Template of Hardware Metapaper

For submission to the *Journal of Open Hardware*

To complete this template, please replace the blue text with your own. The paper has five main sections: (1) Overview; (2) Quality Control; (3) Application (4) Build Details (5) Discussion.

(1) Overview

Title

The title of the hardware paper should focus on the hardware, e.g. “[type of hardware] to automate laboratory protocol X” or “Surface Trawl for Collecting Marine Microplastics.” If the hardware is closely linked to a specific research paper, then “[Hardware type] from [Paper Title]” is appropriate. The title should relate to the functionality of the hardware and the area it relates, to rather than making claims about the usability or user characteristics of hardware, e.g. “Easy-to-use”.
BEE.EEG – an Open Hardware Device to measure behaviour of honey bees (Apis Mellifera) and in hive conditions

Paper authors

1. Last name, first name; *(Lead/corresponding author first)*
Julian Morgan
2. Last name, first name; *etc.*
Karén Timo Tim Menzel
Senior
Ben

Paper author roles and affiliations

1. First author role and affiliation
2. Second author role and affiliation *etc.*

Note: One author can have multiple affiliations and affiliations can include “independent”.

Abstract

A short (up to 250 words) summary of the hardware being described: what problem(s) the hardware addresses, what it does, how it technically/methodologically advances the state-of-the-art, how it was designed and implemented, and its applicability to other issues/research/areas of reuse.

Honey bees are important pollinators in agriculture as well as indicator insects representing the state of the investigated area concerning pesticide use and/or other factors. The rich behavioral repertoire of the eusocial honey bee, in particular their dancing behaviour, can be used as a predictor for the bees' health as a social group. This paper describes the BEE.EEG, a device for recording the dancing behaviour of the honey bee within the hive as well as the activity of bee at the entrance. Furthermore, environmental data like temperature and humidity inside and outside the hive as well as the weight of the hive is recorded. The system saves all data with a GPS time and location stamp on an SD card. The SD card and connected battery can run independently for at least two weeks. The environmental data will be used to correct for the expected amount and characteristics of dances. The BEE.EEG will be distributed across all of Germany in different environmental settings to generate stock data to calibrate the system. The systems high complexity makes it more demanding to reproduce on the hardware level but the resulting data will make up for it by its unprecedented insight into the bee hives communications.

Metadata Overview

Please provide the following overview and self-assessment

Main design files: link to repository with design files and assembly instructions

Target group: e.g. secondary school students, layperson, undergraduate students, scientists in [discipline], trained engineers, use of professional services

Scientists in behaviour, environment and agriculture, citizen science

Skills required: For each main manufacturing method, provide information pairs (method - easy / advanced / specialist). E.g. desktop 3d printing - easy; surface mount PCB - advanced; injection moulding – specialist

3D printing – easy, PDB – advanced (SMD components), source code – specialist (), apiary(advanced)

Replication: Project has been replicated at time of publication? Include reference (possibly way of contact). Links to places where future builds and kits might be found (repository, group website, collaboration site, etc.).

See section "Build Details" for more detail.

Keywords

(required) keyword 1; keyword 2; etc.

Keywords should make it easy to identify who and what the hardware will be useful for. Maximum of 5 keywords. E.g.: microbiology; laboratory; microfluidics; single cell analysis; fluorescence

environment, logger, behaviour, honey bee, STM32

Introduction

An overview of the hardware, what it does, how it was produced, and the research for which it has been used. If the hardware addresses a specific lack or problem, describe the problem in this section and how the hardware addresses the problem. Elaborate how it technically/methodologically advances the state-of-the-art including references to relevant research articles and online references.

The amount of open hardware loggers for honey bee hives is tremendously increasing. More and more devices are capable of wireless data transmission and are solar powered. Nevertheless, the gathered data seem to stagnate concerning its diversity. There are temperature sensors inside and outside the hive as well as humidity sensors. Some include the weight of the hive by load cell scales and fewer the inbound and outbound activity of bees. There are devices recording audio in the hive as the only measure related to behavior so far. We address this lack with our device. The EE.EEG is measuring temperature and humidity inside and outside the hive constantly. It measures the weight of the hive with a three load cell configuration around the clock. The sensors of special interest and real novelty are the multi channel capacitance sensor at the entrance of the hive measuring activity of honey bees in that area and a six channel electric charge measuring array in the hive. This array of J-FET transistors are capable of recording movements of bees in the hive at the area were dancing behaviour is supposed to be taking place. The Capacitance sensor with its two active channels in our device (it generally supports four) samples sata with 100 Hz per second per channel. The transistor based sensor array samples 5000 data points per second per channel. This dence data streams allow for a deeper look into the information flow with in the hive.

