

This manuscript is a post-print version of the document published in:

Fernández-Pacheco, A., Garcia S., Ruipérez-Valiente, J. A., & Castro M. Remote experimentation through Arduino-based Remote Laboratories (In Press). IEEE Revista Iberoamericana de Tecnologías del Aprendizaje (IEEE-RITA).
<https://doi.org/10.1109/RITA.2021.3089916>

<https://ieeexplore.ieee.org/document/9471829>

© 2022 IEEE

Remote experimentation through Arduino-based Remote Laboratories

Sergio Martin, Atilano Fernandez-Pacheco, José A. Ruipérez-Valiente, German Carro and Manuel Castro

Abstract—This paper describes the architecture of an Arduino remote lab that supports the deployment of many Arduino-based experiments, such as a sensors remote lab consisting on eleven sensors and a LCD display connected to an Arduino MEGA, or a 3D RGB LED cube remote lab consisting on 16 RGB LEDs connected to an Arduino UNO-compatible board. The proposed on-line system allows students to write code on a website to be executed on these experiments. The execution results can be observed in real time through an IP camera. The use of this kind of on-line didactic tools is very important to provide high quality on-line education programs on technical fields.

Keywords—Remote lab; Arduino; Sensors; Raspberry Pi

I. INTRODUCTION

One of the most challenging issues in on-line education is how to facilitate experimental activities in technical fields. Laboratory practice is essential in STEM (Science, Engineering, Technology and Mathematics) disciplines to help students assimilate correctly the theory concepts and can acquire the practical skills to be used in their professional career once their degree is finished. Traditionally, on-campus education provides hand-on practice sessions where students can physically conduct experimentation with technical equipment once a week. These lab sessions are vital for the development of the students and the university must assure a good amount, variety and quality of practice time.

Traditionally, laboratory practices have been done in the university facilities (such as a classroom or specific spaces with the necessary equipment) with a previously stipulated fixed schedule that do not provide much flexibility. These laboratory practices usually take place in small groups to facilitate that teachers can provide a more personalized attention and ensure a suitable and safe use of the technical equipment. In these traditional laboratories, equipment, teacher attention and access to the lab is limited due to budgetary problems [1] [2].

According to many authors [3] [4] [5] [7], one potential solution to these problems can be found in the use of virtual and remote laboratories [8] [9]. The use of these on-line labs in engineering education is becoming more and more important everyday because of the need of providing flexible, scalable and high quality access to acquire practical competences with

real instrumentation at anytime and anyplace, not only within a classroom [10].

This article is an extension of the paper published by the authors [11] as part of the EDUCON 2019 conference. This new article includes a reworked presentation of the related work regarding the use of Arduino in education, and it considerably extends the experimentation section describing with more details the 3D LED cube remote lab and the sensors lab developed under the proposed architecture. The abstract, introduction and conclusions sections have been also extended or modified.

In this case study, the proposed remote lab will be used by Electronics Engineering students of a distance learning university as part of their mandatory laboratory sessions within the Microprocessor and Microcontrollers subject. We would like to denote that this paper will share guidelines and lessons learned that will be of interest for a wider audience of students, teachers and professionals who are interested in remote learning and technology laboratories.

The main goals of this paper are:

- Describe the architecture of an Arduino remote lab based on Raspberry Pi.
- Describe the configuration of this remote lab to support the deployment of several experiments, such as a 3D LED cube or a Sensors lab.

II. RELATED WORK

A. The use of Arduino in education

There are currently several universities where Arduino is being used as a resource in those classes dedicated to embedded systems. The authors in [12] started in 2010 to introduce the Arduino platform as a possibility for students to complete the projects of their class. According to these authors, a high number of students chose Arduino as a platform for the implementation of their projects instead of FPGAs or PIC microprocessors. In addition, both the variety of projects as well as their complexity grew when compared to years prior to the inclusion of Arduino. Among the projects carried out at this university with Arduino there are a spectrum analyzer or a paintball gun that shoots autonomously, where the Arduino UNO controls the servos [13].

Another example of using Arduino as part of the curriculum of a class can be found at the Technological

University of Bratislava [14]. In this university, the Boe-Bot commercial plate [14] was used for years for the development of robots. However, students reported that using BASIC as the language to develop their projects was a disadvantage. Therefore, the class coordinators eventually decided to adapt their curriculum to use the Arduino board instead of the hardware that was previously used, and the results were quite satisfactory. The coordinators reported that with few lessons it, was already possible to program a robot that was able to follow a line while avoiding different obstacles.

