2026 IEEE International Symposium on Measurement in Networking and Communications (MNC) | 979-8-3315-6214-4/26/$31.00 ©2026 IEEE | DOI: 10.1109/MNC69143.2026.11680787

Integrating IEEE 1451 Smart Transducers into IoT Environments using Node-RED Library

Helbert da Rocha
Dept. of Electromechanical Engineering
University of Beira Interior
IT-Instituto de Telecomunicações
Covilhã, Portugal
helbert.rocha@ubi.pt

Eugene Song
Communications Technology Lab. (CTL)
National Institute of Standards and Technology (NIST)
Gaithersburg, MD, USA
eugene.song@nist.gov

António Espírito-Santo
Dept. of Electromechanical Engineering
University of Beira Interior
IT-Instituto de Telecomunicações
Covilhã, Portugal
aes@ubi.pt

Riccardo Brama
ICT Consultant
Carovigno (BR), Italy
r.brama@ieee.org

***Abstract*—The interoperability and integration challenges to quickly develop Internet of Things (IoT) applications include heterogeneous IoT smart transducers (sensors and/or actuators) communicating through standardized or proprietary protocols, the need to integrate legacy sensor devices/systems, and the demand for rapidly developing friendly user interfaces/dashboards, therefore, integrating diverse IoT smart transducers with IoT platforms/middleware is a solution to overcome these challenges above. This paper introduces the development and orchestration of a specialized library of Node-RED nodes for enabling interoperable, standards-compliant transducer interfacing, specifically designed to support the Institute of Electrical and Electronics Engineers (IEEE) 1451.0-2024 and 1451.1.6-2025 standards. These nodes provide a low-code abstraction layer that simplifies the instantiation, discovery, and management of smart transducers. The proposed library categorizes nodes into three functional domains: Network Nodes, Transducer Nodes, and Util Nodes for synchronous and asynchronous telemetry acquisition. Util Nodes allow conversion from hexadecimal to readable formats (e.g., int, float, string), including a dedicated TEDS (Transducer Electronic Data Sheet) decoding engine and a router for persistent message queue telemetry transport (MQTT) topic management. A temperature-monitoring scenario demonstrates how the node library eliminates the need for manual binary header manipulation and bit-shifting. The primary contribution lies in this modular-based flow architecture, which allows users to transition from a physical sensor connection to a live-streaming, units-corrected dashboard in minutes. Performance evaluations demonstrated minimal computational overhead, with decoding nodes executing in 0.07 to 0.42 ms, while network Round Trip Time (RTT) latencies ranged from a median of 465 ms (NCAP discovery) to 1228 ms (PHY TEDS extraction).**

Keywords—IEEE 1451, Node-RED Nodes, Low-Code IoT.

I. Introduction

Smart transducers (sensors and actuators) are widely used across the Internet of Things (IoT) and play a key role in real-time monitoring and control applications. The main challenges of IoT smart sensors and actuators include diverse connectivity, interoperability, and cybersecurity. The solutions to overcome these challenges include adopting standardized interfaces and communication protocols, conducting testing and certification. The Institute of Electrical and Electronics Engineers (IEEE) 1451 family of standards provides a common, network-independent communication interface for connecting smart sensors and actuators to microprocessors and control networks. This framework is essential for modern IoT applications by addressing the complexities of device heterogeneity.

The IEEE 1451.0-2024 [1] defines common functions of IoT smart sensors and actuators, sensor network devices, network services, transducer services, and transducer electronic data sheet (TEDS) formats to help achieve interoperability in network and transducer interfaces. Also defines the universal unique identification (UUID), security framework, and time synchronization framework. IEEE 1451.1.6 [2] defines a standardized method for transporting IEEE 1451.0-2024 command and reply messages over a sensor network using the popular message queue telemetry transport (MQTT) protocol to handle IEEE 1451 sensor communications. Fig. 1 (a) shows a four-layer architecture of IEEE 1451.0 and 1451.1.6-based IoT smart sensor networks, which consist of APP (application), MQTT broker, network-capable application processors (NCAP), and transducer interface modules (TIM). These IEEE 1451-based devices equipped with UUID, TEDS, and standardized interfaces and communication protocols can achieve global identity, time synchronization, secured network communications, and interoperability. Fig. 1 (b) shows the four-layer architecture of the IoT framework, which consists of the

