# Learning CAN bus communication with a remote laboratory

Ignacio Del Villar
Department of Electrical, Electronic and Communications Engineering
Public University of Navarra
*Pamplona, Spain*
ignacio.delvillar@unavarra.es

Luis Rodriguez-Gil
LabsLand
*Bilbao, Spain*
luis@labsland.com

Pablo Orduña
LabsLand
*Bilbao, Spain*
pablo@labsland.com

***Abstract*— CAN bus is a very important communications protocol. Its usage is extended to many applications and it is key to know some fundamentals in an introductory course to industrial communications. A collaborative experience between the Public University of Navarra has permitted students of Industrial Technologies Engineering and Electrical and Electronic Engineering to access a remote laboratory where they can progress in the development of a project where they must implement a CAN bus client and server for monitoring the parameters of a vehicle. More than twenty parameters can be monitored and the remote laboratory is 24 hours a day accessible, which permits students to work at home and progress more in the multiprotocol project they develop at the university. The interface and some code examples are shown, along with some statistics that demonstrate the utilization of the tool by the students with the aid of some evaluation measurements taken by the lecturer. Finally, a survey was performed to the students in order to understand better the potential of the remote laboratory assisted teaching as well as to know where to improve it in the next future.**

**Keywords— Remote Laboratory, CAN bus, Arduino, OBDII**

## I. INTRODUCTION

Industrial communications is a domain where many different protocols coexist to cover multiple types of applications [1]. In addition, due to the increasing dynamicity of industry, it is necessary to adapt to new and different needs. In this sense, Ethernet and wireless based protocols are probably the two main technologies that have contributed to revolutionize the industry during the last years. As a result, many protocols have decayed, whilst others survive, i.e. those that adapt to the new paradigm of industry. One good example is Controller Area Network (CAN) bus, the standard communications protocol in vehicles [2], and a widely used protocol in domains such as factory automation, aviation and telematics [3]. The explanation for the success of CAN bus is that it is a simple, inexpensive and robust network technology [1].

The simplicity and importance of CAN bus makes it an interesting protocol for engineers in degrees where the fundamentals of industrial communications are explained, such as Industrial Technologies Engineering (ITE) and Electrical and Electronic Engineering (EEE) in the Public University of Navarra [4]. However, during the last years it has been observed that the hours in the laboratory are not enough for less skilled students to gain a deep knowledge on CAN bus. That is why it was decided to complement the face-to-face laboratory with a remote laboratory experience, which permits students to work in an environment that resembles reality.

In previous years the students developed a CAN bus communication between two nodes in class, along with other communications, in a multiprotocol communication project [4]. This year, during the development of the CAN bus protocol part of the project, they can test their programs in the remote laboratory both in class and at home, which permits the students to progress more in the development of their project. This is one of the main advantages a remote laboratory offers, i.e. the 24 hour a day availability of a laboratory that is real (the only difference with a traditional laboratory is that the elements are controlled remotely), at the same time costs are reduced in view that the laboratory can be shared by many institutions around the world [5], [6]. This concept has been used in remarkable initiatives such as the European Go Lab project, aimed at improving the STEM education [7].

## II. METHODOLOGY

The methodology is based on the previous collaborative experience between the Public University of Navarra and LabsLand [8]. LabsLand is a company that provides access to educational laboratories and equipment, online. For that purpose, its platform connects schools and universities with real laboratories hosted at different universities and institutions around the world [9]. The CAN bus communication laboratory described in this work is built upon the more general Arduino Board remote laboratory [10]. This decision is based on the previous experience with CAN bus communication in the Public University of Navarra hands-on laboratory developed on the basis of Arduino, which at the same time is an adequate platform for learning with the help of many bibliographical facilities and scripts available in the Internet [4], [11]. Regarding the Arduino Board remote laboratory, it provides students access to an Arduino controller where they can upload their own code and that has certain peripherals connected, including potentiometers, a display, switches, buttons and others.

