Remote laboratory experiments of Analog Electronics based on 'RedPitaya'

Carlos J. García-Orellana, Miguel Macías-Macías, Horacio González-Velasco,
Antonio García-Manso and Ramón Gallardo-Caballero
Departament of Electrical Engineering, Electronics and Automatics
University of Extremadura
Badajoz - SPAIN
Email: cjgarcia@unex.es

***Abstract*—Virtual teaching is becoming relevant in the offer of many Universities. 'Virtual Campus' platforms allow the distribution of contents and the communication with students. Nevertheless, part of the problems are found when we want to carry out laboratory experiments. These experiments play a fundamental role in Electronics teaching, and although they can be complemented with simulation, they can not be replaced. The alternative proposed in this work is the use of remote laboratories. Particularly, we present the development of a remote laboratory experiment of Analog Electronics using a system known as 'RedPitaya'. The student can interact with the previously prepared assembly with an environment adapted to this aim.**

I. Introduction

Practical laboratory experimentation in the University studies are based on traditional labs (real) and, generally, are costly in time and money. Besides this labs need the maintenance of very delicate infrastructures. On the other hand, this methodology requires the physical presence of the students and imposes an inflexibility of schedules. These factors can, in a lot of situations, cause students leave university studies.

There are solutions for these problems called remote and virtual laboratories or labs. A remote lab physically exists and can be manipulated remotely via internet, using web cams and specific hardware for local data acquisition. On the other hand, a virtual lab (remote or local) only uses computer software to recreate the behaviour of experimentation plants that only exist in computers used for their simulation [1].

The classification of laboratories depending on the form to access its resources (local or remote) and the physical existence of the system where the experiment is performed (real or virtual) can be summarized in the following table [2]:

	Real	**Virtual**
Local	In-class lab sessions with real plants	In-class lab sessions with simulated plants
Remote	Remote operation in a real plant	Remote lab with simulated plants

In detail, the use of remote labs has the following advantages over traditional teaching:

- *Economic*: It is not necessary to multiply laboratory workstations. With remote labs students can experiment with only one laboratory workstation.
- *Flexible working hours*: With remote labs students can experiment at different times using the same laboratory workstation.
- *ICT skill learning*: The use of remote labs facilitates, not only allows the learning of the skills of the different subjects, but also helps the training in cross-disciplinary skills (generic competencies), like the use of ICTs (Information and Communication Technology).
- *Constructivist learning*: A remote lab allows to use a constructivist learning methodology generating meaningful and autonomous learning.
- *Scientific interest*: This kind of technologies are experimenting an increasing interest by the scientific community. This has caused the increasing of communications related to remote labs in conferences and journals, even in areas with no relation to the subject of teaching and didactic.

Remote labs are based on a client-server environment, where clients (students) request services or contact with a web server. Generally this server must comply with the following guidelines [3]:

- *Scheduling*: Students must book within a pre-set timetable before performing the session lab.
- *Authentication*: Only registered users with an advanced booking can access to the session labs.
- *User interface*: The server based in a web browser must facilitate the communication between the client and the experiment.
- *Database*: Server must save information about users, timetables, profiles, results and reports.

In this sense, there are remote labs management tools based on Moodle [4] consistent with the Virtual Campus of our University. This Virtual Campus is used by a lot of professors of the Spanish Universities to manage our teaching [5]. There are several plugins for Moodle that allow the use of remote labs, like 'mod_OpenDesktop','mod_vpl' (*virtual programming lab*) and 'mod_ejsapp' (*easy javascript simulations*).

The use of remote labs in many areas of science and engineering has increased over the last years. Important examples are: Electrical Engineering [6], Analog Electronics [7], [17], [18], Robotics [8], Automation and Control Engineering [9], Physics [10]. In Spain, there are very interesting works and groups with wide experience in this area like UNILabs [11] developed by UNED (Distance-Education National University) and WebLab-Deusto [12] developed by the University of Deusto. Both platforms are integrated by a network of universities that share their laboratories. At international level there are significant projects like the Global Online Laboratory Consortium (GOLC) [13], the remote lab network (RexNet) [14] and, specially, the iLab platform [15] developed by the Massachusetts Institute of Technology (MIT). The increasing interest in this area is also shown by the financing of European research rojects, MARVEL [16], PEARL [19].