The dynamics of activity and dancing behaviour are strongly correlated with weather conditions. therefore, we are using the temperature at and in the hive together with the openly available weather data from the official weather service. This data in combination with the biologically relevant shifts in weight will be useful to get a relation of what certain conditions call for at the behavioral end of the sensors (e.g. activity and dancing probes). With such an predictive approach changes in behaviour related to unknown environmental changes can be investigated. the plan is to use many BEE.EEGs across different environmental conditions and do big data analysis.

old system from audio is not controllable enought (syncrony across6 channels) expensive / power hungry

open hardware systems (arduino raspi) not powerfull enougth, not stable

like this stuff:
https://ac.els-cdn.com/S0925857416300829/1-s2.0-S0925857416300829-main.pdf?_tid=4ef17d26-caf4-48c1-9a45-3a41d48a2eb1&acdnat=1524492441_104a2e0b1ed7115a7fb5f59d62283a31

Overall Implementation and Design

Describe how the hardware was implemented/created, with relevant details of the architecture and design, including general materials (*Note: specific details on materials can be addressed in later section "Availability of materials and methods"*). The use of diagrams and pictures of the assembled hardware is appropriate. Please also describe any variants and associated implementation differences.

Note: This is not meant to be an assembly instruction. Assembly instructions, detailed material lists, and construction files must be deposited in an appropriate repository (see Repositories document) and referenced in section "Build Details".

Subsections

We encourage the use of subsections to highlight the modular functionality of the design or to highlight different design achievements. These questions might help you find appropriate subsections: What parts of the hardware could be used as a component of another project or a variation of this hardware? What solutions do you find most noteworthy given your experiences during the design iterations? What separate actions does the hardware perform?

Software on Microcontroller

Many beehives are located at remote locations. So BEE.EEG has to use batteries. That means the power consumption has to be a low as possible. For that reason, higher performance and more flexibility it was decided to use a STM32F4 Microprocessor [3] and not a Raspberry Pi or Arduino.

To make the code more structured a state machine was used (See figure x).

At startup the BEE.EEG transitions into the **initializing-state** where:

- The SD-card is mounted
- The RTC (Real Time Clock) is synchronised with the GPS time. If no GPS time is available after waiting 5 minutes it will go into the **errorHandling-state**.
- The load-cells are initialized

When no error occured in the Initializing-state it transitions to the configuring-state. In the **configuring-state**:

- The configuration struct is loaded
- It is verified that the it is time for a recording:
  -- If not: It switches to the **nightLogging-state** (one measurements of all sensors (except electric field and activity) is done and saved to file
  -- If yes: The total number of electric field measurements is calculated (5000 per second per channel) and the timer for electric field measurements is initialized

If debugging is needed the system can switch to the **testing-state** where the sensors can be tested. In operation mode the system switches from the configuring-state to the **dayLogging-state** where:

- Where the timer for the electric field measurements is started
  → The data is written to a buffer
- In a loop:
  -- The log files are created (length is defined in the configuration struct)
  -- Sensors are readout
  -- The buffer (containing the electric field measurements) is saved (that causes an interruption of 20-250ms)

When the total number of electric field measurements is reached the system switches to the **finalising-state** where:

- The microprocessor is set into standby mode to save power
- After standby mode it switches to the initialising-state

If an error occurs the system switches to the **errorHandling-state** where an error message is printed (over UART) and it switches to the initialise-state

