



Comparative Analysis of Sensor Reading Accuracy Between Arduino and ESP32 Microcontrollers for TDS Sensor Measurements

Abdul Rahman^{1,*}, Agus Ulinuha¹

¹ Department of Electrical Engineering, Universitas Muhammadiyah Surakarta, Jawa Tengah, Indonesia

Article Info

Article history:

Received Jan 07, 2026

Revised Feb 13, 2026

Accepted Mar 11, 2026

OnlineFirst Mar 27, 2026

Keywords:

Arduino UNO

ESP32

TDS Sensor

ABSTRACT

Purpose of the study: This study aims to compare the performance of Arduino UNO and ESP32 microcontrollers in TDS sensor measurements, evaluating reading accuracy, response speed, and reading stability to recommend the most suitable microcontroller platform for water quality monitoring applications.

Methodology: Tools & Hardware: TDS sensor, Arduino UNO (ATmega328P, 10-bit ADC), ESP32 NodeMCU (12-bit ADC), TDS meter (reference instrument). Methods: Waterfall methodology, literature review, direct observation, potentiometer testing, and voltage measurement. Water samples tested at 101, 201, 300, 406, and 515 PPM concentrations, each with three repetitions at room temperature (~20–21°C).

Main Findings: Arduino UNO achieved average voltage errors of 0.21%–2.12% across all TDS concentrations, compared to ESP32's 13.1%–54.32%. ESP32 misread 101 PPM as 48 PPM (54.32% error). Arduino UNO's standard deviation was 0.064 ADC versus ESP32's 2.019 ADC 31.5 times more variable confirming Arduino UNO's superior accuracy and stability for TDS sensor measurements.

Novelty/Originality of this study: This Arduino UNO achieved average voltage errors of 0.21%–2.12% across all TDS concentrations, compared to ESP32's 13.1%–54.32%. ESP32 misread 101 PPM as 48 PPM (54.32% error). Arduino UNO's standard deviation was 0.064 ADC versus ESP32's 2.019 ADC — 31.5 times more variable confirming Arduino UNO's superior accuracy and stability for TDS sensor measurements.

This is an open access article under the [CC BY](https://creativecommons.org/licenses/by/4.0/) license

© 2026 by the author(s)



Corresponding Author:

Abdul Rahman,

Faculty Electrical Engineering, University Muhammadiyah of Surakarta,

Ahmad Yani Road, Kartasura, Sukoharjo, Jawa Tengah, 57169, Indonesia

Email: abdulrahmansal3@gmail.com

1. INTRODUCTION

Advancements in information and communication technology have fostered various innovations that simplify human activities, including microcontroller technology. Technology is defined as the implementation of scientific knowledge embodied in devices, systems, and methods to facilitate human life. A microcontroller is a compact, single-chip integrated computer system that combines a central processing unit (CPU), memory, and input/output modules into a single unit. This technology enables efficient data processing from various sensors, making it widely applicable in fields such as industry, agriculture, healthcare, and research [1], [2].

In this era of digitalization, the demand for accurate, fast, and stable monitoring systems has increased significantly, particularly in applications involving sensor readings to monitor vital parameters such as TDS, temperature, humidity, and water quality. Sensor-based monitoring systems require a microcontroller platform

capable of delivering optimal performance in terms of data reading accuracy, responsiveness, and overall system stability. Arduino Uno and ESP32 are two popular microcontrollers extensively used in the development of sensor-based monitoring systems. The Arduino Uno utilizes an 8-bit architecture with an ATmega328P processor operating at 16 MHz and a 10-bit ADC resolution, making it an economical and user-friendly choice for basic applications. On the other hand, the ESP32 features a 32-bit dual-core architecture with speeds up to 240 MHz, a 12-bit ADC resolution, and integrated WiFi and Bluetooth connectivity, offering higher processing capabilities and comprehensive connectivity features [3]-[7].

The fundamental differences in characteristics between these two platforms specifically in architecture, processing speed, and Analog-to-Digital Converter (ADC) resolution raise important questions regarding which platform is more optimal for specific monitoring applications. In the context of monitoring systems, the TDS sensor is a critical component that requires accurate and stable data acquisition to provide reliable information, particularly in applications such as water quality monitoring, hydroponic systems, and industrial food quality control [8]-[11]. This study aims to compare the performance of Arduino Uno and ESP32 microcontrollers in TDS sensor readings based on three primary parameters: reading accuracy, response speed, and reading stability. The results of this comparison are expected to provide practical recommendations for monitoring system developers in selecting the most suitable microcontroller for their specific application needs.

2. RESEARCH METHOD

This study employs an experimental quantitative research approach with a comparative design. The research compares the performance of two different microcontroller platforms (Arduino UNO and ESP32) in measuring Total Dissolved Solids (TDS) using the same sensor under controlled laboratory conditions. The experimental design follows a systematic testing protocol to ensure the validity and reliability of the comparative analysis. The research adopts the Waterfall methodology as the development framework, which consists of sequential phases: literature review, hardware construction, data acquisition, and analysis. This structured approach ensures systematic progression and enables early detection of potential errors during the development process. The study involves the following variables [8].

