OpenWuR - An Open WSN Platform for WuR-based Application Prototyping

Felix Cabarcas
Electronics Engineering Department
Pontificia Universidad Javeriana
fcabarcas@javeriana.edu.co

Juan Aranda
Exact Sciences and Engineering School
Universidad Sergio Arboleda
juan.aranda@usa.edu.co

Diego Mendez
Electronics Engineering Department
Pontificia Universidad Javeriana
diego-mendez@javeriana.edu.co

Abstract

Recently, Wake-up Radio (WuR) based systems have gained interest from the academic and industrial institutions, given their ultra-low power consumption during channel listening and highly responsive features, which is required in several Wireless Sensor Network (WSN) applications, such as environmental monitoring. However, few WuR-based platforms have been reported in the literature for rapid prototyping and testing of new WSN techniques without relying only on simulation tools. In this paper, an open-source WSN platform based on the WuR paradigm is presented, with dual-stack support tuned for the 915MHz and 2.4GHz ISM bands, ContikiOS as the operating system, and an on-board power monitoring system for performance evaluation purposes. This power management system differentiates the power consumption between the WuR module and the sensor node, allowing the extrapolation of the results, with respect to the consumption of other WuR prototypes reported in the state-of-the-art. The experimental results validated the flexibility of the proposed system and the interesting potential of such a platform for prototyping purposes.

Keywords

WuR, WSN, hardware, prototyping, OOK.

1 Introduction

Wireless Sensor Networks (WSNs) are considered the keystone of Internet of Things (IoT) based solutions such as those deployed for structural and environmental monitoring applications. Typically, WSNs are built on battery-powered sensor nodes (SNs), therefore, energy-efficiency is a crucial goal to be considered in any WSN design and deployment [5]. Besides, most critical monitoring applications demand from the WSN to be responsive upon the occurrence of an event, i.e., SNs should propagate the event and its associated data with minimal latency via a multi-hop network [8].

Consequently, energy-efficiency and responsiveness are two critical features to consider when designing WSNs. The former has been widely tackled through the implementation of sophisticated power saving techniques based on low-duty-cycling (LDC). However, this approach suffers from power expensive idle-listening and overhearing issues. Besides, LDC applications require a sophisticated synchronization mechanism for the nodes to exchange data packets. Nevertheless, LDC techniques seek to minimize the active period of SNs to lower their energy consumption during their whole operation at the cost of a higher latency. Therefore, a wake-up radio (WuR) approach has recently arisen to counteract the LDC issues.

The WuR approach aims to minimize the latency for energy-efficient WSNs via the implementation of an always-on and ultra-low-power radio system within the sensor node architecture to manage the duty-cycling of the node–which is usually configured in its deepest low power mode–, through an external interrupt signal known as the Wake-up Signal (WuS). When an event occurs, the source node wakes up the next hop in the multi-hop path via a WuS packet and then, it disseminates the event and its associated data to the sink node. This behavior allows WSNs to be responsive and energy-efficient during their whole operation.

Recently, several efforts have been reported in the literature regarding prototypes and networking techniques based on WuR [5]. However, most of them have not been released to the market, and most techniques remain in simulation environments. According to Piyare *et al.* in [5], an unified and open-source system and a network architecture under the WuR approach for WSNs are still missing, where techniques such as routing and MAC protocols can be implemented, without relying on simulation tools, but on an actual testbed. Therefore, in this paper, we present OPENWUR, an open WSN platform based on the WuR paradigm, which aims to support the rapid prototyping and testing phases that should be currently included in any research in the field.

The main contributions of this paper are twofold: (i) an open hardware WuR-based WSN platform is provided with support of a dual-stack tuned for the 915MHz and 2.4GHz ISM bands, ContikiOS and its fully networking stack–an open-source operating system for IoT based solutions–, and an on-board power monitoring system to facilitate the performance evaluation of new techniques; (ii) a performance evaluation of the platform is provided within a linear net-

International Conference on Embedded Wireless Systems and Networks (EWSN) 2020
17–19 February, Lyon, France

ISBN: 978-0-9949886-4-5

Table 1. Qualitative comparison of different WSN platforms based on WuR.

