



OPEN Energy-efficient architecture for perception layer of IoT system

Kamarul Hafiz Kamaludin & Widad Ismail✉

This paper presents an energy-efficient power distribution mechanism for the sensor block in IoT nodes, which are typically battery-powered and designed to perform environmental sensing and periodic data transmission. While prior research has primarily focused on optimizing the radio module to minimize energy consumption, the continuous power supply to the sensor block—and the resulting leakage current—has received comparatively little attention. This oversight contributes to unnecessary energy loss and degrades network performance, including reduced data reliability and node availability. To address this challenge, an Adaptive Switching Mechanism (ASM) based on P-channel MOSFET high-side switching is proposed to dynamically control power delivery to the sensor block. Simulation results show that the proposed ASM can reduce energy consumption by up to 79.25% and improve runtime by 381.96% compared to conventional sensor node designs. Furthermore, the switching energy is significantly lower than that of low-side switching using N-channel MOSFETs. Experimental validation in a real-world application confirms these findings, demonstrating an energy reduction of up to 70.96% and a battery runtime improvement of approximately 244.31%.

Keywords Index terms, Internet of things (IoT), MOSFET, WSN, Microcontroller, Sensor

The Internet of Things (IoT) has become increasingly synonymous with modern lifestyles, particularly in domains that necessitate continuous event monitoring. The abbreviation “IoT” is now widely recognized as emblematic of contemporary technological integration, especially in applications requiring persistent observation and data acquisition. At its core, IoT encapsulates the convergence of physical objects, internet connectivity, and the automation of predefined operational tasks associated with these entities. IoT-enabled sensing systems are predominantly deployed in environments where precise and uninterrupted monitoring is imperative. Nevertheless, such settings frequently pose substantial challenges that impede direct human intervention, thereby necessitating autonomous solutions. To address these constraints, IoT devices are typically powered by battery modules and augmented with energy harvesting mechanisms, which collectively aim to prolong their functional lifespan.

The determination of battery capacity and the selection of appropriate energy harvesting strategies are inherently context-dependent, influenced by the specific operational scenario and the intended application of the IoT system. These considerations must encompass a range of critical parameters, including installation costs, maintenance intervals, requisite technical expertise, and the anticipated longevity of the device. Notably, the integration of energy-intensive electrochemical sensors, if not judiciously managed, may precipitate frequent operational interruptions, escalate maintenance expenditures, and compromise overall system reliability.

Electrochemical sensors can be broadly classified into several categories, including amperometric, potentiometric, impedimetric, photoelectrochemical, and electrogenerated chemiluminescence types^{1–5}. Among these, potentiometric sensors—particularly coated-wire electrodes (CWEs) and ion-selective electrodes (ISEs)—are extensively utilized in the aquaculture industry for continuous monitoring applications due to their sensitivity and selectivity^{2,4}.

In typical aquaculture deployments, electrochemical sensors require additional electrical isolator circuits to facilitate the simultaneous measurement of multiple parameters within a shared water resource. Given that these probes operate in aqueous environments, they are inherently vulnerable to electrical short-circuiting, which may lead to inaccurate micro-voltage readings or, in more severe cases, catastrophic failure of the entire computing system². Each isolator circuit generally consumes between 15 and 20 mA of current during operation, contributing to the overall power demand of the system³.

Energy efficiency has been identified as one of the five principal challenges within the broader IoT efficiency taxonomy⁴. Consequently, the architecture of IoT systems must be meticulously designed to accommodate the constraints of real-world operational environments. In aquaculture applications, the necessity for individual

Auto-ID Laboratory (AIDL), School of Electrical & Electronic Engineering, Universiti Sains Malaysia (USM), 14300 Nibong Tebal, Pulau Pinang, Malaysia. ✉email: eewidad@usm.my

electrical isolation per sensor significantly impacts the energy profile of the node. This is particularly problematic when sensor block circuits continue to draw power during low-power states, such as when the microcontroller and radio module enter sleep mode, thereby degrading the overall energy efficiency of the system^{3,4}.

The primary objective of this study is to advance the architectural design of the perception layer in conventional wireless IoT nodes by introducing an event-driven, energy-efficient power distribution framework for sensor blocks, referred to as the Adaptive Switching Mechanism (ASM). A secondary objective is to evaluate the current consumption behavior and the switching profile of the high-side P-channel MOSFET, including its transient response and energy performance.

The principal contributions of this research are delineated as follows:

- 1. Adaptive switching mechanism (ASM):** This study proposes a novel dynamic power supply control mechanism for sensor blocks, termed ASM, which utilizes a P-Channel MOSFET within the sensor node architecture. The mechanism addresses a critical yet frequently overlooked limitation in existing commercial sensor nodes—namely, the continuous power supply to sensor blocks regardless of operational state.
- 2. Energy efficiency and battery performance gains:** Through the implementation of the ASM, substantial improvements in energy efficiency and battery longevity were achieved. Simulation results demonstrate that the proposed mechanism can reduce energy consumption by up to 79.25% and improve runtime by 381.96% compared to conventional sensor-node designs. Real-world deployment results further validate these findings, confirming up to 70.96% energy savings and a 244.31% increase in battery runtime, thereby establishing the ASM as an effective and practical solution for low-power IoT applications.
- 3. Dual validation and practical applicability:** The dual validation approach—comprising both simulation and empirical field testing—confirms the robustness and practical viability of the proposed ASM framework. Its integration into IoT perception layer deployments significantly enhances node availability and ensures more reliable data transmission, thereby contributing to the long-term sustainability and operational resilience of IoT-based wireless sensor networks.

The remainder of this paper is structured as follows: "Related works" provides a review of related work concerning the characteristics and architectures of existing wireless IoT node. "Energy-efficient system solution" presents the energy-efficient system architecture, simulation and experimental setup. "Results and discussion" discusses the simulation and experiment-based results of the proposed system. Finally, "Conclusion" concludes the paper by summarizing the principal findings and highlighting their implications for future research in this field.

Related works

Recent advancements in IoT technologies have facilitated the development of water quality monitoring (WQM) systems that integrate multiple electrochemical sensors within a single wireless node. These systems aim to provide real-time, multi-parameter environmental data critical for aquaculture and freshwater resource management. However, a recurring limitation in existing designs is the lack of attention to energy efficiency, particularly concerning the continuous power supply to multiple sensor circuits.

Typical IoT-based aquaculture monitoring system

For instance, the system investigated in⁵ incorporates a suite of electrochemical sensors namely pH, Oxidation-Reduction Potential (ORP), Dissolved Oxygen (DO), and Total Dissolved Solids (TDS) such that all powered directly via a + 5 VDC output from a DC buck converter. While effective in sensing, this configuration lacks dynamic power control, resulting in persistent energy draw even during idle periods.

Similarly, the study presented in⁶ employs pH and DO sensors powered directly through a fixed power module, without the integration of a dedicated power management unit. This static power delivery approach contributes to unnecessary energy consumption, especially in scenarios where sensor activity is intermittent.

An Arduino-based WQM system described in⁷ utilizes pH and TDS sensors, with all electrochemical circuits powered by a 5 V regulator IC. The schematic diagram provided in the study confirms the absence of any switching or adaptive power control mechanisms, further reinforcing the trend of continuous power delivery in low-cost implementations.

The design approach outlined in⁸ mirrors that of⁵, integrating pH, ORP, DO, and electrical conductivity (EC) sensors into a solar-powered IoT node. Although the use of renewable energy is commendable, the system's reliance on a 5 V solar energy subsystem introduces limitations. As highlighted in⁹, energy harvesting becomes infeasible during night-time hours, thereby placing increased demand on the energy storage system and potentially compromising long-term operational reliability.

The schematic diagram presented in the study *Design and Implementation of a Full-Time Artificial Intelligence of Things-Based Water Quality Inspection and Prediction System for Intelligent Aquaculture*¹⁰ further exemplifies this trend. It demonstrates the integration of standard electrochemical sensors such as pH, DO, and ORP that are commonly employed in WQM applications. Notably, the schematic reveals that all sensor circuits are powered via a continuously supplied power source, without any adaptive control mechanism.

Moreover, the systems described in^{11–15} adopt design methodologies analogous to those in⁵ and⁷, wherein continuous power delivery to all electrochemical sensor circuits is facilitated through the onboard + 5 VDC supply integrated within the microcontroller circuit. This design choice, while straightforward, contributes to persistent energy consumption and limits the scalability of such systems in energy-constrained environments.

The study presented in¹⁶ provides quantitative insight into the power consumption of individual components within an Arduino-based WQM node. The microcontroller exhibited a current draw of 65.91 mA, while the pH and DO sensors consumed 23.16 mA and 43 mA, respectively. The system architecture diagram confirms that all

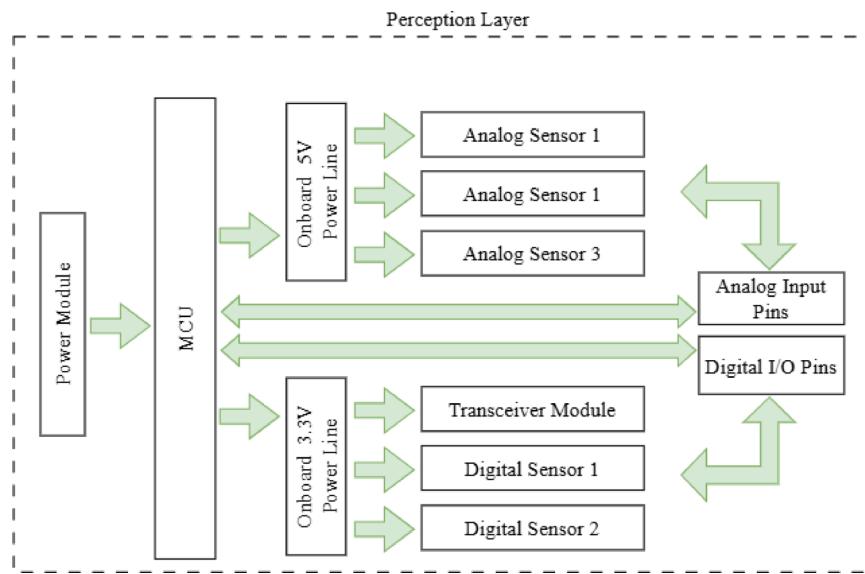


Fig. 1. Typical block diagram of IoT system architecture in aquaculture.

References	Authors/work	Electrochemical sensor	Continuous power leakage	Dedicated energy controller	Electrical isolator	Limitations
⁵	Tolentino et al.	pH, ORP, DO, TDS	+ 5 VDC on 4 electrochemical sensors	None	None	Multiple sensor integration increases energy consumption
⁶	Huan et al.	pH, DO	+ 3.3 VDC on 2 electrochemical sensors	None	None	Regulator IC continuously supplies + 3.3 VDC to all sensor blocks
⁷	Araneta et al.	pH, TDS	5 V on 2 electrochemical sensors	None	None	Inefficient energy consumption via direct 5 V DC regulator
⁸	Olmedo et al.	pH, ORP, DO, EC	5 V on 4 electrochemical sensors	Solar harvesting controller	None	Energy harvesting not feasible during nighttime hours [9]
¹⁰	Hu et al.	pH, ORP, DO	5 V on 3 electrochemical sensors	None	None	Lacks power management; contributes to overall inefficiency
¹¹	Duangwongsa et al.	pH, DO	+ 5 VDC on 2 electrochemical sensors	None	None	Direct power supply leads to inefficient energy utilization
¹²	Shete et al.	pH, DO	+ 5 VDC on 2 electrochemical sensors	None	None	Continuous energy supply causes inefficiency
¹³	Zaini et al.	pH, DO	+ 5 VDC on 2 electrochemical sensors	None	None	Direct energy utilization on all sensors
¹⁴	Anusha et al.	pH, TDS	+ 5 VDC on 2 electrochemical sensors	None	None	Absence of dedicated energy controller
¹⁵	Ajith et al.	pH, DO	+ 5 VDC on 2 electrochemical sensors	None	None	Continuous energy supply to multiple sensors degrades efficiency
¹⁶	KP et al.	pH, DO, Salinity, Ammonium	+ 5 VDC on 3 electrochemical sensors	Solar harvesting controller	None	Requires large battery capacity due to high energy consumption
¹⁷	Jais et al.	pH, DO, Salinity	+ 5 VDC on 3 electrochemical sensors	None	None	Continuous leakage current due to lack of energy controller

Table 1. Summary of related works.

electrochemical sensor circuits are directly interfaced with the onboard + 5 VDC supply of the Arduino UNO module, further reinforcing the absence of energy-aware design considerations.

Another study, titled *Improved Accuracy in IoT-Based Water Quality Monitoring for Aquaculture Tanks Using Low-Cost Sensors: Asian Seabass Fish Farming*¹⁷, adopts a similar design approach. Despite its focus on sensor accuracy, the system does not incorporate any explicit mechanism for energy-efficient control, thereby perpetuating the issue of continuous power draw.

Based on the aforementioned IoT-based aquaculture studies, a generalized block diagram representing the IoT system architecture for aquaculture monitoring featuring multiple electrochemical sensors for multi-parameter water quality assessment is presented in Fig. 1. This diagram serves as a reference model for identifying architectural gaps and motivating the development of energy-aware solutions such as the ASM proposed in this work.

The comparative evaluation summarized in Table 1 identifies a critical limitation across contemporary IoT node architectures employing multiple electrochemical sensors: the absence of dedicated energy management

and distribution mechanisms within the sensor subsystem. A survey of recent implementations reveals a pervasive omission of both power domain partitioning and electrical isolation for individual sensors. Although this omission is often rationalized by the use of a single sensing element to capture key electrochemical parameters, the lack of isolation introduces significant measurement inaccuracies under concurrent multi-sensor operation at same spot due to inter-sensor electrical interference.

Low power IoT-based monitoring system

As mentioned by¹⁸, energy inefficiency in IoT systems primarily arises from communication, data processing, and sensing components, with communication being the most energy-intensive. Consequently, recent research efforts have focused on optimizing network connectivity through energy-efficient mechanisms, such as adaptive sleep mode schemes, which minimize power consumption by allowing devices to remain inactive during idle periods while ensuring timely wake-up within defined duty cycles.

Building upon these energy-saving strategies, the study in¹⁹ presents an energy-efficient IoT system utilizing the EFM32GG microcontroller, which supports multiple low-energy modes to optimize battery usage. Key peripherals such as timers, real-time clocks, and communication interfaces remain active during low-power states, while transistor-based power gating further reduces energy consumption. In addition, a TPS63000 power management IC (PMIC) ensures stable voltage regulation from a 3.6 V, 35 Ah lithium primary cell. The Low-power Autonomous Module (LAM) operates at 20 mA under normal conditions and peaks at 50 mA during brief radio transmissions, enabling an estimated operational lifespan of approximately two months. This demonstrates the effectiveness of combining low-power microcontroller features with efficient power regulation for prolonged IoT device operation.

As demonstrated in²⁰, the proposed improved duty cycling algorithm effectively reduces energy consumption under special conditions, such as cloudy weather, where solar energy availability is limited. This adaptive strategy enables sensor nodes to maintain efficient operation even in suboptimal energy harvesting environments, thereby enhancing system resilience. In addition to energy-aware scheduling, the study introduces an efficient path selection mechanism based on the residual energy levels of individual nodes. By prioritizing communication routes with higher remaining energy, the algorithm balances power usage across the network and mitigates premature node failures. To validate the effectiveness of these techniques, simulations were conducted using Network Simulator 2 (NS2), yielding performance metrics that confirm notable improvements in energy efficiency, network reliability, and overall operational sustainability.

