INTEGRATION FRAMEWORK FOR SENSOR THROUGH IOT EMBEDDED MICROCONTROLLER IN AUTOMATION SYSTEM FOR HYDROGEN GENERATOR MANAGEMENT

Francisco Javier Folgado Gaspar, Diego Orellana Martín, Isaías González Pérez, Antonio José Calderón Godoy
Escuela de Ingenierías Industriales, Universidad de Extremadura, {ffolgar, dorellana, igonzp, ajcalde}@unex.es

Abstract

This paper presents a framework to integrate a traditional sensor within an Industry 4.0 featured automation system through an Internet of Things (IoT) embedded microcontroller. In particular, an Arduino microcontroller is connected to a water level sensor required for the performance of a hydrogen generator. By means of the IoT communication protocol Modbus TCP, the signal of the sensor is sent to an industrial programmable logic controller that automates the operation of a Smart Microgrid where the hydrogen generator is fed by a photovoltaic array. The developed system is described concerning hardware, software and communications, together with experimental results.

Keywords: IoT, Industry 4.0, Arduino, microcontroller, automation, Modbus TCP, electrolyzer, hydrogen.

1 INTRODUCTION

The Internet of Things (IoT) is a key enabling technology in the Industry 4.0 [1]. The IoT consists of a variety of devices with embedded systems connected to the telecommunications network, e. g. the Internet, which have the ability to generate and automatically send information without direct human intervention [12]. Therefore, massive data acquisition, storage, transmission and processing must be accomplished under these concepts.

For the real-world deployment of IoT technology, low-cost embedded platforms are being increasing applied. For instance, Arduino, Genuino, Adafruit, Raspberry Pi, BeagleBone, and Odroid are available in the market with diverse information in the Internet for their configuration.

Regarding Arduino, it is an open-source single-board microcontroller widely used in the context of IoT and Do It Yourself (DIY) projects related to electronics prototyping, data acquisition, control, etc. It was created in 2005 and its schematics and software are available under different types of licenses, which has led to the existence of myriads of programs, codes, sketches and projects freely available on the Internet. Within the well-known three-layered IoT architecture comprising Perception, Network and Application layers, the Arduino microcontroller is commonly placed in the Perception level [4,16]. Arduino boards are sometimes referred to as IoT hardware platforms [2].

On the other hand, industrial automation systems are mainly based on Programmable Logic Controllers (PLC), which are electronic programmable microprocessors hardened for harsh conditions and continuous operation.

In industrial environment, Arduino has been scarcely used due to some limitations like lack of standard signals levels (voltage, current) as well as weak hardware connections [7]. However, some cases of application and integration of Arduino in industry-focused scenarios have been reported in previous literature. For example, in [5] an Arduino with a K-type thermocouple is used as industrial IoT sensor for high-temperature measurement in an industrial oven for twenty-two hours. In [18], Arduino is included within a fog-computing framework for process monitoring in the so-called cyber manufacturing. Data from an accelerometer is collected through Arduino boards in an Industry 4.0-enabled spring factory [13]. A solution based on the protocol Open Platform Communications for data exchange between industrial supervisory systems and Arduino boards is reported in [7]. In [15] the architecture of automation systems using low-cost embedded platforms is presented, including Arduino for control purposes. Applications of Arduino in the mining industry are reviewed in [11]. Continuous monitoring and logging of the temperatures of a photovoltaic generator is solved by Arduino and digital sensors in [17]. The integration of sensors by means of single-board architectures, like Arduino, in different scenarios is reviewed in [3], which shows the wide applicability of this type of systems.

Despite the aforementioned literature, there is still the need for integration frameworks in order to promote the utilization of Arduino in Industry 4.0 and IoT scenarios taking advantage of its numerous benefits. In fact, the integration of Arduino microcontrollers and PLC brings diverse benefits and empowers the development of innovative scenarios [7].

To this aim, this paper presents a framework to integrate a traditional water level sensor into a PLC-based automation system through the IoT microcontroller Arduino and the protocol Modbus Transport Control Protocol (TCP). The sensor is needed to inform about the amount of water accumulated in a phase separator stage of a hydrogen generator. When the level reaches the top value, it is imperative to purge the water. In fact, the water purge is critical for the proper operation of the electrolyzer as well as for security reasons. Previous literature about data acquisition and sensing of hydrogen generators scarcely indicates the applied equipment, which is commonly solved by a PLC [6]. Therefore, using IoT technology constitutes a novelty in this regard.

