



Low-cost Smart Home Design and Evaluation for Internet of Things and Supervisory Control Education

MATTHEW GRAFF¹, SHENG-JEN HSIEH^{2,3*}, WILLIAM KERNEY¹,
CHIA-YING KUO³

¹*Department of Business Industry and Information Technology, Clovis Community College, Fresno, CA 93730, USA*

²*Department of Engineering Technology & Industrial Distribution, Texas A&M University, College Station, TX 77843, USA*

³*Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX 77843, USA*

**hsieh@tamu.edu*

Abstract: The rapid evolution of Industry 4.0 and the Industrial Internet of Things (IIoT) has created a significant demand for engineers and technicians skilled in embedded networking, system integration, and supervisory control. However, academic institutions often struggle to provide hands-on training in these areas due to the prohibitive cost and proprietary nature of commercial industrial training rigs. This paper presents the design and evaluation of a low-cost educational platform that utilizes a smart home context to demonstrate fundamental industrial SCADA (Supervisory Control and Data Acquisition) concepts. By leveraging commercial off-the-shelf (COTS) components—including ESP32 microcontrollers and Raspberry Pi servers arranged in an MQTT-centric architecture—the proposed framework bridges the gap between simple sensor-to-cloud prototyping and complex industrial automation. The study details the system design, which supports essential industrial functions such as alarm management, set-point control, and historical data trending via a standard publish-subscribe protocol. It embeds this platform in a structured laboratory sequence. Financial analysis demonstrates that the platform can be constructed for approximately \$185 per station, representing a cost reduction of nearly 80% compared to entry-level commercial trainers. Entry-exit survey data from students indicate measurable gains in SCADA/IIoT knowledge, tool familiarity, and career interest, suggesting that this architecture offers a technically feasible, scalable, and pedagogically effective solution for engineering and technician programs seeking to deploy rigorous IIoT laboratory experiences to larger student cohorts without relying on restrictive vendor licenses.

Keywords: Industry 4.0 education, industrial internet of things (IIoT), SCADA systems, engineering laboratory design, low-cost educational platforms, smart home automation, MQTT protocol

© 2026 under the terms of the J ATE Open Access Publishing Agreement

Introduction

The rapid growth of the Internet of Things (IoT), Industry 4.0, and cyber-physical systems (CPS) [1] has created strong demand for engineers and technicians who can design, deploy, and maintain interconnected sensing, actuation, and supervisory control infrastructures [2-4]. Recent studies on Industry 4.0 integration in engineering curricula and continuing education repeatedly highlight a widening gap between the skills required by industry—such as embedded networking, edge computing, data acquisition, and system integration—and the background of many graduating engineers and technicians [2-3, 5-8]. In particular, IoT engineers and technicians are expected to combine competencies in embedded hardware, communication protocols,



cybersecurity, and data analytics, yet many programs still emphasize theory or isolated course projects rather than end-to-end, system-level practice [3, 6, 9-10].

Supervisory control and data acquisition (SCADA) systems and Human–Machine Interfaces (HMIs) occupy a central role in industrial automation, energy systems, and infrastructure management [11-12]. Traditional SCADA platforms from major vendors (e.g., Siemens WinCC, GE iFIX/CIMPLICITY, Inductive Automation Ignition) offer powerful capabilities for real-time monitoring, alarming, and control. While powerful, these platforms often present financial and licensing barriers that limit their deployment in large undergraduate cohorts [13]. These systems are widely used in large-scale industrial settings and in specialized training programs. Still, their cost and complexity make them difficult to adopt in resource-constrained teaching laboratories, especially at institutions that wish to support open-ended student projects or to provide each student team with its own physical setup [11, 14-15].

In many engineering and technician programs, the resulting educational gap is twofold. First, IoT is often introduced through low-stakes, single-board prototyping exercises (e.g., simple sensor-to-cloud demos), which may not expose students to supervisory control concepts such as alarm management, set-point scheduling, or coordinated multi-device operation [9, 16-17]. Second, SCADA and HMI topics, when they appear at all, are frequently taught using vendor-specific simulators or shared industrial trainers that limit student access and hands-on time [11, 12, 14]. There is therefore a need for platforms that are simultaneously low-cost, replicable, and rich enough to illustrate the full loop from physical sensing and actuation to networked communication and supervisory decision-making [11, 14, 17].

A smart home is an ideal context for such a platform. It is familiar and immediately understandable to students, yet it naturally involves multiple IoT subsystems (lighting, HVAC, security, environmental monitoring) and lends itself to supervisory control scenarios (e.g., scheduling, energy management, safety interlocks). Recent educational and commercial smart home kits demonstrate that a home-scale environment can be used to explore IoT and automation concepts in an accessible way [15, 17]. By designing a smart home model that uses the same architectural concepts as industrial SCADA systems—but with low-cost, commercial off-the-shelf (COTS) components—it becomes possible to demystify industrial control technologies while keeping the platform accessible for classroom and project-based use [14-15, 18].

