This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

Smart Sensor: SoC architecture for the Industrial Internet of Things

Marcelo Urbina, Tatiana Acosta, Jesús Lázaro, Armando Astarloa, and Unai Bidarte

***Abstract*—Nowadays, the concept of intelligent manufacturing is being introduced, based on the integration of new advanced technologies such as the Internet of Things (IoT), distributed control, data analysis, and cyber-security in the manufacturing area, with the aim of improving manufacturing processes and the articles produced. In this sense, new intelligent devices (Smart Sensors) should be developed that integrate several detection methods (sensors), real-time data analysis and wired and/or wireless connectivity. The main contribution of this paper is the design, implementation and experimental verification of an architecture of a Smart Sensor that satisfies the operational requirements needed by the Industrial Internet of Things (IIoT). Considering the software and hardware adaptability that a Smart Sensor should have, this work takes advantage of the characteristics of the current Field Programmable Gate Arrays (FPGA) and SoC to implement a Smart Sensor for the IIoT. In this sense, the proposed Smart Sensor architecture incorporates real-time operation features, the ability to perform local data analysis, high availability communication interfaces such as High-availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP), interoperability (industrial protocols) and cyber-security. The architecture was implemented with hardware available in the market, IP cores and Python libraries developed by third parties. Finally, to validate the applicability of the architecture in the industry, two test environments were implemented. In the first case, interoperability, high availability, synchronization, and local data processing are validated. The second case aims to determine the delay when adding encryption (cyber-security) in layer 2 communications.**

***Index Terms*—FPGA, Internet of Things, High-availability Seamless Redundancy (HSR), Industrial communication, Smart Sensor, synchronization, cyber-security, Parallel Redundancy Protocol (PRP).**

I. Introduction

ADVANCES in digital electronics and communications networks have allowed sensors to cease to be a simple element that generates an electrical signal associated with a physical phenomenon. Today, sensors also digitize electrical signals, process data and transmit information using a communications protocol. Considering these new features a current sensor (Smart Sensor) is a sophisticated computational platform that can process locally the information collected by transducers and transmit it to other devices through a network infrastructure.

Marcelo Urbina and Tatiana Acosta are with the Department of Electrical and Electronics, Universidad de las Fuerzas Armadas ESPE, Sangolquí Av. General Rumiñahui s/n, Ecuador, e-mail: wmurbina@espe.edu.ec, teacosta@espe.edu.ec.

Jesús Lázaro, Armando Astarloa and Unai Bidarte are with the Department of Electronic Technology, University of the Basque Country, Bilbao 48013, Spain, e-mail: jesus.lazaro@ehu.eus, armando.astarloa@ehu.eus, unai.bidarte@ehu.eus.

Corresponding author: Marcelo Urbina (wmurbina@espe.edu.ec).

Sensors and in particular the Smart Sensor are essential elements for the development of more complex systems, such as the Cyber-Physical System (CPS). There are many definitions of CPS, for example, Lee [1], describes CPS as systems that integrate computing and physical processes, Rajkumar [2] defines CPS as physical and engineering systems whose operations are supervised, coordinated, controlled and integrated into a computing and communication core. Jifeng [3] defines CPS as "3C" systems that integrate Computing, Communication, and Control. Baheti [4] describes CPS as a new generation of systems that integrate computational and physical capabilities, which can interact with humans through new modalities.

In summary, CPS are complex, multidisciplinary systems, characterized by integrating physical components (sensors, actuators), processing (control, data analysis) and communications, which can interact with other systems and humans. A typical diagram of the architecture of a CPS is presented in the Fig. 1. In this diagram three main parts are identified.

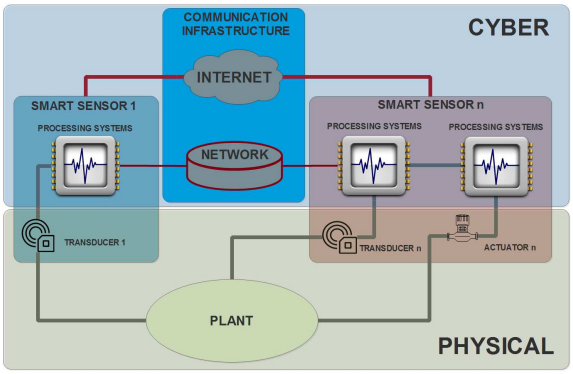

Figure 1. CPS Architecture. The CYBER part is composed of the smart sensor processing systems and the communications infrastructure. The PHYSICAL part is composed of transducers, actuators and processes that control and monitor.

First, the "physical" part refers to physical phenomena that require supervision or control, in short, the plant of a system, which may include mechanical elements, biological or chemical processes, or human operators. Second, we have the computer processing platforms (Smart Sensors), which consist of transducers and actuators (connectors between the "physical" and "Cyber" part), one or more processing systems, and depending on the complexity of one or more operating systems. Finally, the third element is the communications infrastructure, which provides the mechanisms for the different Smart Sensors to exchange information. Together, the processing systems and communications structure make up the "Cyber" part of a CPS [5], [6].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

It should be noted that smart sensors are more frequent and relevant today, largely due to the need to use more and more information in the control of industrial processes. In addition, the new technological advances in the field of electronics, information technologies and computing make possible the construction of this type of systems [7]. These devices need new technological infrastructures such as networks to connect to the cloud, communication protocols to exchange data between machines and with users, Cloud tools to generate new applications for the manufacturer or its ecosystem, enabling the development of new business models. The development platforms should facilitate the development of applications in a faster way, accelerating the "Time to Market", reducing project risk and facilitating the scalability of applications in changing environments such as the industrial sector.

Furthermore, most smart sensors require the integration of different technologies and design methodologies used in various domains, such as electronics, communication, control, wireless networking, distributed computing, real-time systems, security and model-oriented engineering [8]. Some of the concepts and technologies, such as wireless networks and Ethernet-based communications, used in other SMART environments (e.g., SMART-CITIES, SMART-GRID) can be used directly in the industrial sector. However, the particular problems and requirements of this sector, such as Real-time (RT), high availability, interoperability, standardization, flexibility, safety, and cyber-security need to be considered [9], that are entirely addressed by Smart Sensors.

Smart sensors are necessary devices for the development of the Industrial Internet of Things (IIoT) [10]. In IIoT, in addition to the real-time operation, high availability, interoperability, and cyber-security characteristics that Smart Sensors provide, IIoT also takes advantage of the enormous amount of data that Smart Sensors generate and Machine-to-Machine (M2M) communication to incorporate automatic learning and Big Data technologies into the production system [11]. The philosophy behind IIoT is that machines with a high level of intelligence are better than humans at capturing and communicating data accurately and consistently. With this data, companies can more quickly detect malfunctions and problems, even before they occur, saving time and money. Particularly, in the manufacturing industry, the IIoT will allow better product quality control while maintaining traceability and efficiency in the supply chain, with the aim of achieving sustainable and ecological production.

The main contribution of this paper is the design, implementation and experimental verification of an architecture of a Smart Sensor, which can be implemented in a SoC platform, considering the operational requirements that the IIoT needs. In this context, the proposed architecture incorporates real-time operation features (latency, determinism, synchronization), the ability to perform local data analysis, high availability communications (HSR/PRP), interoperability (industrial protocols) and cyber-security.

The paper is organized as follows. Section II describes the requirements and characteristics of a Smart Sensor needed in IIoT. Section III presents an in-depth comparison of architectures and designs of existing and proposed Smart Sensors in IIoT. Section IV details the implementation of the architecture in an FPGA. In Section V, the proposed architecture is experimentally validated in a test environment to verify applicability in industry. Finally, Section VI discusses the conclusions and future work in this field.

