

Article

A Fully Open Source Remote Laboratory for Practical Learning

Bastien Letowski ^{1,*}, Camille Lavayssière ², Benoît Larroque ¹, Martin Schröder ³ and Franck Luthon ²

¹ SIAME, Université de Pau et des Pays de l'Adour, E2S UPPA, 64000 Pau, France; benoit.larroque@univ-pau.fr

² LIUPPA, Université de Pau et des Pays de l'Adour, E2S UPPA, 64600 Anglet, France; camille.lavayssiere@iutbayonne.univ-pau.fr (C.L.); franck.luthon@iutbayonne.univ-pau.fr (F.L.)

³ Institut für Energietechnik, Technische Universität Berlin, 10587 Berlin, Germany; m.schroeder@tu-berlin.de

* Correspondence: bastien.letowski@univ-pau.fr

Received: 11 September 2020; Accepted: 28 October 2020; Published: 3 November 2020



Abstract: Nowadays, communication and web services are part of our lives, including education, and e-learning applications are used more and more. However, teaching in engineering sciences requires an important amount of practical experiments. Hence, remote laboratories are an attractive solution, offering new tools for both teachers and learners. Our objective is to propose a fully open source remote laboratory for generic practice, in all fields of engineering education or applied sciences. This work is based on an open source Supervisory Control and a Data Acquisition platform programmed with Python. Interoperability is one of the main issues considered here to ensure a wide compatibility with multi-communication protocols and multi-instruments used in practical labwork.

Keywords: e-learning; industrial engineering; lab work; student motivation; online education; virtual instrument; distance laboratory; SCADA; network; Raspberry Pi

1. Introduction

One of the main goals of remote laboratories (e-labs) is to extend the possibilities of practical experimentation in scientific academic degrees. Moreover, post-secondary student profiles are diverse but well suited to new Information Technology. Another important point is related to student cohort massification in higher education [1,2] that induces the need to carry out pedagogical innovation [3]. To answer these issues, new trends in engineering education deal with distance education to complete traditional face-to-face education [4]. Remote laboratory has been used for more than twenty years now [5] and is still massively used for learning technologies, as shown by numerous publications these last years. These remote laboratories led to many benefits for both teacher and student: all-time accessibility, enhanced learning processes, practical example in classroom lecture, etc [6]. Thus, they are well adapted to flexible student's digital environment of today. Nevertheless, the remote laboratory is much more difficult to implement than a virtual laboratory that is often associated with remote laboratory. Virtual laboratory is only a computer based model for simulations, which is cheaper and easier to implement as well as to share [7]. Accordingly, developments to improve standardization, simplicity, and cost of remote laboratory are important research issues.

The state of the art is the focus for two main topics: open source remote control laboratory and open source Supervisory Control And Data Acquisition (SCADA). As remote laboratory is mainly developed by university or training center, the open source and publication aspects are automatically considered, for instance, [8] or [9] and more recently [10,11]. In [12], a remote laboratory for control education is implemented using common communication and programming languages: HTML 5,

JavaScript for client side, JSON for data transfer, and PHP/MysQL for the server side. Everything is open source designed, even the hardware part with Arduino boards. The project SEITI RMLab is programmed with the same logic (HTML5, JSON, etc.) running an electromechanical practical work [13]. Another open source project is presented in [14] where a solar tracking system using Raspberry Pi and open source hardware cards are used to obtain a low cost remote system. The same solutions are used in a remote smart grid demonstrator [15]. In the other way, solution spreading to collaborators is common. Some projects have already validated their solution like the UNED network [7] or the PEMCWebLab [16] and more recently Lab@Home [17] to only cite some examples.

On the other hand, our project takes inspiration from the SCADA system and particularly in open source SCADA. Proprietary software is expensive and there is a problem of interoperability with infrastructures or devices. The open source SCADA platform, Thingener.IO, designed for Internet of Things (IoT), is implemented in [15,18]; however, this solution is not entirely free, even if the cost is acceptable. In fact, low or medium size industries face the same issues than engineering education centers: for example, ref. [19] works on Object Linking and Embedding (OLE) for process control (OPC) with a Python language. The work of [20] demonstrates an important effort to work on sensors/actuators with open source software and hardware, and they use the free tool Easy Java and JavaScript Simulations (EJS) to carry out their experiment.

To introduce the intersection of remote laboratory and SCADA system literature, some smart grid applications have been considered because these projects intermix SCADA architecture and energy monitoring on the education laboratory. These applications have an important need for control and supervision of power generation and use. The Smart Electric Power System (SEPS) Laboratory [21] is an example; however, it is programmed with LabVIEW, namely a proprietary solution. In the same manner, a remote laboratory based on SCADA architecture has been proposed by [22] using Programmable Logic Controller (PLC) to manage the laboratory and a Visual Studio with an ASP.NET solution for the web interface.

Otherwise, ref. [23] have demonstrated a low cost SCADA system with open source hardware (Arduino) and common programming language (HTML5, JavaScript, and PHP) on a microgrid test bench for academic and research applications. Similarly to open source SCADA, interoperability issue has been studied in remote laboratories to carry out standardization on different communication protocols, programming software, and data exchange [24].

In relation to the state of the art, it appears that there is not a real open source system (entirely open) and all developments are specifically dedicated for one system or experimental laboratory. Most of the time, proprietary software is used (e.g., Matlab or LabVIEW). Some authors have already addressed low cost and open source solution, but, to the best of our knowledge, our work is one of the first projects to work on fully open source and modular solution to offer a wide interoperability. In this logic, software (server and controls) and hardware solution have been designed to ensure interoperability with hardware on the server side and with software on both the server side and client side.

This article is organized as follows: Section 2.1 describes our solution for remote laboratory where the architecture overview is presented. Then, a description of the open source programming solutions of the remote laboratory in reference to the SCADA system and communications is provided. Hardware elements of our solution are explained. Section 3.1 briefly describes two kinds of practical laboratory work that can be implemented in the plugin concept and illustrates our approach with an application to electronics labwork. Finally, key results are summarized and future studies are introduced.

2. Materials and Methods

2.1. System Description

The project, Laborem, has evolved since 2017 from a LabVIEW based solution [25] to an open source solution [26–28].

Laborem concerns both learners and teachers and has multiple applications in education as graphically summarized in Figure 1. Remote laboratory can be a solution for learners with motor or auditory disabilities or learners in continuing education, and it completes and enhances online courses with laboratories. To offer access to Laborem to the public with visual disabilities, we will plan to offer Braille support and large font size option for the web HMI. Dropping out students might prefer not to work in a common class environment. More precisely, Laborem is used with undergraduate students which are very receptive to distant control and a game-like scenario. For teachers, Laborem is implemented both for entire laboratory courses, or to prepare practical work before face-to-face practical work. It represents a new educational tool to mix practice with theory, as well as to offer a multi support enhanced learning process.