Independent Variables: Microcontroller type (Arduino UNO vs. ESP32), TDS concentration levels (101, 201, 300, 406, and 515 PPM). **Dependent Variables:** Voltage output from TDS sensor (V), ADC reading values, TDS measurement values (PPM), Standard deviation, Measurement error percentage. **Controlled Variables:** TDS sensor type, Ambient temperature (20-21°C), Measurement repetition (3 times per test). In this study, the author employs a literature review and the waterfall methodology. A literature review is a systematic research method used to collect, identify, evaluate, and analyze documents relevant to the research topic. In the context of this comparative study of TDS sensors using two different microcontrollers (Arduino UNO and ESP32), the literature review serves as the theoretical and empirical foundation for understanding the characteristics of each component and the appropriate testing parameters.

This research utilizes the waterfall method to ensure the process remains sequential and orderly, allowing for efficient identification of any failures or errors during the development process. The method applied in this study involves direct observation of the sensor outputs used to measure and compare the TDS sensor performance with the Arduino UNO and ESP32 NodeMCU microcontrollers. The research flowchart is illustrated in Figure 1.

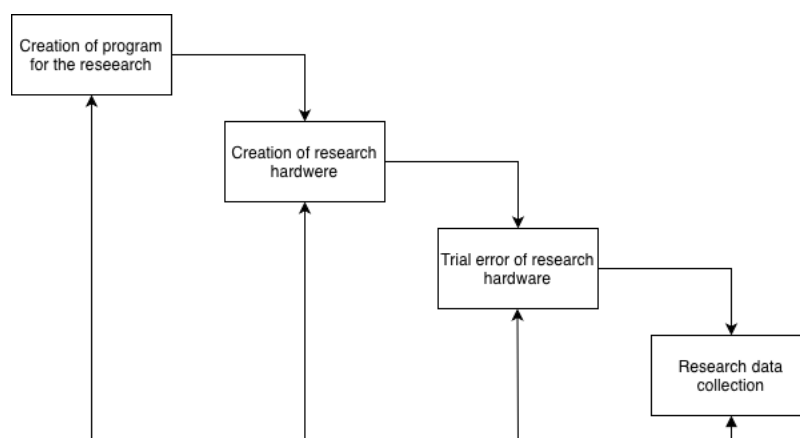


Figure 1. Waterfall Method

Based on Figure 1, the stages are detailed as follows: 1) Literature Review and Reference Gathering: Collecting and understanding references related to the design, construction, and testing of the device.

Hardware Construction: This stage involves the mechanical and electrical assembly of the device; 2) Data Acquisition: Conducting research data collection by using measurement tools to compare values; 3) Creation of journal publications : Analyzing the research data obtained from the measurements against.

Arduino Uno is an ATmega328P-based microcontroller development board developed by Arduino.cc. Arduino Uno is an open-source platform that allows users to create interactive electronic projects easily. This board is designed for beginners and professionals who want to develop embedded systems with a simple and intuitive approach. The Arduino UNO has a storage capacity of 32Kb and has an 8-bit resolution. The ESP32 is an advanced microcontroller with built-in Wi-Fi and Bluetooth connectivity features, making it particularly suitable for Internet of Things (IoT)-based applications such as Off-grid solar PV systems with solar trackers. In this system, the ESP32 acts as the main control center that integrates various sensors and actuators, such as LDR sensors to detect light intensity, high-torque servo motors to drive solar panels, and relays to remotely control electrical loads. TDS sensor is an electronic device that functions to measure the total amount of dissolved solids in a solution. Dissolved solids include a variety of minerals, salts, metals, cations, and anions dissolved in water. TDS measurement is essential in a wide range of applications such as drinking water quality monitoring, hydroponic systems, aquariums, and industrial water treatment [12]-[16].

3. RESULTS AND DISCUSSION

This study aims to conduct a comparative analysis of the performance of Total Dissolved Solids (TDS) sensors when integrated with two different microcontroller platforms: the Arduino UNO and the ESP32. The comparison was performed by measuring parameters that reflect the quality of the measurement system, including accuracy, voltage (V), and reading stability. The selection of these two microcontrollers was based on fundamental differences in their technical specifications, particularly the Analog-to-Digital Converter (ADC) resolution, which theoretically influences the quality of data acquisition from analog sensors such as the TDS sensor [17]-[19].

In this research process, measurements were carried out to observe the results of the TDS sensor used in the comparative analysis between the Arduino UNO and ESP32 microcontrollers. These measurements utilized a TDS meter as a reference tool to provide accurate measurement results. The calculations used to determine the average error and accuracy are presented in Equations 1 and 2.

$$\text{Average error} = \frac{(\text{Reference Value} - \text{Measured Value})}{\text{Measured Value}} \times 100\% \quad \dots (1)$$

$$\text{Accuracy} = 100\% - \text{Average error} \quad \dots (2)$$

3.1. TDS Sensor Potentiometer Data

This section presents the analog output data or potentiometer values generated by the TDS sensor when connected to the Arduino Uno and ESP32. The observation focuses on the stability of the voltage readings from the TDS sensor. The difference in Analog-to-Digital Converter (ADC) resolution between the Arduino Uno (10-bit) and the ESP32 (12-bit) serves as the primary parameter in evaluating the level of initial detection precision before the data acquisition process is conducted [20].

