Design and Performance Evaluation of a Low-Cost Edge-To-Cloud Telemetry System for Industrial Monitoring

Emina E. Etomi[1], Benjamin O. Akinloye[1], Donatus U. Onyishi[1]

Federal University of Petroleum Resources (FUPRE), P.M.B. 1221, Effurun, Nigeria

DOI: https://doi.org/10.51584/IJRIAS.2026.11080034

Received: 20 August 2026; Accepted: 25 August 2026; Published: 01 September 2026

ABSTRACT

Industrial facilities increasingly rely on continuous telemetry acquisition to support condition monitoring, operational awareness, and predictive maintenance. However, many existing edge-to-cloud monitoring systems employ complex communication architectures involving message brokers and middleware, which can increase deployment complexity and computational overhead for low-cost embedded platforms. This study presents the design, implementation, and experimental evaluation of a lightweight edge telemetry system that directly transmits industrial sensor measurements from an ESP32 DevKit V1 to the InfluxDB Cloud time-series database using HTTP. The developed platform integrates an MLX90614 infrared temperature sensor for real-time temperature monitoring, while vibration and electrical measurements were incorporated to validate the complete multi-parameter telemetry pipeline. Telemetry data were transmitted over Wi-Fi, stored in InfluxDB Cloud, and visualized through Grafana for continuous monitoring. Experimental validation was conducted for approximately eleven hours using a one-second sampling interval under representative industrial operating conditions. Performance evaluation demonstrated a mean sensor acquisition time of 1.62 ms, stable free heap memory averaging 230,129 bytes, a mean HTTP communication latency of 2.63 s, and a mean Wi-Fi received signal strength of −22.20 dBm. Successful HTTP response code 204 and continuous telemetry storage confirmed successful edge-to-cloud communication throughout the monitoring period. Analysis of the exported telemetry dataset further demonstrated stable sensor acquisition, firmware execution, wireless connectivity, and cloud data storage. Compared with conventional broker-based architectures, the developed system provides a simpler direct edge-to-cloud telemetry architecture while supporting continuous communication and scalable time-series data storage. The findings demonstrate the capability of low-cost embedded hardware to support continuous industrial telemetry and provide a practical foundation for scalable edge-to-cloud industrial monitoring applications.

Keywords— Edge computing, Industrial Internet of Things (IIoT), ESP32, industrial telemetry, InfluxDB, Grafana, time-series database, condition monitoring.

INTRODUCTION

The Industrial Internet of Things (IIoT) has transformed industrial monitoring by integrating intelligent sensors, embedded computing, wireless communication, and cloud services for continuous acquisition and analysis of operational data. Industrial equipment such as motors, pumps, compressors, and turbines generate measurements including temperature, vibration, current, pressure, speed, and flow rate that support condition monitoring, fault diagnosis, predictive maintenance, and operational decision-making [1], [2], [3], [4].

Advances in embedded systems and wireless communication have further encouraged edge computing for industrial telemetry. By performing data acquisition and preliminary processing close to monitored equipment before transmitting data to cloud platforms, edge computing can reduce communication overhead and provide timely access to operational information [5], [6], [7]. The ESP32 is widely used for such applications because of its integrated Wi-Fi, multiple communication interfaces, low power consumption, computational capability, and support for multiple sensors [6], [7], [8], [9], [10].

However, practical industrial telemetry requires more than sensor interfacing and cloud connectivity. Edge devices must continuously acquire and process multi-sensor measurements, format telemetry data, manage limited resources, and maintain reliable wireless transmission. Increased sensor count and acquisition frequency can introduce communication latency, memory constraints, packet loss, and reduced acquisition throughput, making systematic performance evaluation necessary [9], [10], [11], [12].

Although several ESP32-based monitoring systems have been reported, many studies have focused primarily on sensor integration, cloud connectivity, and visualization rather than quantitative evaluation of the telemetry platform. Metrics such as acquisition time, communication latency, throughput, memory utilization, transmission reliability, wireless signal quality, and long-duration operational stability are often briefly reported or omitted [7], [10], [11], [12], [13]. In addition, many edge monitoring architectures employ middleware such as MQTT brokers, Node-RED servers, or protocol gateways, which can increase system complexity and deployment requirements [14]. Comparatively fewer studies have evaluated direct HTTP-based communication with cloud time-series databases under continuous operating conditions [5], [6], [7], [12], [13].

To address these limitations, this study presents the design and performance evaluation of a low-cost edge-to-cloud telemetry system for industrial monitoring. The developed platform integrates an ESP32 DevKit V1 with temperature, vibration, and electrical sensing, direct HTTP communication, InfluxDB Cloud for time-series data storage, and Grafana for real-time visualization. An industrial electric motor was employed as the monitored asset to validate the telemetry architecture under representative operating conditions. Unlike studies that primarily demonstrate monitoring functionality, this work experimentally evaluates the telemetry platform using sensor acquisition time, HTTP communication latency, firmware memory stability, wireless communication quality, HTTP response status, and telemetry packet characteristics during approximately eleven hours of continuous operation. The findings provide quantitative evidence of the operational performance of a lightweight direct edge-to-cloud telemetry architecture for industrial monitoring applications.

LITERATURE REVIEW

A. Industrial Edge Telemetry Systems

Industrial edge telemetry systems support continuous acquisition, processing, transmission, and visualization of operational data from distributed industrial assets [15], [16]. Unlike fully centralized monitoring architectures, edge telemetry places sensing and preliminary processing close to physical equipment, improving responsiveness and reducing communication overhead while supporting condition monitoring, predictive maintenance, and operational decision-making [15], [17], [18].

A typical architecture integrates industrial sensors, edge acquisition devices, communication networks, data storage, and visualization platforms. Sensors acquire variables such as temperature, vibration, current, voltage, pressure, and flow rate, while edge devices perform local acquisition and processing before transmitting measurements to local or cloud platforms [17], [18], [19], [20]. Edge processing may also include filtering, aggregation, feature extraction, and protocol conversion, reducing unnecessary network traffic and improving communication efficiency [20], [21], [22].

Industrial telemetry has increasingly adopted multi-sensor and wireless architectures. Heterogeneous sensors provide complementary information about equipment condition but increase requirements for synchronization, communication scheduling, and data management [19], [23], [24]. Wireless technologies including Wi-Fi, ZigBee, LoRaWAN, and NB-IoT provide deployment flexibility but may be affected by interference, congestion, latency, and communication reliability in industrial environments [16], [22].

Despite these advances, many studies remain application-oriented and provide limited quantitative evaluation of the telemetry platform itself [17], [18], [21]. System-level characteristics such as acquisition throughput, communication latency, memory utilization, transmission reliability, data integrity, and long-term stability are often reported only briefly [15], [17], [21], [22]. This limits quantitative assessment of low-cost telemetry platforms intended for continuous industrial operation.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

B. Low-Cost Embedded Platforms for Industrial Monitoring

Low-cost embedded platforms provide sensing, local processing, and communication close to industrial assets and are particularly useful for retrofit monitoring of equipment without integrated digital interfaces [25], [26], [27]. Common platforms include Arduino, Raspberry Pi, STM32, ESP8266, ESP32, and FPGA-ARM systems. Arduino-based platforms are widely used for basic sensing and control [27], while Raspberry Pi systems provide greater computing capability for edge analytics, gateways, and local data processing [28], [29], [30]. STM32 and FPGA-ARM platforms are suited to applications requiring low-level hardware control or high-speed deterministic processing [6], [9], [31].