Given these trends and constraints, this work addresses the following central challenge: How can we design and rigorously evaluate a low-cost, yet functional smart home platform that can effectively support both fundamental IoT education (sensing, networking, device programming) and advanced supervisory control education (SCADA/HMI concepts such as alarm handling, set-point management, and multi-device coordination)? The problem is non-trivial because the platform must simultaneously satisfy competing requirements: (i) hardware and software must be inexpensive and readily available; (ii) the system architecture must be realistic enough to resemble modern IoT-enabled SCADA systems; and (iii) the resulting platform must be pedagogically effective in helping students learn and apply core concepts across multiple courses and disciplines [2, 11, 14].

To address this challenge, the paper pursues four specific objectives. First, we design and build a functional tabletop smart home model using COTS and low-cost components, focusing on microcontroller platforms such as ESP32 and single-board computers such as Raspberry Pi that are widely available and well supported by the education and maker communities. Second, we integrate these devices into a networked architecture that exposes telemetry and control points to a web-based HMI—implemented on a Raspberry Pi–hosted SCADA gateway—so that students can interact with the system in a manner analogous to industrial SCADA consoles. Third, we document and evaluate the total Bill of Materials (BOM) to demonstrate that the platform can be replicated at relatively low cost, making it feasible for multiple teams or institutions to deploy. Finally, we empirically assess the platform’s technical and educational value, evaluating its cost-effectiveness relative to commercial industrial trainers, its usability and scalability for lab deployment, and its preliminary educational impact using entry–exit survey data on student knowledge, confidence, and career interest.



This paper is organized as follows: The remainder of this Introduction section surveys existing IoT educational platforms and industrial trainers, discusses the principles of SCADA/HMI in education, and motivates the use of low-cost, open-source technologies such as MQTT and Raspberry Pi. The Methods section then describes the proposed smart home platform in detail, including the overall system architecture, the hardware components and Bill of Materials, the software implementation of the IoT devices, and the design of the laboratory sequence. The Results and Discussion section presents the prototype construction, functional and supervisory control tests, and an evaluation of the platform's cost-effectiveness, usability, scalability, and preliminary educational efficacy based on student survey data. Finally, the Conclusion summarizes the main findings, highlights current limitations, and outlines directions for future improvement and extension of the platform.

State-of-the-Art in IoT Educational Platforms

IoT educational platforms have evolved substantially over the past decade, spanning low-cost maker-oriented kits to industrial-grade training systems [9, 16-17]. At the entry level, Arduino-based platforms—such as the Arduino Explore IoT Kit, Arduino PLC Starter Kit, and Oplà IoT Kit—are widely adopted due to their affordability, ease of use, and extensive community ecosystem [9, 16]. These kits typically bundle a microcontroller board with sensors, actuators, and cloud-connectivity tools, enabling students to build and program IoT devices for realistic applications, including smart homes, environmental monitoring, and basic industrial automation [6, 9, 16-17, 19]. Because these platforms are supported by rich online documentation and open-source libraries, they are especially attractive for introductory courses and project-based learning in IoT and embedded systems [6, 9, 16].

Complementing these low-cost kits, industrial trainers aim to replicate real-world automation environments more closely. These systems often integrate programmable logic controllers (PLCs), industrial HMI panels, and field-grade sensors and actuators, along with support for standard industrial communication protocols such as Modbus and OPC UA [11, 12, 14]. In doing so, they provide students with highly authentic hands-on experience that aligns closely with industrial practice [11, 14]. However, this realism comes at a cost: industrial trainers generally require proprietary software licenses, ongoing maintenance, and specialized technical support, and individual training stations can cost from one to several thousand U.S. dollars [13-14]. This price point can be prohibitive for many institutions, especially those wishing to equip multiple parallel lab stations or support open-ended student projects [14].

In addition to cost, complexity and scalability also differentiate these platforms. Arduino-based and similar maker kits are intentionally designed for beginners, with step-by-step project guides and a gentle learning curve that emphasizes basic wiring, programming, and cloud integration [6, 9, 16]. They can be readily scaled from simple sensor nodes to more complex multi-device systems, depending on course goals [6, 9, 17]. Industrial trainers, while powerful, often assume that learners have some prior exposure to automation concepts and can navigate vendor-specific configuration tools, wiring practices, and diagnostic procedures [11, 12, 14]. For novice students, this complexity can become a barrier rather than an enabler [11, 14]. Moreover, industrial trainers are typically configured for a limited set of scenarios (e.g., a fixed process or production line), which can make them less flexible for instructors who wish to rapidly prototype new experiments or adapt the lab to emerging topics [14].

Taken together, current IoT educational platforms illustrate a clear trade-off: low-cost, Arduino-style kits provide accessibility and scalability but tend to focus on device-level IoT and simple cloud connectivity, while industrial trainers offer realistic exposure to automation and SCADA-like environments at the expense of cost, flexibility, and accessibility [6, 9, 14]. This trade-off motivates the need for intermediate platforms that are both affordable and capable of supporting richer system-level concepts such as supervisory control, alarm management, and coordinated operation of multiple devices [14, 20]. A smart home-scale platform built from low-cost components and designed explicitly for educational use can help bridge this gap [15, 17, 18].

