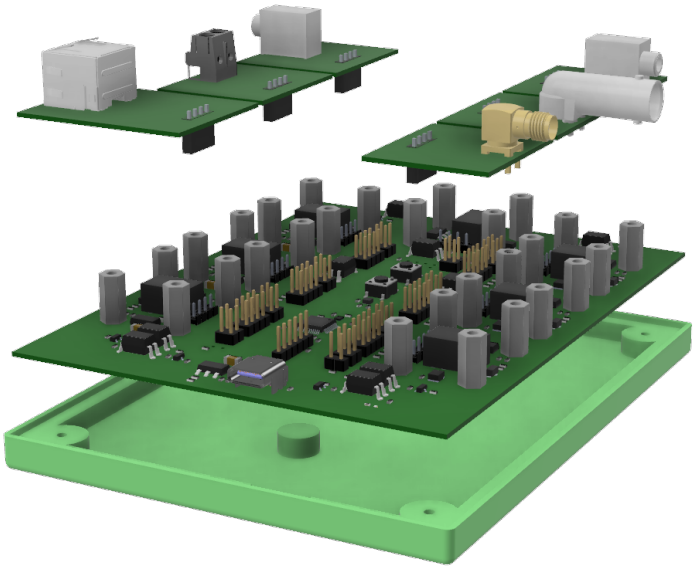


Figure 5: Frontal view of the components of the OSB. 1) 3D-printed mount, 2) PCB, 3) Modular daughterboards that can be attached depending on the required connector types.

Figure 6a illustrates the optocoupler circuit. The emitter LED is driven from the MCU through a current-limiting resistor while the phototransistor drives an N-Channel MOSFET output stage. This ensures a precise output voltage. Otherwise, if I_C is higher than 50mA, the voltage U_{CE} rises, causing the low level to be greater than the threshold voltage $U_{th,low}$.

A DC-DC converter is used to galvanically isolate the supply voltages. For this purpose, a component capable of providing a 9V output voltage was selected. We added the adjustable *LDO LMS8117ADT-ADJ/NOPB* to configure the output voltage within a range of 1.8 V to 7.5 V due to the given input voltage of 9V.

The maximum output voltage is determined by the chosen LDO, as demonstrated in Equation 1 [19]:

$$V_{OUT,max} = V_{IN} - 1.5V \quad (1)$$

Figure 6b shows the application of the used LDO. The output voltage V_{OUT} can be adjusted by the value of the R_2 and is calculated using Equation 2.

$$V_{OUT} = V_{ref}\left(1 + \frac{R_2}{R_1}\right) + \left(I_{adj} \cdot R_2\right) \quad (2)$$

I_{adj} is 60 µA. The OSB offers fixed voltage levels of 1.8 V, 3.3 V, or 5 V, which can be adjusted by connecting different resistor values through jumpers. We selected these voltage levels since they are the most common levels used by most sensors. Furthermore, if these provided values are unsuitable for the corresponding application, a specific resistor value for R_2 that yields the desired output voltage can also be added to the board.

In case higher voltages than 7.5 V are needed, an external voltage can be applied through a DC plug connected to MD3. To accommodate this, jumpers are available for selecting either the adjusted level voltage or the externally applied voltage. If the connected device provides its own pull-up voltage, there is the alternative of positioning the jumper in the unconnected state. In this case, the transistor conducts when a trigger signal is detected, resulting in a grounded output signal.

Apart from the OSB generating output signals, we also allow the board to receive various edge types (falling, rising, or any edge) as input signals that can trigger a measurement or an event message. For that, two of the MDs are equipped with a reverse optocoupler as input that is sampled by the microcontroller. The direction of the MD (input or output) can be configured with a separate jumper. Furthermore, when the port is configured as input, and the jumper labeled V_{int_I} is set, the OSB supplies the level voltage for the input signal, and it can be triggered by short-circuiting the port. In this case, the OSB is triggered by a falling edge. If, this jumper is not set, the connected MD must provide its voltage of 3.3 V at the input port, whose level change also determines the signal form for triggering the OSB.

The motherboard is complemented by different daughterboards that provide adapters for several commonly used connector types (Figure 3). They can be flexibly mounted to the motherboard with a connector strip and the matching socket. For better stability, they are mechanically fixed via four screws with spacer bolts.

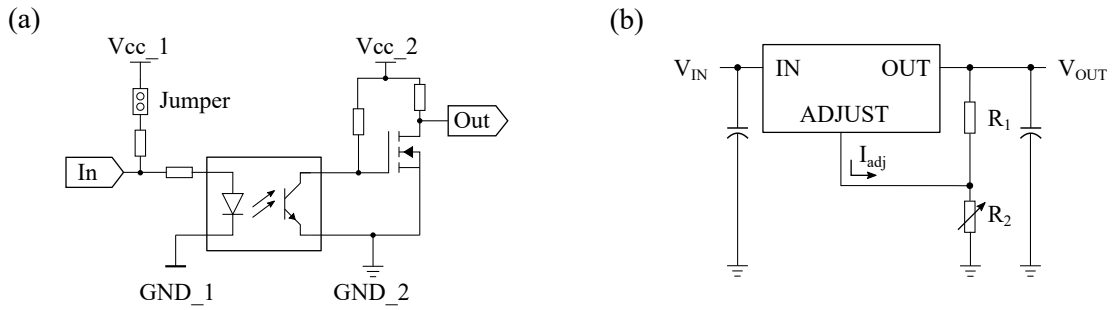


Figure 6: (a) Schematic optocoupler, (b) Adjustable regulator R2 set by Jumper (Texas Instruments Incorporated, 2016).