Another important point is to provide simplified access to the laboratory, independently of opening hours, and, in particular, each student can work at his own pace. Only one test bench is necessary to perform a wide range of practical works within one course; this represents a low-cost solution and the opportunity to share with other educational centers.

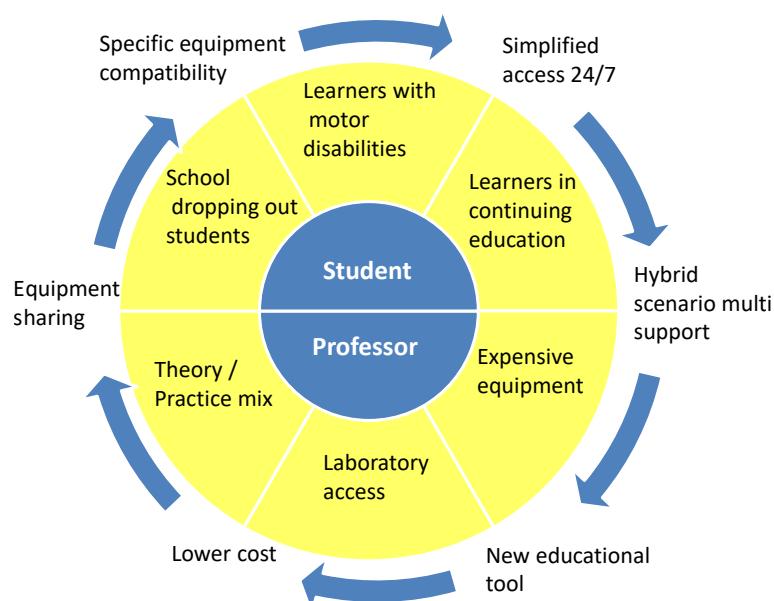


Figure 1. Laborem applications in education.

The initial prototype of Laborem used a technically advanced solution running on the Windows Server operating system. The proprietary LabVIEW software from National Instruments was used to program and to create web remote access, which allows for control benchmark equipment and to exploit data results. The software cost, the powerful server needed, and the nonstandard protocol used by LabVIEW to communicate data through internet led us to develop an entirely new system to migrate to an open source and light software.

Table 1 compares the original Laborem version with the current version, highlighting the differences and evolution. To summarize our solution, firstly, the cost of the solution is reduced by utilizing open source software, a low cost server (Raspberry Pi) and developing our acquisition card. Secondly, the interface compatibility is improved employing standard web protocols while good hardware compatibility is kept thanks to open source and widely used libraries. With this solution,

new perspectives were opened such as integration with a Learning Management System (LMS) and creation of an e-lab network to share practical works between universities.

Table 1. Comparison between former Laborem and the current solution.

Issues	Laborem 2017	Laborem 2019
Cost	Expensive: LabVIEW, powerful server, NI data acquisition system	Cheap: Raspberry Pi as a server, Python programming, free drivers
Pedagogical efficiency	High: used with student (tests since 2012) 40 students each year	Possibly higher: better integration with Moodle, not yet tested
Availability and Access	Not easy, nonstandard web communication	Full compatibility with web browser (HTML, JavaScript)
Various equipment compatibility	Excellent, based on NI Database (proprietary)	Good with PyScada and VISA protocols (open source)
E-lab network	None	Under construction, network with Burkina Faso, Germany, Spain

The Laborem architecture is centered on the server, here a Raspberry Pi which hosts the web Human Machine Interface (HMI) accessible by a student. It communicates with the motherboard to select which circuit will be tested. A robotic arm is used for circuit modification, two cameras mimic student's eyes to see all the test bench equipment and ongoing measurements. The overview is presented in Figure 2.

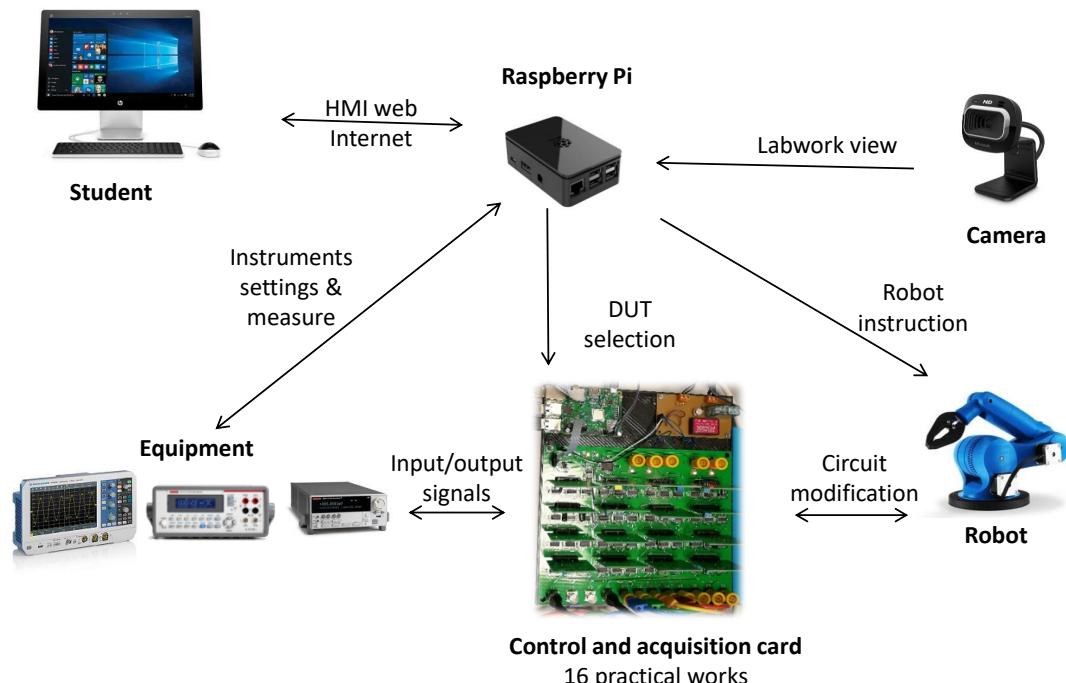


Figure 2. Overview of Laborem system architecture (DUT means Device Under Test).

2.2. Software Architecture

2.2.1. Programming Language

The core software is PyScada, a free and open source SCADA application written in Python with a web HMI [29].

The high-level free and open source Python framework used is Django, which allows for creating database-driven web applications. Algorithms, settings, and data models are written in Python. The Django framework core can be seen as a Model-View-Controller (MVC) architecture. As described in Figure 3, this Python based MVC architecture uses an Object-Relational Mapper (ORM) to manage the data model and interact with the database, an URL dispatcher to receive the HTTP requests, and a web template system to generate dynamic HTML5 pages. Alongside HTML5, the application uses JavaScript and CSS (Cascading Style Sheets) to enable interactive web pages.

