Experimental Modeling and Measurements of Networked Wireless Sensors for Power Consumption

Jin-Shyan Lee
Department of Electrical Engineering
National Taipei University of Technology
Taipei, Taiwan
e-mail: jslee@mail.ntut.edu.tw

Jorge Portilla, Gabriel Mujica
Centro de Electronica Industrial
Universidad Politecnica de Madrid
Madrid, Spain
email: jorge.portilla@upm.es
email: gabriel.mujica@upm.es

Yuan-Heng Sun
Information and Communications Labs
Industrial Technology Research Institute
Hsinchu, Taiwan
e-mail: gilbertsun@itri.org.tw

Abstract— **Power efficiency is a critical design issue in wireless sensor networks. In order to analyze the power consumption of a single node, a system model of networked wireless sensors is thus required. Based on a Petri net framework, this paper has preliminarily applied a systematic approach to the modeling and measurement of power consumption for ZigBee-equipped sensors. Moreover, several experiments have been conducted to measure the real power consumption of wireless sensor networks. It is believed that the experimental measurements presented in this paper would benefit application engineers in analyzing and understanding the power consumption of wireless sensor networks.**

Keywords-energy consumption; experimental measurements; Petri nets; sensor models; wireless sensor networks; ZigBee.

I. Introduction

An earlier version of this paper was presented at the International Conference on Systems and Networks Communications (ICSNC) and was published in its proceedings [1]. This paper extends the previous work by conducting more experiments regarding the power consumption of wireless sensor networks.

Recently, there has been an increasing emphasis on developing distributed Wireless Sensor Networks (WSNs) with self-organization capabilities to cope with device failures, changing environmental conditions, and different sensing and measurement applications [2]-[5]. WSNs consist of hundreds or even thousands of networked wireless sensors, which are linked by radio frequencies to perform distributed sensing tasks. In general, since these wireless sensors are equipped with batteries, energy consumption is a major design issue. Researchers have attempted to determine the best topology, the optimal way of routing, or whether the sensor node should aggregate data or not. All these topics are investigated with the intention of prolonging network lifetime from a global networking point of view [6]-[8].

On the other hand, from a single node point of view, energy conservation could be achieved by applying some power management techniques. However, in order to propose methods, by which power consumption can be minimized in networked wireless sensors, it is first necessary to gain an accurate understanding of their energy consumption characteristics. Thus, a system model of wireless sensors is required so as to analyze the energy consumption of a single node.

Starting from measurements carried out on the off-the-shelf radio, Bougard *et al.* [9] evaluated the potential of an IEEE 802.15.4 radio for use in an ultra-low power sensor node operating in a dense network. Their resulting model has been used to optimize the parameters of both the physical and medium access control layers in a dense sensor network scenario. Also, based on the empirical energy consumption measurements of Bluetooth modules, Ekstrom *et al.* [10] presented a realistic model of the radio energy consumption for Bluetooth-equipped sensor nodes used in a low-duty-cycle network. Their model gives users the possibility to optimize their radio communication with respect to energy consumption while sustaining the data rate. From a hybrid system point of view, Sousa *et al.* [11] modeled and analyzed the power consumption of a wireless sensor node in sensor networks using differential hybrid Petri Nets (PNs). With the discrete event evolution, the continuous battery discharge profile is updated and the remaining battery capacity is estimated. Moreover, their Petri net model was further applied to the design and evaluation of several dynamic power management solutions [12]. Based on Petri nets, Shareef and Zhu [13] also developed a model of a wireless sensor node that can accurately estimate the energy consumption. They used this model to identify an optimal threshold for powering down a sensor node of a specific wireless sensor application.

Most of the previous work focused on developing a conceptual sensor model and provided limited results on realistic measurement or comparative experiments. By applying our previously proposed Petri net framework in [14], this work has preliminarily modeled the energy consumption of a ZigBee-equipped sensor node. Then, a basic experiment has been conducted to measure the real power consumption and provide input parameters to the PN model, which could be applied to further simulations of ZigBee-based WSNs.

This is the sense to use the PN model to describe the power consumption of sensor nodes. Furthermore, by using the developed modular Cookie platforms [15]-[16], more experiments regarding measurements of power consumption for wireless sensor networks have conducted to analyze different power profiles to reach ultra-low-power states. It is believed that the experimental measurements presented in this paper would benefit application engineers in analyzing and understanding the power consumption of wireless sensor networks.

The rest of this paper is organized as follows. Section II introduces the Petri net modeling of wireless sensors. Next, preliminary experiments are provided in Section III. Then, Section IV gives more experimental measurements of power consumption for wireless sensor networks. Finally, Section V concludes this paper.

II. Petri Net Modeling of Wireless Sensors

This section will introduce the basic PN concepts, the typical PN modeling, the MultiParadigm Modeling (MPaM) methodology, and then illustrate the behavior modeling of networked wireless sensors.

A. *Basic PN Concepts* [5]

A PN is identified as a particular kind of bipartite directed graph populated by three types of objects. They are places, transitions, and directed arcs connecting places and transitions. Formally, a PN can be defined as

$$G = (P, T, I, O) \tag{1}$$

where,

$P = \{p_1, p_2, ..., p_m\}$ is a finite set of places, where $m > 0$;

$T = \{t_1, t_2, ..., t_n\}$ is a finite set of transitions with $P \cup T \neq \varnothing$ and, where $n > 0$;

$I : P \times T \to N$ is an input function that defines a set of directed arcs from *P* to *T*, where $N = \{0, 1, 2, \dots\}$;

$O : T \times P \to N$ is an output function that defines a set of directed arcs from *T* to *P*;

A marked PN is denoted as (G, M_0), where M_0: $P \to N$ is the initial marking. A transition *t* is enabled if each input place *p* of *t* contains at least the number of tokens equal to the weight of the directed arc connecting *p* to *t*. When an enabled transition fires, it removes the tokens from its input places and deposits them on its output places. PN models are suitable to represent the systems that exhibit concurrency, conflict, and synchronization.

Several important PN properties include boundness, which means no capacity overflow, liveness, which shows the freedom from deadlock, conservativeness, which indicates the conservation of non-consumable resources, and reversibility, which represents the cyclic behavior.

The concept of liveness is closely related to the complete absence of deadlocks. A PN is said to be live if, no matter what marking has been reached from the initial marking, it is possible to ultimately fire any transition of the net by progressing through some further firing sequences. This means that a live PN guarantees deadlock-free operation, no matter what firing sequence is chosen. Validation methods of these properties include reachability analysis, invariant analysis, reduction method, siphons/traps-based approach, and simulation [17].