The ESP32 provides an attractive balance of processing capability, integrated wireless communication, peripheral interfaces, and affordability. Its Wi-Fi connectivity and GPIO, ADC, SPI, I²C, UART, and PWM interfaces support multi-sensor acquisition without extensive external communication hardware [26], [32]. ESP32-based systems have consequently been applied to industrial monitoring, environmental sensing, automation, and predictive maintenance applications [10], [26], [27], [32]. However, successful sensor acquisition and dashboard visualization alone do not demonstrate suitability for sustained industrial telemetry. An edge controller must simultaneously manage acquisition, data formatting, wireless communication, buffering, and error handling within limited processing and memory resources [6], [26], [32]. Evaluation of acquisition throughput, communication latency, memory utilization, transmission reliability, timing stability, and continuous operation is therefore necessary to establish practical system capability.

C. Multi-Sensor Data Acquisition in Edge-Based Industrial Monitoring

Industrial equipment condition is generally represented by multiple mechanical, electrical, and environmental variables. Multi-sensor monitoring therefore combines measurements such as temperature, vibration, current, voltage, pressure, flow rate, and rotational speed to provide a more comprehensive representation of equipment behaviour [1], [3], [33]. Edge nodes facilitate this process by integrating sensor acquisition, preliminary processing, buffering, and communication close to the monitored equipment [1], [3].

Heterogeneous sensors may be interfaced through ADCs, GPIO, I²C, SPI, or UART depending on their measurement and sampling requirements [34]. Acquisition strategies including sequential polling, timer-driven sampling, interrupts, buffering, filtering, and preprocessing are used to coordinate multiple sensing channels and maintain data consistency [1], [2], [33]. However, increasing the number of sensors and sampling frequency also increases demands on processor scheduling, memory, synchronization, communication bandwidth, and transmission reliability [3], [6], [35]. Although multi-sensor monitoring has been demonstrated across industrial and related IoT applications [3], [26], [33], quantitative assessment of the acquisition subsystem is less frequently reported. Metrics such as sustained acquisition throughput, latency, synchronization, resource utilization, communication reliability, and long-term stability remain important for determining whether low-cost edge platforms can support continuous industrial telemetry [1], [3], [26].

D. Communication Architectures and Performance Evaluation in Edge Telemetry Systems

Reliable communication is essential because acquired measurements must be transmitted consistently to storage and visualization platforms. Industrial telemetry systems employ communication technologies according to requirements for latency, throughput, scalability, reliability, and implementation complexity [3], [25], [36]. MQTT is widely adopted because of its lightweight publish-subscribe architecture and Quality of Service mechanisms [4], [37], while CoAP and LoRaWAN support constrained or long-range applications. HTTP remains suitable for direct communication with web services and cloud databases and can simplify the architecture by eliminating an intermediate message broker [36], [38].

Time-series databases complement these communication architectures by providing efficient storage and querying of continuously generated timestamped measurements. InfluxDB supports time-series data ingestion and integration with visualization platforms such as Grafana for real-time and historical monitoring [17], [29], [39]. Communication performance nevertheless depends on network conditions, packet size, protocol

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

implementation, edge processing overhead, and server-side ingestion [1], [3], [39]. Performance evaluation should therefore consider the complete telemetry pipeline rather than communication protocols in isolation. Relevant metrics include latency, acquisition throughput, processor and memory utilization, packet delivery reliability, wireless communication quality, and data integrity [26], [39], [40]. Although existing studies have examined individual sensing, communication, and cloud components, comparatively fewer have evaluated these characteristics simultaneously across a complete low-cost edge-to-cloud telemetry system operating continuously. These gaps motivated the present study, which evaluates a direct edge-to-cloud telemetry architecture from multi-sensor acquisition through wireless communication, time-series storage, and visualization using quantitative measures of acquisition time, HTTP communication latency, memory stability, wireless communication quality, HTTP response status, and telemetry packet characteristics.

METHODOLOGY

System Architecture

The developed edge telemetry system provided continuous acquisition, transmission, storage, and visualization of operational data from an industrial induction motor. Thermal, mechanical, and electrical measurements were acquired by an ESP32-based edge node and transmitted directly to InfluxDB through HTTP POST over Wi-Fi. The stored telemetry data were visualized using Grafana for real-time monitoring and historical analysis. The overall architecture is shown in Fig. 1. The sensing subsystem comprised an MLX90614 infrared temperature sensor, MPU6050 vibration sensor, and INA226 current and power monitor connected to the ESP32 through a shared I²C bus. During each acquisition cycle, sensor measurements were serialized using the InfluxDB Line Protocol and transmitted directly to the database without an intermediate message broker or gateway. The system was deployed on a three-phase squirrel-cage induction motor driving a fire hydrant pump under normal industrial operating conditions. System performance was evaluated in terms of sensor acquisition, HTTP communication, memory stability, wireless communication quality, response status, and telemetry packet characteristics.

