Prototyping Intelligent IoT Smart Hubs on Repurposed TV Boxes: A Low-Cost Approach to Practical Engineering Education

Cláudio R. M. Silva
Communication Engineering
Department of the Rio Grande do Norte
Federal University
Natal/RN, Brazil
claudio.muniz@ufrn.br

João Paulo N. dos Santos
Network Convergence Laboratory of
the Rio Grande do Norte Federal
University
Natal/RN, Brazil
joao.paulo.nascimento.047@ufrn.edu.br

Felipe Augusto C. M. Silva
Network Convergence Laboratory of
the Rio Grande do Norte Federal
University
Natal/RN, Brazil
felipe.silva.088@ufrn.edu.br

Abstract*—Digital Transformation and the Internet of Things (IoT) are interrelated concepts. These initiatives have substantially expanded the capabilities of IoT, rendering such systems more intelligent and autonomous across diverse automation domains. Although these advances have had considerable impact across industrial, residential and commercial domains, they also have presented interesting opportunities and challenges within educational environments. The reason for this is a growing demand for curricula that foster practical competencies, enabling students to design, implement and interact with intelligent, networked systems. This work presents the design and implementation of a low-cost IoT smart hub based on modified TV boxes, featuring natural language support through integration with NLP tools. The methodology consists of a prototyping approach that involves a set of validation tests on a control plant defined in an educational environment. The results have shown that the implementation is technically viable and pedagogically effective for hands-on engineering education.

***Keywords*—Natural language processing, IoT, LLM, MQTT**

I. Introduction

The increasing convergence of Internet of Things (IoT), Artificial Intelligence (AI) and Natural Language Processing (NLP) is transforming how intelligent systems are designed, deployed and experienced. This technological convergence has enabled the emergence of cognitive cyber-physical systems [1][2].

The integration of IoT with AI-driven reasoning allows connected systems not only to perceive and process information but also to understand context, make decisions and interact naturally with humans through language [3][4].

In the context of engineering education, this convergence introduces new possibilities for developing hands-on, interdisciplinary learning environments that combine connectivity, computation and cognition.

According to a recent report from the Organization for Economic Co-operation and Development (OECD) [5], It is expected that future engineers develop important competencies such as problem-solving, critical thinking and collaborative innovation to operate effectively in an AI-enabled technological domain.

Unfortunately, the widespread integration of these technologies into teaching laboratories remains limited, largely due to the high cost and proprietary nature of commercial IoT and AI development kits [6].

A promising alternative available to certain institutions involves repurposing consumer electronic devices, such as Android-based TV boxes, into functional IoT edge nodes capable of running AI-enhanced services. When reconfigured with Linux-based operating systems, these devices can host open-source frameworks that support IoT communication protocols and AI inference engines [7]. This is made possible by a public donation program led by the Brazilian Federal Revenue Service in Brazil that redirects seized electronic equipment to educational institutions. This initiative not only enables the sustainable reuse of hardware but also promotes digital inclusion and contributes to reducing electronic waste.

Repurposing decommissioned TV boxes to work like IoT Smart Hubs offers a practical and sustainable way to extend the lifespan of consumer electronic devices [7]. These devices, once reconfigured with open-source, Linux-based operating systems, can host IoT communication frameworks and AI inference engines, transforming otherwise discarded hardware into fully functional edge-computing nodes.

An Intelligent IoT Smart Hub is a central device designed to connect, manage and locally process data from multiple IoT sensors and actuators [8]. This kind of device incorporates embedded processing capabilities and artificial intelligence algorithms, enabling edge computing and autonomous decision-making. By processing data at the edge, they lower latency, enhance privacy and optimize bandwidth usage, making them essential components in IoT ecosystems.

This paper presents the development of an intelligent IoT Smart Hub implemented on a repurposed Azamerica I7 Gamer TV Box, which integrates REST APIs, MQTT communication, NLP and a local large language model (LLM) to enable human–machine interaction. A proof-of-concept prototype was iteratively designed, implemented and evaluated through educational laboratory experiments involving IoT sensors and network convergence protocols.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 describes the system architecture and the prototyping methodology; Section 4 presents the experimental setup and the educational evaluation; and Section 5 concludes the paper and outlines future research directions.

II. Related Work

The development of low-cost IoT platforms has been widely explored in the context of engineering education due to their accessibility, flexibility and ability to promote hands-on learning.

The present work is the fourth iteration in a series of research initiatives that aim to progressively enhance the educational value and technological capabilities of embedded systems using open-source tools and repurposed hardware.