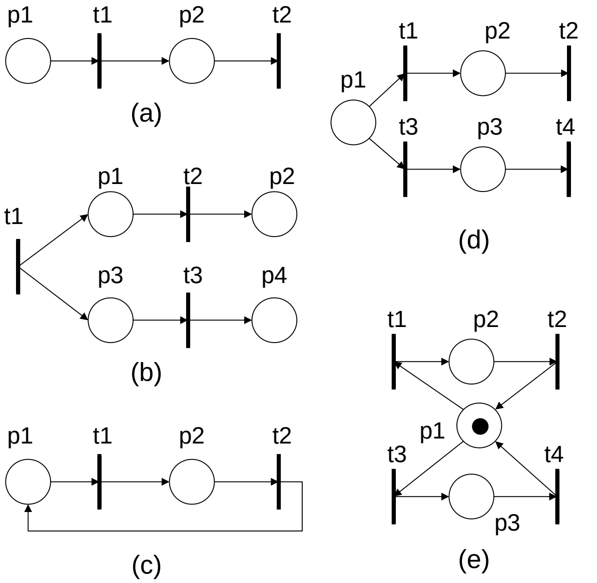


Figure 1. Basic PN models for (a) sequential, (b) concurrent, (c) cyclic, (d) conflicting, and (e) mutually exclusive relations [18].

B. *Typical PN Modeling* [18]

At the modeling stage, one needs to focus on the major operations and their sequential or precedent, concurrent, or conflicting relationships. The basic relations among these processes or operations can be classified as follows.

- *Sequential:* As shown in Figure 1 (a), if one operation follows the other, then the places and transitions representing them should form a cascade or sequential relation in PNs.
- *Concurrent:* If two or more operations are initiated by an event, they form a parallel structure starting with a transition, i.e., two or more places are the outputs of the same transition. An example is shown in Figure 1 (b). The pipeline concurrent operations can be represented with a sequentially-connected series of places/transitions, in which multiple places can be marked simultaneously or multiple transitions are enabled at certain markings.
- *Cyclic:* As shown in Figure 1 (c), if a sequence of operations follow one after another and the

completion of the last one initiates the first one, then a cyclic structure is formed among these operations.

- *Conflicting:* As shown in Figure 1 (d), if either of two or more operations can follow an operation, then two or more transitions create the outputs from the same place.
- *Mutually Exclusive:* As shown in Figure 1 (e), two processes are mutually exclusive if they cannot be performed at the same time due to constraints on the use of shared resources. A structure to realize this is through a joint place marked with one token plus multiple output and input arcs to activate these processes.

C. *MultiParadigm Modeling (MPaM)* [14]

To deal with specific and complicated problems, we have to integrate heterogeneous modeling arts, thereby resulting in the MPaM methodology. It is based on a proposition of giving different entities of a complex system the most appropriate modeling abstractions [14]. From a viewpoint of MPaM, the PN is adopted to design and analyze coordination controllers in a discrete-event domain. The primary motivation for employing PN as hybrid models is the situation that all those good characteristics that make discrete PN a valuable discrete-event model still be available to hybrid systems. Examples of these characteristics include: PN does not need the exhaustive enumeration of the state space at the design stage and can finitely model systems with an infinite state space. Moreover, PN provides a modular description where the structure of each module is maintained in the composed model. Furthermore, discrete states of PN are modeled by a vector and not by a symbolic label, thus linear algebraic techniques may be adopted for system analysis.

Figure 2 represents the previously proposed PN framework for modeling a system in discrete-event and discrete-time domains [14]. Each operation is modeled with a *command* transition to start the operation, a progressive *working* place, a *response* transition to end the operation, and a *completed* place. Note that the start transition (drawn with a dark symbol) is a controllable event as “command” input, while the end transition is an uncontrollable event as “response” output. The working place is a Hierarchical Hybrid Place (HHP, drawn with a triple circle), in which the state equations of the systems to be controlled are contained and interacted through the boundary interface. The interaction between event-driven and time-driven domains is realized in the following way: a token put into the working place triggers a discrete (or continuous) time process with the corresponding equations. Thresholds are monitored concurrently. Each threshold is corresponding to a transition, that is, the response transitions. When the threshold is reached or crossed, it indicates that the associated event is happening, and the corresponding transition is fired. Next, a new marking is evaluated, and the combination of the hybrid system restarts.

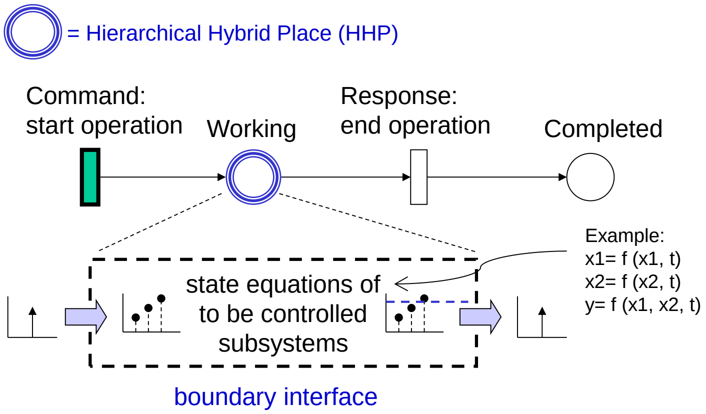


Figure 2. Multiparadigm modeling within a Petri net framework [14].

D. *Behavior Modeling of Networked Wireless Sensors*

In general, radio communication is the most energy consuming part of a wireless sensor as compared with its sensing and computation tasks. Hence, our model focuses on the operations of packet transmission and reception. By applying the design procedure in [14], the PN model of a networked wireless sensor is constructed as shown in Figure 3, which consists of 17 places and 14 transitions, respectively. The corresponding notations are described in Table I. The model is based on a scenario where a sensor node periodically transmits and receives some data towards, for example, a base station.