Multiple physical copies of these laboratories are deployed through LabsLand in various institutions around the world (in universities in Spain, South Africa, and Costa Rica, among others). To ensure that all this pre-existing infrastructure can be leveraged for the Arduino CAN BUS remote laboratory, the CAN BUS remote laboratory has been implemented as an extension or variation of the standard Arduino Board remote laboratory (Fig. 1 shows an instance of the Arduino Board laboratory in the University of Fort Hare in South Africa).

In addition, two different user interfaces have been implemented: a CAN BUS client and a CAN BUS server perspective. Students can choose one or the other depending on which side's logic they are working on. For simplicity, the laboratory does not allow students to program both at the same time. Instead, if they want to develop both a client and a

server, they will need to first program one, and then the other, separately.

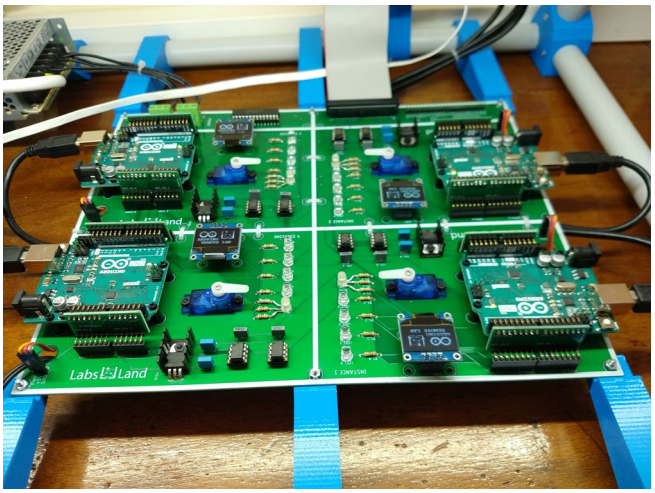

Fig. 1. Instance of the Arduino Board laboratory in University of Fort Hare (South Africa). It contains 4 Arduinos, which permits that 4 students connect in parallel with the remote laboratory.

The remote lab can also provide statistics of the number of accesses and time spent by students in the platforms, which along with a survey will permit to evaluate the acceptance of the methodology by the students.

Currently, the remote laboratory is prepared for obtaining, with a client code written with Arduino, 23 parameters of the car via OBD-II messages created in the Arduino script and integrated in CAN bus frames. Students will implement the program to extract the parameters of the car in an easier way with the remote lab, which will permit to deepen the knowledge on the field and to dedicate more time for developing other types of communication in the project developed in the face-to-face laboratory compared to other years.

In order to facilitate the implementation of the programs, they are given two samples codes for two can bus nodes where one of the nodes sends a message and the other receives this message [4].

When students access the online platform, they can choose among three modes of operation: standard Arduino, where they can use Arduino in a normal way, CAN bus client mode and CAN bus server mode (see Fig. 2). These last two modes are the ones implemented for the CAN bus remote laboratory described in this work.

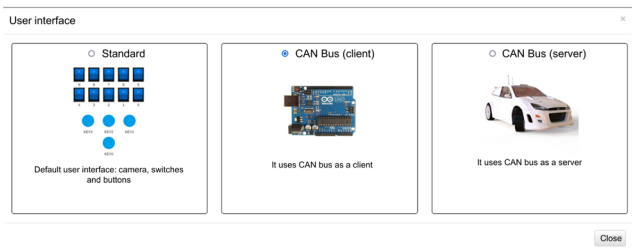


Fig. 2. User Interface where CAN bus client and server mode can be selected

Once they select CAN bus client or CAN bus server mode, they enter an interface where they can introduce their program and later load it into the Arduino board.

Fig. 3a shows an example for the client program the students can develop (the user interface is already given to the student).