Firmware

The firmware for the MCU is programmed into the flash memory using the Serial Wire Debug (SWD) interface. It processes the computer's commands to configure desired presets of the OSB, generates synchronization signals, and sends them to the output ports during a measurement.

There are seven available output ports, each capable of generating a synchronization signal independently using a specified waveform. The OSB is connected to a PC via USB for both power supply and serial communication. A comprehensive overview of the command interface can be found in the accompanying material.

The measurement starts when the configured start source is triggered. This can either happen by a button press, when a start command is received from the USB interface, or when a start command with the selected signal form is received at the two MDs capable of sampling input signals (MD5 or MD6).

In addition, if an event trigger occurs at one of the available sources (Button1, Button2, MD5 or MD6) and event reception is enabled for that specific source, the OSB immediately transmits a message to the computer via the USB interface, which can be saved in the GUI. The content of the message indicates the triggering of this source, which is edge-controlled. However, if the event source is either the MD5 or MD6 input port, the OSB will only respond if the received signal format aligns with the chosen one in the GUI.

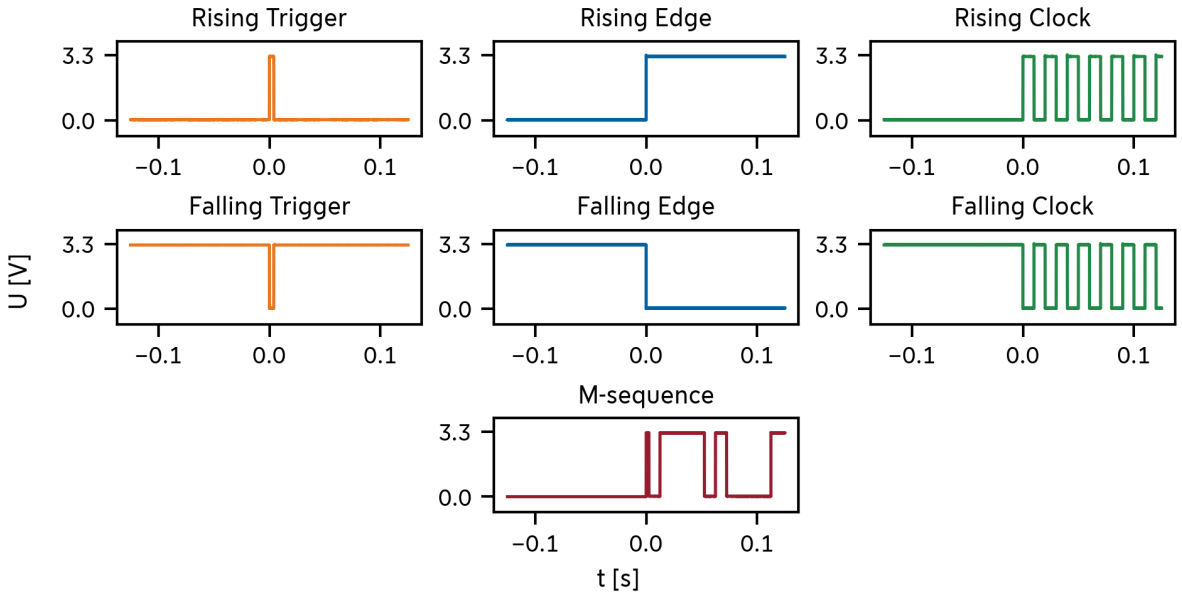


Figure 7: Available synchronization signals with voltage level set to 3.3 V.

Each of the MD ports can be configured to output one of the following synchronization signals, all of which are plotted in Figure 7:

- *Rising Trigger*: A signal that causes a logic level switch from Low to High for the defined pulse duration at the beginning of the measurement.
- *Falling Trigger*: A signal that causes a logic level switch from High to Low for the defined pulse duration at the beginning of the measurement.
- *Rising Edge*: A signal that alters the voltage level from Low to High for the entire measurement duration and then reverts to the default state after the measurement concludes.
- *Falling Edge*: A signal that alters the voltage level from High to Low for the entire measurement duration and then reverts to the default state after the measurement concludes.
- *Rising Clock*: A periodic clock signal that runs at the designated frequency throughout the measurement, starting with a rising edge.
- *Falling Clock: A periodic clock signal that runs at the designated frequency throughout the measurement, starting with a falling edge.*
- *M-Sequence*: A pseudo-random binary sequence that is created using a linear feedback shift register (LFSR). The sequence length I_s relies on the degree n of the LFSR polygon [20]:
  $I_s = 2^n - 1$

The M-sequence can be used to efficiently align two measurements. The cross-correlation of the M-sequence yields periodic peaks whose spacing depends on the sequence length and the bit transmission frequency. To measure the time delay between two recorded sensors, the cross-correlation between their recorded M-sequences with the same degree is computed. If no delay occurred, a peak of the cross-correlation function occurs at $t_0 = 0$. However, if there is a delay, the peak occurs at $t_0 \neq 0$, and the signal must be shifted by t_0 to eliminate the delay. Figure 8a shows the peaks occurring once without delay (dotted) and once with a delay of 100 ms (solid). When permanently recording the M-sequence during a measurement, sampling rate deviations due to sensor drifts can be compensated. For this, a Fast Fourier Transform (FFT) needs to be computed on the M-sequence to determine the actual sampling rate. Afterwards, the drift between desired sampling rate and actual sampling rate can be computed and the signal can be resampled to obtain time-series data sampled at the actual sampling rate. Due to this characteristic, the M-sequence is particularly beneficial for synchronizing long-term measurements. The OSB can generate M-sequences with a maximum degree of $n = 11$, resulting in a maximum sequence length of $2^{11} - 1 = 2047$ samples. For recordings exceeding this maximum length, the M-Sequence is periodically repeated. The set M-sequence is permanently sent with the adjusted sampling rate. The larger n, the wider the spacing between the periodic peaks of the cross-correlation, as illustrated in Figure 8b. Therefore, setting a higher value for n is recommended to get a better temporal resolution.