Principles of Supervisory Control and SCADA in Education



Supervisory Control and Data Acquisition (SCADA) systems form the backbone of modern industrial automation in sectors such as manufacturing, power systems, water treatment, and transportation [11, 18, 20]. Conceptually, a SCADA system integrates three key layers: field devices (PLCs, RTUs, sensors, and actuators) that interact with the physical process; communication infrastructure that transmits measurements and commands; and supervisory software that aggregates data, displays it through a Human–Machine Interface (HMI), and executes higher-level control logic. Through this architecture, operators can monitor real-time process variables, acknowledge alarms, adjust setpoints, and coordinate large-scale operations from a centralized control room or distributed interfaces [11, 12, 16].

Within this architecture, core SCADA/HMI concepts have strong educational value. Supervisory control focuses on decision-making at a higher level than local PID loops or embedded control, emphasizing topics such as alarm philosophies, interlocks, scheduling, and manual overrides. The HMI provides the graphical layer through which these supervisory actions occur: it visualizes process variables, trends, alarms, and states using mimic diagrams, plots, and interactive widgets [11, 12, 16]. Data acquisition and historian functions complete the loop by capturing measurements from remote devices, storing them in databases, and enabling subsequent analysis and reporting [11, 14, 18]. Together, these elements give students a system-level view that combines control theory, networking, user interface design, and data analytics [11, 14].

However, this realism comes with significant barriers. Commercial industrial trainers are often prohibitively expensive for general engineering programs and rely on restrictive licensing models that prevent students from accessing the software outside the laboratory [13, 14, 16]. This economic friction forces many institutions to choose between high-cost, limited-access vendor trainers or low-fidelity simulation software [11, 15, 18].

Applying SCADA and supervisory control concepts to a familiar, human-scale context—such as a smart home—offers a promising alternative. Home automation scenarios (e.g., monitoring and controlling lights, thermostats, and security systems) naturally map onto SCADA notions of tags, alarms, trends, and setpoints, but in a setting that students can easily relate to their daily lives [15, 17, 18]. Building a “home SCADA” system using low-cost hardware allows students to experience the full data acquisition–to–HMI pipeline: they configure sensing and actuation devices, define alarm conditions, design dashboards, and implement simple supervisory logic (for example, scheduling lighting scenes or implementing energy-saving modes) [14, 15, 18]. This context makes abstract SCADA concepts tangible and engaging, while still providing a strong conceptual bridge to larger industrial systems [11, 14, 18].

Low-Cost Prototyping and Open-Source Hardware

Low-cost prototyping platforms and open-source ecosystems have transformed how educators design laboratory experiences in IoT and automation. Single-board computers such as the Raspberry Pi provide an inexpensive yet capable computing platform that can host databases, web servers, SCADA-like applications, and communication brokers [9, 14, 16]. At the same time, microcontrollers such as the ESP32 combine digital and analog I/O with integrated Wi-Fi and Bluetooth, making them ideal as field-level IoT or RTU nodes in small supervisory systems [9, 17]. Both platforms support widely used programming environments (e.g., C/C++, MicroPython, and Python) and are backed by large global communities that produce tutorials, example code, and troubleshooting resources [6, 9, 16-17, 21].

On the software side, flow-based tools such as Node-RED have become central to rapid IoT prototyping. Node-RED enables users to build automation workflows by visually connecting nodes representing devices, services, and data-processing blocks [13, 20]. This paradigm lowers the barrier to entry for students, who can focus on system logic rather than boilerplate integration code [20]. MQTT, a lightweight publish–subscribe messaging protocol designed for constrained devices and unreliable networks, complements these tools by enabling efficient, decoupled communication between sensors, actuators, gateways, and cloud



services [13, 20]. Together, Node-RED and MQTT support real-time data acquisition, event-driven control, and flexible integration with web services and databases [13-14, 20-21].

Open-source HMI and dashboard frameworks further extend this ecosystem. Node-RED Dashboard, Grafana, and custom HTML/JavaScript front-ends running on Raspberry Pi can all be used to build SCADA-like interfaces that visualize real-time data and provide simple control widgets, without requiring proprietary licenses [13-14, 18, 20]. These tools are highly configurable: instructors and students can design multi-page dashboards, implement authentication, and integrate historical trends or analytics, all on commodity hardware [13-14, 18]. Because the entire stack—from microcontroller firmware to HMI—can be developed with free or open-source software, institutions can replicate and adapt the platform with minimal financial overhead [14, 18].

These technologies collectively justify their selection for a low-cost smart home SCADA educational platform. From a cost perspective, Raspberry Pi and ESP32 boards, paired with commodity sensors and relays, are significantly cheaper than industrial PLCs and HMI panels, while Node-RED, MQTT, and open-source HMIs are free to use [6, 13-14, 16-18]. Strong community support and extensive online resources reduce the learning curve for both instructors and students, helping them troubleshoot and extend the system. The ease of programming, particularly with Python and visual flow-based tools, enables students with limited programming backgrounds to participate meaningfully in lab activities. Finally, the flexibility and extensibility of this stack make it suitable not only for basic smart home projects but also for more advanced experiments in topics such as cybersecurity, data analytics, or integration with cloud platforms [2-4, 14, 18].