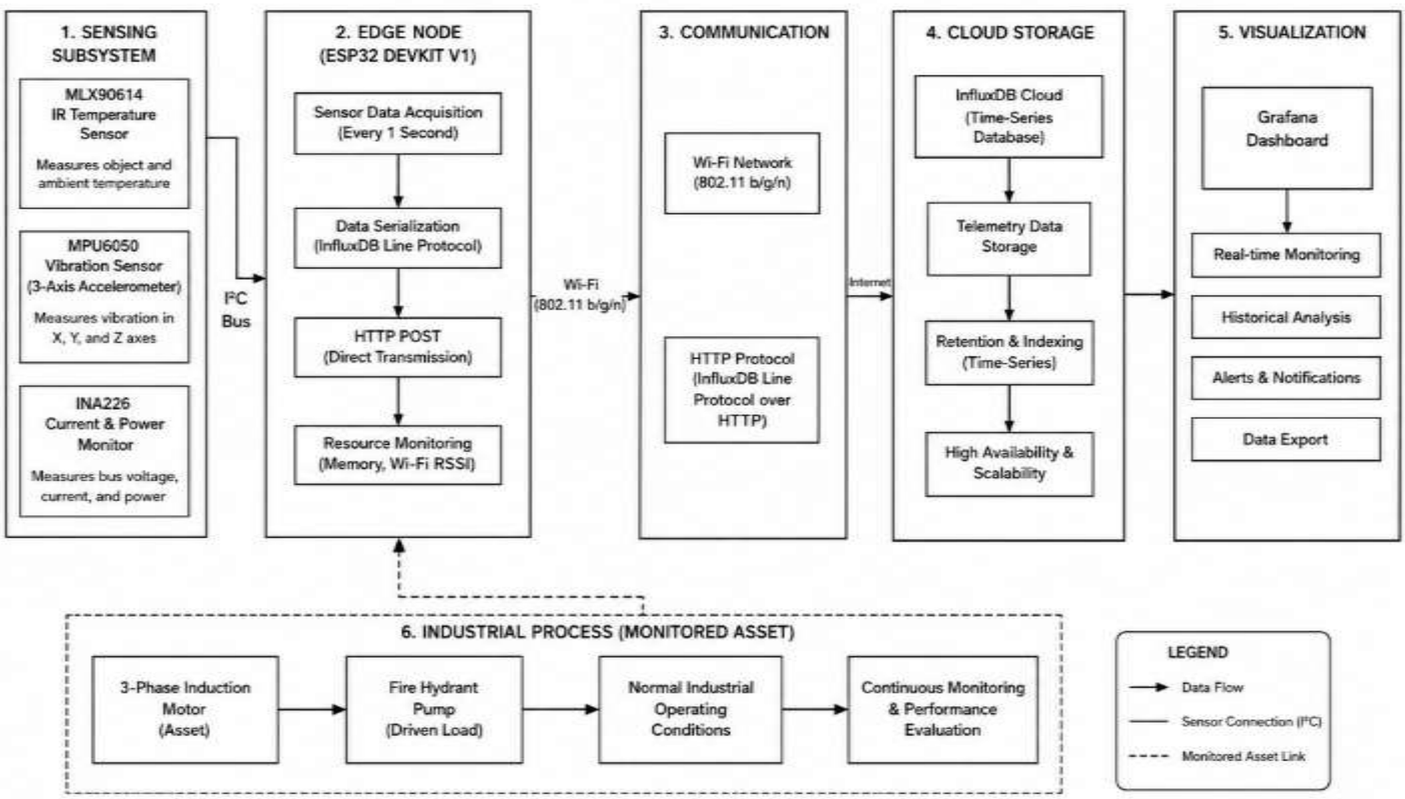

Figure 1: Overall Architecture of the Developed Edge Telemetry System

B. Hardware Design

The hardware subsystem integrated an ESP32 DevKit V1 with the MLX90614, MPU6050, and INA226 sensing modules, as illustrated in Fig. 2. The ESP32 served as the edge controller, providing the processing, sensor interfacing, and wireless communication required for telemetry acquisition and transmission.

The MLX90614 provided non-contact motor surface temperature measurements, while the MPU6050 acquired three-axis acceleration for vibration monitoring. The INA226 measured bus voltage, load current, and electrical power. The sensors communicated through a shared I²C bus using unique device addresses, reducing wiring complexity and ESP32 pin requirements while supporting multi-sensor acquisition. Fig. 2 shows the block diagram of the developed edge telemetry system.

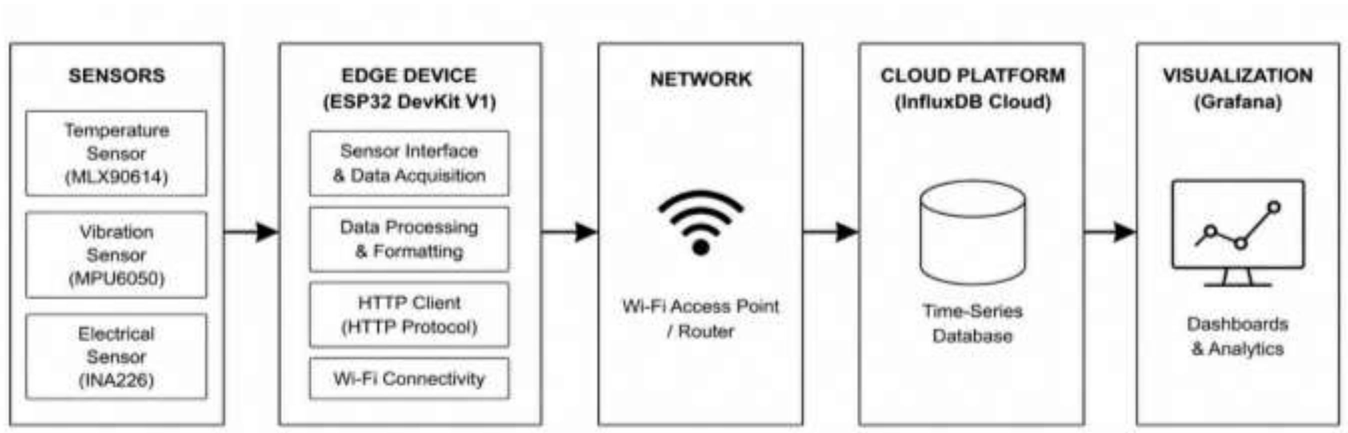

Figure 2: Block Diagram of the Developed Edge Telemetry System

C. Software Architecture

The firmware was developed using Arduino IDE 2.3.8 to manage sensor initialization, data acquisition, validation, serialization, and communication. At startup, the ESP32 initialized the I²C interface and connected sensors before establishing Wi-Fi connectivity.

During operation, temperature, vibration, voltage, current, and power measurements were periodically acquired and validated. The measurements were serialized using the InfluxDB Line Protocol and transmitted directly to InfluxDB Cloud using HTTP POST. The HTTP response was checked to confirm successful storage, while communication failure initiated reconnection before retransmission. The software workflow is illustrated in Fig. 3, and the principal software components are summarized in Table 1.

Table 1: Principal Software Components Used in the Developed Edge Telemetry System

Software Component	Library / Tool	Version	Function
Development Environment	Arduino IDE	2.3.8	Firmware development, compilation, and deployment
ESP32 Development Framework	ESP32 Arduino Core (Espressif)		ESP32 hardware abstraction and peripheral support
I²C Communication	Wire Library	Built-in	Communication with MLX90614, MPU6050, and INA226 sensors
Temperature Sensor Driver	Adafruit MLX90614 Library	Latest compatible	Acquisition of motor surface temperature
Vibration Sensor Driver	MPU6050 Library	Latest compatible	Acquisition of three-axis acceleration measurements
Current and Power Sensor Driver	INA226 Library	Latest compatible	Acquisition of voltage, current, and power measurements

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

Wi-Fi Communication	WiFi Library	Built-in	Wireless network connection management
HTTP Communication	HTTPClient Library	Built-in	HTTP POST transmission to InfluxDB
Time Management	TimeLib	Latest compatible	Timestamp generation for telemetry records

