Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

ArPi Lab: A Low-cost Remote Laboratory for Control Education

Martin Kalúz * **Ľuboš Čirka** * **Richard Valo** * **Miroslav Fikar** *

* *Institute of Information Engineering, Automation and Mathematics*
STU in Bratislava, Radlinského 9, 812 37 Bratislava, Slovakia
martin.kaluz@stuba.sk

Abstract: In this paper, we present a cost effective approach to remote experimentation. We propose ArPi Lab – remote laboratory for education in area of process control. This lab is built on very cheap hardware components, including single-board computers Raspberry Pi and open prototyping platforms Arduino based on 8-bit micro-controllers. This approach combines several different software technologies. These are HTML 5 and JavaScript for client-side application, PHP and MySQL for laboratory server implementation, JSON as structure for data transfer, and C language for experiment server and micro-controller programming. ArPi Lab provides three different types of educational physical systems. Three thermal plants, one magnetic levitation, and one hydraulic tank system are available for remote laboratory experiments. Each part of ArPi Lab's hardware architecture can be controlled in the meaning of power supply. For this purpose we propose an efficient power management model, designed to solve occasional hardware and communication failures in such kind of laboratories, where physical absence of supervising person can result in serious malfunctions or security issues.

Keywords: Remote laboratory, Control education, Raspberry Pi, Arduino

1. INTRODUCTION

Control education passed through a rapid development in recent years. A whole set of new methodologies of student-subject interaction were set by the upsurge of modern information technologies. This phenomenon has affected almost all fields of education and science, but mostly those within technical areas.

At the same time, new ways of practical laboratory experimentation were set, extending the approaches of hands-on practices to remote. Nowadays, the use of remote laboratories is considered as the standard for education in the electrical engineering (Pradyumna et al., 2012), robotics (Prieto and Mendoza, 2013), automation and control engineering (Santana et al., 2013), physics (Zimin et al., 2013), and many others. Several noteworthy projects of remote laboratories and their networks are available on the Internet. Very interesting remote labs were developed at the National Distance Education University in Madrid, Spain (UNED, 2013). The UNILabs (former UNED Labs) provide wide spectrum of virtual and remote instruments for control education and show the cooperation between five Spanish universities. Impressive network of remote laboratories hosted on different universities is also provided by the University of Deusto (WebLab-Deusto research group, 2013). These laboratories are mostly focussed on, but not limited to, electronics and robotics. WebLab-Deusto is the remote laboratory management system dedicated to develop, publish and share remote labs. Another renowned projects with high impact are e.g. MIT iLabs (Harward et al., 2008), LabShare (Lowe et al., 2009), and AutomatL@bs (Vargas et al., 2011).

Since the remote and virtual instrumentation became very popular, the whole branch of development has focused on low-cost devices. These involve the use of field programmable gate arrays (FPGA), complex programmable logic devices (CPLD), peripheral interface controllers (PIC), and various types of opened prototyping platforms. These platforms are based on embedded micro-controllers, and in recent years they spread all over the market. Most renowned of them are AVR based prototyping boards Arduino. Moreover, not only the micro-controllers are getting through rapid development these days, but also cheap alternatives to standard computers.

In related works, we often meet with solutions based on micro-controllers and small computers. In paper by Georgitzikis et al. (2012), the application of Arduino boards, extended by WiFi communication capability, is used to create the wireless sensor network as the part of the *Web of Things*. In similar manner, Ursutiu et al. (2010) have introduced an interesting way of Web instrumentation using Tag4M, the WiFi tag as the portable extension for renowned iLab framework. Al-Busaidi (2012) uses the Arduino Mega board as embedded control system for biped robot, and introduces an interesting communication method between robot and MATLAB environment. Another use of Arduino is described in Barber and Crespo (2013), where authors apply this micro-controller as a cheap hardware interface to link Simulink models to physical laboratory systems without need of expensive DAQ cards. Neto et al. (2012) use the same controller extended with Ethernet module to build remote laboratory for DC motor PID control, with network communication capability. Campos et al. (2012) show a low-cost platform for Web-based experimentation developed on PIC micro-controllers. They allow students to exercise with different electrical devices like light diodes, LED panels, switches, and servo drives. Very nice example of inexpensive experimentation setup is given in paper by Sobota et al. (2013), where the REXduino platform is introduced. This platform is based on Raspberry Pi computer equipped with REX Control System and connected to physical instrument by Arduino board.