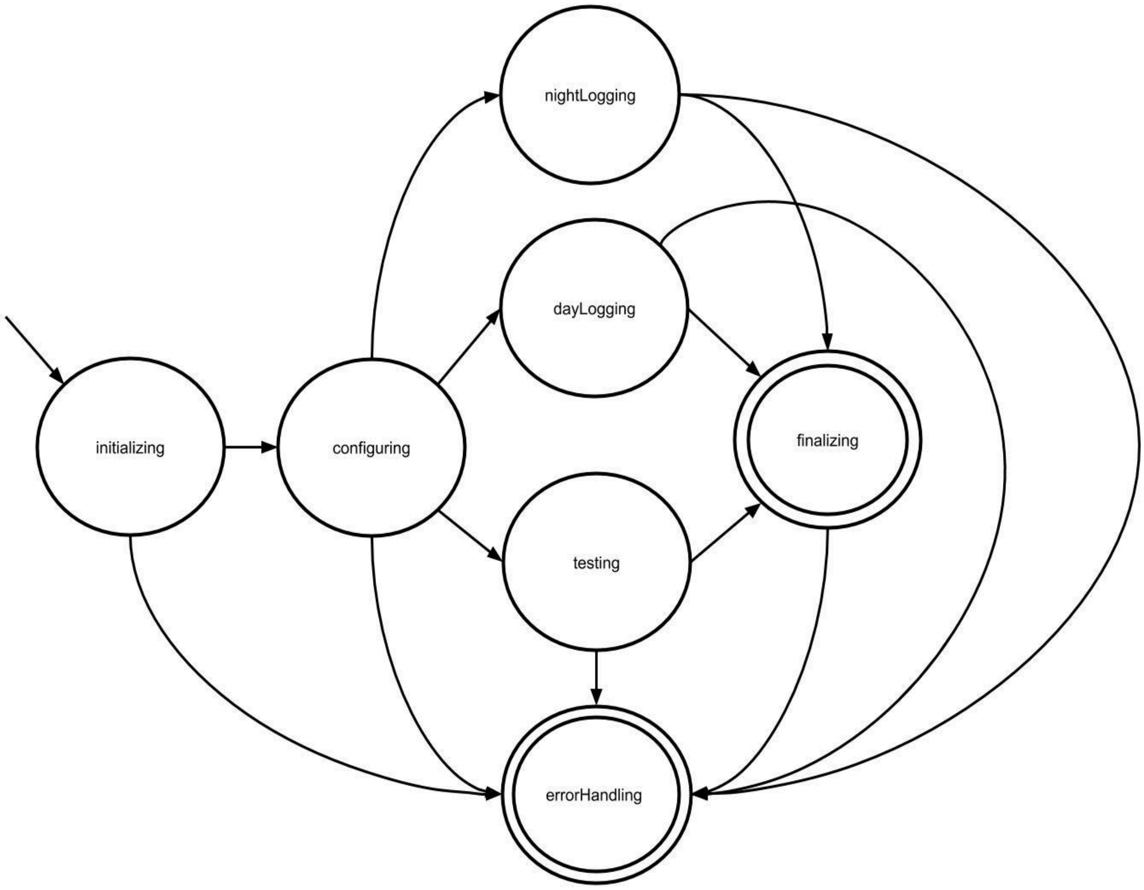
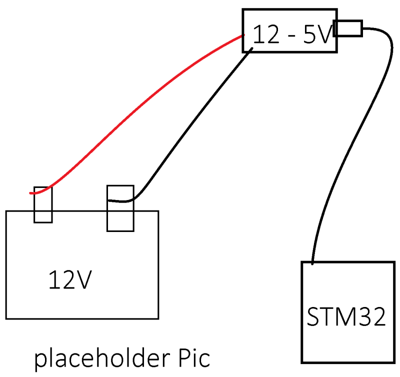


Figure X: State machine representing the software lifecycle of the BEE.EEG device

Hardware

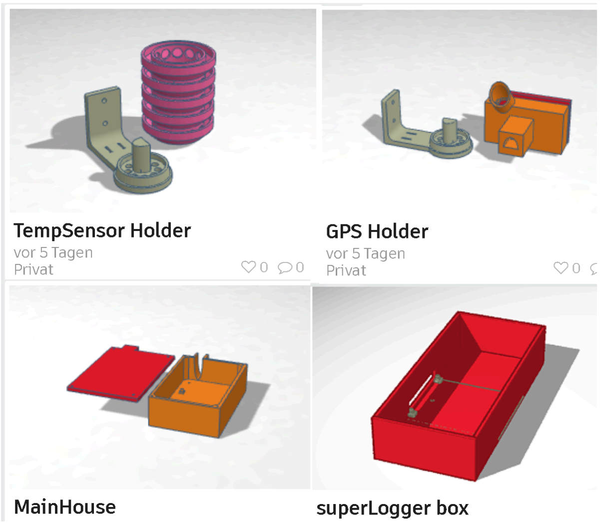
Power Supply

The BEE.EEG is powered by a generic 12V car battery with 80 Ah capacity. It is connected with clamps to a 12V DC to 5V DC USB converter like the ones found in cars. This converter is using a switch mode power supply unit.Those units are often noisy but the one used here is either low noise or has a high switching frequency. The converter also contains a fuse for safety.A USB mini cable connects the converter to the STM32 board with contains a linear voltage regulator to supply itself and the PCB on top to with 3.3 V. All ICs on the custom PCB have appropriate decoupling capacitor nearby. Excessive use of decoupling can be found around the analog area of the PCB.



