



## COMPARATIVE PERFORMANCE ANALYSIS OF HARDWARE AND SOFTWARE AES ENCRYPTION IMPLEMENTATION IN WSN

**Ljubomir Vračar, Milan Stojanović**

University of Niš, Faculty of Electronics Engineering, Niš, Serbia

ORCID iDs: Ljubomir Vračar  
Milan Stojanović

 <https://orcid.org/0000-0003-0258-7012>  
 <https://orcid.org/0000-0002-3474-7968>

**Abstract.** *Wireless Sensor Networks (WSNs) have been among the most researched areas in the last ten or twenty years. Different sensor technologies and wireless communication contributed to the progress in this field, but two features of these systems are still insufficiently developed. These are data security and energy consumption. Data is particularly vulnerable during transmission, whether it is communication between nodes or between a base station and a server. One of the most promising solutions for this problem is data encryption. However, data encryption negatively affects sensor nodes' energy consumption, so there is a compromise between data security and energy consumption. Symmetric encryption algorithms are more effective than asymmetric for these applications. Advanced Encryption Standard (AES) is the most frequently used symmetric algorithm. This paper analyzes the performance of hardware and software implementation of the AES encryption algorithm on the sensor node. The hardware implementation is based on the integrated circuit ATECC608A, while the microcontroller firmware includes the software implementation. The analyzed parameters include execution time, energy consumption, and memory usage, for four working frequencies (8 MHz, 16 MHz, 32 MHz, and 64 MHz), and different supply voltages (2.6 V, 3.3 V, and 5 V). The results show that the hardware implementation works faster for the lower working frequency (up to 12 MHz). The execution time of software implementation is reduced approximately 50 % with doubled frequency, while the reduction is less, and inconstant in hardware implementation. The hardware implementation consumes more energy than software, and the difference rises with the working frequency. These differences are 20 %, 40 %, 60 %, and 75 % at the same frequencies considered for execution time analyzing. The hardware implementation occupies 4.58 % of the memory of the microcontroller (PIC18F45K22), while 10.13 % is needed for the software implementation. These results can provide a good starting point for choosing an appropriate encryption implementation method.*

**Key words:** *Wireless Sensor Network, Data Security, Encryption, Advanced Encryption Standard (AES), Energy Consumption*

Received August 12, 2024; revised November 12, 2024, December 17, 2024 and December 30, 2024; accepted January 01, 2025

**Corresponding author:** Milan Stojanović  
University of Niš, Faculty of Electronics Engineering, Niš, Serbia  
E-mail: [milan.stojanovic@elfak.ni.ac.rs](mailto:milan.stojanovic@elfak.ni.ac.rs)

## 1. INTRODUCTION

The Internet of Things (IoT) paradigm is the technology that provides the internet connection, and exchanging information between millions of smart devices [1]. Wireless Sensor Networks (WSNs) are one of the most essential subnetworks of the IoT. A WSN is a self-organized network consisting of nodes that collect real-time data from their surroundings and communicate through a wireless medium. The three functions accomplished by the sensor nodes are sensing, data processing, and communication. There are constant increases in WSNs applications, including indoor and outdoor deployment [2]. Some of them are smart home systems, environmental conditions monitoring, intelligent transportation surveillance, military applications, industrial monitoring, and agricultural monitoring [3]. As WSNs continue to grow, it is necessary to provide efficient security mechanisms. To increase the effectiveness of these networks, additional security is needed to guarantee the confidentiality, integrity, and authenticity of the data [4].

Recent developments in wireless sensor network technology have created new data transmission security issues. In wireless broadcast media, the nature of broadcast communication permits message interception, eavesdropping, and modification. WSNs are vulnerable to various threats and attacks, just like traditional wireless networks, which can result in delays, communication disruptions, and information loss. The standard method for network communication security is to build an end-to-end trust relationship between the message's sender and recipient. The three security properties necessary in WSNs are authentication, confidentiality, and integrity.

*Authentication:* Underlying system for network access control. When a node attempts to connect, the base station must be aware of and confirm its legitimacy. By using authentication, an attacker cannot join the network and send false messages.

*Confidentiality:* Keeping information private from unwanted access in order to protect the confidentiality of the messages that are exchanged and prevent adversaries from learning about it. Verify that the message originated from the specified source.

*Integrity:* It confirms that received data are not changed during transmission through the network. Cryptographic hash functions or encryption algorithms can be used to accomplish these requirements.

Traditional security solutions require a lot of computing energy and large memory. Therefore, they cannot be applied to WSNs because of significant resource constraints. Sensor nodes are usually deployed in unattended environments without access to an electrical grid and powered by batteries. The battery recharging or replacement is not easy, so the sensor node's energy consumption should be reduced maximally. Besides limited available energy, there are additional constraints of sensor nodes, such as processing power and memory. As wireless sensor nodes have limited sensing, processing, and transmission capabilities, the implemented security mechanism should minimize the execution time, consume low energy, and occupy the program memory as little as possible.

Current researchers focused on developing security mechanisms for these resource-constrained networks [5]. Many methods for securing data transfer have been revealed, such as security protocols [6], digital signatures, and encryption [7]. The current researches on WSNs security mainly include encryption algorithms. Data encryption is the process of transforming data into a format that can only be read by authorized personnel who possess a decryption key. Before encryption, the data is referred to as plaintext, while after encryption, the data is called cipher text.

Encryption algorithms include two kinds: symmetric encryption (DES, AES, Skipjack, TEA, XTEA) and asymmetric encryption (Elliptic curve, RSA) [8]. Symmetric encryption algorithms encrypt the data with a symmetric key (private key). The sender and receiver use the same key to encrypt and decrypt data. Symmetric encryption algorithms include two basic categories: block ciphers and stream ciphers. A block cipher encrypts a block of plaintext and produces a block of encrypted data, while a stream cipher can encrypt plaintexts of varying sizes. The data is encrypted and decrypted using two keys in asymmetric encryption algorithms. The sender encrypts data using the public key, and the receiver decrypts data with his private key. Hybrid encryption, used in some applications, represents the combination of symmetric and asymmetric encryption algorithms. Both types of algorithms have pros and cons. [9]. Symmetric encryption algorithms are low-cost and high-efficiency. Asymmetric algorithms are complex and take more execution time, but they deliver a higher level of security. Even symmetric encryption algorithms provide a lower security level than asymmetric they are usually used in WSNs because of resource limitations. AES is high speed encryption algorithm, and it provides more security than others symmetric encryption algorithms. Because of efficient execution, low memory demands and high resiliency to attacks, AES is one of the most frequently used data protection methods in IoT systems and wireless sensor network nodes [10, 11].

