NordDesign 2022
August 16-19, 2022
Kgs. Lyngby, Denmark

Promoting open research by adapting proprietary hardware to open-source software

Cathrine Bartnes[1], Agnes Ockernahl[2], Henrikke Dybvik[3] , Martin Steinert[4]

[1] The Norwegian University of Science and Technology (NTNU), Norway
cathrbar@stud.ntnu.no
[2] The Norwegian University of Science and Technology (NTNU), Norway
amockern@stud.ntnu.no
[3] The Norwegian University of Science and Technology (NTNU), Norway
henrikke.dybvik@ntnu.no
[4] The Norwegian University of Science and Technology (NTNU), Norway
martin.steinert@ntnu.no

Abstract

Physiological sensors can give valuable information about the human response when interacting with engineering products or systems, which is essential for engineers to know when developing new innovations. Observing multiple physiological reactions to the same stimuli will help researchers gain a better understanding of how the human body responds. However, there is limited availability of multimodal physiological sensor setups. This article presents an approach utilizing the open-source distribution Lab Streaming Layer (LSL) which records, synchronizes, and exports high-quality data from multiple sensory measures. The challenge is to adapt proprietary hardware to utilize LSL, as most sensors are pre-programmed to interact with their own software. The aim is to reduce the cost associated with subscriptions to proprietary software, by switching to the open-source alternative which promotes higher quality, greater reliability, greater flexibility, and lower cost.

The article presents a case where the proprietary sensors, electrocardiogram (ECG) and galvanic skin response (GSR) provided by Shimmer, adapt to the LSL network. Given the multidisciplinary aspect of the case both Wayfaring, Extreme Programming (XP), and Scrum were used as development methodologies. A description is provided of the software architecture for collecting data from the two Shimmer sensors and how the signals are transmitted to the LSL network.

In addition, the article encourages the use of open-source resources and discusses the benefits that open research brings. Open research offers similar potential as open-source development in terms of rapidity and innovation. The advantage of allowing researchers to benefit from other researchers' findings and results leads to advances that would not have been possible in a closed, or proprietary, environment. In light of the insights acquired in the development process, questions regarding companies' attitudes toward open research are addressed. Moreover, the article adds to the field of open research by broadening the possibilities for experimental setups facilitating the acquisition of new knowledge.

Keywords: Open-source, Open research, LSL, Multimodal experiments, Physiology sensors

1 Introduction

Open research offers the same potential as open-source development in terms of rapidity and innovation. Even though science's credibility would benefit if everyone was more transparent, many individual researchers lack strong incentives to be more transparent (Nosek et al., 2015). Using other researchers' findings and results allows for further innovation and lead to advances that would not have been possible in a closed environment. Keeping the research transparent and available will contribute to reproducibility and validation. In the same way open-source has its set of terms to fulfil (Open Source Initiative, n.d.-b), the Transparency and Openness Promotion (TOP) Committee has developed shared standards for open practices to change the current incentives to drive researchers toward more openness (Nosek et al., 2015).

Physiological sensors can assist engineers in improving the design and structure of systems and products to enable more intuitive interaction, resulting in better products and, implicitly, improved productivity. Several sensors currently come with a software solution that is tailored to the sensor in question. Many physiological sensors are pre-programmed to only interact with their proprietary software. In addition to making it difficult to synchronize sensory data from various sensors across several platforms, the cost of leasing proprietary software can be high in the long run. This makes conducting experiments with multiple physiological sensors time-consuming and expensive. Moreover, because the firmware is often costly, only the wealthy can conduct research. The loss of potential research due to easily accessible resources is far too great, and the area could benefit from all contributions (Pinti et al., 2020). As a result, the economic factor is dividing the field.

This article discusses how the use of open-source resources, namely Lab Streaming Layer (LSL), can benefit the research domain and reduce cost by removing the use of proprietary software. The advantages and drawbacks of open-source software are addressed, demonstrated by a development case where proprietary hardware is adapted to open-source software. The case presents how the proprietary physiological sensors, developed by Shimmer Research (Shimmer Research, n.d.-b), are integrated into the open LSL network. This incorporation of open-source software solutions contributes to a low-cost experiment setup.

