Power Gating and Its Application in Wake-Up Radio

Nahit Pawar
CNRS SAMOVAR UMR 5157
Télécom SudParis, Évry, France
nahit.pawar@telecom-sudparis.eu

Thomas Bourgeau
KB DIGITAL AG
Gstaltenrainweg 80, Riehen, Switzerland
thomas.bourgeau@kb-digital.ch

Hakima Chaouchi
CNRS SAMOVAR UMR 5157
Télécom SudParis, Évry, France
hakima.chaouchi@telecom-sudparis.eu

Abstract

Internet of Things (IoT) intelligent devices such as wireless sensor actuator nodes (WSANs), pervades virtually every aspect of our lives. These devices are expected to be operated on a limited energy budget, therefore reducing energy consumption is considered an important task from both hardware and software standpoint. Out of all operations performed by WSANs, the wireless communication is known to consume a significant portion of the energy budget. Moreover, in between transmission the WSAN spends most of its time in inactive (sleep) state consuming only a fraction of the energy budget, but if not properly taken into account can play a decisive role in battery life, as the time spent in inactive state increases.

In this paper, we analytically analyze the importance of inactive current on lifetime of battery operated wireless node. As a result of this analysis, we propose a technique of power gating to reduce inactive current. We implement this technique in a wireless node platform to showcase an effective decrease in inactive current by 90-95%. In addition, we propose a wireless node architecture that integrates power gating (to reduce inactive power) with wake-up radio (to reduce communication power) enabled wireless node to further enhance battery life.

Categories and Subject Descriptors

B.m [**Hardware**]: MISCELLANEOUS

General Terms

Design, Experimentation, Measurement

Keywords

IoT, Wireless Sensor Actuator Nodes, Power Gating, Wake-up Radios, Power Optimization, Hardware Platform

1 Introduction

Internet of Things (IoT) encompasses a large network of heterogeneous devices, especially resource constrained devices used in wireless sensor actuator nodes (WSANs). Moreover the advancement in technologies like wireless communication, ultra-low power processors & system-on-chips (SoC), embedded sensors & actuators, Radio frequency IDentification (RFID) has enabled the development of low power WSANs that can be operated on batteries, supercapacitors or energy harvested from the environment. Nonetheless WSANs still requires innovative energy aware hardware and software solutions to achieve multi years of battery lifetime without interruption. In WSANs wireless communication consumes considerable amount of power as compared to any other operation performed by WSANs and is considered an important component that influence battery life, therefore it becomes necessary to come up with systematic solution to analyze and reduce communication power.

One method is to model the wireless power consumption of an IoT end-devices, although it is not an easy task as it requires many technology dependent parameters [19], nevertheless the power models are good for comparing various wireless technologies at an early stage of technology selection. For example, [13] presents analytical models of LoRaWAN end-device's current consumptions derived from measurements performed on existing prevalent LoRaWAN hardware platform, these models are useful to study the impact of various LoRaWAN physical and Medium Access Control (MAC) parameters (data rates, acknowledge transmission, payload size and bit error rate) on power consumption and ultimately to have a rough estimate of battery life. There are also other techniques to reduce power at the protocol level, for example [11] proposes an energy efficient protocol stack for multi-hop communication for LPWANs technology.

From a design perspective, an energy efficient ultra low power wake-up receivers (WURx) [14, 18] are gaining a lot of attentions for reducing power consumption by relieving the main radio from continuously monitoring the channel for incoming messages. The WURx acts as an auxiliary receiver and wake-up the wireless node from sleep using interrupt, only after detecting a potential incoming message.

We also believe that apart from wireless communication power, much less attention has been paid to the importance of end-device's sleep current on battery life, especially when

the data is transmitted very infrequently there by end-device spends most of its time in inactive (sleep) state, which makes inactive current a significant contributor to battery life.

In this paper, we implement in our platform a hardware technique know as *power gating* [20] to reduce power (leakage power) between two consecutive transmit, when microcontroller (MCU) is in inactive state. We showcase with this technique and using the measurement results presented in [13] the significant increase in battery life, when the inactive period is large and inactive current starts to dominate.

We further argue that this technique can be useful and can complement existing wake-up radios (WUR) receivers [17, 18] with minimal changes in the circuit design but requires further experimental validation and performance evaluation.