II. Requirements for the IIoT

The approach of a device architecture that provides a solution to connectivity, scalability, interoperability, data collection and analysis requirements that are specific to the IoT is difficult. This challenge becomes even more complicated if the requirements of devices and industrial applications are incorporated into the design. This means that there are some challenges to consider that have not yet been addressed. In this sense, IIoT needs an architecture that can handle strict requirements regarding real-time execution (latency, determinism, synchronization), high availability, interoperability, data analysis and cyber-security. These requirements are described below, and a possible solution is proposed.

A. Real-time operation

The "Real-Time" concept in the field of automation implies a response to events or signals in a predictable time after their occurrence. For example, fast digital control loops may require reaction times of less than ten microseconds. In this sense, it is necessary to have mechanisms that allow the synchronization of the different elements that make up a network of sensors to guarantee the operation of a system in real time.

Traditionally, the most widely used synchronization method (especially in the electricity sector) was defined by the Inter-Range Instrumentation Group (IRIG-B) [12], [13], but it required dedicated infrastructure which increased costs considerably. Also, complex calibration processes were required to compensate for the variable propagation delays of the signals [14]. In this context, synchronization protocols were developed that are transmitted over the same data network, to reduce implementation costs. The first synchronization protocol to be proposed was Network Time Protocol (NTP), which was used in applications where the required accuracy was below milliseconds [15]. In NTP the accuracy depends on the extent and complexity of the network. In a LAN the accuracy is in the order of milliseconds [16], while in a WAN it can be increased up to ten milliseconds [16]. However, for industrial networks, clock synchronization requires greater precision. In this sense, the IEEE 1588 Precision Time Protocol (PTP) standard [17], first adopted in 2002 for automation and measurement applications, establishes a method for clock synchronization with microsecond precision. Subsequently, in 2008, the second version of the standard was ratified to address telecommunications, automation and control applications, power generation, transmission and distribution systems. Using the IEEE 1588-2008 protocol, a master device distributes its reference time to the rest of the nodes, over conventional data networks such as Ethernet, so that the secondary nodes are synchronized both in phase and frequency.

One of the objectives of the Smart Sensor architecture proposed in this work is the incorporation of synchronization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

technologies such as IEEE 1588-2008. In this way, any node connected to the network, even the simplest, will have a shared clock with precision in the nanosecond range.

B. High availability

The basic method of increasing availability is to add redundancy to those elements where failures occur most frequently. In this context, in the field of communications, the standard for High availability automation networks IEC 62439-3 has been developed and is now published in its second edition [18]. This part of IEC 62439 specifies two redundancy protocols designed to provide seamless recovery in case of single failure, the two protocols contemplated are Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR). These protocols are the only ones that offer zero recovery time from a network failure, and this means that there will be no loss of information [19], [20].

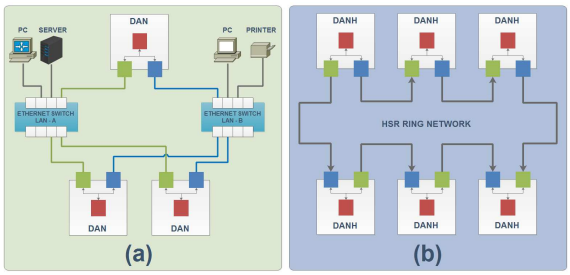

Figure 2. High availability automation networks. **(a)** Parallel Redundancy Protocol (PRP) and **(b)** High-availability Seamless Redundancy (HSR)

For a device to be able to operate with HSR or PRP protocols, it must have two Ethernet interfaces. HSR and PRP are Zero Packet Loss (ZPL) protocols because when all packets are duplicated, in case of loss of connection between two nodes, the destination node will always receive a packet through one of the interfaces.

In the case of PRP protocol, each port of a node is connected to two independent and conventional Ethernet networks (LAN A and LAN B), the structure of a PRP network is shown in Fig. 2(a). PRP nodes are called Dual Attached Nodes (DANs) [21] . PRP frames can run on the infrastructure of a conventional Ethernet network. In PRP networks it is possible to connect devices with a single Ethernet interface called Single Attached Nodes (SANs), such as computers, printers, etc.

HSR, as well as PRP, offers redundancy by sending packets over two Ethernet ports, but unlike PRP, HSR nodes form a ring through their two Ethernet ports [20], the structure of an HSR network is shown in Fig. 2(b). In this case, the nodes are called Doubly Attached Bridging Nodes HSR capable (DANHs) [19]. The DANH nodes, besides managing the elimination of redundant frames, and the emission of supervision frames, must also forward non-target frames. The latency times set in the standard for frame relay are very demanding (cut-through). For this reason, unlike PRP, HSR frames cannot be relayed over conventional Ethernet equipment [19].

C. Interoperability.

International organizations focus their efforts on setting standards with the aim of unifying the architecture of communications systems, for example in the electricity sector have raised the standard IEC 61850 [22], in the industrial IEC 60870 [23], among others. These standards provide solutions for future implementations, but today all industrial sectors continue to use old equipment that meets their goals, so companies do not plan to upgrade their facilities. Until the old devices are replaced, interfaces that can handle various types of industrial communication protocols must be incorporated into the Smart Sensor. In an industrial environment, the interface and communication protocol are called fieldbus. Two kinds of fieldbus can be identified: Ethernet-based fieldbus and serial fieldbus.

Ethernet-based fieldbus is used in environments where higher bandwidth, higher performance, real-time operation and the ability to connect more nodes to the network over longer distances are required. It also supports information technology (IT) (e.g., TCP/IP, UDP, SNMP, FTP...). Examples for the Ethernet fieldbus are Profinet, EtherCAT, Modbus / TCP, Ethernet / IP, Powerlink Ethernet, Sercos III, CC-Link IE, among others [24]. Some of these protocols, in order to add real-time operating features to them have to be implemented entirely in hardware.

In contrast, in low data rate applications and networks with few devices, serial fieldbus is used. The RS-485 and CAN standards are the basis for the implementation of this type of fieldbus. Examples of serial fieldbus are Modbus, Profibus, Interbus, DeviceNet, CC-link, among others [24].

As already mentioned, one of the major problems when considering an automation solution is the diversity of devices with different communication protocols. For this reason, it is necessary to develop tools that allow access to industrial process data in a standardized way. One such tool is Object Linking and Embedding (OLE) Process Control (OPC), which is a standard that establishes a common language for the exchange of information between different devices in the area of industrial automation [25]. OPC is used to address one of the most significant challenges of the automation industry: how to communicate devices, drivers, and applications without falling into the usual problems of proprietary protocol-based connections [26]. OPC-UA is the evolution of the OPC standard, and this version natively incorporates security mechanisms to establish secure and reliable communications independent of the platform manufacturer. The capabilities offered by OPC (OPC-UA latest update) show that standardization is indispensable and marks a revolution in automation systems [27].

D. Data analysis.

Nowadays, trends such as globalization, energy efficiency, and customized products are forcing production lines to be more efficient, faster and highly personalized. To address what has been called intelligent manufacturing or the fourth industrial revolution, the need arises to develop tools and facilities, with the ability to adapt in the shortest possible time to new

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

production requirements. In this sense, the Smart Sensors must modify their behavior based on the conditions of their current context. For this purpose, Smart Sensors must have the ability to acquire and build knowledge, using, for example, successful experiences from previous executions. Technologies such as cloud computing and Big Data could be used to provide learning capabilities and adaptation to Smart Sensors [28]. Big Data (BD), involves the analysis, management and intelligent manipulation of a huge amount of data that cannot be processed or analyzed using a traditional infrastructure (software, hardware). Intelligent data exploration using description, prediction and optimization models improves the management of available resources. Cloud Computing (CC), it is a shared platform of computational resources such as servers, storage, and applications, and access to these resources is possible from any mobile or fixed device with Internet access. Cloud solutions will become increasingly crucial to the development of the fourth industrial revolution, for example, cloud storage makes any information accessible from anywhere, facilitating access to and analysis of data not only between factories but across the entire value chain. In short, everything in and around a manufacturing operation (suppliers, the plant, distributors, even the product itself) is digitally connected, providing a highly integrated value chain [29].