Contrary to above mentioned works, we provide solution that differ in two general points. Firstly, those mentioned works, where authors presents strictly low-cost remote laboratories, are predominantly focused on electronics and its applications. Our solution is primary dedicated to process control and automation applications where the use of cheap micro-controllers and development platforms is still not common in the context of remote laboratories. Secondly, even the published works with control applications, where development expenses are low, still usually contain architectural parts like standard server computers and expensive interfacing devices. In our approach, all hardware components, even those operational like laboratory servers, are built on devices not exceeding the price of several dozens €.

In this paper, we introduce the remote laboratory *ArPi Lab* which is physically built on single-board computers Raspberry Pi and Arduino development boards. It uses several different software and communication technologies. Two types of HTTP services are introduced on two communication layers. The first is PHP powered Apache-based laboratory server on the top layer of architecture and second type is embedded HTTP server used for experimental nodes. Communication is served by the asynchronous (AJAX) and synchronous calls based on JSON. The storage of experimental data and configuration of nodes is served by the MySQL database located in laboratory server. A whole laboratory uses an advanced power management mechanism, which also described in this paper. At the end of the paper, we show an example of students' assignment performed in remote laboratory.

2. MOTIVATION

ArPi Lab is the main result of project named "Low-cost hardware architecture for implementation of remote control". This project is supported by the grant program "Young scientist" of Slovak University of Technology in Bratislava and it applies to PhD students and young scientists to financially cover up their research.

The fact that budget of project is €1000, has created an interesting challenge for us. We claimed to develop a fully operative remote laboratory with architecture capable to handle various physical laboratory experiments, and simultaneously we claimed to deploy 5 particular experiments designed for control education. Furthermore, if we speak about the cost of the laboratory, it is worth to mention that half of project's funds were spent on IP cameras, which are not considered as necessary part of architecture. This fact pushes the real expenses on architecture development somewhere to €500.

The main motivation is to prove that even with very limited financial resources, the cheap and effective development methods can be used to build realistic remote laboratories. Inexpensiveness can be considered as the main benefit of ArPi Lab. On the other hand, low costs are negatively reflected to development phase where creators of remote laboratory must work with the raw hardware which is, in many cases, literally featureless in its initial form. On the contrary, use of commercial ready-made technologies can significantly reduce the effort on development and implementation, but can also rapidly increase the price. These facts truly follow the well known argument "the more you pay, the more you get".

Another motivating challenge was to develop a laboratory which is architecturally opened for future extensions and changes. This goal can be fulfilled by using general purpose approaches. To simplify future development we tried to follow the idea of very popular *Plug and Play* concept (García-Zubia et al., 2008) as close as possible. ArPi Lab's architecture allows developers to connect any kind of experimental device controllable by common electrical signals, and simultaneously to provide their easy incorporation to laboratory just with few simple configuration steps. To sum up, ArPi Lab architecture is opened in the meaning of "connect and configure" and no further architectural or software changes are required.