This paper analyzes the performance of the AES encryption algorithm in the WSN node implemented in hardware and software. The main research objectives are:

- Implement AES encryption algorithm on two ways in the sensor node. The first, represents the hardware implementation, based on the external integrated circuit ATECC608A. The second way, software implementation, is performed inside the microcontroller's firmware;
- Design an experimental setup to measure the execution time, energy consumption, and memory demands of these two implementation methods;
- Consider these characteristics for different working conditions, including the three power supply levels (2.6 V, 3.3 V, and 5 V), and four microcontroller's working frequencies (8 MHz, 16 MHz, 32 MHz, and 64 MHz).

The experimentally obtained results presented in this paper could help to choose an appropriate encryption method according to available resources. That could reduce the development time and ensure the minimum energy consumption, extending the operation lifetime of a sensor node without reducing its performance.

The rest of the paper is organized as follows. Section 2 contains the literature review of data protection principles in WSN. Section 3 describes the basic specifications of AES and its implementation in the observed sensor node. The experimental setup for measurement is shown in Section 4, and the results are presented and compared in Section 5. Finally, Section 6 concludes the paper.

## 2. RELATED WORK

One approach for providing security of data during transmission in WSN is based on routing protocols. A Secure and Energy-Efficient Scheme for routing data known as SEER, which provides end-to-end data security and strives to minimize energy consumption, is proposed in [12]. This scheme, designed to be implemented at the hardware level, reduces the processing overhead, saves battery power, and provides a high-security level against

data interception attacks. Another approach for reducing power consumption without a negative impact on a WSN's performance is the combination of security protocol and encryption algorithm [13]. The proposed Lightweight Security Algorithm (LSA) is a hybrid algorithm created by combining the Security Protocol for Sensor Networks (SPINS) with the Secure IoT (SIT) encryption. Routing protocols are convenient security methods for sensor networks consisting of a large number of sensor nodes. Their main drawback is the inevitable trade-off between energy consumption and a high level of security.

In some applications, when it is not necessary to have a data content secret, a digital signature is used to provide data authentication, confidentiality, and integrity. These algorithms are based on asymmetric encryption. The most widely used algorithm in WSNs is the Elliptic Curve Digital Signature Algorithm. The implementation of this algorithm and its variations are examined and assessed in terms of computational cost and security level, in papers [14, 15]. Asymmetric algorithms are used for data encryption also. In the paper [16], authors present encryption approaches based on the distribution of keys using Rivest Shamir Adleman (RSA) and Elliptic Curve Cryptography (ECC) algorithms. These approaches were compared to find the minimum energy consumption in WSNs with different MAC protocols. The comparison between these two kinds of encryption, considering execution time and amount of data, is presented in [17]. The improved version of the ECC algorithm, Improved Elliptic Key Cryptography (IEKC), is presented in the paper [18]. Experimental results show that the suggested algorithm is effective regarding packet delivery ratio, end-to-end latency, throughputs, and energy consumption. The ECC encryption algorithm with the new key distribution scheme is presented in paper [19].

The survey [20] highlights the limitations of the most important strategies, which provide encryption-based security services for WSNs. As mentioned, symmetric encryption algorithms are usually used in WSNs because they demand fewer resources than asymmetric ones. Paper [21] examines symmetric encryption algorithms, Skipjack, XXTEA, and AES. A comparison of their characteristics, such as memory usage, energy consumption, and execution time, was performed using the TOSSIM simulation program. The study [22] analyzes symmetric encryption algorithms (TEA, XTEA, and RC5) for WSN security using the NS2 simulator environment. The effectiveness of four of the most common encryption algorithms (RC4, DES, AES, and RSA) on power consumption and encryption/decryption speed has been conducted in [23], at different sizes of data blocks and keys. The survey presented in [24] covers a comprehensive comparison among block ciphers and a comparison between hardware vs software solutions of the AES. The results show that the AES is the most trusted solution for WSN security. The energy consumption of various AES encryption types, including software-based AES-ECB (Electronic Codebook Mode), hardware-based AES-ECB, and hardware-based AES-CCM (Counter with CBC-MAC Mode), were measured and analyzed for different payload lengths [25]. The performance and power usage of the four different hardware and software implementations of the AES with Sparrow v3.2 sensor nodes based on the ATmega128RFA1 are compared in paper [26]. The hardware encryption module that uses the AES algorithm and the radio transceiver are integrated into this microcontroller. The findings demonstrate that implementing AES on hardware is faster and uses less energy than implementing it on software. The low-power AES algorithm, implemented on the ASIC platform, is presented in [27]. The good tradeoff between computation speed and energy efficiency of the AES was achieved by combining several optimized

components in the AES core and some modifications in the core architecture, balancing throughput, area, and power consumption. The lightweight text encryption algorithm proposed in [28] shows reduced energy usage and time complexity compared to the AES. This energy-efficient algorithm is based on the dynamic salt key (an arbitrary string of letters, numeric digits, or special characters appended to a password).

Symmetric and asymmetric algorithms are used together in some applications to improve security and reduce energy consumption. An energy-aware scheme for control of the security level, based on both symmetric and asymmetric encryption algorithms, is presented in paper [29]. The proposed scheme chooses an algorithm according to the amount of energy in the energy-harvesting sensor node. In order to increase security, a node with more energy than a predetermined threshold encrypts the data using asymmetric encryption (Elliptic Curve Integrated Encryption Scheme - ECIES). The node uses the less energy-intensive symmetric algorithm (AES) to encrypt data when the remaining energy falls below a threshold value. By applying the ideas of hybrid cryptography and clustering-based routing, paper [30] offers a methodology for enhancing energy efficiency and protecting data in WSN. The ECC algorithm was used for key generation, while data encryption/decryption was completed using a combination of AES and ECC cryptography. The proposed method shows a better speed of operation, improved security level, and a high energy efficiency level.