TABLE I
NOTATION FOR PETRI NET OF A WIRELESS SENSOR IN FIGURE 3

Place	Description	Transition	Description
p1	Node in sleep mode	t1	Cmd: start startup sequence
p2	MCU running at 16MHz	t2	Re: end startup sequence
p3	Startup sequence completed	t3	Cmd: start running MCU at 32MHz
p4	MCU running at 32MHz	t4	Re: end running MCU
p5	MCU running completed	t5	Cmd: start CSMA/CA operation
p6	Radio in RX mode	t6	Re: end CSMA/CA operation
p7	CSMA/CA operation completed	t7	Cmd: start transmitting packets
p8	Radio in TX mode	t8	Re: end transmitting packets
p9	Packet transmission completed	t9	Cmd: start receiving packets
p10	Radio in RX mode	t10	Re: end receiving packets
p11	Packet reception completed	t11	Cmd: start processing packets
p12	Processing packets	t12	Re: end processing packets
p13	Processing packets completed	t13	Cmd: start shutdown sequence
p14	MCU running at 16MHz	t14	Re: end shutdown sequence
p15	Shutdown sequence completed		
p16	MCU is available		
p17	Radio is available		

III. PRELIMINARY EXPERIMENTS

This section will firstly show the measurement setup and experimental results. Then, the comparisons between the measurement and PN model will be described.

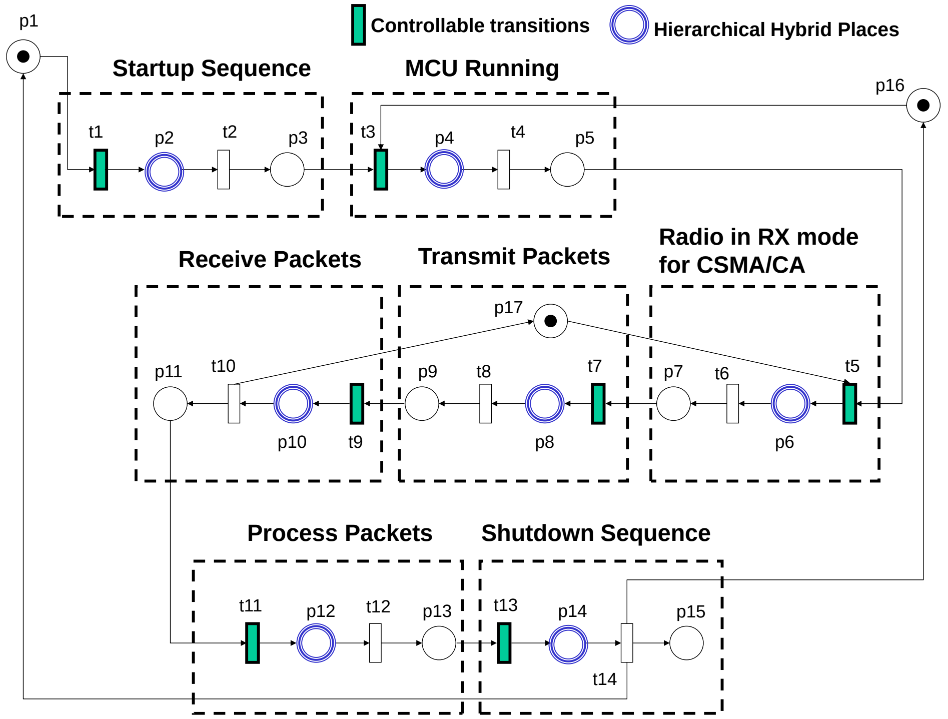


Figure 3. Petri net model of a networked wireless sensor.

A. *Measurement Setup*

In this section, the energy consumption of a wireless sensor as computed via its Petri net model will be compared against real measurements collected from a ZigBee-equipped sensor node. The measurement setup in [19] has been adopted as shown in Figure 4, in which a ZigBee End Device is the Device Under Test (DUT) and powered by a power supply. The energy consumption measurements are performed at the End Device, which periodically (every 0.5 sec in our measurement) wakes up and sends data to the coordinator (base station). The voltage across a 10 Ohm resistor is monitored to determine the current draw of the system. The measurement system has been calibrated with both a digital oscilloscope and a digital multimeter to ensure an accurate measurement. Figure 5 shows the hardware setup during energy consumption measurement.

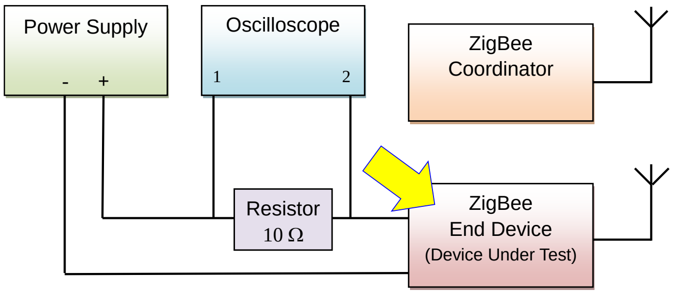


Figure 4. Measurement configuration.

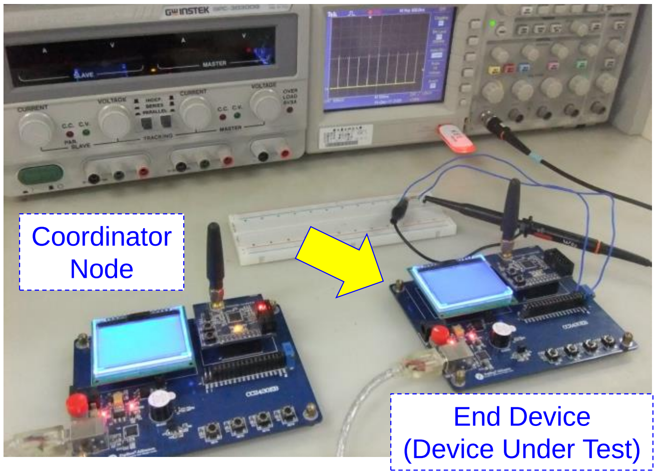
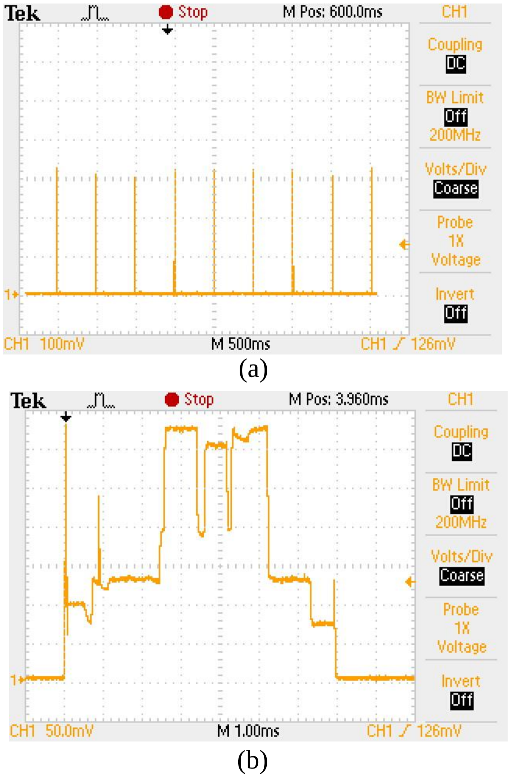


Figure 5. Hardware setup during energy consumption measurement.