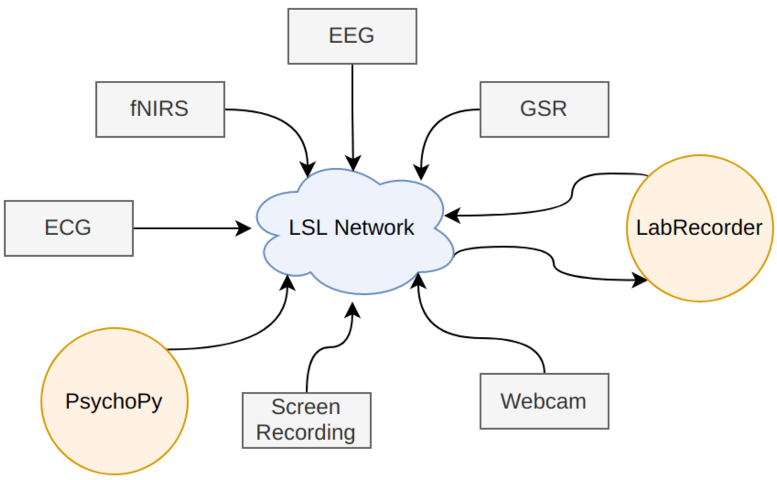
The case is inherently multidisciplinary, combining engineering, computer science, signal processing, and physiology. Thus, the development process had to embrace agile development which is influenced by the methodologies of Wayfaring (Gerstenberg et al., 2015; Steinert & Leifer, 2012), Extreme Programming (Sommerville, 2011), and Scrum (Schwaber, 1997).
Within engineering design, the case illustrates how both software and product development overlap.

2 Case presentation and theoretical background

The case presented in this article takes advantage of the already existing experiment setup (Dybvik et al., 2021), and proposes a solution where it extends the current functionality. Physiological sensors can give valuable information on the human response in interaction with machines and computers (Balters & Steinert, 2017). To gain a better understanding of the human response pattern, it is encouraged to observe multiple physiological responses to the same stimuli. Applying data triangulation, which combines multiple data sources and research methods, will contribute to reducing or at least detecting bias and error (Blessing & Chakrabarti, 2009). Thus, the results will be seen as more credible since each response confirms the other, ensuring that the correct conclusions are drawn. The full experiment setup currently consists of the sensors electroencephalography (EEG), functional near-infrared spectroscopy (fNIRS), galvanic skin response (GSR), and electrocardiogram (ECG), which are recorded by proprietary software. Given that the existing experiment setup already has enabled streaming of data to the LSL network, it is only the GSR and ECG sensor that must be adapted to the open-source software distribution LSL. The overarched aim is to collect all measurement data solely by utilizing open-source software. As such, what remains is to implement a custom script to adapt the configuration of GSR and ECG, so the data can be streamed to the LSL network. The case presented here is the development process of retrieving and collecting GSR and ECG data, collected by Shimmer hardware units, and enabling LSL compatibility.

To get a better understanding of the experiment setup and the integration of the various sensors, the necessary technical information is provided. “Figure 1 – Architecture of experiment setup” shows how the LSL network is connected to hardware (represented by boxes) and open-source software (shown by circles).

Figure 1 – Architecture of experiment setup



2.1 Lab Streaming Layer

The Lab Streaming Layer is an open-source system developed by Christian Kothe, David Medine, Chadwick Boulay, Matthew Grivich, and Tristan Stenner. The system handles time-synchronization, networking, and real-time access in addition to viewing and disk recording of the data (Kothe et al., 2019). The system comes with a core library called “liblsl,” a recording program called LabRecorder, file importers, and a wide variety of applications suited for various measurement units. LabRecorder collects all sensor streams available to the LSL network into one single file provided with the Extensible Data Format (XDF)(*Extensible Data Format (XDF)*, 2015/2021). The XDF format is designed specifically for bio signal data, given the general-purpose container format for multi-channel time-series data with extensive associated meta-information.