There are some other approaches for data encryption that tend to reduce the complexity of operation. A paper [31] describes an energy-efficient encryption algorithm with a 64-bit block length and a 128-bit key length that uses fundamental operations like exclusive-OR (XOR) and shifting. It includes simple operations and demands low-resource implementation suitable to sensor node devices. A simple and fast-stream cipher encryption system that utilizes permutation vectors as encryption keys, is introduced in the paper [32]. Experimental results show that the proposed system is very secure, simple, and energy-efficient compared to other algorithms (AES, DES, 3DES). A novel data encryption based on multiple pairwise shared keys with short length, presented in paper [33], shows much faster speed than conventional symmetric encryptions. The security of WSNs can be improved by encryption based on chaotic systems. In paper [34], the authors propose a WSN-improved identity encryption based on a hybrid chaotic mapping system applied to encrypt the data. Some other technologies, such as blockchain [35, 36], can be implemented in WSN to achieve the required security level. However, blockchain is typically resource-hungry for energy, computation, and memory. The main benefits of the different data protection methods, noted in this section, are summarized in Table 1.

**Table 1** The different data protection methods described in other papers

Ref.	Protection method		Testing environment	The observations
[12]	Rounting protocols	SEER	MATLAB	<ul style="list-style-type: none"><li>▪ Increased lifetime of sensor node</li><li>▪ Reduced the number of messages</li><li>▪ Reduced energy consumption</li></ul>
[13]		SPINS protocol + SIT encryption	MATLAB	<ul style="list-style-type: none"><li>▪ Reduced the key generation time</li><li>▪ Reduced energy consumption during transmission</li><li>▪ Improved data security</li></ul>

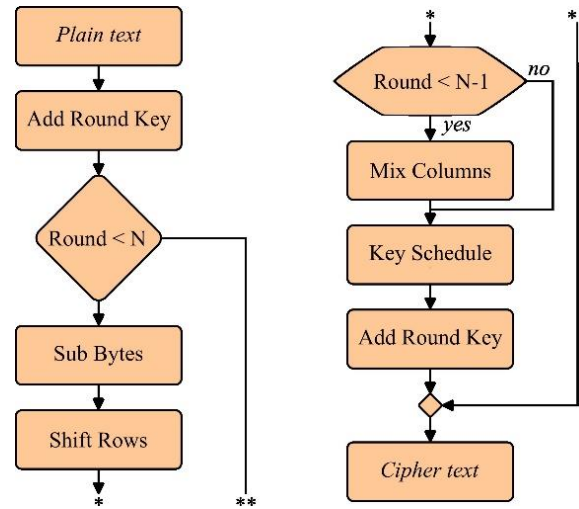
[14]	Asymmetric encryption	ASCW based on ECDSA	Sensor node with Zigbee	<ul style="list-style-type: none"> <li>Management with acceptable key length,</li> <li>Secure communication on node level,</li> <li>Reduced the cost of risk and security threats</li> </ul>
[15]		Modified ECDSA	FPGA	<ul style="list-style-type: none"> <li>Reduced computational cost,</li> <li>Increased security (compared to ECDSA)</li> </ul>
[16]		BMAC, LMAC protocols with AES, RSA, ECC	OMNet++	<ul style="list-style-type: none"> <li>Reduced energy consumption,</li> <li>Less packet lost</li> </ul>
[17]		RSA vs. ECC	Not specified	<ul style="list-style-type: none"> <li>RSA requires a less execution time, and smaller amount of stored data over ECC</li> </ul>
[18]		Improved Elliptic Key Cryptography (IEKC)	Not specified	<ul style="list-style-type: none"> <li>Effective regarding packet delivery ratio, end-to-end latency, throughputs, and energy consumptions</li> </ul>
[19]		ECC	NS2	<ul style="list-style-type: none"> <li>The new key distribution system</li> <li>Reduced the key storage</li> <li>Improved secure connectivity and resistance to attacks</li> </ul>
[21]	Symmetric encryption	Skipjack, XXTEA, AES	TOSSIM	<ul style="list-style-type: none"> <li>Skipjack – 4.7 % longer execution time, and 5 % more consumed energy,</li> <li>XXTEA – 5.1 % longer execution time, and 6 % more consumed energy,</li> <li>AES – 8.4 % longer execution time, and 13 % more consumed energy, Compared to non-encryption system</li> </ul>
[22]		TEA, XTEA, RC5	NS2	<ul style="list-style-type: none"> <li>Network throughput is inversely related to security (if the number of keys increases the network throughput decreases)</li> </ul>
[23]		RC4, DES, AES	ARM9TDMI	<ul style="list-style-type: none"> <li>Distance has a small impact on energy consumption,</li> <li>Data size has a direct impact on energy consumption</li> <li>Energy consumption (descending): AES-256, AES-192, AES-128, DES-CBC, RC4</li> </ul>
[25]		SW-AES-CB, HW-AES-ECB, HW AES-CCM	LAUNCHXL-CC1310 (48 MHz)	<ul style="list-style-type: none"> <li>Execution time rises with the size of plain text</li> <li>Current consumption: AES-ECB-SW = non AES AES-ECB-HW, 1.14 mA more than non AES</li> </ul>
[26]		AES-ECB, AES-CBC, AES-CFB, AES-CTR	AtMega128R FA1 (16 MHz)	<ul style="list-style-type: none"> <li>Execution time rises with the size of plain text</li> <li>Execution time is longer for software than hardware implementation</li> <li>Energy consumption is higher for software than hardware implementation</li> </ul>

[27]		AES (optimized S-Box)	ASIC (10 MHz)	<ul style="list-style-type: none"> <li>Proposed implementation of AES energy consumption:               <ul style="list-style-type: none"> <li>38.46 % less than Canright Sbox</li> <li>20-25 % less than LBS-CRS</li> <li>20-25 % less than DSE-CRS</li> <li>13 % less than DSE</li> </ul> </li> </ul>
[28]		Lightweight, energy-efficient secure text encryption using the dynamic salt key	Java, Intel(R) Pentium(R)	<ul style="list-style-type: none"> <li>High level of security while requiring minimal communication and computational resources,</li> <li>Reduced execution time (841 ms vs. 1066 ms – compared to AES),</li> <li>Reduced memory usage (8307 bytes vs. 13178 bytes – compared to AES)</li> </ul>
[29]	Hybrid (Asymmetric + symmetric encryption)	ECIES + AES	TelosB sensor node	<ul style="list-style-type: none"> <li>Increased both the encryption level and energy efficiency by utilizing the remaining energy, compared to other techniques</li> </ul>
[30]		ECC + AES	Not Specified	<ul style="list-style-type: none"> <li>A better speed of operation, improved security level, and a high energy efficiency level</li> </ul>
[31]	Other methods	Energy Efficient Encryption based on exclusive-OR (XOR) and shifting	Verilog	<ul style="list-style-type: none"> <li>Low-resource implementation, enough security decreases the power consumption compared to HIGHT</li> </ul>
[32]		Synchronous Dynamic Encryption System (SDES)	Not Specified	<ul style="list-style-type: none"> <li>Less energy for encryption and decryption, compared to AES, DES, and 3DES</li> </ul>
[34]		Improved identity encryption, based on chaotic systems	Not Specified	<ul style="list-style-type: none"> <li>Shorter execution time, and less memory resources required, compared to RC5, and CWSN</li> </ul>