In the Analog Electronics case, there are several examples of remote labs, from which we can highlight ISILab [17]. In ISILab lab, sessions are previously assembled and students interact with the instruments available, but they does not change the electronic circuit. On the other hand, we have those based in VISIR [7], [18], in which the students can even build the electronic circuit (within given limits) by using a switching matrix controlled by a web interface. Most of analog electronic remote labs have some common features: they use instruments with advanced connectivity (GPIB, LXI, PXI, ...) and in a lot of cases they use 'LabView' as a control software. These characteristics result in a higher costs of each remote lab session.

Other works are focuses on the use of low cost systems for the web servers of the labs and experiments . Specifically in [19] the authors describe ArPi Lab, a fully operational remote lab at a cost of around 1000 Euros, most of which were invested in web cams used to record the experiments. This system consists of a Raspberry Pi as lab server and Arduinos YUN and UNO with Ethernet shields as experiment servers.

In this work we present an analog electronics remote lab based on 'RedPitaya'. We do not describe the part related to the authentication and scheduling. In Section II we will show the most important characteristics of 'RedPitaya' and the structure of its 'ecosystem'. In Section III the details of the implemented session labs will be described. Finally, the conclusions and future work will be exposed in Section IV.

II. 'RedPitaya' and its 'ecosystem'

'RedPitaya' [20] is a hardware board with a size slightly larger than a credit card, designed to be used as a measurement and control instrument. It can be accessed through ethernet, and it is devised to be used with a web browser as interface. For the capture and generation of signals, both high and low speed ADCs and DACs are included, which allow its interaction with the environment in lots of applications, with SMA-F standard connectors. On the other hand, it includes a dual-core ARM which allows to run a version of Linux operating system for the management of all the system, and also includes a high performance FPGA with which the more demanding digital signal processing is performed. Finally, an environment for the development of our own applications is provided, what turns it into a very versatile platform, ideal for the development of an autonomous system for laboratory experiments.

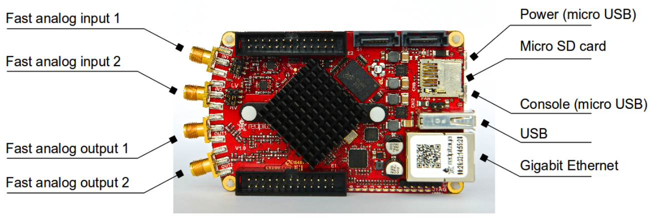

Fig. 1. RedPitaya, with the location of its fundamental elements.

First, the hardware included in the platform will be described in detail, what will allow us to discover the capabilities that it provides. Later, we will analyze the open-source software freely distributed with the device, to finish presenting the methodology that must be followed in the development of our own applications (or in the modification of the existing ones).

Hardware description

As mentioned above, the hardware of 'RedPitaya' consist of a board with dimensions similar to those of a credit card (107 x 60 x 21 mm), whose image is shown in figure 1. The system is built around a *system-on-chip* (SoC) Zynq 7010 by Xilinx [21], which includes, in the integrated circuit itself, an ARM processor and a FPGA. This makes it completely programmable, both the software run by the processor and the hardware implemented in the FPGA.

As described in [21], this SoC comprises two parts: the processing system (PS) and the programmable logic (PL), both separated, and interconnected by a high-speed interface. The processing system has a dual-core ARM Cortex A9 working at 667 Mhz as its main component, and provides the computer over which both the operating system and the applications are run. Although the SoC itself has some on-chip double-port RAM memory, accessible also by the FPGA, the main memory of 512 MB (DDR3 RAM, 4 Gb) is on an external chip, connected through the specific memory interfaces. In addition to the processor, the PS also includes input/output peripherals and interfaces for standard buses: two up-to-gigabit ethernet medium access control (MAC) peripherals, two USB 2.0 OTG ports, two SD/SDIO 2.0 controllers, two SPI ports, two high-speed UARTs (up to 1 Mb/s), two master and slave I2C interfaces, and several GPIO banks, that can be made accessible by the programmable logic or from the outside. This way, the system has a great capability of connecting to external devices, through the connectors described in figures 1 and 2.