2.2 Shimmer

Shimmer Research is a leading provider of wearable sensor products including technology services (Shimmer Research, n.d.-b). They provide a proprietary software solution called ConsensysPRO, which records and displays sensor measurements, in addition to various APIs. The physiological sensors used in the experimental setup described in this article are Shimmer3 GSR+ and EXG. The units come pre-programmed with the LogAndStream firmware, which is the configuration allowing for serial connection through Bluetooth as well as capturing data onto the SD card of the units (Shimmer Research, 2018). The firmware is developed in the proprietary integrated development environment Code Composer Studio (CCS) by Texas Instruments, which is the distributor of the microcontroller used by the Shimmer3 units (Shimmer Research, 2017). The IDE supports C and C++, which is the programming language used to configure the units (Texas Instruments, n.d.-b). Shimmer openly provides a wide range of source code, including the LogAndStream firmware, available through their GitHub account.

2.2.1 GSR

A Galvanic Skin Response (GSR) sensors measure electrodermal activity (EDA), which is the automatic activation of the sweat glands in the skin, due to emotional arousal or the introduction of a new stimulus (Balters & Steinert, 2017; Shimmer Research, n.d.-a).

The Shimmer3 GSR+ unit is equipped with a Texas Instruments MSP430F5437A microcontroller, which among other features consists of a 16-bit RISC architecture and a 12-bit analogue-to-digital converter (ADC) (Texas Instruments, n.d.-c). The mentioned features determine the data signal type, u12, and u16, which must be provided when extracting the data from the sensors with the Struct package.

2.2.2 ECG

An electrocardiography (ECG) sensor measures the electrical activity of the heart. Electrodes are placed on the body's surface and the lead, the difference between them, is measured to determine the electrical potential. Both the sympathetic- and the parasympathetic nervous system can adapt the heart rhythm as a reaction to internal or external triggers (Balters & Steinert, 2017).

The Shimmer3 unit is equipped with two ADS1292R chips from Texas Instruments, which is a 24-bit ADC with integrated respiration impedance (Texas Instruments, n.d.-a).

2.3 Python

The programming language used when implementing the custom scripts of this experimental setup is Python. The language is widely supported, and several packages and libraries are written in the language. The libraries included in the setup are sys, pySerial, struct, and pylsl.

2.3.1 pySerial

The pySerial package is used to connect to the Shimmer devices, through a serial port.

2.3.2 Struct

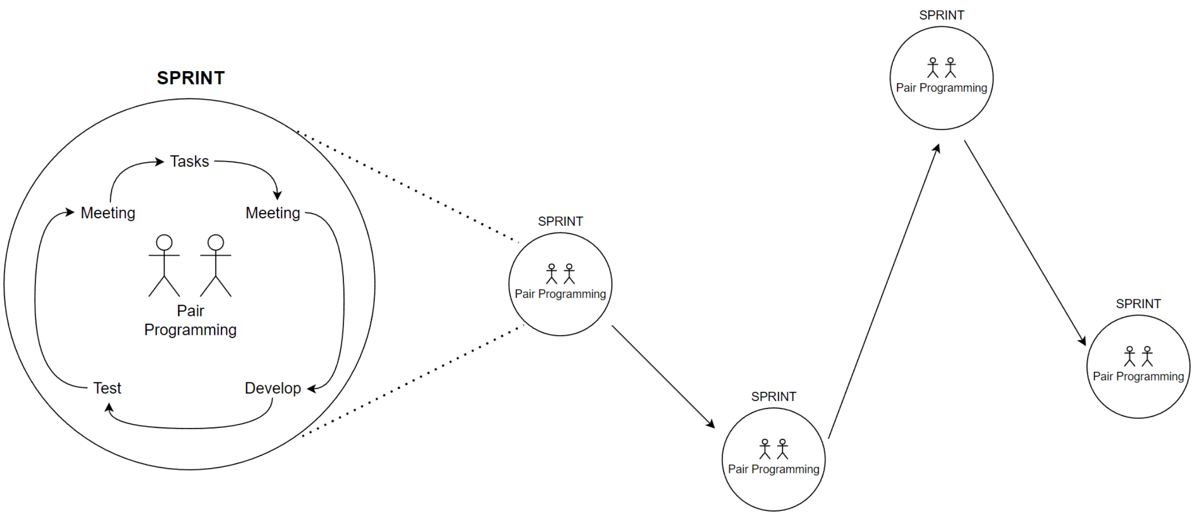
Struct is a library converting C/C++ structs to Python values. Due to the configurations made in CCS, the C structs captured by the Shimer units must be converted to Python byte objects to read the incoming data. Struct uses Format Strings, which specifies the expected layout when packing and unpacking data. The Format String consists of Format Characters and “special characters”, which specify the type of data being packed/unpacked and the byte order, size, and alignment of the data. The data types retrieved from the Shimmer3 units must correspond to the Format Characters to get the right output.