B. *Measurement Results*

Figure 6 (a) shows the power consumption during sleep and awake states. The time base on the oscilloscope is set to 500 ms per division, and it can be seen that it is about 0.5 sec among each current peak, showing the power consumption when the device is awake to send the data to the coordinator. Figure 6 (b) is a zoomed version of Figure 6 (a) and shows the current consumption during the active modes in more details. This snapshot has a time base of 1 ms per division. The duration of the active mode is about 7 ms. According to the measurement results, the consumed energy and duration of each operation can be estimated.



(a)

(b)

Figure 6. Measurement results for the division scale at (a) 500 ms and (b) 1 ms.

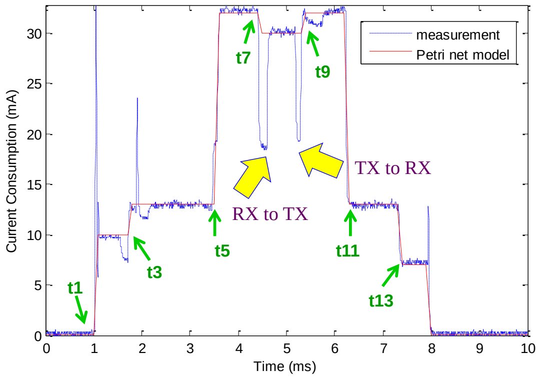


Figure 7. Comparison of energy consumption between measurement and Petri net model.

C. Discussions

With the measured sets of consumed current and duration for each transition as the inputs to the Petri net model, the energy consumption can be obtained as shown in Figure 7. In general, the energy consumption of the Petri net model is close to the practical measurement with a mean difference of less than 1%. However, several peak currents appear during the state transitions, especially the startup sequence t1. The current peaks show the energy consumption when the sensor node is triggered for data transmission. Moreover, note that between transitions t7 and t9, there are two V-shaped gullies, which present the energy consumption of the transceiver turnaround operations, which are the RX to TX and the TX to RX, respectively. Future work would attempt to model such detailed behaviors.

Obviously, the description of the power consumption of a single node during a standard RX/TX procedure is a very isolated scenario. For example, the power consumption of a node would significantly change when a collision is happening during transmission with a subsequent packet loss, which requires a repeated transmission. Further work would consider more practical interactions between the nodes so as to simulate the power consumption of a whole sensor network for different scenarios.

IV. More Experiments Regarding Power Consumption of Wireless Sensor Networks

This section will show more experiments regarding the measurement of power consumption of wireless sensor networks. In this case, the main target is to analyze the different power profiles of a modular WSN hardware platform, which have been designed to reach ultra-low-power states.

A. Developed Cookie Platforms

The implementation of a flexible and configurable processing layer is then compared to an already existing processing solution to show the main benefits of creating a more complex power management profile within the sensor node. The baseline structure of the proposed setup lies on the Cookie platform [15]-[16], which is a modular and fully adaptable hardware architecture mainly composed of four layers: The processing layer, which is the core of the node and includes the microcontrollers of processors to carry out the computation tasks in accordance with the application requirements; the communication layer, which integrates the wireless module to establish an energy-efficient connection with the rest of the participant devices and the base station in the WSN; the sensor layer, which provides the interface with the target environment by including the sensing capabilities to monitor the target physical magnitudes (if needed); and finally the power supply layer, which is in charge of providing the rest of the layers with the required voltage level based on their intrinsic design and operational stages, also serving as an intelligent power manager for more advanced Cookie configurations.

As shown in Figure 8, the vertical connectors that support the modularity of the Cookie allow a very flexible and adaptable prototyping hardware ecosystem to speed up the development and integration of heterogeneous technologies within the same platform. Thus, it fosters the reusability of hardware components with the inclusion of novel techniques for WSN applications.

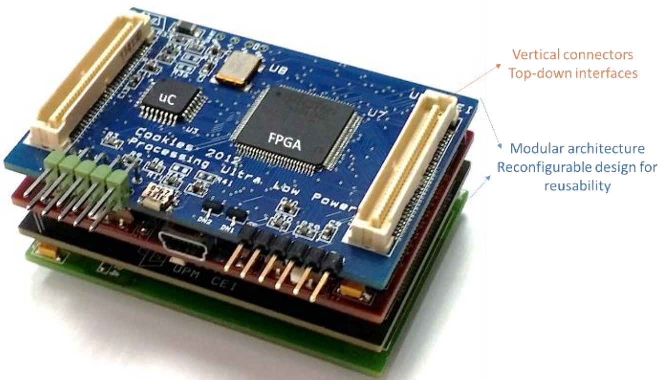

Vertical connectors
Top-down interfaces
Modular architecture
Reconfigurable design for reusability
uC
FPGA
Cookies Processing 2012 Ultra Low Power

Figure 8. Cookie WSN platform composed of 4 modular layers, including an ultra-low-power processing design for very-long-term networks lifetime.

In line with the modular hardware architecture, the Cookie nodes also provide a software support platform to abstract the low-level control of the hardware elements that integrates the sensor devices, as well as speeding up the prototyping of WSN applications and network deployment [20]. The baseline structure of the software layer is composed of a complete set of libraries and functional components that follows the modularity of the hardware layers, including controllers for every processing, sensing, and communication technologies integrated into the Cookie node. These support libraries have been ported to the new design of the ultra-low power processing layer in order to produce a seamless integration of the available hardware elements with the WSN application profiles of the Cookie platform.

The design of this Cookie layer considers three main aspects to exploit the modularity, reconfigurability, and adaptability to different application requirements. The first one relies on the ability of the embedded system to system to adapt to different experimental configurations by allowing for several connectivity arrays of the processing elements to the rest of the platform. This means that, depending on the system requirements, the processing layer can be modified to offer different functional properties. Figure 9 shows the main interconnection structure that allows the definition of several Cookie processing configurations within the same hardware layer.