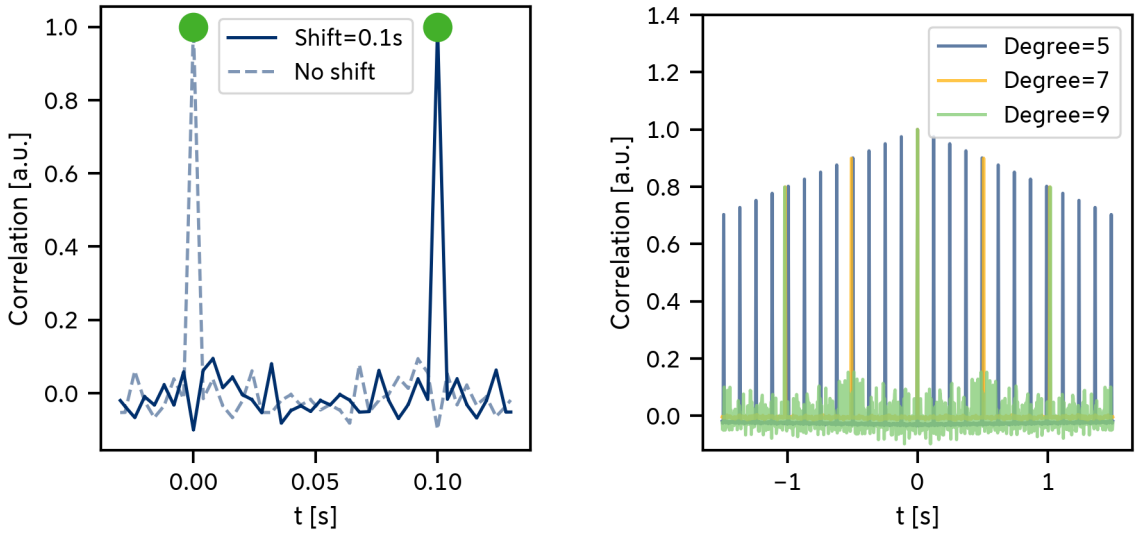


Figure 8: a) Cross-correlation of an M-sequence (n = 9, f = 250 Hz) with and without a delay of t = 100 ms, respectively and (b) Auto-correlation peaks of M-sequences (f = 250 Hz) with different degrees (n = 5, 7, 9). The higher the degree, the higher the delay that can be corrected unambiguously.

The maximum frequency that can be generated is 5.1kHz due to the limited clock configurations of the MCU.
Additionally, the output signals to the MDs can be individually delayed by a pre-defined time.
The command for setting the desired delay in ms is two bytes long. Thus, the time delay can be set for a maximum duration of $2\ \text{ms} = 65.536\ \text{ms} \approx 1.1\ \text{min}$.

Software

For users familiar with programming, the OSB can be configured and controlled programmatically using the provided serial command interface[2]. However, to allow all users to operate the OSB easily and intuitively, we developed a GUI for communication with the PCB. The GUI was implemented using PySide, a Python package that provides bindings to the cross-platform GUI development kit Qt[3].
When starting the GUI, it scans automatically for the connected OSB, establishes a serial port connection, creates a log file for the session at a pre-selected location, and opens the main window. All settings made in the GUI are directly sent to the OSB by a virtual serial port via USB. In parallel, a background thread constantly listens to messages from the OSB to notify the user about status changes or possible errors. The main GUI window resembles the PCB layout (Figure 9). The activated ports are highlighted in green for a clearer identification. Every active port has a settings button that allows users to choose between manual and automatic configuration of different settings, such as out- put signal type, frequency, or delay. For automatic configuration, arbitrary further sensors can be added programmatically apart from the default devices provided by us (Table 1). Furthermore, the user is guided with instructions on jumper placement and connector usage that are required for the selected device.
The two ports that are also capable of receiving input signals provide a configuration option as an event listener. In this case, every time the specified edge signal is received, an event will be logged and written to the log file. This feature can be used as a convenient way to annotate study phases using an external switch.

[2] https://github.com/empkins/open-sync-board-hardware#command-interface
[3] https://doc.qt.io

Additionally, different options are provided to start and stop synchronized measurements, either directly from the GUI, by pressing a button, or triggered by an external input signal that is fed into one of the two input ports. Timestamps of all events are written to a log file automatically.
The current OSB configuration can be exported to a file and imported in a subsequent session to be reused across various participants in a study or between multiple studies.