On the other hand, in the section of programmable logic a hardware equivalent to a FPGA Artix-7 is included, with 28000 logic cells (17600 LUTs), RAM memory blocks up to 240 KB, and 80 blocks specific for DSP. In this FPGA, the interface with the high-speed ADC and DAC converters is implemented, and allows to program other hardware processing of the signal (as digital filtering), for functions which require intensive computation or very high speed.

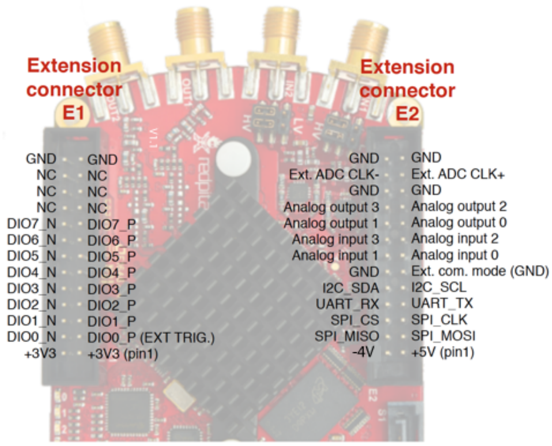

Fig. 2. Extension connectors of 'RedPitaya', where we can find slow analog inputs and outputs, I2C and SPI buses, and the digital inputs/outputs.

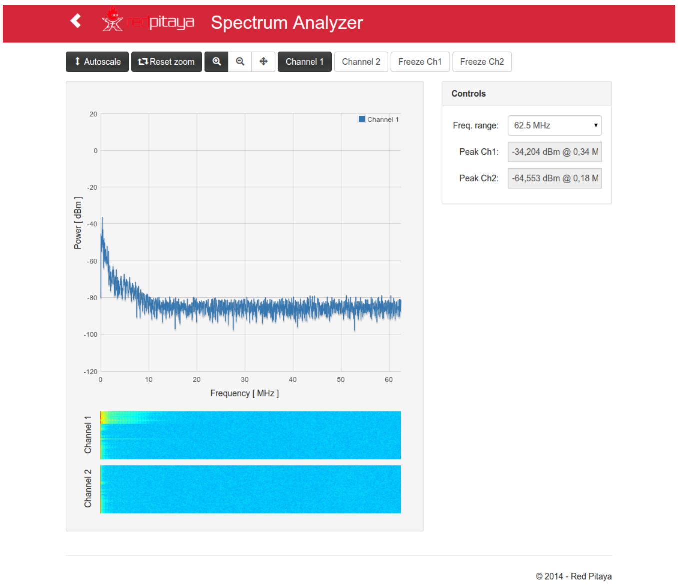

Fig. 3. Web interface of the spectrum analyzer application provided by 'RedPitaya'.

For the connection of analog signals 'RedPitaya' includes two high-speed analog-to-digital converters, that can sample up to 125 MS/s with a resolution of 14 bits. Those converters are connected to the FPGA that, after a first processing of the samples (digital filtering and trigger), sends them to the SP through the interconnection bus inside the SoC. The connectors used for these channels are SMA-F type, and can be seen in the figure 1. This way, an analog input with 1 MOhm/10 pF impedance is achieved, which allows a bandwidth of up to 50 MHz, with a maximum voltage scale of ± 1 V or ± 20 V (selectable with jumpers). Besides, 'RedPitaya' has four additional low-speed analog inputs, each capable of sampling at up to 100 kS/s and 12 bits of resolution, available through the extension connector.

Regarding the outputs, 'RedPitaya' also distinguishes between high and low speed. On one hand, two 14 bits DAC are attached to the corresponding SMA-F connectors, and are able to work at a rate of 125 MS/s, with a maximum voltage range of ± 1 V and an output impedance of 50 Ohm. On the other hand, in the extension connector we have four analog outputs with 12 bits of resolution and 100 kS/s, which can generate voltages in the range of 0 to 1.8 V.