The second aspect is related to the inclusion of an 8051-based processing architecture, which is in line with the defined HW-SW framework of the Cookies for resource-constrained sensor network applications, and for the efficient implementation of lightweight reprogramming strategies for extending the overall network lifetime [21]. The third aspect lies integrating ultra-low-power hardware elements within the system architecture, seeking the balance between flexibility and power-awareness.

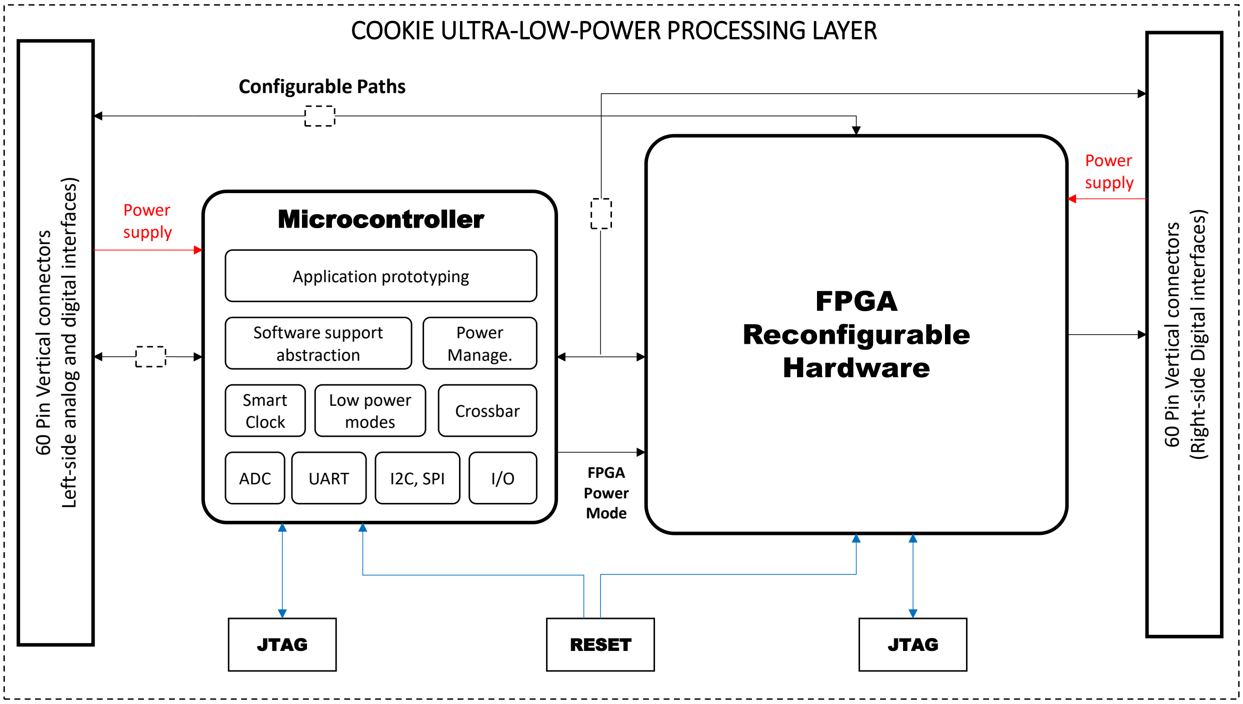

COOKIE ULTRA-LOW-POWER PROCESSING LAYER
Configurable Paths
60 Pin Vertical connectors
Left-side analog and digital interfaces)
Power supply
Microcontroller
Application prototyping
Software support abstraction
Power Manage.
Smart Clock
Low power modes
Crossbar
ADC
UART
I2C, SPI
I/O
FPGA Power Mode
FPGA Reconfigurable Hardware
Power supply
60 Pin Vertical connectors
(Right-side Digital interfaces)
JTAG
RESET
JTAG

Figure 9. Configurable structure of the ultra-low-power Cookie processing layer considering flexibility and adaptability to different resource-constrained application contexts.

Based on the interconnection structure shown in Figure 9, the designed processing layer provides five different operational modes, whose configurations can be made by using bridge connections among layer paths according to the WSN based system constraints definition for the target application/experiment, as follows.

- *Microcontroller with ADC signal connections*: this mode only includes the main processing element and it is thought to be used in those applications where the power consumption is the most critical aspect to be considered. This mode dedicates some of the port lines of the microcontroller for analog input signal processing from the left side of the Cookie interface (for instance, considering the analog sensor from the sensing layer).
- *Microcontroller without ADC signal connections*: this is a variation of the former mode intended to provide the maximum number of digital input/output ports to the rest of the platform.
- *FPGA as the core processor:* in this case, a trade-off between fast signal processing, digital connections availability and power consumption is sought. Thus, the I/O pins of the FPGA are spread between both vertical interfaces of the Cookie platform.
- *Microcontroller with ADC signal connections and FPGA*: this mode offers the widest possibility for platform experimentation in a single layer configuration array, so as to be able to combine the functional component capabilities and peripherals of the microcontroller with hardware block implementations for co-processing and debugging tasks in the FPGA. In this case, both analog and digital sensor signals are contemplated.
- *Microcontroller and FPGA*: It offers a similar approach as the aforementioned mode though more ports of the microcontrollers are dedicated to digital interfaces of the Cookie node.

B. *Measurement Setup*

As mention before, the experimental analysis is firstly focused on a very configurable processing layer that provides an extended set of different power modes.

Figure 10 shows the experimental setup to evaluate the power consumption of the processing layer based on the configuration of the different power modes depicted in Table II. Thus, several functional scenarios have been set up in order to characterize different power consumption profiles of the designed low-power processing layer, considering not only the various configurations that the hardware layer can adopt, but also the operational modes of the microcontroller and the FPGA elements. To do so, the experimental setup was composed of the implemented Cookie layer plus an extension board that allows supplying the different voltage levels of the modular platform from an external source (as shown in Figure 9), so that no additional consumption offsets can significantly shift the encountered measurements.