For the implementation of the algorithms of data analysis, specialized programming languages are used in the processing of large volumes of data, such as SAS, which is a leader in data analysis but is proprietary software, R and Python used in academic environments or research, and they are open source. Although Python was created as a general-purpose programming language, it has a series of libraries and development environments for each of the phases (obtaining, processing, cleaning, analysis, modeling, visualization) of the data analysis process. This feature added to the power, open source character and ease of learning have made it the first choice for developing data analysis applications, leaving SAS and R in second place [30].

E. Cyber-security.

Communication networks and cyber-security are critical parts in the deployment of reliable and efficient automation systems. This security problem is very complex and needs to be addressed from a multilayer approach: devices, systems, networks, users, software applications, among others. To the extent that communications interact with teams that conduct critical operations, cyber-attacks that put the integrity of the communications management teams at risk may end up resulting in significant economic and even personal losses. The following references describe attacks that have reached automation systems [31]–[33].

The use of secure industrial protocols is recommended to increase the security of industrial communications. During the last ten years, several organizations have raised specifications in the industrial field to solve security problems in the different areas of operation and communication of industrial environment. For example the OPC Foundation raises OPC UA (Open Connectivity Unified Architecture), The International Electrotechnical Commission (IEC) through the IEC 62351 series defines standards to provide security mechanisms to communication protocols that were defined by the IEC working group TC57 [34].

Currently, the IEC TC57 working group and researchers around the world are developing several safety proposals that were not initially considered in the standard. For example, the processing times of encryption algorithms based on Secure Hash Algorithm (SHA) do not allow meeting the time requirements of high priority frames. In this sense, a research team analyzed the security problems in the PTP synchronization protocol and proposed replacing the old Message Authentication Code (MAC) algorithms based on Secure Hash Algorithm (SHA) with new algorithms based on the Advanced Encryption Standard (AES) [35]. This work stands out for proposing a global security solution that can be implemented in any industrial Ethernet protocol that works at Layer 2 level. This contribution should be taken into account in particular in the Smart Sensor architecture that is proposed.

III. Implementation.

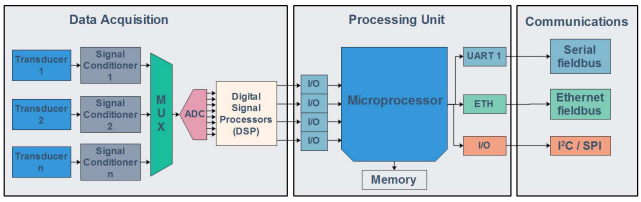

Figure 3. Generic architecture of an intelligent sensor. The **Data Acquisition** module is responsible for collecting data. **Processing Unit** is responsible for controlling all the elements that make up a Smart Sensor. **Communications** module is responsible for transferring the data generated in the Smart Sensor.

A Smart Sensor consists of a processing unit, a data acquisition module and communications module [36]. Fig. 3 shows a high-level block diagram of the architecture of a generic Smart Sensor.

The data acquisition module is responsible for collecting data (such as temperature, pressure, image, sound) from the physical environment and sends it to the processing unit. This module is composed of one or several transducers, signal conditioners, Analog to Digital Converter (ADC) and Digital Signal Processors (DSP). At present all the elements that make up the acquisition module can be integrated into a single circuit called MicroElectro-Mechanical Systems (MEMS). MEMS uses microfabrication technology to integrate miniaturized mechanical and electromechanical elements into electronic devices, such as accelerometers and gyroscopes [36], [37].

The processing unit is responsible for controlling all the elements that make up a Smart Sensor. It also manages the use of resources such as the communications module, memory, Input/Output (I/O) peripherals and application execution. There are several types of devices that can be used as processing units, for example, microcontrollers, SoC and FPGA. The choice of each depends on the complexity and functionality of the Smart Sensor [36], [38]. Also, in the processing unit is where the data is manipulated using specialized software, and the results can be sent to central stations or presented

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

graphically to the user or in tables easy to interpret. Software developed for a smart sensor must be able to adapt to changes in the structure of the smart sensor and be easy to use. It is required collaboration and cooperation between informatics experts, those responsible for the management of the system to which the application is directed and users.

The communications module is responsible for transferring the data generated in the Smart Sensor to local or remote control and monitoring stations. Depending on the amount and distance of data transmission, the communications module incorporates several low (RS-232, RS-485) and high-speed (Ethernet, SPI) interfaces. The universality together with the communication protocols that can be implemented over Ethernet make it the ideal means to link different devices in the industrial environment. Ethernet allows sensors in the process network to be interconnected directly with the devices in the management network, eliminating the use of protocol converters (gateways) that increase runtimes and limit their use in real-time applications.

Technological advances in the fields of FPGA and SoC has revolutionized the way electronic systems are designed. The FPGA has evolved from being a simple tool for the creation of prototypes to being an essential solution for the development of devices that require high processing capacities, real-time operation requirements, interoperability, flexibility, safety, and high availability. The current FPGAs, in addition to the large resources they integrate (millions of logical cells, various types of memory and peripheral interfaces), also integrate ARM processors implemented in silicon. Thanks to these new FPGA manufacturing techniques, the sensors are getting smaller and can perform more processing, allowing the execution of more complex applications, such as machine learning algorithms and data analysis, introducing the Big Data concept in the field of sensors, giving rise to so-called Smart Sensors. The flexibility of the FPGA offers the possibility of adding these new features, in [39]–[43] some developments in this respect are outlined. [39] Describes the design of neural networks in FPGA. [40] Intel proposes the use of OpenVINO as a tool for the development of artificial intelligence and machine learning projects. The use of accelerators for use as IP cores in machine learning is proposed in [41], [42]. Finally in [43] we can find an extensive review on this area of research. At the software level, the Linux operating system offers possibilities to extend the capabilities of the Smart Sensor, for example with the use of development tools such as Python it is possible to provide the system with the ability to execute industrial communication protocols such as Modbus. The fundamental challenge of this work is to take advantage of the features of the current FPGAs that integrate silicon-embedded processors to propose a hardware-software architecture of a Smart Sensor.

The proposed work considers the use of a SoC platform (microprocessor + FPGA) for the implementation of an intelligent sensor for the IIoT. In the proposed architecture, all IIoT requirements have been considered, in this sense, the architecture incorporates: a) A processing unit (microprocessor) to run the architecture management software, and to perform data processing and analysis (Big Data). b) An IP Core 1588 to ensure synchronism in the order of nanoseconds. c) An IP Core HSR / PRP to provide high availability in communications and an IP core for industrial communications such as PROFINET to ensure interoperability with other devices. d) An IP Core to perform asymmetric encryption of layer 2 Ethernet frames using the AES-GCM algorithm. e) I/O module are also included in order to add more functionality to the Smart Sensor. The use of an encryption module fully implemented in hardware reduces the use of resources in the microprocessor.

The scheme of the Smart Sensor architecture proposed for the IIoT is presented in the Fig. 4. Five modules can be identified in the architecture to support the different functionalities that Smar Sensor must have, which are presented below:

The processing module (PS), allows executing the necessary software to manage all the components of the architecture, to run specific libraries of a communications protocol and to perform data processing and analysis (Big Data). The PS has Ethernet and serial interfaces (RS-232, I^2C) to communicate with the exterior or with the internal modules (HSR/PRP, IEEE 1588, etc.). In the processing module, a 1Gbps Ethernet port (GMAC1) can be identified. This port will be used as an interface to access a local network or the Internet, to provide the system with access to services such as web, FTP, database, cloud, among others. Additionally, the GMAC0 is used to interconnect the PS with the IP Cores that are implemented in the PL. There are also two serial interfaces, the first (UART0) is used to implement industrial communications via serial field buses (Modbus, Profibus), and the second (UART1) is used as a terminal for monitoring, configuring and controlling the Smart Sensor.