The study presented in²¹ proposed an NMOS-based power gating approach for regulating power distribution in sensor nodes, highlighting its efficiency and cost-effectiveness. However, a key limitation of this method is the inflexibility of its control logic such that any design errors necessitate hardware modifications, thus requiring high precision during the circuit design phase. In the study on power gating applications²², NMOS transistors demonstrated advantages in size and switching speed compared to PMOS transistors; however, PMOS transistors exhibited significantly lower leakage currents, making them more energy-efficient and better suited for ultra-low-power applications such as IoT sensor nodes.

In addition, NMOS transistors require additional components for high-side switching, such as a gate driver IC, which may increase the overall cost. In contrast, PMOS transistors can be used directly as high-side switches and can be triggered by standard 5 V microcontroller signals without the need for additional circuitry. Considering the complete high-side switching circuit, PMOS transistors typically consume less energy and offer lower implementation costs compared to their NMOS counterparts.

To further enhance energy sustainability, as highlighted in¹⁶, extending the battery life of WSN nodes can be achieved through energy harvesting techniques that utilize ambient energy sources. Among these, solar energy harvesting is particularly effective for outdoor deployments such as aquaculture sites, where sunlight is readily available. Moreover, photovoltaic cells offer a practical means of recharging batteries and are available in diverse sizes and configurations, enabling seamless integration into various system architectures. The incorporation of solar panels into water quality monitoring (WQM) systems supports continuous operation, minimizes the need for battery replacement, and reduces maintenance frequency and thereby lowering reliance on on-site technician interventions and enhancing system longevity.

While energy harvesting presents a promising strategy for extending the operational lifespan of IoT nodes, relying exclusively on ambient energy sources is not always practical. This limitation stems from the direct correlation between energy consumption and battery capacity sizing. In high-demand scenarios, larger battery capacities are required to compensate for intermittent energy availability, which in turn increases system cost, physical footprint, and maintenance complexity—particularly in remote or resource-constrained environments. To sustain prolonged operation, especially in energy-intensive deployments, larger battery capacities are required. However, this increase in battery size not only elevates the overall system cost but also leads to higher maintenance demands, thereby impacting the long-term viability of remote IoT node installations.

To address the limitations observed in conventional IoT node designs, the present study adopts a power gating strategy analogous to the technique proposed in¹⁹, wherein transistor-based switching is employed to regulate power delivery to the sensing blocks. In parallel, the duty-cycling algorithm introduced in²⁰ is also considered as part of this study, given its relevance to energy-efficient scheduling. Notably, both referenced approaches primarily focus on optimizing radio module protocols and network path selection to enhance energy efficiency. Building upon these foundations, the proposed system extends the conventional duty-cycling scheme by applying an event-driven power gating mechanism to the integrated sensing blocks. As a result, the proposed mechanism not only improves overall energy efficiency but also supports more precise battery capacity sizing. By aligning power usage with actual sensing activity, the system reduces reliance on oversized batteries, which

in turn lowers maintenance costs and enhances the long-term sustainability of IoT node deployments in energy-constrained environments.

Energy-efficient system solution

In alignment with energy-efficient design methodologies, this study introduces a customized architecture for the IoT perception layer that integrates a dedicated energy control mechanism. A proprietary expansion board (also known as shield board), referred to as USM Smart Aqua was developed to embed a P-channel MOSFET-based switching system, enabling dynamic high-side power control for each integrated electrochemical sensor and its corresponding electrical isolation circuitry.

To overcome the limitations of conventional duty cycling in IoT nodes, the USM Smart Aqua is proposed. This solution extends the sleep–wake signalling functionality of the radio module by interfacing it directly with its ASM control logic algorithm. Through this integration, ASM dynamically regulates power distribution to the sensing circuits based on the node's operational state. Consequently, sensor blocks are energized only during active communication phases, thereby reducing energy consumption during idle periods and improving overall system efficiency. The operation of the ASM is governed by the sleep–wake control signal of the ATmega328P microcontroller unit (MCU), enabling real-time power gating of the electrochemical sensor interface in synchronization with the activity of the radio communication module. This coordinated control strategy effectively minimizes quiescent current draw during inactive periods and significantly enhances the overall energy efficiency of the sensing subsystem. The following subsections provide a detailed explanation of the system architecture, simulation methodology, and experimental validation setup.

System architecture

The architecture of the enhanced IoT sensor node within perception layer, equipped with ASM operation is depicted in Fig. 2. This improved architecture is introduced to address the limitations identified in existing IoT-based sensor node designs, particularly the lack of dynamic energy control mechanisms. The proposed ASM, which integrates a P-channel MOSFET (IRF9540N), is designed to regulate power distribution from the onboard power source to the sensing circuits in a dynamic and event-driven manner. An energy-efficient controller algorithm is developed and embedded within the microcontroller to govern ASM operation at defined intervals, synchronized with the sleep–wake duty cycle of the communication module.

In this study, the Arduino Uno microcontroller is utilized as the sensor node platform, chosen for its simplicity, operational stability, and lower architectural complexity relative to alternatives such as the ESP32 and STM32. The platform also offers flexibility for future expansion, particularly in terms of analog sensor integration. Currently, only two analog input pins are in use—A0 for the pH sensor and A2 for the temperature sensor leaving three additional analog pins available for potential future applications. The DO sensor utilizes the I²C communication interface, thereby preserving the majority of digital pins for potential integration of additional digital sensors in future system expansions.

As concluded by²³, AVR microcontrollers are well-suited for applications that demand low power consumption and reliable performance, while offering a broad set of peripherals for general-purpose use. Their efficiency and simplicity make them ideal for battery-operated and embedded systems. The ESP32 stands out as a highly versatile microcontroller, featuring built-in Wi-Fi and Bluetooth connectivity, which makes it

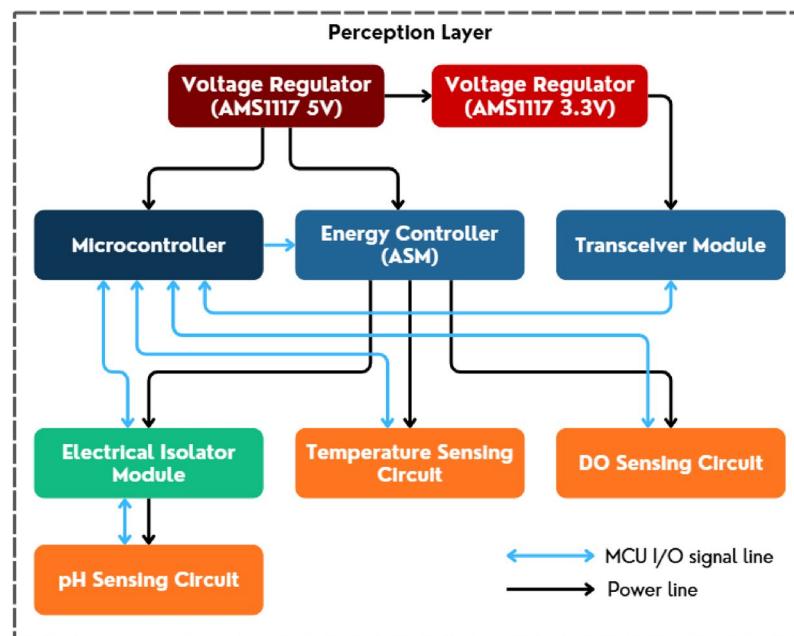


Fig. 2. Architecture of enhanced IoT sensor node with proposed ASM.

particularly attractive for Internet of Things (IoT) applications. In comparison, STM32 microcontrollers provide high performance, extensive peripheral support, and a high degree of integration, making them suitable for more complex and compute-intensive embedded systems.

Building upon this comparison, the Arduino Uno, based on the AVR microcontroller, offers a stable and relatively linear analog-to-digital converter (ADC). Its six analog input channels are consistently available, as the board lacks wireless functionality that could compromise ADC performance. In contrast, the ESP32's ADC exhibits notable non-linearity, particularly at the lower and upper voltage ranges, and is more prone to electrical noise. Therefore, for high-accuracy applications, the ESP32 often necessitates additional calibration or the use of an external ADC. Moreover, a critical limitation is that the ADC2 channels become unavailable when the Wi-Fi module is active. Additional constraints include the fact that some GPIOs (34–39) are input-only, while others (such as GPIOs 0, 2, and 15) serve as strapping pins, which, if improperly configured, may disrupt the boot process. Nevertheless, while STM32 microcontrollers may require a steeper learning curve due to more complex development environments, they offer significantly greater low-level control and configurability, making them highly suitable for advanced embedded applications.

On the other hand, the XBee Pro module is employed due to its ability to provide a simplified and homogeneous network architecture compared to the more complex Zigbee protocol. To establish a robust and scalable WSN platform suitable for real-world aquaculture environments, the DigiMesh protocol is utilized. As explained in²⁴, unlike Zigbee, DigiMesh eliminates the parent-child hierarchy by enabling all nodes to operate as routers, thereby ensuring a more uniform and resilient network. This mesh topology not only reduces latency but also simplifies routing, enhances fault tolerance, and allows for more flexible node placement with high scalability—making it highly beneficial for dynamic and distributed IoT deployments in aquaculture settings. The experiments conducted in this study demonstrated that DigiMesh networks exhibit higher throughput compared to traditional hierarchical protocols such as Zigbee. As a result, DigiMesh is particularly advantageous for applications requiring reliable and efficient data transmission, such as real-time monitoring in aquaculture environments.

In addition, the developed energy-efficient algorithm dynamically manages the sleep–wake cycles using an enhanced duty-cycling scheme implemented via ASM. Specifically, the algorithm disables power delivery to the sensing circuits during the transceiver's sleep mode by employing an enhancement-mode P-channel MOSFET as a power switch. The ASM is controlled through a low interrupt signal generated by the microcontroller, which activates or deactivates the MOSFET accordingly. When energized, this switching action connects the MOSFET's drain (+5 V) to the regulated 5 V supply line from the onboard voltage regulator, thereby enabling power delivery to the sensor circuits. The designated sensing circuit consists of a pH sensor (SEN0169 from DFRobot), a DO sensor (from Atlas Scientific), and a temperature sensor (DS18B20 from DFRobot). Using only two electrochemical sensors, the system requires just one electrical isolator module to prevent interference between the pH and DO circuits, enabling accurate simultaneous sensing. Figure 3 illustrates the energy-efficient algorithm flowchart of the enhanced IoT sensor node.

The ASM is designed to effectively disconnect the sensor circuitry during sleep mode, thereby eliminating leakage current and improving overall energy efficiency. Equation (1) is utilized to calculate the leakage current of the IoT node during idle, which will be analyzed in the subsequent Sect.

$$I_{leakTotal} = I_{leakMCU} + I_{leakTcvr} + \sum_{i=1}^N I_{leakSensor} \quad (1)$$

where $I_{leakTotal}$ denotes total leakage current in the IoT node system, $I_{leakMCU}$ denotes leakage current from the microcontroller, $I_{leakTcvr}$ denotes leakage current from the transceiver, $I_{leakSensor}$ denotes leakage current from the i^{th} sensor and N denotes number of sensors.

To ensure data accuracy during simultaneous sensing events, electrical isolator modules are applied to the pH sensing circuit. The power supplies for both the electrical isolator and the electrochemical sensors are dynamically controlled by the ASM in energy controller mechanism block. During operation, the ASM functions is calculated according to the control strategy defined in (2):

$$P_{sensor}(t) = f_{ASM}(S_{MCU}(t)) \quad (2)$$

where the ASM is modeled in (3) as a time-dependent digital control function, f_{ASM} , which governs the power state of the sensor block within the IoT sensor node.

$$f_{ASM}(S_{MCU}(t)) = \begin{cases} ON, & \text{if } S_{MCU}(t) = LOW \\ OFF, & \text{if } S_{MCU}(t) = HIGH \end{cases} \quad (3)$$

This function maps the MCU control signal, $S_{MCU}(t)$, to the corresponding sensor power state, $P_{sensor}(t)$. As described in Eq. (4), the system's operational mode is governed by the logic level of $S_{MCU}(t)$, where a logic HIGH indicates that the transceiver is in sleep mode, thereby placing the sensors into a low-power state. Conversely, a logic LOW corresponds to the transceiver's active (wake) state, which triggers the sensors to power on:

$$S_{MCU}(t) \in \{LOW, HIGH\} \quad (4)$$

The P-channel MOSFET within the ASM block is configured to supply power through a high-side switching mechanism. Most electrochemical sensors require either +5 VDC or +3 VDC for proper operation. The switching behavior of the P-MOSFET is governed by the state of $S_{MCU}(t)$, which controls the high-side switching action.

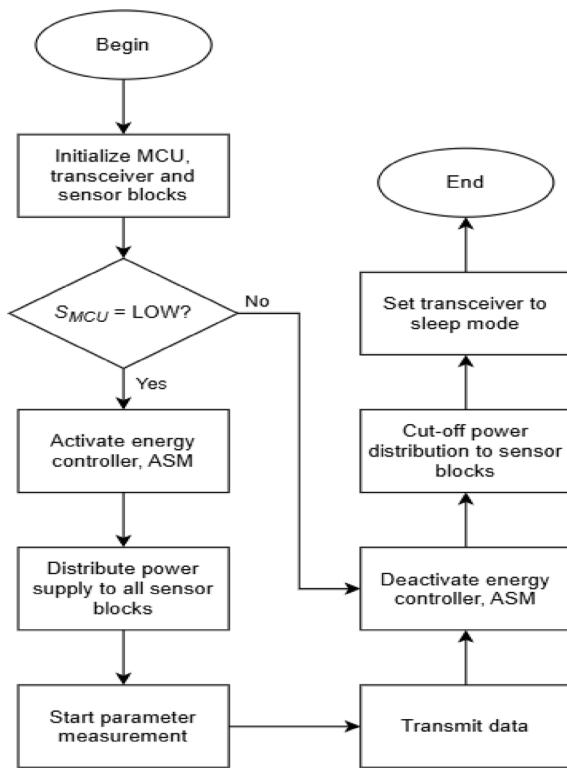


Fig. 3. Flowchart of the energy-efficient algorithm for enhanced IoT sensor node.

In this configuration, the source terminal of the P-channel MOSFET is connected to the onboard positive supply voltage of the IoT node, while the load is positioned between the drain and ground. This arrangement enables the load to be connected or disconnected from the power supply by modulating the gate voltage relative to the source.

Simulation setup

To validate the effectiveness of the proposed architecture, a power consumption test scheme was simulated using MATLAB. The primary objective of this simulation is to evaluate and compare the energy performance between a standard sensor node (StSN) and the proposed energy-efficient enhanced sensor node (EeSN). This comparative analysis provides insight into the energy-saving potential of the enhanced design under equivalent operating conditions. Table 2 presents the detailed parameters used in the simulation test bench for the power consumption analysis. Prior to executing the simulation setup, the total leakage current of the sensor node, denoted as $I_{leakTotal}$, was calculated using (1) from the previous subsection. The resulting value of $I_{leakTotal}$ was determined to be 373.6 mA, serving as a baseline reference for further evaluation of the system's power efficiency.

Both sensor node configurations were modelled using a common power source—an 8.4 V battery with a rated capacity of 4800 mAh—to ensure consistent testing conditions. Each node was equipped with a temperature sensor and two electrochemical sensors for monitoring pH and DO, with the pH sensor interfaced through a dedicated electrical isolator module. As only two electrochemical sensors are used, a single electrical isolator module is sufficient. This setup reflects real-world deployment scenarios where electrical isolation is required for reliable multi-parameter sensing in agricultural and aquaculture applications.