TDS Sensor Potentiometer Data on Arduino UNO

Testing was performed to evaluate the performance of the TDS sensor using the Arduino UNO microcontroller at four different voltage levels (25%, 50%, 75%, and 100% of the full scale). Each level was tested three times to validate the consistency and repeatability of the system.

Table 1. TDS Sensor Potentiometer Data on Arduino UNO

No.	Position (%)	Ref (V)	Step	Arduino UNO			
				ADC	V	StdDev (ADC)	ERROR (%)
1	25	1.256	1	256	1.2561	0.04	0.0080
		1.256	2	256	1.2561	0.08	0.0080
		1.256	3	256	1.2561	0.04	0.0080
	Average		256	1.2561	0.0533	0.0080	
2	50	2.51	1	517	2.5269	0	0.6733
		2.51	2	517	2.5269	0	0.6733
		2.51	3	517	2.5269	0	0.6733
	Average		517	2.5269	0	0.6733	
3	75	3.751	1	774	3.783	0	0.8531

No.	Position (%)	Ref (V)	Step	Arduino UNO			
				ADC	V	StdDev (ADC)	ERROR (%)
4	100	3.751	2	774	3.783	0	0.8531
		3.751	3	774	3.783	0	0.8531
		Average		774	3.783	0	0.8531
		4.945	1	1018	4.9757	0.21	0.6208
		4.945	2	1018	4.9757	0.22	0.6208
		4.945	3	1018	4.9757	0.18	0.6208
		Average		1018	4.9757	0.2033	0.6208

At this low voltage level, the Arduino UNO demonstrates excellent performance with an error of only 0.0080%, the lowest error recorded across all tests. Identical ADC values (256) across the three repetitions indicate perfect consistency. A standard deviation of 0.0533 ADC points to minimal fluctuations that are almost negligible. At the 50% position, and reaching the 75% level, it achieves a standard deviation of 0 ADC, representing perfect consistency. The error increases slightly to 0.8531%, which is the highest error in this test, yet remains below the 1% threshold. At maximum voltage (near the 5V Vref), the Arduino UNO continues to show very good performance. The standard deviation increases to 0.2033 ADC, which is still very low but indicates slight variation in the high range.

TDS Sensor Potentiometer Data on ESP32

Testing was conducted to evaluate the performance of the TDS sensor using the ESP32 microcontroller at four different voltage levels (25%, 50%, 75%, and 100% of the full scale). Each level was tested three times to validate the consistency and repeatability of the system.

Table 2. TDS Sensor Potentiometer Data on ESP32

No.	Position (%)	Ref (V)	Step	ESP32			
				ADC	V	StdDev (ADC)	ERROR (%)
1	25	0.825	1	856	0.6902	1.8	16.3394
		0.825	2	855	0.6895	1.76	16.4242
		0.825	3	855	0.6897	1.96	16.4000
		Average		855.3333	0.6898	1.84	16.3879
2	50	1.65	1	1882	1.5173	1.5	8.0424
		1.65	2	1883	1.5175	1.63	8.0303
		1.65	3	1883	1.5176	1.71	8.0242
		Average		1882.6667	1.5175	1.6133	8.0323
3	75	2.475	1	2909	2.3449	1.72	5.2566
		2.475	2	2909	2.345	1.56	5.2525
		2.475	3	2909	2.3447	1.75	5.2646
		Average		2909.0000	2.3449	1.6767	5.2579
4	100	3.12	1	4082	3.2897	3.03	5.4391
		3.12	2	4082	3.2899	3.06	5.4455
		3.12	3	4082	3.2898	2.75	5.4423
		Average		4082.0000	3.2898	2.9467	5.4423

At the 25% position, the ESP32 exhibits very poor performance with an error reaching 16.39%. A significant underestimation occurs where the voltage, which should be 0.825V, is only read as 0.6898V, representing a loss of approximately 0.135V or 16.4% of the actual value. The error decreases drastically by half (8.03%) compared to the 25% position, indicating a significant improvement in the mid-range. However, the error remains 8% higher than the reference voltage. The error continues to decline to 5.26%, demonstrating a consistent upward trend in accuracy. While the identical ADC values (2909) across the three tests are positive, the standard deviation remains high at 1.68 ADC.

3.2. TDS Sensor Measurement Data

1st Measurement Data

The first tests were conducted to evaluate the capability of the Arduino UNO and ESP32 in measuring the output voltage from the TDS sensor using a water sample with a concentration of 101 PPM. The TDS sensor

voltage is a critical parameter that is subsequently converted into TDS values in PPM. The testing was performed at room temperature (~20-21°C) with three repetitions to validate consistency.