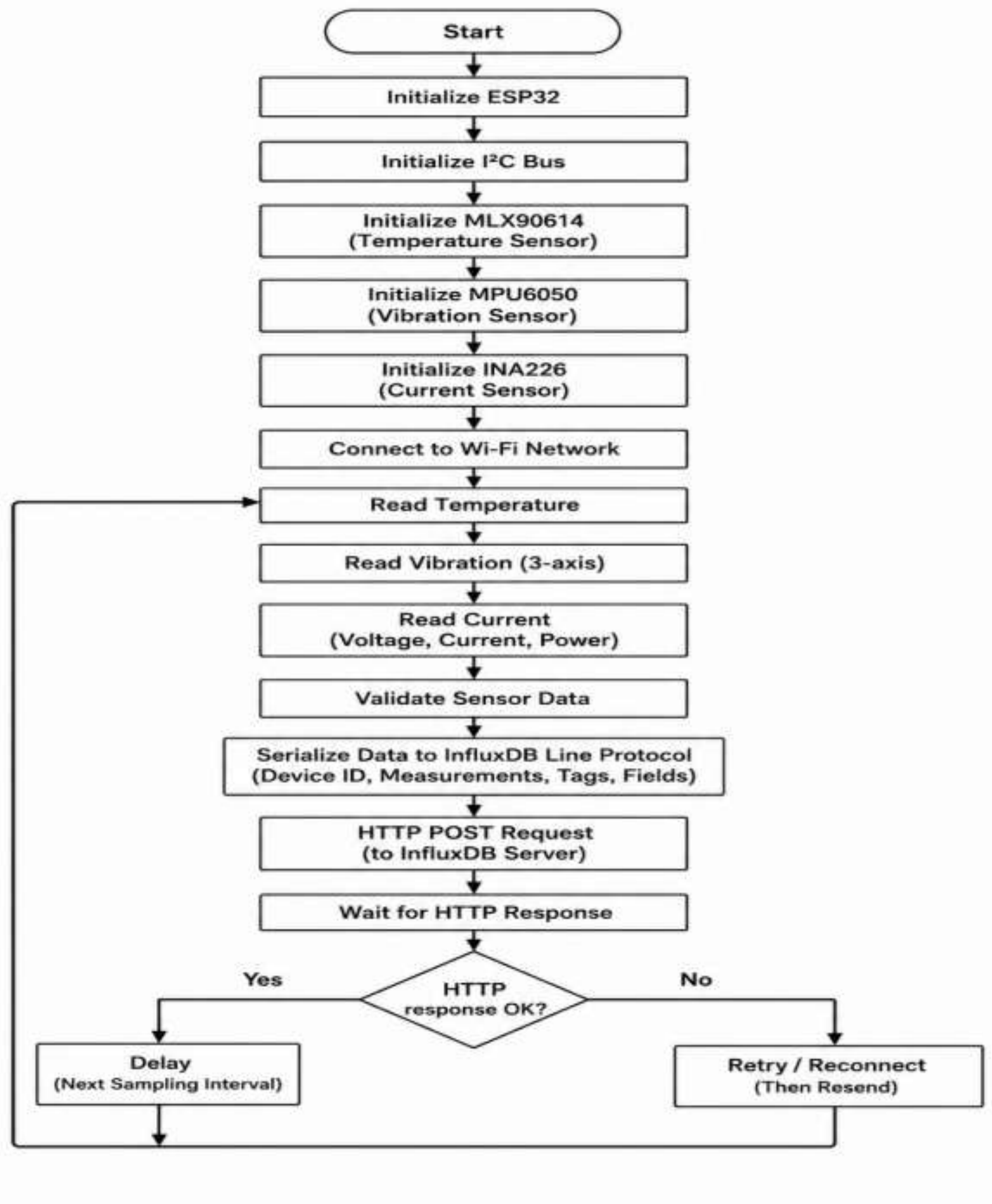

Figure 3: Software Flowchart of the Developed Edge Telemetry System

D. Communication and Security Implementation

Communication between the ESP32 and InfluxDB was implemented using HTTP POST over IEEE 802.11 b/g/n Wi-Fi. The ESP32 connected to the wireless access point using WPA2-PSK authentication, while InfluxDB authenticated incoming requests using an access token. Telemetry measurements were formatted using the InfluxDB Line Protocol and stored within the configured organization and bucket.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

Wi-Fi connectivity and HTTP responses were monitored throughout operation. Successful responses confirmed that telemetry records had been written to the database, while unsuccessful communication initiated reconnection and retransmission. Grafana retrieved the stored telemetry from InfluxDB for real-time and historical visualization. The communication process is illustrated in Fig. 4, while Table 2 summarizes the communication parameters.

Table 2: Communication Parameters of the Developed Edge Telemetry System

Parameter	Configuration
Wireless Standard	IEEE 802.11 b/g/n
Communication Protocol	HTTP POST
Network Security	WPA2-PSK
Data Format	InfluxDB Line Protocol
Database	InfluxDB 2.x
Authentication	Access Token
Data Organization	Organization and Bucket
Visualization Platform	Grafana
Communication Model	Direct Edge-to-Database
Transport Protocol	TCP/IP

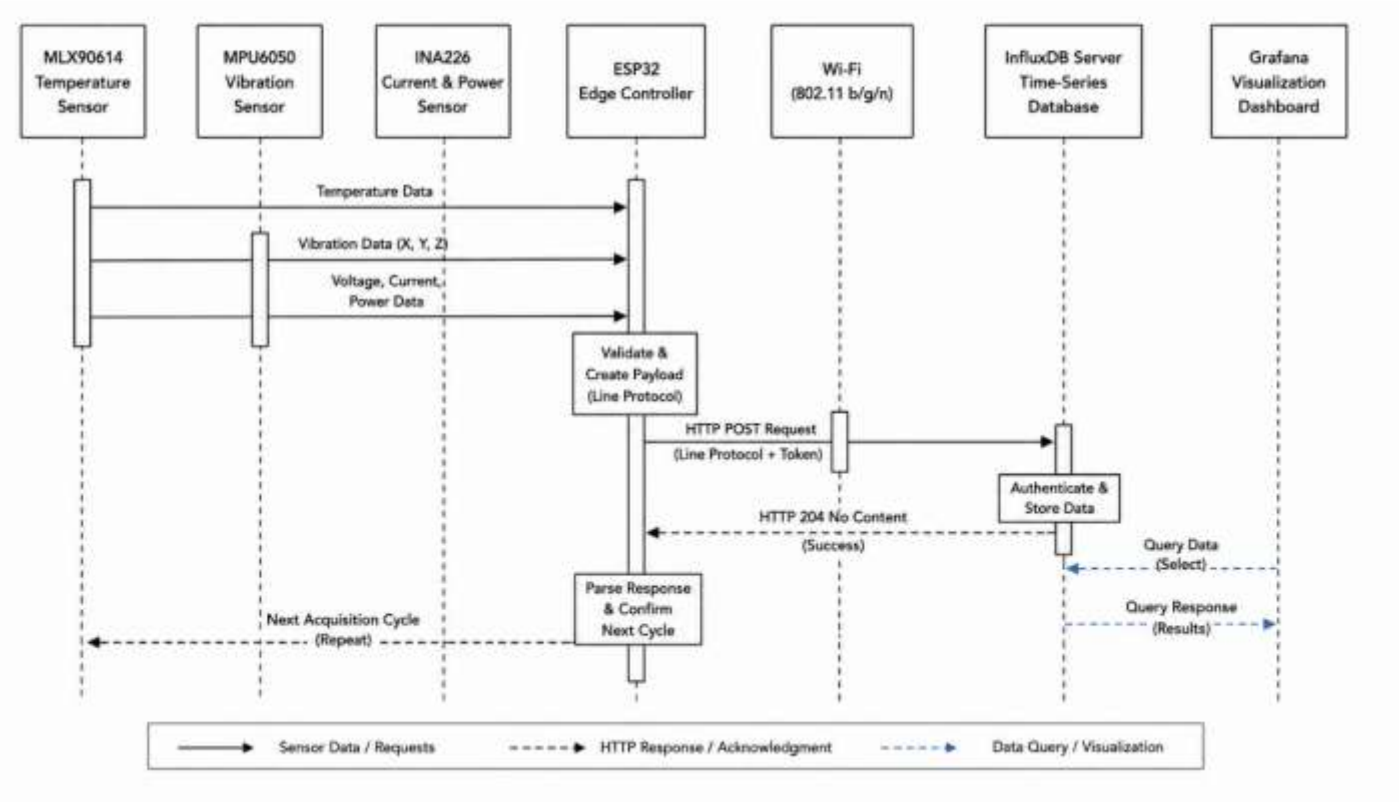

Figure 4: Sequence Diagram of the Developed Edge Telemetry System

E. Experimental Setup

The developed telemetry system was deployed on a three-phase Baldor Reliance squirrel-cage induction motor driving a fire hydrant pump under normal field operating conditions. The ESP32 DevKit V1 served as the edge controller, while the MLX90614, MPU6050, and INA226 provided temperature, vibration, and electrical measurements, respectively.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

Wireless communication was provided through an MTN 4G LTE router operating over IEEE 802.11 b/g/n Wi-Fi. Telemetry records were transmitted directly to InfluxDB Cloud using HTTP POST and visualized through Grafana. The system operated at a 1-s sampling interval for approximately 11 h.