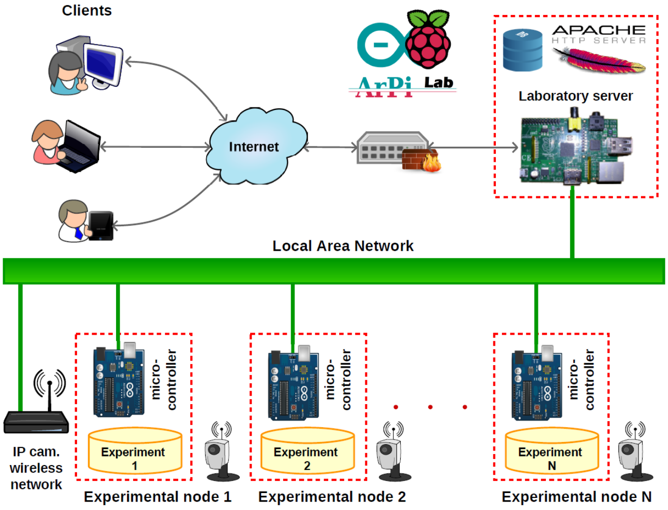


Fig. 1. Local architecture of ArPi Lab

3. ARPI LAB DESCRIPTION

As mentioned above, ArPi Lab is a general purpose remote laboratory designed for practical experimentation in automation and process control related education.

3.1 Hardware in General Purpose Architecture

The local architecture of ArPi Lab has the branched structure and it is shown in Fig. 1. The hardware devices used in architecture are: single board computer Raspberry Pi[1] as the Laboratory server and communication forwarder; micro-controller boards Arduino YÚN[2] and

---

[1] http://elinux.org/RaspberryPiBoard
[2] http://arduino.cc/en/Main/ArduinoBoardYun

Arduino UNO[3] equipped with Ethernet modules[4] as the experiment servers.

The laboratory server of ArPi Lab acts as the public communication gateway between clients and laboratory nodes. It performs the following tasks:

- provides the Web services for clients
- manages communication (forwarding)
- secures the private network (laboratory nodes)
- verifies the validity of transferred data
- provides administration/configuration of ArPi Lab

In the ArPi Lab, an Arduino micro-controller boards are used as devices for direct interaction with physical laboratory systems. To control them over electrical signals, Arduino UNO provides 14 digital pins that can be used as both, the inputs and outputs. Moreover, 6 of them can be configured to work in PWM mode (Pulse Width Modulation), to substitute analogue signals. All of the digital pins have 8-bit resolution, so they can emulate (or receive) signals in range of 0-255 in integer representation or 0-5 V in voltage. Arduino UNO also provides 6 pure analogue inputs that use 10-bit resolution (0-1023 or 0-5 V).

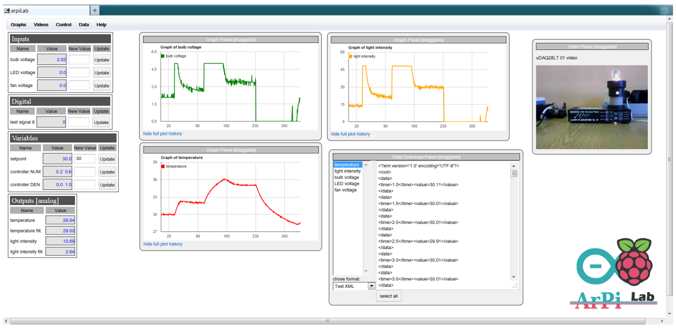
3.2 Software and Communication

All devices used in ArPi Lab architecture use a non-commercial open-source software. Client side implementation of ArPi Lab uses JavaScript powered Web application for direct interaction between the user and remote experiment. It is built as the JavaScript powered HTML5 Web page with semi-dynamic Document Object Model and event driven internal logic. In the comparison with other Web-based laboratories, the ArPi Lab provides a whole new concept of interface construction. This concept is called on-fly content generation. Graphical user interface (GUI) is show in Fig. 2 and it is unified for all experiments in laboratory. When a student enters lab, client side application loads configuration data for requested experiment from database and automatically constructs the user interface. It consists of interactive tables and draggable windows, so students can make their own layout for comfortable usage. GUI consists of:

- table of input signals and additional variables (view/update);
- table of output signals (view only);
- signal trends;
- video streams from IP cameras;
- control algorithm window;
- logging history window;
- data download window.