Based on the related literature that we have reviewed and our own experience, the greatest benefits of using Arduino within the curriculum of a class are:

- Ease of configuration. Once you have installed the Arduino board driver, it is very straightforward to make use of it within the programming environment.
- There are lots of examples for controlling peripherals in the same development environment that can be used as a guide.
- Accessible and free software.
- A very active and broad Arduino community that shares the code of the projects. This is a great advantage for the students, since they have many examples easily available and they can find responses to their issues.
- Multi operating system support, which facilitates being able to work on Windows, Linux or Mac.
- Low maintenance cost. In case of damage to the microcontroller, a new one can be acquired to replace it at a low cost. There are microcontrollers with the bootloader included and others without it. The latter can be used in the case of having a microcontroller programmer.
- Students can build prototypes of their projects quickly. The possibility to obtain real and tangible results in a short time, increases the motivation of the student, thus improving the learning process.

Therefore, it is important that educational institutions can have access to high quality Arduino labs where the students can learn skills related to this technology with the objective of improving the technical competencies of students and their future employment prospects.

B. Remote laboratories

These labs are often not tied to a specific topic; instead they cover broad areas of knowledge and applications (e.g., programing, control system, etc), and hence the labs can be used across a variety of classes. The general objective of these labs is enabling the possibility for students to put in practice the theoretical knowledge that they have acquired through the formal lectures through the use of a virtual interface that permits the student to interact with the remote lab. During this process, students are able to perform hands on experimentation with a real scenario, facilitating the theory to practice transfer and consolidating this knowledge.

These labs specify different roles and access permissions, which are often organized in two security levels: The first one specifies which users can access the lab and the second one specifies the actions that each user can do within the application. Moreover, all labs must introduce some sort of self-regulated and security control that has as objective minimizing errors that could cause harm to the equipment and devices. In the case of the software that is used to interact with the remote lab, the system must control the possible case scenarios, so that no inadequate actions are executed.

The presence of this kind of remote labs has greatly increased in the Spanish university UPV/EHU, where students have at their disposal a good number of these labs. As conclusion, remote labs have lots of advantages [15]:

- There are no time restrictions and provide a high degree of flexibility for the student.
- It is possible to include easy controls and feedback to avoid inadequate usage that could be potentially harmful for the equipment.
- It can increase the maximum amount of students who can use these labs by far since there are no physical space restrictions. This factor has a limitation, which is the number of concurrent users.
- It can increase the number of students who can use this equipment to reduce the cost per student and thus increasing quality in other areas where more investment is required.

According to previous work [6], the main requirements that a remote laboratory must fulfill are the following:

- High-availability. One strong characteristic is the possibility to be used without restriction of time and with high availability. It is critical that this feature is strongly taken into account as part of the design and architecture of the laboratory.
- Concurrency. The lab architecture needs to accommodate the usual case scenario where several users are interacting with the lab simultaneously; all users should be able to work at the same time. Otherwise, there will be a queue of requests and the system will take time to process.
- Low operating cost. Since they will be used with high frequency, the construction and maintenance of these labs should have low operational and financial cost.
- The remote lab must have a self-regulation control system to avoid an inadequate use. In other words, the system must protect itself from being damaged by the instructions from students.

C. Arduino remote labs

Carro *et al.* [16] developed a remote lab for primary school students to control the color of a LED light connected to an Arduino. This lab did not allow to program the Arduino itself, but it just provided a web interface to select the mixture of colors to be displayed on the LED. Thus, this lab was not

designed to teach Arduino programming, but to facilitate an explanation regarding light components.

Sarik and Kymissis [17] introduce a distributed lab (or take-home lab) for Arduino programming. It consisted on providing an Arduino kit to students that they could take home. The main advantage of these take-home labs is that students can work with the real hardware connected to their computers. These setups also have some limitations that we detail next:

- These kinds of take-home laboratories are only viable for electronic designs that are not very complex, since students do not have access to specific equipment needed to prepare the designs. However, a feasible possibility is to mount these complex labs only once and to share them with as many students as needed.
- Take-home labs are also not viable when the proposed experiment includes expensive components, since it would imply to buy as many expensive components as students. This multiplies the cost of the lab. However, when working with remote labs only one kit is needed instead.
- Maintenance becomes more complex since the transport of delicate electronic designs from school to home and back, may often unintentionally incur in breaking some pieces. However, when we work with remote labs, there is no transportation risk for the equipment.
- Remote labs make possible to reuse the same laboratory for several classes at the same time.
- When a student uses a take-home lab on their own, the instructions programmed by the student could damage the lab if there is a mistake in the code. For example, when programming the movement of a robotic arm, the student may damage one of the articulation motors if the student does not control for the arms' torsion limit. Remote labs allow to include an extra software layer that can control these dangerous actions.