The baseline configuration, referred to as the Standard Sensor Node (StSN), excludes the ASM model and continuously supplies power to all sensing blocks. In this configuration, all sensing components remain active, except for the XBee Pro transceiver and the MCU, which enter sleep mode during predefined low-power intervals. In contrast, the Enhanced Energy-efficient Sensor Node (EeSN) incorporates the ASM within the simulation framework to dynamically regulate power delivery according to transceiver activity, thereby enhancing overall energy efficiency. During sleep mode, all sensing components, XBee Pro transceiver and the MCU in the EeSN are powered down. Note that the simulation was performed on an Arduino Uno R3; consequently, during power-down mode the node's typical current consumption remains at a minimum of 13.5 mA owing to the continuous operation of the onboard USB interface chip. The energy performance of the EeSN node was quantitatively assessed based on measured current consumption across all operational components, transmit power level and transmission modes.

To address the variability of operational conditions, the system's energy performance was evaluated under three representative duty-cycle scenarios: 8.33%, 16.67% and 25%. The duty cycle $duty_{cycle}$ was computed using

Variable parameter	Value
Sensor 1 current (pH), $i_{sensor1}$	23.16 mA ¹⁶
Sensor 2 current (DO), $i_{sensor2}$	43 mA ¹⁶
Sensor 3 current (temperature), $i_{sensor2}$	20.75 mA ¹⁶
XBee Pro Transmit current based on power level, i_{tx}	Low (60 mA), Middle (120 mA), High (215 mA) ²⁵
XBee Pro Idle current, i_{rx}	29 mA ²⁵
XBee Pro Sleep current, i_{sleep}	2.5 μ A ²⁵
XBee Pro data rate, $data_{rate}$	10 Kbp ²⁵
Packet Size, P_{size}	38 bytes
Packet Size to bits, bit_{value}	304 bits
XBee Pro transmit time, $t_{tx_calculated}$	0.03 s
MCU current (ATmega328P), i_{mcu}	9.2 mA ²⁶
MCU current (ATmega328P), i_{mcu_sleep}	66 μ A ²⁶
USB interface chip current (ATmega16U2), i_{usb}	13.5 mA ²⁷
Isolator current (Atlas Scientific), i_{iso}	20 mA ²⁸
Supplied voltage, V_{batt}	8.4 V
Battery capacity, C_{batt}	4800 mAh
Battery efficiency, η	0.9
Cycle period in 1 h, t_{total}	3600 s
Awake time, t_{awake}	t_{awake1} (300 s), t_{awake2} (600 s), t_{awake3} (900 s)
Sleep period, t_{sleep}	t_{sleep1} (3300 s), t_{sleep2} (3000 s), t_{sleep3} (2700 s)
Total cycle number, N_{cycle}	24
Duty cycle, $duty_{cycle}$	$duty_{cycle1}$ (8.33%), $duty_{cycle2}$ (16.67%), $duty_{cycle3}$ (25%)
Node density	10, 25, 50, 75, 100 nodes
Traffic load levels	Light / Medium / Heavy

Table 2. Matlab simulation test bench parameter.

$$duty_{cycle} = \frac{t_{awake}}{t_{awake} + t_{sleep}} \times 100 \quad (5)$$

where t_{awake} and t_{sleep} correspond to the node's active and idle (sleep mode) durations, respectively, within a one-hour cycle. The nominal configuration uses $t_{awake} = 300$ s and $t_{sleep} = 3300$ s, resulting in $duty_{cycle} = 8.33\%$. Two additional cases— $t_{awake} = 600$ s and $t_{awake} = 900$ s—were introduced to emulate increased sampling rates and more frequent transmissions, corresponding to 16.67% and 25% duty cycles.

For wireless communication, both sensor nodes utilize XBee Pro S3B transceivers. The ATmega328P microcontroller unit is adopted for this simulation, consistent with prior studies. The MCU is configured to remain continuously active for 300 s and turn into sleep for 3300 s. The transmission time is computed based on the ratio of packet size to transmission speed, as described in²⁹. Accordingly, the XBee Pro transmit time (s), $t_{tx_calculated}$ is calculated as follows:

$$t_{tx_calculated} = \frac{bit_{value}}{data_{rate}} \quad (6)$$

This simulation framework facilitates a comprehensive evaluation of the proposed energy control mechanism's effect on power consumption under realistic application scenarios. The overall simulation procedures are presented in the following Algorithm 1.

1:	Input: Variable parameters
	Output: Energy analysis results
2:	Begin
3:	Compute battery usable energy using (8)
4:	Compute power consumption for each component
5:	If node type is EeSN then
6:	Compute energy consumption per hour using (29)
7:	Else if node type is StSN then
8:	Compute energy consumption per hour using (30)
9:	End if
10:	Accumulate total energy
11:	Compute estimated runtime using (31), (32)
12:	Output results
13:	End

Algorithm 1. Simulation of energy analysis.

At the beginning of simulation, the battery energy capacity E_{bat_total} is first converted from milliamper-hours (mAh) to watt-hours (Wh) using the relation

$$E_{bat_total} = \left(\frac{C_{batt}}{1000} \right) \times V_{batt} \quad (7)$$

where C_{batt} and V_{batt} are battery capacity and supplied voltage, respectively. Considering an efficiency factor $\eta = 0.9$, the usable energy E_{bat_usable} is given by:

$$E_{bat_usable} = E_{bat_total} \times \eta \quad (8)$$

Assuming the system is active for $t_{awake} = 300$ s and $t_{total} = 3600$ s, the sleep duration is calculated as:

$$t_{sleep} = t_{total} - t_{awake} \quad (9)$$

Thus, for hourly energy analysis, the $t_{sleep} = 3300$ s. The following parameters are converted to hours for subsequent energy calculations:

$$t_{awake_hr} = \left(\frac{t_{awake}}{t_{total}} \right) \quad (10)$$

$$t_{sleep_hr} = \left(\frac{t_{sleep}}{t_{total}} \right) \quad (11)$$

$$t_{tx_hr} = \left(\frac{t_{tx_calculated}}{t_{total}} \right) \quad (12)$$

The remaining energy consumption, P for each system component is calculated using the following equation:

$$P = I \times V_{batt} \quad (13)$$

where I is the current drawn by the component, and V_{batt} is the battery voltage. This calculation provides the power consumption values for each component, specifically $P_{sensor1}$, $P_{sensor2}$, $P_{sensor3}$, P_{mcu} , P_{mcu_sleep} , P_{usb} , P_{tx} , P_{rx} , P_{sleep} and $P_{isolator}$. Subsequently, the hourly energy consumption, E_{hourly} of each aforementioned component in both the StSN and EeSN systems is calculated using (14), based on their respective operating durations. This enables a detailed comparison of the energy consumption profiles of the two system configurations.

$$E_{hourly} = P \times t_{hr} \quad (14)$$

The value of t_{hr} is substituted by using the results of (10), (11), (12) respectively, as the following:

$$E_{EeSN_sensor1} = P_{sensor1} \times t_{awake_hr} \quad (15)$$

$$E_{StSN_sensor1} = P_{sensor1} \times t_{total} \quad (16)$$

$$E_{EeSN_sensor2} = P_{sensor2} \times t_{awake_hr} \quad (17)$$

$$E_{\text{StSN_sensor2}} = P_{\text{sensor2}} \times t_{\text{total}} \quad (18)$$

$$E_{\text{EeSN_sensor3}} = P_{\text{sensor3}} \times t_{\text{awake_hr}} \quad (19)$$

$$E_{\text{StSN_sensor3}} = P_{\text{sensor3}} \times t_{\text{total}} \quad (20)$$

$$E_{\text{mcu}} = P_{\text{mcu}} \times t_{\text{awake_hr}} \quad (21)$$

$$E_{\text{mcu_sleep}} = P_{\text{mcu_sleep}} \times t_{\text{sleep_hr}} \quad (22)$$

$$E_{\text{usb}} = P_{\text{usb}} \times t_{\text{total}} \quad (23)$$

$$E_{\text{tx}} = P_{\text{tx}} \times t_{\text{tx_hr}} \quad (24)$$

$$E_{\text{rx}} = P_{\text{rx}} \times t_{\text{awake_hr}} \quad (25)$$

$$E_{\text{sleep}} = P_{\text{sleep}} \times t_{\text{sleep_hr}} \quad (26)$$

$$E_{\text{EeSN_iso}} = P_{\text{isolator}} \times t_{\text{awake_hr}} \quad (27)$$

$$E_{\text{StSN_iso}} = P_{\text{isolator}} \times t_{\text{total}} \quad (28)$$

$$\begin{aligned} E_{\text{total_EeSN}} = & E_{\text{EeSN_sensor1}} + E_{\text{EeSN_sensor2}} + E_{\text{EeSN_sensor3}} \\ & + E_{\text{mcu}} + E_{\text{mcu_sleep}} + E_{\text{usb}} + E_{\text{tx}} + E_{\text{rx}} + E_{\text{sleep}} + E_{\text{EeSN_iso}} \end{aligned} \quad (29)$$

Then, the total energy consumption per hour of sensor nodes is computed as the follows:

$$\begin{aligned} E_{\text{total_StSN}} = & E_{\text{StSN_sensor1}} + E_{\text{StSN_sensor2}} + E_{\text{StSN_sensor3}} \\ & + E_{\text{mcu}} + E_{\text{mcu_sleep}} + E_{\text{usb}} + E_{\text{tx}} + E_{\text{rx}} + E_{\text{sleep}} + E_{\text{StSN_iso}} \end{aligned} \quad (30)$$

As a final step in the node energy analysis, the estimated runtime t_{runtime} (in hour) of the sensor nodes is computed using the following equations:

$$t_{\text{runtime_EeSN}} = \frac{E_{\text{bat_usable}}}{E_{\text{total_EeSN}}} \quad (31)$$

$$t_{\text{runtime_StSN}} = \frac{E_{\text{bat_usable}}}{E_{\text{total_StSN}}} \quad (32)$$

The subsequent analysis investigates the energy-efficient performance of the network. The simulations are based on an energy model for XBee Pro nodes powered by an 8.4 V, 4800 mAh battery with 0.9 efficiency. Each node's energy consumption depends on its operating mode: transmit (TX) at maximum current, receive (RX) and idle/standby at 29 mA, and deep sleep at 0.0025 mA.

For the lifetime simulation, the goal is to compute the time until the first node dies. The network lifetime is estimated using.

$$t_{\text{node}} = \frac{E_{\text{bat_usable}}}{P_{\text{avg}}} \quad (33)$$

where P_{avg} is the average power consumption across all modes:

$$P_{\text{avg}} = \frac{V(i_{\text{tx}}t_{\text{tx}} + i_{\text{rx}}t_{\text{rx}} + i_{\text{idle}}t_{\text{idle}})}{t_{\text{total}}} \quad (34)$$

In retransmission simulations, the extra energy caused by packet retries is calculated as.

$$E_{\text{retrans}} = r_{\text{mean}} \times N_{\text{packet}} \times E_{\text{tx_per_packet}} \quad (35)$$

where r_{mean} is the mean number of retries per packet and $E_{\text{tx_per_packet}}$ is the energy consumed per transmission.

The node density simulation studies how network lifetime scales with the number of nodes. Networks with 10, 25, 50, 75, and 100 nodes are simulated, and the lifetime is measured as the time until the first node dies. Assuming traffic per node is constant, the lifetime per node remains

$$t_{\text{node}} = \frac{E_{\text{bat_usable}}}{P_{\text{avg}}} \quad (36)$$

For traffic load analysis, different traffic levels (light, medium, heavy) are simulated by defining packet rates for each level. The total transmit and receive time is computed as.

$$t_{\text{tx, total}} = N_{\text{packet}} \times t_{\text{tx_per_packet}} \quad (36)$$

$$t_{rx,total} = N_{packet} \times t_{rx_per_packet} \quad (37)$$

and the energy consumption for each load is using.

$$E_{traffic} = V (i_{tx} t_{tx,total} + i_{rx} t_{rx,total}) \quad (38)$$

In summary, the simulation workflow starts by defining node parameters, including currents, voltage, battery, and efficiency. Baseline energy consumption and lifetime are computed first, followed by simulations of retransmissions, node density scaling, and traffic load effects. The outputs are then combined to analyze network lifetime, energy cost, and scalability, providing a quantitative framework to evaluate system viability under different operating conditions. Algorithm 2 details the procedural steps used in this analysis.

1:	Input: V_{batt} , C_{batt} , η , i_{tx} , i_{rx} , i_{sleep} , $duty_{cycle}$, $node_densities = [10, 25, 50, 75, 100]$, $traffic_levels = [\text{light, medium, heavy}]$
	Output: $lifetime_node$, $E_retrans$, $E_traffic$, $lifetime_vs_density$
2:	Begin
3:	Compute battery usable energy using (8)
4:	Compute baseline node lifetime using (33), (34)
5:	For each node do
6:	Compute t_{node} , P_{avg}
7:	End For
8:	Retransmission energy calculation using (35)
9:	For each packet do
10:	Retries \leftarrow random_retry_rate()
11:	Compute $E_{retrans}$
12:	End For
13:	Node density analysis
14:	For each density in $node_densities$ do
15:	Simulate_network(density)
16:	Compute_lifetime_per_node()
17:	Store_network_lifetime()
18:	End For
19:	Traffic load analysis calculation using (36), (37), (38)
20:	For each traffic in $traffic_levels$ do
21:	$N_{packets} \leftarrow$ get_packet_count(traffic)
22:	Compute $t_{tx,total}$, $t_{rx,total}$, $E_{traffic}$
23:	End For
24:	Generate reports
	End

Algorithm 2 . Simulation of energy-efficient performance of the network.

The investigation of the ASM is further extended through simulations evaluating key MOSFET performance metrics, including transient response, switching losses, conduction characteristics, and leakage current. The analysis focuses on P-channel (IRF9540N) and N-channel (IRFZ44N) MOSFETs. Circuit-level simulations were conducted using LTspice to model and assess the ASM performance with MOSFETs as the core switching elements. Manufacturer-provided SPICE models, as referenced in^{30,31}, were utilized to ensure realistic device behavior and accurate simulation outcomes.

In the simulation setup, the IRF9540N P-channel MOSFET (PMOS) is configured as a high-side switch, with its source terminal connected to a 5 V supply (VBUS) and its drain connected to a 100Ω resistive load. The gate is driven by a pulsed voltage signal alternating between 0 V and 5 V, emulating digital control logic for switching the device on and off. A 100Ω resistor is placed at the gate to model the driver's output impedance, while a $10\text{ k}\Omega$ pull-up resistor ensures proper gate-source voltage biasing when the control signal is inactive.