The motivation of this work arose when developing an automation and management system for a hydrogen generator, e.g. electrolyzer within a renewable energy microgrid. The sensor could be directly coupled to the PLC responsible of managing the microgrid. However, through the Arduino board, the sensor becomes a smart node able to be connected into an Ethernet network. Moreover, the level sensor requires certain circuitry for signal adaptation to be connected to a digital input of the PLC due to the fact that they manage 5V and 24V, respectively. Even more, the information gathered by the Arduino is available not only for the PLC but also for other nodes like Supervisory Control and Data Acquisition (SCADA) systems.

The structure of the rest of the paper is as follows. The second section describes the materials and methods used in the research. Section 3 expounds the developed framework, at both hardware and software levels, together with illustrative results. Finally, the main conclusions are commented.

2 MATERIALS AND METHODS

In this section, the main hardware and software components that have been used in the research are briefly described.

To begin with, the water level sensor an electro-optic sensor model ELS-900 of the manufacturer Gems Sensors [19]. This sensor is devoted to detect the level that water reaches within a gas-liquid phase separator. This separator hosts the generated hydrogen, which contains small amounts of water. The sensor provides a signal of 5V while the water does not reach the height where it is placed, so, when the water rises up to such height the signal is 0V. The main technical features of this device are summarized in Table I and its physical aspect can be seen in Figure 1:

Table I: Main features of water level sensor.

Feature	Values
Input voltage	5 V
Current consumption	18 mA
Operating temperature	-40°C to +125°C
Operating pressure	0 to 17 bar
Repeatability	± 1 mm
Housing and prism material	Polyethersulfone

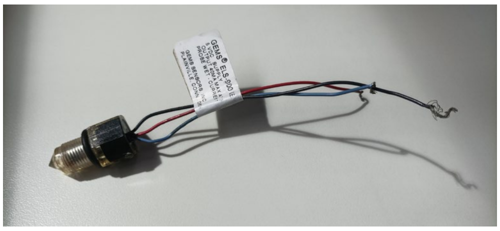


Figure 1: Physical aspect of the water level sensor.

The model or Arduino microcontroller that has been selected for the reported application is the Arduino MEGA 2560. It is based on the microcontroller ATmega2560 and incorporates 54 digital input and output ports, together with 16 analogue input ports, a 16 MHz crystal oscillator, a power jack, and a reset button. In order to provide Ethernet connectivity, an Ethernet shield model W5100 is mounted, exchanging data with the microcontroller through the Serial Peripheral Interface (SPI) bus.

The PLC of high-performance series S7-1500 of Siemens is the industrial controller used in the presented system. The Central Processing Unit (CPU) corresponds to the model S7-1516 which provides 5 MB of memory for data, 1 MB of memory for programs, a port for fieldbus communication (PROFIBUS) and three ports for Ethernet connectivity (RJ45). For the connection of digital and analog sensors/actuators, digital and analog modules can be coupled to the CPU.

 https://doi.org/10.17979/spudc.9788497498418.1063

Regarding communications, the protocol Modbus TCP is the most modern version of the industrial de facto standard Modbus. Created in 1979 by Modicon, it rapidly spread in industrial floor for open communication between controllers and SCADA systems. Namely, Modbus TCP is the consequence of evolution and adaptation to new decentralized frameworks [9], follows client/server architecture, and has been signaled as an IoT communication protocol in recent literature [8,10].

With respect to the software packages involved in the proposal, to begin with, TIA Portal is the engineering software for programming and configuring Siemens controllers. On the other hand, the Arduino Integrated Developing Environment (IDE) is a cross-platform application written in Java, which can be used to code and configure any Arduino board. The IDE is freely available and runs in a computer to which the board must be connected via Universal Serial Bus (USB) communication.

3 IMPLEMENTATION OF MODBUS-BASED INTEGRATION FRAMEWORK

The developed framework is described in detail in this section. Firstly, the block diagram depicted in Figure 2 illustrates the interconnection between the involved nodes.

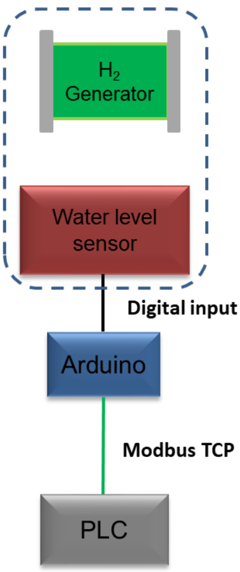


Figure 2: Block diagram of the developed framework.