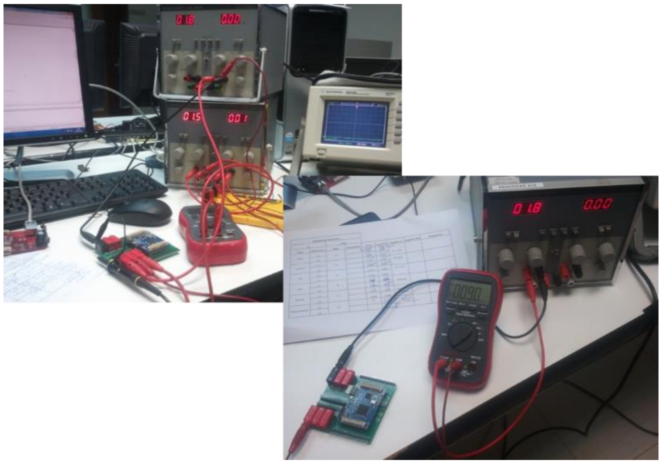


Figure 10. Experimental setup for the characterization of the WSN ultra-low sensor node design.

On one hand, the C8051F930, which is an 8051-based 8-bit microcontroller from Silicon Labs with Crossbar technology [22] is the main processing element that includes the software support platform for managing the Cookie sensor nodes. The C8051F930 is composed of a 64 kB flash program memory, internal and external RAM memories of 256 bytes and 4 kB respectively, 10-bits ADC, four 16-bits timers and Smart Real Time Clock, UART and 2 SPI, and most particularly it supports a voltage supply range of 1.8 V to 3.6 V.

On the other hand, a flash-based FPGA Igloo AGL030V5 from Actel (Microsemi) [23] that serves as a co-processing element for performing faster control tasks that need to be implemented in hardware. The advantage of this technology is the ability to power-down the whole FPGA without losing the hardware implementation (so no need to download the bitstream once the system is power-up again), thus allowing deep power consumption modes without penalizing performance. The AGL030V5 is a 1.2 V to 1.5 V low power flash technology programmable logic device composed of 30000 system gates, 768 D-flip-flops, 81 user I/O pins, and Flash*Freeze sleep mode control without needing to disconnect supply voltages.

The combination of both technologies allows promoting a wider spectrum of configuration, stand-by and functional scenarios that will ultimately have a positive impact in the node lifetime, hence extending the overall long-term operability of the network.

TABLE II
POWER CONSUMPTION RESULTS COMPARING THE DIFFERENT uC POWER MODES AGAINST NORMAL OPERATION, CONSIDERING DIFFERENT CONFIGURATIONS FOR THE MICROCONTROLLER AND THE INPUT VOLTAGE SUPPLY LEVEL

uC		Current consumption		uC configuration	
Operational mode	Voltage Supply (V2)	I_{DOWN} (mA)	I_{NORMAL} (mA)	uC Code	Details
Sleep	2.5	0.011	2.180	Sleep mode trigger (via smartclock) 846 bytes	Low-power oscilator (20 MHz) / 4 ==>> 5 MHz. Timer 2. SmaRTClock in 40 KHz self-oscillate Mode. Port Match. Comparator 0. II/O port decoder.
	1.8	0.005	2.140		
Suspend	2.5	0.105	2.174	Suspend mode trigger (via smartclock) 847 bytes	Low-power oscilator (20 MHz) / 4 ==>> 5 MHz. Timer 2. SmaRTClock in 40 KHz self-oscillate Mode. Port Match. Comparator 0. I/O port decoder.
	1.8	0.087	1.930		
Idle	2.5	0.520	0.850	Iddle mode trigger (via Timer interruption) 237 bytes	Low-power oscilator (20 MHz) / 8 ==>> 2,5 MHz. Timer 0. I/O port decoder.
	1.8	0.510	0.820		
Stop	2.5	0.500	0.830	Stop mode trigger (via external reset) 208 bytes	Low-power oscilator (20 MHz) / 8 ==>> 2,5 MHz. Timer 0. I/O port decoder.
	1.8	0.500	0.810		
Normal	2.5	1.230	1.230	uC basic test 48 bytes	Low-power oscilator (20 MHz) / 8 ==>> 2,5 MHz. I/O port decoder.
	1.8	0.900	0.900		
Programming	2.5	---	2.74 (Erasing) 4.6 (Writing)	uC basic test 48 bytes	Low-power oscilator (20 MHz) / 8 ==>> 2,5 MHz. I/O port decoder.
	1.8	---	2.58 (Erasing) 4.5 (Writing)		

TABLE III
EXPERIMENTAL RESULTS CONSIDERING THE COMBINATION OF THE SLEEP POWER MODES OF THE FPGA AND THE MICROCONTROLLER

uC		FPGA		I Consumpt. (mA)
Mode	V2 (V)	Mode	V1->VCC (V)	
nc		Normal	1.5	0.094
Normal	1.8	Flash*Freeze	1.5	0.0044 (V1), 0.92 (V2)
Normal	1.8	Normal	1.5	0.154 (V1), 1.05 (V2)

C. *Measurement Results*

Both Tables II and III summarize the power-mode profile of the ultra-low power consumption layer, where a combination of two main cores produces a fine-grained configuration set. Tests were also performed considering, on one hand, the inclusion of both components into the hardware layer (uC + FPGA) and on the other hand, the FPGA as the only processing element of the board.

Table II shows the results regarding the configuration that only includes the C8051F930, so V2 was used to characterize the current consumption profile of the layer in such conditions, taking into account 1.8 and 2.5V as voltage supply levels. The current consumption in active mode depends on the number of activated peripherals of the microcontroller, whereas the power-down transition modes are triggered by using various sources of activations, as detailed in the uC configuration column. For instance, differences in terms of power consumption in active modes can be distinguished with respect to the ADuC841-based layer, where I_{normal} raises up to 11mA in normal operation (with VDD=2.8V, 11.0592MHz), while 20uA in power down mode.

Table III depicts the current consumption considering V1 = 1.5V for VCC and VCCBIx (3.67MHz clock), V2 = 1.8V for the uC, and the flash*freeze activation for power-down mode on the AGL030v5, whose results clearly show the difference in consumption when switching the FPGA to the ultra-low power mode and the normal operation (around 100uA and 5uA, respectively) both in FPGA and uC+FPGA layer configuration.