Platform	WuS (TX/RX)	Data (TX/RX)	Dual-Stack Switching	Open HW	Operating System	Power Monitoring	Sensors
Zippy [7]	OOK, 434MHz	RF, 434MHz	×	×	×	×	External
ePanStamp[2]	OOK, 868MHz	RF, 868MHz	×	✓	×	×	×
Kratos [6]	OOK, 868MHz	LoRa, 868MHz	✓	✓	ContikiOS	×	On-board
OPENWUR - this work	**OOK, 915MHz**	IEEE 802.15.4, 2.4GHz	✓	✓	ContikiOS	On-board	External

work topology regarding power consumption and latency. Additionally, OPENWUR has been implemented with commercially available electronic devices to ease its replication by other researchers. The rest of the paper is organized as follows: In Section 2, we discuss the related work regarding WuR-based WSN platforms. Then, Section 3 provides details of the proposed system, and in Section 4, its implementation and evaluation results are presented. Finally, we conclude the paper and discuss future remarks in Section 5.

2 Related Work

In recent years, some WuR prototypes have been proposed in the literature for WSN applications [5]. However, most of these proposals have focused solely on the WuR hardware, i.e., the design and evaluation of dedicated circuits that continuously monitor the wireless channel and perform communications duties in a responsive and energy-efficient way, without considering the integration into a complete and open WSN platform for research and rapid prototyping purposes. However, some approaches have recently been proposed tackling this issue.

The Zippy platform, proposed in [7], includes a passive On/Off Keying (OOK) based demodulator tuned for the 434MHz ISM frequency band based on the AS3930 chip as the wake-up receiver (WuRx). Besides, for data communication, the CC110L transceiver was included with two RF ports connected to a single antenna via an ADG904 RF switch. Finally, Zippy provides an on-demand flooding technique for asynchronous network wake-up, and synchronous data dissemination over multi-hop WSNs. This platform has not yet been released to the research community compared to ePanStamp, an open-hardware platform [2]. ePanStamp incorporates a semi-active WuRx shield based on the AS3933 chip and OOK demodulator tuned for the 868MHz band with a receiver sensitivity of -60dBm. The PanStamp sensor node is provided for WuS and data transmissions at 868MHz. Similar to [7], the ePanStamp platform does not run an operating system.

Kratos [6] is an open-source platform for LoRa networks (prototyping and testing), which includes the SX1276 transceiver that supports the OOK modulation and the LoRa physical layer for data and WuS transmissions in the 868MHz band. Besides, Kratos provides an on-board passive WuRx with an OOK demodulator tuned for the 868MHz band with a receiver sensitivity of -50 dBm, coupled to an 8-bit ultra-low power MCU for selective triggering. Finally, Kratos runs ContikiOS, which was ported for LoRa networks applications, including its fully networking stack.

In Table 1, we present a qualitative comparison between our work and the aforementioned proposals. OPENWUR provides an open WSN platform that includes a semi-active WuRx shield based on the AS3933 chip and an OOK demodulator tuned for the 915MHz ISM band with a receiver sensitivity of -50 dBm. This shield is attached to a well-known sensor node, Zolertia's Re-Mote, for WuS transmission in the 915MHz band and data communication in the 2.4GHz band. Similar to [6], the proposed platform runs ContikiOS and its full networking stack. To our knowledge, our work is the first approach to propose a WuRx hardware that operates in the 915MHz band and to provide a dual-stack switching ported in ContikiOS to manage both the 915MHz and 2.4GHz transceivers. Besides, compared to other proposals, OPENWUR incorporates an on-board power monitoring module, which independently measures the energy consumed by the sensor node and WuRx in order to separately evaluate their performance.

3 Proposed Architecture

In this section, we provide an overview of the proposed WSN platform especially designed to prototype WuR-based applications. OPENWUR[1] is a platform that points to simplicity in its design and implementation, along with a low power and low-cost WuRx. In Figure 1, we present the overall architecture of the proposed platform, which encompasses four main blocks, which are described below regarding their design considerations: the OOK demodulator, the AS3933 low-frequency (LF) WuRx, the sensor node and the power measurement system.

3.1 OOK Demodulator

The complexity and reproducibility are key factors that allow designers to adjust and simplify their WuR circuit design, allowing faster prototyping. However, the WuR design with ultra-low energy consumption depends on the modulation scheme used for the WuS transmission, the RF front-end architecture and the chosen operating frequency [5].