Along with this, the proposed platform enables easy, cost effective, battery operated and low power solution for experimental LoRaWAN [12] field studies and can also be used to validate the power models [19] of LoRaWAN technology and better estimate battery life. Although our platform uses LoRaWAN radio, but the power gating technique is independent of any wireless technology and other hardware components. Moreover, our solution can also be used to complement the technique presented in [11], where [11] reduces active state current and our solution reduces inactive state current.

The paper is organized as follows. We first, In Section 2 describe in brief the hardware details of our platform. Then in Section 3, power gating technique is explained in conjunction with platform power management and operation. In Section 4, an effective decrease in platform's inactive current consumption using power gating is compared and analyzed against current reduction using power down modes. Next, in Section 5, we describe analytically the importance of inactive current on the lifetime of battery-operated end-devices. Then in Section 6, we explain the effects of power gating on wasted power during wake-up. In Section 7, we propose the integration of power gating in a wake-up radio enabled wireless node. Finally, Section 8 concludes with future work.

2 Platform Details

The platform consists of two boards that together forms a battery operated wireless node. As shown in Figure 1, the *first board* (referred to as B1) combines on a single board an Arduino compatible MCU (Atmega328p) with LoRaWAN radio (RFM95W module). The bootloader inside the MCU is same as *Arduino Pro Mini 3.3V-8MHz* therefore it is possible to use the Arduino IDE to create applications and debug via external FTDI serial adapter, similar to Arduino Pro Mini. The *second board* (referred to as B2) is a power-supply and power-management board for B1. It uses nanoPower boost converter (MAX17223 [1]) to convert voltage from two 1.5V AAA battery (3.0V in series) into regulated 3.3V output voltage for B1. The value of the boost converter inductor is carefully selected to reliably convert input voltage ranging from 2.0V to 3.0V into regulated 3.3V at 200 mA output current. For power management (Section 3), a nano-power system timer (TPL5111 [8]) is used to temporarily disable power from B1, the timer's time interval is programmable using external resistor.
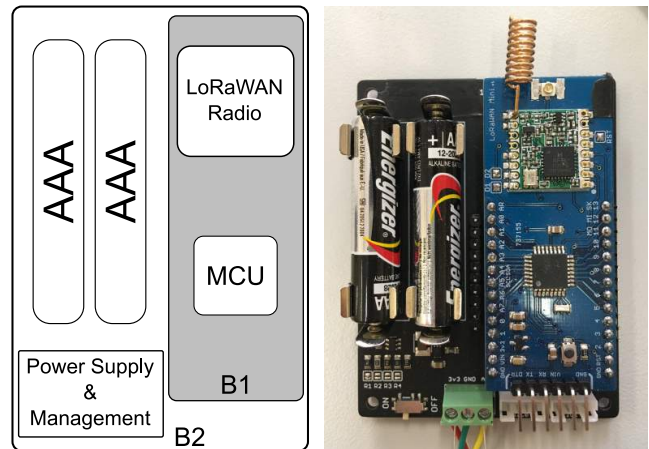

Figure 1. Platform block diagram

3 Power Gating and Platform Power Management & Operation

Power gating [15, 20] is a technique to reduce power consumption by temporarily and selectively shutting down current from the circuits/subcircuits that are not in use. This allows reducing overall system standby current during inactive (sleep) state. Figure 2 illustrates a generalized block digram of a typical system utilizing power gating. The power gating controller generates the required power gating signals to selectively and temporarily disable power from various subcircuits/subsystems using electronic switches.

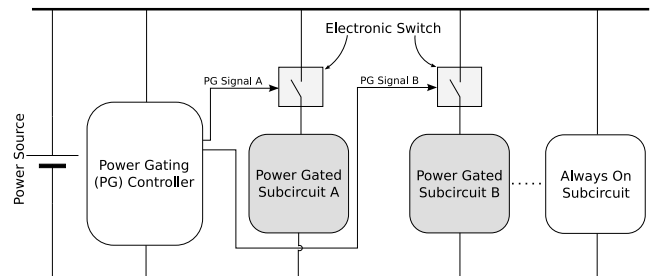The power management of our platform is based on the

Figure 2. Illustrative block diagram of power gating

principle of power gating, where the power is temporarily removed from the whole circuit (B1) during inactive state and making sure the power is back again when B1 requires power for operation.