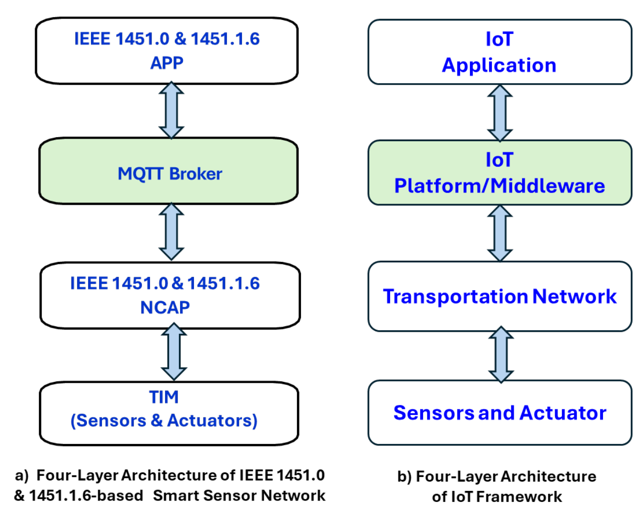

Fig. 1 Four-layer architecture of IEEE 1451.0 and 1451.1.6-based smart sensor network and IoT framework.

sensors and actuators layer, the transportation network layer, the IoT platform/middleware layer, and the IoT application layer [3]. The transducer layer comprises various sensors and controllers, including temperature and humidity sensors. The transportation network layer enables communication of information from sensors and actuators to the IoT Platform, and of control commands from the platform to sensors and actuators. The communication network can use a variety of transmission technologies, including wireless and wired networks. The IoT platform layer includes device management, connectivity and management, data sharing, information exchange, and processing analysis. The IoT application layer is responsible for providing application-specific services for use across scenarios such as Industry 4.0, Smart Homes, and Smart Cities. Security is cross-cutting for all four layers to ensure secure IoT applications.

To speed up IoT application implementations, interoperability and integration challenges persist: heterogeneous smart transducers communicating via different standardized protocols, the need to integrate diverse sensors/devices with IoT platforms/middleware, and the demand for rapidly developed user interfaces/dashboards for non-technical end users/operators. Therefore, to integrate diverse IoT sensors/devices with IoT platform/middleware and quickly prototype and orchestrate complex data flows from diverse IoT sensors/devices is a solution to overcome the challenges mentioned above. Node-RED [4] is an open-source, low-code middleware/platform built on Node.js that uses a visual, flow-based programming model to connect hardware, APIs, and online services. Key advantages of Node-RED for IoT include visual programming and ease of use, rapid prototyping and built-in dashboarding, multi-protocol support (e.g., HTTP, MQTT, TCP/UDP), a vast Node-RED node library, lightweight, and cross-platform operation [5]. It is widely used in IoT applications for its efficiency in handling real-time data and its accessibility to both developers and end users. To address these challenges, the primary contributions are a custom IEEE 1451 Node-RED library, a low-code abstraction layer that eliminates manual binary payload construction, and a validated, automated multi-stage transducer discovery pipeline.

This paper focuses on integrating IEEE 1451 smart transducers with IoT Node-RED middleware/platform by defining and developing IEEE 1451-based Node-RED nodes embedded inside the Node-RED platform, which will create highly interoperable, scalable, and plug-and-play IoT applications. Related works are described in Section II. Section III describes architecture and implementation. Section IV describes use cases and evaluation results. The summary and conclusion are described in Section V.

II. Related Work

There are works developed using the IEEE 1451 family of standards for transducer management and communication. Nishi and Lee introduced a standards-based sensor network for IoT applications, based on the IEEE 1451.0 and 1451.1.6 draft standards [6]. A few use cases are shown, along with results, to verify certain network service specifications for the sensor network based on 1451.0 network services and 1451.1.6 MQTT interfaces, thereby improving draft specifications and achieving interoperability. Rocha et al. [7] presented an interoperable IEEE 1451 semantic digital twin (DT) using the JavaScript Object Notation for Linked Data context within the NCAP to semantically encode data by referencing an external IEEE 1451 ontology. Rocha et al. [8] introduced the IEEE 1451.0-based Web of Things (WoT) ontology. The 1451.0-based WoT thing model consists of identification and context, behaviors, properties, data schema, security, time synchronization and protocol bindings such as NCAP and TIM. An example of a WoT thing description (TD) for an IEEE 1451.0-based smart sensor is provided. Song et al. [9] presented an IEEE 1451-based smart transducer digital twin framework (DTF), consisting of an IEEE 1451-based smart transducer physical twin (PT), its corresponding DT, and bidirectional information exchange between them for real-time monitoring and control. The IEEE 1451.0 and 1451.1.6-based smart transducer DT is developed using Node-RED and the MQTT protocol. Song et al. [10] introduced a harmonization model and implementation of interactions among IoT devices with diverse interfaces/connectivity and communication protocols to achieve interoperability. A harmonization implementation of interactions of the IEEE 1451-based IoT devices and MODBUS devices has been provided in the paper to test and verify that the proposed harmonization model works well. This harmonization model and implementation of interactions among IoT devices will provide a solid foundation for the IEEE P1451.99 standard specification. These research and development works mentioned above mainly focus on the implementation of IEEE 1451 standards-based sensor networks using MQTT and Node-RED, not the extension of Node-RED with 1451 functional nodes.