3 Agile development methodology

Three different development methodologies have been used throughout the development process. The multidisciplinary aspect of the case requires knowledge from several fields, including engineering, computer science, signal processing, and physiology. Wayfaring (Dybvik et al., 2021; Steinert & Leifer, 2012) is used as a development method within engineering design, and Extreme Programming (XP) (Sommerville, 2011) within software development, and Scrum (Gonçalves, 2018; Schwaber, 1997) as a development management method. Together, the development process makes use of the multidisciplinary environment and is an excellent example of how software development crosses with product development in the context of engineering design.

Embracing an agile development methodology was necessary to handle unknown variables, continuously new requirements, and serendipitous events, which are inevitable within product development. The Wayfaring method suggests an open approach to engineering problems and presents a journey of probing ideas. The unknown unknowns are in Wayfaring handled in Probes, where each probe is a prototype where new ideas are developed and tested. The equivalent process in XP is called a Cycle, where small frequent releases and continuous integration are encouraged by following a process consisting of meetings, development, and testing. XP consists of multiple principles, which ensures that the requirements are met. These practices include pair programming, and continuous testing and integration of code, all of which we employed in our development process. The different approaches and principles are managed by Scrum, where the focus is on integrating the complex environment of the case. The Scrum framework is made up of Sprints, which are development iterations similar to the Probe in Wayfaring. An illustration of how all methodologies are incorporated is provided in "Figure 2 - An illustration of how all development methodologies are incorporated into the process".

Figure 2 – An illustration of how all development methodologies are incorporated into the process. Inspired by the illustration of the Wayfaring process (Gerstenberg et al., 2015; Steinert & Leifer, 2012)



4 Software architecture

The agile development of the implementation of the experiment setup resulted in multiple prototypes, where each Probe/Sprint increased the understanding of the Shimmer sensors in addition to improving the previous result. The development process provided a better understanding of the Shimmer units and how to integrate the proprietary hardware into the LSL network. The result is a functioning API for multimodal sensor experiments, utilizing the LSL distribution.

All hardware must output a data stream that is recognized by the LSL network to take advantage of the open-source distribution LSL. The software architecture of the development process is provided to show how the data is captured from the Shimmer units and transmitted to the LSL network.

4.1 Shimmer software architecture

PySerial and Struct must be used in a specific manner to collect appropriate data since the Shimmer devices are pre-programmed with the firmware LogAndStream, written in C.
The architecture of the code is modular, which facilitates further development and implementation. The key features of the scripts include how pySerial and Struct functions are combined to collect the appropriate data and are as follows:

- When running the scripts, the comport where the device is connected must be included.
- It is important that the pySerial function read() and struct function unpack() is used together. Read() collects the requested data from the sensor and unpack() converts the data from C and presents the data in Python.
- The pySerial function write() and struct function pack() must be used together. Write() sends a request to the sensor, which must be converted to C using the pack() function.
- For every call to the sensor, an acknowledgement response must be called with the wait_for_ack() function. The function ensures that the request is processed correctly by the sensor and that it is ready to proceed.

4.2 LSL software architecture

To send data from Shimmer to LSL, a new stream info object must be created. The necessary parameters which must be defined include name, content type, channel count, nominal sampling rate, channel format, and source id. The following is the correct sequence in which to implement the code for creating an LSL stream from a serial port connection:

1. Open a serial port.
2. Create an LSL outlet, by defining the core stream information.
3. Read the data from the serial port.
4. Parse the signal accordingly.
5. Push the signal into the LSL outlet.