The experimental configuration and wireless network specifications are presented in Tables 3 and 4, respectively, while a section of the Grafana dashboard for the sensors is shown in Fig. 5.

Table 3: Experimental Configuration of the Developed Edge Telemetry System

Parameter	Configuration
Experimental Asset	Three-phase squirrel-cage induction motor driving a fire hydrant pump
Motor Manufacturer	Baldor Reliance Industrial Motor
Motor Model (Cat. No.)	M3538
Motor Specification	34A61-872
Motor Rated Power	0.5 HP (1/2 HP)
Rated Voltage	230/460 V
Rated Current	2/1 A
Rated Speed	1725 RPM
Operating Frequency	60 Hz
Number of Phases	Three-phase
Edge Controller	ESP32 DevKit V1
Temperature Sensor	MLX90614 Infrared Temperature Sensor
Vibration Sensor	MPU6050 MEMS Accelerometer
Electrical Sensor	INA226 Current and Power Monitor
Sensor Communication Interface	Shared I²C Bus
Wireless Communication	IEEE 802.11 b/g/n Wi-Fi
Wireless Router	MTN 4G LTE Router
Communication Protocol	HTTP POST
Time-Series Database	InfluxDB Cloud
Visualization Platform	Grafana
Firmware Development Environment	Arduino IDE Version 2.3.8
Sampling Interval	1 s
Monitoring Duration	11 h
Expected Telemetry Records	39,600 records
Data Storage Method	Time-series database with automatic server-side timestamping

Table 4: Wireless Network Configuration

Parameter	Specification
Router	MTN 4G LTE Router
Cellular Network	4G LTE / 3G
Wi-Fi Standard	IEEE 802.11 b/g/n
Wi-Fi Frequency	2.4 GHz
Maximum Download Rate	Up to 150 Mbps
Maximum Upload Rate	Up to 50 Mbps
Maximum Connected Devices	10
Battery Capacity	6–8 hours (fully charged)

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

Figure 5: Grafana visualization of the Developed Edge Telemetry System

F. Performance Evaluation

The performance evaluation focused on sensor acquisition time, HTTP communication latency, firmware memory stability, wireless communication quality, HTTP response status, and telemetry packet characteristics. Sensor acquisition time and HTTP latency were obtained from firmware timing measurements, while free heap memory and Wi-Fi RSSI were recorded directly from the ESP32 to assess memory stability and wireless communication quality, respectively. HTTP response codes were monitored to verify successful cloud communication, while telemetry packet size was recorded to characterize the transmitted data payload. These parameters were analysed during continuous system operation to evaluate the performance of the developed edge-to-cloud telemetry platform.

RESULTS AND DISCUSSION

A. Experimental Validation of the Developed Edge Telemetry System

The developed edge telemetry system was experimentally validated on a three-phase squirrel-cage induction motor driving a fire hydrant pump under normal industrial operating conditions. Temperature, vibration, and electrical measurements served as telemetry inputs for evaluating the complete acquisition and communication pipeline. The ESP32 acquired the sensor measurements and transmitted the resulting telemetry records directly to InfluxDB Cloud using HTTP POST over Wi-Fi, while Grafana provided real-time visualization. The evaluation focused on sensor acquisition time, HTTP communication latency, firmware memory stability, wireless communication quality, HTTP response status, and telemetry packet characteristics.

Successful HTTP response code 204 confirmed that telemetry records formatted using the InfluxDB Line Protocol were accepted by InfluxDB. The exported dataset was subsequently analysed to evaluate the operational performance of the developed telemetry platform. Table 5 summarizes the principal performance results and monitored variables.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

Table 5: Summary of Telemetry System Performance and Monitored Variables

Performance Metric	Mean	Minimum	Maximum	Standard Deviation	Unit
Acquisition Time	1.62	1.00	2.00	0.49	ms
HTTP POST Latency	2627.31	1639.00	12906.00	1063.98	ms
Free Heap Memory	230129.02	229532.00	231280.00	125.79	bytes
Wi-Fi RSSI	−22.20	−47.00	−11.00	13.90	dBm
Telemetry Packet Size	435.73	429.00	440.00	2.02	bytes

B. Sensor Acquisition and Embedded System Performance

The developed platform maintained continuous multi-sensor acquisition throughout the monitoring period. Temperature, vibration, and electrical measurements were acquired as input variables to evaluate the operation of the telemetry pipeline. The mean sensor acquisition time was 1.62 ms, with recorded values between 1 and 2 ms, indicating that multi-sensor data acquisition required only a small fraction of the configured one-second sampling interval.

Firmware memory also remained stable during continuous operation. Available heap memory averaged 230,129 bytes and ranged from 229,532 to 231,280 bytes. The limited variation in free heap memory indicated stable memory utilization during sensor acquisition, data serialization, HTTP communication, and cloud interaction, with no progressive reduction observed during the monitoring period.

