IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

VR-Driven Digital Twin Automation System

Muhammed Yaseen Sidhik Department of Computer Applications

1MS24MC062 Ramaiah Institute of Technology

Bangalore, India

Dr Manjunath M Department of Computer Applications Ramaiah Institute of Technology

Bangalore, India

Abstract: Conventional Internet of Things (IoT) management systems are inherently limited by flat, two-dimensional interfaces, which fail to provide the spatial context necessary for intuitive facility oversight. Although recent deployments combining web dashboards and edge devices have produced rudimentary smart environments, most of these implementations employ static buttons restricted to unidirectional control. Such architectures perform poorly under the conditions of characteristics of large enterprise facilities, where visual locating and geographical representation of physical endpoints are necessary. This paper presents a fully decoupled Virtual Reality (VR) Digital Twin architecture that synchronizes immersive 3D spatial computing with physical edge devices in real-time. Utilizing the Three.js library and the WebXR API, the system enables users to interact with contextual holographic switchboards via VR headsets within a precise 1:1 graphical replica. Under experimental evaluation, the deployed edge hardware utilizing custom memory buffering to bypass enterprise SSL restrictions achieved an average end-to-end synchronization latency of 850 milliseconds. A desktop spectator client is coupled with this architecture to render geospatial telemetry heatmaps in real-time. Taken together, these components constitute a scalable, practically deployable framework for bridging virtual spatial computing and physical electromechanical actuation.

Index Terms: Digital Twin, WebXR, Internet of Things (IoT), Spatial Computing, Three.js, Edge Computing, Realtime Database.

1.Introduction

The management of physical campus infrastructure and industrial facilities relies heavily on decentralized, manual interventions or abstract two-dimensional dashboards. Traditional IoT management systems disconnect the user from the physical environment by relying on rudimentary toggle switches on mobile applications. These interfaces fail to provide an intuitive, 1:1 geographical representation of the endpoints they control. This conventional methodology frequently eludes the natural human understanding of physical space, leading to operational inefficiencies in large enterprise environments.

Research interest in automating facility management has grown substantially. The Digital Twin architecture has emerged as a critical frontier in modern facility management and spatial computing. A Digital Twin serves as a living, interactive 3D replica of a physical space that directly monitors and controls real-world appliances in real-time.

A fundamental limitation persists across much of this domain: the reliance on heavy, standalone executable software to render 3D environments, and the inability of low-cost microcontrollers to securely handshake with modern cloud infrastructure. This paper introduces a highly distributed, three-tier full-stack application ecosystem to address this limitation. Rather than relying on flat UI panels, the platform renders a highly precise, PBR-textured spatial model directly in the browser. Users interact with contextual holographic switchboards that trigger sub-second physical state changes in the real world.

II. Literature Review

The modernization of IoT management practices can be broadly characterised as a progression from localized physical interventions toward data-driven, cloud-assisted analysis. Existing smart home systems heavily rely on 2D architectures like Blynk, Tuya, or basic MQTT dashboards. While these models yield optimized performance metrics for basic on/off toggles, they lack physical context; users cannot visually locate the hardware within a complex architectural footprint.

The integration of Virtual Reality introduced spatial context as an alternative approach. Studies exploring VR integration heavily rely on standalone Unity applications. While these models yield highly immersive graphics, they require massive localized computing power and heavy application downloads, making them inaccessible for rapid browser-based civic deployment.

Despite these advances in software, real-world applicability remains constrained by hardware-dependent limitations. Frameworks utilizing microcontrollers like the ESP8266 frequently encounter severe operational bottlenecks: specifically, the lack of sufficient RAM to perform Secure Socket Layer (SSL) or Transport Layer Security (TLS) handshakes required by modern enterprise cloud providers like Google or AWS. This often results in kernel panics and continuous reboot loops during deployment.

III. Problem Statement

Several unresolved challenges continue to limit the effectiveness of automated facility management systems. Standard IoT dashboards are spatially blind, creating a disconnect between the user and the physical appliance. Furthermore, establishing a bidirectional bridge between a 3D interface and a physical edge device requires navigating severe cryptographic memory limitations on the microcontroller. Finally, resolving the cognitive dissonance between standard software logic (where 'true' equates to ON) and the physical Active-Low circuitry of electromechanical relays is necessary to prevent data corruption.

This architecture is proposed to address these specific shortcomings. By employing WebXR and Three.js, the system resolves spatial context natively in the browser. Simultaneously, the application provides custom memory buffering and active-low logic inversions at the edge to ensure the physical hardware safely reflects the virtual state.