Conversely, the IRFZ44N N-channel MOSFET (NMOS) is configured as a low-side switch, with its source terminal connected to ground and its drain connected to a 100Ω resistive load. The gate is also driven by a pulsed voltage signal alternating between 0 V and 5 V. A 100Ω gate resistor models the driver's output impedance, and a $10\text{ k}\Omega$ pull-down resistor ensures correct gate-source voltage biasing when the control signal is inactive. In this analysis, higher efficiency is anticipated compared to the PMOS configuration. However, high-side switching is

Variable parameter	NMOS value / description	NMOS value / description
SPICE Model	irfz44.lib	irf9540nslpbf.lib
VBUS	5 V (Supply voltage)	5 V (Supply voltage)
RLOAD	100 Ω (Load resistance)	100 Ω (Load resistance)
RGATE	100 Ω (Gate resistor)	100 Ω (Gate resistor)
RPULL	10k Ω (Pull-down to GND)	10k Ω (Pull-up to source)
RPAR	10 mΩ (Drain-source parasitic resistance)	10 mΩ (Drain-source parasitic resistance)
C_{par}	10 pF (Drain-source parasitic capacitance)	10 pF (Drain-source parasitic capacitance)
Gate Logic	0 V = OFF, 5 V = ON	0 V = ON, 5 V = OFF
MOSFET Model	X1 D_MOS VGATE 0 irfz44n	X1 D_MOS VGATE VS irf9540nslpbf
Simulation Directive	.tran 0 5 m 0 50u	.tran 0 5 m 0 50u

Table 3. LTspice test bench parameter.

Metric category	LTspice directive parameter
Turn-on delay	t_{d_on} (s), $t_{d_on_ns}$ (ns)
Rise time	t_{rise} (s), t_{r_ns} (ns)
Turn-off delay	t_{d_off} (s), $t_{d_off_ns}$ (ns)
Fall time	t_{fall} (s), t_{f_ns} (ns)
Switching losses	E_{on} (J), E_{on_uj} (μJ), E_{off} (J), E_{off_uj} (μJ)
Conduction metrics	R_{on_est} (Ω), $R_{on_est_mOhm}$ (mΩ), V_{drop_avg} (V), P_{cond} (W), P_{cond_mW} (mW)
Leakage current	I_{leak_avg} (A), $I_{leak_μ A}$

Table 4. LTspice measurement.

not considered in the NMOS transient analysis, as it requires additional components such as a gate driver IC which increase both circuit complexity and overall energy consumption. The full set of configuration parameters used for this analysis is summarized in Table 3, while Table 4 details the specific measurement parameters.

To quantify the switching response of the MOSFETs, the turn-on delay (t_{d_on}) is defined based on gate and drain voltage transitions, as observed in LTspice transient simulations. For the NMOS, the turn-on delay is determined as the time difference between:

- Trigger: Gate voltage (VG) rising to 90% of the supply voltage (0.9 • VBUS).
- Target: Internal drain voltage (V_{D_int}) falling to 10% of the supply voltage (0.1 • VBUS).

$$t_{d_on}^{(NMOS)} = t_{VD_int} = 0.1 \bullet VBUS, fall - t_{VG} = 0.9 \bullet VBUS, rise \quad (39)$$

For the PMOS, which uses an inverted gate signal, the turn-on delay is defined as the time between:

- Trigger: Gate voltage (VG) rising to 10% of the supply voltage (0.1 • VBUS).
- Target: Drain voltage (V_D) falling to 90% of the supply voltage (0.9 • VBUS).

$$t_{d_on}^{(PMOS)} = t_{VD} = 0.9 \bullet VBUS, fall - t_{VG} = 0.1 \bullet VBUS, rise \quad (40)$$

The rise time (t_{rise}) is defined as the interval during which the drain voltage transitions from 90% to 10% of the supply voltage (VBUS) during the MOSFET turn-on phase. The NMOS rise time is determined from the drain voltage waveform as it falls during the turn-on event. Specifically:

- Trigger: $V_{D_int} = 0.9 \bullet VBUS$ (falling edge)
- Target: $V_{D_int} = 0.1 \bullet VBUS$ (falling edge)

The rise time is calculated as:

$$t_{rise}^{(NMOS)} = t_{VD_int} = 0.9 \bullet VBUS, fall - t_{VD_int} = 0.1 \bullet VBUS, fall \quad (41)$$

This represents the time interval during which the internal drain voltage drops from 90% to 10% of the supply voltage. For PMOS, the rise time is measured using (36), analogously using the drain voltage (V_D):

- Trigger: $V_D = 0.9 \bullet VBUS$ (falling edge)

Target: $V_D = 0.1 \bullet VBUS$ (*falling edge*)

$$t_{rise}^{(PMOS)} = t_{VD} = 0.9 \bullet VBUS, fall - t_{VD} = 0.1 \bullet VBUS, fall \quad (42)$$

The turn-off delay (t_{d_off}) is defined as the time interval between the gate voltage falling below a defined threshold and the corresponding recovery of the drain voltage to its high level. For NMOS, the turn-off delay is determined using (43) based on:

- Trigger: Gate voltage (VG) falling to 10% of the supply voltage ($0.1 \bullet VBUS$).
- Target: Internal drain voltage (V_{D_int}) rising to 90% of the supply voltage ($0.9 \bullet VBUS$).

$$t_{d_off}^{(NMOS)} = t_{VD_int} = 0.9 \bullet VBUS, rise - t_{VG} = 0.1 \bullet VBUS, fall \quad (43)$$

The PMOS turn-off delay is expressed as (44) and measured using the following conditions:

- Trigger: Gate voltage (VG) falling to 90% of the supply voltage ($0.9 \bullet VBUS$).
- Target: Drain voltage (V_D) rising to 10% of the supply voltage ($0.1 \bullet VBUS$).

$$t_{d_off}^{(PMOS)} = t_{VD} = 0.1 \bullet VBUS, rise - t_{VG} = 0.9 \bullet VBUS, fall \quad (44)$$

The fall time (t_{fall}) in this context refers to the duration of the low-to-high transition of the drain voltage during the MOSFET turn-off event. Despite the naming, this corresponds to the signal rising from 10% to 90% of the supply voltage, as the MOSFET ceases conduction and the drain returns to its high state. The NMOS fall time is measured based on the internal drain voltage (V_{D_int}) as follows:

- Trigger: $V_{D_int} = 0.1 \bullet VBUS$ (*rising edge*)
- Target: $V_{D_int} = 0.9 \bullet VBUS$ (*rising edge*)

The NMOS fall time is calculated using the expression:

$$t_{fall}^{(NMOS)} = t_{VD_int} = 0.9 \bullet VBUS, rise - t_{VD_int} = 0.1 \bullet VBUS, rise \quad (45)$$

This interval captures the time taken for the drain voltage to return to its high level during the MOSFET's turn-off process. The same threshold levels are applied to the external drain voltage (V_D) for PMOS fall time measurement using (45):

- Trigger: $V_D = 0.1 \bullet VBUS$ (*rising edge*)
- Target: $V_D = 0.9 \bullet VBUS$ (*rising edge*)

$$t_{fall}^{(PMOS)} = t_{VD} = 0.9 \bullet VBUS, rise - t_{VD} = 0.1 \bullet VBUS, rise \quad (46)$$

Switching energy losses quantify the transient power dissipation that occurs during the MOSFET's transition between on and off states. These losses are determined by integrating the instantaneous power over the respective switching intervals. The switching energy is split into:

- Turn-on energy (E_{on}) - measured during the MOSFET's activation period, and.
- Turn-off energy (E_{off}) - measured during the deactivation period.

The instantaneous power dissipation of NMOS is defined as:

$$\mathcal{P}_{NMOS}(t) = v_{ds}(t) \bullet i_{ds}(t) = V(V_{D_int}) \bullet I(V_{dsense}) \quad (47)$$

where $v_{ds}(t)$ is the drain-source voltage across the MOSFET, and $i_{ds}(t)$ is the drain current measured via the shunt sense path.

- Turn-on energy is computed by integrating the power over the interval between the turn-on delay and the end of the rise time:

$$E_{on}^{(NMOS)} = \int_{t_{d_on}}^{t_{d_on} + t_{rise}} \mathcal{P}_{NMOS}(t) dt \quad (48)$$

- Turn-off energy is similarly calculated over the fall interval:

$$E_{off}^{(NMOS)} = \int_{t_{d_off}}^{t_{d_off} + t_{fall}} \mathcal{P}_{NMOS}(t) dt \quad (49)$$

The instantaneous power dissipation of PMOS accounts for the polarity of the current during P-channel conduction, defined as:

$$\mathcal{P}_{PMOS}(t) = V(V_S, V_D) \bullet [-I(V_{dsense})] \quad (50)$$

The negative sign ensures that the power remains positive when current flows from source to drain.

- Turn-on energy for the PMOS is calculated as:

$$E_{on}^{(PMOS)} = \int_{t_{d_on}}^{t_{d_on} + t_{rise}} \mathcal{P}_{PMOS}(t) dt \quad (51)$$

- Turn-off energy is determined by:

$$E_{off}^{(PMOS)} = \int_{t_{d_off}}^{t_{d_off} + t_{fall}} \mathcal{P}_{PMOS}(t) dt \quad (52)$$

The conduction characteristics of the MOSFETs were evaluated using time-averaged metrics computed over a steady-state interval, specifically from $t_1 = 200 \mu s$ to $t_2 = 900 \mu s$, during which the device remained fully on. The effective on-resistance (R_{on_est}) is calculated as the average ratio of voltage drop to load current over the defined interval:

$$R_{on_est} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \frac{v_{drop}(t)}{i_{load}(t)} dt \quad (53)$$

where NMOS is assigned with:

$$v_{drop}(t) = V(V_{D_int}) \\ i_{load}(t) = I(V_{dsense})$$

and the PMOS is assigned with:

$$v_{drop}(t) = V(V_S, V_D) \\ i_{load}(t) = I(R_{load})$$

The average voltage drop across the MOSFET during conduction is given by:

$$V_{drop_avg} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} v_{drop}(t) dt \quad (54)$$

The average conduction power loss is computed as the time-averaged product of drain-source voltage and current:

$$P_{cond} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} v_{ds}(t) \bullet i_{ds}(t) dt \quad (55)$$

For NMOS:

$$v_{ds}(t) = V(V_{D_int}) \\ i_{ds}(t) = I(V_{dsense})$$

For PMOS:

$$v_{ds}(t) = V(V_S, V_D)$$

$$i_{ds}(t) = -I(V_{dsense})$$

Finally, the leakage current was measured during the OFF-state interval of each MOSFET to evaluate subthreshold conduction and drain-source leakage under zero gate drive. The selected measurement window for both NMOS and PMOS was from $t_1 = 1.1\text{ ms}$ to $t_2 = 1.9\text{ ms}$, a period during which the gate voltage is held high (for NMOS) or low (for PMOS), fully turning the device off.

The average leakage current of NMOS is computed as:

$$I_{leak_avg}^{(NMOS)} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} i_{ds}(t) dt \quad (56)$$

where $i_{ds}(t) = I(V_{dsense})$. The PMOS average leakage current is measured using the negative of the sensed current, since conduction direction is reversed:

$$I_{leak_avg}^{(PMOS)} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} -i_{ds}(t) dt \quad (57)$$

Experimental setup and implementation

The experimental setup was carefully designed to replicate real-world requirements and environmental conditions typical of aquaculture practices. The key water quality parameters monitored, namely pH, DO, and temperature, were selected to complement the improved architecture presented in the earlier section, as illustrated in Fig. 2. The sensor node was configured to operate in two distinct modes: active mode for data acquisition and transmission, and sleep mode for conserving energy during idle periods. This dual-mode operation supports the enhanced duty-cycling strategy outlined in earlier sections. The conducted experiments were designed to validate the hypothesis that energy consumption in IoT-based sensor nodes can be significantly reduced by dynamically regulating power distribution to the sensing block by employing ASM. The measured current components include I_{mcu} , I_{mcu_sleep} , $I_{sensor1}$, $I_{sensor2}$, $I_{sensor3}$, I_{iso} , I_{tx} , I_{rx} and I_{sleep} . The USB interface current, I_{usb} , is assumed to be 13.5 mA²⁷, as the USB interface chip (ATmega16U2) is integrated into the Arduino Uno R3 design.

Low-side current sensing is preferred for measuring very small currents, such as those drawn by the XBee Pro S3B transceiver during sleep mode, because the shunt voltage remains close to 0 V and amplifier offset errors are minimal. In contrast, high-side sensing is necessary when ground continuity must be preserved, but it becomes less accurate at microampere levels due to increased noise and offset. For example, with a 10 μA least significant bit (LSB) and a 0.1 Ω shunt resistor, low-side sensing provides better resolution. High-side configurations under these conditions require careful handling of common-mode rejection, supply noise, and offset calibration. Therefore, the INA231 current sensor is configured in a low-side setup to accurately capture the XBee Pro S3B's sleep current, which is exceptionally low. This configuration ensures reliable energy analysis in ultra-low power scenarios. The supplied voltage is maintained at 3.3 V, consistent with prior studies³², where the XBee Pro's sleep current is measured at 0.6 mA. As the next step, the average idle current for both sensor nodes are measured.

The sink node was positioned on a table approximately 1 m above ground level and located 10 m from the deployed sensor nodes. Each sensor node transmitted data packets using an XBee Pro transceiver, with a maximum payload size of 38 bytes. The data packet structure comprises 7 bytes for an authentication code, 4 bytes for the node name, and 7 bytes for delimiters (denoted by the symbol '\$') used to separate individual data fields. Each sensor reading is allocated up to 5 bytes. An example of a transmitted packet is: "\$Test123\$1001\$19.59\$13.84\$30.19\$0.00\$".

The pH 7 buffer solution was used as the test medium during the experimental procedures. This choice reflects typical environmental conditions and ensures consistent baseline measurements for evaluating sensor accuracy and system performance under practical deployment scenarios. The sensor nodes operated in a 60-second duty cycle, comprising 30 s of active sensing and 30 s of sleep. During each active phase, 100 measurements were taken and averaged to generate one representative sample of water quality data. Each sensor was captured over 20 independent samples to ensure statistical reliability and to account for potential variability in sensor readings and system performance. All tests were conducted under controlled indoor conditions at the Auto-ID Laboratory (AIDL), Universiti Sains Malaysia. Tables 5 and VI provide a comprehensive overview of the test-bench parameters and the equipment used in the experimental setup. Several parameters are partially preconfigured, as previously listed in Table 5.

Leveraging the system architecture outlined earlier, we developed an IoT shield board (USM Smart Aqua) for a wireless sensor node featuring energy-efficient capabilities. This shield board is compatible with any Atmel ATmega328P microcontroller module based on the Arduino Uno platform. As shown in Fig. 4, the double layer PCB layout was designed specifically for the monitoring of key aquaculture parameters in these experiments.

The sensor node, referred to as the Enhanced Energy-efficient Sensor Node (EeSN), is equipped with the proposed shield board that integrates the energy-saving features outlined in the previous sections. In contrast, the Standard Sensor Node (StSN) represents an unmodified Arduino Uno-based sensor node, serving as the baseline for comparison. To evaluate and compare power performance, current consumption was initially measured by monitoring the voltage drop across a shunt resistor using a high-precision current sensor (INA231, 25 μA resolution), combined with time-duration capture via a digital oscilloscope. To ensure consistency and

Item description	Specification
Sensor node platform	Arduino Uno-based microcontroller
Proposed shield board	USM Smart Aqua
Transceiver	XBee Pro S3B (900 MHz ISM band)
Network topology	DigiMesh
Transceiver operating mode	API 1 (without escaped characters)
Transceiver power level	PL = 4
Transceiver sleep mode	SM = 1
Sensor node power source	8.4 V, 4800 mAh, battery-powered
Electrochemical sensor 1	DFRobot SEN0169-V2 pH sensor
Electrochemical sensor 2	Atlas Scientific EZO DO sensor kits
Auxiliary sensor 3	DFRobot DS18B20 Temperature sensor
Electrical isolation 1	DFRobot Analog signal isolator
Active duration (verification measurement)	$t_{active} = 30$ s
Active duration (real-world)	$t_{active} = 300$ s
Sleep duration (verification measurement)	$t_{sleep} = 30$ s
Sleep duration (real-world)	$t_{sleep} = 3300$ s
Total sampling number	20 cycles
Calibration for pH	pH 7 Buffer Solution
Calibration for DO	Zero Dissolved Oxygen Solution

Table 5. Experimental tests bench parameter.