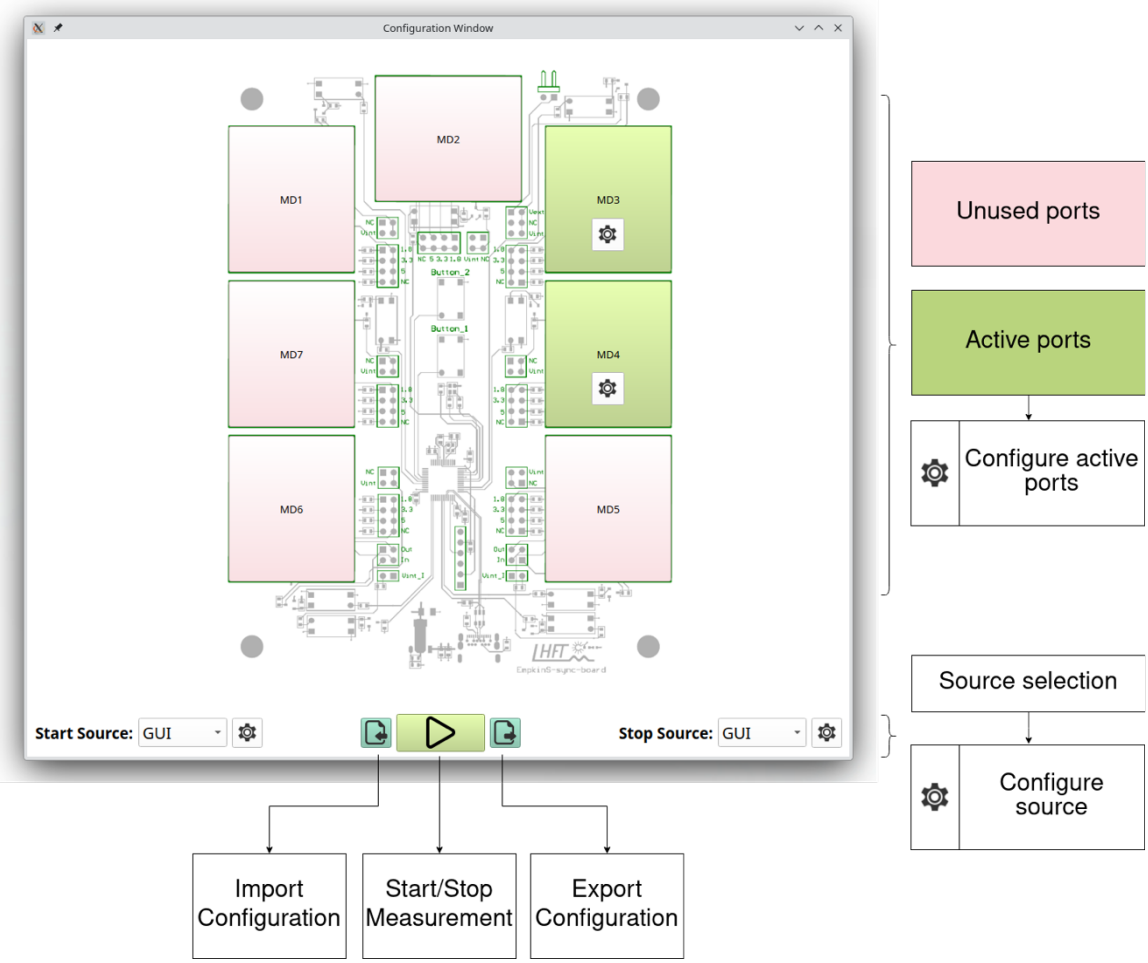


Figure 9: Overview of main functionalities of OSB GUI: Select active ports, specify their configuration, select start and stop modality, save and import previous configurations.

Postprocessing

To facilitate the alignment of OSB-recorded data, we provide the Python module *empkins_io.sync* along with comprehensive examples[4]. This module includes an interface for adding the recorded data of a measurement session and specifying the utilized synchronization signal. Once a primary dataset is selected, the remaining data can be aligned with respect to the primary timestamps. Additionally, there is an option to cut all data to the synchronization region or to the length of the shortest recorded data stream. To utilize this interface, it is necessary to load the recorded data in Python, a task that can be easily accomplished for all common data formats. Furthermore, it is implied that all data streams share the same synchronization signal.

[4] https://github.com/empkins/empkins-io/blob/main/examples/synchronization/ Synchronization.ipynb

(2) Quality control

Calibration

Ideally, the MCU should transmit the signals to the output ports immediately after triggering the start of the measurement through the selected start source. However, in practice, there is a slight delay detectable. To determine the delay, we experimentally measured the time lag between the start signal and the detected output signal at MD1. Utilizing an oscilloscope (Saleae Logic Pro 16) with a sampling rate of 1 562 500 samples $\cdot$ s^{-1}, we iterated the measurement N = 7 times for each input. We obtained average delays of 1.7 ms ± 0.3 ms (Button 1), 1.8 ms ± 0.4 ms (Button 2), 1.6 ms ± 0.2 ms (MD5), and 1.8 ms ± 0.2 ms (MD6) between the triggering of the start signal and the generation of the output signal. This delay is mainly caused by the firmware code, which is structured as simple as possible to ensure maximal readability and easy adaptability. Further, this delay does not impact the synchronization between the single recordings, but only refers to the start delay of the measurements. For human-centered research, such measurements are usually taken for a period of at least several minutes up to several hours, making a start delay in milliseconds range negligible.
Furthermore, the MCU doesn't simultaneously transmit the synchronization signals to the output ports. They experience slight delays attributed to the clock frequency of the MCU since instructions can only be processed sequentially and not in parallel. To measure the time lag between OSB outputs, and thereby evaluate the precision of the OSB, we generated rising edge signals on each MD and captured the output using the same oscilloscope (Saleae Logic Pro 16) with a sampling rate of 781 250 samples $\cdot$ s^{-1}. The results are shown in Figure 9a. It shows that the OSB yields an approximate delay of 6 µs between each active MD. In case all MDs are activated, there is a maximum delay of 42 µs from the moment the synchronization signal is transmitted at MD1 to when it reaches MD7. This delay was confirmed by repeated test measurements (N = 10). Furthermore, the experiment revealed that the edge steepness of the synchronization signal generated by the OSB is approximately 2.552 µs.