Other authors, such as Mejías Borrero *et al.* [18], developed a framework to convert a laboratory experiment in a remote laboratory. This framework allows interacting with Arduino input/output pines and configuring a webcam. However, this framework does not allow the student to write the Arduino code to be executed.

A similar problem is found on the work of Kalúz *et al.*, [19] as well as in Cardoso and Sousa [20]. Their labs make use of Raspberry Pi and Arduino as the base to create remote laboratories. However, the experiments are not focused on allowing the student to write the code to be executed on the Arduino board. The Arduino board is pre-programmed to manage some specific experiment (hydraulic plant or a model of magnetic levitation) allowing the student to modify some of the parameters applied to the experiment, but not the code itself. Thus, the aim of this work is not to allow students to learn and improve their Arduino programming skills, but to experiment with the configuration of different parameters to be executed on a remote lab.

III. ARCHITECTURE

In this section, the different architectural elements are explained. The main device is an Arduino board that is connected on the one side to other devices such as the led cube, mechanic crane, sensors, etc., and on the other side to a Raspberry Pi that is acting as a web server.

The connection between the Raspberry Pi and the Arduino board is done through the USB port. The number of Arduino boards that can be connected to the Raspberry Pi server is limited due to the number of USB ports available.

The administrator needs to setup the remote lab indicating the different Arduino boards connected. For accomplishing this task, the administrator has to add one line for each new Arduino connected. The format line is as follows:

“Port, name device, microprocessor, programmer, baud, default program”

The description of each one of these parameters is the following:

- Port - It is the device file, which will be used by the Arduino board.
- Name device - It is the name of the device that users will visualize in the web site.
- Microprocessor - This parameter indicates the type of Arduino processor, for example atmega328p, atmega2560, etc.
- Programmer - This parameter indicates the type of programmer to be used in the connection, for example Arduino or wiring -D.
- Baud - It is necessary to define the connection speed. This parameter is very important because a bad duration can cause errors when the program is being loaded to the device.
- Default program - This parameter indicates the reset default program to be executed when the user's time has expired. This is a binary program (.hex) that performs all necessary tasks in the device to reset it and leave it in a stable state for the next user. This program setups the device with its default configurations.

Some examples of default configuration lines for different devices, such as a 3D LED cube based on an ATmega328p microprocessor and an Arduino MEGA lab based on ATmega2560, are the following:

```
/dev/ttyUSB0,Cubo Arduino,atmega328p,arduino,57600,ini_cubo.hex
```

```
/dev/ttyACM0,Sensores,atmega2560,wiring -D,115200,ini_sensores.hex
```

Although the system can support a lot of users concurrently, only one can be executing instructions at the same time. Then, it is necessary to introduce mechanisms of synchronization and queue in order to orchestrate the different executions. Thus, when the instructions of a user are being executed, the particular experiment used (i.e., sensors lab, led cube) is blocked for other users during ten minutes or until the

current user finishes the session with the lab. During that time, if other user tries to execute another program, the user will receive a notification indicating that the action cannot be performed and how much more time the device will remain blocked. In this case, the user enters a queue waiting for their turn to gain access to the remote lab based on the arrival time i.e., first to arrive first to gain access.

On the other hand, the Raspberry Pi is the hardware chosen as application server. The rationale for this decision is very similar to the one previously stated for Arduino selection. It is possible to deploy a full server with high performance characteristics at a reduced cost. Moreover, Raspberry Pi's developer's community is well established and widespread. Raspberry is based on Linux system and thus making it is very easy to deploy as a web server.

The main steps performed by the webserver to execute students' source code into the Arduino board are very similar to the ones done by the Arduino IDE when compiling and executing a source code to an attached board. The main difference is that a makefile script does all the steps instead of being executed by the own Arduino IDE. These are the main steps involved:

1. Prepare parameters: A makefile script prepares all compilation parameters taking into account microcontroller, port, bauds, etc.
2. Compilation: The makefile script invokes the Arduino compiler to convert the ".c" source code to a ".o" object.
3. Linking: The linker module looks for the libraries used in our source code and link them to our compiled code to generate a binary file that we can upload to the Arduino board.
4. Upload and execution: Once the binary file is created, we upload it to the Arduino board to be executed.

Other interesting characteristic is the possibility to migrate with few simple steps to a traditional Linux server. Therefore, Raspberry Pi is a really great choice in order to deploy a web sensor [10]. In addition, the architecture (Figure 1) adds one WebCam so that the student can visually observe the physical actions that their instructions are causing on the equipment.