Communication principles applied in ArPi Lab are shown in Fig. 3. User's actions in client application are processed to the asynchronous HTTP requests, handling the data in JSON structures. They are sent to laboratory server and processed by a set of PHP scripts. Each request contains the authorization key which grants uses to access a particular laboratory node (experiment). Therefore, each request can be considered as the separate authorization procedure. Data received by laboratory server are parsed from JSON to a set of associative arrays and passed to validator. Request validator is a PHP script that compares authorization key with database and accordingly collects the configuration settings for the particular experiment. Then it checks if request satisfies the predefined structure of data and their correctness. Last part of data processing is the engineering unit conversion and addition of internal credentials (private node key). If this procedure is successful, the node settings and data are sent back to forwarder core which prepares JSON structure for experiment server. Internal data exchange between laboratory server and experiment server is also processed by HTTP request, but unlike the public layer, it uses a synchronous communication. Requests are sent from forwarder to embedded Web server running on ATmega328p, which also contains the algorithms for physical interfaces and control. The response with the process data is returned to client through the same services as the request.

Two types of data sampling from micro-controller to GUI can be chosen in ArPi Lab. The first is intended for systems with slow dynamics, using one data point per sample, and the second one for fast systems, using batch data acquisition. For example, if control sampling of experiment node is 1 millisecond and data sampling for GUI is 500 milliseconds, then one batch of data contains 500 points.



Inputs

Name	Value	New Value	Update
bulb voltage	2.02		Update
LED voltage	0.0		Update
fan voltage	0.0		Update

Digital

Name	Value	New Value	Update
test signal 8	0		Update

Variables

Name	Value	New Value	Update
setpoint	30.0	30	Update
controller NUM	0.2 0.6		Update
controller DEN	0.0 1.0		Update

Outputs [analog]

Name	Value
temperature	28.64
temperature filt.	29.03
light intensity	13.69
light intensity filt.	2.64

```
<?xml version='1.0' encoding='UTF-8'?>
<root>
<data>
<time>1.0</time><value>30.11</value>
</data>
<data>
<time>1.5</time><value>30.01</value>
</data>
<data>
<time>2.0</time><value>30.01</value>
</data>
<data>
<time>2.5</time><value>29.91</value>
</data>
<data>
<time>3.0</time><value>30.01</value>
</data>
<data>
<time>3.5</time><value>30.01</value>
</data>
```

Fig. 2. Unified user interface of ArPi where students can perform their tasks

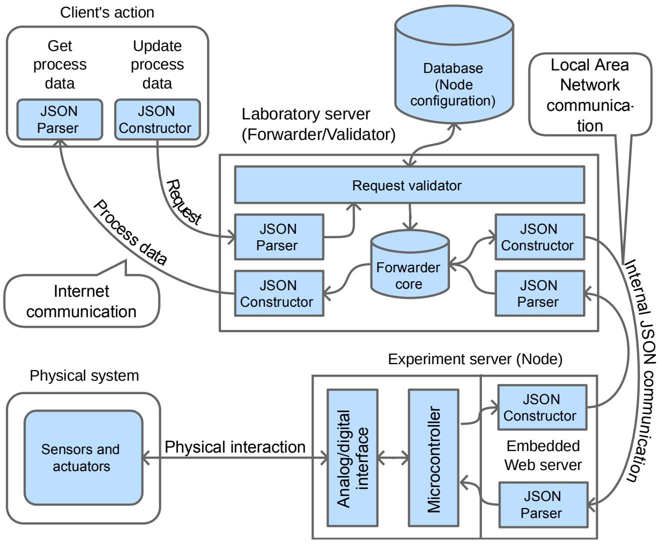


Fig. 3. Communication between client and experiment

---

[3] http://arduino.cc/en/Main/ArduinoBoardUno
[4] http://arduino.cc/en/Main/ArduinoEthernetShield

3.3 Physical Systems

Since the physical interface of Arduino allows both digital and analogue signals, any kind of laboratory device, that operates on 0-5 V can be directly connected to experiment server and used in remote laboratory. Additionally, for devices with interfaces operating on different voltages, wide spectrum of converters can be used. As mentioned in previous sections, we have implemented five remote experiments in ArPi Lab. These are represented by three different physical systems:

- thermal plant uDAQ28/LT (Fig. 4);
- magnetic levitation CE152 (Fig. 5);
- hydraulic tank system uDAQ28/3H (Fig. 6).