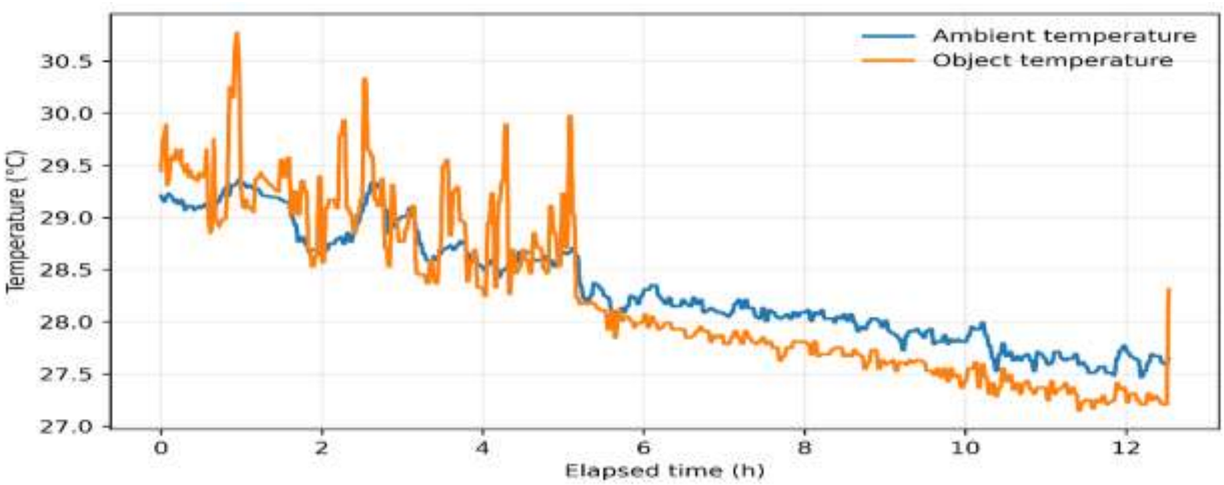

Figure 6: Telemetry Variables Recorded During Continuous Industrial Monitoring

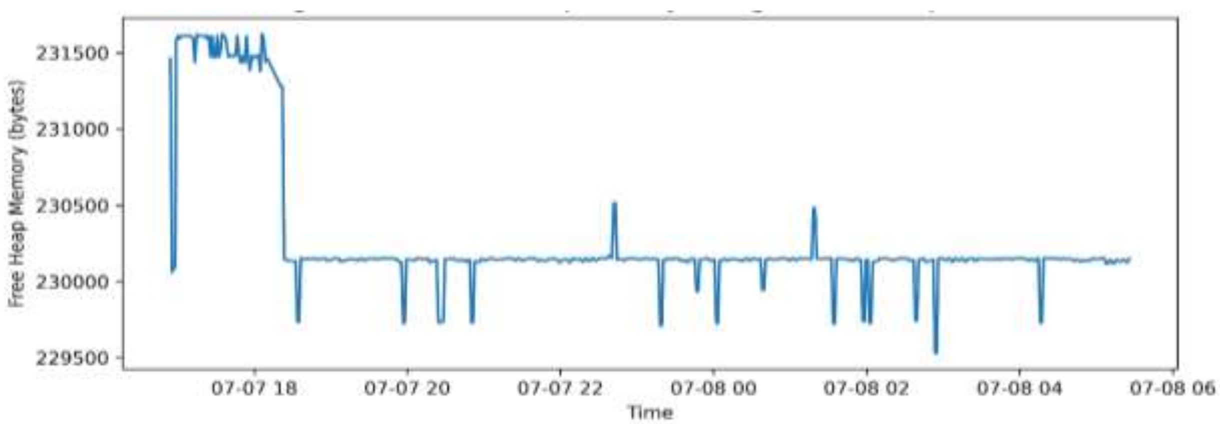

Figure 7: Available Heap Memory During Continuous Operation

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

C. Communication and Wireless Performance

The mean HTTP POST latency was 2627.31 ms, with recorded values ranging from 1639 to 12906 ms. The observed variation reflected the characteristics of Internet-based communication involving Wi-Fi connectivity, cellular backhaul, Internet routing, and cloud-server response time. Despite these variations, successful HTTP 204 responses and continuous storage of telemetry records in InfluxDB confirmed successful edge-to-cloud communication during the monitoring period.

Wireless connectivity remained stable, with a mean RSSI of −22.20 dBm and recorded values between −47 and −11 dBm. These measurements indicated a strong wireless connection between the ESP32 and the MTN 4G LTE router during system operation. Together, the HTTP response status, communication latency, and RSSI measurements demonstrated the communication performance of the direct HTTP-based telemetry architecture.

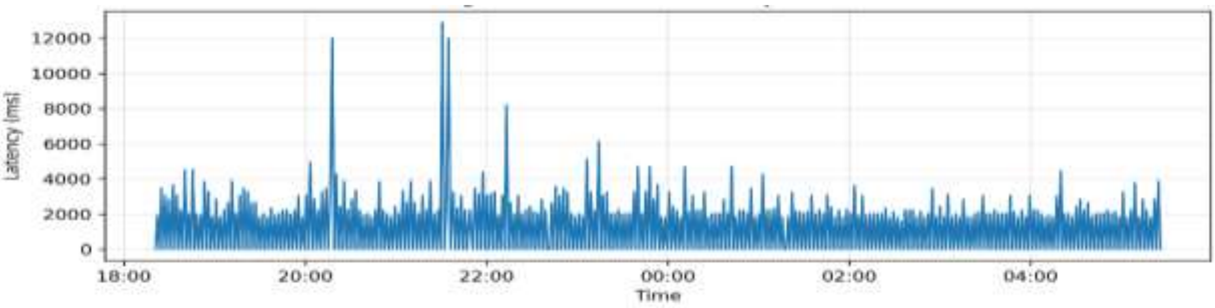

Figure 8: HTTP POST Latency During Continuous Telemetry Transmission

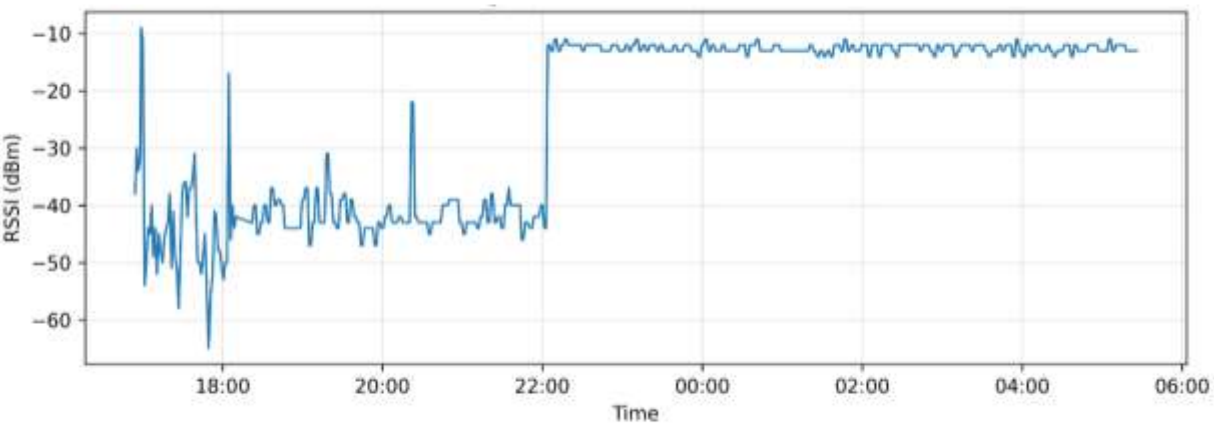

Figure 9: Wi-Fi RSSI During Continuous Operation

D. Telemetry Packet Characteristics and Multi-Sensor Operation

The developed platform successfully incorporated temperature, vibration, and electrical measurements within a common telemetry pipeline. These sensor variables were used to demonstrate multi-parameter acquisition and transmission rather than to evaluate the operating condition of the monitored motor. The continuity of the recorded measurements confirmed that different sensing modalities could be acquired, serialized, transmitted, and stored using the developed platform.

The transmitted telemetry packets had a mean size of 435.73 bytes, ranging from 429 to 440 bytes, with a standard deviation of 2.02 bytes. The limited variation in packet size reflected the consistent structure of the InfluxDB Line Protocol payloads during continuous operation. The results therefore demonstrate stable telemetry formatting and successful integration of multi-sensor acquisition, direct HTTP transmission, cloud storage, and real-time visualization.