5 Open-source contribution and discussion

By utilizing LSL and enabling proprietary hardware to stream data to open-source software distributions, the development process demonstrates how to create a low-cost, multimodal physiological sensor experiment setup. New research opportunities are offered by switching from proprietary software. Higher quality, greater reliability, greater flexibility, lower cost, and the end of predatory vendor lock-in are among the promises made by open source (Open Source

Initiative, n.d.-a). Utilizing LSL broadens the researcher's options by removing the financial constraint of leasing proprietary software. LSL provides several tools and resources on how to incorporate various sensors and customize the software to your need. The platform is already compatible with a wide range of sensor software, allowing data streams from numerous sensors to be synchronized. LSL is widely used in research and has an established community.

5.1 Advantages and drawbacks of open source

There are significant benefits to sharing research and source code. First and foremost, it promotes low-cost collaboration and innovation. Transparency and open access to information and materials allow researchers to build on each other's discoveries and ideas, which facilitates the acquisition of new knowledge. Thus, transparency promotes faster development and improvements. Another advantage is the community connected to the different projects. Collaborations, derivations, and discussions all contribute in unexpected ways to improving others' work. Approaching “the open source way” encourages everyone to accept failure as a means of improvement, and expects everyone else to do the same (Opensource.com, 2022). Forums, blog posts, papers, and journals all contribute to the ongoing growth and maintenance of projects.

However, if a project does not acquire sufficient support and has difficulty building a community, it will quickly fail. One of the principles of “the open-source way” is the idea of meritocracy. Diverse perspectives are essential to identifying the best ideas, and the most successful projects are those that receive community support and effort, even if they are not reached by consensus (Karsten, 2020). A project's survival is based on the motivation and interest in maintaining the project. In the absence of a community, or if the original contributor no longer maintains the project, it will become obsolete, if not deprecated.

When creating proprietary software or conducting research, the creation of a community is not as important as it is for open-source developers. One of the most obvious benefits of proprietary software is that it is maintained and improved by employees who are completely committed to the project. Subscription fees contribute to covering the costs of maintaining the project. Some companies also utilize the advantage to disclose some of their research or software while keeping some of the functionality confidential. In that way, they are still in control of the development in addition to keeping the economic benefits (Vanhaverbeke et al., 2008).

5.2 Adapting proprietary hardware to open-source software

Our development process evoked several questions worth discussing. Are there nuances as to what is considered open source? How many resources should a researcher use to reproduce and contribute to further development? Can Shimmer be considered open source? It is important to clarify that this is not meant as a critique against Shimmer, but rather as an evaluation of how companies take advantage of open-source properties without fully committing to open research.

Understanding the firmware of the Shimmer sensors took a lot of time and effort. Despite offering a large amount of documentation and user manuals to the customer, the information given is difficult to understand and interpret. In addition, the documentation offered by Shimmer is poorly constructed. We had to search through several user manuals and source code scripts to find associated information to fill the information gap. Shimmer makes configuring the various sensor units extremely difficult without their Consensys software tool. They do, however, claim to be "committed to transparency in explaining how our systems work, what they can and cannot do, and why, and providing and actively maintaining a huge body of product documentation", but simultaneously clarify that their support does not include detailed explanations of the engineering principles behind their software and hardware (Shimmer, n.d.).

Researchers need to know that their sensory data is correct, and Shimmer being selective in their use of transparency is an issue. The firmware settings are crucial to know as it is the foundation of what output is presented. Open-source software can be considered to be more secure and stable than proprietary software. Given the fact that open-source software can be viewed and modified by anyone, allows anyone to detect and correct errors or omissions that were missed by the original authors (Heron et al., 2013). The Shimmer units consist of hardware developed by Texas Instruments, which again is configured through their integrated development environment. It is difficult to determine what the original raw data are because various firmware handles data differently, and the corresponding software program used to show the data takes liberties and alters the output to present it in an orderly manner. There are many steps along the way where information on how to manipulate the raw data can be hidden. Due to the binary format of the firmware source code and the lack of documentation explaining the configuration, it is complicated to figure out the settings. The sensor data's integrity determines the credibility. Quality assurance provided by the firm is one of the key arguments for keeping firmware and software products proprietary. However, any solution may turn out to be just as reassuring as any proprietary solution with proper documentation and accessible source code.