Recent advancements in environmental monitoring, such as the system proposed by Cabuk [11], leverage IoT sensors to track ambient conditions and transmit data to cloud infrastructures. Their approach utilizes Node-RED alongside modern wireless protocols to streamline data collection and real-time visualization. Thomas et al. [12] introduced a smart home automation system that employs MQTT, Node-RED, and a Raspberry Pi with a sensor kit. Uzougbo et al. [13] investigated the integration approach of Node-RED and IoT analytics for real-time data processing and visualization in the context of the IoT. Developing an industrial IoT system using Node-RED and IoT analytics to process and visualize real-time data from multiple sensors demonstrates efficiency, reduces development time and complexity, and provides a flexible, scalable solution. Guerbaoui et al. [14] presented a smart environmental monitoring system that leverages IoT technology and data analytics to monitor temperature and humidity levels in real-time. The MQTT protocol is used to transmit sensor data to a cloud-based HiveMQ Broker. Node-RED that subscribes to MQTT topics is used to process and visualize sensor data in real time on the dashboard, allowing access to actionable insights into environmental conditions. The benefits include improved decision-making, proactive maintenance, and enhanced productivity. These research and development works mentioned above mainly focus on IoT applications using MQTT and Node-RED, did not integrate IEEE 1451 with Node-RED. This paper primarily focuses on integrating IEEE 1451 smart transducers with the Node-RED middleware/platform by defining and developing IEEE 1451-based Node-RED nodes embedded within the platform, thereby creating highly interoperable, scalable, and plug-and-play IoT applications. Previous works that utilize generic MQTT nodes for IoT integration uniquely embed IEEE 1451 binary serialization into native visual nodes, advancing the state-of-the-art by removing the need for manual data parsing.

III. System Architecture

The system architecture is based on the IEEE 1451 family of standards, connecting the physical transducers to a higher-level monitoring and control environment. There are five main components, as shown in Fig. 2:

- APP with Node-RED (Management Client): Acts as the high-level application manager. It provides the user dashboard and coordinates the multi-stage discovery and data acquisition pipeline.
- MQTT Broker: The intermediary for the IEEE 1451.1.6 communication, enabling asynchronous, topic-based messaging between Node-RED and the physical NCAPs.
- NCAP: Acts as a gateway. It handles the logical conversion between transducer-specific data and the network-side protocol. In this implementation, the physical NCAP implements the IEEE 1451.0 logical layer.
- TIM: The low-level hardware containing sensors/actuators and their associated TEDS (Transducer Electronic Data Sheets). It communicates with the NCAP via the Transducer Independent Interface (TII).
- Smart transducer: a combination of NCAP with TIM, which is shown in Fig. 2.

The Node-RED runtime assumes that the APP interfaces directly with the NCAP via the MQTT broker. APP acts as the intelligent orchestration layer handling the standard network-facing operations and hardware-facing controls. Node-RED handles message routing and discovery, maintains state, polls data, and processes event callbacks without forcing the user to manually craft byte-level payloads. The extension was

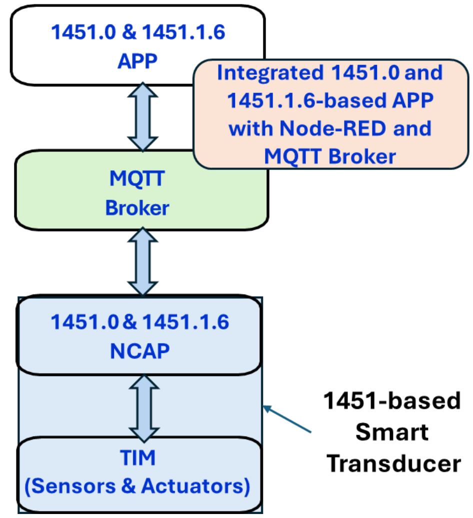

Fig. 2. Four-layer architecture of integrated MQTT and Node-RED with IEEE 1451.0 and 1451.1.6-based IoT framework.

engineered using a hybrid design pattern specifically tailored to mirror the structure of the IEEE 1451 standards. Each node in the library encapsulates the construction of a standards-compliant IEEE 1451.0 binary message, shielding the developer from the byte-level details of header assembly. The full picture of the system is presented as a sequence diagram, as shown in Fig. 3. As shown in Table I, the nodes implemented include the IEEE 1451 Network Nodes, Transducer Nodes, and Utils Nodes.