Different modulation schemes can be employed for the WuR module, such as OOK, amplitude-shift keying (ASK) or frequency-shift keying (FSK). After reviewing the available literature in the area [2, 7], we decided to use an OOK demodulator-based receiver circuit using a voltage multiplier, which was implemented with commercially available components, as shown in Figure 2. Although ASK and FSK modulation schemes offer better noise immunity compared to OOK, they have higher energy consumptions than OOK [1]. The OOK demodulator includes two sub-modules to fulfill its duties: an impedance matching stage and an envelope detector tuned for the 915MHz band.

[1] `https://github.com/diegomecha/OpenWuR`

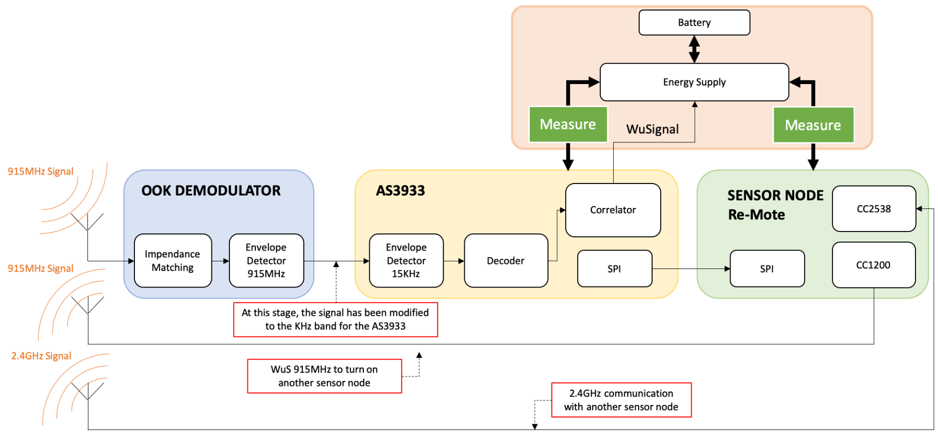

Figure 1. OPENWUR overall architecture.

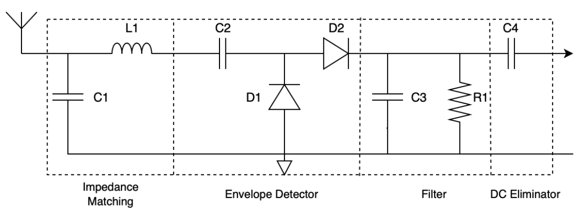

Figure 2. Typical OOK demodulator circuit.

3.1.1 *Impedance Matching*

In order to achieve the maximum power transfer between the antenna and the voltage multiplier, an impedance coupling or matching circuit is required as part of the OOK demodulator, as shown in Figure 2. If an RF signal hits the input side of the circuit, part of the signal is reflected, and another is transmitted through the circuit. The main goal of the impedance matching circuit is to reduce or eliminate the reflected RF signal. This procedure implies to measure how much power is reflected, i.e., the ratio of the reflected electromagnetic field to the incident field, known as the reflection coefficient (S11). It is crucial to keep in mind that the impedance matching circuit is strictly linked to the selected operating frequency, e.g., 915MHz. Therefore, although a receiver could operate in different frequency domains, the circuit is limited by the S11 parameter.

Given that the OOK demodulator circuit is a non-linear and time-varying circuit, finding an analytical model for the precise impedance is not trivial. However, it is possible to determine an approximate value of the circuit impedance by measuring the S11 parameter with a vector network analyzer (VNA). After several measurements, we obtained an impedance of $Z = 4.21 - j * 63.3\Omega$ for the OOK demodulator. Based on this result, we designed an L–match impedance matching circuit with $L = 27nH$ and $C = 4.7pF$, which provides a reflection coefficient of $-7.4150dB@915MHz$, as shown in Figure 3.

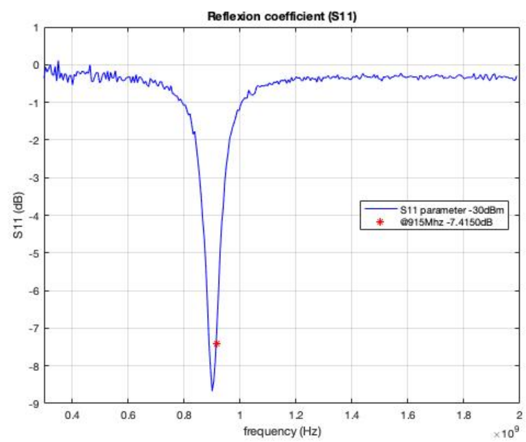

Figure 3. S11 coefficient for the L–match impedance matching circuit.