Companies, such as Shimmer, benefit from open research by providing numerous scripts and documentation to any end-user by encouraging technological development without compensating for it. The advantage of the strategy is that companies, such as Shimmer, learn early on about new technologies. As a result, companies can scan a broader range of technologies or new market developments, rather than writing options on internal projects alone (Vanhaverbeke et al., 2008). They retain control and economic benefits by keeping the firmware settings to themselves. The business model appears to supply enough content to encourage the customer's interest in contributing to further development before they reach the level of confidentiality.

The answer to whether Shimmer is open source or not, is not definitive. However, they do embrace open innovation by publishing large amounts of documentation and harvesting the benefits that come with it. The development process uncover that the sensor units were decodable and configurable. Given the vast amount of documentation, we are confident that the data acquired are accurate data. However, one question remains unanswered: to reproduce and contribute to further development, how many resources should a researcher use? On one side, the public has access to a lot of useful information about how to handle and interpret the units' output. The question is whether the resources available, such as time and knowledge, are sufficient. On the other side, one could argue that the process is only limited to oneself. The information is available, but do you have the time and knowledge to comprehend and understand it? During the development process, the process of collecting information was time-consuming. The documentation was available, and anyone could decode and interpret the data eventually. Consequently, if you are motivated enough, it is only a matter of willpower to persevere until all questions are addressed. However, if this is the mindset to reproduce and contribute to development, all resources will run out. We assert that the relationship between resources invested, and the outcome of the final result should be fairly balanced. Within the development process described in this article, this was not.

6 Conclusion

This article adds to the field of open research by broadening the possibilities for experimental setups facilitate the acquisition of new knowledge. With the primary goal of demonstrating the benefits of open-source and how it may promote open research, we have shown how open platforms, such as LSL, may be used to save costs by eliminating the cost of leasing proprietary software.

The development case opens more research opportunities and contributes to the open research field. The accessibility provided by the case should encourage researchers to take advantage of the open-source distribution of LSL. Given that all implementation from the development process is open-source, other researchers can build upon and modify it, creating the opportunity to customize it to other purposes. Even though there is a lack of incentives to be transparent (Nosek et al., 2015), it contributes to more reliable research, and better quality, due to peer review.

Code availability

The code retrieving GSR and ECG data and sending it to the LSL network is publicly available: https://github.com/catthiba/Multimodal-sensor-setup

References

Balters, S., & Steinert, M. (2017). Capturing emotion reactivity through physiology measurement as a foundation for affective engineering in engineering design science and engineering practices. *Journal of Intelligent Manufacturing*, *28*(7), 1585–1607. https://doi.org/10.1007/s10845-015-1145-2

Blessing, L. T. M., & Chakrabarti, A. (Eds.). (2009). DRM: A Design Reseach Methodology. In *DRM, a Design Research Methodology* (pp. 13–42). Springer. https://doi.org/10.1007/978-1-84882-587-1_2

Dybvik, H., Kuster Erichsen, C., & Steinert, M. (2021). Description of a Wearable Electroencephalography + Functional Near-Infrared Spectroscopy (EEG+FNIRS) for In-Situ Experiments on Design Cognition. *Proceedings of the Design Society*, *1*, 943–952. https://doi.org/10.1017/pds.2021.94

Extensible Data Format (XDF). (2021). [Python]. Swartz Center for Computational Neuroscience. https://github.com/sccn/xdf (Original work published 2015)

Gerstenberg, A., Sjöman, H., Reime, T., Abrahamsson, P., & Steinert, M. (2015). A Simultaneous, Multidisciplinary Development and Design Journey – Reflections on Prototyping. In K. Chorianopoulos, M. Divitini, J. Baalsrud Hauge, L. Jaccheri, & R. Malaka (Eds.), *Entertainment Computing—ICEC 2015* (Vol. 9353, pp. 409–416). Springer International Publishing. https://doi.org/10.1007/978-3-319-24589-8_33