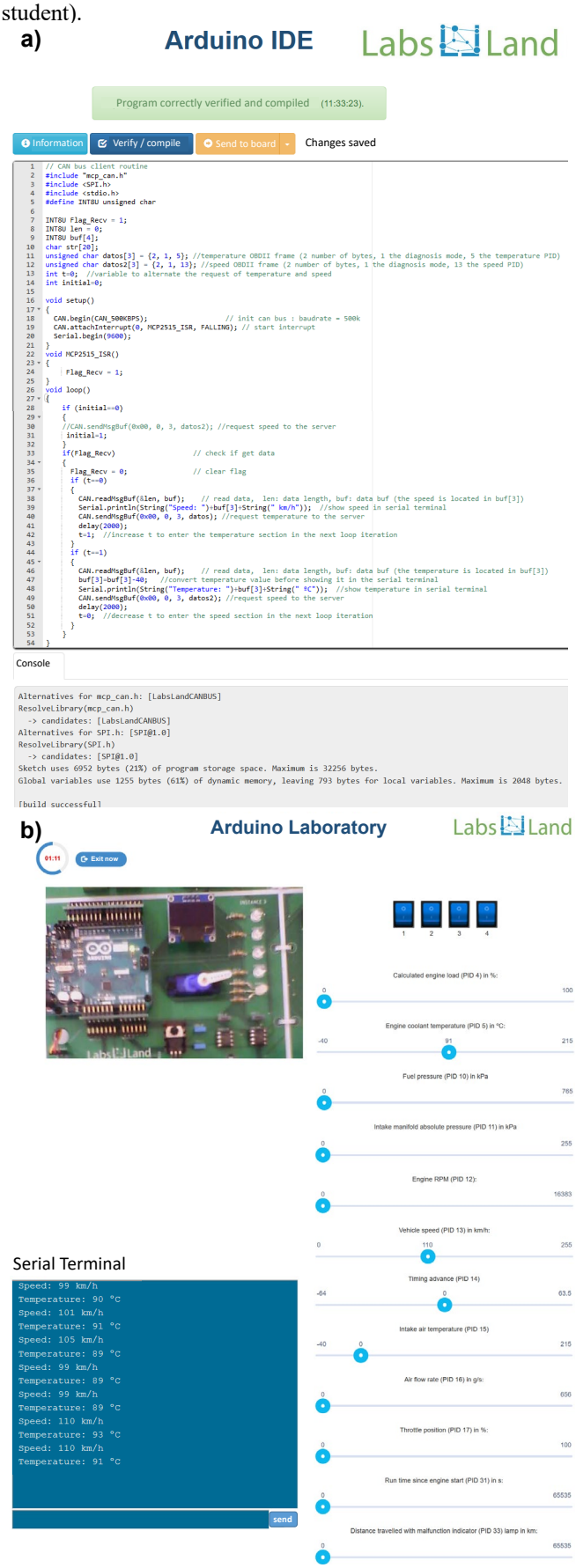


Fig. 3. Arduino client script and interface after transfering the code to the Arduino board.

This client code focuses on requesting information from two parameters: the engine coolant temperature and the speed. For temperature, according to the OBDII protocol, in the data section of the CAN bus frame three bytes must be sent: 2, 1, 5, where the first one is the number of bytes after it, the second one is the mode of operation and last one corresponds with the parameter identificator (PID). On the other hand, 2, 1, 13 is sent for requesting speed since the PID for speed is 13. This information is introduced is the CAN.sendMsgBuf function.

The other key function in the script is *CAN.readMsgBuf*, which is responsible for receiving the information from the server and storing it in the variable *buf*. According to OBDII, the temperature ranges from -40 to 205º C and this parameter can be implemented with a single byte located in the last position of the data section of the CAN bus frame (the data section is stored in the vector *buf* in the Arduino script). This byte is called variable A according to the OBDII protocol and, under the operation A-40, it gives the temperature. For instance, if A is 35, it means that the temperature is 35-40=-5 ºC. Regarding the motor speed, there is no need to adapt the value included in A. The range of operation is 0 to 255 ºC and if A contains 95, this means that the vehicle speed is 95 km/h.