Measuring equipment	Model
Digital oscilloscope	Agilent technologies DSO-X 2002 A
Digital multimeter	FLUKE 115 True RMS
Current sensor	INA231

Table 6. List of used equipment for experimental setup.

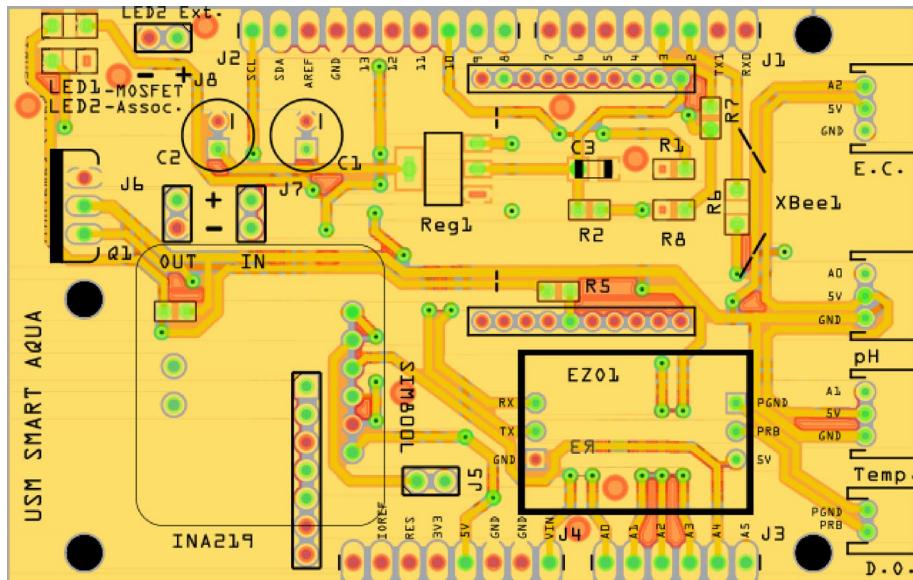


Fig. 4. PCB layout design of SmartAqua.



Fig. 5. Current consumption measurement.

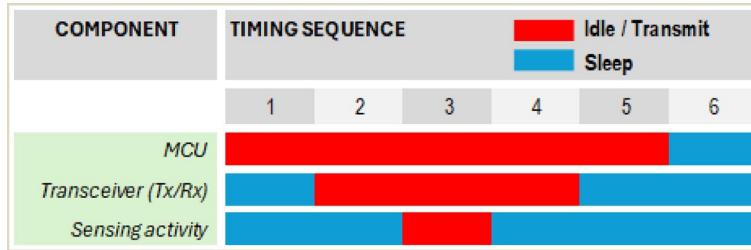


Fig. 6. Operational sequence of each sensor node component.

measurement reliability, the results were further validated using a FLUKE 115 True RMS digital multimeter (DMM), as illustrated in Fig. 5.

In this experimental setup, low power sleep mode is enabled on the MCU based on the operational conditions of both the EeSN and StSN systems. The entire system is designed to enter full sleep mode during designated sleep periods to optimize energy efficiency. To reflect real-world operating conditions, the system follows a duty cycle comprising 300 s of active time and 3300 s of sleep time. Figure 6 illustrates the operational sequence of each sensor node component during the energy consumption analysis.

As illustrated in the following Fig. 7, all sensors are positioned within a single measurement vessel containing a pH 7 buffer solution to replicate the real-world deployment conditions in a fishpond.

The average active current consumption over 20 measurement cycles is then calculated by current sensor using Ohm's Law, as expressed in (58), based on the measured voltage drop V_{shunt} across the shunt resistor R_{shunt} :

$$I_{shunt} = \frac{V_{shunt}}{R_{shunt}} \quad (58)$$

Using (58), the individual current components, I_{mcu} , I_{mcu_sleep} , $I_{sensor1}$, $I_{sensor2}$, $I_{sensor3}$, I_{iso} , I_{tx} , I_{rx} and I_{sleep} are calculated and summed. This total is then combined with the USB interface chip current (13.5 mA²⁷, I_{usb}) to obtain the overall active current, I_{active} . The corresponding active energy per hour E_{active} is then calculated using (59) as follows:

$$E_{active} = \frac{V_{batt} \times I_{active} \times t_{active}}{3600} \quad (59)$$

where $t_{active} = 300$ denotes the active duration (in seconds) within one hour. Then, the computation of sleep energy per hour, denoted as E_{sleep} , is carried out using (60):

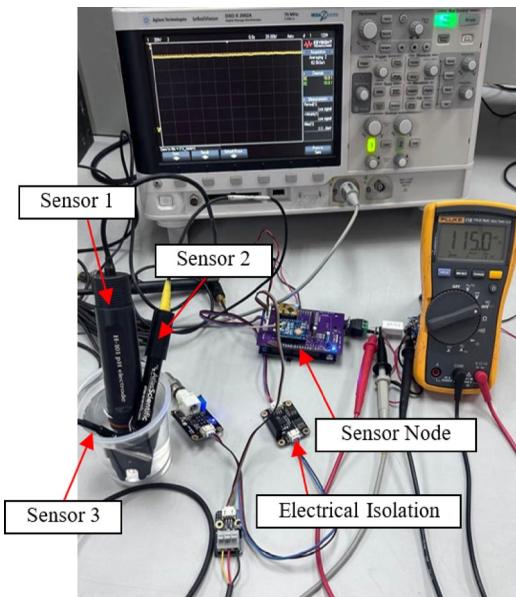


Fig. 7. Electrochemical sensors placement.

$$E_{sleep} = \frac{V_{batt} \times I_{sleep} \times t_{sleep}}{3600} \quad (60)$$

where I_{sleep} is calculated using (61) and $t_{sleep} = 3300$, denotes the sleep duration (in seconds) within one hour. To this end, the average energy consumption per hour for both EeSN and StSN sensor nodes is calculated using (61):

$$E_{total} = E_{active} + E_{sleep} \quad (61)$$

The percentage values for reduction, improvement, and efficiency are calculated using (62), (63), and (64), respectively, as follows:

$$\text{Reduction (\%)} = \frac{E_{total_StSN} - E_{total_EeSN}}{E_{total_StSN}} \times 100 \quad (62)$$

$$\text{Improvement (\%)} = \frac{t_{runtime_EeSN} - t_{runtime_StSN}}{t_{runtime_StSN}} \times 100 \quad (63)$$

$$\text{Efficiency (\%)} = \frac{E_{total_EeSN}}{E_{total_StSN}} \times 100 \quad (64)$$

A statistical analysis of the sleep current (I_{sleep}), identified as the most significantly affected component, was performed using MATLAB to assess the power consumption characteristics of the StSN and the EeSN. The mean current was calculated using (65), which defines the average current consumption during the respective operational state.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (65)$$

Where n denotes the total number of samples collected during the experiments.

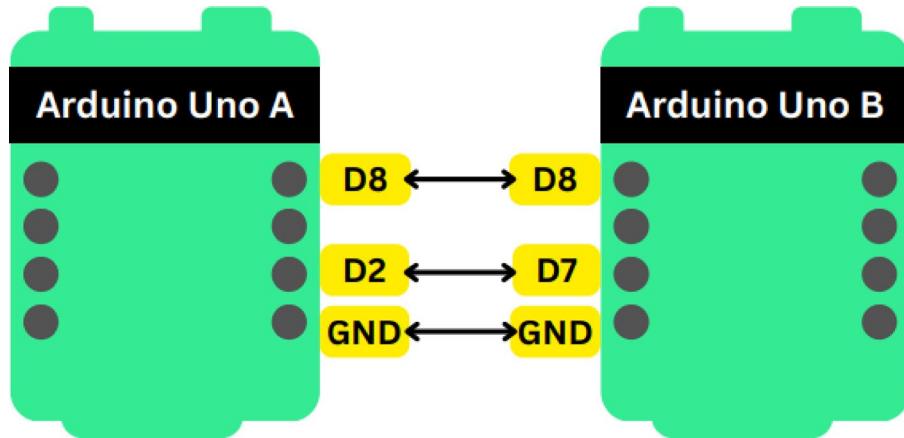
To quantify the variability in the measurements, the standard deviation σ was computed using (66), based on 20 independent samples per component:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (66)$$

To evaluate the reliability of these estimates, 95% confidence intervals (CIs) for the mean were calculated using the t-distribution as (67):

$$CI_{95\%} = \bar{x} \pm t^* \cdot \frac{\sigma}{\sqrt{n}} \quad (67)$$

Item description	Specification
Active duration (measurement), t_{active}	3 s
Sleep duration (measurement), t_{sleep}	30 s
Transmit Packet Size, P_{size_tx}	4 bytes
Receive Packet Size, P_{size_rx}	38 bytes
Total packets, $Packet_{total}$	20 packets
Node distance, d_{node}	10 m

Table 7. Experimental setup for PDR and RTT analysis.**Fig. 8.** Connection diagram to measure Arduino Uno wake-up latency.

The standard error of the mean (SEM) was obtained using (68) by dividing the standard deviation σ by the square root of the sample size:

$$SEM = \frac{\sigma}{\sqrt{n}} \quad (68)$$

The CI was then defined as $mean \pm t^* \cdot SEM$, with $t^* \approx 2.093$ for $n = 20$. This is expressed in (69):

$$\left[\bar{x} - t^* \cdot SEM, \bar{x} + t^* \cdot SEM \right] \quad (69)$$

The experiment was extended to analyze network performance in terms of packet delivery ratio (PDR) and round-trip time (RTT) latency. The experimental parameters largely replicate the configuration presented in Table 5, with several modifications as detailed in Table 7. The testbed was set up at the Auto-ID Laboratory, Universiti Sains Malaysia, with a distance of 10 m between the end node and the sink node. Both nodes were placed on a table approximately 1 m above the ground in an indoor non-line-of-sight (NLOS) environment, with a wooden door and a concrete wall acting as obstacles between them. The end nodes, namely EeSN and StSN, were configured to transmit ping “PING” packet and then will receive 38-byte packets at 30-second intervals.

The PDR is calculated based on the following equation:

$$PDR (\%) = \frac{Packet \text{ Successfully Received}}{Total \text{ Packets Sent}} \times 100 \quad (70)$$

The round-trip time (RTT) latency is computed as:

$$RTT_{\mu s} = t_{rcv} - t_{send} \quad (71)$$

where both the sending timestamp (t_{send}) and the receiving timestamp (t_{rcv}) are measured in microseconds. The RTT value in milliseconds is then given by:

$$RTT_{ms} = \frac{RTT_{\mu s}}{1000} \quad (72)$$

Wake-up latency was measured using two Arduino Uno boards. Arduino A was programmed to enter four sleep modes which are Idle, ADC Noise Reduction, Power-Save, and Power-Down modes via the Low Power library,

and was woken using an external interrupt on pin D2. Upon wake-up, Arduino A generated a short 10 µs pulse on pin D8, which was recorded by Arduino B to determine latency in microseconds. A one-second pre-sleep delay and a ten-second interval between successive wake-ups ensured stable operation and clear differentiation of each sleep mode. Measurements were repeated across multiple cycles to obtain representative latency values. Figure 8 shows the connection diagram used to measure the wake-up latency for both boards.

Results and discussion

This section presents the simulation results, experimental measurements, and corresponding analysis. It also discusses cross-platform extrapolation, emphasizing the portability of the sensor node design across various microcontroller platforms that is not limited to the Arduino Uno. By considering alternative microcontrollers with lower power profiles, enhanced processing capabilities, or integrated communication features, the system demonstrates adaptability for broader deployment scenarios and optimized performance.

Simulation and experimental results

Table 8 compares the hourly energy consumption and estimated runtime of the EeSN and StSN architectures based on simulation. Regardless of the transmission (TX) power level (Low: 60 mA, Mid: 120 mA, High: 215 mA), the energy consumption results remained consistent across all settings. The energy consumed by the EeSN was consistently 0.2155 Wh, while the StSN consumed 1.0387 Wh at all power levels. This uniformity in energy consumption can be attributed to the extremely short period of active transmission time (just 0.03 s) per transmission. Because of this brief duration, the variation in TX current had minimal impact on the overall energy consumption, leading to nearly identical energy results across all TX power levels.

The energy reduction between the StSN and EeSN remained 79.25% across all TX power levels, highlighting the significant energy savings achieved with the EeSN configuration. Additionally, the efficiency improvement was consistently 20.75%, which means that the EeSN consumed only 20.75% of the power used by the standard node (StSN) during active period. This demonstrates a substantial reduction in power consumption, indicating the superior efficiency of the EeSN with ASM model, which operates at a fraction of the energy cost compared to the traditional StSN.

The subsequent duty-cycle analysis illustrates how the energy consumption and efficiency of the EeSN and StSN configurations vary as the duty cycle increases. As the device remains awake for longer periods (increasing t_{awake}), the energy consumption of both configurations' increases; however, the relative energy savings of the EeSN remain substantial across all duty cycles.

- At a duty cycle of 8.33% (300 s awake), the EeSN consumed 0.2155 Wh, whereas the StSN consumed 1.0387 Wh, corresponding to a 79.25% reduction in energy consumption. The efficiency improvement at this duty cycle was 20.75%, indicating that the EeSN used only 20.75% of the power required by the StSN. The EeSN achieved a runtime of 168.37 h (7.02 days), compared to 34.94 h (1.46 days) for the StSN, representing a 381.96% improvement.
- At a duty cycle of 16.67% (600 s awake), the EeSN energy consumption increased to 0.317 Wh, while the StSN consumed 1.0654 Wh. The energy reduction decreased to 70.24%, whereas the efficiency improvement increased to 29.76%, meaning that the EeSN required only 29.76% of the StSN's power consumption. The EeSN runtime decreased to 114.46 h (4.77 days), while the StSN runtime remained at 34.06 h (1.42 days). The resulting runtime improvement for the EeSN was 236.04%, which, although lower than at the 8.33% duty cycle, remains considerable.
- At a duty cycle of 25% (900 s awake), the EeSN consumed 0.4186 Wh and the StSN consumed 1.0921 Wh, yielding an energy reduction of 61.67% and an efficiency improvement of 38.33%. Even at this elevated duty cycle, the EeSN continues to outperform the StSN in terms of energy efficiency. The EeSN achieved a runtime of 86.69 h (3.61 days), whereas the StSN operated for 33.23 h (1.39 days), resulting in a 160.91% improvement in runtime.

Overall, although the absolute energy consumption increases with higher duty cycles, the EeSN consistently provides significant energy savings and offers up to a 381.96% improvement in runtime, with only a modest reduction in the magnitude of these gains as the awake duration increases. Figures 9 and 10 present the comparison of energy analysis results for both nodes.

Prior to the energy consumption experiments, a calibration procedure was performed for each electrochemical sensor to ensure the accuracy of the measurement data. The results of the calibration are presented in Fig. 11, serving as reference values for validating sensor performance throughout the testing process. The average ambient temperature recorded during the procedure was 30.03 °C. Both sensors exhibited high accuracy, with the pH sensor showing a standard deviation of 0.02 for EeSN and 0.01 for StSN. Furthermore, the DO sensor demonstrated excellent accuracy, recording a value of 0, which aligns with the Zero Buffer Solution reference.