The power management is purely in hardware and is independent of wireless technology. As illustrated in Figure 3, the B2 has two important components - boost converter (power converter & electronic switch) and system timer (power gating controller). The boost converter generates the regulated 3.3V for the proper functioning of B1 and the system timer. The system timer waits for the MCU (B1) to generate power disable signal and after receiving the signal, the timer disables the boost converter, thereby removing power from B1, this process is called *"self destruction"*.

Since the system timer is continuously powered, therefore it is still ticking and after a predefined time interval it enables the boost converter and the power is back again. Before disabling the power the MCU performs the necessary IoT application task as programmed by the user. Figure 4 shows the timing diagram of platform operation. This technique is useful for star networks like LPWAN (LoRaWAN, Sigfox, etc.) and for applications with infrequent transmission, for example - Smart Cities, Smart Agriculture, etc.
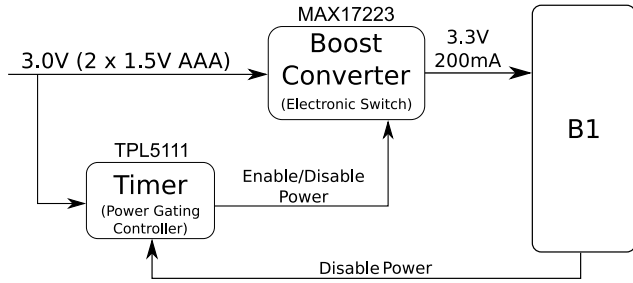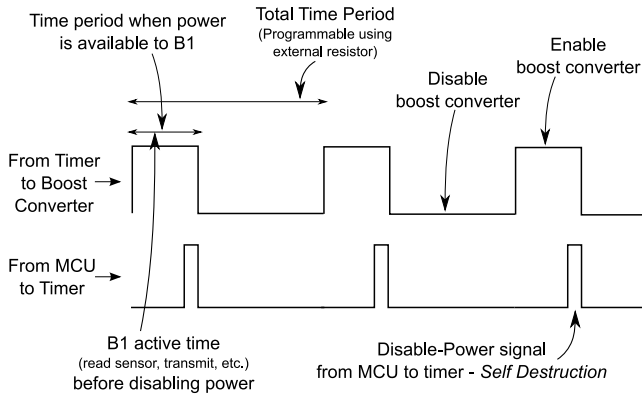

Figure 3. Block diagram of B2

Figure 4. Platform operation timing diagram

4   Platform Power Consumption

The advantage of technique mentioned in Section 3 is that between two consecutive transmit/receive the total shutdown current drawn $I_{T_{SD}}$ from the batteries is effectively reduced to the current consumed by the system timer I_{ST}, shutdown current of boost converter $I_{BC_{SD}}$ and an unknown platform dependent leakage current $\delta_{leakage}$ - see Equation (1). The $\delta_{leakage}$ is calculated as the difference between analytical and measured current consumption of the platform, this allows to take into account the quality of overall platform circuit design and is usually very small. After obtaining the values of I_{ST} and $I_{BC_{SD}}$ from their respective device datasheet, the $I_{T_{SD}}$ is approximately equal to Equation (2).

$$I_{T_{SD}} = I_{ST} + I_{BC_{SD}} + \delta_{leakage} \qquad (1)$$

$$I_{T_{SD}} = 35nA + 0.5nA + \delta_{leakage} \qquad (2)$$

In contrast with other techniques, where we utilize various power down modes of the components to reduce the effec-

tive power during sleep, the total power down (sleep) current ($I_{T_{PD}}$) consumed is given by Equation (3). Where I_{BC_A} is boost converter active current (always on to supply sleep current) and $I_{MCU_{PD}}$ & $I_{R_{PD}}$ are MCU & Radio power down currents respectively. In this case the system timer (I_{ST}) is used to wake up MCU from sleep. After obtaining the values of I_{ST}, I_{BC_A}, $I_{MCU_{PD}}$ & $I_{R_{PD}}$ from their respective device datasheet, the $I_{T_{PD}}$ is approx. equal to Equation (4).

$$I_{T_{PD}} = I_{ST} + I_{BC_A} + I_{MCU_{PD}} + I_{R_{PD}} + \delta_{leakage} \qquad (3)$$

$$I_{T_{PD}} = 35nA + 500nA + 100nA + 200nA + \delta_{leakage} \qquad (4)$$

It is important to note that Equation (3) does not include current consumed by external components attached to B2 such as sensors, actuators, etc. as they are application dependent. One can imagine the addition of more sleep currents if these external components are attached to the platform, where as Equation (1) is independent of various sub-system sleep and leakage currents. Also in general the total platform's sleep current depends on the type of components used and the overall circuit design, therefore Equation (3) varies from one platform to another. Table 1 lists sleep currents for few popular hardware platforms. Note from Table 1 that sleep current of [10] and [3] differs considerably even though they have same MCU and Transceiver, this is because [10] is a module and requires additional hardware components (power supply, debug circuit, etc.) before it can be used.