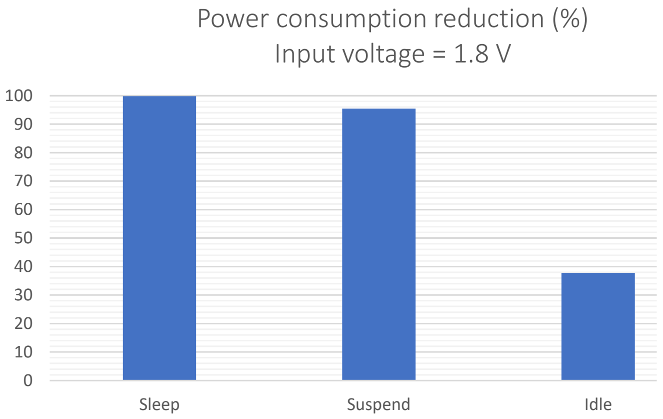


Figure 11. Comparison of the different consumption improvement obtained experimentally for every power mode, and for an input supply voltage of 1.8 V.

Figure 11 depicts the comparative results regarding the percentage of real consumption of the processing element when considering an input supply voltage of 1.8 V and the three main low-power modes: Sleep, Suspend and Idle. The experimental results show that the sleep modes provides a 99.7% of reduction in comparison with the normal operation (and 94.3% with respect to the next configurable mode), which represents an important improvement for very-long-term operability of the WSN, where the sensor node can stay in ultra-low power mode for longer periods of times. On the other hand, 95.4% and 37.8% improvement are respectively obtained for Suspend and Idle modes, so a trade-off between sensor node activity and energy balance are also possible with these intermediate states.

The comparison of both processing layers for resource-constrained applications has experimentally carried out by measuring the power consumption when performing the transition between wireless data transmission and putting the communication module in sleep mode. Figure 12 shows the results comparing the power consumption in both processing layers, and including the same application code into the microcontrollers (wake-up the wireless communication module, periodic sensing data transmission and putting the module back to sleep mode). The experimental outcomes clearly exhibit the improvement of the ultra-low power design, having a current consumption of 3.23 mA when the wireless module is in sleep mode, whereas the other processing layer obtains a value of 24.9 mA.

D. Discussions

The combination of the intrinsic flexibility of the modular hardware platform with the design and implementation of an ultra-low power processing layer provides important benefits in terms of energy savings as well as a trade-off between computing capabilities and long-term lifetime awareness. The reduction of the power consumption in every mode is very noticeable when comparing first with the next configuration step (obtaining more than 90% in the deepest low-power modes, that is, suspend and sleep, and more than 80% between idle and suspend) and second with a more general-purpose (microcontroller + FPGA) design without particular ultra-low power strategies (up to 87 % depending on the configuration to be adopted). The results show that it is indeed possible to refine the power-computing balance both at hardware and software levels without a strong penalization in one of the figures of merits sought. For instance, a very extreme configuration based on the only microcontroller and running with 1.8 V can be applicable in some use cases, although shifting to another configuration point may be possible if considering the experimental outcomes against the desired power consumption boundaries. Therefore, a more computational-effective solution without penalizing the energy cost is feasible.

In this sense, the level of reconfigurability and adaptability can be highlighted in three ways, according to the application needs: Selection of the processing elements to be included and their interfaces in line with the five supported combinations; the power supply levels in accordance with the threshold limits and considering the results of the power consumption experiments for every operational state; and the management of the different low-power modes depending on the duty-cycle and QoS requirements for the target scenario. While for the first two cases the decisions can be taken at design and pre-deployment time (although dynamic voltage scaling might be a possibility to be considered as well), the latter exhibits a very powerful opportunity to reduce power at runtime and in an adaptive fashion.

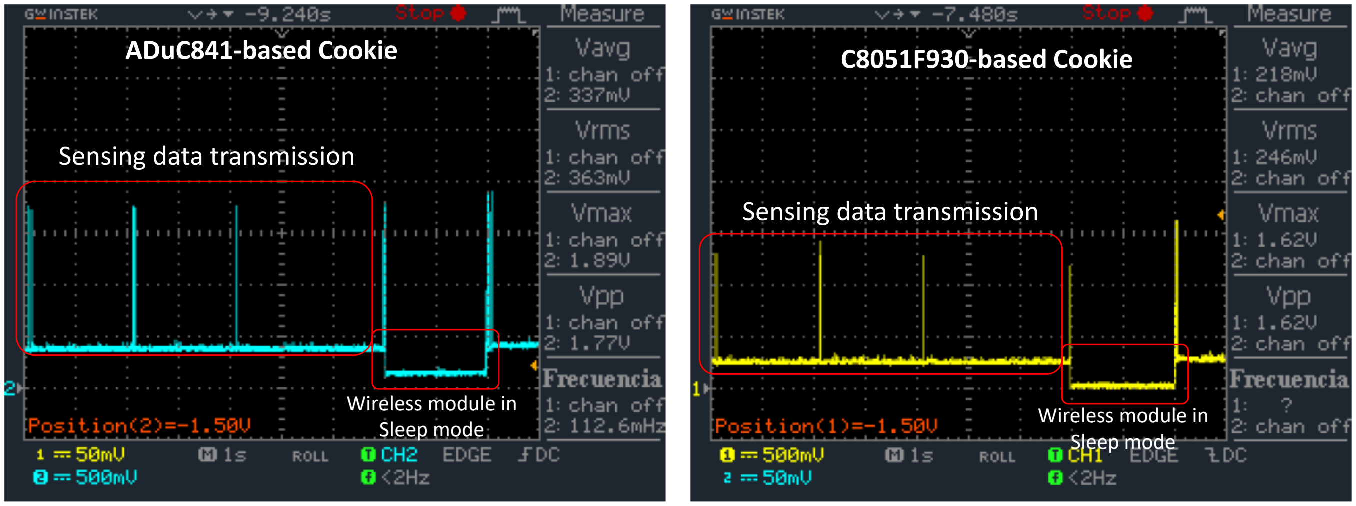


Figure 12. Experimental comparison of two processing layers when performing the same WSN application and power modes transitions.

V. Conclusion and Future Work

In this paper, a systematic approach to modeling and measurement of energy consumption for wireless sensors has been presented. In the prior work, the sensor operation is modeled using the Petri nets. Then, a preliminary experiment has been conducted to measure the real power consumption and provide input parameters to the Petri net model. The comparative results indicate the Petri net model has approximated the real measurement under the assumed scenarios. Besides the periodical operations demonstrated in this paper, the measurement scheme is also useful for other specific applications and could be fed back to the Petri net model as a calibration source. Since the proposed Petri net model in this paper is mainly designed for packet transmission and reception, as future work, operations of sensing and computation tasks could be further considered so as to make the model much more realistic. Also, with a given battery, the proposed model could be further applied to the lifetime estimation for periodical operations.