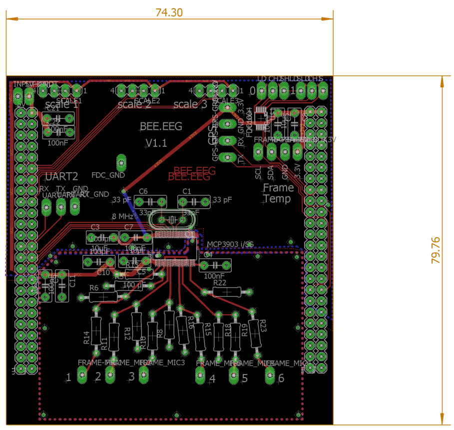
Housing

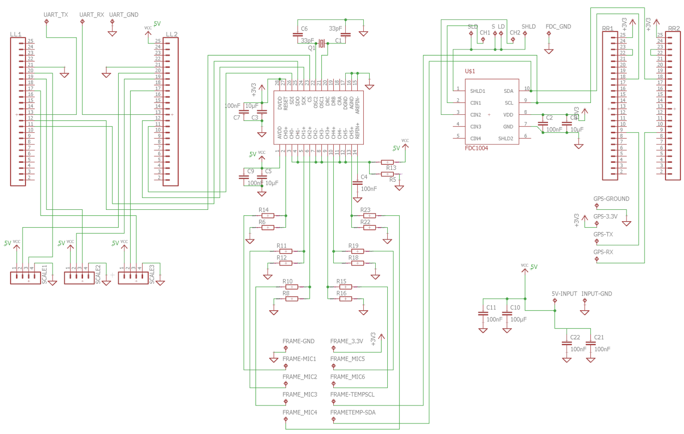
The BEE.EEG device is part of a bee hive and is situated outside under the influence of temperature, humidity, rain and snow. Therefore a 3D printed watertight housing is suggested. Appropriate housing was printed in white PLA plastic and sealed with silicone. This guarantees high whether resilience and the peripheral electronics like GPS and outer temperature sensors can be accessed by breaking the cheap produced housing and reprint.



PBC

The custom designed PCB accommodates all connections to external devices as well as two active components (ADC and capacitance sensor) with the appropriate passive components. The two active ICs are only available in SMD footprint making soldering of the components somewhat difficult for inexperienced replicators.





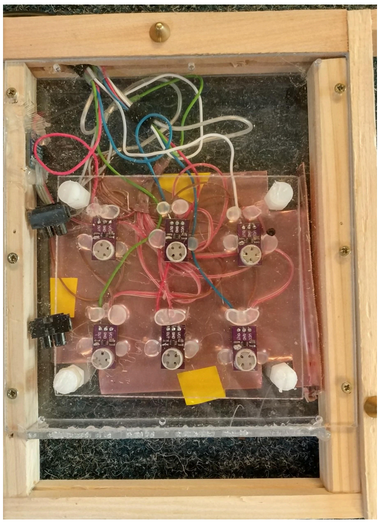
ADC

A 24 bit, 6 channel synchronous Delta Sigma ADC was used to digitize the signals from the microphone preamps. After the PCB was designed it became obvious that 24 bit are way over the top and unnecessarily hard to denoise. nevertheless, the ADC is sampling the data in a sufficient manner therefore we will stick with it for now. As shown in Fig.#ADC# the analog input and the analog pins of the ADC are concentrated on the lower part of the PCB. Power supply is on top and logic below. For future developments the board can be broadened to the left and right to access more pins from the STM32 board.

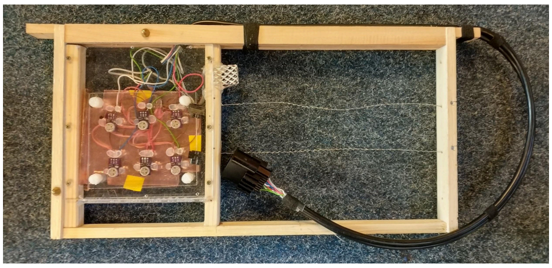
mic boards (dance sensor, plexi) J-FET: 2SK596