Table 1. Sleep Current for various IoT Devices

Device Name	MCU	Transceiver	Sleep Current
MultiTech mDot [2] *	STM32F411 ⋆	SX1272 [4]	$40\mu A$
iM880B-L [10] *	STM32L151 •	SX1272	$1.85\mu A$
NetBlocks XRange [3] †	STM32L151	SX1272	$70\mu A$
iM222A [9] *	CC2530 °		$1\mu A$
* Module, † Platform, ⋆ [5], • [6], ° [7]			

5   Importance of Inactive Current - An Analytical Evaluation

It is possible to generalize the average current (I_{avg}) consumption of any hardware platform, let us consider that the platform is programmed for periodic message transmission (IoT monitoring application) with period T_{Notif} (notification period), than I_{avg} can be calculated as :

$$I_{avg} = \frac{1}{T_{Notif}} \int_0^{T_{Notif}} i(t)dt = \frac{1}{T_{Notif}} \left(\int_0^{T_{active}} i_{active}(t) \right.$$
$$\left. + \int_{T_{active}}^{T_{Notif}} i_{inactive}(t) \right) dt \qquad (5)$$

T_{active} is the total time spent in active state (wake-up, read sensor, transmit, etc.), where $i_{active}(t)$ and $i_{inactive}(t)$ are current consumption profile in active and inactive state respectively. Also $T_{inactive}$ can be obtained as :

$$T_{inactive} = T_{Notif} - T_{active} \qquad (6)$$

It is important to note that, $i_{active}(t)$ depends on numerous factors such as processing unit, operating voltage & frequency, transceiver, application algorithm, wireless network & technology, temperature, etc. and therefore it is difficult to model the behavior accurately which requires careful measurement setup & methods [16], nevertheless [19] presents analytical models of energy consumption for various operations in wireless sensor device like - data acquisition (regular and event driven), data processing (hardware dependent), data communication (point-to-point - SIGFOX & LoRa and time synchronized network - TSCH) and also [13] presents analytical models for both acknowledged and unacknowledged LoRaWAN transmission for various spreading factors (SF) based on the measured current consumption on an actual hardware platform.

On the other hand $i_{inactive}(t)$ can be considered constant throughout $T_{inactive}$ and is equivalent to the summation of all the sleep currents of various components available on the platform and active currents of always-On components (for example, Equation (3)). Therefore Equation (5) can be simplified as shown in Equation (7), where $I_{inactive}$ is total sleep current of the platform and $\frac{T_{active}}{T_{Notif}}$ is duty cycle.

$$
\begin{aligned}
I_{avg} &= \frac{1}{T_{Notif}} \left(\int_0^{T_{active}} i_{active}(t)dt + I_{inactive} \cdot T_{inactive} \right) \\
&= \frac{1}{T_{Notif}} \int_0^{T_{active}} i_{active}(t)dt + I_{inactive} \left(1 - \frac{T_{active}}{T_{Notif}} \right)
\end{aligned}
\tag{7}
$$

Further using Equation 7, we can calculate the theoretical lifetime, denoted by $T_{lifetime}$, of a battery-operated end-device as shown in Equation (8), where C_{bat} is battery capacity expressed in mAh (milliampere hour)

$$
T_{lifetime} = \frac{C_{bat}}{I_{avg}}
\tag{8}
$$

One of the main goal of IoT embedded designers and application developers is to increase $T_{lifetime}$ by systematically modifying the factors that influence I_{avg}. It is easy to visualize from Equation (7) the factors - T_{Notif}, T_{active}, i_{active} and $I_{inactive}$ - that determines $T_{lifetime}$ for a given C_{bat}. In this paper we consider the impact of T_{Notif} and I_{sleep} on $T_{lifetime}$.