The most frequently used laboratory training process in education at our department is thermal plant uDAQ28/LT (Huba et al., 2006). In ArPi Lab we have connected three of these devices to provide possibility of simultaneous parallel student sessions. Another training device is a three-tank system (uDAQ28/3H, Huba and Halás (2011)). It allows students to work with three tanks, fed by pumps from main reservoir and separated by controllable valves.

ArPi Lab also contains CE152[5] magnetic levitation system, that can be used to teach control of unstable systems with very fast dynamics.

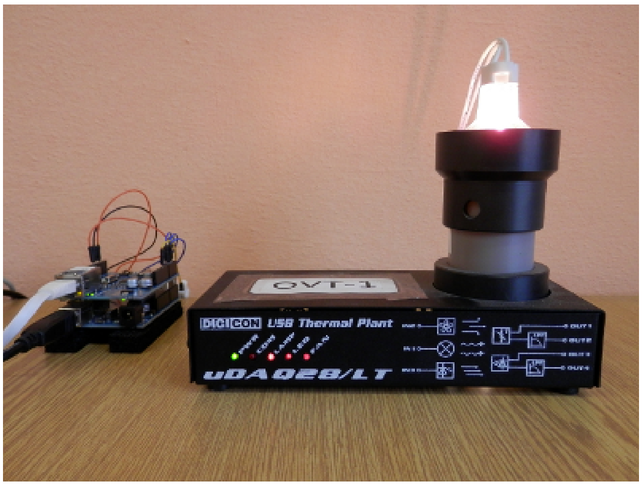


Fig. 4. Thermal plant uDAQ28/LT

3.4 Performance

The fast control loops can be achieved thanks to surprising computational performance of Arduino's embedded microcontrollers ATMEL ATmega328, which are used as experiment servers.

In ArPi Lab, each Arduino includes code for:

- Web server emulation;
- serving Ethernet communication;
- JSON parsing and construction;
- signal interfacing (I/O read/write);
- several switchable controller algorithms.

Even despite the fact, that all listed tasks are performed in single thread loops, we were still able to achieve sampling period of micro-controller at approx. 5 ms. This sampling is sufficient for most of process control related systems.