The main hardware of this project are the manipulated electret microphone capsules. These capsules can be accessed as a component mounted on an appropriate preamplifier breakout board from china. These boards bandpass filter and lower the impedance of the signal from the microphones, the result is a stabilized signal with a higher signal to noise ratio. The microphones here are manipulated by cutting of the dielectric membrane. afterwards this microphone consists basically of the internal J-FET. Its geometry however is very useful to us, therefore we use microphone capsules instead of using the pure transistor. The preamplifier is adapted to our needs from a audible bandpass filter to a filter with an lower high pass cutoff frequency by replacing the responsible capacitor with a solder bridge. The relevant signal that can be picked up by this arrangement in two different ways. The bees charge up their exoscelletor in flight and this static electric charge can change the potential of the gate pin of the J-FET, it is floating. It could also been shown (not published) that the static field on the plexiglass sheet in front of the gate pin is disturbed by a bees body water content. It implies that the sheet is an electric field lense, with so far neither completely understood properties nor predictable strength or geometry since the static field of the plexiglass sheet itself will change over time. However, movements of the bees body might be sensed this way, while movements of the wings, containing no water, are sensed by the gate by their own electric filed. Further testing and calibration will clarify this topic in the future.

J-FET sensitivity 110dB noise voltage / very linear across frequencies 1 dB 110Hz-1kHz, harmonic distortion -> 1.2% / 25MOhm @ 1kHz input impedance