TX level	TX current (mA)	EeSN energy (Wh)	StSN energy (Wh)	Energy reduction (%)	Efficiency improvement (%)
Low ²⁵	60	0.2155	1.0387	79.25	20.75
Mid ²⁵	120	0.2155	1.0387	79.25	20.75
High ²⁵	215	0.2155	1.0387	79.25	20.75

Table 8. Simulation results of energy consumption and estimated runtime.

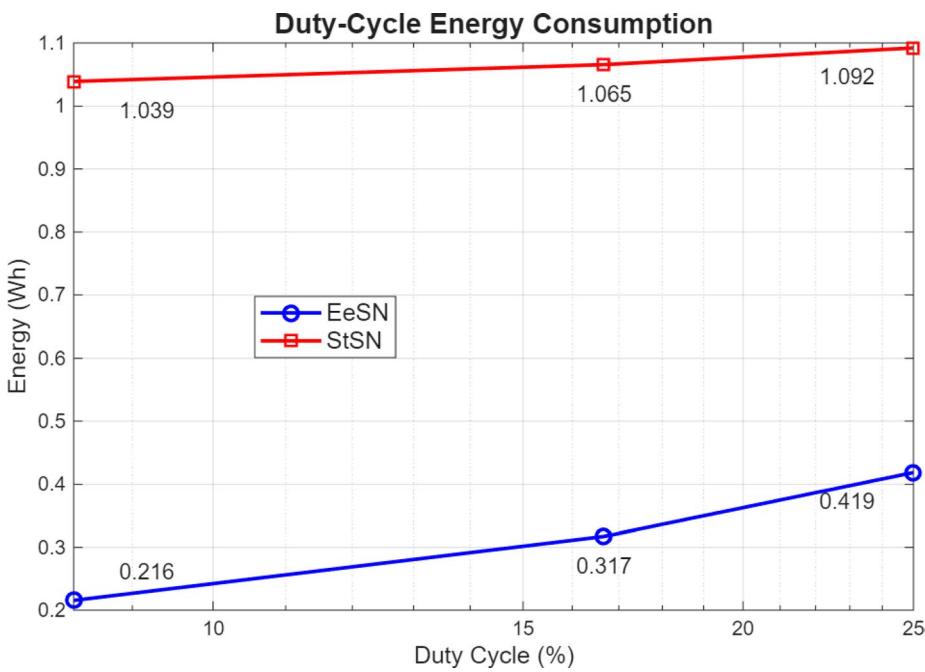


Fig. 9. Comparison of energy consumption.

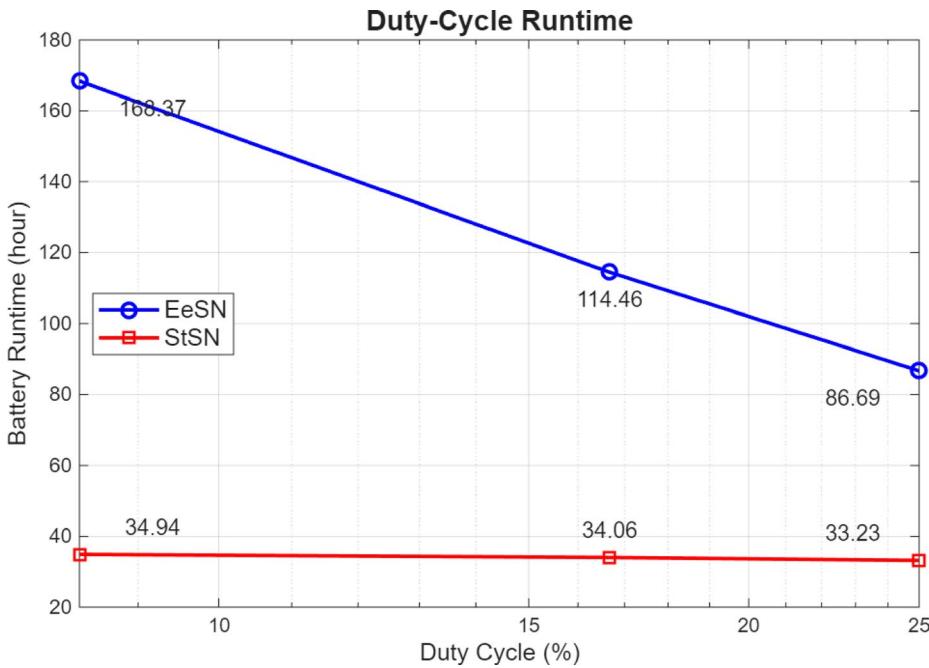


Fig. 10. Comparison of estimated runtime.

Figure 12 presents a consolidated bar graph comparing the measured current consumption across all components of the sensor node system. The near-identical results between both nodes validate the reliability of the proposed EeSN design, confirming its performance is comparable to that of a standard node. Figure 13 displays the sensor measurements collected from each node, further confirming consistent performance and stable operation throughout the evaluation. The pH readings ranged between 7.10 and 7.20, while the DO sensor recorded values between 4.50 mg/L and 5.50 mg/L. The temperature sensor measured values within the range of 29.05 °C to 29.30 °C. All sensors demonstrated expected performance following the completion of calibration procedures, ensuring measurement accuracy and system stability throughout the evaluation period.

A detailed analysis of the average measured current consumption for each sensor node component is presented in Tables 9 and 10, respectively.

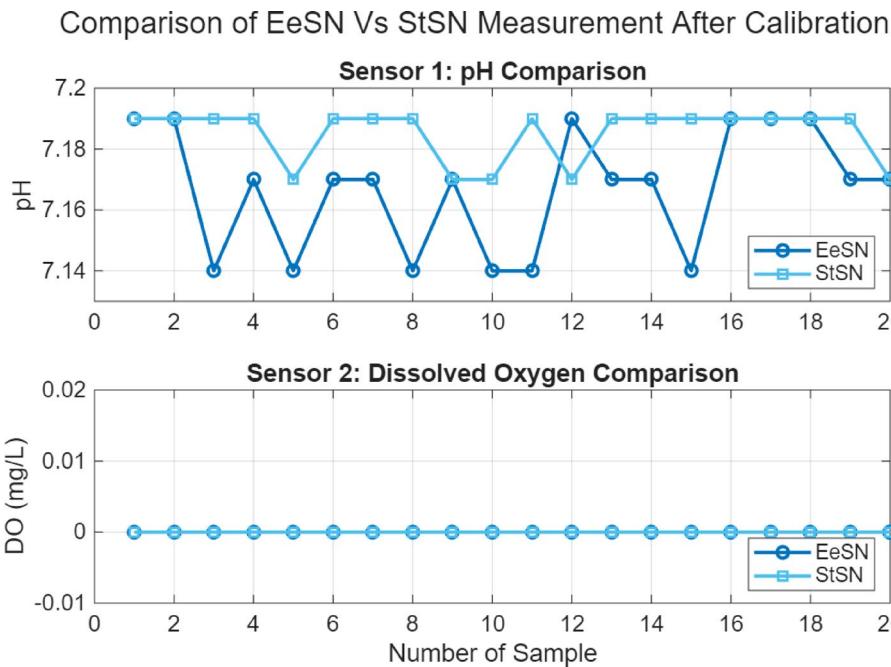


Fig. 11. Comparison of sensor measurement after calibration procedure.

For the EeSN node, the total energy consumed by active components, including the MCU, sensors, transceiver, isolation circuitry, and USB interface, was 0.1837 Wh, while the sleep components contributed an additional 0.0038 Wh. This resulted in a total energy consumption of 0.1875 Wh, corresponding to an estimated battery runtime of 193.53 h (approximately 8.06 days). In contrast, the StSN node exhibited significantly higher energy consumption, with active components consuming 0.6418 Wh and sleep components 0.0038 Wh, yielding a total of 0.6456 Wh and a reduced battery runtime of 56.21 h (2.34 days). Comparative analysis indicates that the EeSN node achieves a 70.96% reduction in total energy consumption relative to the StSN node. This reduction translates into a 244.31% improvement in battery runtime and a 29.04% increase in overall energy efficiency. A comparison of energy consumption analysis and estimated battery runtime, based on both simulation and experimental results, is illustrated in Figs. 14 and 15, respectively.

The comparison of calculated current consumption during the idle condition demonstrates that the EeSN achieves a statistically significant and practically substantial reduction in operating current relative to the StSN. In this phase, both nodes enter the XBee Pro's sleep mode via the DTR interrupt pin following successful data transmission. The EeSN further reduces power usage by gating the power supply to all sensing blocks through the ASM.

The StSN exhibited a mean current draw of 60.06 mA ($SD = 0.38$ mA), with a 95% confidence interval (CI) ranging from 59.90 to 60.23 mA. In contrast, the EeSN showed an extremely low and tightly distributed current consumption, with a mean of only 0.49 mA ($SD = 0.01$ mA) and a 95% CI of [0.49, 0.49], indicating highly stable and repeatable low-power operation. A two-tailed t-test yielded a P-value ≈ 0.001 , confirming that the difference in current consumption between StSN and EeSN is statistically significant at the 95% confidence level. Moreover, the effect size, expressed as Cohen's $d = -223.67$, indicates an extremely large practical effect, far exceeding conventional thresholds and highlighting that the EeSN operates at a radically lower current level than the StSN. Table 11 presents a summary of the statistical analysis of the idle condition results, as detailed below:

Table 12 presents the MATLAB simulation results of energy-efficient network performance analysis. The network simulations were conducted to evaluate the performance of the proposed protocol under varying network conditions. The network lifetime analysis revealed that the time to first node death was 619,544 s (approximately 172.1 h), indicating a highly energy-efficient operation under the simulated conditions. This extended lifetime demonstrates the protocol's effectiveness in conserving node energy, even as the network scales in size.

Retransmission behaviour was minimal, with an average of 0.26 retries per packet. The associated energy cost for retransmissions was measured at 223.56 J, suggesting that communication overhead due to packet loss is negligible and does not significantly impact overall energy consumption.

Node density scaling tests indicated that the network lifetime remained constant at 619,544 s across all node densities ranging from 10 to 100 nodes. This invariance implies that the protocol maintains robustness and energy efficiency regardless of network size.

Finally, the energy consumption under different traffic loads was evaluated. Light, medium, and heavy traffic scenarios consumed 855.93 J, 870.78 J, and 941.62 J, respectively. The gradual increase in energy expenditure with traffic load is expected due to higher transmission demands, yet the network demonstrates stable and predictable energy performance.

Measured Current Consumption Comparison: EeSN vs StSN

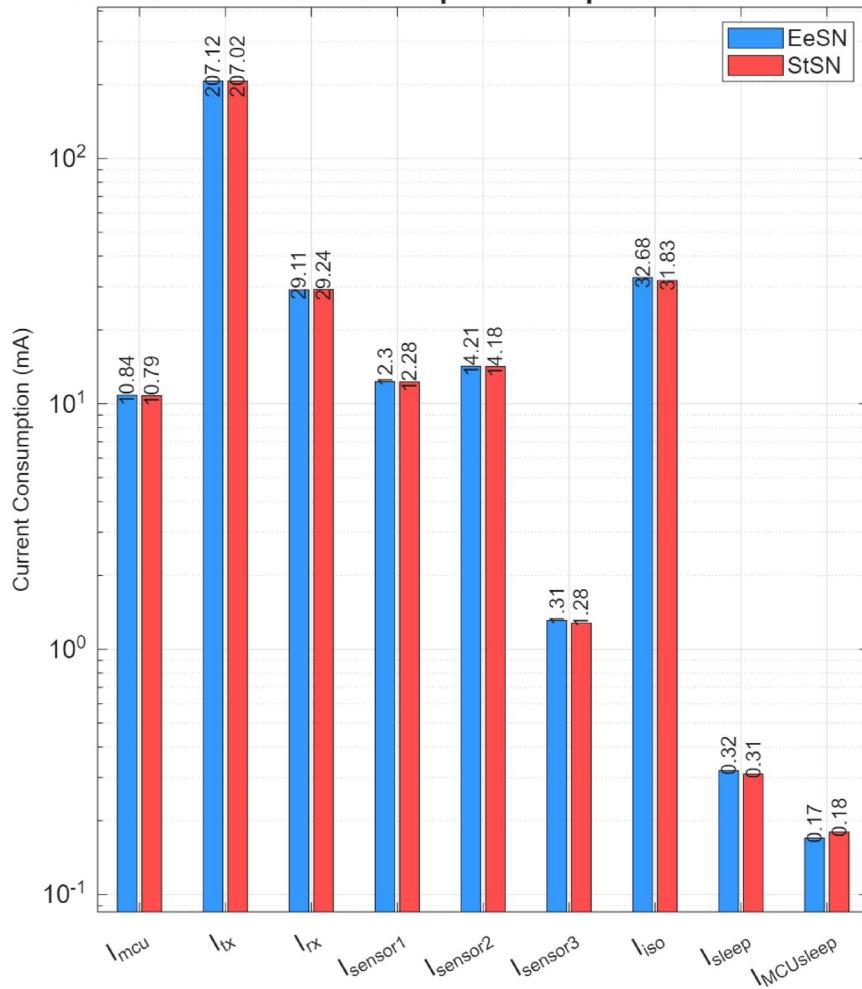


Fig. 12. Comparison of measured current consumption of all components within the sensor node system.

The subsequent simulation on transient response and leakage current characteristics of both N-channel and P-channel MOSFETs are illustrated in Fig. 16, whereas the switching losses and conduction metrics are depicted in Fig. 17. A summary of the corresponding simulation results is provided in Table 13.

The NMOS exhibited a turn-on delay of 839.88 ns and a rise time of 565.03 ns, while the PMOS showed a faster turn-on delay of 196.37 ns but a longer rise time of 992.34 ns. Turn-off delays were comparable which are 669.55 ns for NMOS and 605.07 ns for PMOS respectively, though the PMOS had a significantly longer fall time (2417.73 ns vs. 529.75 ns).

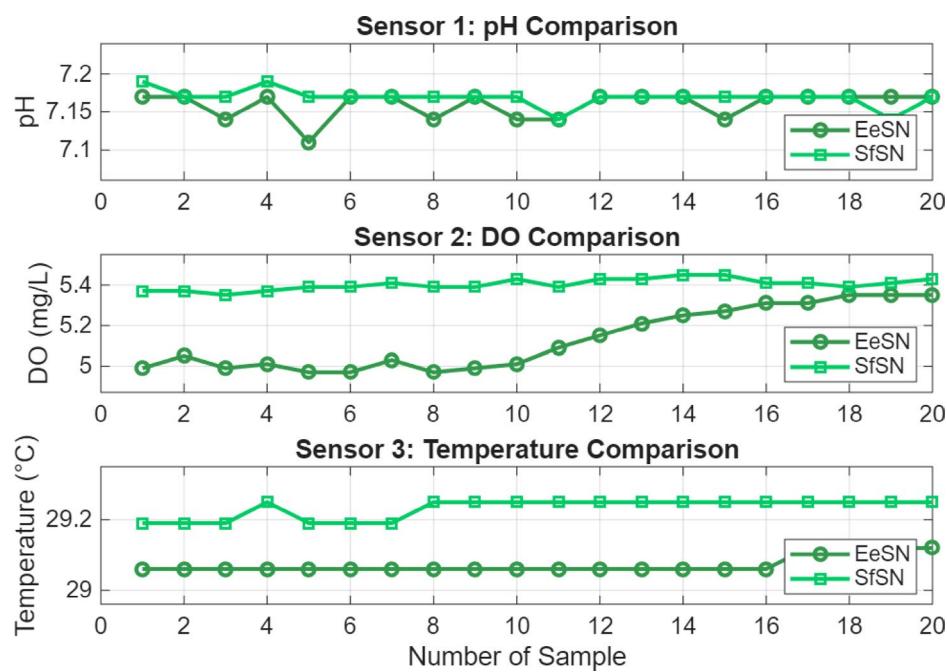
Switching energy measurements showed the NMOS incurred $6.42 \mu\text{J}$ (E_{on}) and $6.02 \mu\text{J}$ (E_{off}), whereas the PMOS required only $2.50 \mu\text{J}$ and $6.09 \mu\text{J}$, respectively. In conduction, as expected, the NMOS demonstrated lower losses, with an $R_{\text{on_est}}$ of $31.73 \text{ m}\Omega$, $V_{\text{drop_avg}}$ of 1.59 mV , and P_{cond} of 0.079 mW . In contrast, the PMOS exhibited higher values: $227.76 \text{ m}\Omega$, 11.36 mV , and 0.567 mW . Leakage currents were measured at $28.80 \mu\text{A}$ for NMOS and $45.20 \mu\text{A}$ for PMOS in the OFF state.