This last point is very important because it allows the student to connect the virtual instructions with the physical equipment by visualizing the testing, which also helps to understand and to solve issues more easily. Moreover, it helps the student to get closer to the laboratory and avoid the feeling of using just virtual components instead of real instrumentation.

Regarding user experience, the time needed to compile the source code written by the student depends on the length of the code and the libraries used, but most of the programs that our students have to compile take around three seconds to be compiled. This compilation time is the same one that it may take when using a real Arduino board attached to our own computer instead of using a remote lab.

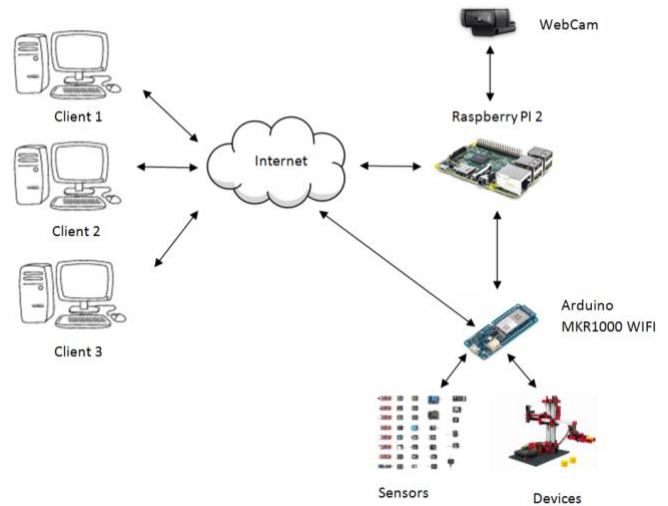


Fig.1 Remote lab architecture [3].

Once the code is compiled, it must be uploaded to the board and enable the device to start its execution. This process takes approximately another additional three seconds. Analogously, if we have a real Arduino board connected to our computer, the execution time required when uploading the binary code to the board is approximately the same. Thus, the student does not have any inconvenient from the time-to-execution point of view.

From the latency perspective, once the code is being executed, there is a 500ms delay in the webcam broadcast transmission. However, this is not very relevant and it does not affect the practice execution, since the proposed practices to the students do not make use of any input mechanisms, it only make use of visual outputs, such as LEDs or LCDs displays. Thus, there is no real delay from the student point of view; the practice execution is visualized smoothly.

Finally, regarding the support of concurrent users, within the current architecture only one student can concurrently execute code in each experiment, the rest of students have to wait in a queue to gain access.

IV. DESCRIPTION OF THE EXPERIMENT

The number of experiments that the students can perform with this remote lab is very diverse and different. All of them are Arduino-based remote experiments that can be accessed by any web browser (Firefox, Chrome, Opera, etc.) without the need of installing extensions or any other software. Even more, it is multi-platform and students may access from Windows, Linux or Mac operating systems. In fact, there are no platform, hardware or software entrance barriers and it is not necessary to use a high performing computer to have a good user experience. In this section, the experiments are explained in detail.

The user interfaces and available operations within the experiments are very simple, so that anyone could use this program without much difficulty. The laboratory website is divided in two parts. On the left side, there is a code section where the students can write their Arduino code (Figure 2).

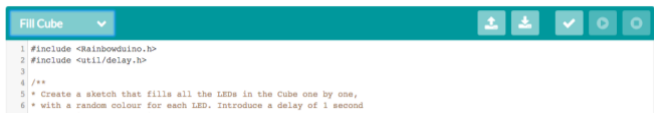


Fig.2 Details of the left side of the remote lab GUI.

There are some buttons to allow the student to compile the code and execute it on the remote Arduino board. This area of the GUI also includes a drop down menu with some predefined usage examples of the particular experiment, as it can be seen in Figures 3 and 4.

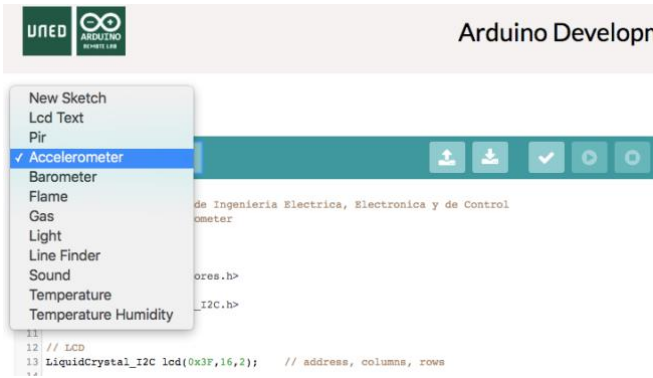


Fig.3 Details of the drop down menu with code examples for the sensors lab that are ready to be loaded in the editor



Fig.4 Detail of the drop down menu with some example codes for the LED cube lab ready to be loaded in the editor.