In summary, the existing literature and technology landscape indicate that there is a compelling opportunity to combine low-cost, open-source IoT tools with SCADA/HMI concepts in a familiar smart home setting. Doing so can address the cost and accessibility limitations of traditional industrial trainers while providing richer system-level learning outcomes than most introductory IoT kits alone.

There has been some related work in the area of open source platforms for SCADA education. For example, the CREATE initiative has developed a SCADA platform that allows colleges to connect their renewable energy-generating systems [22]. However, the focus of CREATE's initiative is on fostering student engagement by allowing them to generate, visualize, and analyze long-term, large data sets. In addition, CREATE uses industrial automation hardware (e.g., Allen-Bradley PLCs) and the IGSS SCADA platform from Schneider Electric, which are relatively expensive. The proposed project focuses specifically on designing, implementing, and evaluating a low-cost smart home platform tailored for IoT and supervisory control education. The platform uses relatively low-cost hardware, in combination with Inductive Automation's Ignition software, which is widely recommended and used by local industry partners. Inductive Automation offers educational institutions free access to the full Ignition SCADA platform.

Methods

System Architecture

The smart home training platform is implemented as a three-tier IIoT/SCADA architecture, as illustrated in Figure 1. At the field layer, each smart home station is built around a KEYESTUDIO ESP32 Smart House kit connected to a Wi-Fi network. The ESP32 acts as an IoT endpoint that interfaces with multiple onboard sensors and actuators—such as environmental sensors and indicator lights—and exposes them as MQTT topics. In this layer, the ESP32 microcontroller serves as a low-cost remote terminal unit (RTU): it periodically publishes sensor telemetry to the broker and subscribes to command topics for actuator updates.

The communication and middleware layer is hosted on a Raspberry Pi 4 or Pi 400, and is connected to a compact lab router. The router provides an isolated local network for the ESP32 clients and the Raspberry Pi, while the Pi runs an MQTT broker as the central messaging hub. All field devices connect to the broker over Wi-Fi and exchange messages via a topic hierarchy that distinguishes between houses, device types, and signal



directions (telemetry vs. commands). This decoupled publish–subscribe pattern allows the platform to scale from a single smart home to a “smart neighborhood” with multiple ESP32 houses sharing the same broker and supervisory system.

At the supervisory layer, the Raspberry Pi also hosts an Ignition gateway from Inductive Automation, which functions as the SCADA/HMI server. The system uses Ignition’s MQTT Engine to subscribe to the topics provided by the ESP32 devices and map them into tags that can be visualized and controlled within Ignition views. Students and instructors access these views through a web browser on a PC or, optionally, via a Raspberry Pi touchscreen configured as a local HMI panel. From these HMIs, users can monitor real-time sensor values, send manual control commands (e.g., toggle smart home lights), and, where configured, view trends or status indicators. This three-tier structure—field devices, MQTT middleware, and a SCADA/HMI layer—mirrors modern industrial IIoT deployments while remaining compact and affordable for instructional use.