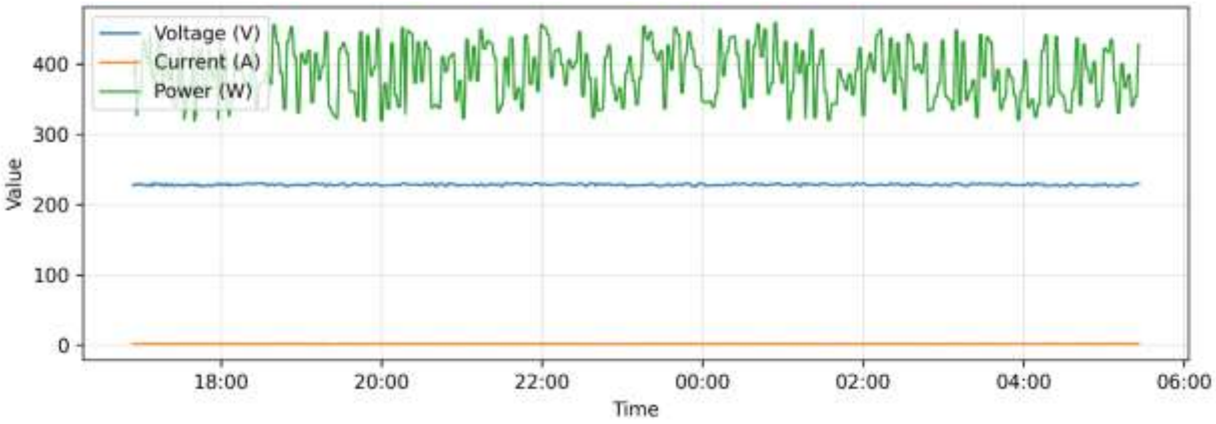

Figure 10: Electrical Telemetry Variables Recorded During Continuous Monitoring

CONCLUSION

This study presented the design, implementation, and performance evaluation of a low-cost edge-to-cloud telemetry system for industrial monitoring. The developed platform integrated an ESP32 DevKit V1 with multiple sensing devices and transmitted telemetry directly to InfluxDB Cloud using HTTP POST over Wi-Fi, while Grafana provided real-time visualization. The system was validated on an industrial induction motor under normal operating conditions for approximately eleven hours at a one-second sampling interval.

Experimental evaluation focused on the performance of the telemetry platform rather than the operating condition of the monitored motor. The system achieved a mean sensor acquisition time of 1.62 ms, while available heap memory remained stable at an average of 230,129 bytes. Mean HTTP POST latency was 2.63 s, and mean Wi-Fi RSSI was −22.20 dBm. Successful HTTP 204 responses confirmed successful cloud communication, while the limited variation in telemetry packet size demonstrated consistent data formatting during operation.

The developed system employed a direct edge-to-cloud architecture without an intermediate MQTT broker, Node-RED server, or protocol gateway. This reduced the number of intermediate communication components while retaining multi-sensor acquisition, wireless transmission, time-series storage, and real-time visualization. Overall, the experimental results demonstrate the capability of the developed platform to support continuous industrial telemetry using low-cost embedded hardware and direct HTTP-based cloud communication.

REFERENCE

1. E. Rosas, B. Arratia, Á. Martín Furones, J. Prades, P. Manzoni, and J. M. Cecilia, “Edge-enabled GNSS-IR for efficient water level monitoring in harsh environments,” Internet of Things (The Netherlands), vol. 34, Nov. 2025, doi: 10.1016/j.iot.2025.101766.
2. X. Chai et al., “Development and Experimentation of a Universal Array-Style Grain Flow Sensor for Multiple Crops Based on Random Forest Flow Regression Model,” Electronics (Switzerland), vol. 13, no. 23, Dec. 2024, doi: 10.3390/electronics13234791.
3. F. A. Delgado-Rajó, C. M. Travieso-González, and R. Hernández-López, “Scalable and low-power edge architecture with Wi-Fi HaLow and on-device spectrograms generation for flexible urban bioacoustics monitoring,” Internet of Things (The Netherlands), vol. 36, Mar. 2026, doi: 10.1016/j.iot.2025.101864.
4. R. Kufakunesu, H. C. Myburgh, and A. De Freitas, “Priority-Based Data Flow Control for Long-Range Wide Area Networks in Internet of Military Things,” Journal of Sensor and Actuator Networks, vol. 14, no. 2, Apr. 2025, doi: 10.3390/jsan14020043.
5. J. Lorincz, M. Kusačić, E. Čusto, and Z. Blažević, “A Comprehensive Multiple Linear Regression Modeling and Analysis of LoRa User Device Energy Consumption,” Journal of Sensor and Actuator Networks, vol. 15, no. 1, p. 5, Dec. 2025, doi: 10.3390/jsan15010005.

6. Y. R. Choi, H. Y. Jeong, and S. Moon, "Hybrid Framework for Secure Low-Power Data Encryption with Adaptive Payload Compression in Resource-Constrained IoT Systems," Sensors, vol. 26, no. 7, Apr. 2026, doi: 10.3390/s26072253.
7. B. Mostafa, H. Haj Ahmad, Y. Rabaiah, and M. Elseddik, "FedCARE: Fuzzy-Supervised Federated Inference with Confidence Gating for Resilient IIoT Sensor Networks," Sensors, vol. 26, no. 12, Jun. 2026, doi: 10.3390/s26123904.
8. Kumar Arigela, C. Banapuram, and N. Venu, "Remote based Home Automation with MQTT: ESP32 Nodes and Node-RED on Raspberry Pi," in 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2024 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 313–318. doi: 10.1109/I-SMAC61858.2024.10714655.
9. D. Zhai, Q. Zhai, S. Liu, X. Liu, and T. Guo, "Research on Fire Source Recognition and Fire Extinguishing Algorithms Based on Multimodal Fusion and Lightweight Model Deployment," Sensors, vol. 26, no. 13, p. 3988, Jun. 2026, doi: 10.3390/s26133988.
10. Z. Liu, K. Perin, and G. Li, "An IoT-Enabled Smart Pillow with Multi-Spectrum Deep Learning Model for Real-Time Snoring Detection and Intervention," Applied Sciences (Switzerland), vol. 15, no. 24, Dec. 2025, doi: 10.3390/app152412891.
11. S. Leone, F. Lago, and G. Lavia, "Design and Experimental Validation of a Unidirectional Cable-Driven Exoskeleton for Upper Limb Rehabilitation," Applied Sciences (Switzerland), vol. 15, no. 22, Nov. 2025, doi: 10.3390/app152211996.
12. C. H. Goh, J. Anthony, and C. Ballard, "Variable-Gravity RoboREHAB for Gait and Sit-to-Stand Rehabilitation: A System-Level Integration Study with Simulation and Benchtop Prototype Evidence," Applied Sciences (Switzerland), vol. 16, no. 9, May 2026, doi: 10.3390/app16094336.
13. O. Paladino, F. Fissore, and M. Neviani, "A low-cost monitoring system and operating database for quality control in small food processing industry," Journal of Sensor and Actuator Networks, vol. 8, no. 4, Oct. 2019, doi: 10.3390/jsan8040052.
14. Hermanudin, F. Ekadiyanto, and R. Sari, "Performance Evaluation of CoAP Broker and Access Gateway Implementation on Wireless Sensor Network," IEEE Region Ten Symposium (Tensymp), Aug. 2019. doi: 10.1109/TENCONSpring.2018.8692050.
15. J. Rosa-Bilbao, "A Low-Code Containerized Edge Architecture for IIoT Telemetry Orchestration: Mitigating Cloud API Rate Limits Through Dual-Path Routing," Sensors, vol. 26, no. 10, May 2026, doi: 10.3390/s26103082.
16. H. Klaina, I. Picallo, P. Lopez-Iturri, and A. Biurrun, "IIoT Low-Cost ZigBee-Based WSN Implementation for Enhanced Production Efficiency in a Solar Protection Curtains Manufacturing Workshop †," Sensors, vol. 24, no. 2, Jan. 2024, doi: 10.3390/s24020712.
17. O. Khattach, O. Moussaoui, and M. Hassine, "End-to-End Architecture for Real-Time IoT Analytics and Predictive Maintenance Using Stream Processing and ML Pipelines," Sensors, vol. 25, no. 9, May 2025, doi: 10.3390/s25092945.
18. R. Mofidul, M. M. Alam, M. H. Rahman, and Y. M. Jang, "Real-Time Energy Data Acquisition, Anomaly Detection, and Monitoring System: Implementation of a Secured, Robust, and Integrated Global IIoT Infrastructure with Edge and Cloud AI," Sensors, vol. 22, no. 22, Nov. 2022, doi: 10.3390/s22228980.
19. D. Bruneo and F. De Vita, "Detecting Faults at the Edge via Sensor Data Fusion Echo State Networks," Sensors, vol. 22, no. 8, Apr. 2022, doi: 10.3390/s22082858.
20. J. Xue, Y. Huang, Y. Guo, Z. Lin, and B. Tian, "Dual-Time-Scale Cloud–Edge–End Collaborative Task Offloading for Multi-AGV Intelligent Warehousing in Industrial Internet of Things," Sensors, vol. 26, no. 12, Jun. 2026, doi: 10.3390/s26123936.D. P. Mtowe, L. Long, and D. M. Kim, "Low-Latency Edge-Enabled Digital Twin System for Multi-Robot Collision Avoidance and Remote Control," Sensors, vol. 25, no. 15, Aug. 2025, doi: 10.3390/s25154666.
21. E. Shahri, P. Pedreiras, and L. Almeida, "A Scalable Real-Time SDN-Based MQTT Framework for Industrial Applications," IEEE Open Journal of the Industrial Electronics Society, vol. 5, pp. 215–235, 2024, doi: 10.1109/OJIES.2024.3373232.