Table 3. Arduino UNO TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
1	101	0.228	20.6	1	44	0.2197	96.8	0.24	3.64035	4.158415
	101	0.228	20.6	2	46	0.2249	98.9	0.16	1.35964	2.079207
	101	0.228	20.6	3	46	0.2249	98.9	0.1	1.35964	2.079207
Average					45.33	0.223	98.2	0.166	2.11988	2.772277

Table 4. ESP32 TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
1	101	0.228	20.6	1	44	0.2197	96.8	0.24	3.64035	4.158415
	101	0.228	20.6	2	46	0.2249	98.9	0.16	1.35964	2.079207
	101	0.228	20.6	3	46	0.2249	98.9	0.1	1.35964	2.079207
Average					45.33	0.223	98.2	0.166	2.11988	2.772277

In the results of the 1st test, the voltage measurement on the TDS sensor from the Arduino UNO system showed a more stable result with an average voltage value of only 0.233 V from the reference value of TDS of 0.228 V. Meanwhile, the ESP32 obtained the average measurement of the voltage value on the TDS sensor of 0.106 V from the reference value of TDS 0.233 V. The Arduino UNO system got an average voltage error value of 2.11% and the ESP32 system got an average voltage error value by 54.32%.

2nd Measurement Data

The 2nd test was carried out to measure the output voltage of the TDS sensor on a water sample with a concentration of 201 PPM. The test was performed at room temperature conditions (~20-21°C) with three repetitions to validate consistency.

Table 5. Arduino UNO TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
2	201	0.444	20.6	1	91	0.445	186.5	0.17	0.22522	7.21393
	201	0.444	20.6	2	91	0.4449	186.4	0.16	0.20270	7.26368
	201	0.444	20.6	3	91	0.4449	186.4	0.18	0.20270	7.26368
Average					91	0.4449	186.4	0.17	0.2102	7.2471

Table 6. ESP32 TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
2	201	0.476	21	1	427	0.3444	146.0	11.06	27.6470	27.3333
	201	0.476	21	2	427	0.3446	146.1	11.98	27.6050	27.2885
	201	0.476	21	3	426	0.3439	145.8	13.45	27.7521	27.4328
Average					426.6	0.3443	146.0	12.163	27.6681	27.3516

In the results of the 2nd test, the voltage measurement on the TDS sensor from the Arduino UNO system showed a more stable result with an average voltage value of only 0.4449 V from the reference value of TDS of 0.444 V. Meanwhile, the ESP32 obtained the average measurement of the voltage value on the TDS sensor of 0.3443 V from the reference value of TDS 0.476 V. The Arduino UNO system got an average voltage error value of 0.21% and the ESP32 system got an average voltage error value by 27.66%.

3rd Measurement Data

The 3rd test was carried out in measuring the output voltage of the TDS sensor on a water sample with a concentration of 300 PPM. The test was performed at room temperature conditions (~20-21°C) with three repetitions to validate consistency.

Table 7. Arduino UNO TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
3	300	0.64	20.6	1	131	0.6448	262.7	0.29	0.75	12.4333
	300	0.64	20.6	2	132	0.6454	263	0.18	0.84375	12.3333
	300	0.64	20.6	3	132	0.6454	263	0.23	0.84375	12.3333
Average					131.66	0.6452	262.9	0.2333	0.8125	12.3667

Table 8. ESP32 TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
3	300	0.635	21.1	1	621	0.5009	206.2	10.34	21.1181	31.25
	300	0.635	21.1	2	622	0.5018	206.5	8.71	20.9763	31.14
	300	0.635	21.1	3	622	0.5018	206.5	8.86	20.9763	31.136
Average					621.66	0.5015	206.4	9.303	21.0236	31.175

In the 3rd test results, the voltage measurement on the TDS sensor from the Arduino UNO system showed a more stable result with an average voltage value of only 0.645 V from the TDS reference value of 0.64 V. Meanwhile, the ESP32 obtained an average voltage measurement value on the TDS sensor of 0.501 V from a TDS reference value of 0.635 V. The Arduino UNO system got an average voltage error value of 0.81% and the ESP32 system got an average voltage error value by 21.02%.

4th Measurement Data

The 4th test was carried out to measure the output voltage of the TDS sensor on a water sample with a concentration of 406 PPM. The test was performed at room temperature conditions (~20-21°C) with three repetitions to validate consistency.

Table 9. Arduino UNO TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
4	406	0.845	20.6	1	175	0.859	346	0.46	1.65680	14.7783
	406	0.845	20.6	2	175	0.8602	346.5	0.12	1.79881	14.6551
	406	0.845	20.6	3	176	0.8602	346.5	0.07	1.79881	14.6551
Average					175.33	0.8598	346.3	0.2166	1.75147	14.6962

Table 10. ESP32 TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
4	406	0.901	21.2	1	947	0.7636	305.82	9.2	15.2497	24.6748
	406	0.901	21.2	2	946	0.7625	305.4	12.69	15.3718	24.7783
	406	0.901	21.2	3	944	0.7613	304.9	7.97	15.5049	24.8940
Average					945.66	0.7624	305.3	9.9533	15.3755	24.7824

In the results of the 4th test, the voltage measurement on the TDS sensor from the Arduino UNO system showed a more stable result with an average voltage value of only 0.859 V from the reference value of TDS of 0.845 V. Meanwhile, the ESP32 obtained an average measurement of the voltage value on the TDS sensor of 0.762 V from the reference value of TDS of 0.901V. In the Arduino UNO system, the average voltage error value was 1.75% and in the ESP32 system, the average voltage error value was 24.78%.