We can calculate the asymptotic theoretical upper bound of $T_{lifetime}$ denoted by $\hat{T}_{lifetime}$, using Equation (7) & (8) and taking the limit $T_{Notif} \to \infty$, $\hat{T}_{lifetime}$ is given by :

$$
\begin{aligned}
\hat{T}_{lifetime} &= \frac{C_{bat}}{I_{inactive}} \\
I_{inactive} &= \lim_{T_{Notif} \to \infty} I_{avg}
\end{aligned}
\tag{9}
$$

The implication of Equation (7) & (9) is that, the contribution of $I_{inactive}$ in I_{avg} starts to dominate as T_{Notif} becomes large as compared to T_{active} i.e $\frac{T_{active}}{T_{Notif}} \ll 1$. Furthermore, while designing an IoT end-device one can easily gather sleep currents ($I_{inactive}$) of various components from manufacturer's datasheet and can easily predict $\hat{T}_{lifetime}$ early in design phase. Also it is relatively easy to measure $I_{inactive}$ accurately using inexpensive equipments.

Now we illustrate the effect of $i_{active}(t)$, T_{active} and T_{Notif}

on $T_{lifetime}$ for a given $I_{inactive}$. Without loss of generality and for realistic assumption of $i_{active}(t)$ and T_{active}, we decided to reuse the measurement data of LoRaWAN transmission from [13] and is summarized in Table 2, where $C_{active} = \int_0^{T_{active}} i_{active}(t)dt$. We also assumed a battery capacity of 2400 mAh, which is equivalent to AA battery capacity. As illustrated in Figure 5, for relatively lower values of $i_{active}(t)$ and T_{active} (i.e SF7) 50 percent of $\hat{T}_{lifetime}$ (1000 days) is achieved earlier as we increase T_{Notif} ($T_{Notif} \approx 27$ min for SF7 and $T_{Notif} \approx 85$ min for SF12).

Next, we illustrate the effect of $I_{inactive}$ on $T_{lifetime}$ as a func-

Table 2. LoRaWAN Transmission Measurements

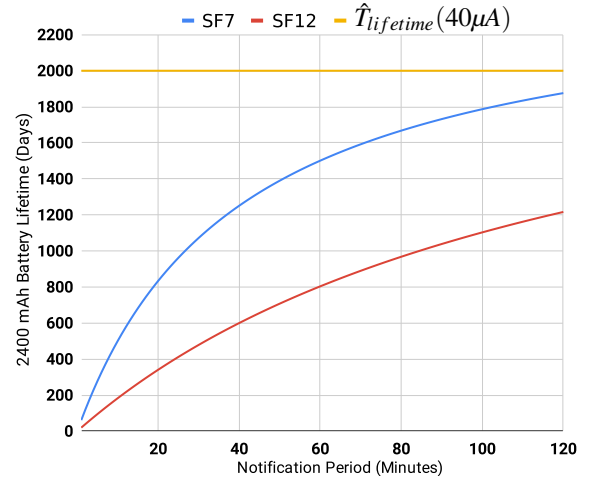| Parameters		C_{active}	T_{active}	$I_{inactive}$
SF	Payload			
7	242 bytes	95.93 mA.s	2.934 sec	$40\mu A$
12	51 bytes	304.15 mA.s	5.577 sec	$40\mu A$
Common Parameters - Class A protocol, Bandwidth BW = 125 KHz, 8-symbol preamble length				

Figure 5. $T_{lifetime}$ for LoRaWAN SF7 & SF12 transmission as a function of T_{Notif}

tion of T_{Notif} for a given $i_{active}(t)$ and T_{active}, again for realistic assumption and without loss of generality we use the data of Table 2. Figure 6 illustrates $T_{lifetime}$ for various $I_{inactive}$ as a function of T_{Notif} for LoRaWAN SF7 & SF12 transmission. Note that $I_{inactive} = 50nA$ (Equation 2) corresponds to the situation, where power gating is used to reduce inactive current and $I_{inactive} = 40\mu A$ as used in [13].

It is interesting to note that for lower values of T_{Notif}, ≤ 9 min for SF7 and ≤ 20 min for SF12 there is not much difference in $T_{lifetime}$ because C_{active} dominates I_{avg} and hence $T_{lifetime}$. The implication of this result is that in an application where there is a requirement of frequent transmission which corresponds to short T_{Notif}, one should try to reduce $i_{active}(t)$ & T_{active} rather than inactive current in order to obtain the desired $T_{lifetime}$.

