Integration of Open Source Arduino with LabVIEW-based SCADA through OPC for Application in Industry 4.0 and Smart Grid Scenarios

Isaías González Pérez[a], A. José Calderón Godoy[b] and Manuel Calderón Godoy[c]
Industrial Engineering School,University of Extremadura, Avenida de Elvas, Badajoz, Spain

Keywords: SCADA, LabVIEW, Arduino, Open Source, OPC, Ethernet, Industry 4.0, IoT, Smart Micro-grid.

Abstract: Modern innovative concepts around Digital Information and Communication Technologies (DICTs), like Industry 4.0, the Internet of Things or Smart Grids, are impacting the scientific and technological worlds and, hence, in control and automation arenas. These trends involve networked interconnection and continuous data flow between a number of hardware and software actors. In parallel, open source technology has gained increasing attention from last years, especially due to the widespread presence of the open source hardware Arduino microcontroller. Focusing on industrial advanced frameworks, Supervisory Control and Data Acquisition (SCADA) systems are required to exchange data with new smart devices, sensors and/or actuators. Arduino boards are commonly used as development platforms for such smart devices. Therefore, communication solutions must be designed towards the convergence of open source hardware and widely used traditional SCADA-devoted software. This paper presents a system that seamlessly integrates Arduino boards into a LabVIEW-based SCADA system through Ethernet connection. The open connectivity provided by the Open Platform Communications (OPC) protocol enables such integration. The proposed framework is a novelty in scientific literature. The development of the system is reported and initial results are provided to demonstrate the feasibility of the proposal.

1 INTRODUCTION

The ever-growing expansion of Digital Information and Communication Technologies (DICTs) has created a set of modern and innovative paradigms like the Internet of Things (IoT), Cyber-Physical Systems (CPSs), Big Data, and Cloud computing. Regarding control and automation-related arenas, two intimately linked concepts have arisen as a consequence of their application to industrial environments, namely Industry 4.0 and Industrial IoT (IIoT).

In industrial facilities, Supervisory Control and Data Acquisition (SCADA) systems carry out the paramount tasks of data gathering and display to the operator, enabling a continuous surveillance/tracking of the process behaviour and status of the involved components. A network of data acquisition and control/automation devices exchanges data with a software application that processes them. This way, the operator is capable of monitoring the automated system behaviour by means of real-time information through numerical and/or graphical visualizations. The introduction of DICTs in these facilities has enhanced the functions afforded by SCADA systems as well as the amount of interconnected devices and data flows.

In parallel, open source technology has gained increasing attention from last years (González et al., 2017), even contributing to the progressive real implementation of such modern trends. For instance, as asserted by Martínez et al. (Martínez et al., 2017), open source hardware and software projects are key accelerators for the industry adoption of IoT. Focusing on open source hardware equipment, a small, cheap and easy-to-configure microcontroller has reached a widespread presence: Arduino (Arduino Online). It acts as core or as auxiliary device in a great number of applications in the fields of data

[a] https://orcid.org/0000-0001-5645-3832
[b] https://orcid.org/0000-0003-2094-209X
[c] https://orcid.org/0000-0001-8380-8547

Pérez, I., Calderón Godoy, A. and Godoy, M.
Integration of Open Source Arduino with LabVIEW-based SCADA through OPC for Application in Industry 4.0 and Smart Grid Scenarios.
DOI: 10.5220/0007795301740180
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 174-180
ISBN: 978-989-758-380-3

acquisition, automation and engineering in general (Calderón et al., 2016). Some recent examples are devoted to educational remote laboratories (Mejías et al., 2017; Chacón et al., 2017), ZigBee-based wireless networks (Pereira et al., 2015), CPSs (García et al., 2016), greenhouse control (Robles et al., 2017), or monitoring of fuel cells (Calderón et al., 2016; Segura et al., 2017).

Nonetheless, despite the versatility of Arduino boards, for industrial locations they present several weaknesses like signals levels non-compliant with industrial ranges, not suitability for rail-mounting, low robustness, etc. (Puigt, 2015).

Within the context of the abovementioned new paradigms, SCADA systems are required to exchange data with new smart devices, sensors and/or actuators. Precisely, Arduino boards are commonly used as development platforms for such smart devices. Therefore, communication solutions must be designed towards the convergence of open source hardware and widely used traditional SCADA-devoted software. A widely used SCADA software environments is Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) of National Instruments (LabVIEW Online). This package is considered as representative due to the fact that it has a worldwide presence and support for thousands of technologies and instruments (Arpaia et al., 2015).