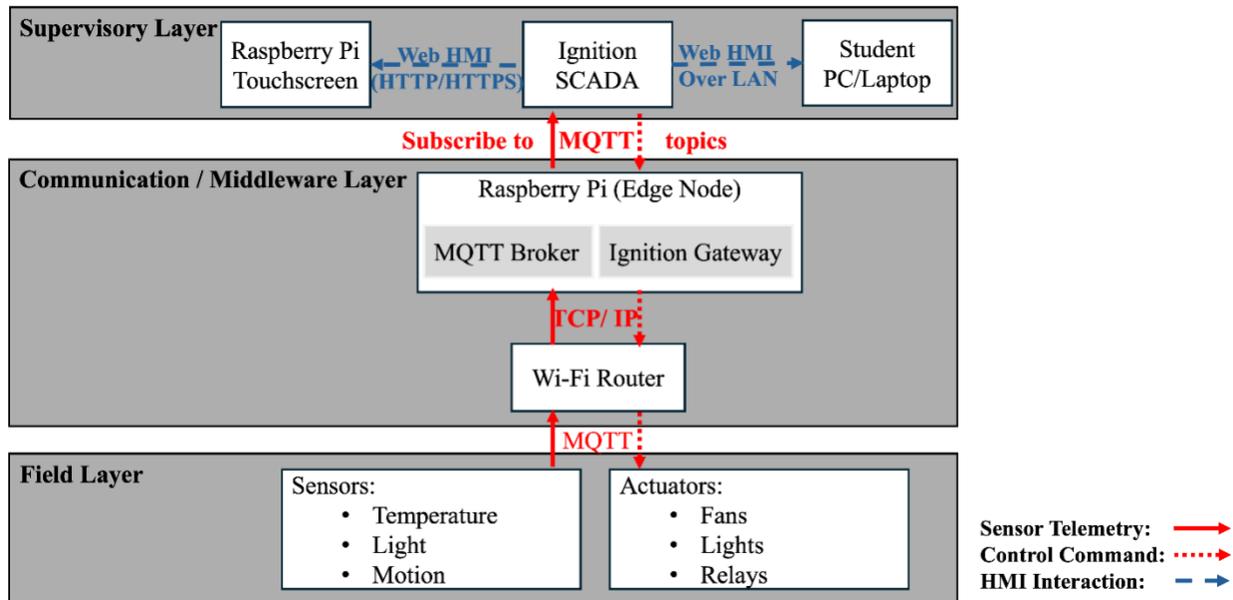


Fig. 1. Three-tier architecture of the low-cost smart home IIoT/SCADA platform. ESP32-based smart home kits act as field devices, publishing sensor data and receiving actuator commands via MQTT over Wi-Fi. A Raspberry Pi hosts both the MQTT broker and the Ignition SCADA gateway, which exposes tags to web-based HMI clients (Raspberry Pi touchscreen and student PCs) over the local lab network.

Hardware Components and Bill of Materials (BOM)

The hardware design emphasizes the use of commercial off-the-shelf components that are both readily available and inexpensive, consistent with the project’s stated goal of introducing Industry 4.0 concepts at minimal cost. The reference kit includes:

Raspberry Pi 4 or Pi 400 as the edge computing node and host for the MQTT broker and Ignition gateway. KEYESTUDIO ESP32 Smart House kit as the primary smart home field device, bundling the ESP32 microcontroller with a variety of sensors, actuators, and a compact training board.

Compact Wi-Fi router (e.g., GL.iNet GL-AR300M16-Ext) to provide an isolated lab network and DHCP/Wi-Fi connectivity for the ESP32 and Raspberry Pi.



Optional enhancements include a 7" Raspberry Pi touchscreen display and SmartiPi Touch 2 case for a kiosk-style local HMI, as well as an Opto 22 GRV-RIO-LC module for advanced industrial I/O demonstrations, which is intended for more advanced college-level training rather than the baseline low-cost configuration.

For the core smart home station described in this work, the baseline BOM therefore consists of the Raspberry Pi, ESP32 smart house kit, and router, yielding an approximate per-station hardware cost on the order of \$185 before tax and shipping. Basic accessories such as microSD cards, power supplies, and cables are either bundled with these devices or available at a marginal additional cost.

Software Design and Implementation

The IIOT-4.0-Project repository mentioned in this section is available online at: <https://github.com/CCC-Industry4/IIOT-4.0-Project?tab=readme-ov-file#readme>.

1) IoT Device Code

At the field layer, the ESP32 microcontroller on the smart house board is programmed using the Arduino framework. The Arduino directory in the IIOT-4.0-Project repository contains sketches that configure the ESP32 as an MQTT client, initialize the smart house I/O pins, and implement periodic acquisition of sensor values along with subscription to actuator command topics. Each sketch connects to the lab Wi-Fi network, establishes an MQTT session with the broker running on the Raspberry Pi, and publishes measurements (e.g., sensor readings) at defined intervals using a structured topic naming scheme. Similarly, it subscribes to one or more command topics and updates outputs such as LEDs or relays based on incoming messages. This design allows students to inspect and modify the device behavior at the code level, reinforcing their understanding of both embedded programming and message-oriented communication.

2) Control Logic on the Central Node

While flow-based tools like Node-RED are excellent for rapid prototyping and are widely used in the maker community, this proposed platform specifically incorporates Inductive Automation's Ignition for the supervisory layer to expose students to an industry-standard SCADA environment. By using commercial-grade software that is prevalent in manufacturing sectors, the platform ensures that learners not only understand the underlying MQTT protocols but also gain marketable proficiency with the actual tools they will encounter in the workforce.

To implement this, the Raspberry Pi acts as the central IIoT/SCADA node. After installation via the provided `install.sh` script, the Pi runs both the MQTT broker and the Ignition gateway. MQTT Engine within Ignition is configured to subscribe to the relevant topic space and automatically generate tags for each exposed measurement and command point, effectively binding the ESP32 smart home signals to the SCADA environment. Control logic can then be implemented at the supervisory level using Ignition's expression bindings, tag change scripts, or simple alarm and event configuration. For the core smart home use case, this typically consists of:

- Direct bindings between HMI control widgets (switches, buttons) and command tags that drive actuators on the ESP32.
- Derived tags or scripts that implement simple rule-based logic, such as turning a device on when a sensor crosses a threshold or grouping multiple loads into "scenes."

While the repository's public materials primarily provide a working configuration for monitoring and basic control, the same mechanism readily supports the implementation of more advanced logic—such as thermostat behavior or automatic occupancy-based lighting—as part of future curriculum extensions.

3) Supervisory Control and HMI Design

The HMI is implemented using Ignition's web-based visualization tools. The Ignition directory in the repository includes project resources such as gateway backups and view definitions that illustrate a smart home-style dashboard. In a typical configuration, the main view presents an overview of the house with



clickable controls for lights and other devices, real-time numerical or gauge displays of sensor values, and optional status indicators. Students access this interface from a browser on a lab PC or from an attached Raspberry Pi touchscreen, interacting with the system as a control room operator would in an industrial setting. The resulting operator interface, shown in Figure 2, functions as a centralized supervisory dashboard. It aggregates real-time telemetry from the distributed ESP32 nodes, displaying current environmental metrics such as temperature alongside device status indicators. The interface allows users to perform direct supervisory actions—such as toggling the lighting subsystem or adjusting the temperature setpoint via a slider—which are immediately published back to the edge devices via MQTT.