IEEE 1588 module is used to support applications with low synchronization time and timestamping. This IP provides an exceptional synchronization mechanism that requires only an Ethernet connection for nanosecond range synchronization. All processes are carried out using hardware modules and do not need any software to manage their operation. This IP core can run on CPU-less boards and can be embedded into any Ethernet IP Switch or IP cores compatible with Transparent Clock operation. The IEEE 1588 module is used to perform time stamping and supports the HSR protocol.

The communications module, allows managing high availability communications (HSR/PRP) and industrial communications (Profinet, EtherCAT, EtherNet / IP, among others). This IP implements Ethernet connectivity ensuring zero-delay recovery time in case of network failure and no-frame lost. The IP supports the latest version of High-availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP) standards in combination with redundant IEEE 1588-2008. The communications module used consists of four Ethernet interfaces, two of which are used as logic inputs to handle high-availability communication protocols (HSR / PRP), and the other two allow the connection of Ethernet devices (SAN) that do not have HSR/PRP functionality, through the Redbox configuration.

The encryption module has the advantage of not requiring the use of a microprocessor for its operation. The encryption module is a cryptographic engine that is capable of encrypting, decrypting and authenticating the Ethernet frames of layer 2 using the AES-GCM algorithm implemented in the hardware.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

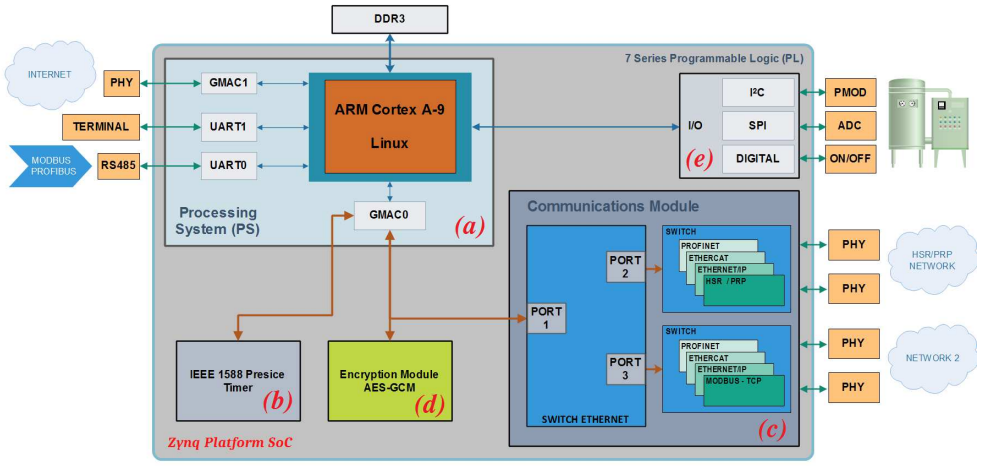

Figure 4. Block diagram of the architecture of a Smart Sensor for the IIoT.

This module uses symmetric key cryptography, which relies on the 128-bit AES-GCM algorithm to provide encryption and data authentication. AES-GCM has been selected for its cryptographic capabilities, resource utilization and performance achieved, especially in hardware implementations, although with minor changes the module could support almost any cryptographic algorithm. This cryptographic module is an all in hardware solution that provides a great balance between performance and resources. Also, this approach allows to minimize latency and increase efficiency at the same time compared to other software-based cryptographic solutions.

The architecture also has I/O ports to add interfaces for sensor reading using I^2C communications or Analog/Digital (A/D) converters.

Hardware description language (VHDL) was used to describe the hardware of the proposed architecture. Each module of the architecture corresponds to and IP core developed by SoC-e [44]. The amount of resources needed to implement the hardware described in Fig. 4, is presented in Table I.

Table I
FPGA RESOURCES USED FOR THE SMART SENSOR ARCHITECTURE IMPLEMENTATION

Resources	Communications Module	IEEE 1588 Module	Encryption Module	Full Design	Available	Percent (%)
Slice LUTs	42780	1408	4748	48936	53200	91.98
Slice Registers	28493	2350	4138	34971	106400	32.87
F7 Muxes	2231	65	-	2296	26600	8.63
F8 Muxes	874	31	-	905	13300	6.80
Block RAM	46	6	5.5	57.5	140	41.07

As it can be seen, the most used resources in the implementation are the Slice LUTs and represent 91.98% of those available in an XC7Z020 FPGA [45], which is the one used in the implementation and validation of the Smart Sensor that is proposed. It is also noted that the communications module is the part of the architecture that uses most of the resources.

The amount of power required for the whole system is 6.12 W, while the FPGA requires 2.53 W. This device is intended for industrial environments such as an electrical substation, where operation requirements (latency, synchronism, bandwidth) are stringent and communication links between devices are wired, therefore, the proposed Smart Sensor architecture is not designed to be powered directly by batteries, in case of power failure they are connected to a UPS backup system so the operation is not interrupted.

The software tools used for the design are listed below:

- Vivado 2017.2: It is used for the hardware design and to interconnect the interfaces of the PS with the peripherals implemented in the PL.
- Software Development Kit (SDK): It is used to define both the device tree and the FSBL (First Stage Boot Loader), and to generate the zynq boot image needed to run the Linux operating system on the SoC.
- Python and Java: It is used to program, to compile and to execute applications.

IV. RELATED WORK AND COMPARISON WITH EXISTING SCHEMES

With the evolution of technology towards Industry 4.0 [46] and IIoT, a great variety of research have been carried out on Smart Sensors in fields as diverse as transportation industry (automotive, air, maritime and aerospace) [47], intelligent buildings [48], energy networks [49], agriculture [50], medical devices [51], defense [52], and industry [53], [54].

In this regard, a study was carried out on the development of IIoT devices that consider the requirements analyzed in Section II. In table II, a comparison is presented between the proposed work and similar works of Smart Sensor designs and architectures for use in the IIoT. All the developments presented above implement different intelligent sensor architectures with sufficient processing capacity to perform data analysis and digital communication capability. In general, these architectures are characterized by the use of a single intelligent device that is responsible for both the conditioning of the signal from the transducers and the attention to the digital communication protocol implemented.

The works [55]–[57] present an intelligent sensor architecture based on a microprocessor, that provides flexibility at the software level by allowing for adding new functionalities to the device. The type of microprocessor and its processing capability will determine the amount and complexity of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

applications that can be added to the device, such as data access mechanisms (web, FTP, among others.) and security for layer three frames (VPN, IPSec). On the other hand, all the architectures incorporate serial, Ethernet and wireless interfaces that allow interoperability with other devices. As negative points, these designs do not incorporate the mechanisms used to guarantee the synchronization, high availability, and security of layer 2 communications used in industrial control systems.

With a different approach [58] and [59] propose the use of an FPGA as a platform to implement an intelligent sensor. The use of FPGAs allows guaranteeing the execution of the processes in exact times (determinism) and with greater rapidity. The negative point of this approach is that the designs are specific to an application. Without a microprocessor, it is difficult to change the software and add new features to the device. Finally, in [60]–[62] they propose the use of a SoC platform that incorporates processors implemented in silicon and an FPGA. This configuration takes advantage of software flexibility that allows an architecture with a microprocessor. At the hardware level (FPGA), the choice of internal architecture will allow offering more significant benefits to the device, for example, high availability communications (IP Core HSR / PRP), synchronization (IP Core 1588), among others. As a negative point, these architectures [60]–[62] do not incorporate all the features required by a device for IIoT. For example, in [60], [61], they do not incorporate synchronization mechanisms and high availability communications.