Gonçalves, L. (2018). Scrum. *Controlling & Management Review*, *62*(4), 40–42. https://doi.org/10.1007/s12176-018-0020-3

Heron, M., Hanson, V. L., & Ricketts, I. (2013). Open source and accessibility: Advantages and limitations. *Journal of Interaction Science*, *1*(1), 2. https://doi.org/10.1186/2194-0827-1-2

Karsten, W. (2020, December 16). *The open source way 2.0—Presenting the Open Source way*. Opensource.Com. https://www.theopensourceway.org/the_open_source_way-guidebook-2.0.html

Kothe, C., Medine, D., Boulay, C., Grivich, M., & Stenner, T. (2019). *Quick Start—Labstreaminglayer*. https://labstreaminglayer.readthedocs.io/info/getting_started.html

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D. P., Hesse, B., Humphreys, M., … Yarkoni, T. (2015). Promoting an open research culture. *Science*, *348*(6242), 1422–1425. https://doi.org/10.1126/science.aab2374

Open Source Initiative. (n.d.-a). *About the Open Source Initiative*. Retrieved 15 February 2022, from https://opensource.org/about

Open Source Initiative. (n.d.-b). *The Open Source Definition*. Retrieved 1 December 2021, from https://opensource.org/docs/definition.php

Opensource.com. (2022). *What is open source?* Opensource.Com. https://opensource.com/resources/what-open-source

Pinti, P., Tachtsidis, I., Hamilton, A., Hirsch, J., Aichelburg, C., Gilbert, S., & Burgess, P. W. (2020). The present and future use of functional near-infrared spectroscopy (fNIRS) for cognitive neuroscience. *Annals of the New York Academy of Sciences*, *1464*(1), 5–29. https://doi.org/10.1111/nyas.13948

Schwaber, K. (1997). SCRUM Development Process. In J. Sutherland, C. Casanave, J. Miller, P. Patel, & G. Hollowell (Eds.), *Business Object Design and Implementation* (pp. 117–134). Springer London. https://doi.org/10.1007/978-1-4471-0947-1_11

Shimmer. (n.d.). Contact Support. *Shimmer Wearable Sensor Technology*. Retrieved 1 December 2021, from https://shimmersensing.com/support/wearable-sensing-support/

Shimmer Research. (n.d.-a). *GSR User Guide rev 1.13*.

Shimmer Research. (n.d.-b). *Shimmer Wearable Sensor Technology*. Shimmer Wearable Sensor Technology. Retrieved 15 February 2022, from https://shimmersensing.com/

Shimmer Research. (2017). *Shimmer User Manual*.

Shimmer Research. (2018). *LogAndStream for Shimmer3 Firmware User Manual*.

Sommerville, I. (2011). *Software engineering* (9th ed). Pearson.

Steinert, M., & Leifer, L. J. (2012). ‘Finding One’s Way’: Re-Discovering a Hunter-Gatherer Model based on Wayfaring. *International Journal of Engineering Education*, *28*(2), 251.

Texas Instruments. (n.d.-a). *ADS1292R data sheet, product information and support | TI.com*. Retrieved 14 February 2022, from https://www.ti.com/product/ADS1292R

Texas Instruments. (n.d.-b). *CCSTUDIO-C2000 IDE, configuration, compiler or debugger*. Retrieved 21 November 2021, from https://www.ti.com/tool/CCSTUDIO-C2000

Texas Instruments. (n.d.-c). *MSP430F5437A data sheet, product information and support*. Retrieved 21 November 2021, from https://www.ti.com/product/MSP430F5437A#features

Vanhaverbeke, W., Van de Vrande, V., & Chesbrough, H. (2008). Understanding the Advantages of Open Innovation Practices in Corporate Venturing in Terms of Real Options. *Creativity and Innovation Management*, *17*(4), 251–258. https://doi.org/10.1111/j.1467-8691.2008.00499.x