3.1.2 *Envelope Detector*

In the OOK modulation scheme, the data signal is transmitted by the source node as a sequence of 1's and 0's, represented by an RF signal with a fixed-amplitude or non-amplitude, respectively, as shown in Figure 4, allowing to save energy during the sequences of zeros, given that the transceiver can be turned off. On the receiver side, the modulated RF signal is detected by the rising edge of the signal. Given the simplicity of the OOK modulation scheme, implementing an OOK envelope detector circuit for WuRx in hardware is inexpensive and relatively straightforward because just a few discrete components are required to build it.

Besides the envelope detector, a low pass filter and a DC eliminator stages have also been added, as shown in Figure 2. The former removes the unwanted frequencies through the discharge capacitors C2 and C3 and the resistor R1. The latter helps to remove the DC components via the capacitor C4.

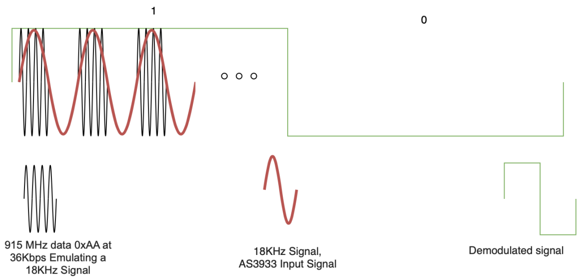

Figure 4. Example of a sub-carrier modulation scheme.

Table 2. OOK demodulator component values.

Component	Value
C1	4.7pF
L1	27nH
C2	8pF
C3	8pF
R1	100kΩ
C4	220pF

The values of the capacitors and resistor have been selected based on the study performed in [4] and via simulation tools (see Table 2).

3.2 AS3933 Low Frequency WuRx

After the OOK demolutator block, an AS3933 chip is integrated to work as the wake-up receiver module, as shown in Figure 1. This IC is mainly responsible for address decoding and correlation, in addition to implementing the low-frequency (LF) WuRx hardware in the 15–150kHz frequency range. Since the OOK demodulation failure probability increases with the frequency, it has been decided to use the lowest frequency range available for this chip (15–23kHz), with a central operating frequency of 18kHz. We chose the AS3933 IC because it is commercially available and provides a long battery lifetime and a long wake-up range.

The AS3933 can decode a 16/32-bit Manchester wake-up pattern, which is included in the demodulated incoming signal, that should follow a well-defined structure (i.e., Carrier Burst + Preamble + User-defined Pattern) to properly operate. Additionally, an internal clock calibration is required for decoding purposes, given that the AS3933 does not include an external oscillator.

The decoder has a set of timing rules regarding the carrier burst, preamble, and pattern duration, which depends on the carrier (f_{CRR}) and clock (f_{CLK}) frequencies. The clock frequency to be selected should fulfill $f_{CLK} = f_{CRR} * 14/8$, and for $f_{CRR} = 18kHz$, we get $f_{CLK} = 31.5kHz$ and the corresponding periods of $T_{CLK} = 31.75\mu s$ and $T_{CRR} = 55.55\mu s$. Finally, the equations shown in Table 3 are mainly used to determine the minimum length of the different parts of the sequence in order to fulfill the AS3933 protocol rules.

Considering $n = 7$, by utilizing Table 3 we get a bit time of $222\mu s$ and a minimum carrier burst of $3.365ms$, equivalent to 15.14 bits. However, since this is not a valid value, we utilize a minimum carrier burst of $3.544ms$, equivalent to 16 bits. Similarly, by using 13 bits we get a minimum duration

Table 3. Timing for the AS3933 IC.

Sequence Part	Minimum Duration
Bit Time	$n * T_{CLK}$
Minimum Carrier Burst	$92 * T_{CLK} + 8 * T_{CRR}$
Minimum Preamble	2.3 ms

of the preamble of $2.86ms$. Finally, considering that the 16-bit address pattern would have a duration of $3.544ms$, the total sequence would take $10ms$.