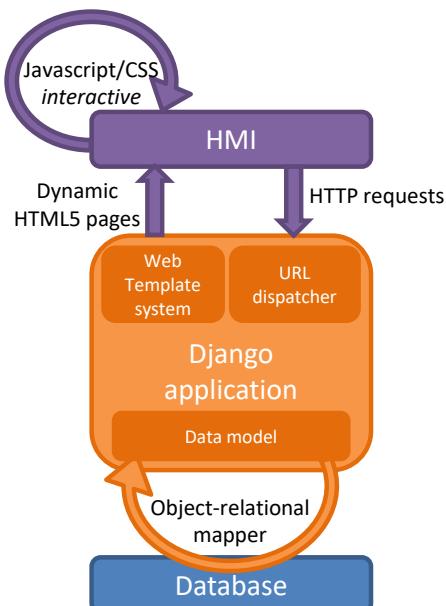


Figure 3. Django Model-View-Controller.

The SCADA system with HTML5 HMI allows students to interact with various instruments and view the data on any recent web browser, independently of the operating system of the final user.

In PyScada, all the processes are working asynchronously and communicate through the database. It means that, on one side, the HMI queries the values to display and writes the user actions to the database. On the other side, scripts that send and request data to the equipment also query the database to know the user action. Alongside these, scripts feed the database with instrument answers relative to the original request. These processes are managed by a master process named the scheduler.

The Raspberry Pi 3 [30], which is a single board computer (SBC), replaces the previous personal computer (PC) that was used as server. It is an all-in-one, low-cost hardware solution, shared by a large community, so it is an interesting and widely chosen solution, for example [8,12,19,31] use a Raspberry, whereas [7] uses a BeagleBone Black board.

2.2.2. Communication Protocols

In order to have interoperability with many instruments, the system has to be compatible with several communication protocols (VISA, Modbus, etc.) and several hardware interfaces (USB, Serial, Ethernet, etc.). PyScada offers a wide range of protocols for SCADA systems based on the wide Python open source community and its useful applications such as PyUSB, PySerial, PyVISA, and PyModbus. All the protocols implemented and used are graphically illustrated in Figure 4.

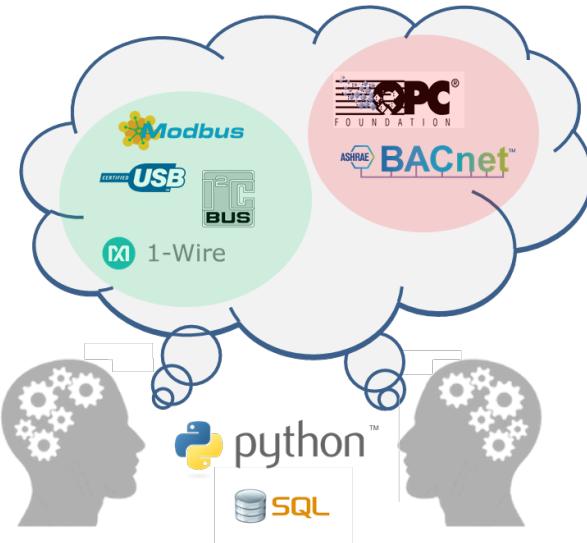


Figure 4. Protocols currently used by Laborem in a green circle and future work in a red circle.

The developed solution is a multi-protocol and multi-instrument system. It works like a bank of instruments where one can easily choose the instruments connected to the Raspberry Pi independently of the hardware interface used. For instance, one can switch between two oscilloscopes, HP 54603B and Tektronix MDO3014, using GPIB and USB connection, respectively, where each instrument has a dedicated handler file. Each handler has the same functions which are adapted to capabilities of each model and brand specific SCPI instructions (Standard Commands for Programmable Instruments). For example, the Tektronix MDO3014 is able to give directly the phase between two signals, whereas, for the HP 54603B, it is necessary to digitize these two signals and therefore use a NumPy function in order to compute the phase.

2.3. Modular Hardware

The hardware setup on the server side is shown in Figure 5.

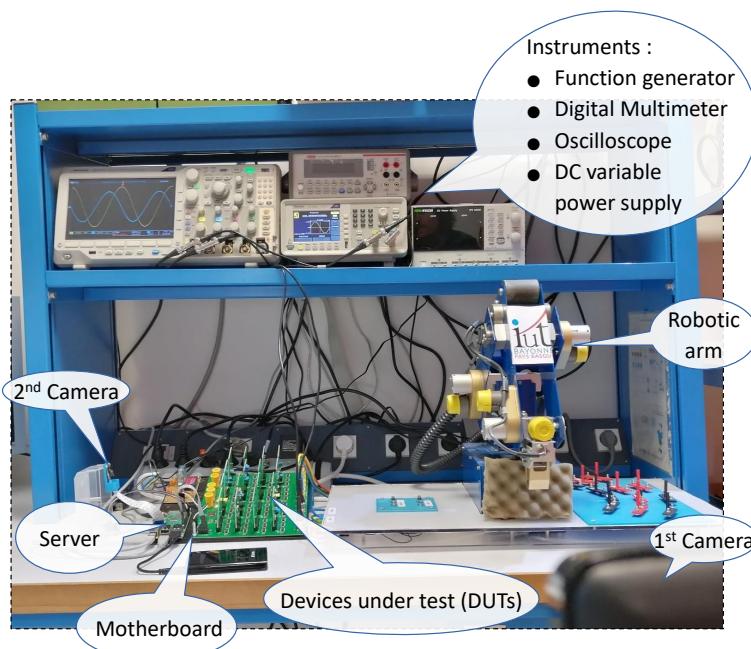


Figure 5. Hardware resources on server side: instruments, motherboard with connected plugs (DUTs), robot, camera.

2.3.1. Modular Motherboard

The motherboard interconnects laboratory devices and practical workbench, such as device under test (DUT). It receives and manages all experiment information. Each plug is a DUT and can be easily replaced by the teacher to adapt the scenario to the learner's profile. The learners will have access to 16 different DUT to achieve their work. For example, DUT in electronics can be a low-pass filter, a high-pass filter, a band-pass filter, an active Sallen-Key filter, a Wien bridge, etc. A multiplexer enables the plug selection according to server request. A specific power card supplies the energy required for the motherboard and the plugs from single phase 230–110 Vac to several Direct Current (DC) power supplies (+15 V, -15 V, 5 V and 3.3 V). The block diagram of the motherboard is described in Figure 6.