It is important to highlight that the architectures proposed in the works shown in the table II, none of them completely satisfy all the requirements of IIoT and, in particular, do not have security mechanisms for layer 2 Ethernet frames.

The difference between our work and the literature is that the proposed architecture aims to be general enough to provide a solution to the requirements posed by the IIoT. Any of the application scenarios presented in the works presented in the table II can be re-implemented using our architecture.

Table II
COMPARISON WITH RELATED WORKS THAT FOCUS ON THE DESIGN OF INTELLIGENT SENSORS FOR THE IIOT.

Feature	[38]	[41]	[39]	[40]	[42]	[43]	[44]	[45]	This work
Platform									
Microprocessor	√	-	√	√	-	-	-	-	-
FPGA	-	√	-	-	√	-	-	-	-
Microprocessor + FPGA	-	-	-	-	-	√	√	√	√
Real-time									
Processing Speed	*low*	*low*	*high*	*high*	*high*	*high*	*high*	*high*	*high*
Synchronization	X	X	√	X	X	X	X	√	√
Interoperability									
Serial Interfaces	√	X	√	√	X	√	√	√	√
Ethernet interfaces	√	X	√	√	√	√	√	√	√
Data analysis	X	X	√	X	X	X	X	X	√
High availability	X	X	X	√	X	X	X	√	√
Cyber security	*Layer 3*	X	*Layer 3*	*Layer 3*	X	*Layer 3*	X	X	*Layer 2-3*

V. VALIDATION

As indicated in the previous section to implement the proposed Smart Sensor architecture, it is necessary to have an electronic card that integrates a SoC with a significant amount of resources. With this consideration, an electronic card that integrates an FPGA of the Xilinx Zynq-7000 family was used. This FPGA contains a dual-core ARM Cortex-A9 ARM processor that will allow executing an operating system (e.g., Linux) in one of the processors and executing applications that have time constraints in its execution in the second processor. The FPGA also has a section of Programmable Logic (PL) of the last generation of 28nm.

For the management of all hardware resources, at the software level in the processing module was installed a Linaro operating system (linaro-vivid-developer-20150914-710), which is a version of Linux for embedded systems. Software needed to load the bitstream, device tree and FSBL is developed in Xilinx SDK. Finally, all the necessary software (Python and JAVA) are installed to compile and execute tools for data analysis, industrial protocols (Modbus) and monitoring (web server). For example, with Python, the MODBUS industrial protocol was implemented through the minimalModbus library. Communication modes that can be implemented with this library are Modbus-ASCII and Modbus-RTU. MinimalModbus library contains all functions required to manage the requirements of a serial Modbus communication [63]. Python was used to run the data analysis application. The NumPy, SciPy, Matplotlib and Pandas libraries, among others, approach all the necessary stages to implement a data analysis application. These libraries have tools for data collection, classification, processing, and presentation. Also, it has automatic learning functions, real-time execution and can be executed in embedded systems that run the Linux operating system that support Python.

In order to verify the operation of the Smart Sensor architecture, two test environments have been implemented. First, the operation of the Smart Sensor is verified regarding interoperability, high availability, synchronization, and local data processing. Finally, a test environment was implemented to demonstrate the encryption of layer 2 Ethernet frames and determine the delay that this adds in the communications.

A. Case 1

This test environment was implemented to validate the approach of integrating several communication interfaces (interoperation), high availability, synchronization and data analysis into a device. The design has been developed with the hardware and software tools available in the market.

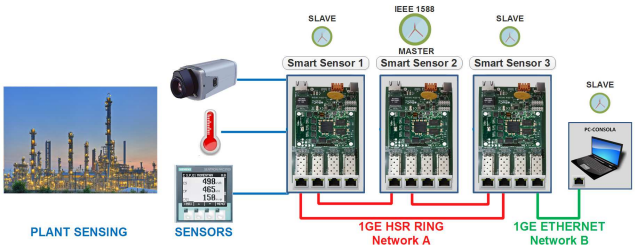

Figure 5. Concept-proof setup for Smart Sensor. Case 1, Smart Sensor 1 collects sensor data through the digital, analog, Ethernet (video) and serial interfaces (MODBUS). Smart Sensor 2: It works as HSR node (DANH) and master IEEE 1588. Smart Sensor 3: It works as a Redundancy Box (RedBox) to connect the HSR ring with the regular Ethernet network. The PC is used to access the data stored in the Smart Sensor 1 through a web server.

The Concept-proof setup used is presented in Fig. 5. In the schematic, three smart sensors can be identified interconnected

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

via a 1 Gbps HSR network that provides zero seconds of delay in the event of failure. To remotely access the data generated by the smart sensors, a 1 Gbps Ethernet link is used between the smart sensors (3) and a computer. The IEEE-1588 IP core used allows obtaining synchronization times in the order of 20 nanoseconds, in [64] it is shown a detailed description of the synchronization times obtained with this IP core.

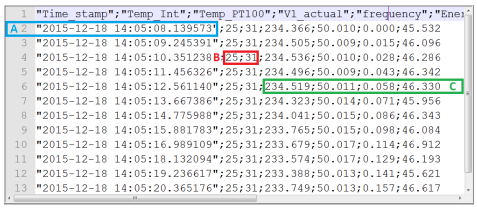

Figure 6. Smart Sensor data acquisition. a) Timestamping data capture. b) Temperature data (PT100) and internal temperature of the smart sensor. c) Energy parameters.

The three nodes (Smart Sensor) connected in the HSR network implement the architecture shown in the figure 5, and additionally each node executes the following functions:

Smart Sensor 1: It collects sensor data through the Ethernet (video) and serial interfaces (RS232, RS485, I^2C). The data that are captured in real time is, a) temperature using a PT100 and b) electrical parameters, energy consumption data and network frequency using the SENTRON PAC3100 device, which uses a Modbus interface to transmit the information. This information is processed internally to perform control actions (in this case control of a traffic light). This node also performs data analysis using the python libraries. This application compares the current status of the configuration being monitorized with the plant model and forecasts the system failure point. Also, graphical reports of the results are generated internally and can be accessed remotely from the computer.

Smart Sensor 2: It works as HSR node and it is used to complete the HSR ring and implement Master IEEE-1588 clock.

Smart Sensor 3: It works as a Redundancy Box (RedBox) to connect the HSR ring with the regular Ethernet network, that is, it allows to transform the computer into an HSR compatible device.

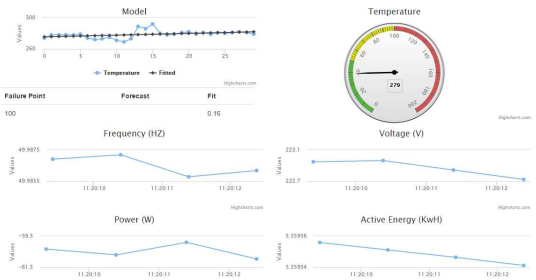

Figure 7. Graphical report of the data analysis performed by Smart Sensor.

Once the equipment was configured, basic tests were performed to verify the operation. In the Fig. 6, the temperature data (PT100), time stamping data and the energy parameters that are collected by the Smart Sensor 1 are displayed, which are processed and stored locally.

With the PC that is connected to the Smart Sensor 3 (RedBox), the data stored in the Smart Sensor 1 is accessed remotely. This information is observed graphically using a web browser. The results are shown in the figure 7.

B. Case 2

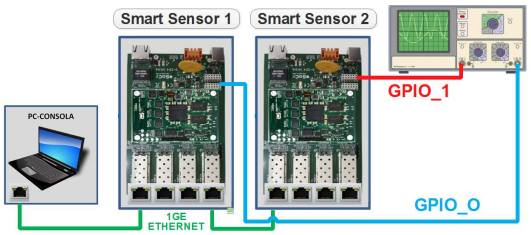

Figure 8. Concept-proof setup for Smart Sensor. Case 2, Smart Sensor 1: reads the state of the GPIO_0 pin and generates an Ethernet frame and executes the encryption. Smart Sensor 2: Process and decrypt the ethernet frame and write the data to the GPIO_1 pin. The PC is used as a terminal to run applications on the Smart Sensor.