3.3 Sensor Node

The Sensor Node block is in charge of performing communication tasks, such as packet transmission and reception. There are two ways of communication between two nodes: (i) a 915MHz packet is received (or transmitted using the CC1200 radio) in order to activate the WuRx and wake up the corresponding node; and (ii) once both nodes are up and ready to communicate, the 2.4GHz radio (CC2538) is used to exchange data packets. In order to manage these two bands, we implemented a software-based dual-stack using the main 2.4GHz stack integrated into Zolertia's Re-Mote and the RIME stack available in ContikiOS for packet routing purposes. Since ContikiOS does not allow two communication stacks, for the 915MHz communication, it was necessary to activate the corresponding stack manually every time it was required.

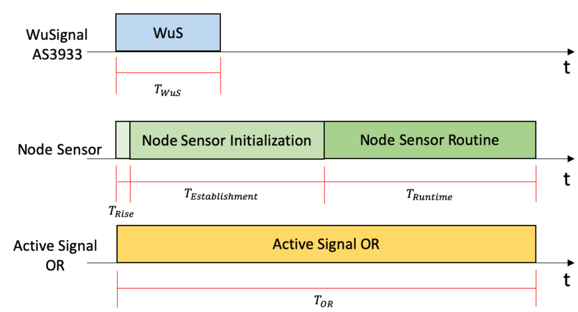

Figure 5. Simplified time diagram.

In OPENWUR, the Sensor Node block is turned on/off via the Power Measurement System based on a boolean decision rule. To that end, we implemented an OR-gate (SN74LVC1G32) to control when the power is supplied to the sensor node, by checking two signals: (i) the WuSignal generated by the AS3933 block once a correct sequence has been received, and (ii) one digital signal generated by the node while it is on, as shown in Figure 5. The signal generated by the AS3933 remains active for $100ms$, enough time for the node to wake up and generate its digital output to the OR-gate and maintain the power flowing to the node. After, the sensor node performs its tasks, it turns off automatically, deactivating the OR gate and sleeping until another WuSignal is produced. The active time of the node is calculated based on $T_{WuS} = 100ms$, $T_{Rise} = 19ns$, $T_{Establishment} = 560ms$ and $T_{Runtime}$ as shown in Figure 5. The node runtime is variable since it depends on the data communication tasks. Finally, the active OR signal time, T_{OR}, is defined as $T_{Rise} + T_{establishment} + T_{Runtime}$.

Figure 6. The implemented OPENWUR platform.

Although, this method is considerably more complex than simply putting the sensor node into a deep sleep mode (which is not completely feasible when using an operating system such as ContikiOS), it allows the node to consume $0W$ when it sleeps, significantly reducing the energy consumption in the network and extending the lifetime of the batteries.

3.4 Energy Supply and Power Measurement System

This block is responsible for supplying power to the whole platform based on the corresponding activation signals and to measure the energy consumed by the WuRx and Sensor Node independently, as shown in Figure 1, which is intended for energy efficiency studies. OPENWUR has been designed to be battery-powered with a 3.7V@1000mAh lipo battery. As mentioned before, the activation circuit is implemented with an OR-gate and a voltage regulator (MIC5205 3.6V) with an enable control input. Each measurement module consists of a TI INA219 DC current shunt and power monitor, which allows us to capture precise power measurements via an I2C communication interface, to also easily change its resolution via programmable registers.

4 Implementation and Evaluation

In this section, we present the implementation and validation of each module independently. Afterwards, OPENWUR is evaluated by implementing a simple linear network.

4.1 Modular Evaluation

Figure 6 shows the implemented OPENWUR platform. The WuR shield sits on top of the Zolertia's Re-Mote node and includes the OOK demodulator, LF WuRx, and the power monitoring system, which were implemented using commercially available electronic components and integrated within a PCB manufactured in a FR4 substrate ($\varepsilon = 4.4$, $h = 0.1588cm$) and 50Ω paths set as a grounded coplanar waveguides.

4.1.1 OOK Demodulator Sensitivity

We measured the sensitivity of the designed OOK demodulator circuit using a Rohde & Schwarz SMW200A vector signal generator. Table 4 presents the output of the circuit depending on the input power, varying from $-50dB$ to $0dB$. As a result, the designed demodulator provides a sensitivity of $-50dB$, approximately.