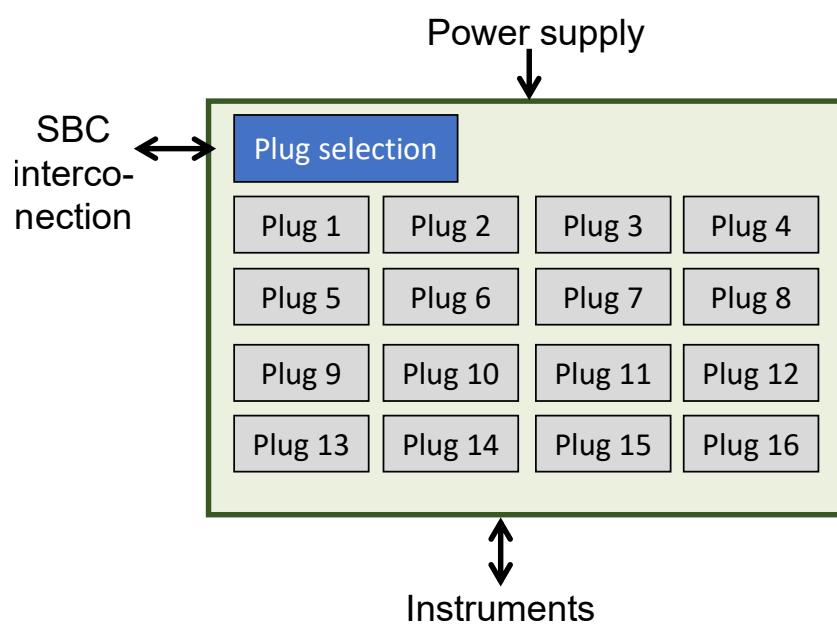


Figure 6. Block diagram of the modular motherboard.

The Printed Circuit Board (PCB) was firstly designed with Multisim and Ultiboard (National Instruments proprietary software), but new ongoing projects are made with KiCad (open source software). All of the PCB designs are open source. Connectors and plugs are designed in PCI Express logic. This is a well-known electronic standard, it simplifies PCB design, and reduces cost because it only uses one connector (female). In addition, this solution is widely used and has a good mechanical resistance. This choice ensures hardware interoperability with a simple connector that fits all applications and robustness for a long life utilization.

The motherboard is designed to limit human needs. The teacher needs to set up the configuration of the practical workbench at the beginning of a new course and to select the 16 different plugs that can be used by students. The only specification is that all plugs need to have the same Input/Output (I/O) configuration to work concurrently. Then, the I/O configuration is specified in the Laborem administration interface by the teacher, selecting for each of the 12 I/O ports the type of the connection. It could be, for example: a channel for oscilloscope, another for the function generator and also General Purpose Inputs/Outputs (GPIO) to interact with the Raspberry. The teacher can save various I/O configurations that match different groups of plugs to easily reuse them in a future utilization.

2.3.2. Generic Plugs

Plugs are designed for all kinds of applications. The DC power supplies from the motherboard are available inside plugs to be fully compatible with TTL or CMOS logics in digital electronics (+5 V and +3.3 V, respectively). The 15 V symmetric supply is used for analog signals that can be positive and negative and typically used with Operational Amplifiers. Moreover, the 15 V can be used for a local power supply if more power is required, for instance to locally heat a temperature sensor. Finally, 12 General Purpose Inputs/Outputs (GPIO), both analog and digital, are available and can be implemented to build experimental workbench or circuit. The plug schematics is described in Figure 7.

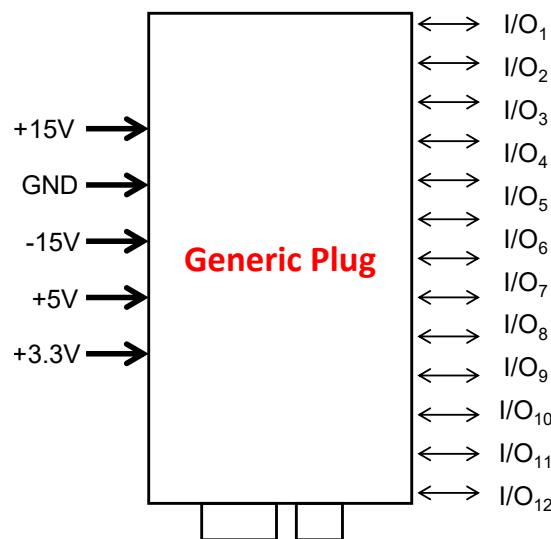


Figure 7. Generic plug schematic with 12 GPIO.

Similarly to the motherboard, plugs are designed to minimize physical human actions; however, it is important to remote control devices within plugs. Thus, these controlled devices mimic student actions to remotely modify the DUT, to activate/deactivate part of an electronic circuit, to adjust a load or to vary a motor speed, etc. For instance, a practical work using a Wheatstone bridge has been implemented with digital potentiometer as already presented at [28]; in this case, the plug is designed as follows:

- Analog:
 - input signal (waveform generator)
 - output signal (oscilloscope)
 - electrical component wired between 2 I/O (resistance, capacitor)
- Digital:
 - I2C bus command for digital potentiometer
 - logic signal (low/high) to control switches.

To conclude this section about the hardware, all elements are stored in a “Laborem Box”, an open source plastic box designed with Solidworks and realized by a 3D printer in our fablab at the University of Pau, Anglet, France. Then, pictures of the all-in-one Laborem box are presented in Figure 8. The approximate size is 20 cm (width), 25 cm (length), and 10 cm (thickness).