Because the HMI is served via a web gateway, additional clients can be added with minimal effort, and views can be extended to include trend charts, alarm banners, or multiple smart homes (for the “neighborhood” scenario). The project thus exposes learners not only to the idea of supervisory control but also to practical considerations in HMI design: grouping related controls, providing feedback, and ensuring that the interface remains usable even as the number of devices grows.

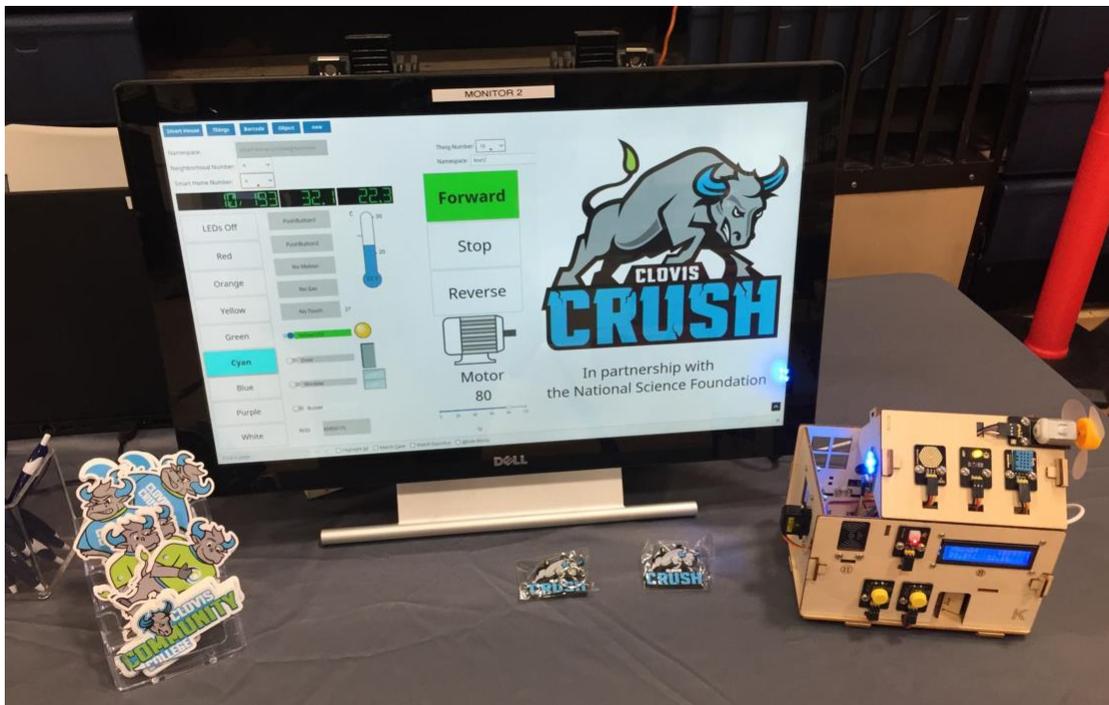


Fig. 2. The web-based supervisory interface developed using the Ignition Perspective module. The dashboard integrates real-time data monitoring (current temperature), binary control (light/fan/motor), and analog parameter adjustment (temperature setpoint), providing students with a functional example of an industrial HMI adapted for a smart home context.

Experiments

In this section, we describe how the smart home platform was instantiated as a sequence of laboratory activities and how these activities were structured to progressively develop students’ understanding of IoT networking, MQTT-based communication, and supervisory SCADA concepts. Rather than serving only as a hardware demonstration, the prototype was deliberately embedded in a lab curriculum that moves from basic system bring-up to integrated supervisory operation.

Prototype Construction



The physical prototype follows the hardware and configuration blueprint defined in the IIOT-4.0-Project, but its role in the course is more than simply “wiring the system together.” The construction phase is designed as the students’ first encounter with the notion of a distributed IIoT architecture: a router providing an isolated lab network, a Raspberry Pi acting as the edge node, and an ESP32-based smart home kit serving as the field device.

In the corresponding lab activity, students first initialize the router and configure the Raspberry Pi with the provided installation script, which automatically deploys the MQTT broker and the Ignition gateway. They then power and connect the KEYESTUDIO ESP32 Smart House kit to the same Wi-Fi network and verify basic connectivity. At this stage, the emphasis is not on low-level electronics assembly—the smart house board encapsulates most sensors and actuators—but on understanding how each physical element maps to its logical role in the three-tier architecture (field, communication, supervisory). By the end of the lab, students can visually identify each tier on the bench and explain how data and control signals are expected to flow between them.