4.1.2 AS3933 Decoder

We validated the LF WuRx implemented with the AS3933 chip, regarding the decoding of the sequence of bits described in Section 3.2 and the generation of the activation signal. Figure 7 shows the output behavior of the AS3933 decoder, having as input the signals processed by the OOK demodulator, where it is possible to see the wake-up pattern, i.e., carrier burst, preamble and user-defined pattern (`0x9669`). The node address is implemented through the user-defined pattern, which is validated through the AS3933 correlator to generate the activation signal for the power measurement system. The results show that the activation signal rises when the correlator validates the address, which indicates that the designed OOK demodulator and decoder work as expected.

Table 4. OOK Demodulator Sensitivity

Power [dBm]	-50	-45	-40	-35	-30	-25	-20	-15	-10	-5	-0
V_{OUT} [mV]	21.9	23	27.1	39.5	72.1	146	297	591	1140	2170	4000

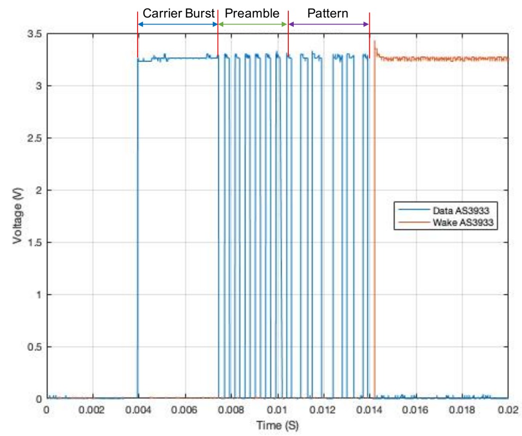

Figure 7. Decoding wake-up signal behavior.

4.1.3 Power Measurement System

We validated the reliability of the power measurement system against theoretical and experimental results using a simple voltage divider circuit, by measuring the current through a known resistance value. As a result, the system can provide measurements with 2.53% of average error, which is acceptable for energy study purposes in WSN prototyping.

4.1.4 Sensor Node

As mentioned before, the sensor node is implemented using Zolertia's Re-Mote Rev-A platform, which includes a 32MHz ARM Cortex-M3 with 512KB programmable flash and 32KB of RAM, and transceivers in the 2.4GHz ISM band (TI CC2538) and the 868/915MHz ISM band (CC1200) to allow dual band operation. From the software perspective, the sensor node runs the well-known and lightweight operating system for IoT applications, ContikiOS, with its fully networking stack for collecting, routing, and packet-forwarding, and the additional 915Mhz stack integrated at the application level of the operating system.

4.2 Complete OPENWUR Evaluation

For the platform evaluation, we have configured a simple linear network of two nodes powered by 3.7V@1000mAh lipo batteries. The communication is performed within one hop, where the source node emits a 45-byte WuS packet in the 915MHz band to activate the destination WuRx and turn on the node. After the node wakes up, an 8-byte data packet is exchanged using the 2.4GHz transceiver. Both the WuS and data packets are generated only once. We defined two quantitative metrics: *Power Consumption* as the energy consumed by the node and the WuRx in the different operational stages, and *Latency* as the time from the moment the source node generates the WuS packet to the moment the data packet is received at the destination node. The latency has been measured using two trigger signals via a digital oscilloscope. Finally, we have conducted five independent experiments to average the metrics.

Figure 8 shows the power consumption profile of the destination sensor node in different stages. The node consumes $53mW$ during its awake and operating system loading stages, which takes $560ms$, approximately between the marks at $0.75s$ and $1.3s$ in the graph. During the data packet reception and acknowledgment transmission, it consumes $120mW$, roughly between the $1.3sand1.8s$ marks. Regarding the WuRx module, the energy consumption is around $10.5\mu W$ during the listening stage and $23.5\mu W$ while demodulating and decoding the WuS, approximately 10000 times less than that of the sensor node. In the state-of-the-art, it is possible to find power consumptions as low as $150nW$ [3] for a fully customized WuRx IC, though it is highly dependable on the sensitivity achieved and the addressing capabilities. However, since OPENWUR is oriented for prototyping purposes, whenever someone wants to compare the performance of a given WuR application running on OPENWUR to the state-of-the-art, it is as simple as dividing the power consumption of the WuRx module by a factor of 100.