The Concept-proof setup used in this case is shown in Fig. 8. In the scheme, two intelligent sensors interconnected via a 1 Gbps Ethernet link can be identified. The computer is used as a terminal to run the applications on the Smart Sensors. Finally, to validate the processing times, measurements were made using an oscilloscope. The first channel is connected to GPIO_0 of the Smart Sensor 1 that is configured as input. In addition, the first channel is used as a trigger signal. The second channel is connected to the GPIO_1 of Smart Sensor 2 that is configured as output. The objective of this scheme is to measure the processing time from the moment the Smart Sensor 1 pin GPIO_0 is activated until its status is reflected on the Smart Sensor 2 pin GPIO_1. For this purpose, Smart Sensor 1 has a minimal program that reads the status of the pin (GPIO_0), generates an Ethernet frame with a specific ethertype so that the encryption module identifies it and can perform the encryption process in Smart Sensor 1 and decrypted in Smart Sensor 2. The encryption IP core can identify Ethernet frames with a specific ethertype, extract the payload and encrypt/decrypt it with the AES-GCM algorithm, and finally reconstruct and send the frame with the encrypted data. In Smart Sensor 2, a program was implemented in C that reads the decoded Ethernet frame, extracts the status information from the GPIO_0 Smart Sensor 1 and replicates it in the local GPIO_1.

The execution times of the implemented applications are variable because the Linux operating system does not guarantee a deterministic execution. In order to determine the quality of the solutions, the oscilloscope was used with persistence. Fig. 9 shows the capture of the processing time from the moment the GPIO_0 pin is activated, the encryption (Smart Sensor 1) and decryption (Smart Sensor 2) processes are performed, until its state is reflected in the GPIO_1 pin of Smart Sensor 2. The total time from the change of state in pin GPIO_0 (Smart Sensor 1) until the change in pin GPIO_1 (Smart Sensor 2) is: minimum 211.0 µs and maximum 319.4

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2908264, IEEE Internet of Things Journal

Table III
PROCESSING TIMES OBTAINED IN THE EXPERIMENT. 1) EXECUTION TIME OF THE TEST WITHOUT ENCRYPTION. 2) EXECUTION TIME OF THE TEST WITH ENCRYPTION. 3) DELAY INTRODUCED BY THE ENCRYPTION PROCESS (SMART SENSOR 1) PLUS THE DECRYPTION PROCESS (SMART SENSOR 2).

Frame(Bytes)	(1) Without_encryption (μs)					(2) With_encryption (μs)					(3) (2) - (1) (μs)	
	min	max	$\overline{x}$	σ^2	σ	min	max	$\overline{x}$	σ^2	σ	Δd_min	Δd_max
100	86,21	189,00	106,80	924,73	30,41	96,50	198,80	124,93	988,70	31,44	9.80	10.29
1200	171,80	278,70	187,76	1397,31	37,38	211,00	319,40	249,34	1343,11	36,65	39,20	40,70

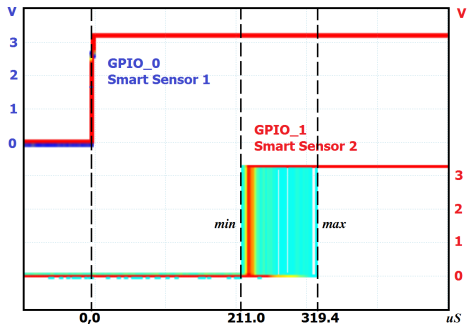

Figure 9. Time from the activation of pin GPIO_0 (Smart Sensor 1) until its status is reflected on pin GPIO_1 (Smart Sensor 2). The processing time using Ethernet frames of 1200 bytes is minimum 211.0 µs and maximum 319.4 µs.

µs, these times correspond to the use of Ethernet frames of 1200 bytes as a mechanism to transfer information between the two Smart Sensors.

For the tests, 100 and 1200 byte Ethernet frames were generated in order to cover a wide range of control frames used in industrial applications. Although the common ones are small frames due to the strict time requirements, for example in the electrical sector the Sample Values (SV) frames are from 160 to 180 bytes and the Generic Object Oriented Substation Events (GOOSE) frames are from 92 to 250 bytes. The Table III summarizes the processing time data obtained from the experimental measurement with the oscilloscope, data of minimum, maximum and average value ($\overline{x}$) are presented, as well as statistical data of variance (σ^2) and standard deviation (σ).

VI. CONCLUSIONS

The latest generation of programmable devices (FPGA, SoC) have allowed the development of electronic devices that are interconnected and are responsible for more complex activities. FPGAs have reached a high level of development regarding performance, energy consumption and cost. The fundamental challenge of this work is to take advantage of the features of the current FPGAs that integrate silicon-embedded processors to implement a hardware-software architecture of a Smart Sensor. The processing of frames with real-time requirements will be implemented as logic circuits (hardware), and the highest level algorithms will be performed in the high-performance processor (software). These two systems are closely linked together on a single chip, and the success of the whole depends on the selected architecture for exchanging information between them. The ability to reconfigure the hardware and the possibility of adding applications based on Linux allows the proposed architecture to be scalable and can be used in different industrial environments. The Smart Sensors architecture presented in this work has demonstrated this approach, and considers the use of a SoC platform (microprocessor + FPGA) for the implementation of an intelligent system for the IIoT.

The FPGA used has two high-performance ARM processors running the Linux operating system and Big Data analysis libraries. In the PL of the FPGA, a communications module with HSR interfaces was implemented to provide redundancy in communications and avoid loss of information. The IEEE 1588 module was also added to perform time marking and maintaining synchronization between the devices in the network. The new architecture offers advantages regarding response time, system integration, as well as flexibility and adaptability to the different applications and communication protocols to be implemented in the future. Finally, the tests performed and the values obtained demonstrate the correct functioning of the intelligent sensor implemented. Tests have also shown that the Smart Sensor implemented on platforms based on FPGA is easily integrated into any industrial environment. Also, it is important to note that the proposed architecture satisfies all the requirements of IIoT, in particular, it has security mechanisms for layer 2 Ethernet frames. In this regard, in the encryption test the response times obtained for 100 and 1200 byte Ethernet frames were 10.29 and 40.70 µs respectively, which allows their use in a large number of industrial applications.

Future work includes expanding the functionalities of the architecture to support a wide variety of applications and developing tools that automate our approach. For example, adding a module for high-speed wireless communications with mechanisms to guarantee high availability and security. Another point to develop is a method that allows reconfiguring the Smart Sensor remotely. The reconfiguration of the FPGA allows implementing different types of hardware/software architectures on the same chip for a wide variety of sensors. Also the dynamic reconfiguration allows to reprogram the FPGA in execution time, it is possible to implement different hardware depending on what the Smart Sensor is detecting, all this when the device is running.

ACKNOWLEGMENT

This work has been supported by the Ministerio de Economía y Competitividad of Spain within the project TEC2017-84011-R, ZE-2017/00022 - NEWCAUTO project co-funded by the Basque Government and by FEDER 2014-2020 funds and it has been carried out inside the Research and Education Unit UFI11/16 of the UPV/EHU and supported by the Department of Education of the Basque Government within the fund for research groups of the Basque university system

IT978-16 and within the project TFactory ER-2014/0016. Also, UPV/EHU and Universidad de las Fuerzas Armadas ESPE PhD scholarship funding are acknowledged.