5th Measurement Data

The 5th test was carried out in measuring the output voltage of the TDS sensor on a water sample with a concentration of 515 PPM. The test was performed at room temperature conditions (~20-21°C) with three repetitions to validate consistency.

Table 11. Arduino UNO TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
5	515	1.057	20.6	1	220	1.077	437.7	0.47	1.892147	15.0097
	515	1.057	20.6	2	220	1.0774	437.9	0.48	1.929990	14.9708
	515	1.057	20.6	3	220	1.0782	438.2	0.49	2.005676	14.9126
Average					220	1.0775	437.9	0.48	1.94260	14.9644

Table 12. ESP32 TDS sensor Measurement Data

No.	TDS (PPM)	TDS Ref (V)	Temperature (°C)	Step	Arduino UNO					
					ADC	TDS V	TDS	StdDev (ADC)	ERROR V (%)	ERROR TDS (%)
5	515	1.029	20.9	1	1108	0.8937	356.8	11.1	13.1486	30.7048
	515	1.029	20.9	2	1110	0.8948	357.3	12.21	13.0417	30.6174
	515	1.029	20.9	3	1109	0.8941	357.0	13.06	13.1098	30.6699
Average					1109	0.8942	357.0	12.123	13.1000	30.6641

In the results of the 5th test, the voltage measurement on the TDS sensor from the Arduino UNO system showed a more stable result with an average voltage value of only 1.07 V from the reference value of TDS of 1.057 V. Meanwhile, the ESP32 obtained the average measurement of the voltage value on the TDS sensor of 0.894 V from the reference value of TDS of 1.029 V. The Arduino UNO system obtained an average voltage error value of 1.94% and the ESP32 system obtained an average voltage error value by 13.1%.

A critical finding of this research is the disparity between theoretical and practical ADC performance. The ESP32's 12-bit ADC theoretically provides 4× higher resolution than Arduino's 10-bit ADC (4096 vs 1024 discrete levels). However, the results demonstrate that higher bit depth does not guarantee better measurement accuracy or precision when ADC linearity, noise, and system architecture are suboptimal [21]-[26]. The Arduino UNO, despite its lower 10-bit resolution, achieves superior practical performance through excellent linearity, low noise, and stable reference voltage. This illustrates an important principle in instrumentation: ADC resolution is only valuable when accompanied by proportional improvements in accuracy, linearity, and noise performance [27]-[31].

For Applications Requiring High Accuracy (< 2% error): Arduino UNO or similar platforms with linear, low-noise ADCs should be selected. The theoretical advantage of ESP32's higher resolution and processing power is negated by poor analog performance [32]-[36]. For Low-Voltage Sensor Applications (< 1V signals): Extra caution is required when using ESP32, as this research demonstrates critical failures in this range. External ADCs with better linearity characteristics or signal conditioning circuits to amplify low-voltage signals above 1V should be considered. For Multi-Sensor IoT Applications: If ESP32 is required for WiFi connectivity, implementing external 16-bit or 24-bit ADC modules (e.g., ADS1115, ADS1256) via I2C or SPI interfaces can overcome the internal ADC limitations while preserving ESP32's connectivity advantages [37]-[41].

4. CONCLUSION

Based on the results of the comparative research on the accuracy of TDS sensor readings using Arduino UNO and ESP32 microcontrollers, several important conclusions can be drawn as follows. The Arduino UNO shows a much superior voltage reading accuracy compared to the ESP32. In the potency test, the Arduino UNO produced an average error of only 0.54% with a maximum error of 0.8531% at the 75% position, while the ESP32 produced an average error of 8.78% with the highest error reaching 16.39% at the 25% position (low voltage range). This difference shows that the Arduino UNO has 16.3 times better accuracy than the ESP32 in TDS sensor voltage readings. Arduino UNO demonstrated excellent reading consistency with an average standard deviation of only 0.064 ADC, even achieving perfect values (StdDev = 0) at 50% and 75% potency tests. In contrast, the ESP32 shows significant variability with an average standard deviation of 2,019 ADCs, which is 31.5 times higher than the Arduino UNO. This high Arduino UNO consistency ensures stable and reliable readings for continuous monitoring. Arduino UNO exhibits excellent ADC linearity over the entire measurement range (0-100%), with a smooth output voltage curve proportional to the input. The ESP32