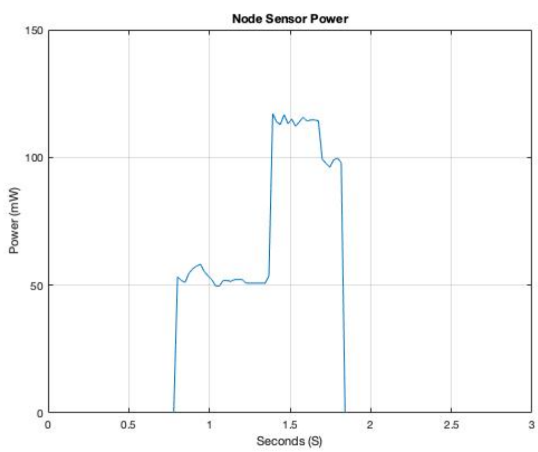

Figure 8. Power consumption profile of the sensor node.

Regarding the latency of sending one packet in a one-hop communication, we obtained a value of approximately $1s$. This result depends strongly on the boot-up process of the operating system and accounts for almost 56% of the total latency. This is a large percentage because in these experiments the routine that the node executes is very simple (receive a packet and send an acknowledgement). However, having the sensor node completely off allows the power consumption to be reduced to $0W$. Therefore, for those WSN applications where data packet transmission is non-deterministic, this latency might be acceptable with respect to power consumption–latency ratio. For more latency-strict applications, the operating system could be discarded and a custom firmware could be implemented. Finally, Figure 9 summarizes the power consumption and time profiles of OPENWUR during the different stages.

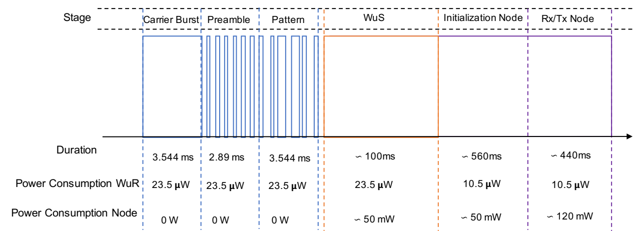

Figure 9. OPENWUR consumption profile.

5 Conclusions and Future Work

In this paper we have presented the development of OPENWUR, an open-source WSN platform based on the WuR paradigm, with dual-stack capabilities (915MHz and 2.4GHz bands), full support for ContikiOS and its networking stack, and with an on-board power monitoring system for performance evaluation purposes. Besides, we have reported several experiments where we evaluated the implemented modules and their integration into OPENWUR within a linear network topology of two nodes regarding energy consumption and latency. The designed power management system allows for independent power measurement of the sensor node and the WuR module, which is very important when prototyping applications in order to understand the full effect of the WuR module in the implemented algorithm. We plan to explore more complex WuR-based routing techniques and their effects in latency and energy when considering addressing and non-addressing schemes in a multi-hop WSN communication.

6 References

[1] J. Ansari, D. Pankin, and P. Mahonen. Radio-Triggered Wake-ups with Addressing Capabilities for Extremely Low Power Sensor Network Applications. 2008.

[2] S. Bdiri, F. Derbel, and O. Kanoun. An 868 MHz 7.5uW wake-up receiver with-60 dBm sensitivity. *Journal of Sensors and Sensor Systems*, 5(2):433–446, 2016.

[3] M. Magno, V. Jelicic, B. Srbinovski, V. Bilas, E. Popovici, and L. Benini. Design, Implementation, and Performance Evaluation of a Flexible Low-Latency Nanowatt Wake-Up Radio Receiver. *IEEE Transactions on Industrial Informatics*, 12(2):633–644, 2016.

[4] H. J. V. Pflug, Hans W. *Wireless Power Transfer: Discrete Rectifier Modeling and Analysis*. 2016.

[5] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli. Ultra Low Power Wake-Up Radios: A Hardware and Networking Survey. *IEEE Communications Surveys Tutorials*, PP(99):1–1, 2017.

[6] R. Piyare, A. L. Murphy, M. Magno, and L. Benini. KRATOS: An Open Source Hardware-Software Platform for Rapid Research in LPWANs. In *14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–4, Oct 2018.

[7] F. Sutton, B. Buchli, J. Beutel, and L. Thiele. Zippy: On-demand network flooding. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 45–58, New York, NY, USA, 2015. ACM.

[8] F. Sutton, R. D. Forno, J. Beutel, and L. Thiele. BLITZ: Low Latency and Energy-Efficient Communication for Event-Triggered Wireless Sensing Systems. *ACM Trans. Sen. Netw.*, 15(2), Mar. 2019.