In the example shown in Fig. 3 and 4, for the sake of simplicity, two variables of one byte are shown, but the program can be easily extended to operate with variables of two bytes or even 4 bytes, like the odometer. In fact, the students work in pairs and each couple of students must develop the program for two different variables, one of one byte and one of two bytes. Consequently, 11 different pairs of students can work with a different project on the same topic. Moreover, the odometer is offered to the students as an optional exercise to increase the mark for those who finish first this part.

Fig. 3b shows the interface that students visualize once they load their scripts in the Arduino board. On the right side of the figure, it is possible to visualize the different variables of the server, which can be controlled by the user in such a way that the client receives the value indicated by the corresponding control. At the bottom left of Fig. 2b, the student can observe the serial terminal, i.e. the information sent by the USB connection of the Arduino board, with serial functions such as *serial.print*. In that section the contents of variable included in the last element of vector *buf* can be printed after performing the necessary conversions, as it was the case in the temperature parameter. Moreover, the student can vary the controls in the right side of the panel while an immediate reaction is generated in the serial terminal (in Fig. 3b the temperature and speed values change due to the movement of their corresponding controls). In addition, the serial terminal includes the possibility of sending messages by the serial port.

Similarly, in Fig. 4a the code for the server is shown, where in this case four different engine coolant temperatures and vehicle speeds are sent to the client by the server. In the case of the temperature, an initial temperature of 88 ºC (128 in A variable) is set and this temperature is increased by 1º C each time the temperature is requested. After 91 ºC, the temperature is restarted again to 88 ºC. Regarding the vehicle speed, it is varied from 40 to 55 km/h in steps of 5 km/h, and after 55 km/h, the speed is restarted to 40 km/h. Obviously, other approaches could be implemented, such as a random variation of the variables or controlling the variables with a potentiometer in the Arduino board. The variables could even be shown in the display of the Arduino board in real time. In other words, the amount of possibilities is enormous, though in a course for students with low programming experience and centered on communications, it is better to focus on learning the basic concepts.