The first stage of this research line was introduced in [9], where an IoT gateway capable of bridging Modbus and MQTT protocols was developed to support integration between legacy industrial systems and cloud infrastructures. This foundational work demonstrated the feasibility of combining traditional automation technologies with modern IoT communication standards.

The second and third stage [10] focused on expanding the system's hardware capabilities, introducing a low-cost MQTT hub with USB support for sensor connectivity. This prototype enabled direct interaction with various USB-based IoT sensors and was designed with educational usage in mind, facilitating student engagement in embedded systems and networking projects. While these earlier studies emphasized integration, affordability and educational deployment, they did not explore user accessibility through natural language interfaces.

The current stage addresses this gap by incorporating natural language processing (NLP) as a key feature of the IoT hub, enhancing user interaction and aligning the platform with current trends in AI-enhanced systems. Several recent studies have investigated the integration of IoT, AI and TV box technologies. [11] explored the role of smart TVs within IoT environments, identifying their capacity to serve as control and data hubs. [12] presents a comprehensive review of recent advancements in landslide monitoring systems using low-cost sensors and LoRaWAN communication within IoT frameworks. [13] proposes the creation of an intelligent energy sustainability hub to explore how IoT, big data, artificial intelligence and cyber-physical systems can be integrated to optimize energy use. [14] explored the synergy between generative AI and IoT systems, particularly the ability to synthesize training data and personalize user interactions in constrained environments.

Compared to these studies, the present work is distinguished by its strong educational orientation, use of repurposed TV boxes and integration of natural language interfaces tailored for learning environments.

III. System Architecture and Implementation

The system architecture was designed to provide flexibility and modularity capabilities, with a focus on the integration of sensors and actuators through voice or text commands interpreted by large language models (LLMs), as well as secure remote control via the MQTT protocol. An overview of the proposed architecture and its main use cases scenario are presented in Fig. 1.

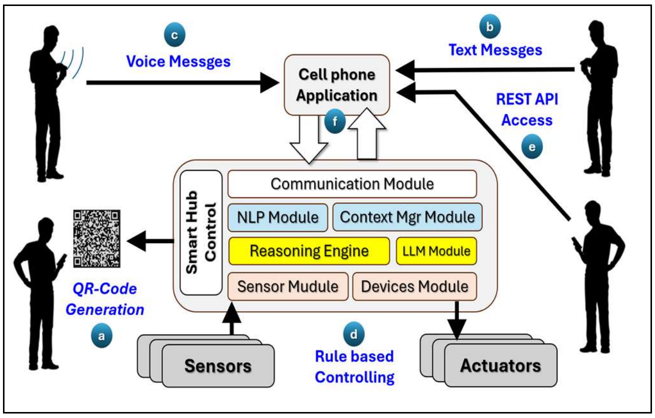

Fig. 1 Main components for the prototype and its use cases

One of the core functionalities of the proposed Smart Hub architecture is device identification and authentication via QR Code Fig. 1 - “a”). A QR Code is automatically generated based on the hub’s unique identification and authentication data, enabling seamless pairing with a companion control application. This application communicates with the Smart Hub through specific MQTT topics, ensuring secure and efficient data exchange. Another key feature is the control of the system via voice or text commands, which are processed either locally or remotely by a large language model (LLM).

These commands are interpreted in natural language, allowing users to interact with the system in a flexible and user-friendly manner and enabling the execution of a wide range of control actions (Fig. 1- “b” and “c”). The Smart Hub enables users to define automation rules in natural language, which are converted to JSON and processed by an internal rule engine to control sensors and actuators autonomously. Its modular architecture ensures interoperability and scalability, using repurposed TV boxes as low-cost, sustainable IoT hardware. Key modules include NLP for interpreting voice/text commands with LLMs (local or cloud-based), MQTT for secure bidirectional communication with a smartphone app and dedicated modules for sensors, actuators and overall coordination. The hybrid NLP approach allows local inference for privacy and efficiency or cloud inference for greater capability. The rule engine evaluates JSON-based conditions to trigger actions, while the general control module manages resources and system orchestration.