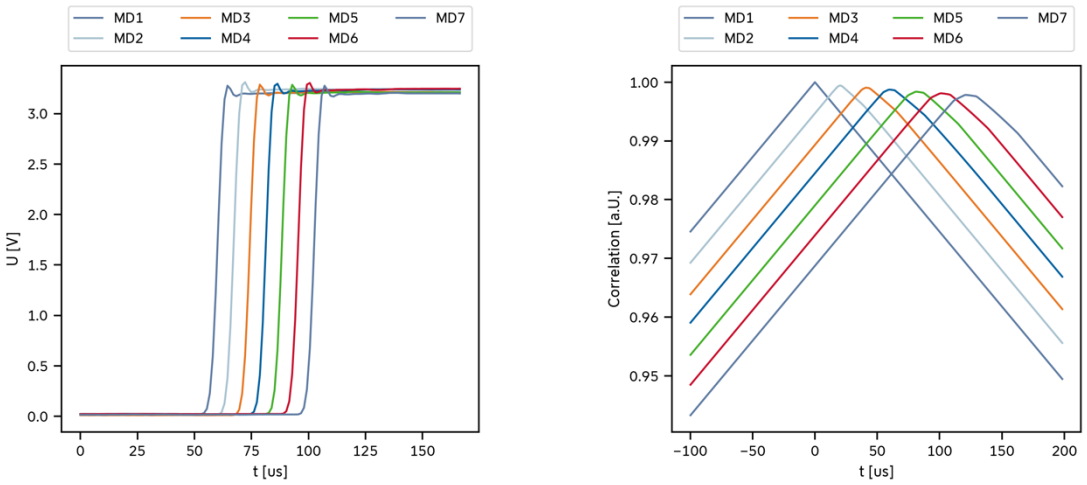


Figure 10: The time delay of (a) the rising edge synchronization signal among all MDs and (b) between the autocorrelation of the M-sequence synchronization signal at MD1 and the cross-correlation of the remaining M-sequences with the reference M-sequence at MD1.

As detailed in the subsection Firmware, M-sequences can be used for synchronization by calculating offsets between their cross-correlation peaks. To quantify the measurement delay for M-sequences as

synchronization signal, we generated an M-sequence (n = 9, f = 250 Hz) on every MD. We recorded the associated output signals using an oscilloscope (Saleae Logic Pro 16). The results were correlated and the correlations averaged over repeated test measurements (N = 10).
The resulting delays in the cross-correlation peaks when using MD1 as reference output are depicted in Figure 10b. Between each output, an approximate delay of 21 µs ± 3 µs was measured, resulting in a maximum delay between MD1 and MD7 of 125 µs ± 14 µs. This delay difference between Figure 10a and Figure 10b arises from how the signal is generated in the firmware.
However, this maximum delay of 125 µs corresponds to an inaccuracy of one sample at a sampling frequency of 8 kHz. This is considerably higher than the commonly used sampling frequency of physiological sensors, such as Electrocardiography (ECG, up to 1 kHz [21]), Electroencephalography (EEG, up to 4 kHz [22]), or Polysomnography (PSG up to 256 Hz [23]). Thus, our OSB is capable of reliably synchronizing different data streams with sub-sample accuracy, even for the case of a maximum synchronization delay of 42 µs for trivial synchronization signals, or 125 µs for pseudo-random synchronization signals, respectively, when all measurement devices are active.

(3) Application

Since the OSB was designed to offer high flexibility and usability, it is suitable for many different application scenarios. To demonstrate the wide variety of possible applications, we present a selection of different use cases on how the OSB is used in research projects involving multimodal human-centered research. Without the modular character of the OSB, custom-build solutions would have been required for all three use cases. On the one hand, the OSB can be applied to synchronize long-term (Use Case 1) and short-term (Use Case 2) lab-based measurements for the validation of novel sensors (Use Case 1) as well as for commercially available sensor systems (Use Case 2). On the other hand, short-term measurements can be synchronously triggered based on an external event (Use Case 3).

In addition to the sensors presented in our three use cases, we tested the compatibility of further sensors with the OSB. Table 2 lists these sensors and how they can be connected to the OSB.

Use cases

Use Case 1: Long-Term Measurement for Sleep Analysis

The gold standard for sleep monitoring, and thus, for identifying possible sleep disorder patterns, is laboratory-based polysomnography (PSG). As it is costly, obtrusive, and not feasible in a home environment, alternative contactless methods such as radar-based sensors are investigated [24, 25, 26]. For testing our prototypes, we are using a *SOMNO HDTM Eco* (*SOMNOmedics*, Randersacker, DE) as gold standard device. The PSG device handles the synchronization of its connected proprietary sensors internally but does not provide an interface for external sensors from other manufacturers or custom-build experimental sensors. Therefore, to synchronize the PSG recordings with the experimental radar sensor, both are connected to the OSB. The PSG is connected to the PCB with a BNC plug (see Table 2). The radar system prototype is connected to the OSB using an RJ45 daughterboard. Since sleep studies are typically long-term measurements, we use an M-sequence as synchronization signal. The recorded data can be aligned for the next data processing steps using our postprocessing package (see the Section *Postprocessing* for further details).