### 3. IMPLEMENTATION OF AES IN A SENSOR NODE

AES was proposed by the National Institute of Standards and Technology in 2002. This algorithm was initially known as "Rijndael," after the two Belgian cryptographers who created it. After the National Security Agency approved it, AES has become a recommended security standard of data encryption [37]. AES is widely used today, and many libraries have been created for it in a variety of programming languages, including Python, Java, C, and C++.

This encryption algorithm is well-known and described in the literature. Therefore, this paper will note the main part of the algorithm without a detailed explanation. AES is a block cipher algorithm that encrypts and decrypts a block of plaintext or cipher text using symmetrical keys [38]. AES operates on a block data size of 128 bits, organized as a four-by-four array of bytes, named a state array. A key can be 128, 192, or 256 bits in size, with 10, 12, or 14 rounds, respectively. Fig. 1 represents the structural algorithm of the AES.



**Fig. 1** The structural algorithm of the AES

*Add Round Key* is performed inside every round and once at the beginning, before the first round. This operation includes the XOR operation between the input data and the key for the current round.

*Sub Bytes* makes a non-linear substitution step replacing the each byte of the current state array by corresponding value from the lookup table (S-Box).

*Mix Columns* is a mixing operation that uses a linear transformation to combine the four bytes in each of the state array's columns. This process is included in all rounds except the final one.

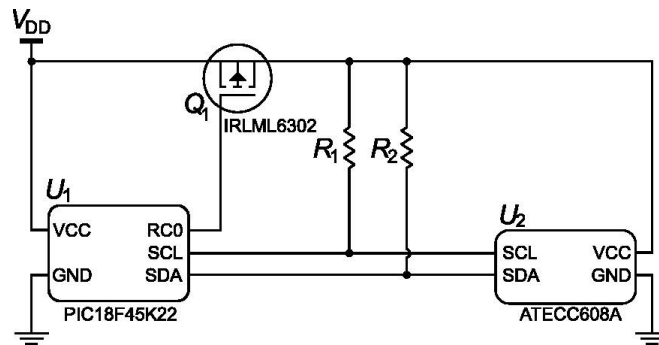
*Key Schedule* process prepares the key for the next round. It includes the three steps: Rotation inside the columns and S-Box and R-con substitutions.

The AES encryption/decryption algorithm is standardized efficient hardware architecture, and implementation methods are the topics that many researchers are focusing on. The hardware and software implementation of the AES encryption algorithm on the WSN node is made in this paper. The microcontroller PIC18F45K22 is used as the processing unit of the sensor node, which power supply voltage is in the range from 2.3 V to 5.5 V. This microcontroller was selected because of well-known architecture and previously experience of the authors. It has an internal oscillator with maximal operating frequency up to 64 MHz, 32 Kb of programmable memory, UART, I<sup>2</sup>C and SPI communication interfaces. It makes him appropriate platform for sensor node in various applications [39].



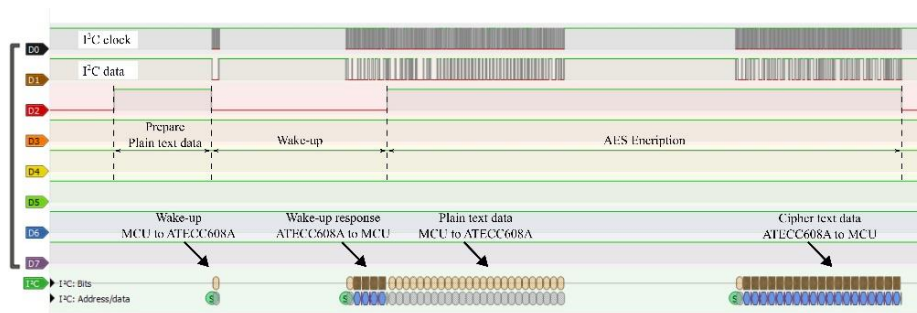
### 3.1. Hardware implementation

The hardware implementation of the AES, in sensor node, is realized using the external integrated circuit ATECC608A. This integrated circuit has a low energy consumption, with the required power supply voltage from 2 V to 5.5 V, and integrates different encryption algorithms such as AES, SHA-256, the random number generator, and the Elliptic Curve Digital Signature. Up to 16 different keys for AES encryption can be stored in ATECC608A. Data exchange between this circuit and the microcontroller is performed via I<sup>2</sup>C communication, with a frequency of the clock signal up to 1 MHz. This integrated circuit was selected because of possibility to encrypt data by using several encryption algorithms. The other reasons are low energy consumption, low-cost, and availability on the market. The microcontroller can turn on/off ATECC608A by controlling the PMOS transistor IRLML6302, as presented on the electronic circuit in Fig. 2. This transistor provides the possibility to turn off the ATECC608A when the encryption process is outperforming to reduce energy consumption.



**Fig. 2** Circuit design of the hardware encryption implementation composed of the microcontroller and integrated circuit ATECC608A

It is necessary to configure the ATECC608A before using it. It includes setting the AES operation and writing the key. The hardware encryption process has three steps, as presented in Fig. 3: data preparation, wake-up, and encryption. The data captured by a



**Fig. 3** The hardware encryption process – data flow between the microcontroller (MCU) and ATECC608A

sensor node is transformed into 16 bytes of plain text data during the data preparation step. In the second step, the microcontroller turns on the ATECC608A, sends the wake-up signal to him, and waits for the appropriate response. The wake-up process takes about 1500 us. After that, the microcontroller sends the 128-bit plain text data to the ATECC608A. The ATECC608A takes about 2030 us to perform encryption, and cipher text data can be read by microcontroller after that.