As noted in³³, a high-side switch configuration places the power device between the supply voltage and the load, which is referenced to ground. When driving inductive loads, a rapid turn-off requires the output voltage to swing negative in order to quickly dissipate stored energy. This necessitates specialized isolation structures and gate drive techniques to minimize turn-off time which is an essential requirement for precise pulse-width modulation control.

Despite the NMOS device offering better electrical efficiency during conduction, the PMOS is preferred for high-side switching in low-power sensor node applications. This preference is primarily due to its lower switching energy, simpler gate drive requirements, inherent support for stable sensor grounding, and direct compatibility with typical microcontroller logic levels, eliminating the need for additional level-shifting circuitry.

The bill of materials (BOM) cost for both NMOS and PMOS high-side switching circuitry is detailed in Table 14, as referred in³⁴.

Next experiment focused on the analysis of packet delivery ratio (PDR) and round-trip time (RTT) latency. The results are presented in Fig. 18. Based on the recorded data, the EeSN node achieved an average RTT latency of 255.85 ms, slightly lower than the StSN node, which recorded an average of 256.62 ms. Both sensor nodes

**Fig. 13.** Comparison of sensor measurement for each sensor node.

Sensor node component	Average current consumption (mA)	Time duration (seconds)	Hourly energy consumption (Wh)
i_{mcu}	10.84	300	0.0076
i_{mcu_sleep}	0.17	3300	0.0013
i_{tx}	207.12	0.03	0.00001450
i_{rx}	29.11	300	0.0204
$i_{sensor1}$	12.3	300	0.0086
$i_{sensor2}$	14.21	300	0.0099
$i_{sensor3}$	1.31	300	0.0009
i_{iso}	32.68	300	0.0229
i_{usb}^{27}	13.5	3600	0.1134
i_{sleep}	0.32	3300	0.0025

Table 9. Energy analysis results of EESN.

Sensor node component	Average current consumption (mA)	Time duration (seconds)	Hourly energy consumption (Wh)
i_{mcu}	10.79	300	0.0076
i_{mcu_sleep}	0.18	3300	0.0014
i_{tx}	207.02	0.03	0.00001449
i_{rx}	29.24	300	0.0205
$i_{sensor1}$	12.28	3600	0.1032
$i_{sensor2}$	14.18	3600	0.1191
$i_{sensor3}$	1.28	3600	0.0108
i_{iso}	31.83	3600	0.2674
i_{usb}^{27}	13.5	3600	0.1134
i_{sleep}	0.31	3300	0.0024

Table 10. Energy analysis results of STSN.

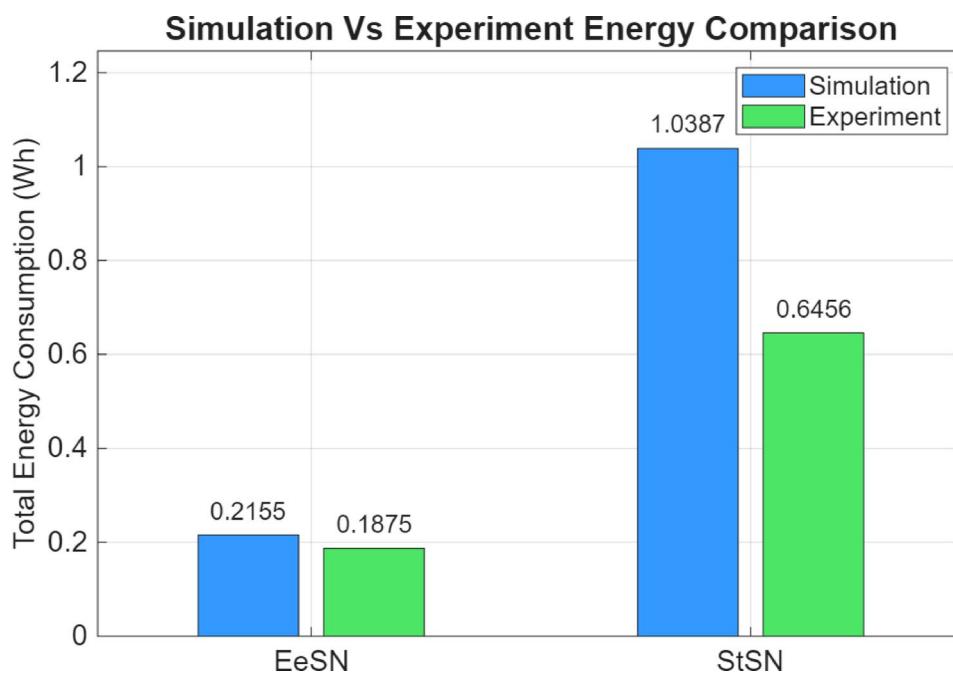


Fig. 14. Comparison of energy analysis based on simulation and experiments.

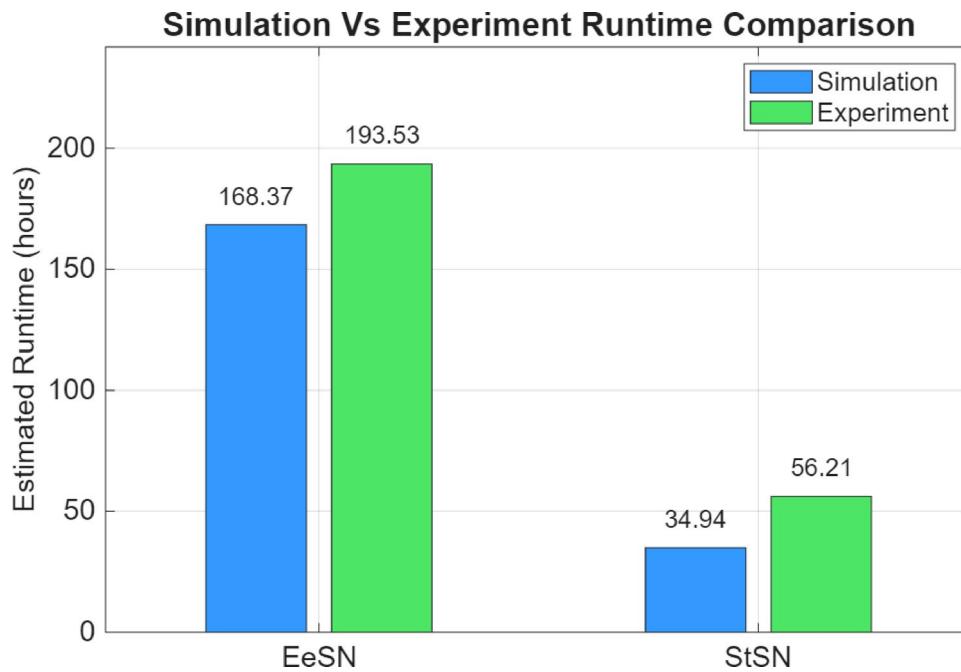


Fig. 15. Comparison of battery runtime based on simulation and experiments.

achieved a PDR of 100%, indicating that the EeSN architecture does not adversely affect PDR or RTT latency performance.

Table 15 presents the measured wake-up latencies for the EeSN and StSN nodes using the Arduino Uno platform. The results indicate that both nodes exhibit nearly identical wake-up latency characteristics across all operating modes. Specifically, in Idle mode, EeSN and StSN show wake-up latencies of 15.31 μ s and 15.25 μ s, respectively, while in ADC Noise Reduction mode, the latencies are 13.75 μ s and 13.69 μ s. For low-power modes, EeSN records 1764.88 μ s and 1764.31 μ s in Power-Save and Power-Down states, respectively, whereas StSN exhibits slightly higher wake-up latencies of 1764.94 μ s and 1764.38 μ s. The close agreement between the two nodes confirms that the employed ASM circuit does not introduce any noticeable delay or degradation in

Metric	StSN	EeSN
Mean current (mA)	60.06	0.49
Standard deviation (mA)	0.38	0.01
95% Confidence interval (mA)	59.90–60.23	0.49–0.49*
Two-tailed t-test p-value	$p \approx 0.001$	–
Cohen's d	– 223.67	–

Table 11. Summary of idle current consumption in StSN and EeSN architectures. *Rounded to two decimal places.

Parameter / condition	Value / result
Time to first node death	619,544 s (172.1 h)
Mean retries per packet	0.26
Energy for retransmissions	223.56 J
Network lifetime (by node density)	10–100 nodes: 619,544 s
	Light: 855.93 J
Energy consumption (traffic load)	Medium: 870.78 J
	Heavy: 941.62 J

Table 12. Result of energy-efficient network performance analysis.

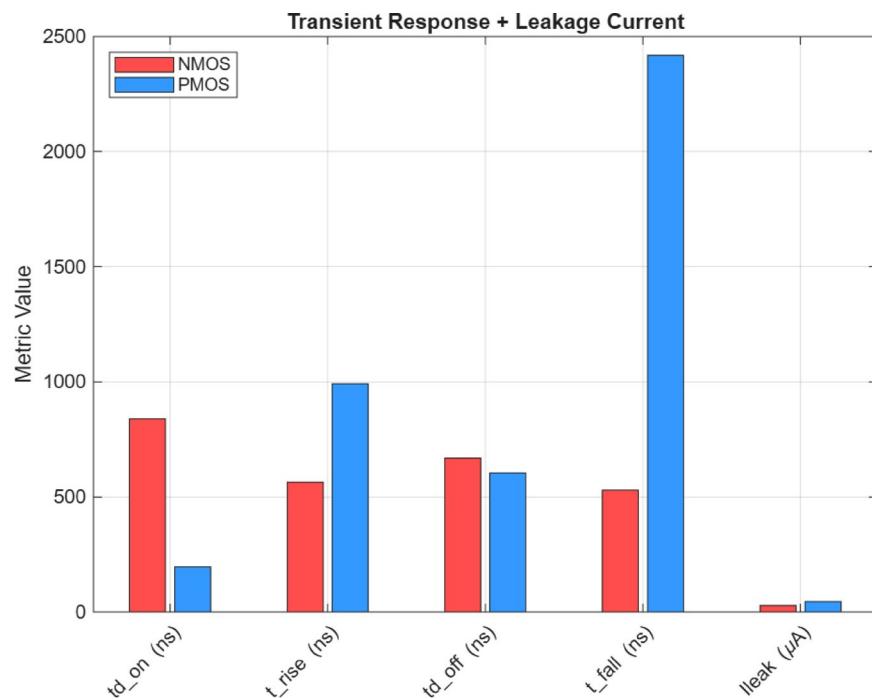
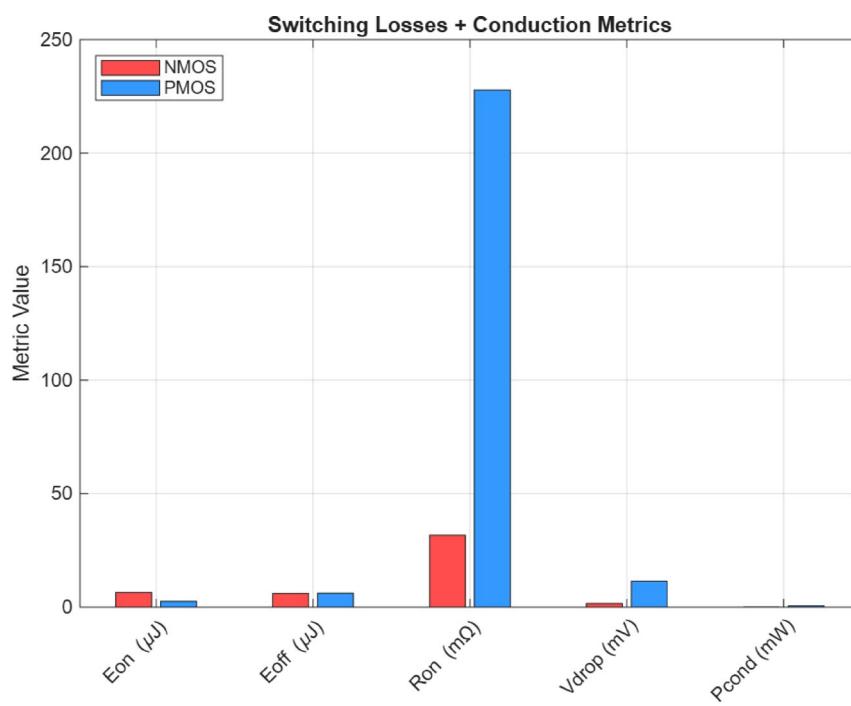


Fig. 16. MOSFET transient response and leakage current analysis.

wake-up latency performance. Hence, the integration of the ASM circuit maintains the original responsiveness of the Arduino Uno across all sleep modes.

Quantitative power and wake-up latency comparison

The quantitative comparison of power consumption and wake-up latency indicates that the EeSN node achieves a balanced performance in terms of peak active current, sleep current, and wake-up responsiveness. The node's peak active current, encompassing both MCU and radio transmission, is 217.96 mA, slightly lower than the extrapolated STM32F1 + XBee Pro combination (226.8 mA) and significantly lower than the ESP32, which typically peaks at 160–260 mA during full CPU and Wi-Fi operation. As summarized in Table 16, the MCU alone draws 10.84 mA, radio transmission adds 207.12 mA, and the combined peak reaches 217.96 mA. In sleep

**Fig. 17.** MOSFET switching losses and conduction metrics analysis.

Measured parameter	NMOS(IRFZ44N)	PMOS (IRF9540N)
t_{d_on} (ns)	839.88	196.37
t_{rise} (ns)	565.03	992.34
t_{d_off} (ns)	669.55	605.07
t_{fall} (ns)	529.75	2417.73
E_{on} (μJ)	6.42	2.5
E_{off} (μJ)	6.02	6.09
R_{on_est} (m Ω)	31.73	227.76
V_{drop_avg} (mV)	1.59	11.36
P_{cond} (mW)	0.079	0.567
I_{leak} (μA)	28.8	45.2

Table 13. LTspice simulation result.