The sensor is connected to the Arduino board via a digital input port, which handles the proper voltage, 5V. This way, the signal provided by the sensor is read by the Arduino and sent to the PLC by means of the Ethernet network using the Modbus TCP protocol. An Ethernet switch carries out the physical implementation of the Ethernet network. To this device, both the Arduino chip and the PLC are connected via common Ethernet wires.

As commented in the previous section, Modbus TCP uses the client/server architecture. In particular, in the developed system, the PLC acts as Modbus server whereas the Arduino develops the client role. In the client side, the library ArduinoModbus.h is used for data exchange through Modbus TCP. Moreover, the library Ethernet.h is required to establish Ethernet communication through the Internet Protocol (IP) address. The flowchart of steps performed in the Arduino sketch is shown in Figure 3. The initial stage consists on reading the signal provided by the sensor in the digital input port to which it is connected. After that, a Modbus TCP connection is established between the client (Arduino) and the server (PLC). Next, the signal gathered from the sensor is sent to the server, which stores it within its memory. Finally, the Modbus TCP connection is closed. These steps are cyclically executed for a continuous data exchange between the IoT microcontroller and the PLC.

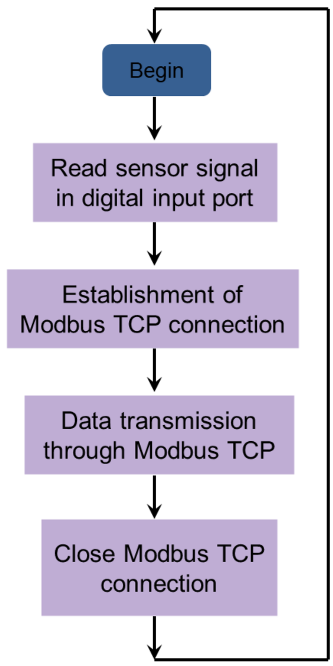


Figure 3: Flowchart of steps performed by IoT microcontroller for data exchange with PLC.

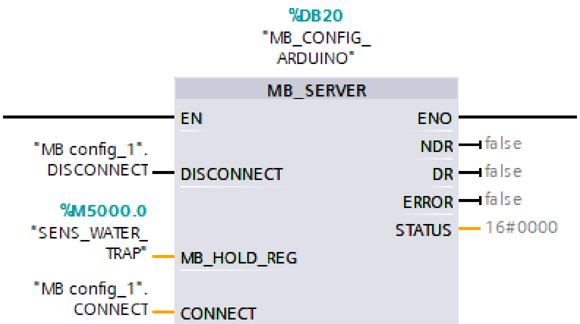
The IoT approach of the Arduino operation is based on interconnection between water level sensor and

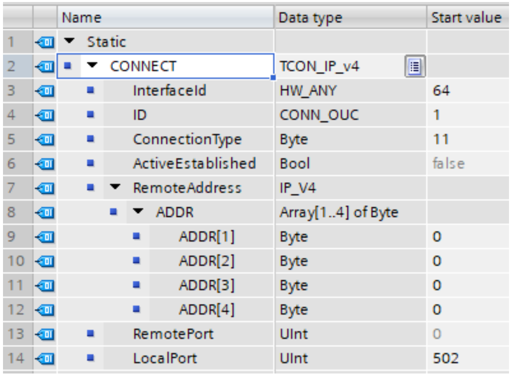
PLC through Modbus TCP protocol. The Arduino board act as a middleware, facilitating the reading of the digital signal and avoiding the need to implement additional electronics to adapt the voltage levels to those of the PLC.

The sensor reading and data transfer are carried out in real time, without the need to rely on databases or cloud applications that store the information to be consulted later by the PLC. In this way, the actions of the Arduino board focus solely on detecting and transmitting the status of the sensor, whose information is vital for the maintenance of the electrolyzer and other devices involved in its operation. Once the information has been received by the PLC, it is dedicated exclusively to executing the electrolyzer management algorithm. A similar IoT application is described in [14], where an Arduino is used as a device to read data from voltage and current sensors installed in different loads in a Microgrid, with the aim of sending this information to a real-time management system via WiFi.