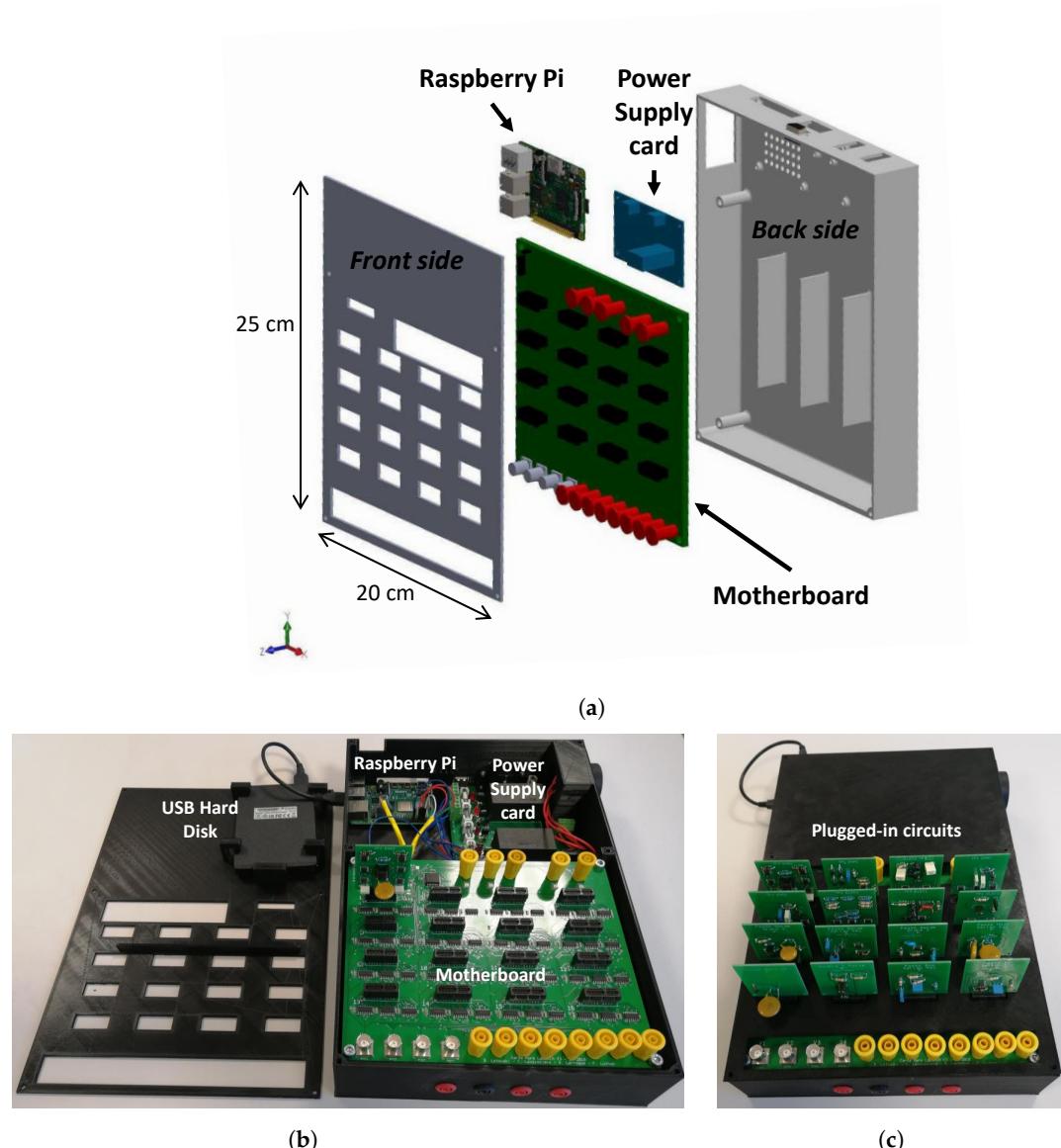


Figure 8. Illustrations of the “Laborem Box”: (a) 3D exploded view; (b) opened box with all elements visible; (c) the box with 16 plugs connected.

3. Results

3.1. Category of Labwork Applications

3.1.1. Small Signals

In Laborem architecture, plugs are sorted into two categories: the first one is related to small signals where everything is on the plug: circuits, DUT, and remote controlled components. Practical experimentation takes place within plugs. This family concerns, for instance, the following topics: basic electronics, instrumentation, industrial data processing such as microcontroller application, automation, etc. These experimentations must be supplied by the “low” power supplies available (+/−15 V maximum as mentioned above) and with a current limited to 1 Ampere.

3.1.2. Large or External Signals

Some other applications have specific prerequisites; this second plug category corresponds to a “large signal” relative to higher power (several Amperes or hundreds of Volt) and “external signal” for applied science laboratories. These applications need conditioning circuits like voltage level adaptation

for interoperability between DUT and control/acquisition signal (digital or analog). This specific requirement is common and has already been addressed in [20].

Power applications use high voltage and high current, so measurements are more complicated. For instance, voltage can be measured thanks to a potential divider (capacitive or resistive divider), transformer, etc. The current is usually measured with a current transformer, Hall effect sensor, shunt resistor, or current probe. The conditioning circuit will be embedded in a specific plug.

In the same manner, other applied science experiments need a conditioning board too, as mechanical laboratory work needs torque, speed, pressure sensors, etc. In that case, plugs are linked with the motherboard and the SBC through GPIO and serial bus.

In that case, local Data Acquisition (DAQ) circuits can be implemented and will communicate directly with the server, without using instruments like digital multimeter or oscilloscopes. This operating mode is close to other proposals in the literature, such as [7,10–12,23].

3.2. Applications to Electronics Labwork

3.2.1. Implementation for an Electronics Filtering Course

Laborem has been dedicated to electronic filtering courses, and the first practical work available is therefore about this subject. The Laborem client's side view is illustrated in Figure 9 where students can select which circuit they are going to experiment.

The screenshot shows the Laborem interface. At the top, there is a header with the logo of the Université de Lorraine, the text "e2s PyScada-Laborem", and navigation buttons "Précédent" and "Suivant". Below the header, a message says "Sélectionnez un montage à étudier et appuyez sur > Suivant". A note below it states: "Si le montage affiche Robot vous pourrez configurer le montage à l'écran suivant." On the left, there is a schematic diagram of a band-pass filter circuit with components R1 (5.6kΩ), C1 (2.2nF), AOP (operational amplifier), R2 (68kΩ), C2 (2.2nF), C3 (2.2nF), and R3 (2.7kΩ). Below the schematic is a photograph of a blue metal frame containing various electronic components and a camera mounted on a robotic arm. To the right of the schematic is a list titled "Liste des montages" with 16 entries, each with a status indicator (Robot, Débutant, Intermédiaire, Avancé). The entries are:

Passe Bas	Débutant
Passe Haut	Débutant
Passe Bande	Intermédiaire
Double T Ponté	Intermédiaire
Passe Bande Ordre 2	Avancé
Passe Bas Sallen-Key	Avancé
Passe Haut Sallen-Key	Avancé
Passe Bande Sallen-Key	Avancé
Passe Haut Robot	Robot Débutant
Passe Bas Robot	Robot Débutant
Passe Bande Robot	Robot Intermédiaire
Double T Ponté Robot	Robot Intermédiaire
Passe Bande Ordre 2 Robot	Robot Avancé
Passe Bas Sallen-Key Robot	Robot Avancé
Passe Haut Sallen-Key Robot	Robot Avancé

Figure 9. Laborem main page with filter selection (in French).

For instance, measurements have been performed on a band-pass active filter among one of the 16 available plugs. The frequency analysis (Bode Diagram) is illustrated in Figure 10, and Figure 11 is the PCB plug picture of the filter.