Finally, 'RedPitaya' also has 16 digital inputs/outputs (extension connector E1) directly connected to the FPGA, and several LEDs that can be used as indicators, all of which are accessible also from the SP through registers mapped in the processor memory.

Included software

As we noted above, the producers of 'RedPitaya' have available in their web a SD memory image with a Linux distribution based on the official kernel maintained by Xilinx, in which the minimal system necessary to put the board to work is included. Once the SD card is introduced into the slot, and the board is connected to the Ethernet, the system will boot and obtain its IP address by DHCP. As it is important to know that address, because all the interface is carried out through a browser in other computer, 'RedPitaya' implements a system to associate your board with an account in their central servers (through the MAC address of the Ethernet), so that every time the board changes the IP, this is communicated to the 'RedPitaya' server, and every time we log into our account, we can consult it. Anyway, this is not completely neccessary, because the IP address can be known through a serial console.

When we connect through the browser to the IP of the board, we can see a list of the applications installed, and we can also install more applications or uninstall those which do not interest us any more. To do this, 'RedPitaya' offers, in their *"Application marketplace"* a series of free applications that include waveform generator, oscilloscope, spectrum analyzer, impedance analyzer or frequency response analyzer, and also offer applications developed by other members of the community. As an example, in figure 3 we can see the interface of the spectrum analyzer application, with one channel activated, and carrying out measurements.

Application development with 'RedPitaya'

In addition to providing specific applications, on the website of 'RedPitaya' we can also find information about how those applications are organized, and how we can develop other new ones by ourselves. The structure of the web applications consist of three basic elements, which we find outlined in figure 4:

- *Web interface*: This first part consist of a file programmed in HTML and JavaScript with JQuery, which the NGINX web server (a high-performance an low-memory-usage web server) sends to the browser (the client) when it asks to start the application. During the execution of the application, the client deals with two data types: signals and parameters, which are interchanged with the

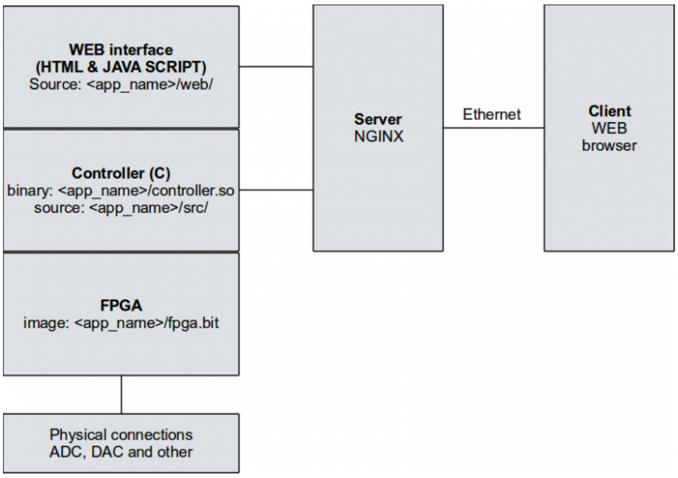

Fig. 4. Structure of the applications developed for 'RedPitaya'.

server using JSON, through GET or POST commands, and which are normally related to data that come from the FPGA, or with parameters to be configured on it. To manage this interchange, specific *indicators* or *location* are defined that, when asked by the client, they make the server to execute a determined module which controls the charge of the following element of the structure: the *controller*.

- *Controller*: is a set of C-programmed functions compiled as a dynamic library, which is loaded by the web server module when any communication with the FPGA is required. It can be, for instance, the transmission of signals captured by the ADC of the board, and that should be visualized in the client (after being processed by the FPGA), or the sending of configuration parameters from the browser to the FPGA itself, all of which is carried out by read and write cycles in concrete registers of it.
- *FPGA*: the control of the ADC and DAC devices, and the digital signal processing that requires higher performance should be done using hardware techniques in the FPGA. All this can be developed using hardware description languages, in the environment that Xilinx provides for this purpose, and with which a configuration file for the FPGA is finally generated. Each application (waveform generator, oscilloscope, spectrum analyzer) will require different configurations and initial processing of the signals, before obtaining the data the web interface has to represent, and so they have different programmings for the FPGA.