The Smart Hub prototype was implemented on a repurposed TV box model, the Azamerica I7. The device was modified to run Armbian Linux. It features 4 GB of RAM and 32 GB of flash storage. Its system-on-chip (SoC) is the Allwinner H6, integrating a 64-bit Quad-core ARM Cortex-A53 CPU. The system architecture was designed in modular form, with each component developed in Python (the server and the windows client application) and java (the smartphone client application) to ensure flexibility and ease of integration. A central control module coordinates the execution and data flow among subsystems. Communication between the Smart Hub and the control application is established via the MQTT protocol, using the paho-mqtt library. The hub subscribes to predefined topics to receive commands and publish status updates. Commands transmitted in voice or text form are processed by the Natural Language Processing (NLP) module, which supports both remote inferences using OpenAI’s APIs and local inference using lightweight models deployed through the LocalAI framework. The rule engine is based on a JSON schema for defining logical rules composed of conditions and actions. It operates in real time to evaluate sensor inputs and execute actions accordingly. Additionally, the onboarding process is facilitated through the automatic generation of a QR code containing the hub’s unique identification and authentication data, allowing secure and user-friendly pairing with the mobile application via MQTT. Sensor control is implemented through enabling integration with temperature sensors using USB interfaces, motion detectors and digital inputs. The actuator module provides control over devices such as relays, responding to either direct user commands or decisions made by the rule engine.

IV. Results

The results are interpreted in the context of the intended IoT scenarios, highlighting both the strengths and the limitations observed during experimental testing. In the first scenario, the Smart hub prototype is accessed by the user using a smartphone application or by using a windows client version running on Windows 11. The user is supposed to authenticate and choose among a set of smart hubs previously stored in the application. After authentication, each smart hub has to be registered by clicking in "Add" button and reading a QR Code printed on a label fixed under the smart hub.

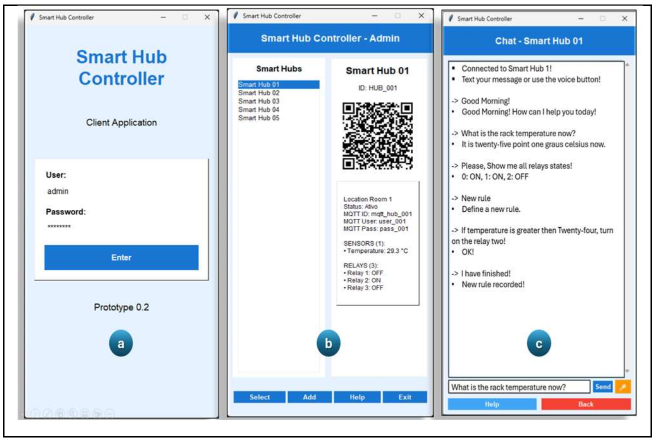

Fig. 2 Smart hub client programa screenshots: authentication screen ("a"), smart hubs previously stored ("b") and Natural Language Interface to the selected smart hub ("c").

A set of five smart hubs were stored in the client program by the process of scanning a QR-code attached to its console (first scenario shown in Fig. 1 - "a"). Fig. 2 shows the main screenshots for the GUIs (graphical user interfaces) used by the client application. The first one is the authentication GUI (Fig. 2 - "a"); The second one is the Smart hubs GUI (Fig. 2 - "b") and the third one is the Natural Language (NLP) interface with the selected smart hub (Fig. 2 - "c"). On NLP interface, a group of previous interactions are logged to show an example of how the smart hub can be accessed to gather information about a control plant and its sensors and actuators/relays. The control plant used in the validation tests is a simple control solution to keep the temperature of rack inside a safe range. See Fig. 3, below.

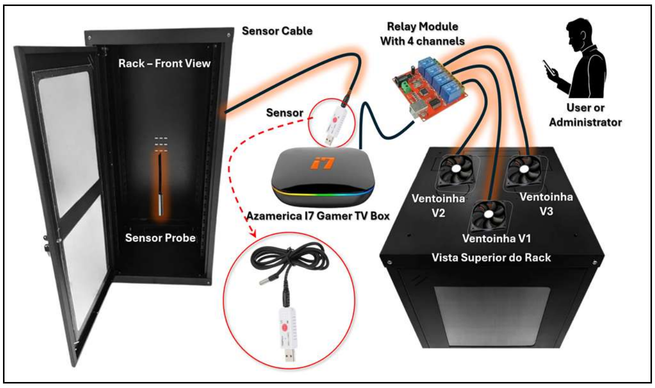

Fig. 3 Smart hub controlling the internal inside a rack of clusters.

It consists of a temperature sensor with a probe and three cooling fans connected to a USB relay module with four channels. This module and the sensor used inside the rack are both connected to two of the USB ports available on the modified TV box console (Fig. 3). The control plant had already been used in previous works to test how traditional smart hubs could be used in this type of scenario. In current experimentation the objective is to verify how feasible it would be to use an NLP device to control the same plant. The challenge now is to train an LLM model to extract commands used in plant control and, at the same time, answer generic user questions made by the user in natural language. The commands that must be extracted from conversations with users involve reading sensors (individually or in group), reading actuator/relay states (individually or in group), defining new rules for the rule engine and consulting information about abnormalities in the ongoing control.