In this case, the integration of Arduino with LabVIEW can be approached by means of the LabVIEW Interface for Arduino (LIFA) toolkit. Examples of the first approach have been reported in (Calderón et al., 2016), (Segura et al., 2017) and (Vivas et al., 2019), where an Arduino microcontroller is used as data acquisition board for measuring hydrogen fuel cells voltage and temperature, whereas a LabVIEW-based interface feeds information to the user.

Nevertheless, this toolkit does not provide Ethernet connection. An option consists on developing a TCP or UDP linkage between the LabVIEW and the Arduino, but this solution involves deep expertise about such kind of communication.

This paper presents a system that seamlessly integrates Arduino boards into a LabVIEW-based SCADA system through Ethernet connection and a standardized protocol widely used for industrial facilities, the Open Platform Communications (OPC) interface.

In order to solve Ethernet communication, a suitable option consists on using the open source software package Arduino OPC Server, developed by I. Martínez Marchena (Arduino OPC Server Online). It enables the communication between Arduino boards and software applications that support OPC connectivity, like LabVIEW.

It must be noted that OPC interface was created in 1996 to handle interoperability in industrial control and automation applications (OPC Foundation Online). Currently, it is considered as one of the main contenders to lead the standardization and systems integration in advanced frameworks (González et al., 2017b; González et al., 2019). According to the literature survey conducted by the authors, there is no paper reporting an OPC communication between LabVIEW and Arduino.

This proposal is framed in a research project to develop and implement a Smart Micro-Grid based on renewable energy sources and its digital replica. To this aim, an automation and supervision system is crucial to manage and operate such challenging facility. Arduino chips need to be connected to a LabVIEW-based SCADA for a proper measurement and monitoring of several of the involved variables.

The goal of this work is to promote the joint utilization of open source tools with industrial supervisory software. Therefore, with the aim of demonstrating the feasibility and validity of the proposed approach, an experimental system has been implemented using an Arduino Mega and a LabVIEW-based SCADA.

In the authors' humble opinion, the integration of the de facto standard software for supervision and instrumentation, LabVIEW, and the versatility of the open source Arduino microcontroller provides a powerful benchmark for R&D activities in the fields of Industry 4.0, IIoT and Smart Grids.

The remainder of the paper is as follows. Section 2 deals with the description of the developed system, both hardware and software subsystems. In Section 3, the development of the system and initial outcomes are reported. Finally, main conclusions and further works are outlined in Section 4.

2 MATERIALS AND METHODS

The developed system comprises hardware and software components therefore; in this section the involved entities are separately described.

2.1 Hardware Subsystem

Among the available Arduino boards (Uno, Yun, Mini, Nano, Mega, Duemilanove, Lilypad, etc.) the model Mega 2560 R3 has been selected. It incorporates an ATmega2560 microcontroller, 16 analog inputs, 54 digital input/output ports, 4 UARTs, a 16 MHz crystal oscillator, a power jack, and a reset button. A USB port enables powering the board and establishing communication with the configuration computer.

The so called shields are expansion cards that provide additional features to the main board and can be directly coupled to it. In this application, an Ethernet shield delivers Ethernet communication by means of a RJ45 port. Concretely, the Hanrun HR911105A Ethernet shield has been chosen. It uses the Serial Peripheral Interface (SPI) protocol to exchange data with the Arduino board.

Also, some ancillary devices are required, such as an Ethernet switch, the corresponding Ethernet wires, and power supplies.

Finally, a PC executes the software applications that are commented in the next subsection. To summarize, this PC is used for configuration tasks of both the hardware and software entities, and also as supervisory station since the SCADA system runs in it.

2.2 Software Subsystem

The Integrated Development Environment (IDE), a free software package, is used to configure the Arduino chip by means of a programming language based on a simplified version of the C++ language. The IDE runs in a PC to which the microcontroller is connected via serial communication.

In order to enable OPC communication, the Arduino OPC Server version 2.0 is used. This freeware and open source package provides OPC Data Access (DA) –compliant communication with various Arduino boards, namely Uno, Yun and Mega. Both serial and Ethernet connections are supported to link the Arduino board and the PC where the server runs.