REFERENCES

[1] E. A. Lee, "Computing Foundations and Practice for Cyber- Physical Systems : A Preliminary Report," Tech. Rep., 2007. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-72.pdf

[2] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems," in *Proceedings of the 47th Design Automation Conference - DAC'10*. New York, USA: ACM Press, 2010, p. 731. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1837274.1837461

[3] L. Hu, N. Xie, Z. Kuang, and K. Zhao, "Review of Cyber-Physical System Architecture," in *2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*. IEEE, 2012, pp. 25–30. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6196100

[4] H.-M. Huang, T. Tidwell, C. Gill, C. Lu, X. Gao, and S. Dyke, "Cyber-physical Systems for Real-time Hybrid Structural Testing: A Case Study," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, 2010, pp. 69–78. [Online]. Available: http://doi.acm.org/10.1145/1795194.1795205http://portal.acm.org/citation.cfm?doid=1795194.1795205

[5] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 12, pp. 4242–4268, 2014. [Online]. Available: http://itiis.org/download.jsp?filename=TIISVol8,No12-1.pdf

[6] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*, 2014. [Online]. Available: leeseshia.org/releases/LeeSeshia{_}DigitalV1{_}08.pdf

[7] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog Computing for Sustainable Smart Cities: A Survey," vol. 0, no. 0, mar 2017. [Online]. Available: http://arxiv.org/abs/1703.07079

[8] S. Yinbiao and et al., "Internet of Things: Wireless Sensor Networks," Tech. Rep. December, oct 2014. [Online]. Available: http://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf

[9] T. Lennvall, M. Gidlund, and J. Akerberg, "Challenges when bringing IoT into industrial automation," in *2017 IEEE AFRICON*. IEEE, sep 2017, pp. 905–910. [Online]. Available: http://ieeexplore.ieee.org/document/8095602/

[10] I-scoop, "The Internet of Things (IoT) – essential IoT business guide." [Online]. Available: https://www.i-scoop.eu/internet-of-things-guide/

[11] M. H. ur Rehman, E. Ahmed, I. Yaqoob, I. A. T. Hashem, M. Imran, and S. Ahmad, "Big Data Analytics in Industrial IoT Using a Concentric Computing Model," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 37–43, feb 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8291112/

[12] "IRIG Standard 200-04, Overview of IRIG-B time code standard," pp. 1–6, 2011. [Online]. Available: http://metis.ipfn.ist.utl.pt/@api/deki/files/184/=TN-102_IRIG-B.pdf

[13] "IRIG Standard 200-04, IRIG serial time code formats," 2004. [Online]. Available: http://www.irigb.com/pdf/wp-irig-200-04.pdf

[14] J. C. Eidson, *Measurement, control, and communication using IEEE 1588*. Springer Science & Business Media, 2006.

[15] C. Ozansoy, A. Zayegh, and A. Kalam, "Time synchronisation in a IEC 61850 based substation automation system," *2008 Australasian Universities Power Engineering Conference*, no. January, pp. 1–7, 2008.

[16] Y.-S. Li, G. Crispieri, and H. Wohlwend, "Using Network Time Protocol (NTP): Introduction and Recommended Practices," International SEMATECH Manufacturing Initiative, Austin,USA, Tech. Rep., 2006. [Online]. Available: http://www.sematech.org/docubase/document/4736aeng.pdf

[17] "IEEE 1588-2008 Standard for a precision clock synchronization protocol for networked measurement and control systems," pp. 1–300, jul 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4579760/

[18] Iso, "IEC 62439-3 Ed.2.0: Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)," http://www.iec.ch/, p. 84, 2012.

[19] J. T. Yu, "A practical and effective approach to implementing High Availability Seamless Redundancy (HSR)," in *2017 IEEE Conference on Dependable and Secure Computing*. IEEE, aug 2017, pp. 392–399. [Online]. Available: http://ieeexplore.ieee.org/document/8073824/

[20] S. Kumar, N. Das, and S. Islam, "High performance communication redundancy in a digital substation based on IEC 62439-3 with a station bus configuration," in *2015 Australasian Universities Power Engineering Conference (AUPEC)*. IEEE, sep 2015, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/7324838/

[21] H. Weibel, "Tutorial on parallel redundancy protocol (prp)," 2010. [Online]. Available: https://www.zhaw.ch/storage/engineering/institute-zentren/ines/forschung-und-entwicklung/time-synchronisation/tutorial-on-prp.pdf

[22] *IEC 61850-1 ed2.0 Communication Networks and Systems for Power Utility Automation - Part 1: Introduction and Overview*, IEC Std., 2013.

[23] V. Skoko, B. Atlagic, and N. Isakov, "Comparative realization of IEC 60870-5 industrial protocol standards," in *2014 22nd Telecommunications Forum Telfor*, vol. 7. IEEE, nov 2014, pp. 987–990. [Online]. Available: http://ieeexplore.ieee.org/document/7034572/

[24] Z. Lin and S. Pearson, "An inside look at industrial Ethernet communication protocols," Texas Instruments, Tech. Rep., 2017. [Online]. Available: http://www.ti.com/lit/wp/spry254/spry254.pdf

[25] D. G. Dunn, M. A. Hildreth, and A. Pilcher, "A modern integrated approach to plant safety, reliability, and production," *IEEE INDUSTRY APPLICATIONS MAGAZINE*, pp. 13–20, mar 2005.

[26] N. K. Verma, T. Sharma, S. Maurya, D. J. Singh, and A. Salour, "Real-time monitoring of machines using Open Platform Communication," in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, jun 2017, pp. 124–129. [Online]. Available: http://ieeexplore.ieee.org/document/7998316/

[27] U. Enste and W. Mahnke, "OPC Unified Architecture," *at - Automatisierungstechnik*, vol. 59, no. 7, pp. 397–404, 2011. [Online]. Available: http://www.degruyter.com/doi/10.1524/auto.2011.0934

[28] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent Manufacturing in the Context of Industry 4.0: A Review," *Engineering*, vol. 3, no. 5, pp. 616–630, oct 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S2095809917307130

[29] R. Buyya, R. N. Calheiros, and A. V. Dastjerdi, *Big Data: Principles and Paradigms*, 2016.

[30] J. Brittain, M. Cendon, J. Nizzi, J. Pleis, J. . Brittain, M. . Cendon, J. . Nizzi, and M. Llamas-Cendon, "Data Scientist's Analysis Toolbox: Comparison of Python, R, and SAS Performance," *SMU Data Science Review*, vol. 1, no. 2, p. 7, 2018. [Online]. Available: https://scholar.smu.edu/datasciencereview/vol1/iss2/7

[31] R. Langner, "Stuxnet: Dissecting a Cyberwarfare Weapon," *IEEE Security & Privacy Magazine*, vol. 9, no. 3, pp. 49–51, may 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5772960/

[32] T. M. Chen and S. Abu-Nimeh, "Lessons from Stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, apr 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5742014/

[33] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Nai Fovino, and A. Trombetta, "A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 179–186, may 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5682374/

[34] F. Cleveland, "IEC TC57 WG15: IEC 62351 Security Standards for the Power System Information Infrastructure," International Electrotechnical Commission, White Paper ver 14, 2012. [Online]. Available: http://xanthus-consulting.com/Publications/documents/IEC_TC57_WG15_White_Paper.pdf

[35] C. Onal and H. Kirrmann, "Security improvements for IEEE 1588 Annex K: Implementation and comparison of authentication codes," in *2012 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication Proceedings*. IEEE, sep 2012, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/6336632/

[36] A. He, "Design and Implementation of Smart Sensors with Capabilities of Process Fault Detection and Variable Prediction," Ph.D. dissertation, The University of Western Ontario, 2017.

[37] M. Perlmutter and S. Breit, "The future of the MEMS inertial sensor performance, design and manufacturing," in *2016 DGON Intertial Sensors and Systems (ISS)*. IEEE, sep 2016, pp. 1–12. [Online]. Available: http://ieeexplore.ieee.org/document/7745671/

[38] M. Chaudhari and S. Dharavath, "Study of Smart Sensors and their Applications," *International Journal of Advanced Research in computer and communication engineering*, vol. 3, no. 1, pp. 5031–5034, 2014.