22. M. Hassan, R. Kirwin, C. S. Rakurty, and A. Mahajan, “Advanced Sensor Technologies in Cutting Applications: A Review,” Feb. 01, 2026, Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/s26030762.
23. C. Zu, W. Zhang, Y. Zhou, C. Peng, S. Teng, and F. Zhao, “Feature Extraction and Unsupervised Classification of Roadway Fracture Signals: A Full-Section Wi-Fi Wireless Monitoring Approach,” Sensors, vol. 26, no. 10, May 2026, doi: 10.3390/s26103018.
24. S. K. Bhoi, S. Chakraborty, B. Verbrugge, and S. Helsen, “Intelligent data-driven condition monitoring of power electronics systems using smart edge–cloud framework,” Internet of Things (Netherlands), vol. 26, Jul. 2024, doi: 10.1016/j.iot.2024.101158.
25. S. D. Brito, G. J. Azinheira, J. F. Semião, N. M. Sousa, and S. P. Litrán, “Non-Intrusive Low-Cost IoT-Based Hardware System for Sustainable Predictive Maintenance of Industrial Pump Systems,” Electronics (Switzerland), vol. 14, no. 14, Jul. 2025, doi: 10.3390/electronics14142913.
26. R. Porras, B. Mobaraki, Z. Liu, T. Muñoz, F. Lozano, and J. A. Lozano, “Assessment of Low-Cost Sensors in Early-Age Concrete: Laboratory Testing and Industrial Applications,” Applied Sciences (Switzerland), vol. 15, no. 15, Aug. 2025, doi: 10.3390/app15158701.
27. F. Laganà, “Design and Simulation-Based Validation of an Embedded Acquisition Architecture for In Situ PCB Integrity Monitoring in Biomedical Devices,” Electronics (Switzerland), vol. 15, no. 4, Feb. 2026, doi: 10.3390/electronics15040833.
28. H. Duran and N. K. Sonmez, “Design and Implementation of the AquaMIB Unmanned Surface Vehicle for Real-Time GIS-Based Spatial Interpolation and Autonomous Water Quality Monitoring,” Applied Sciences (Switzerland), vol. 16, no. 3, Feb. 2026, doi: 10.3390/app16031209.
29. B. Kim, S. Wang, and J. Lee, “Video Display Improvement by Using Collaborative Edge Devices with YOLOv11,” Applied Sciences (Switzerland), vol. 15, no. 17, Sep. 2025, doi: 10.3390/app15179241.
30. T. Sledevič and D. Andriukaitis, “Hardware-Assisted Security Enhancements for an FPGA-ARM Embedded Vision System in IoT Applications,” Electronics (Switzerland), vol. 15, no. 9, May 2026, doi: 10.3390/electronics15091887.
31. D. Hercog, T. Lerher, M. Truntič, and O. Težak, “Design and Implementation of ESP32-Based IoT Devices,” Sensors, vol. 23, no. 15, Aug. 2023, doi: 10.3390/s23156739.
32. Z. Cao, Z. Wu, and J. Gray, “RFID Tag-Integrated Multi-Sensors with AIoT Cloud Platform for Food Quality Analysis,” Electronics (Switzerland), vol. 15, no. 1, Jan. 2026, doi: 10.3390/electronics15010106.
33. J. Jittakort and A. Aurasopon, “A Digitally Controlled Adaptive Current Interface for Accurate Measurement of Resistive Sensors in Embedded Sensing Systems,” Journal of Sensor and Actuator Networks, vol. 14, no. 4, Aug. 2025, doi: 10.3390/jsan14040082.
34. M. Ojo, Y. Wang, C. Bua, M. Sher, and A. Zahid, “A crop digital twin system for predictive growth monitoring and adaptive light control in controlled environment agriculture,” Internet of Things (The Netherlands), vol. 37, May 2026, doi: 10.1016/j.iot.2026.101935.
35. C. Tsallis, P. Papageorgas, D. Piromalis, and R. A. Munteanu, “Industrial Wireless Networks in Industry 4.0: A Systematic Review,” Journal of Sensor and Actuator Networks, vol. 15, no. 1, p. 7, Jan. 2026, doi: 10.3390/jsan15010007.
36. F. Kamoun-Abid and A. Meddeb-Makhlouf, “Enhanced MQTT Protocol for Securing Big Data/Hadoop Data Management,” Journal of Sensor and Actuator Networks, vol. 15, no. 1, Feb. 2026, doi: 10.3390/jsan15010022.
37. Ancans, M. Greitans, and S. Kagis, “An Efficient Communication Protocol for Real-Time Body Sensor Data Acquisition and Feedback in Interactive Wearable Systems,” Journal of Sensor and Actuator Networks, vol. 14, no. 1, Feb. 2025, doi: 10.3390/jsan14010004.
38. W. O’Brien, A. Dooley, M. Penica, S. McGrath, and E. O’Connell, “Data-Driven, Real-Time Diagnostics of 5G and Wi-Fi Networks Using Mobile Robotics,” Journal of Sensor and Actuator Networks, vol. 14, no. 6, Dec. 2025, doi: 10.3390/jsan14060110.
39. M. Tang, X. Li, and J. Shang, “Data-Driven Energy-Saving Methods Based on LoRa-Mesh Hierarchical Network,” Sensors, vol. 26, no. 7, Apr. 2026, doi: 10.3390/s26072226.