Use Case 2: Short-Term Measurement for Stress Assessment

Repeated exposure to acute stress can have a substantial negative impact on mental and physical well-being [27]. The state-of-the-art biomarkers for objective stress response assessment require invasive procedures, such as saliva or blood sample collection.

In our collaborative research center, we explore how data from wearable and ubiquitous sensors can be used as an extension to the established stress biomarkers to gain a more holistic picture of the human stress response. For this purpose, we conduct a laboratory protocol for acute psychosocial stress induction (Trier Social Stress Test [28]) while recording motion and physiological data from multiple sensors. For motion assessment during acute stress, we use an inertial measurement unit (IMU)-based motion capture suit (e.g., *Movella XSens*, Henderson, US). The base station of the *XSens* system offers a BNC plug as trigger input that can be directly connected to the OSB using the corresponding daughterboard. For vital sign monitoring, either a modular system such as *BIOPAC* (*BIOPAC Systems Inc.*, Goleta, US), or wearable sensors can be used. Both mentioned systems can synchronize their own sensors among one another and provide a dedicated channel for receiving an external synchronization signal but require a synchronization unit connecting to both synchronization channels and outputting simultaneous signals to them. Again, the OSB can be used for this purpose out-of-the-box.

For the short duration of laboratory experiments (<2 hours), we use single pulses (Falling/Rising Trigger) as synchronization signals at the start and end of the measurement, respectively. During the measurement, biomarker collection times, as well as study phases, can be annotated using the OSB buttons.

Use Case 3: Triggered Measurement for Laboratory-based Movement Analysis

Marker-based motion capturing systems from manufacturers such as *Vicon* (*Vicon Motion Systems*, Oxford, UK) or *QualiSys* (*Qualisys AB*, Göteborg, SE) and force plates (e.g., *Kistler Group*, Winterthur, CH) are the gold standard for estimation of movement kinematics and kinetics and present in almost all motion capture laboratories.

To investigate patients in ambulant care or analyze the performance of athletes, however, less obtrusive approaches are advantageous. Wearable IMU systems (e.g., *Portabiles GmbH*, Erlangen, DE) represent a viable alternative that is validated and improved in ongoing research [29].

The commercial motion capture systems mentioned above include a synchronization unit that can be mounted to an output port of the OSB. Analogously, IMU systems can be either synchronized using a synchronization unit or a dedicated IMU sensor for primary-secondary synchronization. As force plates integrated in the floor typically only allow to capture a limited number of strides, it is practical to trigger all measurement devices as soon as a study participant enters the area of interest, for instance, using light barriers (e.g., *Pepperl+Fuchs SE*, Mannheim, DE) as external sources. A light barrier could be connected to the synchronization unit of either the marker-based or the IMU-based system. However, to trigger both systems simultaneously based to on the same light barrier, the OSB can be used as central synchronization unit, with the light barrier attached as start source. As soon as the light barrier is crossed, it emits a pulse signal that triggers the OSB to start a measurement session and sends the pre-configured synchronization signals to the connected sensors. Correspondingly, another light barrier could be employed as stop source, to stop the measurement as soon as it is crossed.

Table 1: Exemplary list of tested devices and their properties.

Device	Sensor Type	Plug	Logic Level [V]
BIOPAC MP36, MP160, or similar	Modular system for vital signs	Connection via screw terminal	5.0
SOMNO HD™ Eco	PSG for sleep staging	BNC	6.0
Azure Kinect	Depth camera	TRS Audio Plug	5.0
Intel RealSense	Depth camera	Connection via screw terminal	1.8
Portabiles NilsPod	IMU	BNC	3.3
Xsens	IMU-based motion capture	BNC	5.0
OptiTrack	Marker-based motion capture	BNC	3.3
QualiSys	Marker-based motion capture	BNC	5.0

(4) Build Details

Availability of materials and methods

To manufacture the circuit board of the OSB, the individual electronic and integrated circuits (ICs) components must be procured. These are common parts available for standard online vendors. A comprehensive list of all required components can be found in an inactive bill of materials (BOM) of the OSB provided by *Altium*[5]. This BOM can be accessed without the need to install any software. Furthermore, the PCB design incorporates a milled edge to facilitate the separation of the motherboard from the daughterboards. Therefore, the PCB should be manufactured by an online PCB manufacturer

[5] https://365.altium.com/files/F87B11EB-A8F8-4EBC-8E65-F456D1F7A0D8

that provides this service. Additionally, screws (M3x20 head screws) and spacer bolts (M3x40 hexagonal bolts) with a length of 11 mm are required to securely fasten the daughterboards on the OSB.

Ease of build

The OSB can be ordered pre-populated from a PCB manufacturer. However, developers can also assemble them themselves, but they must possess solid soldering skills to do so. To bolster the OSB's resilience and protect it against external influences, it is advisable to employ a mount in which the OSB is installed. Nevertheless, this requires access to a 3D printer to print the construction of this mount, which can be found in the repository of this project.