On the right side, students find the webcam streaming, which allows them to see the impact of their code execution in the instrumentation as part of the experiment (Figure 5).

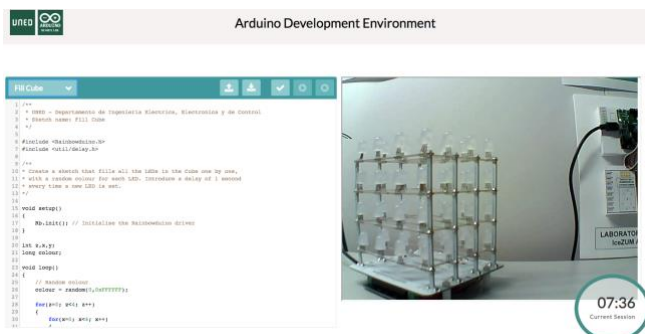


Fig.5 GUI of the 3D LED cube experiment in the Arduino remote laboratory.

The execution workflow only has three steps: upload code, compilation and execution in the board. Thus, the process is simple and it is an appropriate approach to introduce novel students to Arduino programming without the need of buying any equipment.

A. 3D LED cube

The first experiment that we describe is a 3D LED Cube. This experiment consists on a set of 64 RGB LEDs arranged in a 4x4 cube structure (Figure 6).

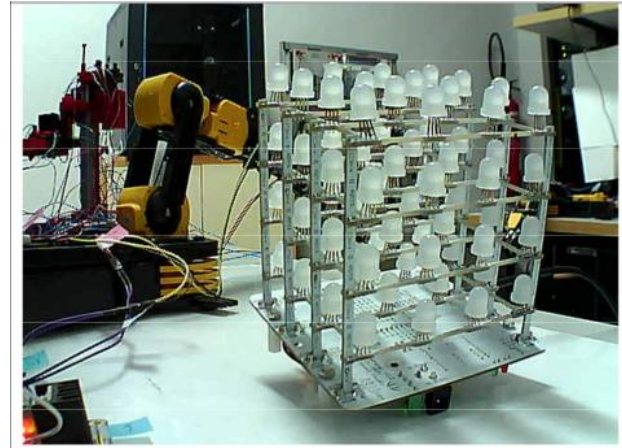


Fig.6 3D LED Cube

Students can write Arduino code to light up or turn off each one of these LEDs having in mind that they can program the color and intensity of each LED at any time. This allows them to create colorful animations based on defined loops.

The competences that students acquire thanks to this remote experiment are the following:

- Students learn how to interact with Arduino GPIO analog pins. Students have to define the intensity and color of each one of the LEDs connected to the GPIO analog pins.
- Students improve competences related to conditional and loop statements. They have to manage 64 RGB LEDs that are organized in a tridimensional arrangement. This requires working with loops, since it is not practical to interact with each LED independently. The creation of animations of several seconds with different colors and intensities makes the student understand the best way to manage the LEDs in groups.
- Students improve their spatial thinking. The fact that they have to work in a tridimensional arrangement makes the students think how to manage the LEDs on the X and Y planes to create meaningful animations.

Some of the practices proposed to students based on this 3D LED cube are the following:

- Random: Create a sketch that fills all the LEDs in the cube one by one, with a random color for each

LED. Introduce a delay of one second every time a new LED is set.

- Countries: Display sequentially several countries flags changing them every four seconds.
- Planes: Define a function to switch all the LEDs in a given Z plane with a desired color. Then define a similar function to switch on all the LEDs in a given Y plane. Using both functions, students must develop a sketch where a blue Z plane go through the whole cube sequentially, that is, first $z=0$, then $z=1$, $z=2$ and finally $z=3$. Only one plane can be set at a given moment. Afterwards, following the same rules, students must define a red Y plane going from $y=0$ to $y=3$. Finally, students must introduce a delay of two seconds every time the plane changes.
- Rainbow: Define an array with the seven colors of the rainbow. Initially the core of the cube (all inner LEDs) must be set to red and the exterior (all external LEDs) to purple. Every two seconds the color in the core is transferred to the exterior and the following color in the array sequence fills the core.
- Spiral: A spiral is a curve, which emanates from a central point, getting progressively farther away as it revolves around the initial point. In this practice, the student must create a three-dimensional inverted blue spiral that emanates from a far point creating a curve to reach the central point. The spiral should be created from the origin point (0,0), representing the Z and Y axis respectively and filling the whole X axis. That means, when the spiral is born, a line in $z=0$, $y=0$ is created, switching on all the LEDs in the X axis. Then the spiral is extended to $z=1$, $y=0$, setting again all the LEDs in the X axis at that point and so on, until the central point is reached and the whole cube is set. Afterwards, a new spiral appears from the same initial point (0,0) shutting off the X-axis at that point and repeating the same process until the whole cube is switched off.