LabVIEW is proprietary software that supports a large number of technologies and protocols, and includes powerful in-built functions. The programs built with LabVIEW are called Virtual Instruments (VIs) and are programmed via a high-level graphical language. Hence, this environment has been chosen to implement the SCADA system that retrieves data from the Arduino. Besides, an additional module is required to establish OPC linkage, the Datalogging and Supervisory Control (DSC) module.

Figure 1 depicts the schematic block diagram of the system. To the open source board, different compatible sensors can be connected for sensing and data acquisition purposes. The OPC server for Arduino and the LabVIEW-based SCADA (OPC client) run in the same PC. Such a PC and the Arduino board are integrated through Ethernet into a Local Area Network (LAN), establishing a continuous data flow between them. In addition, Industrial Control Devices (ICDs), like Programmable Logic Controllers (PLCs), can also be communicated to the supervisory system through the network.

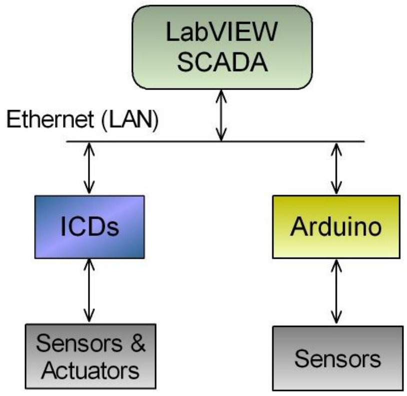

Figure 1: Block diagram of the developed system.

3 DEVELOPED APPROACH AND INITIAL OUTCOMES

This section is devoted to describe the development of the system paying special attention to the configuration of the OPC link between the Arduino and the SCADA. Moreover, achieved initial outcomes in an experimental facility are reported in order to validate the proposal.

3.1 Communication Configuration

The communication established between the microcontroller and the SCADA system must be configured at hardware and software levels. Regarding the first one, as previously commented, an Ethernet network is used to interconnect the Arduino (by means of its shield) and the PC where the supervisory application runs.

At the software level, in the Arduino chip is necessary to incorporate three libraries in the sketch. Namely, the corresponding to the SPI bus (SPI.h), to the Ethernet link (Ethernet.h) and the one devoted to share data through OPC (OPC.h).

By means of the Ethernet library, the IP address, gateway and the subnet are specified so the VI and the Arduino can establish an Ethernet connection. The port for the connection is 80, the default one used for web linkages. Figure 2 illustrates the configured parameters for such a communication.

```
 * MAC address from Ethernet shield sticker under board
 */
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0E, 0xAD, 0x8D };

/*
 * Set your network configuration
 */
IPAddress ip(192, 168, 1, 3);
IPAddress gateway(192,168,1,1);
IPAddress dns_server(10,1,0,1);
IPAddress subnet(255,255,255,0);

/*
 * Server listen port
 */
const int listen_port = 80;
```

Figure 2: Ethernet library configuration.

Concerning the OPC library, its configuration consists on declaring the variables to be shared via OPC so the OPC server makes them available for the client program.

As indicated in Section 2, the Arduino OPC Server allows both serial and Ethernet connections. The first type was used during the starting stage of the development to test the configuration of the libraries and the OPC link. However, the Ethernet connection is the most powerful since it enables the Arduino and the PC to be physically distant but communicated through the network.

The configuration of the Ethernet connection of Arduino in the OPC server is very intuitive; simply the IP address and the port must be introduced as can be seen in Figure 3. It is evident that this IP address must be coincident with the one configured in the Ethernet library.

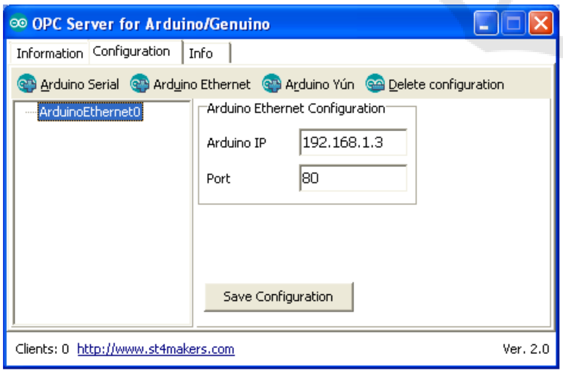

Figure 3: Ethernet communication parameters in the OPC server.