Operating software and peripherals

The firmware and software code are provided and documented in GitHub repositories[6].
After manufacturing the OSB, it is necessary to upload the firmware onto the MCU. Therefore, *stmCubeIDE* must be installed and a programming device is needed. More detailed instructions for flashing the OSB can be found in the corresponding firmware repository[7]. It is possible to rebuild the OSB without installing the PCB design software, Altium Designer. However, if modifications to the hardware design of the OSB are desired, it must be installed.
To configure the OSB through serial communication, the OSB GUI or custom scripts can be used. To operate the OSB GUI, we are providing executable standalones for Windows and MacOS that can be downloaded from the releases page of the corresponding software repository[8]. For other operating systems, the OSB GUI can be downloaded from the Python Package Index, which requires Python to be installed on the system.
For building custom scripts, a detailed overview of the serial communication interface of the OSB is also provided on the landing page of the firmware repository.

Hardware documentation / build instructions / files location:

Archive for hardware documentation and build files
Name: GitHub
Persistent identifier: https://github.com/empkins/open-sync-board
Publisher: Robert Richer
Date published: 2023-04-23

(5) Discussion

Conclusions

In this work, we introduced the Open Sync Board, a versatile, open-source hardware-based solution for synchronizing data from multi-sensor setups with a special focus on flexibility, usability, and accuracy. In contrast to hardware-based synchronization solution that are custom-build for a single specific study or use case, our system constitutes a highly modular synchronization unit that can easily be reused for

[6] https://github.com/empkins/open-sync-board
[7] https://github.com/empkins/open-sync-board-hardware
[8] https://github.com/empkins/open-sync-board-gui/releases

different sensor and connector types, various measurement durations, and externally triggered as well as manually started measurements. We discussed the structure of the accompanying PCB and the configuration options it offers. Further, we explained the provided GUI that ensures a seamless operation even for non-technical users. In addition, we outlined several use cases in the field of biomedical sciences. However, the applicability of the OSB is not limited to such use cases. By publishing the OSB and all accompanying resources and tools to rebuild it under an open-source license, we make it freely accessible to researchers. In doing so, we hope to encourage researchers across different fields to utilize and build upon our work.

Paper author contributions

JB, SB, and RR had the idea for the OSB. MP, JB, and SB designed and constructed the hardware. MP wrote the firmware code. AM designed and implemented the GUI. RR implemented and tested the postprocessing code. MP, JB, SB, RR and NS validated and tested the functionality. NCA, AK, BME, and MV were the technical supervisors. MP and AM wrote the documentation and the manuscript. All authors reviewed the manuscript.

Acknowledgments

We would like to thank Arne Küderle and Liv Herzer for their input and feedback.

Funding statement

This work was funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) – SFB 1483 – Project-ID 442419336, EmpkinS.

Competing interests

RR is Chief Executive Officer of Portabiles GmbH and employed part-time since March 2018. BME is co-founder and co-owner of Portabiles GmbH. Portabiles GmbH does not hold any intellectual property in the EmpkinS Sync Board. All other authors have no competing interests.

References

[1] Gravina, R, Alinia, P, Ghasemzadeh, H, Fortino, G. Multi-Sensor Fusion in Body Sensor Networks: State-of-the-art and Research Challenges. Information Fusion 2017; 35:68–80

[2] Liu, Y, Shukla, D, Newman, H, Zhu, Y. Soft Wearable Sensors for Monitoring Symptoms of COVID-19 and Other Respiratory Diseases: A Review. Progress in Biomedical Engineering 2021; 4(1):012001.

[3] Garcia-Ceja, E, Riegler, M, Nordgreen, T, Jakobsen, P, Oedegaard, K, Tørresen, J. Mental Health Monitoring with Multimodal Sensing and Machine Learning: A Survey. Pervasive and Mobile Computing 2018; 51:1–26.

[4] Will, C, Shi, K, Schellenberger, S, Steigleder, T, Michler, F, Fuchs, J, Weigel, R, Ostgathe, C, Koelpin, A. Radar-Based Heart Sound Detection. Scientific Reports 2018; 8(1):11551.

[5] Plotz, T, Chen, C, Hammerla, N, Abowd, G. Automatic Synchronization of Wearable Sensors and Video-Cameras for Ground Truth Annotation – A Practical Approach. In 2012 16th International Symposium on Wearable Computers 2012 (pp. 100-103).

[6] Shrstha, P, Barbieri, M, Weda, H. Synchronization of Multi-Camera Video Recordings Based on Audio. In Proceedings of the 15th ACM International Conference on Multimedia 2007 (pp. 545–548). Association for Computing Machinery.

[7] Woznowski, P, Fafoutis, X, Song, T, Hannuna, S, Camplani, M, Tao, L, Paiement, A, Mellios, E, Haghighi, M, Zhu, N, Hilton, G, Damen, D, Burghardt, T, Mirmehdi, M, Piechocki, R, Kaleshi, D, Craddock, IA. Multi-Modal Sensor Infrastructure for Healthcare in a Residential Environment. In 2015 IEEE International Conference on Communication Workshop (ICCW) 2015 (pp. 271–277).

[8] Peng, YT, Lin, CY, Sun, MT, Landis, C. Multimodality Sensor System for Long-Term Sleep Quality Monitoring. IEEE Transactions on Biomedical Circuits and Systems 2007; 1(3):217–227.