experienced serious non-linearity problems, especially in the low voltage range (<10% of the V_{ref}), where there was an underestimation of up to 54.32% in the measurement of the real TDS sensor with a concentration of 101 PPM. This problem resulted in the ESP32 only being able to read 45.68% of the actual sensor voltage. Tests with real TDS sensors showed very significant differences in the low voltage range. In a 101 PPM water sample with a sensor output of about 0.23V (7% of the V_{ref} ESP32), the Arduino UNO produced a voltage error of only 2.12% and a TDS error of 2.77%, while the ESP32 failed critical with a voltage error of 54.32% and a TDS error of 52.49%. ESP32 misreads 101 PPM as 48 PPM, an error that can lead to a misclassification of water quality categories. At five different TDS concentration levels (101, 201, 300, 406, and 515 PPM), the Arduino UNO consistently exhibits low voltage errors ranging from 0.21% to 2.12%, with average TDS errors below 15%. The ESP32 exhibits voltage errors that vary from 13.1% to 54.32%, with the worst performance at low concentrations and gradual improvement at high concentrations. This pattern confirms that the ESP32 problem is systematic and related to the characteristics of ADCs in the low voltage range.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Allah SWT for the grace, guidance, and blessing of health that enabled the completion of this research. It is hoped that this work will contribute meaningfully to the advancement of knowledge. The authors would also like to extend their sincere appreciation to the family and relatives for their unwavering motivation, prayers, and moral support throughout the course of this research. Special thanks are due to Dr. Agus Ulinuha, S.T., M.T., the supervisor, for his invaluable guidance, mentorship, and assistance in the completion of this research. The authors are also grateful to Rijal, Ilham, Ridho, Eko, Roy, Rizki, and Tegar for their contributions of knowledge and support during the research process.

AUTHOR CONTRIBUTIONS

For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, A.R., and A.U.; Methodology, A.R.; Software, A.R.; Validation, A.R., and A.U.,; Formal Analysis, A.R.; Investigation, A.R.; Resources, A.R.; Data Curation, A.R.; Writing – Original Draft Preparation, A.R.; Writing – Review & Editing, A.R.; Visualization, A.R.; Supervision, A.R.; Project Administration, A.R.; Funding Acquisition, A.U.”.

CONFLICTS OF INTEREST

The author(s) declare no conflict of interest.

USE OF ARTIFICIAL INTELLIGENCE (AI)-ASSISTED TECHNOLOGY

The authors declare that no artificial intelligence (AI) tools were used in the generation, analysis, or writing of this manuscript. All aspects of the research, including data collection, interpretation, and manuscript preparation, were carried out entirely by the authors without the assistance of AI-based technologies.

REFERENCES

- [1] S. R. Laha, B. K. Pattanayak, and S. Pattnaik, “Advancement of environmental monitoring system using IoT and sensor: A comprehensive analysis. *AIMS Environmental Science*, vol. 9, no. 6, pp. 771-800, 2022. doi: 10.3934/environsci.2022044.
- [2] O. V. Medagedara, and M. H. Liyanage, “Development of an IoT-based real-time temperature and humidity monitoring system for factory electrical panel rooms,” *Engineering Journal of Institute of Engineers Sri Lanka*, vol. 57, no. 1, pp. 21-30, 2024, doi: 10.4038/engineer.v57i1.7636.
- [3] T. Tanti, W. Utami, D. Deliza, and M. Jahanifar, “Investigation in vocation high school for attitude and motivation students in learning physics subject”, *Journal Evaluation in Education (JEE)*, vol. 6, no. 2, pp. 479-490, 2025, doi: 10.37251/jee.v6i2.1452.
- [4] H. M. Forhad, M. R. Uddin, R. S. Chakrovorty, A. M. Ruhul, H. M. Faruk, S. Kamruzzaman, and A. M. Morshed, “IoT based real-time water quality monitoring system in water treatment plants (WTPs),” *Heliyon*, vol. 10, no. 23, 2024, doi: 10.1016/j.heliyon.2024.e40746.
- [5] M. Babiuch, P. Folytynek, and P. Smutny, “Using the ESP32 microcontroller for data processing,” *MATEC Web of Conferences*, vol. 292, p. 01011, 2019, doi: 10.1051/mateconf/201929201011.
- [6] M. Truntic and O. Tezak, “Design and implementation of ESP32-Based IoT devices,” *Sensors*, vol. 23, no. 15, p. 6739, 2023, doi: 10.3390/s23156739.
- [7] I. Rinaldi, A. Depari, P. Ferrari, A. Flammini, A. Musatti, and E. Sisinni, “Characterization and performance evaluation of ESP32 for real-time synchronized sensor networks,” *Procedia Computer Science*, vol. 238, pp. 104–113, 2024, doi: 10.1016/j.procs.2024.05.104.
- [8] M. Tansa, “Monitoring kualitas air sungai (kekeruhan, suhu, tds, ph) menggunakan mikrokontroler ATmega328 [Monitoring river water quality (turbidity, temperature, TDS, pH) using an ATmega328 microcontroller],” *Jambura Journal of Electrical and Electronics Engineering*, vol. 6, no. 1, doi: 10.37905/jjeec.v6i1.23315.