frame geometry temp

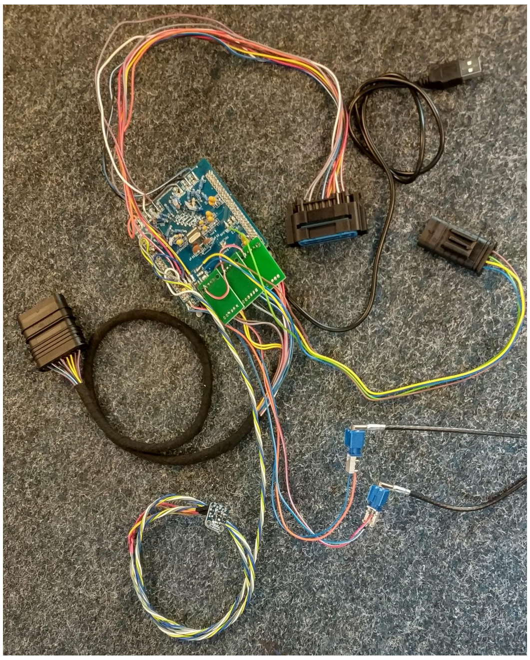


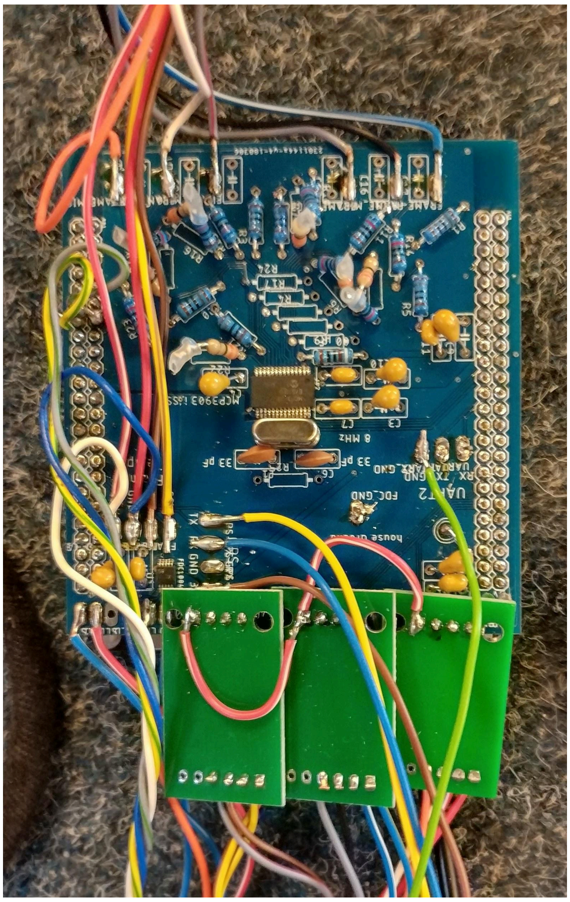
fdc / entrance geometry

scales/geometry

As part of regular telemetry logging, the weight of the hive is measured and recorded on an interval of 1.6 minutes during the active daytime interval and every 5 minutes during the passive night time interval. In order to accurately measure the weight distribution within the hive, three independent scales are used. From a physical component perspective, each scale consists of a double bending beam load cell element which uses a four-wire Wheatstone bridge configuration to connect to a HX711 amplifier. This amplifier is a small breakout board which amplifies the resistance differences received from the load cell and is in turn hardwired directly into the PCB. Recording weight values is accomplished via serial protocol (GPIO). From a software perspective, the driver for this device can be found in the `HX711_scales.c` file. This file contains four functions: (1) `HX711SCALE newScale(GPIO_TypeDef* gpioSck, uint16_t pinSck, GPIO_TypeDef* gpioData, uint16_t pinData, int gain)` which acts a constructor for configuring a new scale where the setup is stored within a `HX711SCALE` C programming structure; (2) `void freeScale(HX711SCALE scale)` which acts as a destructor for the scale to allow the memory allocated on the heap to be freed; (3) `void initScale(HX711SCALE scales)` will initialize the newly configured scale so that it is ready for reading; and (4) `int scaleValue(HX711SCALE scale)` which when called will return the weight integer value from the configured and initialized scale.

gps/outer temp





(2) Quality control

Safety

Describe all relevant safety issues or reference to a risk assessment if included in the hardware documentation. Detail what safety considerations have been included in the hardware design and how these features have been tested, including any official safety standards or criteria that were used in this assessment. If appropriate, discuss the wider context of use of the hardware and safety issues or risks that may arise in the use environment.

The system is powered by DC voltage not exceeding 12 Volts. The system is drawing, when functioning correctly, less than 100 mA. No significant heat is generated. The device is connected to a car battery through a commercial, fused 12V to 5V USB converter. When working with honey bees appropriate care must take place.

Calibration

If the hardware is used for measurements, please detail here how the reliability of measurements, or other hardware properties that are relevant for measurements, has been quantified and explain the results. Be clear about the processes or procedures used to compare the hardware to a standard, as well as the description of the standard calibrated against.
Detail the general procedures in place for users to calibrate their hardware before or during use. What methods can be used to relate user generated data to data from other sources?

Note: Detailed instructions belong in documentation; here, provide insight into how and why the calibration is valid.

Subsections

We encourage the use of subsections within all sections to increase clarity.