Furthermore, more experiments regarding measurements of power consumption for wireless sensor networks have conducted to analyze different power profiles to reach ultra-low-power states by using the developed modular Cookie platforms. Also, one direction of future work is the modeling and measurement of a variety of sensors for power consumption under the structure of the Internet of things [24]. It is believed that the experimental measurements presented in this paper would benefit application engineers in analyzing and understanding the power consumption of wireless sensor networks.

Acknowledgment

This paper was supported by the Ministry of Science and Technology (MOST) of Taiwan under grant MOST-106-2221-E-027-057. The authors and specifically Jorge Portilla would like to thank the “José Castillejo” program for junior postdoc research stays, funded by the Ministry of Education, Culture and Sports of the Government of Spain, grant number CAS17/00259. Also, the authors would like to thank Mr. Yu-Kai Wang from Information and Communications Research Labs, Industrial Technology Research Institute (ITRI), Taiwan, for his help with part of the experiments.

References

[1] J. S. Lee and Y. H. Sun, “Behavior modeling of networked wireless sensors for energy consumption using Petri nets,” in *Proc. IARIA International Conference on Systems and Networks Communications (ICSNC)*, Nice, France, October 2018, pp. 64-68.

[2] J. S. Lee and Y. C. Lee, “An application of grey prediction to transmission power control in mobile sensor networks,” *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2154-2162, Jun. 2018.

[3] W. Q. Guo, W. M. Healy, and M. C. Zhou, “Impacts of 2.4-GHz ISM band interference on IEEE 802.15.4 wireless sensor network reliability in buildings,” *IEEE Trans. Instrum. Meas.*, vol. 61, no. 9, pp. 2533-2544, Sep. 2012.

[4] J. S. Lee and W. L. Cheng, “A fuzzy-logic-based clustering approach for wireless sensor networks using energy predication,” *IEEE Sensors J.*, vol. 12, no. 9, pp. 2891-2897, Sep. 2012.

[5] J. S. Lee, “A Petri net design of command filters for semi-autonomous mobile sensor networks,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 4, pp. 1835-1841, Apr. 2008.

[6] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537-568, May 2009.

[7] J. S. Lee and T. Y. Kao, “An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 951-958, Dec. 2016.

[8] J. S. Lee and C. L. Teng, “An enhanced hierarchical clustering approach for mobile sensor networks using fuzzy inference systems,” *IEEE Internet Things J.*, vol. 4, no. 4, pp. 1095-1103, Aug. 2017.

[9] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene, “Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives,” in *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, Munich, Germany, Mar. 2005, pp. 196-201.

[10] M. C. Ekstrom, M. Bergblomma, M. Linden, M. Bjorkman, and M. Ekstrom, “A Bluetooth radio energy consumption model for low-duty-cycle applications,” *IEEE Trans. Instrum. Meas.*, vol. 61, no. 3, pp. 609-617, Mar. 2012.

[11] J. R. B. Sousa, A. M. N. Lima, and A. Perkusich, “Modeling and analyzing power consumption in sensor networks nodes based on differential hybrid Petri nets,” in *Proc. IEEE Conf. Industrial Electronics Society (IECON)*, Raleigh, NC, Nov. 2005, pp. 389-394.

[12] P. S. Sausen, J. R. B. Sousa, M. A. Spohn, A. Perkusich, and A. M. N. Lima, “Dynamic power management with scheduled switching modes,” *Comput. Commun.*, vol. 31, pp. 3625-3637, Sep. 2008.

[13] A. Shareef and Y. F. Zhu, “Power modeling of wireless sensor nodes based on Petri net,” in *Proc. The 39th Int. Conf. Parallel Processing*, San Diego, CA, Sep. 2010, pp. 101-110.

[14] J. S. Lee, M. C. Zhou, and P. L. Hsu, “Multiparadigm modeling for hybrid dynamic systems using a Petri net framework,” *IEEE Trans. Syst. Man Cybern. A., Syst. Humans*, vol. 38, no. 2, pp. 493-498, Mar. 2008.

[15] J. Portilla, T. Riesgo, A. Abril, and A. de Castro, “Rapid prototyping for multi-application sensor networking”, *SPIE Newsroom*, Nov. 2007, DOI:10.1117/2.1200711.0851.

[16] G. Mujica, R. Rodriguez-Zurrunero, M. R. Wilby, J. Portilla, A. B. Rodríguez González, A. Araujo, T. Riesgo, and J. J. Vinagre Díaz, “Edge and fog computing platform for data fusion of complex heterogeneous sensors,” *Sensors*, vol. 18, no. 11, article ID 3630, 2018.

[17] M. C. Zhou, K. Venkatesh, *Modeling Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*, Singapore:World Scientific, 1998.

[18] J. S. Lee and P. L. Hsu, “Implementation of a remote hierarchical supervision system using Petri nets and agent technology,” *IEEE Trans. Syst. Man Cybern. C. Appl. Rev.*, vol. 37, no. 1, pp. 77-85, Jan. 2007.

[19] B. Selvig, “Measuring power consumption with CC2430 and Z-Stack,” *Application Note AN053*, Texas Instruments. Jul. 2007.

[20] G. Mujica, V. Rosello, J. Portilla, and T. Riesgo, “Hardware-software integration platform for a WSN testbed based on cookies nodes”, in *Proc. IEEE Conf. Industrial Electronics Society (IECON)*, Montreal, Canada, Oct. 2012, pp. 6013-6018.

[21] G. Mujica, V. Rosello, J. Portilla, and T. Riesgo, “On-the-fly dynamic reprogramming mechanism for increasing the energy efficiency and supporting multi-experimental capabilities in WSNs,” in *Proc. IEEE Conf. Industrial Electronics Society (IECON)*, Vienna, Austria, Nov. 2013, pp. 5455-5460.

[22] Silicon Labs, *C8051F92x-F93x Ultra-Low-Power MCUs*. [Online] Available: http://www.silabs.com. [Accessed: 10-May-2019].

[23] Microsemi, *IGLOO Low-Power Flash FPGAs Family,* [Online] Available: http://www.microsemi.com/products/fpga-soc/fpga /igloo-overview. [Accessed: 10-May-2019].

[24] J. Portilla, G. Mujica, J. S. Lee, and T. Riesgo, “The extreme edge at the bottom of the Internet of things: A review,” *IEEE Sensors J.*, vol. 19, no. 9, pp. 3179-3190, May 2019.