In the Arduino code, the definition of the IP address of both Modbus server and client, as well as the TCP port, namely, the Modbus default port (502) are carried out. Besides, a loop is created for reading the sensor signal and processing the Modbus communication.

Concerning the server side, Figure 4 illustrates the aspect of the PLC code for data exchange via Modbus TCP. Namely, the function MB_Server is configured for the PLC to act as Modbus server. To this purpose, the zone of the memory where the information to share is placed, as well as communication parameters must be defined. The IP address of the Modbus client and the logical port are configured, respectively, as 0.0.0.0 and 502 to this purpose.





	Name	Data type	Start value
1	▼ Static		
2	▼ CONNECT	TCON_IP_v4	
3	InterfaceId	HW_ANY	64
4	ID	CONN_OUC	1
5	ConnectionType	Byte	11
6	ActiveEstablished	Bool	false
7	▼ RemoteAddress	IP_V4	
8	▼ ADDR	Array[1..4] of Byte	
9	ADDR[1]	Byte	0
10	ADDR[2]	Byte	0
11	ADDR[3]	Byte	0
12	ADDR[4]	Byte	0
13	RemotePort	UInt	0
14	LocalPort	UInt	502

Figure 4: Modbus TCP server configuration for PLC.

The sensor signal is received from the IoT microcontroller through Modbus and stored in PLC local memory. An electrovalve is used to purge the accumulated water when the level sensor indicates that the maximum water amount has been reached. This way, the condensed water is returned to the water tank that feeds the hydrogen generator stack. The ladder code programmed in the PLC to handle these signals is shown in Figure 5. Once the sensor output drops to 0V, logical level low, a digital output of the PLC is set to high, thus activating the electrovalve.

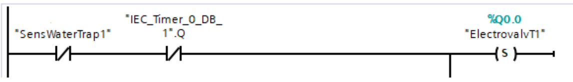


Figure 5: Code of PLC to handle sensor and electrovalve signals.

4 EXPERIMENTAL RESULTS

The physical aspect of the water level sensor mounted in the phase separator is observed in Figure 6. The electrovalve that causes the water purge can be seen at the bottom right of the photograph.

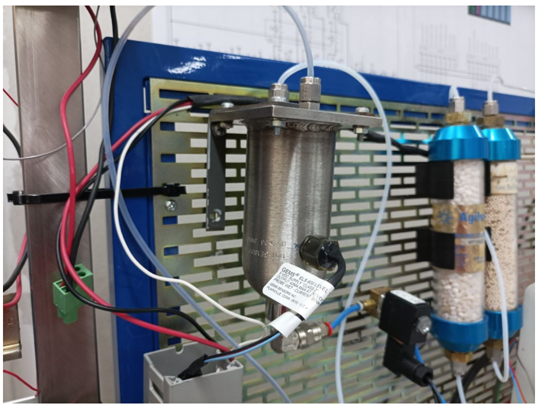


Figure 6: Snapshot of the water level sensor and electrovalve in the phase separator.

https://doi.org/10.17979/spudc.9788497498418.1063

To prove the correct connection and data exchange between the Arduino and the PLC, Figure 7 shows a snapshot centred on the pressure graph of the hydrogen circuit in the monitoring system used for the electrolyser. The figure shows a 5-minute operating period, where it can be seen how the pressure decreases as a result of the activation of the solenoid valve. After that, the pressure increases progressively until it reaches its nominal value within the operating range of the electrolyzer.

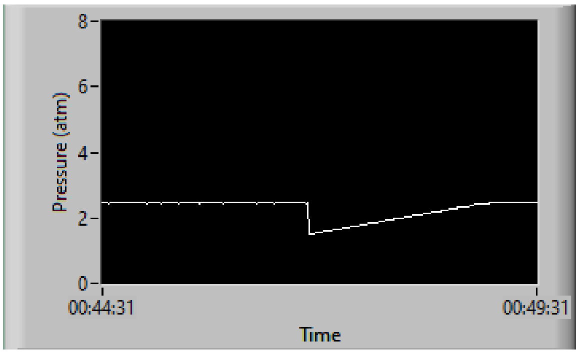


Figure 7: Screenshot of the monitoring system to display the effect of the water level sensor and electrovalve operation.

Thanks to the Modbus server configured in the PLC (Figure 4), the historical data of the electrolyzer operation is stored in an in-house data server. Figure 8 shows the evolution of the system pressure for a period of 7 hours.