Network Nodes encompass operations responsible for topological and data-exchange functions, such as nodes for specific phases of the device life cycle, such as Discovery and Read TEDS and Data, and 1451.1.6 topic Nodes. Data and TEDS access through discrete blocks, with nodes such as Read Sample Data and Read Block Data, supporting synchronous, asynchronous, and callback modes. IEEE 1451.0 network service messages consist of a standardized frame structure: a Network Service Type, a Network Service ID, a Message Type (Command or Reply), a Message Length, followed by a variable-length payload whose fields depend on the specific

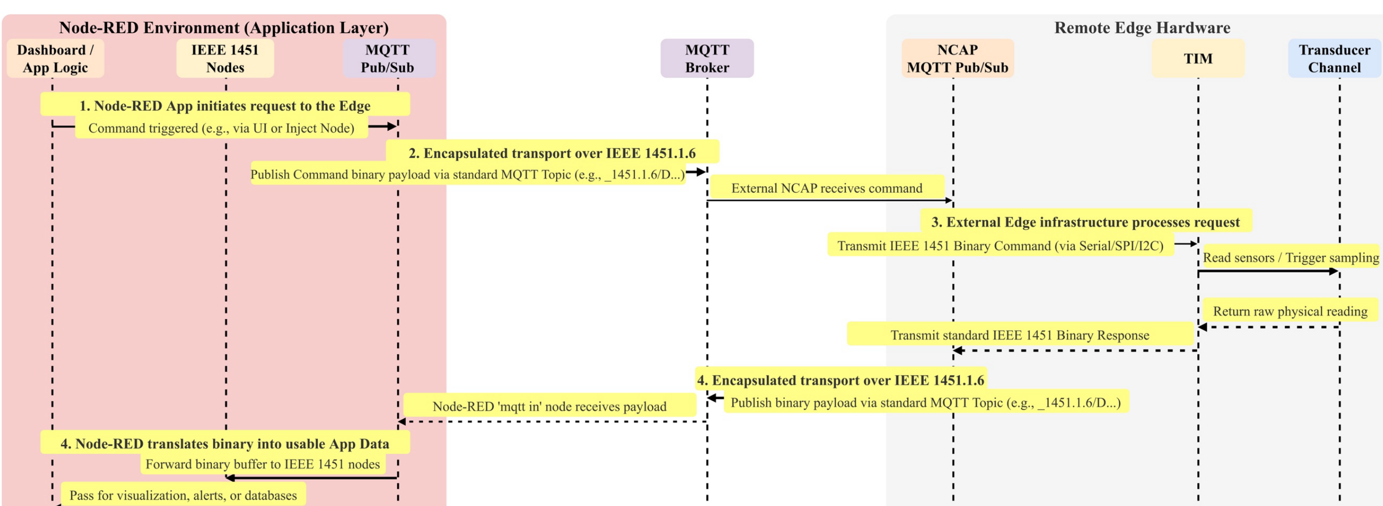

Fig. 3. Sequence Diagram of the Architecture.

service being invoked. As a concrete example, the NCAP Discovery node is considered. When triggered, it internally sets the fixed header bytes Discovery Services and NCAP Discovery, marks the message as a Command, calculates the payload length as 16 octets (the size of the application UUID), and writes these values into a binary Buffer at their specified offsets. The resulting 21-byte frame is then hex-encoded and placed for onward transmission via the native MQTT node. The user's only responsibility is to provide the appID (e.g., UUID) through the node configuration panel or through the incoming message. The binary serialization, field ordering, and length calculation are entirely abstracted. This same pattern scales to more complex operations, such as Read TEDS or TransducerChannels. The node resolves its identifiers dynamically first from the incoming message properties, then from its configuration panel, and finally from the flow context populated by prior discovery stages, enabling a fully automated pipeline where no manual UUID entry is required after the initial bootstrap. The IEEE 1451.1.6 MQTT nodes are customized IEEE 1451.1.6-MQTT-topic bridging nodes, and generated byte payloads are dynamically assigned standard-compliant publication topics based on their target Type of Message (TOM) and Logical Network Site (LNS) hierarchy (e.g., _1451.1.6/D0/UBI-NCAP2). By structuring the topic string precisely to the standard, these frames seamlessly plug into Node-RED's core, generic MQTT in and out communication nodes. This decoupling closely aligns with Node-RED's overarching philosophy, allowing standard network protocols to deliver the proprietary IEEE 1451 datagrams without conflict.