[39] V. Ratford, "Accelerating Machine Learning : Implementing Deep Neural Networks on FPGAs," 2016. [Online]. Available: https://www.embedded-vision.com/platinum-members/xilinx/embedded-vision-training/documents/pages/accelerating-machine-learning-imple

[40] Intel, "FPGAS for Data Analytics," 2019. [Online]. Available: https://www.intel.com/content/www/us/en/storage/products/programmable/applications/data-analytics.html

[41] D. Eaton, "Turning big data challenges into opportunities with FPGA-accelerated computing," 2018. [Online]. Available: https://www.datacenterdynamics.com/opinions/turning-big-data-challenges-opportunities-fpga-accelerated-computing/

[42] K. Neshatpour, "Accelerating Big Data Analytics Using FPGAs," 2015.

[43] A. Shawahna, S. M. Sait, and A. El-Maleh, "FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review," *IEEE Access*, vol. 7, pp. 7823–7859, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8594633/

[44] SoC-e, "SMARTzynq module: 5 Port Gigabit Ethernet Industrial Embedded Switch Module," http://soc-e.com/products/smart-zynq-module.

[45] Xilinx Corp., "Zynq-7000 All Programmable SoC," http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/index.htm, 2012.

[46] K. Henning, W. Wolfgang, and H. Johannes, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0," Tech. Rep., 2013. [Online]. Available: http://www.acatech.de/fileadmin/../Final{_}report{_}{_}Industrie{_}4.0{_}accessible.pdf.

[47] L. F. Herrera-Quintero, K. Banse, J. Vega-Alfonso, and A. Venegas-Sanchez, "Smart ITS sensor for the transportation planning using the IoT and Bigdata approaches to produce ITS cloud services," in *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*. IEEE, apr 2016, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/7520096/

[48] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, feb 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6740844/

[49] M. Jaradat, M. Jarrah, A. Bousselham, Y. Jararweh, and M. Al-Ayyoub, "The Internet of Energy: Smart Sensor Networks and Big Data Management for Smart Grid," *Procedia Computer Science*, vol. 56, no. 1, pp. 592–597, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1877050915017317

[50] M. S. Mekala and P. Viswanathan, "A Survey: Smart agriculture IoT with cloud computing," in *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*. IEEE, aug 2017, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/8211551/

[51] M. Hubl and et al., "Embedding of wearable electronics into smart sensor insole," in *2016 IEEE 18th Electronics Packaging Technology Conference (EPTC)*. IEEE, nov 2016, pp. 597–601. [Online]. Available: http://ieeexplore.ieee.org/document/7861550/

[52] G. Tripathi, B. Sharma, and S. Rajvanshi, "A combination of Internet of Things (IoT) and graph database for future battlefield systems," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, may 2017, pp. 1252–1257. [Online]. Available: http://ieeexplore.ieee.org/document/8230010/

[53] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing Smart Factory of Industrie 4.0: An Outlook," *International Journal of Distributed Sensor Networks*, vol. 12, no. 1, p. 3159805, jan 2016. [Online]. Available: http://journals.sagepub.com/doi/10.1155/2016/3159805

[54] B. Buntz, "The top 20 industrial IoT applications," sep 2017. [Online]. Available: http://www.ioti.com/industrial-iot-iiot/top-20-industrial-iot-applications

[55] S. Nuratch, "The iiot devices to cloud gateway design and implementation based on microcontroller for real-time monitoring and control in automation systems," in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, June 2017, pp. 919–923.

[56] "Dell Edge gateway 5000 series," 2018. [Online]. Available: https://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/Spec_Sheet_Dell_Edge_Gateway_5000_Series.pdf

[57] "Nexcom, "Products"," 2018. [Online]. Available: http://www.nexcom.com/Products/industrial-computing-solutions/iot-solutions/iot-gateway/iiot-edge-gateway-cps-200

[58] Q. Chi, H. Yan, C. Zhang, Z. Pang, and L. D. Xu, "A reconfigurable smart sensor interface for industrial wsn in iot environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1417–1425, May 2014.

[59] C.-H. Chen, M.-Y. Lin, and X.-C. Guo, "High-level modeling and synthesis of smart sensor networks for industrial internet of things," *Computers & Electrical Engineering*, vol. 61, pp. 48 – 66, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790616304001

[60] A. Puhm, P. Roessler, M. Wimmer, R. Swierczek, and P. Balog, "Development of a flexible gateway platform for automotive networks," in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, Sept 2008, pp. 456–459.

[61] T. Gomes, S. Pinto, T. Gomes, A. Tavares, and J. Cabral, "Towards an fpga-based edge device for the internet of things," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, Sept 2015, pp. 1–4.

[62] "Overcoming Smart Grid Equipment Design Challenges with FPGAs," 2018. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01191-smart-grid-design.pdf

[63] "MinimalModbus 0.6: Easy-to-use Modbus RTU and Modbus ASCII implementation for Python," 2014. [Online]. Available: https://pypi.python.org/pypi/MinimalModbus/0.6

[64] N. Moreira, J. Lázaro, U. Bidarte, J. Jimenez, and A. Astarloa, "On the utilization of system-on-chip platforms to achieve nanosecond synchronization accuracies in substation automation systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1932–1942, July 2017.

Marcelo Urbina received the B. Sc. degree in electronics and telecommunications engineering from the Universidad de las Fuerzas Armadas ESPE, Ecuador, in 2007, received the Magister. degree in Communication networks from the Pontificia Universidad Católica del Ecuador, in 2014, received the M.Sc. degree in advanced electronic systems from the University of the Basque Country, Spain, in 2016, where he is currently pursuing the Ph.D. degree in Electronics and Telecommunications. In 2008, he joined the Universidad de las Fuerzas Armadas ESPE as a Researcher and a Lecturer.

Tatiana Acosta received the B. Sc. degree in Electronics in Automation and Control engineering from the Universidad de las Fuerzas Armadas ESPE, Ecuador, in 2002, received the M.Sc. degree in advanced electronic systems from the University of the Basque Country, Spain, in 2016, where he is currently pursuing the Ph.D. degree in Electronics and Telecommunications. In 2002, In 2008, she joined the Universidad de las Fuerzas Armadas ESPE as a Researcher and a Lecturer.

Jesús Lázaro received the M.Sc. and Ph.D. degrees in telecommunications engineering from the University of the Basque Country, Spain, in 2001 and 2005, respectively. In 2001, he joined the University of the Basque Country as a Researcher and a Lecturer. In 2010, he was a Research Visitor with the Configurable Computing Laboratory, Virginia Tech. He is Member of the Applied Electronics Research Team, and the Co-Founder and an entrepreneur of System-on-Chip Engineering S.L. He has participated in over 30 research projects supported by public institutions and private companies, in six of them as a Main Researcher.

Armando Astarloa received the M.Sc. and Ph.D. degrees in electrical engineering from the University of the Basque Country, Spain, in 1999 and 2005, respectively. From 1999 to 2001, he was a Research and Development Engineer with an electronics company. In 2001, he joined the Telecommunications Department, University of the Basque Country, as a Researcher and a Lecturer. In 2008, he was a Research Visitor with the Institute of Microelectronics and Wireless Systems, National University of Ireland. He is Member of the Applied Electronics Research Team, and the Co-Founder and an entrepreneur of System-on-Chip Engineering S.L.

Unai Bidarte received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of the Basque Country, Spain, in 1996 and 2004, respectively. Since 1999, he has been a Professor of Electronic Technology with the University of the Basque Country. He is a Researcher on the Applied Electronics Research Team. He has participated in over 35 research projects.