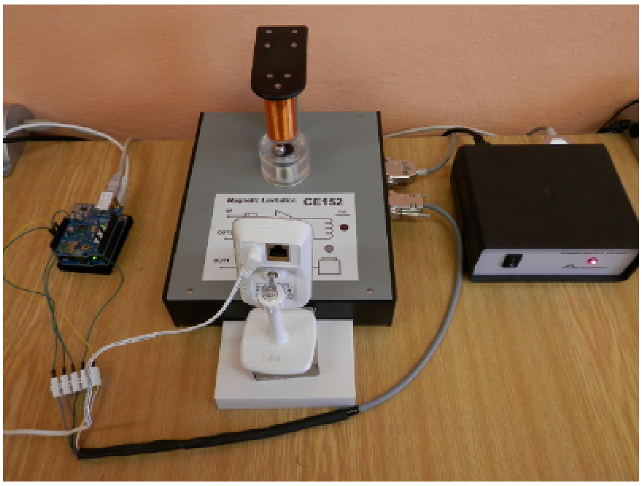


Fig. 5. Laboratory model of magnetic levitation

Fig. 6. Hydraulic tank system uDAQ28/3H

To achieve faster sampling, which is in our case required by magnetic levitation model CE 152 (less than 2 ms), we have used a different type of development board, an Arduino YÚN. This board contains additional ARM processor that runs light-weight Linux distribution Linino and internal communication bridge to ATmega32u4. In this particular case, processing of communication between laboratory server and experiment server is performed in ARM processor, leaving ATmega32u4 enough computational potential to perform control loops in sampling less than 1 ms. Fast output PWM frequencies of ATmega can be reached by scaling of Arduino's internal timers[6]. The default frequency of approx. 500 Hz is not sufficient for PWM control with sampling at 2 ms and less, but by timer scaling, it can be adjusted up to 62 kHz.

3.5 Remote Power Management

In laboratory practice, but also in common life, we meet situations when electronic equipment fails and the only option is restart. This issue is typical for networking devices

---

[5] http://www.humusoft.com/produkty/models/ce152/

[6] http://playground.arduino.cc/Main/TimerPWMCheatsheet

like routers and switches, but can occur for electronics in general. In remote laboratories, this issue is even more significant. Failure of networking device can cause the outage of connection between students and labs, and in worse case, the overall loss of control/administration over the labs.

ArPi Lab uses a sophisticated method of power management to deal with hardware failures and power related issues. Each functional part of hardware architecture is powered by electric source that can be managed and monitored remotely (Fig. 7). For this purpose we use the sets of programmable power outlet strips (Gembird Silver Shield), which are controlled by power management server (PMS) through USB. PMS is the Apache based HTTP Web server running on Raspberry Pi computer. Unlike other hardware parts of ArPi Lab architecture, the PMS is the only device in laboratory that uses separate power and network line for its operation.

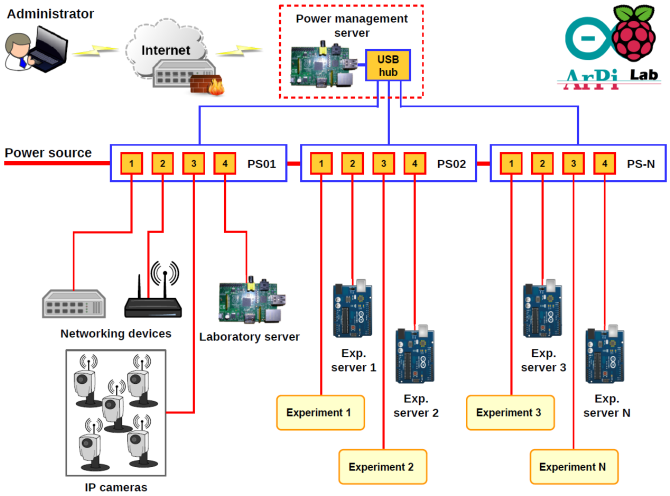


Fig. 7. ArPi Lab power management

If an outage of hardware is reported to administrator, he/she can log-in to PMS's terminal through SSH and can restart or power off the faulty device.

Even if the power management is fully operable through SSH, we plan to extend this system by a secured Web interface for power lines control. Moreover, we consider to implement automatic fault detection to run in PMS for scanning both hardware and communication issues.

4. LABORATORY PRACTICES

ArPi Lab provides experimentation base for several courses taught at Department of Information Engineering and Process Control, STU in Bratislava. Within these courses, students gain knowledge of control design, identification and system dynamics in general. Most of the tasks during the courses period are held in traditional control laboratories, where students can apply their theoretical knowledge on real equipment in their practice sessions.

The following points show an example of students' assignment handled in remote laboratory.

(1) For selected laboratory system, perform the measurement, in order to receive transient characteristics. Do several changes on input signal to get appropriate output behavior.
(2) Chose the structure of mathematical model which provides a good approximation of measured system.
(3) Perform the identification of unknown parameters of model. Use arbitrary identification method based on measured data.
(4) Validate the final form of mathematical model, by comparing it's simulated characteristics with data from real plant.
(5) Chose the appropriate form of controller with integral action (PI, PID) and method for parameter tuning.
(6) Proceed through control task on mathematical model and on real laboratory system. Compare given results.
(7) Write the record of assignment. Discuss the results and justify used methods.