### 3.2. Software implementation

The second way for data encryption in the sensor node is using the algorithm implemented as the microcontroller firmware [40]. In this case, the encryption algorithm and appropriate key are stored in microcontroller memory. The software encryption process includes the same steps as the hardware, except the wake-up operation. Data preparation is the same as in the hardware process, while the data encryption is simplified, because it does not include additional circuits, and the whole process is performed inside the microcontroller.

## 4. EXPERIMENTAL SETUP

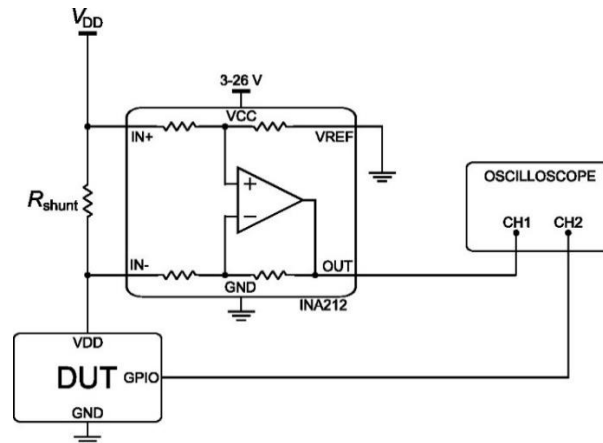
An execution time and energy consumption for both, hardware and software implementation of the AES encryption algorithms were examined in this paper. Time measurement was performed using the oscilloscope. The oscilloscope can accurately measure pulse duration. The microcontroller creates a pulse on its GPIO pin, whose voltage level denotes the start/end of a specified operation. All microcontroller unused pins are configured as output, and set to low voltage levels to prevent unnecessary energy consumption [41]. The energy consumed during encryption can be calculated by Eq. (1) as

$$E = P \cdot t = I \cdot V \cdot t = Q \cdot V, \quad (1)$$

where  $E$  denotes energy,  $P$  is power,  $t$  is the previously mentioned time,  $I$  represent the current sink by the sensor node, and  $V$  is the power supply voltage.  $Q$  represents the total amount of electricity flow through the sensor node. Because the current is not constant, the consumed energy was calculated by Eq. (2) as

$$E = Q \cdot V = V \cdot \int_0^t I \cdot dt, \quad (2)$$

where value of the integral represents the total amount of electricity flow through the sensor node during the encryption process. Current intensity through the sensor node can be measure by different methods, such as shunt resistor, voltage converter based on current mirror [42], circuits based on the constant current source [43], and others [44]. The circuit based on shunt resistor and INA212 integrated circuit, presented in Fig. 4, was used for current measurement. This methods provide the high precision and it has a low impact on load operating voltage.



**Fig. 4** Circuit design for current measurement. (DUT – Device Under Test is the circuit shown in Fig. 2 in hardware, and the microcontroller in software implementation of the AES).

The value of  $R_{shunt}$  needs to be sufficiently small to avoid its influence on the circuit's functionality. However, the same value must be sufficiently large to produce a voltage drop that is precisely detectable. For this purpose, the resistor of 100 m $\Omega$ , whose tolerance is 1 %, was selected.

The current measurement process is performed through the following steps:

- 1) The shunt resistor is connected between IN+ and IN- pins of INA212;
- 2) IN+ pin and IN- pin are connected to the power supply voltage and power supply pin of the device, respectively;
- 3) OUT pin of INA212 is connected to one channel of the oscilloscope (CH1);
- 4) One GPIO pin of the device under test (microcontroller in our case) is connected to another channel of the oscilloscope (CH2).
- 5) The duration of the specific part of the encryption is measured on channel 2 (time between two consecutive changes);
- 6) The current is measured on channel 2, for the same time interval as 5).

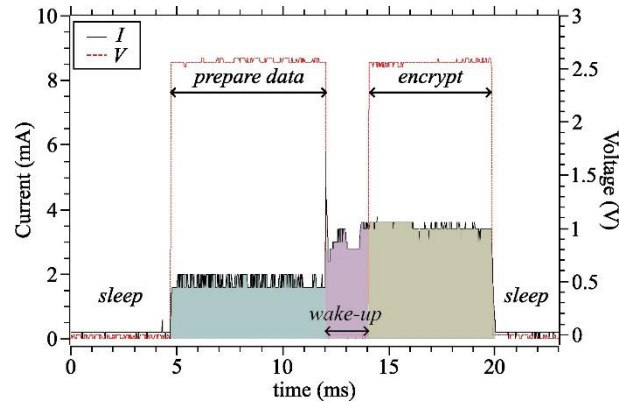
The experimental setup for current measurement based on INA212 operates at the same voltage level as a sensor node. So, it can be made using a microcontroller instead of an oscilloscope and installed in a sensor node, providing real-time consumption measurement. It can be useful for measuring the consumption of the entire WSN, and the lifetime prediction of the specific sensor node.

## 5. RESULTS

The execution time and energy consumption are measured and calculated for three different levels of power supply voltages at four different microcontroller's working frequencies. Although the microcontroller and ATECC608A can operate at 2.3 V and 2 V, respectively, the minimal power supply voltage level is 2.6 V. This voltage level is necessary to provide reliable control of the transistor, which turns on/off the ATECC608A. Two other

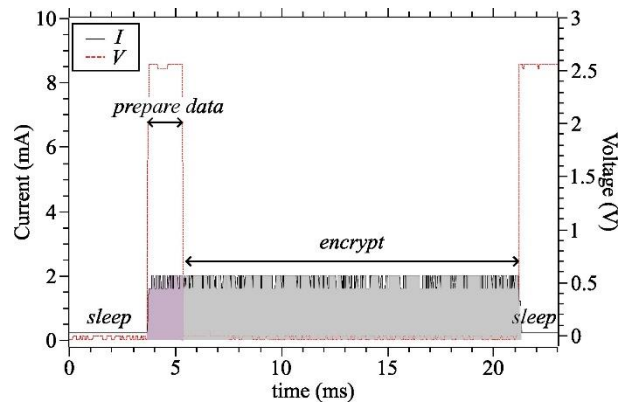
power supply voltages considered in this paper are 3.3 V and 5 V because they are the most common in battery-powered sensor nodes, and most radio transceiver modules operate at these voltages. The selected working frequencies are 8 MHz, 16 MHz, 32 MHz, and 64 MHz, which is the highest working frequency of the used microcontroller.