Furthermore, the effect of $I_{inactive}$ starts to dominate early for lower values of $i_{active}(t)$ & T_{active} as we increase T_{Notif} (≥ 9 min for SF7 and ≥ 20 min for SF12). Interestingly in an application with infrequent transmission which corresponds
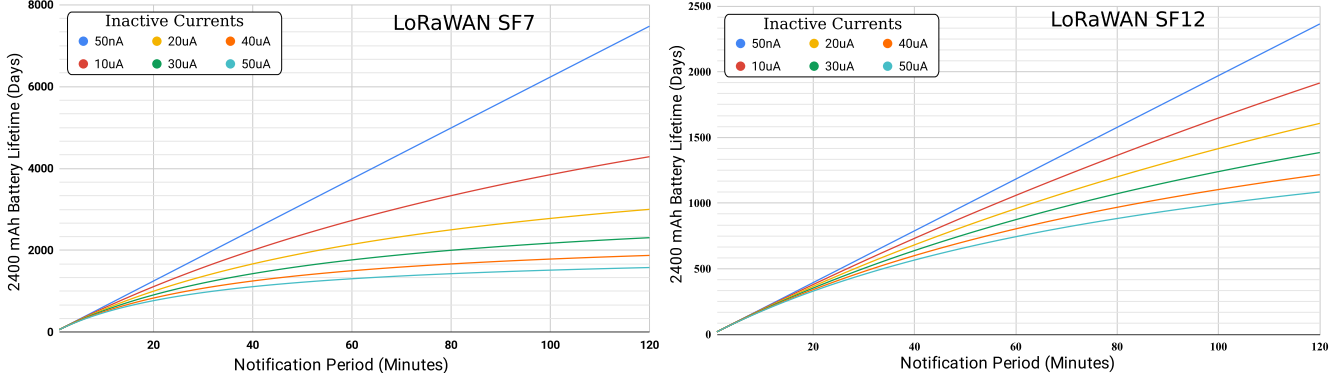
Figure 6. Battery life vs notification period for various inactive currents - LoRaWAN : SF7 and SF12

to large T_{Notif}, where $I_{inactive}$ dominates, one can try to incorporate techniques (for example - power gating) to reduce inactive current rather than $i_{active}(t)$ & T_{active} to obtain better end-device lifetime.

6 The Effects of Power Gating on Wake-up

It is necessary to understand the effect of wake-up from power gating in comparison with wake-up from sleep on power consumption. In general, it is evident that for power gating both wake-up time - T_{wu}^{pg} and the current consumed during wake-up - $i_{wu}^{pg}(t)$ are typically more than its equivalent sleep wake-up parameters - T_{wu}^{s} and $i_{wu}^{s}(t)$ and hence more wasted power during wake-up, as illustrated in Figure 7. This is because, $i_{wu}^{pg}(t)$ and t_{wu}^{pg} involves a significant contribution from transient currents of decoupling capacitors, oscillator & voltage regulator startup time, boot-up time, etc. which can take milliseconds to settle and depends on the hardware platform under consideration. Since it can take considerable time for the power gated circuit to transition from inactive to active state, therefore it is not practical to utilize power gating for frequent data transmission (short T_{Notif}).

Although, it is also possible to selectively power gate sub-circuits (sensors, actuator, transceiver, etc.) instead of whole system, in this situation the processing unit also manages the task of power delivery to various sub circuits, for example it makes sense to power gate transceiver sub-circuit in an application where the sensor data is transmitted only if certain conditions are satisfied. The selective power gating allows fine grain control of power distribution.

We modified Equation 7 to distinguish the effect of wake-up on I_{avg} as shown below :

$$I_{avg} = \frac{1}{T_{Notif}} \left(\int_0^{T_{wakeup}} i_{wakeup}(t)dt + \int_{T_{wakeup}}^{T_{active}} i_{active}(t)dt + I_{inactive} \cdot T_{inactive} \right) \quad (10)$$

Further experimental analysis is need to model the behavior of wake-up from power gating and to evaluate the gain in $T_{lifetime}$ versus T_{Notif} against wake-up from sleep.