B. Sensors lab

The second experiment that we describe is an Arduino Sensor Lab (Figure 7). This remote lab allows students to learn how to program the most popular sensors typically used for Arduino, such as:

- Temperature sensor. The sensor used in this project is already built-in within the experiment, since it includes the LM358 operational amplifier and a $10K\Omega$ resistor. The TTC03 thermistor is used as the temperature sensor, which has a measurement range between -40°C and $+125^\circ\text{C}$ with an accuracy of $\pm 1.5^\circ\text{C}$.
- Pro temperature and humidity sensor. The temperature and humidity sensor pro uses the DHT22 sensor. This is a low cost digital sensor used to measure humidity and temperature. It uses a capacitive sensor for the

measurement of humidity and a thermistor for measuring the temperature. It internally incorporates a chip that performs an analog to digital conversion, so that the sensor provides only a digital output. This is an advantage over other sensors that have an analog output, since this way voltage fluctuations are avoided. The disadvantage of this sensor is that the minimum sampling time is two seconds.

- Passive Infrared (PIR) module. It is an infrared sensor that is able to detect both people and objects. It is widely used in the field of electronic security systems to detect for example the passing of a person to any area previously programmed.
- Light sensor. It is a module that is ready to work directly with it, since it is composed of a voltage divider that uses the light sensor Light Dependent Resistor (LDR) GL5528. The resistance of this sensor decreases when the ambient light increases; the resistance of the LDR lies between 10 and $20K\Omega$ when there is 10 lux and $20M\Omega$ when it is in a dark environment.
- Flame sensor. It is composed of the sensor DFR0076 that detects wavelengths between 760 and 1100 nm of light. It has a measurement range of 60° , and a fairly high sensitivity.
- Microphone module with amplifier. The module consists of a microphone, the LM358 amplifier and a potentiometer. This is used to adjust the gain of the amplifier, which ranges between 20 and 200. The module has an analog output that in Arduino will offer a value between 0 and 1023, and is fed with an input voltage between 4 and 12V.
- MQ2 gas and smoke sensor. This module for the detection of gas and smoke is composed of the MQ2 sensor, which is used to detect concentrations of compressed natural gas; however, it can also detect propane, liquefied petroleum gases, smoke, isobutane and alcohol. It is a sensor with a high sensitivity and a high response speed. The internal operation of the sensor is based on the fact that when it is in an atmosphere with clean air, the internal conductivity is very low; while if it is in an atmosphere where there is a concentration of the gas to be measured, the conductivity of the sensor increases as the concentration of the gas increases.
- PING ultrasonic sensor. The ultrasonic sensor to be used in the project is the TS601P01 sensor. This electric rangefinder has a range of measurement between 3cm and 3m with an accuracy of $\pm 2\text{cm}$. With the implementation of intelligent processing technology through software you can get an improvement in the measurements of the sensor, as well as improve the ability to avoid possible interferences. The generic operation of an ultrasonic sensor is similar to a radar. The sensor emits a pulse of high frequency sound, imperceptible to the human ear,

and it measures the time from the pulse being emitted and bounced back from the surface to the origin.

- **ADXL335 accelerometer.** The ADXL335 sensor is installed in a module that incorporates the necessary pins for the connection with Arduino. This module has six pins, although only five of them are used; which are corresponding to the input voltage of 3.3V (Vcc), ground (GND), X-axis output signal (X), Y-axis output signal (Y) and Z-axis output signal (Z).
- **Real Time Clock.** For projects that work with sensors, it is very important to implement an RTC, since if you want to store the data from the sensors you have to have a temporal fingerprint to be able to perform a study of them afterwards. Therefore it has been decided to describe how this temporary footprint could be programmed for future extensions of the work.
- **Barometric sensor.** It is a module designed to be used directly with Arduino, since it incorporates the circuit for signal conditioning and the outputs and inputs necessary to work with Arduino. This module uses the sensor BMP005, manufactured by Bosch, which allows measuring the barometric pressure and temperature. It is currently one of the lowest cost sensors with greater precision than others found in the market. The conditioning circuit allows the power supply of the module to be made with the 3.3V or 5V of Arduino, and these inputs are used to obtain the 1.8V or 3.6V voltage needed to feed the BMP005 chip. It has a range of measurement of barometric pressure between 300 and 100hPa (90000m and -500m above sea level), with a resolution of 0.03hPa (0.25 meters). In temperature can perform measurements between -40 ° C and + 85 ° C, with a resolution of ± 2 ° C. The BMP005 chip has been manufactured using piezo-resistive technology to have greater protection against electromagnetic fields, achieving high precision, linearity and stability.