Component	NMOS (IRFZ44N)	PMOS (IRF9540N)
Power MOSFET	IRFZ44N – ~\$1.43	IRF9540N – ~\$1.71
Gate Driver IC	IR2184 – ~\$3.37	Not required
Bootstrap Diode	1N5819 – ~\$0.10	Not required
Bootstrap Capacitor	10 μF ceramic – ~\$0.22	Not required
Gate Resistor	100 Ω (0.4 W) – ~\$0.02	100 Ω (0.4 W) – ~\$0.02
Pull Resistor	Pull down 10 k Ω – ~\$0.03	Pull-up 10 k Ω – ~\$0.03
Level-shifting transistor	Not required	BC846B – ~\$0.08
Total cost	~\$4.99	~\$1.84

Table 14. Bill of materials (BOM) cost comparison of high-side switching implementations using NMOS and PMOS transistors.

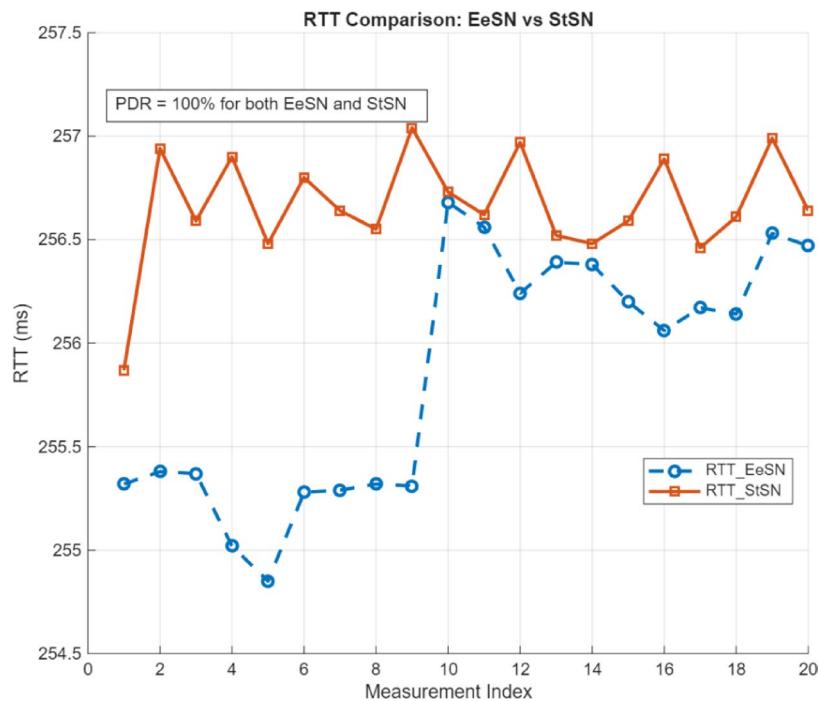


Fig. 18. Packet Delivery ratio (PDR) and round-trip time (RTT) latency analysis.

Node	Idle (μs)	ADC noise reduction (μs)	Power-save (μs)	Power-down (μs)
EeSN	15.31	13.75	1764.88	1764.31
StSN	15.25	13.69	1764.94	1764.38

Table 15. Wake-up latency analysis across ARDUINO Uno mode.

Mode	Current (mA)	Wake-up latency (μs)	Description
MCU Active	10.84	–	CPU fully active, peripherals running
Transmit (TX)	207.12	–	Radio transmitting data
Total Active (MCU + TX)	217.96	–	Combined peak current
MCU Sleep	0.17	–	CPU halted, <i>LowPower.h</i> applied
Power-Down	–	1764.31	Deepest sleep mode

Table 16. EESN node power and wake-up latency characterisiscs.

Mode	Current (mA)	Wake-up latency	Description
Run @ 36 MHz	11.8 ³⁵	–	MCU fully active, peripherals running
Extrapolated Active (MCU + XBee Pro)	226.8	–	Combined MCU + XBee Pro transmit current [25]
Standby	1–3 μA ³⁵	50 μs ³⁵	RTC/VBAT domain only active

Table 17. STM32F102C4 power and wake-up latency characteristics (wiith XBEE Pro).

mode, the MCU consumes 0.17 mA, whereas the deepest Power-Down mode exhibits a wake-up latency of 1.76 ms.

For the STM32F102C4 with XBee Pro, active operation at 36 MHz requires 11.8 mA, with the combined MCU and XBee Pro transmission extrapolated to 226.8 mA. Its standby mode achieves ultra-low current (1–3 μA) and a minimal wake-up latency (50 μs) as presented in Table 17.

The ESP32 exhibits a higher peak current range (160–260 mA) during full operation, while in Hibernate mode, the current is reduced to 5 μA with a wake-up latency below 1 ms, as shown in Table 18:

Mode	Typical current	Wake-up latency	Description
Modem-Sleep (MCU Active)	3–30 mA ³⁶	< 1 µs ³⁶	CPU active, RF disabled
Active (CPU + Wi-Fi TX)	160–260 mA ^{36,37}	–	Full system operation
Hibernate	5 µA ³⁶	< 1 ms ³⁶	Minimal retention state

Table 18. Power and wake-up latency characteristics.

Platform	Peak active current	Lowest-power mode	Sleep current	Wake-up latency
EeSN	217.96 mA	Power-down	0.17 mA	1.76 ms
STM32F1	226.8 mA	Standby	1–3 µA ³⁵	50 µs ³⁵
ESP32	260 mA	Hibernate	5 µA ³⁶	< 1 ms ³⁶

Table 19. Power cross platform comparison of power and latency.

Table 19 summarizes the cross-platform comparison of peak active current, lowest-power mode, sleep current, and wake-up latency:

These results indicate that although the EeSN exhibits higher sleep current than the comparator platforms, it offers a practical balance between communication capability and energy consumption. Overall, the comparison underscores the inherent trade-offs between peak current demand, low-power mode efficiency, and wake-up performance, thereby providing a clear basis for selecting suitable architectures in energy-constrained sensing applications.

Conclusion

This study demonstrates the effectiveness of the proposed Adaptive Switching Mechanism (ASM) in substantially reducing energy consumption and improving operational efficiency in battery-powered IoT sensor nodes. Experimental results confirm its capability, yielding up to 70.96% energy savings, a 244.31% increase in battery runtime, and a power demand of only 29.04% relative to the unmodified node indicating a marked improvement in energy efficiency compared to conventional designs. Although the evaluation was conducted within an aquaculture monitoring scenario, the modular architecture is broadly applicable to various low-power IoT domains, including precision agriculture, environmental monitoring, and smart city infrastructure.

However, the current implementation is limited to a single microcontroller platform and does not account for the long-term effects of frequent switching, which may affect hardware reliability during extended deployments. Additionally, the system's scalability in heterogeneous and dynamic IoT environments where devices vary in capabilities and energy profiles remains unexplored.

Future work will focus on incorporating lightweight, edge-based machine learning models for context-aware and predictive energy management. The integration of programmable power management ICs (PMICs) with dynamic, workload-driven power policies will also be investigated to enhance system flexibility. To assess scalability, robustness, and interoperability, the mechanism will be validated in large-scale, real-world IoT testbeds involving diverse node types. Finally, the degradation implications of fast, dynamic power switching and its impact on real-time performance will be studied, aligning the system with emerging requirements for sustainable and resilient IoT computing.

Data availability

All data generated or analyzed in this study, including the raw and processed datasets, simulation scripts, configuration files, and experiment log files, are available in the Zenodo repository at <https://doi.org/10.5281/zenodo.17705470>. These materials provide all resources necessary to reproduce the figures, tables, and analyses presented in the manuscript.

Received: 1 August 2025; Accepted: 11 December 2025

Published online: 03 January 2026

References

- Baranwal, J., Barse, B., Gatto, G., Broncova, G. & Kumar, A. Electrochemical sensors and their applications: A review. *Chemosensors*, **10**(9), p.363. (2022).
- Electrical Isolation | Atlas Scientific. Atlas Scientific. atlas- Accessed (6 June 2025). (2019).
- Vertical Isolator | Atlas Scientific. Atlas Scientific, atlas-scientific.com/electrical-isolation/vertical-isolator/. Accessed (6 June 2025). (2022).
- Samizadeh Nikoui, T., Rahmani, A. M., Balador, A. & Haj Seyyed Javadi, H. Internet of things architecture challenges: A systematic review. *Int. J. Commun Syst.* **34** (4), e4678 (2021).
- Tolentino, L. K. et al. IoT-based automated water monitoring and correcting modular device via LoRaWAN for aquaculture. *Int. J. Comput. Digit. Syst.* **10**, 533–544 (2021).
- Huan, J., Li, H., Wu, F. & Cao, W. Design of water quality monitoring system for aquaculture ponds based on NB-IoT. *Aquacultural Eng.*, **90**, p.102088. (2020).
- Araneta, A. A. S. Design of an Arduino-based water quality monitoring system. *Int. J. Comput. Sci. Mob. Comput.* **11** (3), 152–165 (2022).

8. Olmedo, E. & Smith, M. March. Development of an IoT water quality monitoring system for data-based aquaculture siting. In *SoutheastCon 2021* (pp. 1–8). IEEE. (2021).
9. Sanislav, T., Mois, G. D., Zeadally, S. & Folea, S. C. Energy harvesting techniques for internet of things (IoT). *IEEE access.* **9**, 39530–39549 (2021).
10. Hu, W. C., Chen, L. B., Wang, B. H., Li, G. W. & Huang, X. R. Design and implementation of a full-time artificial intelligence of things-based water quality inspection and prediction system for intelligent aquaculture. *IEEE Sens. J.* **24** (3), 3811–3821 (2023).
11. Duangwongs, J., Ungsethaphand, T., Akaboot, P., Khamjai, S. & Unankard, S. March. Real-time water quality monitoring and notification system for aquaculture. In *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering* (pp. 9–13). IEEE. (2021).
12. Shete, R. P., Bongale, A. M. & Dharrao, D. IoT-enabled effective real-time water quality monitoring method for aquaculture. *MethodsX*, **13**, p.102906. (2024).
13. Zaini, A., Wulandari, D. P. & Wulandari, R. November. Data Visualization on Shrimp Pond Monitoring System Based on Temperature, pH, and DO (Dissolved Oxygen) with IoT. In *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)* (pp. 1–6). IEEE. (2020).
14. Anusha Bamin, A. M. et al. IoT-Based automatic water quality monitoring system with optimized neural network. *KSII Trans. Internet Inform. Syst. (TIIS)* **18** (1), 46–63 (2024).
15. Ajith, J. B., Manimegalai, R. & Ilayaraja, V. February. An IoT based smart water quality monitoring system using cloud. In *2020 International conference on emerging trends in information technology and engineering (ic-ETITE)* (pp. 1–7). IEEE. (2020).
16. KP, R. A. H. & Harigovindan, V. P. October. Water quality monitoring and reporting system for aquaculture with solar energy harvesting. In *2020 International conference on smart technologies in computing, electrical and electronics (ICSTCEE)* (pp. 66–70). IEEE. (2020).
17. Jais, N. A. M., Abdullah, A. F., Kassim, M. S. M., Abd Karim, M. M. & Muhamadi, N. A. Improved accuracy in IoT-Based water quality monitoring for aquaculture tanks using low-cost sensors: Asian Seabass fish farming. *Heliyon*, **10**(8). (2024).
18. Bembe, M., Abu-Mahfouz, A., Masonta, M. & Ngondi, T. A survey on low-power wide area networks for IoT applications. *Telecommun. Syst.* **71** (2), 249–274 (2019).
19. Hassan, W., Førre, M., Ulvund, J. B. & Alfredsen, J. A. Internet of fish: integration of acoustic telemetry with LPWAN for efficient real-time monitoring of fish in marine farms. *Comput. Electron. Agric.* **163**, 104850 (2019).
20. Dhall, R. & Agrawal, H. An improved energy efficient duty cycling algorithm for IoT based precision agriculture. *Procedia Comput. Sci.* **141**, 135–142 (2018).
21. Pinzi, M. & Pozzebon, A. Low-cost power gating solution to increase energy efficiency optimising duty cycling in wireless sensor nodes with power-hungry sensors. *IET Wirel. Sens. Syst.* **9** (1), 25–31 (2019).
22. Panic, G., Stamenkovic, Z. & Kraemer, R. Power gating in wireless sensor networks. In *2008 3rd International Symposium on Wireless Pervasive Computing* (pp. 499–503). IEEE. (2008).
23. Samiullah, M., Irfan, M. Z. & Rafique, A. Microcontrollers: A Comprehensive Overview and Comparative Analysis of Diverse Types. (2023).
24. Salah, H., Alouneh, S., Al-Assaf, A. & Darabkh, K. Performance evaluation of DigiMesh and ZigBee wireless mesh networks. In *2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 1–6). IEEE. (2018).
25. Digi International Inc. XBee/XBee-PRO RF modules user guide (Document No. 90002173) [Manual]. Digi International. <https://docs.digi.com/resources/documentation/digidocs/pdfs/90002173.pdf> (2020).
26. Microchip Technology Inc. ATmega328P: Automotive microcontrollers (Datasheet No. 7810D-AVR-01/15). https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (2015).
27. Microchip Technology Inc. ATmega8(L) 8-bit AVR microcontroller with 8K bytes in-system programmable flash [Datasheet]. Microch. Technol.. <https://ww1.microchip.com/downloads/en/DeviceDoc/doc7799.pdf> (2012).
28. Vertical Isolator Embedded Voltage Isolator EZQ TM V 1.1. (n.d.). Available at: <https://files.atlas-scientific.com/ezo-vertical-isolator-datasheet.pdf> [Accessed 29 Jun 2025].
29. Ahmed, E., Gani, A., Khan, M. K., Buyya, R. & Khan, S. U. Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges. *J. Netw. Comput. Appl.* **52**, 154–172 (2015).
30. Infineon. IRF9540NS - P-channel power MOSFETs | Infineon Technologies. <https://www.infineon.com/part/IRF9540NS> (2001).
31. Infineon. IRFZ44N - N-channel power MOSFETs | Infineon Technologies <https://www.infineon.com/part/IRFZ44N>. (2001).
32. Piyare, R. & Lee, S. R. Performance analysis of XBee ZB module based wireless sensor networks. *Int. J. Sci. Eng. Res.* **4** (4), 1615–1621 (2013).
33. Sakamoto, K., Nunogawa, Y., Satonaka, K., Kouda, T. & Horiuchi, S. An intelligent power IC with reverse battery protection for fast-switching high-side solenoid drive. *IEEE Trans. Electron. Devices*, **46** (8), 1775–1781 (1999).
34. RS. (n.d.). RS. Retrieved October 21. <https://us.rs-online.com/product/> (2025).
35. STMicroelectronics. STM32F102C4 datasheet. <https://www.st.com/resource/en/datasheet/stm32f102c4.pdf> (2019).
36. Espressif Systems. ESP32 technical reference manual. https://documentation.espressif.com/api/resource/doc/file/NyqlpvzX/FILE/e_sp32_technical_reference_manual_en.pdf (2025).
37. Espressif Systems. ESP32 Series Datasheet (Version 5.1) [PDF], October 14 https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (2025).

Author contributions

Kamarul Hafiz Kamaludin contributed to the conceptualization, methodology design, system implementation, simulation, and experimental validation. He was also responsible for data curation, formal analysis, and the initial draft of the manuscript. Widad Ismail supervised the project and contributed to the conceptual framework, validation of technical content and create new method of measurements and real-time implementation, critical revision of the manuscript, and final approval of the version to be submitted. She also provided the necessary resources and funding acquisition support. Both authors have reviewed and approved the final version of the manuscript and agree to be accountable for all aspects of the work.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to W.I.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026