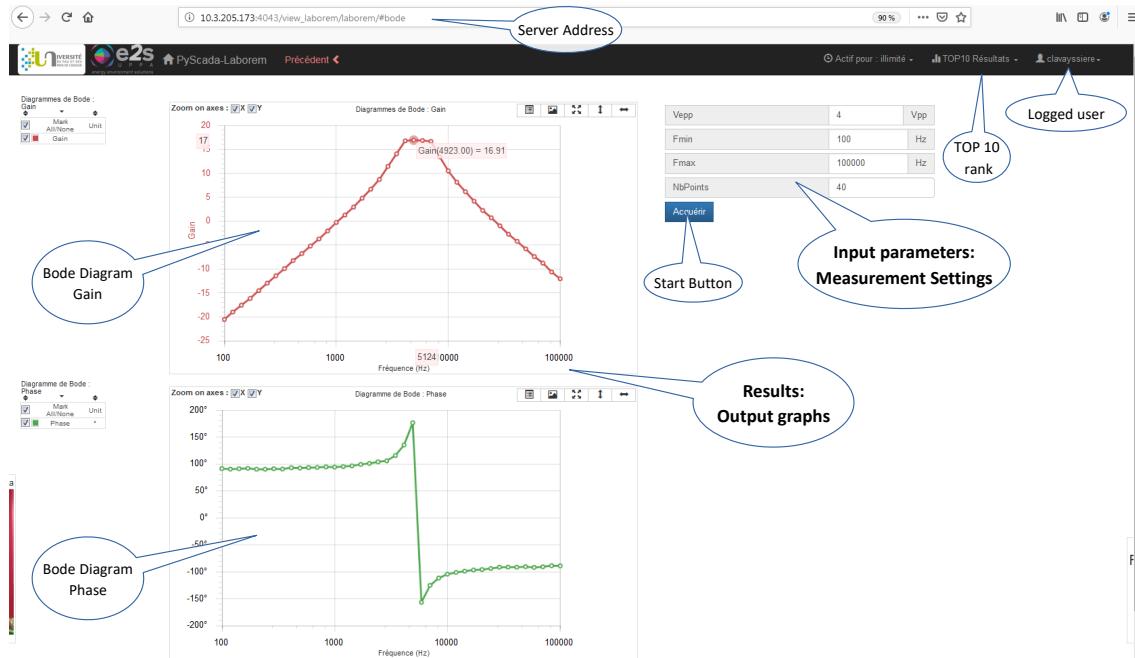


Figure 10. Bode diagram of a second order band pass active filter.



Figure 11. Picture of the tested plug.

3.2.2. Pedagogical Scenario

The pedagogical scenario is based on a game-like approach. It includes a treasure hunt and a Top10 of the best measurements. The typical scenario contains introduction lessons (pre-requisite and new theory), quizzes, remote experiments, and a final exam to evaluate knowledge. Like in e-games, the four basic concepts are: level (beginner, medium, advanced), number of lives given (equivalent to number of allowed repetitions), score, or mark obtained for an activity and time spent in the activity. Moreover, an adaptive scenario is implemented to improve learner immersion and motivation in the student learning process as described in Figure 12.

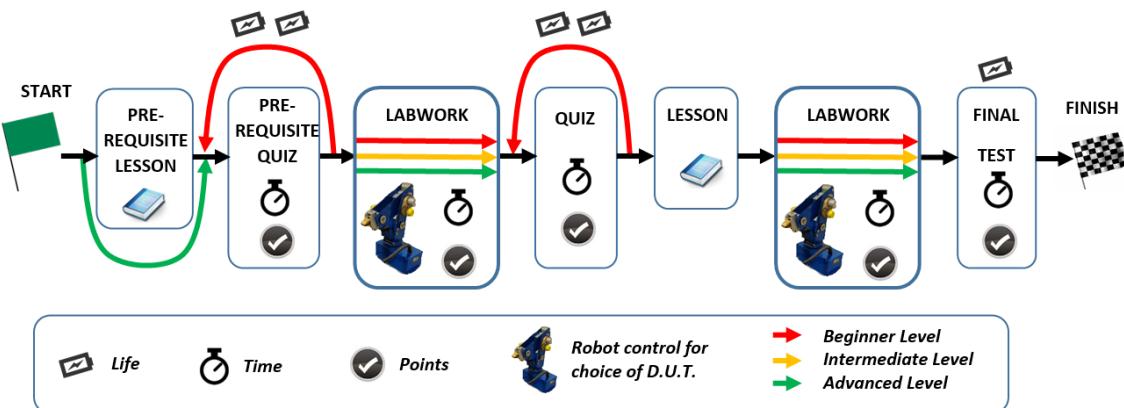


Figure 12. Game-like learning map of Laborem including treasure hunt and student's level adaptive route.

The LMS plans different paths depending on student's results during the sequence of the scenario, allowing teachers to modify the route. For instance, Top 10 activity corresponds to questions about the measured value of typical parameters or specific points. Depending on the correctness of the answer, typically its precision: 5%, 20%, or out of interval, the learner will get more or less points which will be updated in the Top 10. These questions can also be used for partial or quick assessment in preparation to face-to-face experimental laboratory.

The Laborem platform is accessible through any recent web browser. During the development phase, access is limited to the users of Université de Pau. They have easy access to the remote laboratory with the server IP address. The platform authentication is validated by a Django credential or by the university Central Authentication Service (CAS). With the integration of the Laborem platform in a remote laboratory network, access will be opened to other universities. Each student connected to the web page is ranked in a waiting list. The oldest user (known as the "worker") in the list is credited with 5 min to work on the Laborem platform before to let the next user the access to manipulate. If there are other users waiting to access the platform (known as the "viewers"), the "worker" has limited time to use the platform. The "viewers" can see "worker" actions, their ranks in the waiting queue, and the waiting time left. After studying a device, student can answer questions and climb in the user Top 10 ranking. Teachers can have special rights such as moving any other user to the bottom of the waiting list, changing the list of available devices or the list of experiences that students can access, adding or removing questions and answers, adjusting working duration, etc. Usability improvements are done yearly using the results of a survey filled by students.

In the former version, students had the same basic scenario adapted to learner level, evaluated with a quiz at the beginning. Therefore, Moodle is well adapted to this complex scenario, and it is currently under implementation to offer multiple and independent routes. In the case of measuring misinterpretation or test failures, the student is repositioned at the course beginning by the supervisor. Thus, the teacher can follow student's progress and specially help a student in difficulty. A first quiz evaluation is performed just after the introductory lecture, prerequisite of a previous course and basic elements to guide the learner to the most adapted path (e.g., more or less difficult choice of an active filter). This treasure hunt scenario needs numerous exchanges between the remote laboratory and the LMS, as well as several quiz evaluations. Quizzes are used for in-progress formative tests, practical interpretation, and summative tests.