Fig. 5 shows the waveform of current flow through the sensor node during the hardware encryption process, and the voltage level of the selected microcontroller GPIO pin used to measure the time needed for the specified operation.



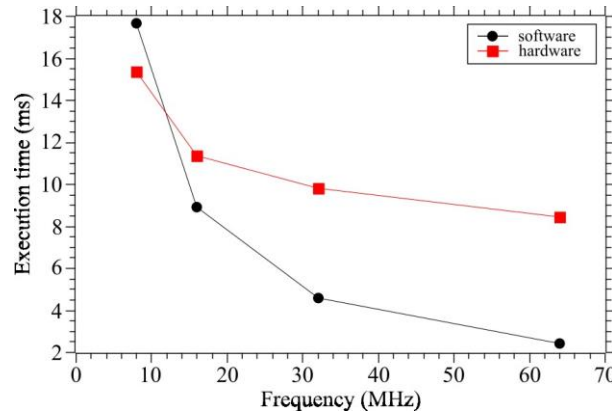
**Fig. 5** Current consumption during the hardware encryption process at constant supply voltage. The duration of each process is indicated by arrows, and the colored areas represent the amount of passed charge

The measured quantities, in the case of the software encryption process, are presented in Fig. 6. In both cases, the power supply voltage is 2.6 V, and the microcontroller working frequency is 8 MHz.



**Fig. 6** Current consumption during the software encryption process at constant supply voltage. The duration of each process is indicated by arrows, and the colored areas represent the amount of passed charge

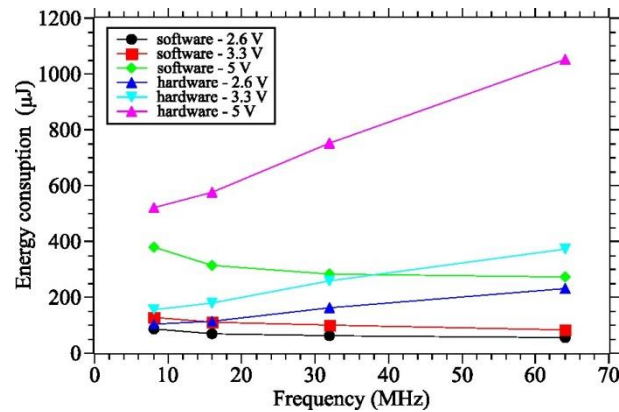
The microcontroller is in sleep mode before and after the encryption process to minimize energy consumption during the inactive state (no measurement or sending data). The total execution time of the hardware encryption process is calculated as the sum of times needed to prepare data, ATECC608A wake-up, and encrypt data. The software encryption process does not include time for wake-up. Fig. 7 shows the dependence of the execution time for both implementations, at noted four different frequencies.



**Fig. 7** Execution time vs. operating frequency for hardware and software implementation of the AES

The execution time decreases with the increase of the microcontroller working frequency. These results are expected, considering that the microcontroller needs less time to do the same operation at a higher working frequency. The reduction of the execution time is more intensive for software than for hardware encryption. When the working frequency is doubled, the execution time of software encryption is reduced by almost 50 %. A similar trend is noted for hardware encryption, but the reduction of execution time is smaller: 25 %, 15 %, and 13 % for frequency changes 8 MHz to 16 MHz, 16 MHz to 32 MHz, and 32 MHz to 64 MHz, respectively. The reason for that is the fixed time interval, which is necessary to ATECC608A perform specific operations (wake-up, encryption), and they do not depend on the microcontroller working frequency. The hardware encryption process takes less time at smaller values of the observed frequencies (approximately up to 12 MHz). Therefore, if the execution time is the primary goal, and the working frequency is smaller than 12 MHz, the hardware encryption process offers better results, while the software encryption operates faster at higher values of working frequencies.

The energy consumption for the observed working parameters is presented in Fig 8. In the case of the same power supply voltage and equal working frequencies, the software encryption process demands less energy than hardware because it does not include additional components and circuits. The both processes, hardware and software, consume more energy if the power supply voltage is higher at the same microcontroller working frequency. This kind of change is expected, considering that energy consumption is calculated according to Eq. (2). Also, the difference in the consumed energy in these two processes is higher at increased power supply voltage.



**Fig. 8** Energy consumption vs. operating frequency for hardware and software implementation of the AES

In the case of the hardware encryption, the consumed energy rises with the increasing working frequency at the constant power supply voltage. The changes are more intense at the highest working frequencies. The main contribution to the increased power consumption is the increase of the current intensity, which is more significant at higher frequencies than the reduction of the execution time. The decreasing of the execution time in the software encryption has more influence on energy consumption than increasing the current intensity at the higher working frequencies. Therefore, the energy consumption decreases slightly if the software encryption is applied. The results in Table 2 show the difference in consumed energy between these implementation methods.

**Table 2** The comparison of energy consumption of hardware and software implementation methods

f (MHz)	The difference in energy consumption, hardware vs. software encryption (%)		
	Voltage (V)		
	2.6	3.3	5
8	+16	+17	+27
16	+40	+39	+45
32	+60	+61	+61
64	+75	+75	+78

The results show that hardware encryption consumes more energy than software, for every working frequencies and power supply voltages. For example, at frequency of 8 MHz and 2.6 V power supply voltage, the hardware encryption process consumes 16 % more energy than software.

In addition, the available memory must be considered when implementing firmware because the microcontroller used in the sensor node has limited space of the programmable memory. In this case, the selected microcontroller has 4 kB memory space designed for the user's instructions. For the hardware and software implementation, 4.58 % and 10.13 % of programmable memory are occupied, respectively. Therefore,

hardware implementation is a better choice when it is necessary to ensure data security with minimal memory usage.

The choice of the implementation method does not depend on the monitored physical parameters and a sensor node's application. The frequency of measurement (execution time), available energy, and microcontroller resources (size of the programmable memory and working frequency) determine which encryption method should be applied. The only constraint for hardware implementation is a microcontroller inside a sensor node equipped with an I<sup>2</sup>C interface, and enough programmable memory for software implementation.