Transducer Nodes control the hardware states. Building these Nodes as discrete Nodes for high-level, single-operation states requires minimal configuration (e.g., TIM Wakeup, TIM Sleep, TIM Reset). Composed Nodes for complex operations are used to avoid UI clutter. Nodes such as Common Commands or Transducer Idle are used for dynamic configuration dropdowns. The Node-RED backend cascades through these nested UI inputs to programmatically aggregate the appropriate CommandClassID and CommandFunctionID bytes at runtime, rather than requiring separate visual nodes.

The IEEE 1451 Util Nodes provide utility nodes that bridge the gap between standards-compliant binary messaging and the event-driven Node-RED environment. Collectively, these Util Nodes provide the connective tissue that allows the command and decoding nodes to operate as a fully automated, zero-configuration pipeline. Some nodes developed for the IEEE 1451 are presented in Table I.

Table I. Nodes developed for the IEEE 1451 Node-RED library.

Functional Domain	Node Name	Function
Network Node	NCAP / TIM / Transducer Discovery	Handles IEEE 1451.0 discovery services, abstracting binary header creation to locate NCAPs, TIMs, and channels.
	Read Sample / Block Data	Generates compliant data acquisition commands supporting synchronous, asynchronous, and callback modes.
	Read TEDS	Requests specific TEDS segments (e.g., Meta, TransducerChannel, PHY) from the target transducer.
	_1451.1.6 Topic Generator	Dynamically assigns compliant MQTT publication topics based on IEEE 1451.1.6 rules to bridge standard MQTT transport.
Transducer Node	Common Commands	Provides high-level control states for hardware management (e.g., TIM Wakeup, Sleep, Reset).
Util Node	Byte to Hex	Normalizes incoming MQTT binary buffers into hexadecimal strings.
	Extract-UUIDs	Performs targeted binary slicing to extract 16-byte hardware UUIDs from flat IEEE 1451.0 payloads.
	Discovery Helper	Enforces sequential ordering of the discovery pipeline and extracts/persists hardware names and UUIDs.
	TEDS Routing Helper	Manages the detection and routing of TEDS messages, securing handover from the discovery phase.
	Data Routing Helper	Handles stateful routing for all synchronous or asynchronous data-related commands.
	TEDS Decoder	A processing engine that parses hexadecimal TEDS TLV structures into human-readable text.
	Data Decoder	Interprets raw hexadecimal payload data and converts it into appropriate types (integer, float, string, etc.).
	Dashboard Formatter	Utilizes TEDS metadata stored in the flow context to automatically perform unit conversions for UI elements.
	Dashboard Logger	Renders a real-time activity table for monitoring network commands and replays.

IV. USE CASE AND EVALUATION RESULTS

The use case is a smart monitoring system. Using the discovery service to discover the NCAPs and its related TIMs (sensors) and start monitoring the environment. It is composed of the Node-RED application running on a MacBook Air M1, the NCAP and the TIM, implemented on an ESP32-WROOM Xtensa dual-core 32-bit LX6 microprocessor running up to 240 MHz, with 448 KB ROM and 520 KB SRAM, connected to a DS18B20 temperature sensor.

The implemented discovery pipeline, shown in Fig. 3, follows the sequence and is compliant with the IEEE 1451.0-2024 discovery services and the IEEE 1451.1.6 MQTT topic mapping. In Stage 1 (NCAP Discovery), the Node-RED application publishes a discovery command (Network Service) to a bootstrap MQTT topic (e.g., *_1451.1.6/D0/UBI*). The remote NCAP replies with the same topic, including its UUID and human-readable name (e.g., UBI-NCAP2), which the Discovery Helper node extracts and persists in the 1451.1.6 Topic Generator node. After a deliberate delay to ensure hardware buffer stability, in Stage 2 (TIM Discovery). The APP issues a TIM enumeration command (Service) on the now hardware-specific topic (*_1451.1.6/D0/UBI-NCAP2*), receiving the connected TIM's UUID and name. Stage 3 (Transducer Discovery) repeats this pattern to retrieve the list of available transducer channels and their names. Upon completion, the Discovery Helper signals a handover to Stage 4 (TEDS Acquisition), where the system sequentially requests the Meta, TransducerChannel, User's Transducer Name, and PHY TEDS segments, along with the Network Service for the discovered transducer. A key design property throughout this sequence is topic symmetry with persistence: per IEEE 1451.1.6, commands and replies share the same topic path, and once the *_1451.1.6* Topic Generator node learns the hardware name during Stage 1, it locks onto that path for all subsequent operations, preventing topic drift back to bootstrap defaults during recurring data loops.