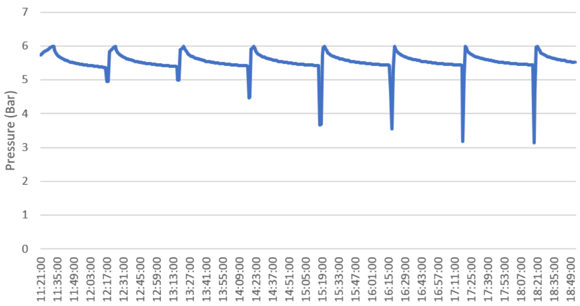


Figure 8: Hydrogen circuit pressure historical data representation

In Figures 7 and 8 it is shown that the pressure drop in the hydrogen circuit is controlled and a total depressurisation of the system does not occur. For this purpose, the PLC uses the signal from the water level sensor together with a timer to keep the working pressure of the electrolyzer at a constant value, preventing the device from needing to re-establish the circuit pressure to continue its operation.

Conventionally, the purging process must be calculated according to the hydrogen flow rate generated, the circuit pressure and the current consumed, involving a greater number of sensors and a high computational cost. The system described in this work implements in a simple manner a control of this process, significantly reducing the number of cycles as well as the purge time necessary for the correct operation of the electrolyzer. This control leads to an increase in the efficiency of the device, by reducing the time and power required to recover the hydrogen circuit pressure to its nominal operating values.

5 DICUSSION AND CONCLUSIONS

This paper has reported the successful connection of a water level sensor to a Modbus-based network for data exchange with a PLC. The sensor is coupled to an IoT microcontroller, in particular, an Arduino board, which manages the communication through the widely applied protocol Modbus TCP.

The validation has been performed within a smart microgrid where hydrogen is produced from solar energy. Namely, the water level of a phase separator of a hydrogen generator is sensed by means of the aforesaid sensor and its signals are processed by a PLC. The fact of applying the approach in hydrogen-related equipment constitutes a challenge given the critical operation of this equipment and the associated safety measures.

The configuration of connection and communication parameters has been easily solved thanks to the ease of use of Arduino. In addition, its low cost facilitates its inclusion in already existent automation systems.
In the reported application, a single Arduino board has been connected to the Modbus TCP network. However, a larger number of IoT-enabled microcontrollers and devices can be integrated due to the widespread support of this protocol.

The PLC-based automation system has been designed under the Industry 4.0 paradigm, so the integration of the IoT microcontroller into the Modbus TCP network constitutes a step forward in the achievement of Industry 4.0 features like digitization, interoperability, and decentralization.

Future works will deal with the deployment of a sensors and actuators network using IoT components.

Acknowledgments

This work has been supported by the FEDER Funds (Programa Operativo FEDER de Extremadura 2014-

2020) through the grant “Ayuda a Grupos de Investigación” (ref. GR21168) of Junta de Extremadura.

Resumen en español

MARCO PARA LA INTEGRACIÓN DE SENSOR MEDIANTE MICROCONTROLADOR EMBEBIDO IOT EN SISTEMA DE AUTOMATIZACIÓN PARA GESTIÓN DE UN GENERADOR DE HIDRÓGENO

Resumen

Este artículo presenta un marco para integrar un sensor tradicional en un sistema de automatización contextualizado en la Industria 4.0 mediante un microcontrolador IoT embebido. En particular, un microcontrolador Arduino es conectado a un sensor de nivel de agua requerido para el funcionamiento de un generador de hidrógeno. Por medio del protocolo de comunicación IoT Modbus TCP, la señal del sensor es enviada a un controlador programable industrial que automatiza la operación de una Smart Microgrid donde el generador de hidrógeno es alimentado por un array de paneles fotovoltaicos. El sistema desarrollado es descrito en cuanto a hardware, software y comunicaciones, junto con resultados experimentales.

Palabras clave: IoT, Industria 4.0, Arduino, microcontrolador, automatización, Modbus TCP, electrolizador, hidrógeno.

References

[1] Alexander, S., Leonid, R., Leonid A., I., Da Xu, L., (2022) Emerging Enabling Technologies for Industry 4.0 and Beyond. Information Systems Frontiers.(September 2021). DOI: 10.1007/s10796-021-10213-w.