In general, when the tasks we want to carry out in our application are directly related to those carried out by the oscilloscope or the waveform generator, the most advantageous situation will be to take one of the already developed applications as a template, and to focus in the parts of web interface and controller, adopting the applications developed by 'RedPitaya' for the FPGA.

III. Work description

The main objective of this work is to evaluate the use of 'RedPitaya' as a single element for the implementation of our analog electronic remote lab. When we say, single element, we refer to we want to use 'RedPitaya' as a substitute of the whole instrumentation and the computer that manage this instrumentation.
In particular, we have focus on the implementation of the user interface with 'RedPitaya' to implement a single remote session lab composed by two electronic circuits based on operational amplifiers. In short, we have used an inverter amplifier and a unity gain buffer. This circuits can be observed in figures 5 and 6.
Both circuits have been implemented in a breadboard with a LM358 operational amplifier. In figure 7 we can observe the mounted circuit. Orange cables correspond to the buffer circuit and blue ones to the inverter amplifier.

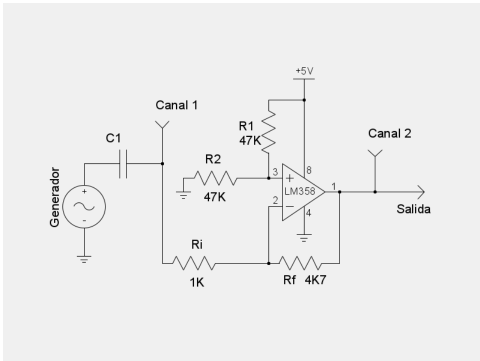

Fig. 5. Outline of the inverter circuit used in the remote laboratory experiment.

To make the 'pilot' test, we have changed an existing application of 'RedPitaya' and adapt it to our requirements. In our case, the application that better fit these requirements is 'Oscilloscope + Generator', that is included as free application with RedPitaya.
We have focused on adapt the web interface to can select one of the two implemented circuits (figures 5 and 6). To make that, we have connected the 'Output 1' of the generator and the channel 1 of the Oscilloscope to both circuits. In this way

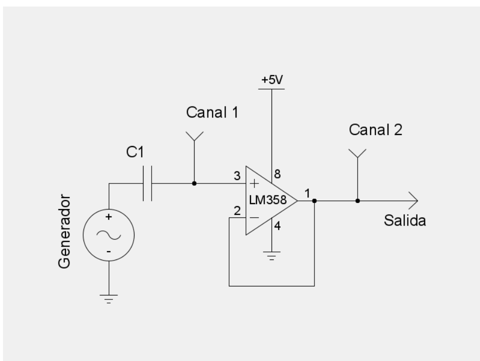

Fig. 6. Outline of the follower circuit used.

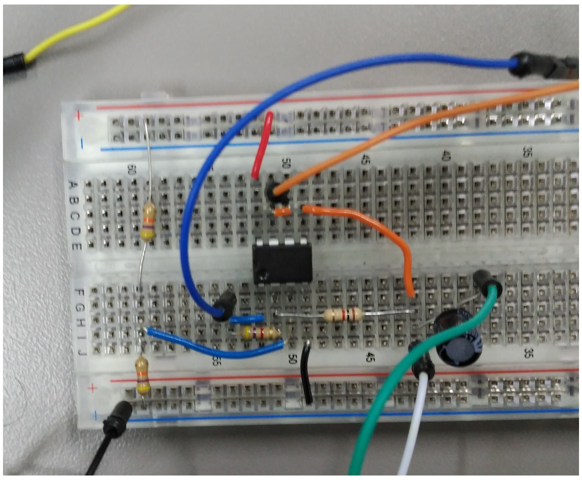

Fig. 7. Image with the arrangement of the circuits used in the laboratory experiment.

we don't have to switch between both circuits. Logically, the circuit outputs have to be switched and to make that we have use an electronic relay.
To drive this relay we can use one of the multiple I/O that have RedPitaya, however, to make more versatile our system, we have used a relays module controlled by a bus I2C. This allow us to use a lot of relays without consume the I/O of our RedPitaya.
In the end, according our approach, to address a remote session lab with RedPitaya we should need develop or change the following points:

- *Web interface*: We need to adapt (or create) the web interface of our application. We think that change an existing application is more convenient. As we commented in section II, this means HTML and JavaScript with JQuery programming and information exchange in JSON format.
- *Controller*:As we also commented in section II, the controller is the module of our application that communicate the FPGA and the Linux system of RedPitaya with the web interface through the web server. In this case we used an existing controlled and we modified it, mainly, to drive the relays module.
- *Control module of the session lab*: Now we are referring to the module that has to be developed for each session lab. In this example, this module has to control the relays module. This module could be more complex, for example, if we want to measure additional voltages by using the A/D converters of RedPitaya.
- *Changes on the FPGA*: This point is not needed in general, especially, if we want to use some of the standard applications. However, we could include new digital modules to make additional filtering of the signals, for example, to simulate a DC multimeter by using one of the analog inputs previously cited.

In the session lab presented in this work we only have modified the first two points. We also have developed an initial

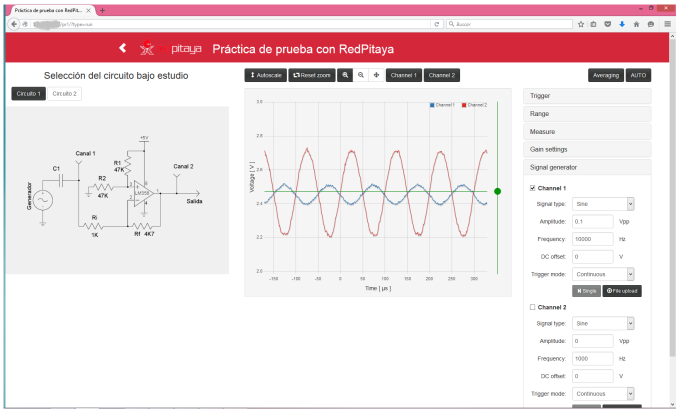

Fig. 8. Web interface of the remote laboratory experiment in which we can study the gain of the inverter amplifier.

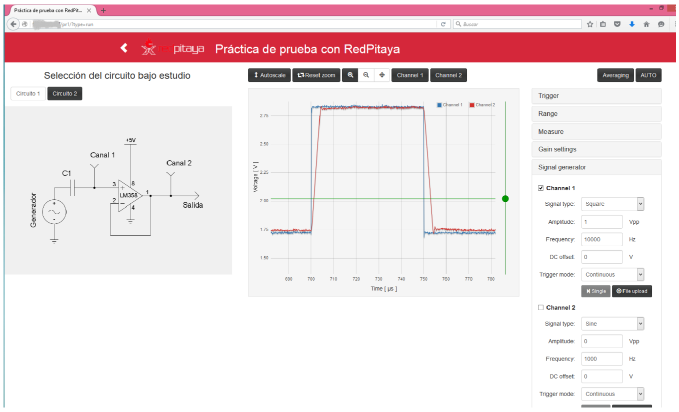

Fig. 9. Web interface of the remote laboratory experiment in which we can study the *slew-rate* of the follower.

version of the third point and we don't have modified the fourth. Each of this subjects are described below.

Web interface

The interface for the student to interact with the experiment is the *Web interface*. In order to analyze the two circuits of this pilot test we mainly need a waveform generator and an oscilloscope. For this reason, our interface shows the controls of the oscilloscope and the generator, along with the schematic of the circuit that is being studied. In figures 8 and 9 we can see that interface for each of the circuits.

As discussed above, we considered the 'Oscilloscope+Generator' application as a base, because it offers us an excellent starting point. It can be seen in figures 8 y 9 that we have added an image of the circuit an two buttons to be able to select among the circuits of the experiment.

Controller

This is the part which is run in the 'RedPitaya' and controls the access to the hardware (the FPGA mainly), collects configuration changes from the web interface and responds to data requests from that interface. In our case, we have had to include the control for the relays through the I2C bus,

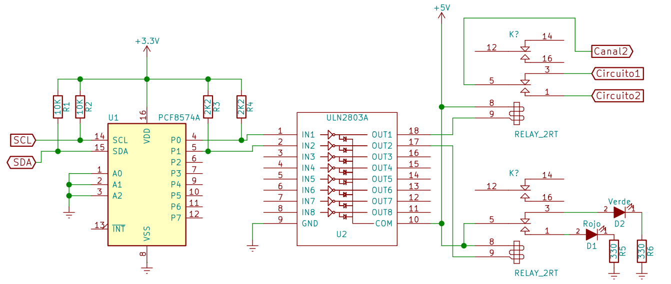

Fig. 10. Outline of the circuit to control the relays used in the remote laboratory experiment test. Two LEDs are used to indicate which circuit is active.