Fig. 4 and Fig. 5 present two sequence diagrams showing how a LLM Model can be used to enable a more flexible and natural way to interface users and a control plant.

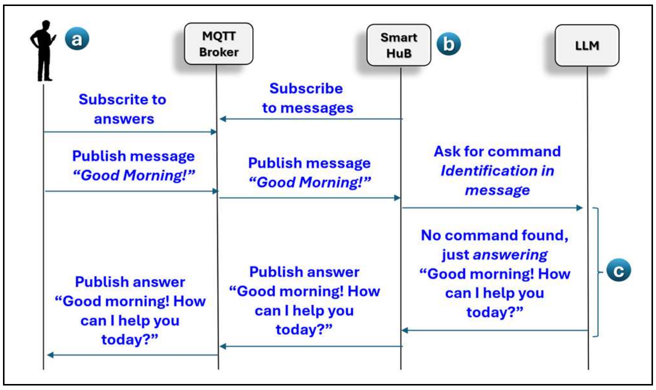

Fig. 4 Sequence Diagram representing first part of the dialog between the client program and the amart hub programa

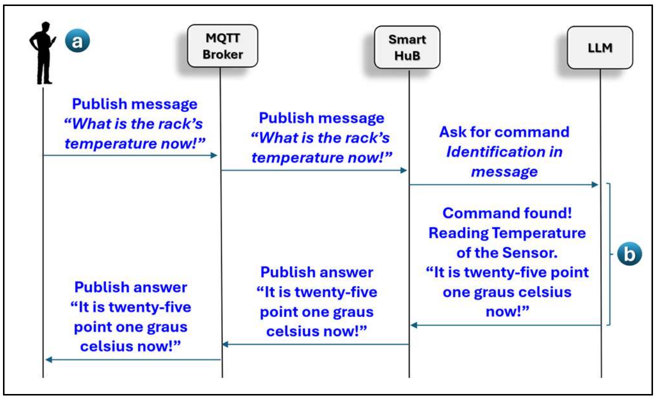

Fig. 5 Sequence Diagram representing second part of the dialog between the client program and the amart hub programa

In Fig. 4 = "c" and Fig. 5 – "b", the LLM module check if there is a command in the prompt or if it should just answer the prompt normally. Besides, considering the scenario shown in Fig. 1 - "d", the rule engine module embedded within a smart hub is responsible for controlling actuators based on the real-time state of connected sensors. Its behavior is defined through a set of rules expressed in a JSON file, allowing for a flexible and human-readable configuration. Each rule typically includes a sensor identifier, a condition to evaluate and an action to perform if the condition is met, such as turning an actuator on or off. At startup, the engine detects sensors and actuators, parses the rule file and validates syntax and semantics. It then runs a monitoring loop, sampling sensor data and evaluating rules to trigger corresponding actions, with conflict resolution and logging for traceability. Commands are sent via hardware interfaces or messaging protocols, while dynamic rule reloading and error handling ensure adaptability and robustness.

The prototype was tested in the control of a server rack temperature in a cloud computing lab scenario.

In MQTT-based applications used to control sensors and actuators, response time and its variation are very useful performance metrics. Response time captures the interval between request and response, while its variation reflects stability and predictability under load. These metrics were selected as the primary performance metric, as it captures the end-to-end duration of client–server interactions from the user's perspective and provides a practical measure of system behavior. See Table I below showing the results for a single client accessing a single smart hub.

TABLE I. ART AND SD FOR THE USAGE SCENARIOS

Usage Scenario	ASR (ms)	ART (ms)	Standard Deviation (ms)
Text Control using local LLM to access sensors	---	364.68	6.24
Text Control using local LLM to turn on/off actuators	---	360.22	2.51
Voice Control using local LLM to access sensors	936.20	360.73	1.16
Voice Control using local LLM to turn on/off actuators	1148.1	368,30	2.14

The current prototype was tested using Ollama with the LLM model "tinyllama:1.1b" (2GB). The final prototype was validated through the calculation of its Average response time (ART), Automatic Speech Recognition time (ASR) and Standard Deviation (SD) for the presented usage scenarios. Table I shows that the system achieves consistently low Average Response Times (ARTs) across all usage scenarios after several experiments with the final prototype. Text-based control for both sensors and actuators also maintained values close to 360 ms with low variability, confirming the efficiency of the client–server communication and ensuring smooth interaction, as response times below 500 ms are generally perceived as very fast. In the voice control scenarios, the additional delay is introduced by the Automatic Speech Recognition (ASR) process, ranging from 936 ms to 1148 ms. However, once recognition is complete, the ART remains comparable to text-based scenarios (~360 ms), indicating system stability and efficiency. Even when considering the total time (ASR + ART), the overall response remains below 1.5 seconds, which is acceptable according to human–computer interaction guidelines. Therefore, all evaluated scenarios can be considered adequate, validating the system.