3.2 SCADA Configuration

Once the Arduino has been configured for both Ethernet and OPC communications, the SCADA has to be parameterized to exchange data with it. The VI is created within a LabVIEW project in order to use the capabilities provided by the DSC module. Thanks to the OPC channel, this stage can be performed through three main steps:

1. Addition of the OPC-shared variables to the LabVIEW project. The Arduino OPC server must be added like a new I/O Server, by selecting it among the available servers registered in the operative system (Figure 4). Once the server is selected, the variables shared via OPC can be chosen, in the present case, the item has been named in the Arduino sketch as AI8, as shown in Figure 5.

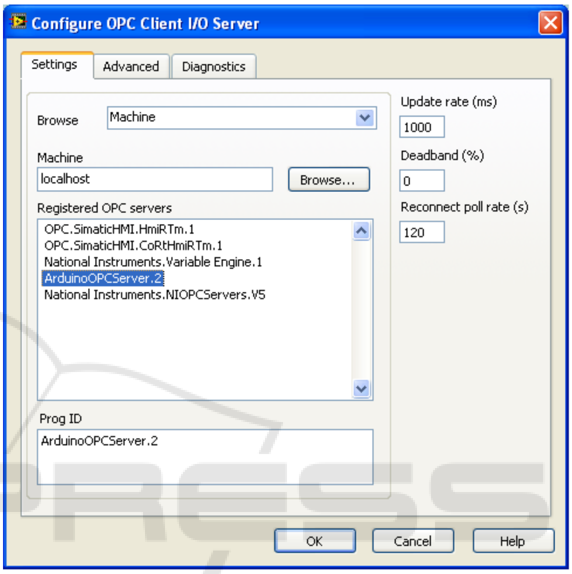

Figure 4: Selection of the Arduino OPC server among the available servers.

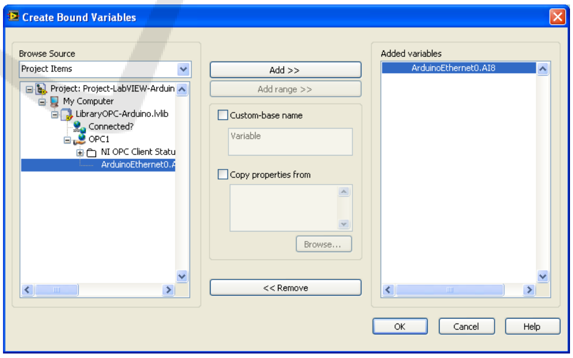

Figure 5: Addition of the OPC-shared variables to the LabVIEW project.

During this phase, the Info tab of the OPC server shows details about the connection of the OPC client, the VI, as can be seen in Figure 6.

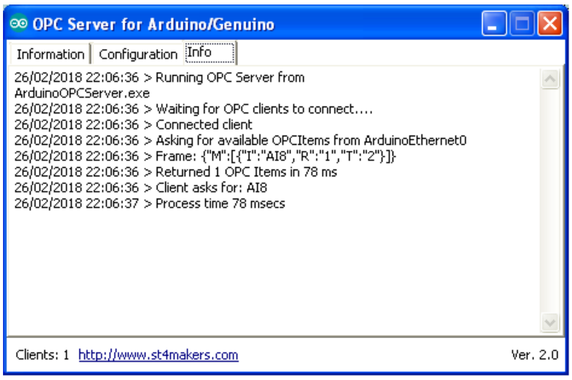

Figure 6: OPC client connection information provided by the Arduino OPC Server.

2. Selection of the shared items via OPC. The items are available via the project library so can be directly chosen within the VI, as can be appreciated in Figure 7. The DSC module provides the shared variable element in order to choose a variable among those that have been defined in the project. In this case, the analogue input of the Arduino is selected.

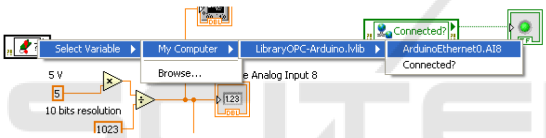

Figure 7: Selection of OPC-shared items in the VI.

3. Design of the interface. This last step consists on carrying out the design and organization of the interface that will provide continuous information to the user. In this sense, some considerations regarding easy-to-use and intuitive distribution of the elements have been taken into account. The design includes the incorporation of every type of element required to visualize the data retrieved from the Arduino like trend graphics, analogue and Boolean indicators. In the present case, as a proof of concept, a graphical chart and numerical indicators have been considered In order to improve the information afforded to the user, a Boolean signal is used to inform about the successful connection of the OPC server. To this aim, the own OPC server provides a bit named Connected?, so its true value indicates that the connection has been established.