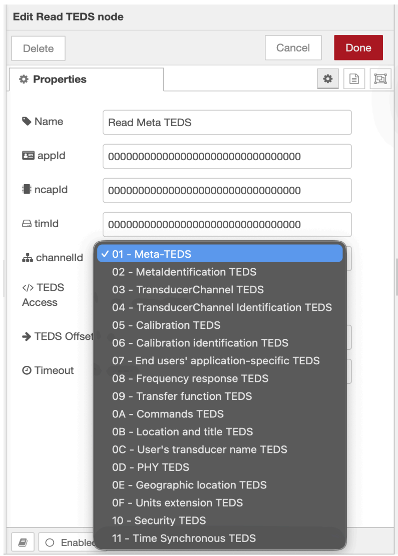

Fig. 4. Read TEDS Node structure.

The Read TEDS command is defined within the Read TEDS node. The appId, ncapId, timId, and channelId are passed in at runtime for this node. The user needs to select the TEDS access code, input the TEDS offset and define a timeout, as shown in Fig. 4. The answer to this message is received by the MQTT node, decided by the Byte to Hex node and passed to the TEDS Decoder node. This node allows automatic detection or selection of one of the IEEE 1451.0-2024-defined TEDS. This routing performance is managed by the TEDS Routing Helper node. The TEDS are interpreted, and the hexadecimal values are parsed and displayed as text in the Node-RED UI using the TEDS decoder node.

The next step is to start reading the TransducerChannel, which is a temperature. The Data Routing Helper was developed to handle all data-related commands and to respond to synchronous or asynchronous requests, such as those from the Read Sample Data node. The answer is passed to the Data Decoder, which was built to interpret the data as an integer. The interpretation of this data is passed to the Dashboard Formatter node and displayed in a text field, gauge or graph, for example. Fig. 5 shows the evaluation scenario using a Node-RED flow and the related descriptions in Table I.

The system latency, as measured using Round Trip Time (RTT) measurements, encompasses the entire network traversal from the APP on a MacBook M1 machine sending IEEE 1451.0 commands over the Internet to an NCAP (ESP-32 WROOM), which receives and replies. Internal computational overhead for data decoding and dashboard rendering is excluded from the RTT and reported separately as decoding latency to distinguish between communication and computation bottlenecks. The TEDS and Data Decode nodes execute with a median time of 0.07-0.42 ms. The RTT latency is presented in Table II.

Table II. RTT Latency Discovery and Read Commands.

Command	Average (ms)	25% (ms)	Median (ms)	75% (ms)
NCAP Discovery	474.02	326.84	465.71	616.24
TIM Discovery	1133.87	1073.21	1117.90	1170.18
Transducer Discovery	1150.94	1126.40	1157.52	1171.79
Read Meta TEDS	1187.02	1157.28	1174.52	1212.28
Read TransducerChannel TEDS	1218.87	1193.06	1227.44	1240.90
Read User's Transducer Name TEDS	1230.40	1197.33	1226.61	1231.90
Read PHY TEDS	1304.24	1226.09	1228.88	1273.76
Read Data Sample	539.31	340.63	454.87	572.46

The IEEE 1451 family of standards library enables a comprehensive layer of abstraction that shields them from the low-level complexity inherent in standards-compliant