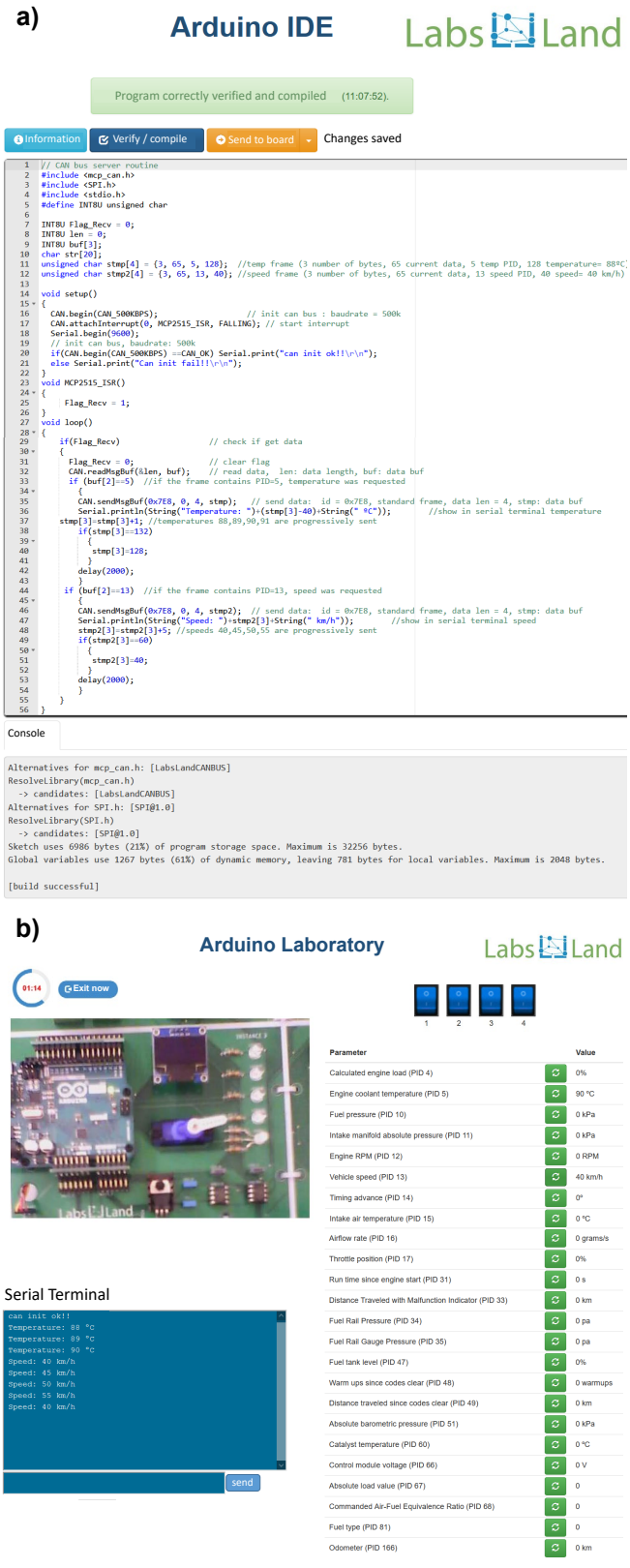


Fig. 4. Arduino server script and interface after transfering the code to the Arduino board.

Fig. 4b shows the server interface after loading the server code in the Arduino board. The interface is similar to the client interface, with the physical Arduino on top of the screen and the serial terminal at the bottom left, showing the serial messages. The main difference is located in the right part of the screen, where this time the end user only clicks on the button of a specific parameter to give the order to the server of sending this parameter to the client. At the same time, the value of the parameter in question is updated any time the button is pressed.

## III. RESULTS

A remote laboratory is a useful tool but students must be encouraged to use it. The previous year, the students were offered the tool without any transfer of their work and effort to the final mark. However, this year they have been forced to show the code and prove that it works. In addition, they have been offered the possibility to increase the final mark. The first couple that submits the client and the server for the two parameters and the odometer increases the final mark by 0.5 points, the second couple increases 0.3 points and the third couple 0.2 points. As a result, the utilization of the lab has increased a lot. In Fig. 5 there is a graph showing the statistics for year 2020 and year 2021 in the two degrees where this subject is taught in the Public University of Navarra: Industrial Technologies Engineering (ITE) and Electrical and Electronic Engineering (EEE). The results clearly indicate the higher utilization of the remote laboratory in 2021 compared to 2020. In addition, it is also observed that students of the degree in ITE are slightly more interested in the topic than students of EEE.

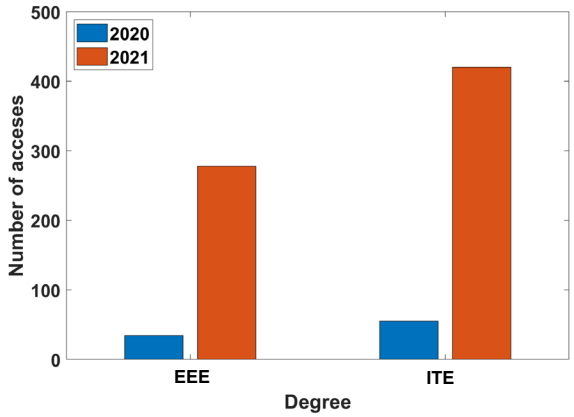


Fig. 5. Statistics of utilization of the CAN bus remote lab in year 2020 and 2021 in Electrical and Electronic and Industrial Technologies Engineering in the Public University of Navarra.