## 6. CONCLUSION

AES is the symmetric encryption algorithm and one of the most frequently used ways to protect data in WSN. It is not energy-demanding and provides a high level of security. Implementation of this algorithm inside the sensor node is possible through hardware or software processes. The hardware implementation includes the additional integrated circuit, which takes data from the microcontroller and encrypts them. On the other hand, the microcontroller performs data encryption in the software implementation. This paper compares the performances of these two implementation ways, considering the execution time, energy consumption, and memory demands at three levels of power supply voltages, and the four microcontroller working frequencies. The sensor node is based on microcontroller PIC18F45K22, and integrated circuit ATECC608A is used in the hardware encryption approach.

The experimentally obtained results show that encryption implemented in hardware takes less execution time for frequencies up to 12 MHz, while the software encryption is faster at higher frequencies. So, if the execution time is the most important during the sensor node operation, and there is no energy constraints, the software encryption implementation is the better solution than hardware. The AES encryption algorithm implemented in software requires less energy than another one realized by hardware components. This encryption process should be applied inside the sensor node if energy saving is the primary goal, such as sensor nodes powered by battery or energy-harvesting power supply. On the other hand, the hardware implementation occupies less programmable memory, which is significant for sensor nodes built on microcontrollers with limited memory space, or sensor nodes that contain computationally demanding algorithms for data analysis.

Our future research will focus on the implementation of hardware and software symmetric encryption algorithms using different microcontrollers and specific integrated circuits (as ATECC608A) to find a less demanding solution for resourced-constrained sensor nodes. Another research direction will include analyzing the energy consumption and other parameters of various radio modules to find the most effective transmission method and minimize energy consumption during data transmission. The obtained result of energy consumption during encryption and data transmission may help to develop the appropriate energy-harvesting power supply.

**Acknowledgement:** *This work was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia [grant number 451-03-65/2024-03/200102].*

## REFERENCES

- [1] P. Asghari, A. M. Rahmani and H. H. S. Javadi, "Internet of Things Applications: A Systematic Review", *Computer Networks*, vol. 148, pp 241-261, 2019.
- [2] A. H. Abdulwahid and M. J. Salih, "Wireless Sensor Networks Applications, Challenges, and Security Requirements", In *Proceedings of 2nd International Multi-Disciplinary Conference Theme: Integrated Sciences and Technologies, IMDC-IST 2021*, Sakarya, Turkey, 2022, pp. 1-11.
- [3] K. Dionisis, N. Christos, V. Dimitrios and K. Grigorios, "Applications of Wireless Sensor Networks: An Up-to-Date Survey", *Appl. Syst. Innov.*, vol. 3, no. 1, p. 14, 2020.
- [4] Z. Huanan, X. Suping and W. Jiannan, "Security and Application of Wireless Sensor Network", *Procedia Comput. Sci.*, vol. 183, pp. 486-492, 2021.
- [5] F. Gutiérrez-Portela, F. Almenárez-Mendoza, L. Calderón-Benavides and E. Romero-Riaño, "Security Perspective of Wireless Sensor Networks", *Rev. UIS Ing.*, vol. 20, pp. 189-202, 2021.
- [6] M. Bhalla, N. Pandey and B. Kumar, "Security Protocols for Wireless Sensor Networks", In *Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, Greater Noida, India, 2015, pp. 1005-1009.
- [7] J. Sen, *Wireless Sensor Networks - Design, Applications and Challenges*. IntechOpen, 2023, Chapter 2.
- [8] M. A. Al-Shabi, "A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security", *Int. J. Sci. Res. Publ. (IJSRP)*, vol. 9, p. 8779, 2019.
- [9] P. P. Santoso, E. Rilvani, A. B. Trisnawan, K. Adiyarta, D. Napitupulu, T. Sutabri and R. Rahim, "Systematic Literature Review: Comparison Study of Symmetric Key and Asymmetric Key Algorithm", In *Proceedings of the 2nd Nommensen International Conference on Technology and Engineering*, Medan, Indonesia, 2018, vol. 420, p. 012111.
- [10] C. Zhang, Y. Jia, L. Zhu and Z. Zhang, "Research on Simple Power Consumption Based on AES Algorithm", In *Proceedings of the 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, Changchun, China, 2023, pp. 1883-1886.
- [11] E. S. Selvapriya and L. Suganthi, "Design and Implementation of Low Power Advanced Encryption Standard Cryptocore Utilizing Dynamic Pipelined Asynchronous Model", *Integration*, vol. 93, p. 102057, 2023.
- [12] M. S. Khan, N. M. Khan, A. Khan, F. Aadil, M. Tahir, M. Sardaraz, "A Low-Complexity, Energy-Efficient Data Securing Model for Wireless Sensor Network Based on Linearly Complex Voice Encryption Mechanism of GSM Technology", *Int. J. Distrib. Sens. Netw.*, vol. 17, no. 5, pp. 1-13, 2021.
- [13] N. Mahlake, T. E. Mathonsi, D. D. Plessis and T. Muchenje, "A Lightweight Encryption Algorithm to Enhance Wireless Sensor Network Security on the Internet of Things", *J. Commun.*, vol. 18, pp. 47-57, Jan. 2023.
- [14] R. Qazi, K. N. Qureshi, F. Bashir, N. U. Islam, S. Iqbal and A. Arshad, "Security Protocol using Elliptic Curve Cryptography Algorithm for Wireless Sensor Networks", *J. Ambient Intell. Human Comput.*, vol. 12, pp. 547-566, 2021.
- [15] K. Venkataraman and T. Sadasivam, "FPQA Implementation of Modified Elliptic Curve Digital Signature Algorithm", *FU: Elec. Energ.*, vol. 32, no. 1, pp. 129-145, 2019.
- [16] M. Tropea, M. G. Spina, F. De Rango and A. F. Gentile, "Security in Wireless Sensor Networks: A Cryptography Performance Analysis at MAC Layer", *Future Internet*, vol. 14, no. 5, p. 145, 2022.
- [17] Y. Cheng, Y. Liu, Z. Zhang and Y. Li, "An Asymmetric Encryption-Based Key Distribution Method for Wireless Sensor Networks", *Sensors*, vol. 23, p. 6460, 2023.
- [18] P. Ramadevi, S. Ayyasamy, Y. Suryaprakash, C. Anilkumar, S. Vijayakumar and R. Sudha, "Security for Wireless Sensor Networks using Cryptography", *Meas. Sensors*, vol. 29, p. 100874, 2023.
- [19] S. Sinha and S. Aggarwal, "Cryptographic Algorithms for Security in Wireless Sensor Networks", In *Proceedings of the 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, London, United Kingdom, 2022, pp. 111-117.
- [20] H. Hayouni, M. Hamdi and T. -H. Kim, "A Survey on Encryption Schemes in Wireless Sensor Networks", In *Proceedings of the 2014 7th International Conference on Advanced Software Engineering and Its Applications*, Hainan, China, 2014, pp. 39-43.
- [21] D. Murat, "Comparison of Encryption Algorithms in Wireless Sensor Networks", *ITM Web Conf.*, vol. 22, p. 01005, 2018.
- [22] S. N. Hussein, A. H. Obaid and A. Jabbar, "Encryption Symmetric Secret Key in Wireless Sensor Network Using AES Algorithm", *Iraqi J. Sci.*, vol. 63, pp. 5037-5045, 2022.
- [23] M. A. Al Sibahee, S. Lu, Z. A. Hussien, M. A. Hussain, K. A.-A. Mutlaq and Z. A. Abduljabbar, "The Best Performance Evaluation of Encryption Algorithms to Reduce Power Consumption in WSN", In