IV. Proposed Method and Design

The system is implemented as a highly decoupled Three-Tier Client-Server Architecture, cleanly segregating the presentation layers, the application processing logic, and the physical execution.

A. System Architecture and Implementation: The presentation layer (Tier 1) is built with HTML5, CSS3, JavaScript, and Three.js, acting as the cross-platform browser client and 3D rendering engine. The application layer (Tier 2) consists of Google Firebase Realtime Database, which hosts the single source of truth for the physical world. The physical edge layer (Tier 3) utilizes an ESP8266 NodeMCU microcontroller programmed in C++ to execute physical commands via 5V electromechanical relays.

B. Hardware and Model Configuration: The primary infrastructural constraint lies within the 50kB RAM limitation of the ESP8266. To prevent memory starvation during Google's TLS 1.2+ encryption handshakes, the firmware is explicitly configured to utilize custom BearSSL (BSSL) memory buffer overrides. The Receive (Rx) buffer is manually allocated to 4096 bytes, and the Transmit (Tx) buffer is clamped to 1024 bytes.

To control the 220V AC load, the firmware implements an Active-Low circuitry paradigm. The C++ device intercepts pure Boolean logic from the cloud and routes it through a ternary inversion operator prior to executing the digitalWrite () command (state? LOW: HIGH). Furthermore, the system employs strict loop isolation, enforcing a 1Hz polling rate to fetch endpoints sequentially, preventing Wi-Fi radio stack collisions.

C. Application Workflow: Upon a VR user intersecting a holographic switch via a Ray caster in the WebXR client, the application encodes a JSON PATCH request to Firebase. Firebase updates the internal NoSQL tree and broadcasts WebSockets to all connected clients. The ESP8266 executes its 1Hz GET request, receives the new Boolean value, and triggers the physical GPIO relay state change. Concurrently, the VR headset translates its spatial origin and pushes live X and Z coordinates to the cloud, allowing a desktop spectator client to render a moving radar marker.

V. Experimental Results and Analysis

System performance was evaluated across end-to-end operational workflows to ensure architectural stability.

A. Quantitative Results: To validate the sub-secondsynchronization requirement, a Round-Trip Time (RTT) latency benchmark was established. Across 50 iterative tests conducted over a standard 2.4GHz broadband network, the system demonstrated an average end-to-end latency of 850 milliseconds between virtual interaction and physical actuation.

IoT microcontrollers were also subjected to a continuous 24-hour polling cycle to profile memory leaks. Utilizing the ESP.getFreeHeap() command, the architecture successfully maintained a stable heap baseline of ~18kB free RAM, confirming that the BSSL buffer overrides effectively flushed memory after every cycle.

B. Qualitative Results: Visual inspection of the WebXR client confirms that the system produces highly accurate 1:1 spatial telemetry. As the VR operator translates physically, the desktop radar marker mirrors the movement seamlessly. During edge-case network degradation testing, the microcontroller exhibited flawless recovery behavior. When the local router was disconnected, the firmware gracefully paused actuation and autonomously re-established the secure SSL handshake upon network restoration within 8 seconds.

VI. Conclusion and Future Work

This paper has described a VR-Driven Digital Twin Automation System that addresses the spatial context limitations of prior IoT deployments. Existing smart-home implementations share a common limitation: two-dimensional interfaces cannot intuitively represent physical environments. By adopting WebXR and Three.js, this architecture produces an immersive 3D replica capable of sub-second state synchronization via decoupled cloud middleware. The system connects high-fidelity graphics to practical edge hardware by successfully overcoming cryptographic memory limitations and active-low hardware inversions.

Several directions for future development have been identified. First, migrating from localized Wi-Fi to Long Range Wide Area Networks (LoRaWAN) would allow the edge hardware to be deployed across massive industrial environments without dependency on local routers. Second, integrating advanced HVAC protocols like BACnet or Modbus would allow the system to dynamically modulate building temperatures based on spatial zone interactions. Third, integrating Generative AI directly into the WebXR interface could provide operators with conversational spatial analytics.

References

[1] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," in Transdisciplinary Perspectives on Complex Systems, Springer, 2017.

[2] P. J. Radita et al., "Integration of WebXR and IoT for Smart Home Applications," IEEE Internet of Things Journal, vol. 8, no. 12, 2021.

[3] Espressif Systems, "ESP8266EX Hardware Design Guidelines," Espressif Technical Documentation, 2020.

[4] Google, "Firebase Realtime Database REST API," Firebase Documentation, 2026.

[5] K. Matsuda, WebGL Programming Guide, Addison-Wesley Professional, 2013.

[6] "Three.js Documentation," Three.js, 2026. [Online]. Available: https://threejs.org/docs/.