7 Power Gating and Wake-Up Radio

In this section we propose a wireless node architecture that integrates power gating circuit of Figure 3 with a wake-
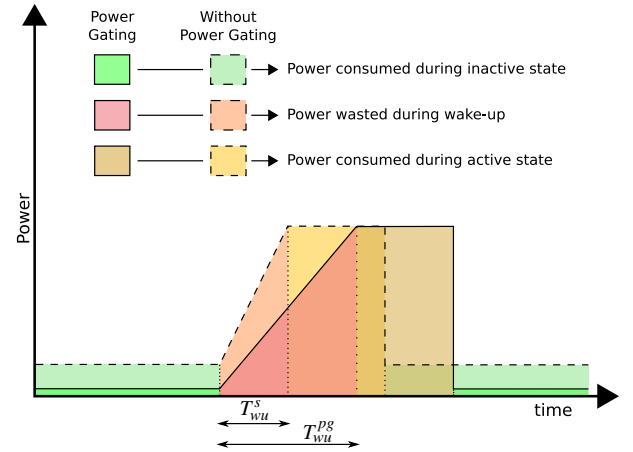

Figure 7. Effect of power gating on wake-up time and power

up radio enabled wireless node.

The wake-up radio is a technique to reduce communication power of wireless nodes whereby an auxiliary receiver circuit known as wake-up radio (WUR) receiver relieves main radio from continuous listening of transmission medium for an incoming messages [14, 18]. Wake-up radios employ an asynchronous wake-up mechanism to notify main processing unit for a potential incoming message. In contrast with wireless node without wake-up radio the idle listening of main radio receiver consumes more energy, therefore one of the main goal in wake-up radio is to design an ultra low power (also, low latency and high sensitivity) receiver circuit that consumes significantly low power in comparison with idle listening of main radio.

Figure 8 illustrates the block diagram of wireless node with WUR and power gating. The power gating circuit comprises of an ultra low power system *timer* that acts as power gating controller and a *boost converter* that acts as an electronic switch to enable or disable power in response to the signal generated by the timer. The timer waits for the processing unit to generate power disable signal and in response to this signal the timer turns off the electronic switch (boost converter), thereby removing power from the main radio and processing unit. After predefined time interval the timer en-

ables the boost converter, thereby restoring the power. The timer's power enable signal is also used as a level-interrupt (Timer-Level-Int) for processing unit to distinguish timer wake-up from WUR wake-up.

On the other hand, the WUR consist of in addition to wake-up receiver an ultra low power microcontroller (ULPμC) for address matching, similar to the one presented in [18]. On successful address match the ULPμC issues a power enable signal which also acts as level-interrupt (WUR-Level-Int) to the main processing unit (PU) to distinguish WUR wake-up from timer wake-up, this allows processing of incoming messages. The main PU can instruct ULPμC to disable power through SPI interface. The OR gate allows wireless node to receive power enable/disable signal from both timer and WUR.

As shown in Figure 8, the proposed wireless node architecture requires only few external components (timer, electronic switch and OR gate) and two extra connections (disable power and interrupt signals) to incorporate power gating circuit in a wake-up radio enabled wireless node.
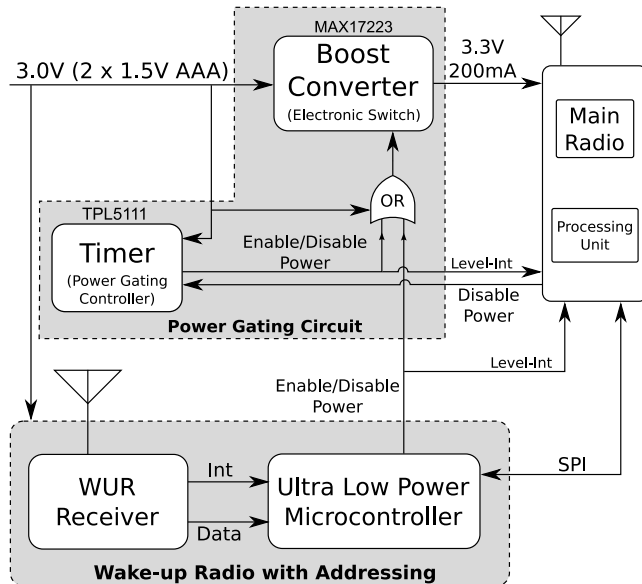

Figure 8. Block diagram of wireless node with wake-up radio and power gating

8 Conclusion

We analytically examined the impact of inactive current of wireless node on battery life. We analyzed that the inactive current starts to dominate battery life, as the time spent in inactive state increases, which is typical in IoT applications with infrequent transmission. We also derived the theoretical upper bound on battery life that depends solely on inactive current for a given battery capacity.