Technically, all control algorithms in remote laboratory run on experiment servers (in Arduino micro-controllers) and they are implemented in digital representation. Students can choose between simple PID (alternatively P, PI or PD) with or without anti-wind-up mechanism, polynomial transfer function controller, and simple relay. Since the algorithms are predefined, students do not have to program control logic, but just tune the parameters.

The role of public availability and accessibility of remote labs is very important especially in educational context. To make our remote laboratories available, accessible, and to provide all necessary features like user and laboratory management, we use the Remote Laboratory Management System (RLMS) WebLab-Deusto developed at University of Deusto (Orduña et al., 2011). Moreover, the WebLab-Deusto provides the possibility of remote labs integration to Learning Management Systems (Orduña et al., 2013) and federation model for sharing of laboratories between different institutions.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have shown an effective and low-cost way to develop and implement remote laboratories. The ArPi Lab was introduced and described to show one of the possible way of future remote laboratory design and development not only for control education. It provides students from different domains with possibility of online experimentation without restrictions that are associated with hand-on laboratories. They can carry out their experiments 24/7 and from any place connected to the Internet. The ArPi Lab experiments are publicly available at `http://weblab.chtf.stuba.sk`.

We have shown that remote management can be applied not only to handle laboratory experiments, but also the operational parts of architecture. For this purpose we have implemented power management model for ArPi Lab, to advance from "remote usage" to "full remote management" model.

In our future work, we consider three ways of ArPi Lab extension. The first one foresees implementation of new educational experiments focused on process control, embedded systems, and robotics. The second intention is to extend power management model by automatic fault detection capability for both, power and network failures, and notification mechanism using e-mail and/or GSM. The third one is to implement feature that allows students to design their own controllers and apply their own control scenarios.

ACKNOWLEDGEMENTS

The authors from Bratislava acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0053/13 and the Slovak Research and Development Agency under the project APVV-0551-11. The first author also acknowledges the financial support from the grant of the Slovak University of Technology in Bratislava.

REFERENCES

Al-Busaidi, A.M. (2012). Development of an educational environment for online control of a biped robot using MATLAB and Arduino. In *13th Int'l Workshop on Mechatronics (MECATRONICS), 9th France-Japan 7th Europe-Asia Congress on and Research and Education in Mechatronics (REM)*, 337–344. Paris, France. doi: 10.1109/MECATRONICS.2012.6451030.

Barber, R. nad Horra, M. and Crespo, J. (2013). Control practices using simulink with Arduino as low cost hardware. In *10th IFAC Symposium Advances in Control Education*, 250–255. University of Sheffield, Sheffield, UK. doi:10.3182/20130828-3-UK-2039.00061.

Campos, B., Angulo, I., Dziabenko, O., Orduna, P., Rodriguez, L., and Garcia-Zubia, J. (2012). Easily deployable low-cost remote lab platform. In *9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 1–4. Bilbao, Spain. doi: 10.1109/REV.2012.6293132.

García-Zubia, J., Angulo, I., Hernández, U., and Orduña, P. (2008). Plug&play remote lab for microcontrollers: WebLab-Deusto-pic. In *7th European Workshop on Microelectronics Education*, 28–30. Budapest, Hungary.

Georgitzikis, V., Akribopoulos, O., and Chatzigiannakis, I. (2012). Controlling physical objects via the Internet using the Arduino platform over 802.15.4 networks. *IEEE Latin America Transactions*, 10(3), 1686–1689. doi:10.1109/TLA.2012.6222571.

Harward, V.J., del Alamo, J.A., Lerman, S.R., Bailey, P.H., Carpenter, J., DeLong, K., Felknor, C., Hardison, J., Harrison, B., Jabbour, I., et al. (2008). The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories. *Proceedings of the IEEE*, 96(6), 931–950.