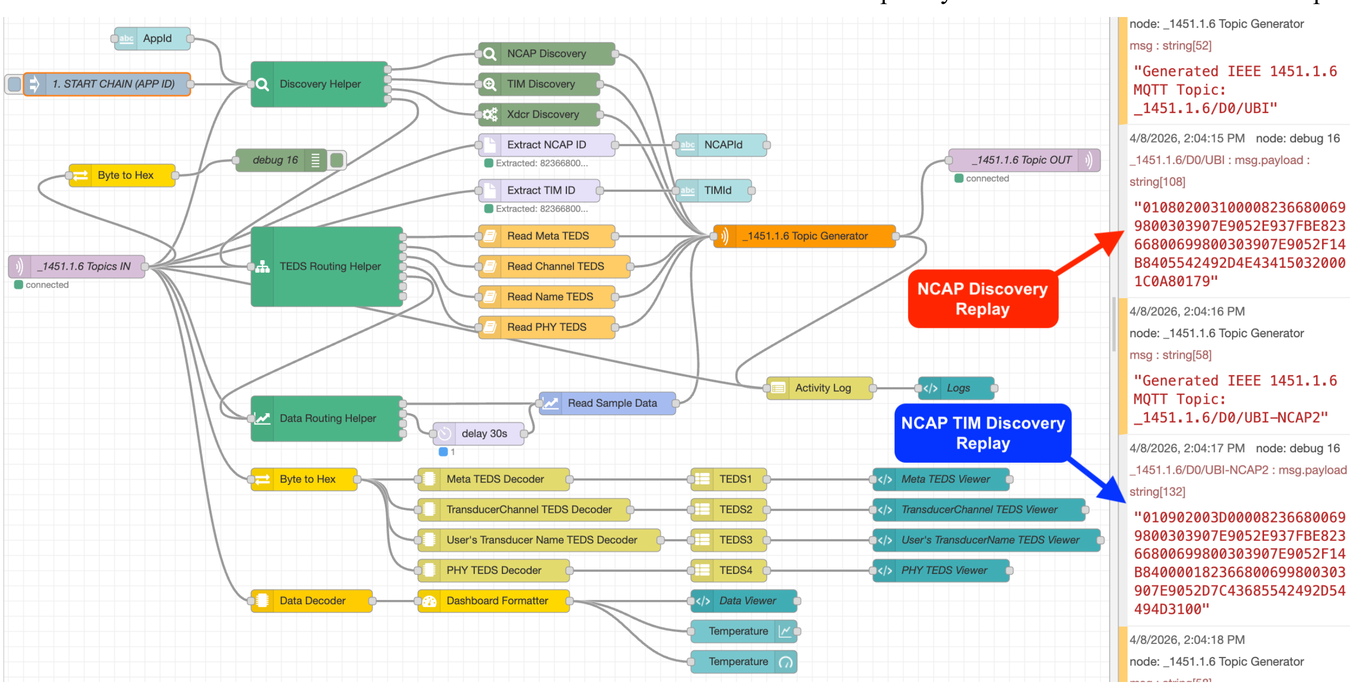

Fig. 5. Evaluation Scenario Node-RED Flow.

transducer communication. The common implementation of IEEE 1451.0 messaging requires users to manually construct byte-encoded binary frames, assembling fixed-length headers with precise field ordering, and interpreting TEDS binary data by parsing type-length-value (TLV) pairs. The IEEE 1451 nodes encapsulate all of this binary serialization internally. A user simply configures high-level parameters through the node's visual configuration panel, and the node programmatically allocates the appropriate buffer size, writes each field to its specified offset, and provides a fully compliant hexadecimal payload. The library automates the dynamic resolution of hardware identifiers throughout the discovery pipeline. UUIDs are persisted in the Node-RED flow context and automatically injected into subsequent data-acquisition commands, eliminating the need to manually copy and paste hexadecimal strings between operations. This low-code library speeds up prototyping and deployment while lowering the barrier to adopting the IEEE 1451 standards, enabling researchers, system integrators, and IoT practitioners to focus on their application logic rather than the intricacies of binary protocol implementation. While continuous MQTT connections required by IEEE 1451.1.6 draw more power than deep-sleep protocols, the integration of targeted 'TIM Sleep' nodes in our library allows users to strategically manage edge-device battery lifecycles.

V. Conclusion

This paper presented the design, implementation, and validation of IEEE 1451 nodes, a modular library of custom Node-RED nodes that provides a low-code abstraction layer for the IEEE 1451.0 and 1451.1.6 standards. The library's primary contribution lies in encapsulating the complexity of binary message construction, multi-stage discovery orchestration, TEDS decoding, and MQTT topic management within reusable, visually composable nodes, enabling a fully functional, standards-compliant sensor pipeline to be deployed in minutes with minimal network overhead (~474 ms for discovery to ~1300 ms for TEDS retrieval) rather than the days of custom integration coding traditionally required. These tools serve as the framework that converts the IEEE 1451 standards from a mere specification reference into a practical, plug-and-play development environment. Permitting democratizing access to the IEEE 1451 standards through a visual, event-driven environment.

The current limitations are related to IEEE 1451.0-2024 and IEEE 1451.1.6 standards. The TEDS decoding engine covers most mandatory TEDS, but does not yet implement the full catalog of TEDS templates defined by the standard. The security model relies on standard MQTT-level mechanisms and does not yet integrate the specialized IEEE 1451.0 security services, such as transducer-level access control and message-level integrity verification.

Future work includes expanding the library to support additional IEEE 1451 physical transport layers and enabling heterogeneous multi-protocol deployments. The TEDS decoder will be expanded to cover the complete set of standardized TEDS templates, enabling automatic sensor linearization and compensation directly within the Node-RED flow. Integration with the WoT architecture will be explored, leveraging the TEDS metadata to automatically generate WoT Thing Descriptions, thereby bridging the IEEE 1451 ecosystem with broader semantic interoperability frameworks. The library will be published to the Node-RED community palette registry to enable adoption, community contributions, and independent validation across diverse IoT deployment scenarios.