3.3 Initial Outcomes

Aiming to demonstrate the validity of the proposal, the temperature of a photovoltaic module has been measured and monitored. This approach constitutes a preliminary stage for the application to a Smart Micro-Grid, as it was commented in the Introduction.

A photograph of the experimental setup is provided in Figure 8. The Ethernet shield and the Arduino board can be observed placed on the left of the module.

The achieved initial outcomes are shown in Figure 9, where the SCADA can be observed working under real conditions. In the presented case, only reading operations have been implemented, i.e., the Arduino acts as a data acquisition system and sends the information to the SCADA.

To this aim, one temperature sensor Lm35 has been connected to one of the analogue input ports of the Arduino board. A numerical field shows the value of the measured temperature. Moreover, a graphical chart illustrates the evolution of such signal over time.

Figure 8: Experimental setup of the Arduino and the photovoltaic module.

In order to reflect the effective data exchange between the nodes (Arduino and LabVIEW SCADA system), Figure 10 shows sthe evolution of the solar irradiance and the temperature of the module during a day of operation. The solar irradiance in the inclined plane is represented in red colour and divided by 200 (*Ginc/200*) in order to be represented in the same graphic. To depict the temperature of the panel, the measurement carried out by the Lm35 sensor and the Arduino board has been chosen. This magnitude is named as *Tp* and corresponds to the blue- coloured curve.

These outcomes prove that the Arduino measures the variations of the module temperature and successfully sends in real time the information to the SCADA system.

It should be remarked that the presented system is expandable and adaptable to accommodate new developments in automation, control, measurement and communications. On the open source view, future enhancements of Arduino libraries and devices are expected to empower the system.

Moreover, Arduino boards can also execute control orders from the SCADA system or even implement control algorithms.

Nevertheless, more evaluations of the proposal as well as long-term operation must be studied.

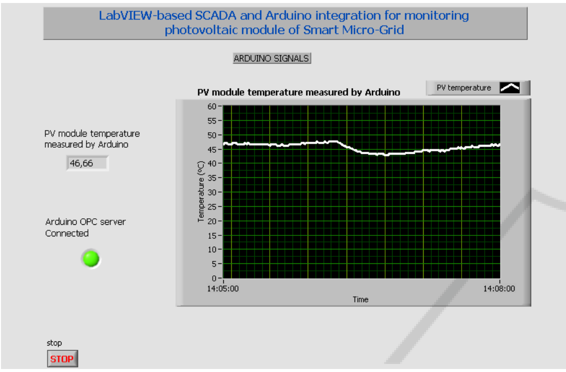

Figure 9: Developed SCADA working.

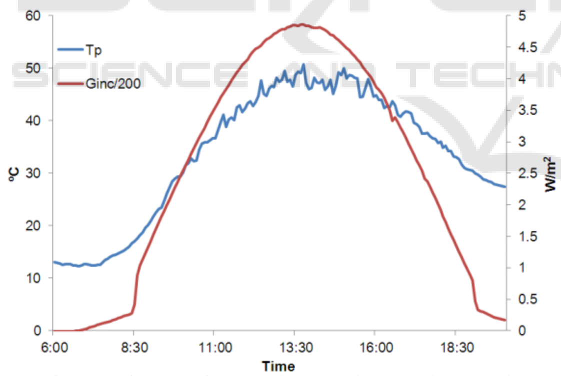

Figure 10: Irradiance and temperature of the photovoltaic module measured by Arduino during a sunny day.

4 CONCLUSIONS AND FURTHER WORKS

This article has presented a successful communication based on Ethernet and OPC to integrate Arduino microcontrollers with LabVIEW-based supervisory systems.

The open source feature of Arduino offers important benefits like low cost, vast amount of information available in the Internet and easy configuration, just to name a few. On the other hand, LabVIEW is a well-known software environment to implement monitoring and supervisory systems widely used by both Academia and industry. Indeed, the presented integrative framework is a novelty in scientific literature.

In fact, two open source tools have been successfully used, at hardware level the Arduino board, and at software level, the OPC server for Arduino. Achieved outcomes about sensing a photovoltaic module temperature have shown a proper operation of the system, demonstrating the feasibility of the proposal.

The presented approach is envisioned to facilitate the integration of open source tools within industrial infrastructures under the frameworks of innovative trends like the Industry 4.0, the IoT and Smart Grids.

Further works are focused on applying the proposal to measure different magnitudes of a Smart Micro-Grid. This way, Arduino will belong to the Advanced Metering Infrastructure (AMI) operating in an integrated manner with the SCADA system.