V. DISCUSSION

The integration of natural language interfaces into IoT hubs offers major educational benefits, lowering the learning curve for students and encouraging intuitive experimentation with sensors and actuators. Using low-cost modified TV boxes, these systems show the feasibility of scalable, accessible tools with minimal infrastructure needs. Challenges include ambiguities in human language, which may cause misinterpretations in safety-critical contexts and reliance on the capabilities of underlying language models. Cloud-based LLMs provide better performance but raise concerns about connectivity and privacy, while local models allow offline use and data control but face hardware limitations. Thus, natural language interfaces are highly suited for education, prototyping and assistive technologies, while in industrial or real-time systems, where precision is essential, structured or graphical interfaces remain preferable.

VI. CONCLUSION

The development of the low-cost IoT smart hub on a modified TV box has proved to be a technically viable solution for hands-on engineering education. The proposed system successfully met all predefined functional requirements and operational scenarios, demonstrating robust performance in environments with constrained computational and financial resources. By leveraging modified TV boxes donated by the Brazilian Federal Revenue Service, the project capitalized on readily available hardware, promoting the sustainable reuse of electronic equipment. This strategy not only minimizes deployment costs, but it is also aligned with institutional goals for technological inclusion and environmental responsibility within public education systems. The integration of NLP interfaces enhanced the accessibility and intuitiveness of the IoT hub, allowing students to interact with sensors and actuators using natural language commands. This feature significantly reduced the learning curve for beginners and stimulated interest in IoT and artificial intelligence topics through practical, real-world applications.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements and future directions" Future Generation Computer Systems*, 2013.*

[2] M. McTear, Z. Callejas and D. Grio*l, "*The Conversational Interface: Talking to Smart Devices*"*. Springer, 2016.

[3] A. Kukulska-Hulme, "Intelligent Assistants in Education: Capabilities and Limitations". In: 19th World Conference on Mobile, Blended and Seamless Learning (mLearn2020), mLearn Conference Proceedings, International Association for Mobile Learning, pp. 123–124. 2020.

[4] P. Schermerhorn and M. Scheutz, "Natural language interactions in Distributed Network of Smart Devices". International Journal of Semantic Computing. vol. 02, n. 04, pp. 503-524, 2008.

[5] OCDE, "The Future of Education and Skills 2030" OECD Publishing, Paris, 2018.

[6] D. Ismael, "Enhancing Online Hands-On Learning in Engineering Education: Student Perceptions and Recommendations". American Society for Engineering Education (ASEE) Annual Conference & Exposition, Baltimore, Maryland, 2023.

[7] G. M. Sato, G. P. C. P. da Luz, L. F. G. Gonzalez and J. F. Borin, "Reaproveitamento de TV Boxes para Aplicação de Contagem de Pessoas na Borda em Cidades Inteligentes" in Anais do 8º Workshop de Computação Urbana (COURB). 2024, pp. 197–209.

[8] J. W. Lee and Y. W. Kim, "A study on smart IoT hub for intelligent signage services using trust information" in *Proc. Int. Conf. Information Networking (ICOIN)*, Chiang Mai, Thailand, 2018.

[9] C. R. M. Silva and F. A. C. M. Silva, "An IoT Gateway for Modbus and MQTT Integration". *SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, Aveiro, Portugal, 2019.

[10] C. R. M. Silva, J. P. N. Santos and F. A. C. M. Silva, "Um Hub e Gateway MQTT de baixo custo para Sensores IoT com interface USB". *In: Anais da XVI Conferência Nacional em Comunicações, Redes e Segurança da Informação (ENCOM). Natal(RN), Brazil, 2024.*

[11] M. Yusufov and I. Kornilov, "Roles of Smart TV in IoT-environments: A survey" in *Proc. 13th Conference of Open Innovations Association FRUCT*, pp. 163–169, 2013.

[12] S. Bagwari et al., "Low-Cost Sensor-Based and LoRaWAN Opportunities for Landslide Monitoring Systems on IoT Platform: A Review" in *Proc. IEEE Access*, 2022.

[13] A. Kluczek and A. Buczacki, "Smart energy sustainability hub in light of Industry 4.0" Energy Reports, vol. 10, Pages 3835-3846, Novr 2023.

[14] S. Sai, M. Kanadia and V. Chamola, "Empowering IoT with Generative AI: Applications, Case Studies and Limitations", *ResearchGate*, 2025.