4. Discussion

Finally, the main reason to stick to real experimental equipment such as Oscilloscope, DC source, Arbitrary Function Generator (AFG), Digital Multimeter (DMM) is explained in the following. Even if this equipment is expensive, students prefer to use real measurement devices as if they were in a traditional face-to-face work. All measurement devices can be implemented while they are remotely programmable. Similarly, these devices are much more accurate than an instrumentation solution with a micro-controller through analog to digital converter (ADC) and a signal processing. They are fully configurable (impedance, scale, DC/AC coupling, etc.) too. The main objective is to train a student to control remote devices as they will have to deal with them in their future professional life. Remote working is currently being used more and more. Remote control is common in the evolution of industry, Industry 4.0, as shown by [32] to improve factory efficiency and for remote maintenance by [33]. Engineering education centers need to take part of this evolution.

Power consumption measurement has been performed to evaluate consumption, for the adequate design of our power supply board, and to know the consumption of each device. Power consumption values of all equipment in our remote laboratory are shown in Table 2. These measurements were performed while the system was running. The overall consumption of the core system is less than 10 W. Then, equipment shall be added; however, it depends on the test bench and the laboratory type. Our test bench is composed of a DC source (ISO-Tech, IPS4303S), a waveform generator (Tektronix, AFG1022), a Digital Multimeter (Keithley, DMM 2110), and an oscilloscope (Tektronix, MDO3014).

Table 2. Remote laboratory consumption.

Hardware	Power (W)
Core system: Raspberry Pi3, external devices (2 webcams, external USB hard drive), motherboard, supply card (relay) and plug with DUT running	9 W
All other equipment: DC source, AFG, DMM, Oscilloscope, etc.	120 W
Total power consumption	129 W

5. Conclusions

The proposed solution is an entirely open source remote laboratory, not only for the software part but also for the hardware. The interoperability issue is a priority to offer a large compatibility and guarantee an easy adaptation for various applications. All Protocols and hardware interfaces compatible with Laborem are illustrated in Figure 13.

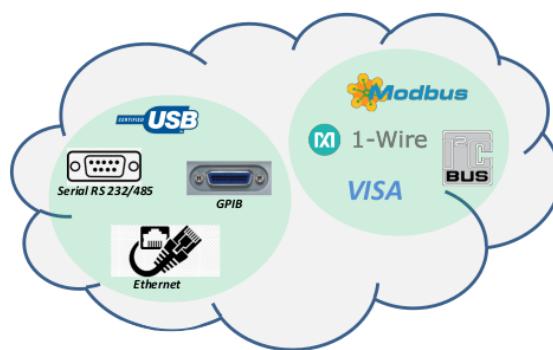


Figure 13. Protocols and hardware interfaces currently used by Laborem.

Laborem represents an all included solution to practical work for undergraduate students (Bachelor of Engineering, B.Eng or Bachelor of Science, B.Sc.). In Laborem experiments related to electronics, most of the undergraduate programs have already been done. Future work will be focused on extending to other subjects such as power electronics, applied sciences, energy, mechanical

engineering, and renewable energy thanks to the generic and modular hardware concept proposed. Concerning the software part, future works are conducted to increase our database of protocols such as BACnet. This special feature will allow for work on energy monitoring of buildings based on multiple electrical sensors exchange based on the open source developed solution.

In relation to the open source and modular proposition, the aim is to unite universities to create a global network. This has been already started with a first partner in Germany at the Technische Universität Berlin (TU Berlin) and two more in Burkina Faso at the University Joseph KI-ZERBO (Ouagadougou) in Autumn 2020.

Finally, all developments are available on the GitHub platform: under “PyScada-Laborem” for the software [34] and under “Laborem-hardware” for electronics and mechanical designs [35]. Thus, everyone can download and implement their own open source solutions for remote laboratories.

Author Contributions: Conceptualization, B.L. (Bastien Letowski), C.L., B.L. (Benoît Larroque) and F.L.; software, C.L. and M.S.; investigation, B.L. (Bastien Letowski), C.L., B.L. (Benoît Larroque), M.S., and F.L.; writing—original draft preparation, B.L. (Bastien Letowski) and C.L.; writing—review and editing, B.L. (Bastien Letowski), C.L., B.L. (Benoît Larroque), and F.L.; supervision, B.L. (Benoît Larroque) and F.L.; project administration and funding acquisition, B.L. (Benoît Larroque), and F.L. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Excellence Initiative of Université de Pau et des Pays de l’Adour—I-Site E2S UPPA, a French “Investissements d’Avenir” program for the financial support of the engineer fellowship Camille Lavayssière.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hornsby, D.J.; Osman, R. Massification in Higher Education: Large Classes and Student Learning. *High Educ.* **2014**, *67*, 711–719. [[CrossRef](#)]
2. Araneda, C.; Pedraja, L. Massification of Higher Education in Chile: Challenges for Chilean Universities. In Proceedings of the 10th annual International Conference of Education, Research and Innovation (ICERI2017), Seville, Spain, 16–18 November 2017; pp. 4356–4361.
3. Gustavsson, I.; Nilsson, K.; Zackrisson, J.; Garcia-Zubia, J.; Hernandez-Jayo, U.; Nafalski, A.; Nedic, Z.; Gol, O.; Machotka, J.; Pettersson, M.I.; et al. On Objectives of Instructional Laboratories, Individual Assessment, and Use of Collaborative Remote Laboratories. *IEEE Trans. Learn. Technol.* **2009**, *2*, 263–274. [[CrossRef](#)]
4. Gomes, L.; Bogosyan, S. Current Trends in Remote Laboratories. *IEEE Trans. Ind. Electron.* **2009**, *56*, 4744–4756. [[CrossRef](#)]
5. Aktan, B.; Bohus, C.A.; Crowl, L.A.; Shor, M.H. Distance Learning Applied to Control Engineering Laboratories. *IEEE Trans. Educ.* **1996**, *39*, 320–326. [[CrossRef](#)]
6. Bencomo, S.D. Control learning: Present and future. *IFAC Proc. Vol.* **2002**, *35*, 71–93. [[CrossRef](#)]
7. Sáenz, J.; Chacón, J.; Torre, L.D.L.; Visioli, A.; Dormido, S. Open and Low-Cost Virtual and Remote Labs on Control Engineering. *IEEE Access* **2015**, *3*, 805–814. [[CrossRef](#)]
8. Bermúdez-Ortega, J.; Besada-Portas, E.; López-Orozco, J.A.; Bonache-Seco, J.A.; de la Cruz, J.M. Remote Web-based Control Laboratory for Mobile Devices based on EJsS, Raspberry Pi and Node.js. *IFAC-PapersOnLine* **2015**, *48*, 158–163. [[CrossRef](#)]
9. Kalúz, M.; García-Zubía, J.; Fikar, M.; Čirka, L'. A Flexible and Configurable Architecture for Automatic Control Remote Laboratories. *IEEE Trans. Learn. Technol.* **2015**, *8*, 299–310. [[CrossRef](#)]
10. Wang, N.; Chen, X.; Lan, Q.; Song, G.; Parsaei, H.R.; Ho, S. A Novel Wiki-Based Remote Laboratory Platform for Engineering Education. *IEEE Trans. Learn. Technol.* **2017**, *10*, 331–341. [[CrossRef](#)]
11. Tran, L.Q.; Radcliffe, P.; Wang, L. A Low Budget Take-Home Control Engineering Laboratory for Undergraduate. *Int. J. Electr. Eng. Educ.* **2019**. [[CrossRef](#)]
12. Kalúz, M.; Čirka, L'; Valo, R.; Fikar, M. ArPi Lab: A Low-cost Remote Laboratory for Control Education. *IFAC Proc. Vol.* **2014**, *47*, 9057–9062. [[CrossRef](#)]
13. Zine, O.; Errouha, M.; Zamzoum, O.; Derouich, A.; Talbi, A. SEITI RMLab: A costless and effective remote measurement laboratory in electrical engineering. *Int. J. Electr. Eng. Educ.* **2019**, *56*, 3–23. [[CrossRef](#)]