In order to reduce inactive current, we introduced a hardware technique of power gating and implemented this technique in our low power wireless node platform as a power management feature. From this technique we were able to achieve 90-95% reduction in inactive current. We also qualitatively analyzed the effects of power gating on power wasted during

wake-up.

Finally, we proposed a wireless node architecture that integrates power gating mechanism in a wake-up radio enabled wireless node to reduce inactive current and thereby further enhancing battery life.

9 References

[1] MAX17223 Nano-Power Boost Converter, Datasheet. https://datasheets.maximintegrated.com/en/ds/MAX17220-MAX17225.pdf. Accessed on 22 October 2019.

[2] MultiTech mDot Long Range LoRa Module. https://www.multitech.com/documents/publications/data-sheets/86002171.pdf. Accessed on 22 October 2019.

[3] NetBlocks XRange SX1272 LoRaNode. https://www.netblocks.eu/xrange-sx1272-lora-datasheet/. Accessed on 22 October 2019.

[4] Semtech SX1272 Transceiver. https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1272. Accessed on 22 October 2019.

[5] STMicroelectronics : STM32F411 MCU. https://www.st.com/resource/en/datasheet/stm32f411ce.pdf. Accessed on 22 October 2019.

[6] STMicroelectronics : STM32L151 MCU. https://www.st.com/resource/en/datasheet/stm32l151qc.pdf. Accessed on 22 October 2019.

[7] Texas Instruments : CC2530 System-on-Chip. http://www.ti.com/lit/ds/symlink/cc2530.pdf. Accessed on 22 October 2019.

[8] TPL5111 Nano-Power System Timer, Datasheet. http://www.ti.com/lit/ds/symlink/tpl5111.pdf. Accessed on 22 October 2019.

[9] WiMOD iM222A IEEE 802.15.4 Radio Module. https://www.wireless-solutions.de/downloads/Radio-Modules/iM880B/General_Information/iM880B_Datasheet_V1_6.pdf. Accessed on 22 October 2019.

[10] WiMOD iM880B Long Range Radio Module. https://www.wireless-solutions.de/downloads/Radio-Modules/iM880B/General_Information/iM880B_Datasheet_V1_6.pdf. Accessed on 22 October 2019.

[11] T. Adame Vázquez, S. Barrachina-Muñoz, B. Bellalta, and A. Bel. HARE: Supporting Efficient Uplink Multi-hop Communications in Self-Organizing LPWANs. *Sensors*, 18(1):115, 2018.

[12] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne. Understanding the Limits of LoRaWAN. *IEEE Communications magazine*, 55(9):34–40, 2017.

[13] L. Casals, B. Mir, R. Vidal, and C. Gomez. Modeling the Energy Performance of LoRaWAN. *Sensors*, 17(10):2364, 2017.

[14] I. Demirkol, C. Ersoy, and E. Onur. Wake-up Receivers for Wireless Sensor Networks: Benefits and Challenges. *IEEE Wireless Communications*, 16(4):88–96, 2009.

[15] F. Fallah and M. Pedram. Standby and Active Leakage Current Control and Minimization in CMOS VLSI Circuits. *IEICE transactions on electronics*, 88(4):509–519, 2005.

[16] D. C. Harrison, D. Burmester, W. K. Seah, and R. Rayudu. Busting myths of energy models for wireless sensor networks. *Electronics Letters*, 52(16):1412–1414, 2016.

[17] M. Magno, F. A. Aoudia, M. Gautier, O. Berder, and L. Benini. WU-LoRa: An Energy Efficient IoT End-Node for Energy Harvesting and Heterogeneous Communication. In *Proceedings of the Conference on Design, Automation & Test in Europe*, pages 1532–1537. European Design and Automation Association, 2017.

[18] 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.

[19] B. Martinez, M. Monton, I. Vilajosana, and J. D. Prades. The Power of Models: Modeling Power Consumption for IoT Devices. *IEEE Sensors Journal*, 15(10):5777–5789, 2015.

[20] P. R. Panda, B. Silpa, A. Shrivastava, and K. Gummidipudi. *Power-efficient System Design*. Springer Science & Business Media, 2010.