- Proceedings of the 2017 International Conference on Computing Intelligence and Information System (CIIS), 2017, pp. 308-312.
- [24] I. K. Dutta, B. Ghosh and M. Bayoumi, "Lightweight Cryptography for Internet of Insecure Things: A Survey", In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2019, pp. 0475-0481.
  - [25] C.-W. Hung and W.-T. Hsu, 2018. "Power Consumption and Calculation Requirement Analysis of AES for WSN IoT", *Sensors*, vol. 18, p. 1675. 2018.
  - [26] C. Panait and D. Dragomir, "Measuring the Performance and Energy Consumption of AES in Wireless Sensor Networks", In Proceedings of the 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), 2015, pp. 1261-1266.
  - [27] M. -H. Dao, V. -P. Hoang, V. -L. Dao and X. -T. Tran, "An Energy Efficient AES Encryption Core for Hardware Security Implementation in IoT Systems", In Proceedings of the 2018 International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh City, Vietnam, 2018, pp. 301-304.
  - [28] V. Elamurugu, and D. J. Evanjaline, "An Efficient and Secure Text Encryption Scheme for Wireless Sensor Network (WSN) Using Dynamic Key Approach", *Int. J. Comput. Netw. Appl.*, vol. 8, pp. 788-794, 2021.
  - [29] J. M. Kim, H. S. Lee, J. Yi and M. Park, "Power Adaptive Data Encryption for Energy-Efficient and Secure Communication in Solar-Powered Wireless Sensor Networks", *J. Sensors*, vol. 2016, p. 2678269, 2016.
  - [30] S. Urooj, S. Lata, S. Ahmad, S. Mehruz and S. Kalathil, "Cryptographic Data Security for Reliable Wireless Sensor Network", *Alexandria Eng. J.*, vol. 72, pp. 37-50, 2023.
  - [31] A. B. Karupiah, Dr. S. Rajaram, "Energy Efficient Encryption Algorithm for Wireless Sensor Network", *Int. J. Eng. Res. Technol. (IJERT)*, vol. 1, pp. 1-7, May 2012.
  - [32] H. Soliman and M. Omari, "Application of Synchronous Dynamic Encryption System (SDES) in Wireless Sensor Networks", *Int. J. Netw. Secur.*, vol.3, pp.160-171, Sept. 2006.
  - [33] L. Harn, C. -F. Hsu, Z. Xia and Z. He, "Lightweight Aggregated Data Encryption for Wireless Sensor Networks (WSNs)", *IEEE Sensors Lett.*, vol. 5, pp. 1-4, April 2021.
  - [34] H. Zhou and H. Bi, "Wireless Sensor Network Security Based on Improved Identity Encryption", *Sci. Program.*, vol. 2022, p. 2308825, 2022.
  - [35] M. S. Jabor, A. S. Azez, J. C. Campelo and A. Bonastre Pina, "New Approach to Improve Power Consumption Associated with Blockchain in WSNs", *PLoS ONE*, vol. 18, p. 0285924, 2023.
  - [36] A. E. Guerrero-Sanchez, E. A. Rivas-Araiza, J. L. Gonzalez-Cordoba, M. Toledano-Ayala and A. Takacs, "Blockchain Mechanism and Symmetric Encryption in A Wireless Sensor Network", *Sensors*, vol. 20, 2020.
  - [37] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback and J. Dray, "Advanced Encryption Standard (AES)", *Federal Inf. Process. Stds. (NIST FIPS)*, Nov. 2001.
  - [38] J. Daemen and V. Rijmen, *The Block Cipher Rijndael: Smart Card Research and Applications*, Springer Berlin Heidelberg, 2000, pp. 277-284.
  - [39] M. Djordjević and D. Danković, "A Smart Weather Station Based on Sensor Technology", *FU: Elec. Energ.*, vol. 32, no. 2, pp. 195-210, 2019.
  - [40] A. Takieldeem and A. M. Fanni, "Implementation of AES Algorithm in MicroController Using PIC18F452", *J. Comput. Eng. (IOSR-JCE)*, vol. 15, pp. 35-38, 2013.
  - [41] B. Ivey, *Low-Power Design Guide*, Microchip Technology Inc., 2011.
  - [42] C. Antonopoulos, A. Prayati, T. Stoyanova, C. Koulamas and G. Papadopoulos, "Experimental Evaluation of a WSN Platform Power Consumption", In Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing, Rome, Italy, 2009, pp. 1-8.
  - [43] A. Di Nisio, T. Di Noia, C. G. C. Carducci and M. Spadavecchia, "High Dynamic Range Power Consumption Measurement in Microcontroller-Based Applications", *IEEE Trans. Instrum. Meas.*, vol. 65, no. 9, pp. 1968-1976, Sept. 2016.
  - [44] Z. Nakutis, "Embedded Systems Power Consumption Measurement Methods Overview", *Matavimai*, vol. 2, no. 44, pp. 29-35, 2009