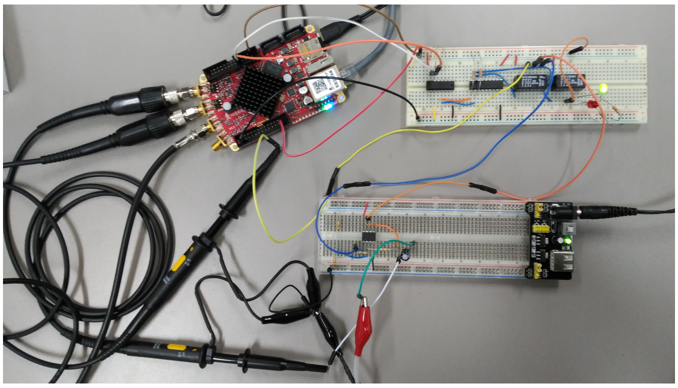

Fig. 11. Image of the global arrangement of this pilot experiment.

for which we have expanded 'configuration change' messages from the web interface, to be able to respond to the circuit change requests from that interface.

Experiment control module

This module is responsible for changing the circuit connections or modifying the measurement points. In the pilot experiment, we have reduced this module to the simplest case (only the change of 'Channel 2' of the oscilloscope), but maintaining the possibility of extending it substantially. To do this, we used an I/O expander based on the I2C bus, and connected to the 'RedPitaya' (terminals 9 and 10 of E2 connector). Particularly we used the chip PCF8574, which provides 8 I/O channels, along with the driver ULN2803 to manage relays. This configuration could be repeated 8 times (changing the I2C identifiers) to have a total of 64 I/O.

In figure 10 we have the schematic that we used to implement this module. In our case, we only needed two relays (though only one was used in the experiment, the other is only for the LEDs that show the active circuit).

In figure 11 we can observe the whole arrangement that we have made for this pilot experience. It can be observed the 'RedPitaya' together with the electronic circuits of the practice and the control module.

IV. Conclusions and Future Work

The main conclusion of this work is that 'RedPitaya' can be satisfactorily used as an integrated element (instrument+PC) for remote laboratory experiments of Analog Electronics. Nevertheless, it has also some downsides.

Among the benefits and points in favour, we could list the following:

- It is a compact, stable, open and economic system (it costs around 220 € without probes) that combines a highly configurable instrument and the computer to manage it.
- It provides good performance for laboratory experiments, combining oscilloscope and generator with A/D additional channels, digital I/O and specific buses.
- It allows to create environments adapted to the laboratory experiment that is to be conducted, creating the instruments needed for that concrete experiment.
- It can be controlled using Matlab, LabView, SciLab, Python, etc. Nevertheless, these methods make it necessary an external computer and we would lost many of the benefits of the platform (it would be another instrument in a computer-controlled laboratory).

Nevertheless, despite these significant advantages, we also have some drawbacks. Although some of them have already been discussed, we point out the following ones:

- 'RedPitaya' is not a laboratory instrument, and though it could be not a problem when conducting a laboratory experiment, we should not forget it.
- The adaptation of the parts of a 'RedPitaya' application to the specific laboratory experiment requires some effort and although once the ecosystem is known it can be easy, at first it could be tedious.
- The oscilloscope input range (± 1 V o ± 20 V) and the generator output range (± 1 V) are quite limited compared to a traditional oscilloscope or to a waveform generator. This restriction must be considered in the design process of the experiment.
- In our case, we have taken as a base an existing application (oscilloscope and generator), which has some disadvantages, for instance it does not have neither AC-DC selector nor cursors, the gain is common for both channels, etc.