Huba, M. and Halás, M. (2011). Hydraulic plants for face-to-face training and remote experiments. In *9th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 79–81. Stara Lesna, Slovakia. doi:10.1109/ICETA.2011.6112590.

Huba, M., Kurčík, P., and Kamenský, M. (2006). *Thermo-optical device uDAQ28/LT*. STU Bratislava, Illkovičova 3, Bratislava.

Lowe, D., Murray, S., Weber, L., and de la Villefromoy, M. (2009). Labshare: Towards a national approach to laboratory sharing. In *20th Australasian Association for Engineering Education Conference*, 458–463. Adelaide, Australia.

Neto, J.M., Paladini, S., Pereira, C.E., and Marcelino, R. (2012). Remote educational experiment applied to electrical engineering. In *9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 1–5. Bilbao, Spain. doi: 10.1109/REV.2012.6293164.

Orduña, P., Irurzun, J., Rodriguez-Gil, L., García-Zubia, J., Gazzola, F., and Lopez-de Ipina, D. (2011). Adding new features to new and existing remote experiments through their integration in WebLab-Deusto. *International Journal of Online Engineering (iJOE), Special issue 2 (REV2011)*, 7, 33–39.

Orduña, P., Uribey, S., Isazaz, N., Sancristobalx, E., Emaldi, M., Martinx, A., DeLongz, K., Baileyz, P., López-de Ipina, D., Castrox, M., and Garcia-Zubia, J. (2013). Generic integration of remote laboratories in learning and content management systems through federation protocols. In *IEEE Frontiers in Education Conference, 2013*. Oklahoma City, Oklahoma, US.

Pradyumna, P., Tarun, C., and Bhanot, S. (2012). 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. doi:10.1109/AICERA.2012.6306740.

Prieto, G.A. and Mendoza, J.P. (2013). Low cost didactic robotic platform based on player/stage software architecture and la fonera hardware. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 8(3), 126–132. doi:10.1109/RITA.2013.2273112.

Santana, I., Ferre, M., Izaguirre, E., Aracil, R., and Hernandez, L. (2013). Remote laboratories for education and research purposes in automatic control systems. *IEEE Transactions on Industrial Informatics*, 9(1), 547 –556. doi:10.1109/TII.2011.2182518.

Sobota, J., Pišl, R., Balda, P., and Schlegel, M. (2013). Raspberry Pi and Arduino boards in control education. In *10th IFAC Symposium Advances in Control Education*, 7–12. University of Sheffield, Sheffield, UK. doi: 10.3182/20130828-3-UK-2039.00024.

UNED (2013). *University Network of Interactive Laboratories*, Nov. 8, 2013 [Online]. URL `http://unilabs.dia.uned.es/`.

Ursutiu, D., Cotfas, D.T., Ghercioiu, M., Samoila, C., Cotfas, P.A., and Auer, M. (2010). Web instruments. In *1st Annual Engineering Education Conference (EDUCON)*, 585–590. Madrid, Spain. doi: 10.1109/EDUCON.2010.5492525.

Vargas, H., Sanchez Moreno, J., Jara, C., Candelas, F.A., Torres, F., and Dormido, S. (2011). A network of automatic control web-based laboratories. *IEEE Transactions on Learning Technologies*, 4(3), 197–208. doi: 10.1109/TLT.2010.35.

WebLab-Deusto research group (2013). *WebLab-Deusto*, Nov. 8, 2013 [Online]. URL `https://www.weblab.deusto.es/web/`.

Zimin, A.M., Korshunov, S.V., Shumov, A.V., and Troynov, V.I. (2013). Remote access laboratories for training of engineers in the 21th century. In *8th International Forum on Strategic Technology (IFOST)*, volume 1, PS–9–PS–13. Ulaanbaatar, Mongolia. doi: 10.1109/IFOST.2013.6616942.