Acknowledgment

Helbert da Rocha and António Espírito-Santo are financially supported by the Project GreenAuto: Green Innovation for the Automotive Industry, n° C644867037 - 00000013, investment project nr. 54, from the Incentive System to Mobilizing Agendas for Business Innovation, funded by the Recovery and Resilience Plan and by European Funds NextGenerationEU. Unless otherwise noted, NIST work was funded solely by the United States Government.

** CERTAIN COMMERCIAL PRODUCTS OR COMPANY NAMES ARE IDENTIFIED HERE TO DESCRIBE OUR STUDY ADEQUATELY. SUCH IDENTIFICATION IS NOT INTENDED TO IMPLY RECOMMENDATION OR ENDORSEMENT BY THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, NOR IS IT INTENDED TO IMPLY THAT THE PRODUCTS OR NAMES IDENTIFIED ARE NECESSARILY THE BEST AVAILABLE FOR THE PURPOSE.

References

[1] IEEE 1451.0-2024 Standard for a Smart Transducer Interface for Sensors and Actuators--Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats.

[2] IEEE 1451.1.6-2025 Standard for a Smart Transducer Interface for Sensors, Actuators, and Devices - Message Queue Telemetry Transport (MQTT) for Networked Device Communication.

[3] IEEE 2413-2019 standard for an Architectural Framework for the Internet of Things (IoT), DOI: 10.1109/IEEESTD.2020.9032420

[4] NodeRED, [Online]. Available: https://nodered.org/

[5] NodeRED features, Pros, Cons & Alternatives, [Online]. Available: https://zeroqode.com/no-code-tools/nodered-review/#145700ef34bc8030b23afd3e7fcf522c

[6] H. Nishi; K. B. Lee, Implementation of IEEE P1451.0 and P1451.1.6 Sensor Networks, IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, Oct. 2022, Brussels, Belgium, DOI: 10.1109/IECON49645.2022.9968652

[7] H. Da Rocha, J. Pereira, R. Abrishambaf, A. Espirito Santo, An Interoperable Digital Twin with the IEEE 1451 Standards. Sensors, 2022, 22, 7590, [Online]. Available: https://doi.org/10.3390/s22197590.

[8] H. da Rocha, E. Song, A. Espirito-Santo, and R. Brama, IEEE 1451.0-based Web of Thing (WoT) Ontology, Proceedings of the 2024 Annual Conference of the IEEE Industrial Electronics Society (IECON), Chicago, IL, US, [online], https://doi.org/10.1109/IECON55916.2024.10905970,

[9] E. Song, Shanaka P, A., H. Nishi, T. Rooth, IEEE 1451-based Digital Twin Framework for Real-Time Monitoring and Control of Smart Homes, IECON 2025, Oct. 14-17, 2025, Madrid, Spain, [Online]. Available: DOI: 10.1109/IECON58223.2025.11221119.

[10] Y. Song, P. Waher, H., da Rocha, R. Brama, H. Nishi, and A. Espirito-Santo, T. Roth, "Harmonization Model & Implementation of Interactions among IoT Devices," IECON 2025 – 51st Annual Conference of the IEEE Industrial Electronics Society, Madrid, Spain, 2025, pp. 1-6, doi: 10.1109/IECON58223.2025.11221746

[11] A. S. Cabuk, Air Quality Regulatory of Primary Care Clinics System Designed with IoT Using the Node-RED, Water Air Soil Pollut, Vol 236, 370, 2025, DOI: https://doi.org/10.1007/s11270-025-08020-z

[12] Thomas, M. Kumar, S. Darshan SL, and Prashanth BS, Towards Comprehensive Home Automation: Leveraging the IoT, Node-RED, and Wireless Sensor Networks for Enhanced Control and Connectivity, the Proceedings of Eng. Proc., 2023, Eng. Proc. 2023, 59(1), 173; https://doi.org/10.3390/engproc2023059173

[13] O. I. Uzougbo, O. A. Olanrewaju, A. K. Kayode, Node-RED and IoT Analytics: A Real-Time Data Processing and Visualization Platform, Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS), A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ), Vol 1, Issue 1, 2024 Publication Edition, DOI: https://doi.org/10.5281/zenodo.13856860

[14] M. Guerbaoui, S. El Faiz, A. Ed-Dahhak, A. Lachhab, B. Benhala, Z. Bakziz, I. Ichou and A. Selmani, From Data to Decisions: A Smart IoT and Cloud Approach to Environmental Monitoring, E3S Web Conf., Vol. 601, January 16, 2025, The 3rd International Conference on Energy and Green Computing (ICEGC'2024), DOI: https://doi.org/10.1051/e3sconf/202560100008