14. Carballo, J.A.; Bonilla, J.; Roca, L.; Berenguel, M. New low-cost solar tracking system based on open source hardware for educational purposes. *Sol. Energy* **2018**, *174*, 826–836. [[CrossRef](#)]
15. Assante, D.; Capasso, C.; Veneri, O. Internet of Energy Training through Remote Laboratory Demonstrator. *Technologies* **2019**, *7*, 47. [[CrossRef](#)]
16. Rojko, A.; Hercog, D.; Jezernik, K. Power Engineering and Motion Control Web Laboratory: Design, Implementation, and Evaluation of Mechatronics Course. *IEEE Trans. Ind. Electron.* **2010**, *57*, 3343–3354. [[CrossRef](#)]
17. Monzo, C.; Cobo, G.; Morán, J.A.; Santamaría, E.; García-Solórzano, D. Lab@Home: The Open University of Catalonia Hands-on Electronics Laboratory for Online Engineering Education. *Electronics* **2020**, *9*, 222. [[CrossRef](#)]
18. Aghenta, L.O.; Iqbal, M.T. Low-Cost, Open Source IoT-Based SCADA System Design Using Thinger.IO and ESP32 Thing. *Electronics* **2019**, *8*, 822. [[CrossRef](#)]
19. Merchan, D.F.; Peralta, J.A.; Vazquez-Rodas, A.; Minchala, L.I.; Astudillo-Salinas, D. Open Source SCADA System for Advanced Monitoring of Industrial Processes. In Proceedings of the IEEE International Conference on Information Systems and Computer Science (INCISCOS), Quito, Ecuador, 23–25 November 2017; pp. 160–165.
20. Mejías, A.; Herrera, R.; Márquez, M.; Calderón, A.; González, I.; Andújar, J. Easy Handling of Sensors and Actuators over TCP/IP Networks by Open Source Hardware/Software. *Sensors* **2017**, *17*, 94. [[CrossRef](#)]
21. Deese, A.S. Development of Smart Electric Power System (SEPS) Laboratory for Advanced Research and Undergraduate Education. *IEEE Trans. Power Syst.* **2015**, *30*, 1279–1287. [[CrossRef](#)]
22. Aydogmus, Z.; Aydogmus, O. A Web-Based Remote Access Laboratory Using SCADA. *IEEE Trans. Educ.* **2009**, *52*, 126–132. [[CrossRef](#)]
23. Vargas-Salgado, C.; Aguilera-Leon, J.; Chiñas-Palacios, C.; Hurtado-Perez, E. Low-Cost Web-Based Supervisory Control and Data Acquisition System for A Microgrid Testbed: A Case Study in Design and Implementation for Academic and Research Applications. *Heliyon* **2019**, *5*, e02474. [[CrossRef](#)]
24. Chacón, J.; Farias, G.; Vargas, H.; Visioli, A.; Dormido, S. Remote Interoperability Protocol: A bridge between interactive interfaces and engineering systems. *IFAC-PapersOnLine* **2015**, *48*, 247–252. [[CrossRef](#)]
25. Luthon, F.; Larroque, B. LaboREM: A Remote Laboratory for Game-Like Training in Electronics. *IEEE Trans. Learn. Technol.* **2014**, *8*, 311–321. [[CrossRef](#)]
26. Lavayssiére, C.; Letowski, B.; Larroque, B.; Luthon, F. Easy Applied Sciences Learning through Open Source Remote Laboratory. In Proceedings of the 11th Int. Conf. Education, Research and Innovation (ICERI2018), Seville, Spain, 12–14 November 2018; pp. 2576–2583.
27. Lavayssiére, C.; Letowski, B.; Larroque, B.; Luthon, F. Laborem—A network of open source remote laboratories for learning. In Proceedings of the 405th International Academic Conference on Engineering, Technology and Innovations (IACETI 2018), Santiago, Chile, 16–17 July 2018; pp. 1–5.
28. Letowski, B.; Lavayssiére, C.; Larroque, B.; Luthon, F. An Open Source Remote Laboratory Network Based On A Ready To Use Solution: Laborem. In Proceedings of the 12th International Conference of Education, Research and Innovation (ICERI2019), Seville, Spain, 11–13 November 2019; pp. 5726–5731.
29. Schröder, M. PyScada Is a Open Source Scada System That Uses the Django Framework as Backend: Trombastic/PyScada. Available online: <https://github.com/trombastic/PyScada> (accessed on 14 June 2020).
30. Raspberry Pi 3 Model B+. Available online: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (accessed on 12 June 2020).
31. Reck, R.; Sreenivas, R. Developing an Affordable and Portable Control Systems Laboratory Kit with a Raspberry Pi. *Electronics* **2016**, *5*, 36. [[CrossRef](#)]
32. Smajic, H.; Wessel, N. Remote Control of Large Manufacturing Plants Using Core Elements of Industry 4.0. In *Online Engineering & Internet of Things*; Auer, M.E., Zutin, D.G., Eds.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2018; Volume 22, pp. 546–551
33. Masoni, R.; Ferrise, F.; Bordegoni, M.; Gattullo, M.; Uva, A.E.; Fiorentino, M.; Carrabba, E.; Di Donato, M. Supporting Remote Maintenance in Industry 4.0 through Augmented Reality. *Procedia Manuf.* **2017**, *11*, 1296–1302. [[CrossRef](#)]

34. PyScada-Laborem GitHub Page. Available online: <https://github.com/clavay/PyScada-Laborem> (accessed on 18 July 2020).
35. Laborem-Hardware GitHub Page. Available online: <https://github.com/bletow/Laborem-hardware> (accessed on 18 July 2020).

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).