Regarding potential future lines of work, we intend to improve the oscilloscope and generator application, adding different scales for the two channels and introducing cursors. Besides, on one hand we want to prepare a compact board with 8 to 16 relays to facilitate the implementation of more complete and systematic lab experiments and, on the other hand, we would like to add a 'multimeter' for measuring certain voltages or currents, using the additional A/D channels of the 'RedPitaya'

Acknowledgments

This work has been partially funded by the Junta de Extremadura and FEDER through the grant GR15101.

References

[1] Calvo I. *Laboratorios remotos y virtuales en enseñanzas técnicas y científicas*, Ikastorratza, e-journal of didactics, ISSN-e 1988-5911, No. 3, 2008.

[2] Dormido S. *Control Learning: Present and Future*. Annual Reviews in Control, vol. 28 (1), 2004 pp 115-136.
[3] Zamora R. *Laboratorios Remotos: Actualidad y tendencias futuras*, Scientia et Technica, year XVII, No. 51, 2012.
[4] Description of the *Moodle* platform [Online]. URL https://docs.moodle.org/all/es/Acerca_de_Moodle.
[5] Javier García-Zubia, Pablo Orduña, Jaime Irurzun, Ignacio Angulo, Unai Hernández-Jayo. *Integración del laboratorio remoto WebLab-Deusto en Moodle*. In MoodleMoot Euskadi, 2009.
[6] Pradyumna, P., Tarun, C., and Bhanot, S. *Remote experimentation of no-load tests on a transformer*, in electrical engineering. In IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA), 1-6. Kottayam, Kerala, India, 2012. doi:10.1109/AICERA.2012.6306740.
[7] Unai Hernandez-Jayo y Javier García-Zubia. *Remote measurement and instrumentation laboratory for training in real analog electronic experiments*, Measurements 82 (2016) 123-134.
[8] Prieto, G.A. and Mendoza, J.P. *Low cost didactic robotic platform based on player/stage software architecture and la fonera hardware*. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 8(3), 2013, 126-132. doi:10.1109/RITA.2013.2273112.
[9] Santana, I., Ferre, M., Izaguirre, E., Aracil, R., and Hernandez, L. *Remote laboratories for education and research purposes in automatic control systems*. IEEE Transactions on Industrial Informatics, 9(1), 2013, 547-556. doi:10.1109/TII.2011.2182518.
[10] Zimin, A.M., Korshunov, S.V., Shumov, A.V., and Troynov, V.I. *Remote access laboratories for training of engineers in the 21th century*. In 8th International Forum on Strategic Technology (IFOST), UNED, 2013.
[11] University Network of Interactive Laboratories, Nov. 8, 2013 [Online]. URL http://unilabs.dia.uned.es/.
[12] WebLab-Deusto research group (2013). WebLab- Deusto, Nov. 8, 2013 [Online]. URL https://www.weblab.deusto.es/web/.
[13] Web page of UNILABS. http://online-lab.org/.
[14] Web page of RexNet. http://www.rexlab.net.
[15] Web page of iLab. http://ilab.mit.edu/.
[16] Müller, D., Ferreira, J. *MARVEL: A Mixed Reality Learning Environment for Vocational Training in Mechatronics*. International Conference on Technology Enhanced Learning (TEL), Nov. 2003, Milano, Italy.
[17] A. Bagnasco, A. Poggi, G. Parodi, A. M. Scapolla, *A Configurable Remote Laboratory for the Flexible Setup of Experiments in Electronics*, in Advances on remote laboratories and e-learning experiences, edited by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, ISBN: 978-84-9830-077-2.
[18] J. Zackrisson, I. Gustavsson, and L. Håkansson, *An Overview of the VISIR Open Source Software Distribution 2007*, Proceedings of the 2007 REV Conference, Porto, Portugal, June 25 - 27, 2007.
[19] Cooper, M., Colwell, C., Amaral, T. *Accessibility and usability in complex web based learning applications: Lessons from the PEARL project* in Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, 2002, pp. 1358-1365.
[20] Web page of 'RedPitaya'. http://redpitaya.com.
[21] *Zynq-7000 All Programmable SoC Overview*. Xilinx, 2016 (available in http://www.xilinx.com).