- [9] Y. Irawan, A. Febriani, R. Wahyuni, and Y. Devis, "Monitoring system of water quality on hydroponic planting media using total dissolved solid (TDS) sensor based arduino uno R3," *Jurnal Ilmu Fisika*, vol. 14, no. 2, pp. 108–115, 2022, doi: 10.25077/jif.14.2.108-115.2022.
- [10] N. D. Susanti, D. Sagita, I. F. Apriyanto, C. E. W. Anggara, D. A. Darmajana, and A. Rahayuningtyas, "Design and implementation of water quality monitoring system (Temperature, pH, TDS) in aquaculture using IoT at low cost," in *Proc. 6th Int. Conf. Food, Agriculture, and Natural Resource (IC-FANRES 2021)*, Atlantis Press, 2022, pp. 7–11, doi: 10.2991/absr.k.220101.002.
- [11] Y. Adityas et al., "Water quality monitoring system with parameter of pH, temperature, turbidity, and salinity based on internet of things," *Jurnal Informatika dan Sains (JISA)*, vol. 4, no. 2, pp. 138–143, 2021, doi: 10.31326/jisa.v4i2.965.
- [12] M. N. Nizam, H. Yuana, and Z. Wulansari, "Mikrokontroler ESP32 sebagai alat monitoring pintu berbasis web [ESP32 microcontroller as a web-based door monitoring tool]," *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 767-772, 2022, doi: 10.36040/jati.v6i2.5713.
- [13] S. Baco, R. Syarifuddin, S. Sajiah, M. A. Rusli, and M. D. Loilatu, "Rancang bangun alat monitoring ph air layak konsumsi berbasis mikrokontroler arduino uno pada depot air minum adipura 1 [Design and construction of a pH monitoring device for potable water based on an Arduino Uno microcontroller at the Adipura 1 drinking water depot]," *Jurnal Teknologi dan Komputer (JTEK)*, vol. 3, no. 2, pp. 339-345, 2023, doi: 10.56923/jtek.v3i02.147.
- [14] W. J. Hong, N. Shamsuddin, E. Abas, R. A. Apong, Z. Masri, H. Suhaimi, S. H. Godeke, and M. N. A. Noh, "Water quality monitoring with arduino based sensors," *Environments*, vol. 8, no. 1, pp. 6, 2021, doi: 10.3390/environments8010006.
- [15] J. Junaidi, T. M. Putra, A. Surtono, G. A. Puazi, S. W. Suciayati, I. Firdaus, and L. Rumiyantri, "Measurement of Physical Parameters of Water Quality in Real-Time Based on Arduino," *Journal of Physics: Conference Series*, vol. 1751, no. 1, pp. 012067, 2021, doi: 10.1088/1742-6596/1751/1/012067.
- [16] A. A. Sneineh, and A. A. Shabaneh, "Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things," *MethodsX*, vol. 11, pp. 102401, 2023, doi: 10.1016/j.mex.2023.102418.
- [17] H. Fan, Z. Chen, T. Xu, F. Maloberti, Q. Wei, and Q. Feng, "14-Bit SAR ADC with on-chip digital bubble sorting calibration technology," *Chinese Journal of Electronics*, vol. 34, no. 1, pp. 125–136, 2025, doi: 10.23919/cje.2023.00.307.
- [18] M. Grossi, "Efficient and accurate analog voltage measurement using a direct sensor-to-digital port interface for microcontrollers and field-programmable gate arrays," *Sensors*, vol. 24, no. 3, pp. 873, 2024, doi: 10.3390/s24031062.
- [19] W. Jiang, W. Zhao, T. Zhou, L. Wang, and T. Qiu, "A review on manufacturing and post-processing technology of vascular stents," *Micromachines*, vol. 13, no. 1, pp. 140, 2022, doi: 10.3390/mi13010140.
- [20] A. Saric, K. Dokic, and M. Cobovic, "Performance evaluation of C/C++, MicroPython, Rust and TinyGo programming languages on ESP32 microcontroller," *Electronics*, vol. 12, no. 1, pp. 143, 2022, doi: 10.3390/electronics12010143.
- [21] N. A. Cloete, R. Malekian, and L. Nair, "Design of smart sensors for real-time water quality monitoring," *IEEE Access*, vol. 4, pp. 3975–3990, 2016, doi: 10.1109/ACCESS.2016.2592958.
- [22] L. Rocher, E. Viala, and C. Roux, "Low-Cost Water Quality Sensors for IoT: A Systematic Review," *Sensors*, vol. 23, no. 10, pp. 4853, May 2023, doi: 10.3390/s23104853.
- [23] I. Essamlali, H. Nhaila, and M. El Khaili, "Advances in machine learning and IoT for water quality monitoring: A comprehensive review," *Heliyon*, vol. 10, no. 6, p. e27920, Mar. 2024, doi: 10.1016/j.heliyon.2024.e27920.
- [24] F. Almajed, F. Almajed, and Q. A. Al-Haija, "An IoT-enabled intelligent water quality monitoring system for tourist safety using machine learning," *Discover Electronics*, vol. 2, no. 1, 2025, doi: 10.1007/s44291-025-00070-1.
- [25] A. H. El-Khozondar, E. H. Bayoumi, and A. A. E. A. Halim, "A smart energy monitoring system using ESP32 microcontroller," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 9, p. 100666, 2024, doi: 10.1016/j.prime.2024.100666.
- [26] V. K. Tran, B. T. Thai, H. Pham, V. K. Nguyen, and V. K. Nguyen, "A proposed approach to utilizing esp32 microcontroller for data acquisition," *Journal of Engineering and Technological Sciences*, vol. 56, no. 4, pp. 474-488, 2024, doi: 10.5614/j.eng.technol.sci.2024.56.4.4.
- [27] D. Chen, X. Cui, Q. Zhang, D. Li, W. Cheng, C. Fei, and Y. Yang, "A survey on analog-to-digital converter integrated circuits for miniaturized high resolution ultrasonic imaging system," *Micromachines*, vol. 13, no. 1, pp. 114, 2022, doi: 10.3390/mi13010114.
- [28] M. A. Mohammedali, and Q. Al-Gayem, "An accurate and fast method for improving adc nonlinearity," *Applied Computational Intelligence and Soft Computing*, vol. 2023, no. 1, pp. 8899666, 2023, doi: 10.1155/2023/8899666.
- [29] M. Zhao, Y. Yuan, L. Luo, and X. Li, "A review: Absolute linear encoder measurement technology," *Sensors*, vol. 25, no. 19, pp. 5997, 2025, doi: 10.3390/s25195997.
- [30] P. K. Ramanathan, S. Ramasamy, M. Ericsson, and S. PV, "Capacitive sensor interfaces: Signal conditioning techniques a review," *Journal of Sensors*, vol. 1015, no. 1, pp. 9146911, 2025, doi: 10.1155/js/9146911.
- [31] B. S. Rawat, N. Kumar, D. Ghosal, D. Aflyatunova, B. Rienäcker, and C. P. Welsch, "Faraday cups: Principles, designs, and applications across scientific disciplines a review," *Instruments*, vol. 9, no. 3, pp. 20, 2025, doi: 10.3390/instruments9030020.
- [32] P. Yan, "Architectural evolution and performance optimization in embedded systems: A comparative analysis of ESP8266, ESP32-S3, and ESP32-C6 platforms," *International Journal of Advance in Applied Science Research*, 4(9), 51-56, 2025.
- [33] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and implementation of ESP32-based IoT devices," *Sensors*, vol. 23, no. 15, pp. 6739, 2023, doi: 10.3390/s23156739.
- [34] N. P. Le, N. B. Khanh, L. N. Giap, T. N. T. An, and B. T. Trung, "A comparative analysis of time synchronization techniques for a hybrid multi-rate power monitoring system," *Engineering, Technology & Applied Science Research*, vol. 15, no. 6, pp. 29992-29998, 2025, doi: 10.48084/etasr.14702.