Scales

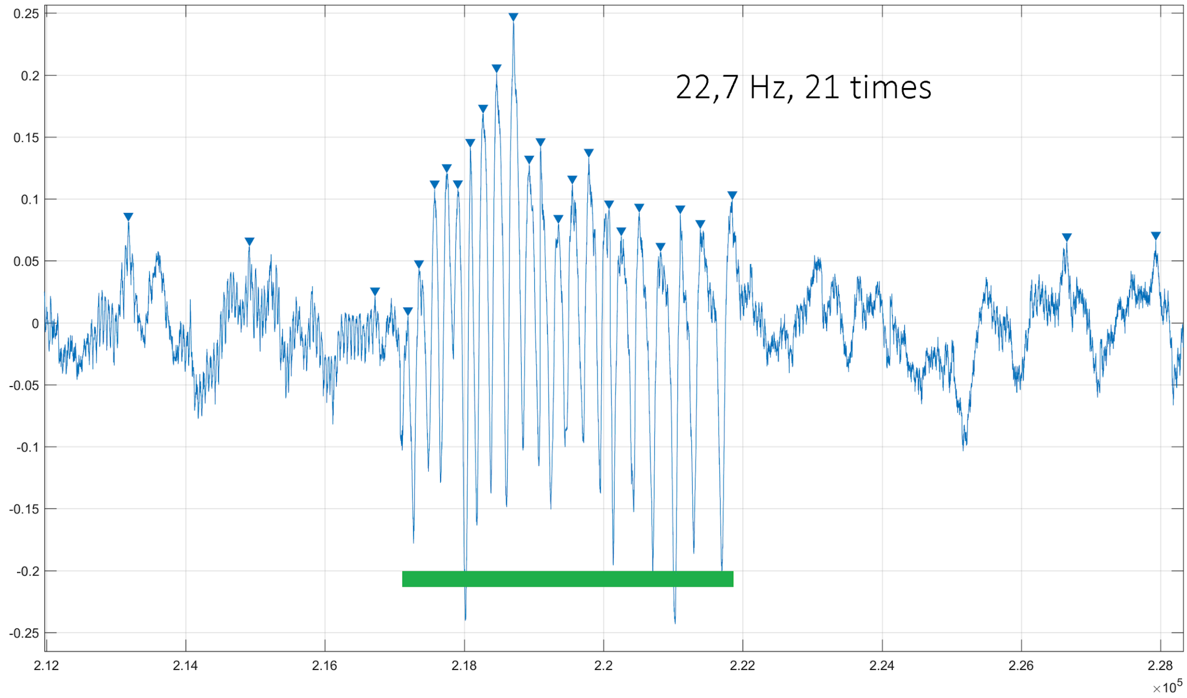
Temperature and Humidity Sensors

digital

activity sensor

VS peronal reference and Video

electric field sensor



General testing

In this section, details can be provided on the testing of hardware functionalities, that are not directly essential for precision operation of the hardware in the given context (which are in turn, where applicable, handled under Calibration), such as automated movements to position the hardware, repeatability of tool exchanges, recyclability, water-tightness, weight or other possibly relevant characteristics. We encourage the authors to characterise all appropriate functionalities of the hardware, if not already described elsewhere (add reference instead). The testing should define the safe/reliable limits in which the components can be operated (e.g. step size and repeatability of linear motion, force ranges, ratio of devices with leaks when built in a workshop, etc). This will enhance the usability of the hardware or method in other contexts.

Again: Detailed instructions belong in documentation; here, provide a summary instead.

(3) Application

Use case(s)

Describe at least one example of an application of your hardware. This should include some evidence of output, e.g. data produced by the use of the device or a picture of other types of results. Outline how the quality control in the previous section enables the use of the hardware in this context. We encourage the inclusion of experiment results or the reference to a publication (published or to-be-published) where these results are detailed. We also encourage pointers to ongoing work.

Note: In the spirit of openness, we require authors to provide (or link to) datasets along with the submitted graphic representations. We do not impose arbitrary limits on inclusion of data so please include sufficient empirical detail and results to ensure your data can be easily verified, analysed and clearly interpreted by the wider scientific community.

Subsections

We encourage the demonstration of different use cases, divided by sub-sections to guide the reader.

Reuse potential and adaptability

Please describe in as much detail as possible the ways in which the hardware could be reused by other researchers both within and outside of your field. This should include the use cases for the hardware, and also details of how the hardware might be modified or extended (including how contributors should contact you) if appropriate. Refer to section "Ease of build" where necessary.
Please provide your thoughts on the adaptability of the hardware design. What tools and procedures would it require for other users to modify your design without your help in order to adapt them for foreseeable or even unforeseeable use.
Also you must include details of what support mechanisms there are in place for this hardware and software (even if there is no support or support community).

(4) Build Details

Availability of materials and methods

Summarise what materials have been used to construct the hardware and what methods to process the materials as well as the assembly. Provide more details or references where important materials or methods are non-standard, not globally available or produced only by one manufacturer.

Ease of build

Have any measures been taken in the design to make the hardware easy to build for other users e.g. reduction of parts, features in the design to make the hardware assembly more reliable?
The here described device contains the least amount of parts as well as the biggest footprints we could access. We used throuhole components whenever possible but the ADC (MCP3903) and capacity sensor (FDC1004) are only available as SMD parts. Further on we decided to use breakout boards, these are extremely easy to use since they come with their

necessary periphery of passive components, but, on the other hand, they are hard to buy from an university account. Most housing parts are 3D printed to ease the build further. The measuring frame needs some basic woodworking and extensive soldering is necessarily involved.

Operating software and peripherals

If hardware requires software, details on the operating software and programming language - Please include minimum version compatibility. Additional system requirements, e.g. memory, disk space, processor, input or output devices.
If the hardware does not require software, detail any required supporting processes or protocols required for use.

BEE.EEG binary to csv converter

To save memory all electric field and activity data is saved binary. So one hour of recording takes 400MB on the SD-Card. If it would record not binary it would need around 1000MB per hour which would limit the runtime of the whole system.

To convert the data into the csv format the software *BEE.EEG binary to csv converter* can be used:

- Link for the Windows executable (If you have Windows and not Python 3): **TODO**
- Link for the Code (Use it if you have Linux): **TODO folder in our repo**

The software converts all binary files in one directory. For each binary file it creates one csv file containing electrical field data, one file for all activity sensor data (per day), and one file containing all other sensor data (per day).