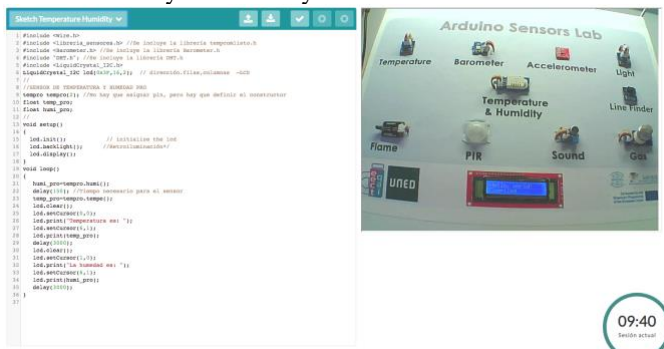


Fig.7 Arduino Sensor Lab interface.

In addition to all these sensors, the experiment includes a LCD screen where the student can write the data from the sensor after the reading. Thus, the result of the execution can be observed through the webcam and students understand, again, that they are using real equipment, not a simulation.

In this case, because students need to learn how to use each sensor, the learning curve is higher than the LED cube lab: its functionality, its libraries and how to read information and process it.

The proposed practices for this remote experiment are incremental. Thus, firstly, students are asked to create a simple function that writes a string in the LCD screen. Once, students have a clear idea of how to write in the LCD screen they can start working with the sensors. Then, they start working with each one of the sensors independently. The operation mode with only one sensor is the following:

1. Setup sensor libraries.
2. Write the code to read from the specific GPIO port where the sensor is connected.
3. Read raw data from sensor.
4. Use the sensor library to translate the raw data to human understandable information.
5. Write the produced information in the LCD screen.
6. Go to step 3 in the loop.

Once the current sensor has been understood, then the student can move on to another sensor. As a result, students would initially become proficient with each sensor separately, to afterwards start mixing functionalities of several sensors at the same time.

The competences acquired by the students after working on this remote laboratory are the following:

- Students learn how to interact with Arduino GPIO analog pins by reading from the sensors connected to these pins.
- Students acquire understanding of the operation mode of the most popular sensors used for Arduino programming
- Students learn how to write in an LCD screen.
- Students improve their programming skills regarding loops and conditional statements.

V. CONCLUSION

The use of remote labs in engineering education is becoming more important everyday because of the need of providing access to acquire practical competences with real instrumentation at anytime and anyplace, not only within the classroom.

This paper has analyzed the state of the art of remote labs for Arduino programming and described the one proposed by the authors. This remote lab includes several experiments, such as a 3D RGB LED cube and a sensors lab. These experiments are being used in an Electronics Engineering Degree in a Distance Learning University as part of the mandatory practices of a class.

Regarding the limitations of this remote system, we can remark that the number of available USB ports limits the

number of Arduino boards connected to the server. However, since the proposed server is Raspberry Pi, a very cheap and cost-effective Linux board with 4 USB ports (in the Raspberry Pi 3 model B), it is possible to have several servers each one with several experiments at a low cost. Moreover, since the literature shows that the use of remote lab increases students' motivation with respect to the use of simulations, the low inversion in this equipment is absolutely justified.

As future work, new experiments will be developed based on this remote lab. New Arduino boards will be connected to the Raspberry Pi server with new educational peripherals, such as a robotic crane, an IoT lab, and image recognition labs, among others.

VI. ACKNOWLEDGMENTS

This work has been co-funded by the Madrid Regional Government, through the project e-Madrid-CM (S2018/TCS-4307). The e-Madrid-CM project is also co-financed by the Structural Funds (FSE and FEDER). Authors also want to acknowledge the support provided by the project IoE-EQ (2017-1-IT01-KA202-006251) and the UNED Industrial School with the grant 2020-IEQ-13. J. A. R-V wants to acknowledge support from the Spanish Ministry of Economy and Competitiveness through the Juan de la Cierva Formación program (FJCI-2017-34926).