- [35] R. Raphael, R. Sarukkalige, S. Narasimhan, and H. Agrawal, "Performance evaluation of chaosfortress lightweight cryptographic algorithm for data security in water and other utility management," *Sensors*, vol. 25, no. 16, pp. 5103, 2025, doi: 10.3390/s25165103.
- [36] G. M. Puentes-Conde, E. Sifuentes, J. Molina, F. Enríquez-Aguilera, G. Bravo, and G. N. Enríquez, "Direct interface circuits for resistive, capacitive, and inductive sensors: A review," *Electronics*, vol. 14, no. 12, pp. 2393, 2025, doi: 10.3390/electronics14122393.
- [37] M. S. Bakare, and K. Abubaker, "IoT-based indoor environmental monitoring system using multi-parameter sensing and ESP32-WROOM integration," *Discover Electronics*, vol. 3, no. 1, pp. 6, 2026, doi: 10.1007/s44291-026-00157-3.
- [38] F. D. Makatita, and N. F. A. Hakim, "MQTT protocol-based ESP-32 smarthome with multi-sensor recognition," *Journal of Electrical, Electronic, Information, and Communication Technology*, vol. 6, no. 1, pp. 29-36, 2024, doi: 10.1007/s44291-026-00157-3.
- [39] M. Sousa, A. Alves, R. Antunes, M. Aguiar, P. D. Gaspar, and N. Pereira, "Interoperable IoT/WSN sensing station with edge ai-enabled multi-sensor integration for precision agriculture," *Agriculture*, vol. 16, no. 1, pp. 69, 2025, doi: 10.3390/agriculture16010069.
- [40] F. Serepas, I. Papias, K. Christakis, N. Dimitropoulos, and V. Marinakis, "Lightweight embedded IoT gateway for smart homes based on an ESP32 microcontroller," *Computers*, vol. 14, no. 9, pp. 391, 2025, doi: 10.3390/computers14090391.
- [41] V. Sámano-Ortega, O. Arzate-Rivas, J. Martínez-Nolasco, J. Aguilera-Álvarez, C. Martínez-Nolasco, and M. Santoyo-Mora, "Multipurpose modular wireless sensor for remote monitoring and IoT applications," *Sensors*, vol. 24, no. 4, pp. 1277, 2024, doi: 10.3390/s24041277.