ACKNOWLEDGEMENTS

This research has been funded by the project IB18041 supported by the Junta de Extremadura in the VI Plan Regional de I+D+i (2017-2020), co-financed by the European Regional Development Funds FEDER (Programa Operativo FEDER de Extremadura 2014–2020).

In addition, authors are grateful to the community that supports Arduino-based developments under open source philosophy. Special thanks are given to I. Martínez Marchena, developer of the open source Arduino OPC Server.

REFERENCES

Arduino Online. Available: www.arduino.cc (accessed on 30 January 2018)

Arduino OPC Server Online. Available: https://www.st4makers.com/opc-server-for-arduino (accessed on 30 January 2018)

Arpaia, P., De Matteis, E., Inglese, V., 2015. Software for measurement automation: A review of the state of the art. *Measurement*, vol. 66, pp. 10–25.

Calderón, A. J., González, I., Calderón, M., Segura, F., Andújar, J. M., 2016. A New, Scalable and Low Cost Multi-Channel Monitoring System for Polymer Electrolyte Fuel Cells. *Sensors*, vol. 16(3), pp. 349.

Chacón, J., Saenz, J., de la Torre, L., Diaz, J. M., Esquembre, F., 2017. Design of a Low-Cost Air Levitation System for Teaching Control Engineering. *Sensors*, vol. 17, pp. 2321.

García, M. V., Irisarri, E., Pérez, F., Estévez, E., Marcos, M., 2016. OPC-UA Communications Integration using a CPPS architecture. *IEEE Ecuador Technical Chapters Meeting*, Guayaquil, Ecuador.

González, I., Calderón, A. J., Barragán, A. J., Andújar, J. M., 2017. Integration of Sensors, Controllers and Instruments Using a Novel OPC Architecture. *Sensors*, vol. 17(7), pp. 1512.

González, I., Calderón, A. J., Andújar, J. M., 2017. Novel Remote Monitoring Platform for RES-Hydrogen based Smart Microgrid. *Energy Conversion and Management*, vol. 148, pp. 489-505.

González, I., Calderón, A. J., Figueiredo, J., Sousa, J. M. C., 2019. A Literature Survey on Open Platform Communications (OPC) Applied to Advanced Industrial Environments. *Electronics*, vol. 8, pp. 510.

LabVIEW Online. Available: http://www.ni.com/en-gb/shop/labview.html (accessed on 30 January 2018)

Martinez, B., Vilajosana, X., Kim, I. H., Zhou, J., Tuset-Peiró, P., Xhafa, A., Poissonnier, D., Lu, X., 2017. I3Mote: An Open Development Platform for the Intelligent Industrial Internet. *Sensors*, vol. 17, pp. 986.

Mejías, A., Reyes, M., Márquez, M. A., Calderón, A. J., González, I., Andújar, J. M. 2017. Easy handling of sensors and actuators over TCP/IP Networks by Open Source Hardware/Software. *Sensors*, vol. 17, 94, 2017.

OPC Foundation Online. Available: https://opcfoundation.org/ (accessed on 20 January 2018)

Pereira, R., Figueiredo, J., Melicio, R., Mendes, V. M. F., Martins, J., Quadrado, J. C., 2015. Consumer energy management system with integration of smart meters," *Energy Reports*, vol. 1, pp. 22-29.

Puigt, M., 2015. How Arduino is Open-sourcing Industry. Presentation at Arduino Day 2015, Fablab Côte d'Opale, Calais, France. Available online: http://www-lisic.univ-littoral.fr/~puigt/LectureNotes/Arduino/ArduinoDay2015_vfinal.pdf.

Robles, C., Callejas, J., Polo, A., 2017. Low-Cost Fuzzy Logic Control for Greenhouse Environments with Web Monitoring. *Electronics*, vol. 6, pp. 71.

Segura, F., Bartolucci, V., Andújar, J. M., 2017. Hardware/Software Data Acquisition System for Real Time Cell Temperature Monitoring in Air-Cooled Polymer Electrolyte Fuel Cells. *Sensors*, vol. 17(7), pp. 1600.

Vivas, F. J., Heras, A., Segura, F., Andújar, J. M., 2019. Cell voltage monitoring All-in-One. A new low cost solution to perform degradation analysis on air-cooled polymer electrolyte fuel cells. *International Journal of Hydrogen Energy*, in press.