[2] Ali, O., Ishak, M.K., Bhatti, M.K.L., Khan, I., Kim, K. Il, (2022) A Comprehensive Review of Internet of Things: Technology Stack, Middlewares, and Fog/Edge Computing Interface. Sensors.22(3):1–43. DOI: 10.3390/s22030995.

[3] Álvarez, J.L., Mozo, J.D., Durán, E., (2021) Analysis of Single Board Architectures Integrating Sensors Technologies†. Sensors.21(18):1–28. DOI: 10.3390/s21186303.

[4] Bellini, P., Nesi, P., Pantaleo, G., (2022) IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies. Applied Sciences (Switzerland).12(3). DOI: 10.3390/app12031607.

[5] Chang, V., Martin, C., (2021) An Industrial IoT Sensor System for High-Temperature Measurement. Computers and Electrical Engineering.95(October 2020):107439. DOI: 10.1016/j.compeleceng.2021.107439.

[6] Flamm, B., Peter, C., Büchi, F.N., Lygeros, J., (2021) Electrolyzer Modeling and Real-Time Control for Optimized Production of Hydrogen Gas. Applied Energy.281:116031. DOI: 10.1016/j.apenergy.2020.116031.

[7] González, I., Calderón, A.J., (2019) Integration of Open Source Hardware Arduino Platform in Automation Systems Applied to Smart Grids/Micro-Grids. Sustainable Energy Technologies and Assessments.36(October):100557. DOI: 10.1016/j.seta.2019.100557.

[8] González, I., Calderón, A.J., Folgado, F.J., (2022) IoT Real Time System for Monitoring Lithium-Ion Battery Long-Term Operation in Microgrids. Journal of Energy Storage.51(January):104596. DOI: 10.1016/j.est.2022.104596.

[9] González, I., Calderón, A.J., Portalo, J.M., (2021) Innovative Multi-Layered Architecture for Heterogeneous Automation and Monitoring Systems: Application Case of a Photovoltaic Smart Microgrid. Sustainability (Switzerland).13(4):1–24. DOI: 10.3390/su13042234.

[10] Jaloudi, S., (2019) Communication Protocols of an Industrial Internet of Things Environment: A Comparative Study. Future Internet.11(3). DOI: 10.3390/fi11030066.

[11] Kim, S.M., Choi, Y., Suh, J., (2020) Applications of the Open-Source Hardware Arduino Platform in the Mining Industry: A Review. Applied Sciences (Switzerland).10(14). DOI: 10.3390/app10145018.

[12] Krzysztof, W., Bieganska, M., Paliwoda, B., Górna, J., (2022) Internet of Things in Industry: Research Profiling, Application, Challenges and Opportunities—A Review. .

[13] Kuo, C.J., Ting, K.C., Chen, Y.C., Yang, D.L., Chen, H.M., (2017) Automatic Machine Status Prediction in the Era of Industry 4.0: Case Study of Machines in a Spring Factory. Journal of Systems Architecture.81(October):44–53. DOI: 10.1016/j.sysarc.2017.10.007.

[14] Legha, M.M., Farjah, E., (2020) IoT Based Load Management of a Micro-Grid Using Arduino and HMAS. Iranian Journal of Electrical and Electronic

Engineering.16(2):228–34. DOI: 10.22068/IJEEE.16.2.228.
[15] Minchala, L.I., Peralta, J., Mata-Quevedo, P., Rojas, J., (2020) An Approach to Industrial Automation Based on Low-Cost Embedded Platforms and Open Software. Applied Sciences (Switzerland).10(14). DOI: 10.3390/app10144696.
[16] Navarro, E., Costa, N., Pereira, A., (2020) A Systematic Review of Iot Solutions for Smart Farming. Sensors (Switzerland).20(15):1–29. DOI: 10.3390/s20154231.
[17] Portalo, J.M., González, I., Calderón, A.J., (2021) Monitoring System for Tracking a Pv Generator in an Experimental Smart Microgrid: An Open-Source Solution. Sustainability (Switzerland).13(15). DOI: 10.3390/su13158182.
[18] Wu, D., Liu, S., Zhang, L., Terpenny, J., Gao, R.X., Kurfess, T., et al., (2017) A Fog Computing-Based Framework for Process Monitoring and Prognosis in Cyber-Manufacturing. Journal of Manufacturing Systems.43(2017):25–34. DOI: 10.1016/j.jmsy.2017.02.011.
[19] ELS-900 Series Electro-Optic Level Switch. (209204).