[9] Herzer, L, Muecke, A, Richer, R, Albrecht, N, Heyder, M, Jaeger, K, Koenig, V, Koelpin, A, Rohleder, N, Eskofier, B. Influence of Sensor Position and Body Movements on Radar-Based Heart Rate Monitoring. In 2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI) 2022 (pp. 1–4). IEEE.

[11] Artoni, F, Barsotti, A, Guanziroli, E, Micera, S, Landi, A, Molteni, F. Effective Synchronization of EEG and EMG for Mobile Brain/Body Imaging in Clinical Settings. Frontiers in Human Neuroscience 2018; 11.

[12] Blum, S, Holle, D, Bleichner, M, Debener, S. Pocketable Labs for Everyone: Synchronized Multi-Sensor Data Streaming and Recording on Smartphones with the Lab Streaming Layer. Sensors 2021; 21(23):8135.

[13] Razavi, M, Janfaza, V, Yamauchi, T, Leontyev, A, Longmire-Monford, S, Orr, J. OpenSync: An open-source platform for synchronizing multiple measures in neuroscience experiments. Journal of Neuroscience Methods 2022; 369:109458.

[14] Michler, F, Shi, K, Schellenberger, S, Steigleder, T, Malessa, A, Hameyer, L, Neumann, N, Lurz, F, Ostgathe, C, Weigel, R, Koelpin, A. A Clinically Evaluated Interferometric Continuous-Wave Radar System for the Contactless Measurement of Human Vital Parameters. Sensors 2019; 19(11):2492.

[15] Schellenberger, S, Shi, K, Steigleder, T, Malessa, A, Michler, F, Hameyer, L, Neumann, N, Lurz, F, Weigel, R, Ostgathe, C, Koelpin, A. A Dataset of Clinically Recorded Radar Vital Signs with Synchronised Reference Sensor Signals. Scientific Data 2020; 7(1):291.

[16] IEC60601-1. General requirements for Basic safety and Essential performance 2014.

[17] Jiang, H, Hu, Y, Ye, S. A Highly Sensitive In Situ CDOM Sensor Based on Frequency Spectrum Shifted m-Sequence Modulation. IEEE Transactions on Instrumentation and Measurement 2020; 69(6):3179-3190.

[18] Hann, G. Introduction to isolation: What is galvanic isolation. Texas Instruments 2018; URL: https://training.ti.com/zh-tw/ti-precision-labs-isolation-what- galvanic-isolation?cu=1135015.

[19] Texas Instruments Incorporated. Linear Voltage Regulator IC Positive Adjustable 1 Output 1A SOT-223 2016; URL: https://www.ti.com/lit/ds/symlink/lms8117a.pdf.

[20] Xinyu, Z. Analysis of M-sequence and Gold-sequence in CDMA system. In 2011 IEEE 3rd International Conference on Communication Software and Networks 2011 (pp. 466-468).

[21] Govindan, R, Massaro, A, Kota, S, Grabowski, R, Wilson, J, Plessis, A. Effect of EKG Sampling Rate on Heart Rate Variability Analysis. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 2019 (pp. 6780-6783).

[22] Zou, B, Zheng, Y, Shen, M, Luo, Y, Li, L, Zhang, L. BEATS: An Open-Source, High-Precision, Multi-Channel EEG Acquisition Tool System. IEEE Transactions on Biomedical Circuits and Systems 2022; 16(6):1287-1298.

[23] Nayak, S, Teja, S, Routray, A, Sarma, MA Non-contact Framework for Shortness of Breath Recognition and effects on Health Parameters during HCI. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) 2020 (pp. 1-7).

[24] Yoo, YK, Jung, CW, Shin, HC. Unsupervised Detection of Multiple Sleep Stages Using a Single FMCW Radar. Applied Sciences 2023; 13(7):4468.

[25] Hong, H, Zhang, L, Gu, C, Li, Y, Zhou, G, Zhu, X. Noncontact Sleep Stage Estimation Using a CW Doppler Radar. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 2018; 8(2):260–270.

[26] Yue, S, Yang, Y, Wang, H, Rahul, H, Katabi, D. BodyCompass: Monitoring Sleep Posture with Wireless Signals. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2020; 4(2):66:1–66:25.

[27] O'Connor, D, Thayer, J, Vedhara, K. Stress and Health: A Review of Psychobiological Processes. Annual Review of Psychology 2021; 72:663–688.

[28] Kirschbaum, C, Pirke, KM, Hellhammer, D. The 'Trier Social Stress Test' – A Tool for Investigating Psychobiological Stress Responses in a Laboratory Setting. Neuropsychobiology 1993; 28:76-81.

[29] Dorschky, E, Nitschke, M, Seifer, AK, Bogert, A, Eskofier, B. Estimation of Gait Kinematics and Kinetics from Inertial Sensor Data Using Optimal Control of Musculoskeletal Models. Journal of Biomechanics 2019; 95:109278.

Copyright notice

Authors who publish in this journal agree to the following terms:

Authors retain copyright and grant the journal right of first publication with the work simultaneously licensed under a Creative Commons Attribution License that allows others to share the work with an acknowledgment of the work's authorship and initial publication in this journal.

Authors can enter into separate, additional contractual arrangements for the non-exclusive distribution of the journal's published version of the work (e.g., post it to an institutional repository or publish it in a book), with an acknowledgment of its initial publication in this journal.

By submitting this paper you agree to the terms of this Copyright Notice, which will apply to this submission if and when this journal publishes it.