Functional Testing

Functional testing is organized as a second layer of learning: once students understand *what* pieces exist, they validate *how* those pieces interact. The first functional lab focuses on establishing end-to-end connectivity. Students confirm that the Raspberry Pi is reachable over the network, that the MQTT broker is running, and that the ESP32 successfully publishes and subscribes to topics. This makes the abstract MQTT publish/subscribe model concrete: if the broker is not reachable, nothing else in the system will behave as expected.

A follow-up lab then shifts attention to tag mapping and HMI binding. Students import the Ignition project configuration, discover MQTT topics as tags, and bind those tags to widgets on the smart home dashboard. Functional correctness is evaluated through observable behavior: toggling a switch on the HMI must immediately change the state of the corresponding light or relay on the smart house board, and manipulating a sensor (e.g., covering a light sensor or changing a knob) must produce responsive updates on the dashboard. By guiding students to reason explicitly about the two main paths—sensor data path (ESP32 → MQTT → Ignition → HMI) and control path (HMI → Ignition → MQTT → ESP32)—this lab operationalizes core IoT/SCADA concepts in a hands-on manner. Formal performance metrics (e.g., latency, message loss) are not the focus here; instead, the primary learning outcome is that students can *trace and debug* end-to-end behavior across all three tiers.

Supervisory Control Scenario Testing

The final set of experiments uses the same platform to introduce supervisory control behavior rather than only point-to-point control. In the “smart home neighborhood” lab, students operate the system from the Ignition HMI as if they were control room operators. They execute manual control scenarios—such as applying different lighting “scenes,” simulating occupancy patterns, or coordinating multiple devices—and observe how their commands propagate through MQTT to the ESP32 devices and back as feedback on the HMI. This reinforces the idea that supervisory systems orchestrate *sets* of field devices according to higher-level goals.

Although the publicly distributed configuration focuses primarily on manual operation, the architecture is explicitly used to motivate basic automatic control and SCADA features. Instructors can, for example, ask students to sketch and then implement a simple thermostat policy (“turn the fan on when temperature exceeds a threshold and off when it falls below it”) using Ignition tag scripts or expression tags. Similarly, alarm management and data trending are introduced conceptually by showing how alarm conditions and tag history can be enabled on existing smart home tags to generate high-temperature alarms or historical plots of sensor values. In this way, the same physical platform supports a progression from device-level control to supervisory logic, giving students a coherent, hands-on experience of how industrial SCADA concepts manifest in a familiar smart home context.



Results and Discussion

Cost-Effectiveness Evaluation

To evaluate the cost-effectiveness of the proposed smart home SCADA platform, we first estimate the Bill of Materials (BOM) for a single lab station, using typical current retail prices in the U.S. market. A representative configuration consists of:

- Raspberry Pi 4 (4 GB RAM) acting as the local SCADA/HMI server and MQTT broker host (~\$60 for the board-only version).
- KEYESTUDIO IoT ESP32 Smart Home Kit, which integrates the ESP32 controller with a smart house board containing LEDs, relays, sensors, and user-input devices (~\$57).
- Mini Wi-Fi router (e.g., GL.iNet GL-AR300M series) to create an isolated lab network and provide Wi-Fi connectivity for the ESP32 (~\$30–\$35).
- Miscellaneous items, including a microSD card for the Raspberry Pi, power supplies, Ethernet cable, and basic accessories (on the order of \$20–\$30).

Using these representative figures, the total hardware cost per station ranges from \$170 to \$190, depending on local pricing and whether institutional discounts or kits are used. This cost is low enough that multiple stations can be deployed within a modest laboratory budget, yet each station still includes a dedicated embedded SCADA server (Raspberry Pi) and a complete, sensor/actuator-rich smart home environment.

From a comparative standpoint, this places the platform between entry-level IoT kits and industrial automation trainers:

- Commercial educational smart-home and IoT kits such as the ACEBOTT QE023 ESP32 Smart Home Education Kit are typically priced around \$100–\$110 per kit.
- Arduino Explore IoT Kit Rev2 and similar university-oriented IoT kits usually retail in the \$135–\$170 range, depending on vendor and bundle contents. These products provide robust IoT learning experiences, but most focus on microcontroller-to-cloud patterns and do not include a dedicated on-premises SCADA/HMI server per station.

On the other end of the spectrum, standalone HMI trainers from vendors like Siemens or Allen-Bradley commonly fall in the \$600–\$900 range. Complete PLC/SCADA training systems are even more expensive, often advertised at \$1,000+ per station, with advanced kits reaching the \$2,000–\$7,000 price band. These platforms provide highly realistic industrial hardware, but their cost makes it difficult to provide one station per small student group.

Given this landscape, the proposed platform achieves a favorable cost–capability balance:

- At roughly \$180 per station, the cost is comparable to, or only moderately higher than, a single advanced IoT kit, yet the station includes:
 - A full Linux-based SCADA/HMI server (Raspberry Pi),
 - A realistic MQTT-based communication layer,
 - A multi-sensor, multi-actuator smart home board for physical interaction.
- Compared to industrial SCADA/PLC trainers, the cost is typically one order of magnitude lower, while still enabling students to experience key SCADA concepts (tags, alarms, trends, operator screens) and field communication through MQTT.