VII. REFERENCES

- [1] Viegas, C., Pavani, A., Lima, N., Marques, A., Pozzo, I., Dobboletta, E., Atencia, V., Barreto, D., Calliari, F., Fidalgo, A., Lima, D., Temporão, G., Alves, G., "Impact of a remote lab on teaching practicum and student learning", *Computers & Education*, vol. 126, pp. 201-216, Nov 2018.
- [2] Liu, D., Valdiviezo-Díaz, P., Riofrio, G., Sun, Y.M., Barba, R., "Integration of Virtual Labs into Science E-learning", *Procedia Computer Science* 75 (2015) 95 – 102, Nov 2015.
- [4] Tawfik, M., Sancristobal, E., Martin, S., Gil, R., Diaz, G., Colmenar, A., Peire, J., Castro, M., Nilsson, K., Zackrisson, J., Hakansson, L., Gustavsson, I., Virtual instrument systems in reality (VISIR) for remote wiring and measurement of electronic circuits on breadboard. *IEEE Transactions on learning technologies* 6 (1), 60-72, 2012.
- [5] Tawfik, M., Sancristobal, E., Martin, S., Diaz, G., Peire, J. and Castro, M., Expanding the boundaries of the classroom: Implementation of remote laboratories for industrial electronics disciplines. *IEEE Industrial Electronics Magazine* 7 (1), 41-49, 2013.
- [6] Sancristobal, E., Pesquera, A., Orduña, P., Ruiz, E. Gil, R., Martín, S., Diaz, G., Albert, M., Colmenar, A., Meier, R., and Castro, M., Virtual and remote industrial laboratory: Integration in learning management systems. *IEEE Industrial Electronics Magazine* 8 (4), 45-58, 2014.
- [7] Joya-Guirado, L., Martin, S., and Castro, M., Architecture of an Open Remote Lab for Digital Electronics. *Dyna* 91 (6), 599-600, 2016.
- [8] Bonilla, M.F.Y., Gutierrez, S.M., Orueta, G.D., Gil, M.A.C., Risk analysis and recommendations for electronic design with Arduino. *DYNA* 92 (6), 607-608, 2017.
- [9] Cubillo, J., Martin, S., and Castro, M., Design of a 3D Printing Remote Lab. *DYNA* 92 (1), 15-15, 2017.
- [10] Gutierrez, S.M., Guirado, L.J., and Gil, M.A.C., *DYNA Ingenieria e Industria* 91 (6), 599-600, 2016.
- [11] Fernández-Pacheco, A., Martin, S., and Castro, M., Implementation of an Arduino Remote Laboratory with Raspberry Pi. 2019 IEEE Global Engineering Education Conference (EDUCON), 1415-1418, 2019.
- [12] Jamieson, P., Arduino for Teaching Embedded Systems. Are Computer Scientists and Engineering Educator Missing the Boat?. Universidad de Miami.
- [12] Balamuralithara, B., Woods, P. C., "Virtual Laboratories in Engineering Education: The Simulation Lab and Remote Lab". *Computer Applications on Engineering Education*, August 2018.
- [13] Balogh, R., Educational Robotic Platform based on Arduino. Proceedings of the 1st international conference on Robotics in Education, RiE2010.
- [14] Grodotzki, J., Ortelt, T.R., Tekkaya, A. E., "Remote and Virtual Labs for Engineering Education 4.0: Achievements of the ELLI project at the TU Dortmund University" *Procedia Manufacturing*, Volume 26, 2018, Pages 1349-1360.
- [15] Čolaković, A., Hadžialić, M., "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues", *Computer Networks* 144, July 2017.
- [16] G. Carro et al., "The color of the light: A remote laboratory that uses a smart device that connects teachers and students," 2014 IEEE Global Engineering Education Conference (EDUCON), Istanbul, 2014, pp. 854-860.
- [17] J. Sarik and I. Kymissis, "Lab kits using the Arduino prototyping platform," 2010 IEEE Frontiers in Education Conference (FIE), Washington, DC, 2010, pp. T3C-1-T3C-5.
- [18] A. Mejías Borrero, M.A. Márquez Sánchez, J. M. Andújar Márquez, M. R. Sánchez Herrera, A Complete Solution for Developing Remote Labs, *IFAC Proceedings Volumes*, Volume 46, Issue 17, 2013, Pages 96-101.
- [19] Martin Kalúz, Ľuboš Čirka, Richard Valo, Miroslav Fikar, ArPi Lab: A Low-cost Remote Laboratory for Control Education. *IFAC Proceedings Volumes*, Volume 47, Issue 3, 2014, Pages 9057-9062.
- [20] A. Cardoso, V. Sousa, P. Gil. Demonstration of a remote control laboratory to support teaching in control engineering subjects. *IFAC-PapersOnLine* Volume 49, Issue 62016, Pages 226-229.