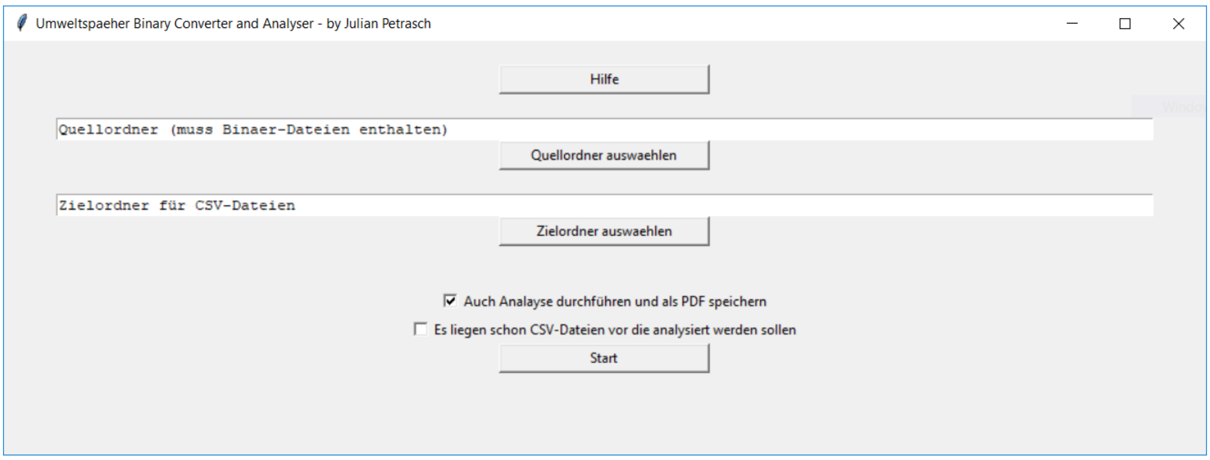


Figure X: **TODO Screenshot from english version**

Dependencies

E.g. other hardware or software projects, modular components, libraries, frameworks, incl. minimum version compatibility. Explicitly state if dependencies are proprietary / closed source.

For the Microcontroller Software: ST-HAL libraries, FATFS

BEE.EEG binary to csv converter: Tkinter

Hardware documentation and files location:

Archive for hardware documentation and build files (required. See *Repositories* document for criteria, DocuBricks and alternative repositories.) *Note: We require the inclusion of modifiable design files as well as a detailed documentation of the design rational of the hardware with assembly instructions. This will be assessed as part of the journal peer review process.*

Name: BEE.EEG

Persistent identifier: https://github.com/julianatastro/BEE.EEG

Licence: Open hardware license under which the documentation and files are licensed - see *Essential author information* for more information

Publisher: Name of the person who deposited the documentation

Date published: dd/mm/yy

Modifiable design files (if different from above)

Name: The name of the emulation environment

Persistent identifier: e.g. DOI, handle, PURL, etc.

Licence: Open license under which the software is licensed here

Publisher: Name of the person who deposited the documentation

Date published: dd/mm/yy

Software code repository GitHub

Name: BEE.EEG

Identifier: https://github.com/julianatastro/BEE.EEG

Licence: Open license under which the software is licensed

Date published: dd/mm/yy

(5) Discussion

Conclusions

Conclusions, learned lessons from design iterations, learned lessons from use cases, summary of results.

Future Work

Further work pursued by the authors or collaborators; known issues; suggestions for others to improve on the hardware design or testing,, given what you have learned from your design iterations.

Paper author contributions

Task (e.g. design, assembly, use cases contribution, documentation, paper writing), contribution, author name.

Acknowledgements

Please add any relevant acknowledgements to anyone else who supported the project in which the hardware was created, but did not work directly on the hardware itself.
Please list anyone who helped to create the hardware and software (who may also not be an author of this paper), including their roles and affiliations.

Funding statement

If the hardware resulted from funded research please give the funder and grant number.

Competing interests

If any of the authors have any competing interests then these must be declared. The authors' initials should be used to denote differing competing interests. For example: "BH has minority shares in [company name], which part funded the research grant for this project. All other authors have no competing interests." Or "BH is selling kits and parts connected to the here presented hardware via platform XX. A fundraising via Crowdfunding platform YY is planned to start commercialisation."
If there are no competing interests, please add the statement:
"The authors declare that they have no competing interests."

References

Please enter references in the Harvard style and include a DOI where available, citing them in the text with a number in square brackets, e.g.

[1] Menzel et al 1960 - 2018

[2] MCP3903 Datasheet: http://ww1.microchip.com/downloads/en/DeviceDoc/25048B.pdf

[3] STM32 Datasheet: http://www.st.com/resource/en/datasheet/stm32f405rg.pdf