As an example of the potential of the remote lab experience, it is also important to indicate that one student had already finished both the client and server prior to attend the first face-to-face laboratory session. Therefore, he just verified the good performance of his designs in the face-to-face laboratory. Regarding the rest of students, this year they spent only two hours in the face-to-face laboratory to finish the CAN bus part of the project whilst the previous year the students spent around 6-8 hours to do the same task. At the same time, the lower time spent did not mean a worse knowledge of the topic since in the theory exam the results were similar to previous year.

In addition, a survey was made to the students at the end of the CAN bus part in which they were asked several questions following a 4-point Likert scale similar to that used in [12], where they could answer: 1. Strongly disagree. 2. Partially disagree. 3. Partially agree. 4. Strongly agree. This scale forces students to take a position in favor or against in any question, which permits to extract a more clear conclusion on the results obtained.

Different questions were asked regarding three topics: lab acceptance, teacher guidance and learning.

Lab acceptance: students were asked on easiness of access to the remote laboratory, the time available once they loaded their program to the Arduino board in order to verify if it works correctly, and the simplicity of use of the tool. This theme obtained an overall average of 2.9.

Teacher guidance: here the questions were focused on whether the teacher was available for answering questions and solving issues, and on whether enough information was given to complete the tasks to solve with the remote laboratory. This theme obtained an overall average of 3.3.

Learning: here students were asked on whether the laboratory permits to delve into CAN bus and OBDII and whether the remote lab tool permits to progress in the project the students must solve in the face-to-face laboratory. Here the average was 2.9.

The average global mark was 3.0, which indicates that students partially agree.

In addition, students were asked to give their opinion on three questions. The first one was what they find more interesting in the remote laboratory. The answer was unanimous: they can access the laboratory at home at any time and they can progress more in the project, which permits them to understand better CAN bus and OBDII. The second question was about the main difficulties. Here the answers were more disperse, though they focused on technical questions such as errors detected with some PIDs or difficulties in uploading the Arduino programs. Finally, there was a place for suggestions and the most repeated suggestion was to dedicate more time in class or in the face-to-face laboratory to explain how to use the remote laboratory with examples.

## IV. CONCLUSIONS

A remote lab where the student can test a CAN bus communication program in client or server mode is a good complement for face-to-face laboratory, which permits to deepen the knowledge acquired by the students on the topic.

In addition, in terms of implication, it has also been demonstrated that students must see a compensation in the final mark for the work performed with the remote laboratory. In the present year the utilization of the remote laboratory has increased a lot compared to the previous year thanks to the maximum 0.5 point increase they can get by completing and testing the client and server code with the remote laboratory, at the same time it is necessary for all to demonstrate that their program works.

Another advantage of the remote laboratory is that it permits to reduce the time students require to finish the CAN bus part in the multiprotocol project they must solve: two hours in the face to face laboratory this year whereas other years around 6-8 hours. The opinion of the students is also positive. They gave to the remote lab experience a mark of 3

in a scale from 1 to 4 and they consider it an interesting tool for progressing at home in the development of the project they must develop in the face-to-face laboratory. However, there is still work to do on solving small technical problems and regarding a more clear explanation in class of the utilization of the remote laboratory.

The remote laboratory experience also opens the path for developing more communication protocols instances, towards a subject where knowledge of all protocols explained in class can be deepened at home. This demanding challenge could become a reality thanks to the concept of sharing laboratories in a global network proposed by LabsLand. With this philosophy, it is not necessary to implement all the technologies in the same university, in this case in the Public University of Navarra. In fact, the Arduino boards used in this Industrial Technologies and Electrical and Electronic Engineering are not located at the Public University of Navarra, while other institutions use the FPGA located at this university. In this sense, it is very important to select adequately the best platform for teaching a specific protocol (for CAN bus it was Arduino), at the same time all institutions share a common project where they complement each other in terms of offering platforms that can be used in many subjects. However, the idea that the CAN bus remote laboratory could be expanded to other protocols that students work in the subject should not mean to replace the face-to-face laboratory, because students need contact with the teacher and mates to progress. Humans are social people. Consequently, the balance between remote and face-to-face experience must be maintained.

## ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Public University of Navarra with the education innovation project “Improving Academic Performance through Remote Laboratory and Flipped Classroom”, supported by the PINNE 2021 initiative.

## REFERENCES

[1] R. Zurawski, Ed., *Industrial Communication Technology Handbook*, Second. CRC Press, 2017.

[2] G. Kornaros *et al.*, “Towards holistic secure networking in connected vehicles through securing CAN-bus communication and firmware-over-the-air updating,” *J. Syst. Archit.*, vol. 109, no. March, p. 101761, 2020, doi: 10.1016/j.sysarc.2020.101761.

[3] A. J. E. and R. M. S. I. Hannah M. Boland, Morgan I. Burgett, “An Overview of CAN-BUS Development, Utilization, and Future Potential in Serial Network Messaging for Off-Road Mobile Equipment,” in *Tecjnology in Agriculture*, Interchopen, 2021.

[4] I. Del Villar, A. B. Socorro-Leranoz, and I. R. Matias, “Work in progress: Multiprotocol system for learning industrial communications,” *IEEE Glob. Eng. Educ. Conf. EDUCON*, vol. 2020-April, pp. 1554–1558, 2020, doi: 10.1109/EDUCON45650.2020.9125125.

[5] R. Heradio, L. De La Torre, D. Galan, F. J. Cabrerizo, E. Herrera-Viedma, and S. Dormido, “Virtual and remote labs in education: A bibliometric analysis,” *Comput. Educ.*, vol. 98, pp. 14–38, 2016, doi: 10.1016/j.compedu.2016.03.010.

[6] J. García Zubía and G. R. Alves, Eds., *Using remote labs in education: Two little ducks in remote experimentation*. Universidad de Deusto, 2012.

[7] T. De Jong, S. Sotiriou, and D. Gillet, “Innovations in STEM education: the Go-Lab federation of online labs,” *Smart Learn. Environ.*, vol. 1, no. 3, pp. 1–16, 2014.

[8] C. A. Mayoz, A. L. Da Silva Beraldo, A. Villar-Martinez, L. Rodriguez-Gil, W. F. M. De Souza Seron, and P. Orduna, “FPGA remote laboratory: Experience of a shared laboratory between UPNA and UNIFESP,” *Proc. - 2020 14th Technol. Appl. to Electron. Teach. Conf. TAEE 2020*, 2020, doi: 10.1109/TAEE46915.2020.9163773.

[9] P. Orduña, L. Rodriguez-Gil, J. Garcia-Zubia, I. Angulo, U. Hernandez, and E. Azcuenaga, “LabsLand: A sharing economy platform to promote educational remote laboratories maintainability, sustainability and adoption,” *Proc. - Front. Educ. Conf. FIE*, vol. 2016-Novem, pp. 4–9, 2016, doi: 10.1109/FIE.2016.7757579.

[10] A. Villar-Martinez, L. Rodriguez-Gil, I. Angulo, P. Orduna, J. Garcia-Zubia, and D. Lopez-De-Ipina, “Improving the Scalability and Replicability of Embedded Systems Remote Laboratories through a Cost-Effective Architecture,” *IEEE Access*, vol. 7, pp. 164164–164185, 2019, doi: 10.1109/ACCESS.2019.2952321.

[11] A. D’Ausilio, “Arduino: A low-cost multipurpose lab equipment,” *Behav. Res. Methods*, vol. 44, no. 2, pp. 305–313, 2012, doi: 10.3758/s13428-011-0163-z.

[12] P. E. Hertzog and A. J. Swart, “Arduino - Enabling engineering students to obtain academic success in a design-based module,” *IEEE Glob. Eng. Educ. Conf. EDUCON*, vol. 10-13-Apri, no. April, pp. 66–73, 2016, doi: 10.1109/EDUCON.2016.7474533.