The cost per student depends on how many students share a station. For typical group sizes of 3–4 students per station, the amortized hardware cost is on the order of \$45–\$60 per student (ignoring the fact that the hardware can be reused across multiple semesters). If a lab deploys, for example, six stations, the total hardware outlay is roughly \$1,000–\$1,200, which is comparable to the price of a single industrial trainer but now supports



multiple teams working in parallel. Over several cohorts, this significantly improves hardware access per student relative to a single shared industrial training rig.

Overall, the cost analysis suggests that the platform:

1. Meets the “low-cost” objective in absolute terms (sub-\$200 per complete SCADA-capable station).
2. Delivers a richer system-level learning environment per dollar than entry-level IoT kits, due to the inclusion of a dedicated SCADA/HMI server and structured labs.
3. Remains far cheaper than full industrial PLC/SCADA trainers, making it feasible for widespread deployment in teaching labs, outreach workshops, and even take-home projects in advanced courses.

Educational Efficacy Assessment

To obtain an initial measure of the platform’s educational impact, we administered an entry and exit survey to 9 college students who used the smart home IIoT/SCADA station as part of the lab sequence. The survey asked students to self-rate their knowledge or familiarity with key technical topics (Ignition, MQTT, Raspberry Pi/ESP32/Arduino, sensors and actuators, smart homes) as well as their confidence in applying these concepts (e.g., configuring MQTT, connecting IIoT components, troubleshooting) and their interest in Industry 4.0–related careers. All items were rated on a Likert-type scale, with higher values indicating greater knowledge, familiarity, or confidence. The self-efficacy items were patterned after Mamaril et al.’s Engineering Skills Self-Efficacy Scale [23], but contextualized to the topics of interest. Table 1 summarizes the average entry and exit ratings and the corresponding gains (exit minus entry) for each survey category.

Table 1. Pre/post survey results for smart home IIoT/SCADA lab

Category	Entry Avg	Exit Avg	Gain (Exit - Entry)
Ignition	1.89	2.78	+0.89
MQTT	1.44	2.22	+0.78
Pi / ESP / Arduino	2.00	2.33	+0.33
Sensors & Actuators	3.00	3.44	+0.44
Smart Home	1.44	2.56	+1.11
MQTT Confidence	2.78	1.89	-0.89
Ignition Confidence	1.56	3.11	+1.56
IIoT Connection Confidence	2.67	2.44	-0.22
Troubleshooting Confidence	2.22	2.78	+0.56
Career Interest	3.22	3.67	+0.44

Overall, the results indicate “measurable gains in technical knowledge and system familiarity”. Students reported substantial increases in their understanding of Ignition as a SCADA/HMI platform (+0.89) and of MQTT as the underlying messaging protocol (+0.78). Their conceptual understanding of the “smart home” as an IIoT system increased even more (+1.11), suggesting that the physical smart home context was effective in making abstract IIoT and SCADA ideas concrete.

More modest but positive gains were observed in familiarity with embedded hardware (Pi / ESP / Arduino, +0.33) and with sensors and actuators (+0.44). These smaller gains are consistent with the fact that many students had some prior exposure to microcontrollers and basic electronics before the lab sequence, while SCADA/HMI and MQTT were largely new to them.



In contrast, the “confidence measures show a more nuanced pattern”. Students’ confidence with Ignition increased strongly (+1.56), as did their troubleshooting confidence (+0.56), reflecting the hands-on time spent configuring tags, dashboards, and resolving connectivity issues. However, reported MQTT confidence decreased (−0.89), and confidence in connecting IIoT components showed a slight decline (−0.22), even as knowledge scores increased. A plausible interpretation is that, as students encountered the real complexities of MQTT topic structures, broker configuration, and multi-device interaction, they revised their initial self-assessment and developed a more realistic view of their skill level. In other words, the labs appear to have moved some students from “unconscious incompetence” to “conscious incompetence,” a common pattern when learners are exposed to the full stack of a new technology.

Finally, “career interest in Industry 4.0 and IIoT” increased modestly (+0.44), indicating that the platform did not merely transmit technical concepts but also helped some students see Industry 4.0 as a relevant and attractive domain. While this shift is smaller than the gains in Ignition-related knowledge and confidence, it is notable given the short duration of the intervention.

Although the sample size was limited, the survey results provide initial quantitative evidence that the smart home IIoT/SCADA platform, combined with the associated lab sequence, enhances students’ understanding of key technologies and their awareness of how these technologies fit together in a system. At the same time, the mixed confidence results highlight the need for continued support and practice around MQTT and multi-device integration, which can inform the design of future lab activities.

Usability and Scalability

Beyond raw cost, the value of an educational platform depends on how easily students and instructors can use it, and how well it can grow with curriculum needs. Here we discuss usability from the standpoint of HMI interaction and lab workflow, and scalability in terms of both hardware and pedagogical extension.

1) Usability: HMI and Lab Workflow

The platform’s usability is strongly shaped by three design decisions:

1. **Web-based HMI accessible from any browser-capable device.** By hosting the HMI on the Raspberry Pi and exposing it over the local network, students can interact with the system from laptops, lab PCs, or tablets without installing vendor-specific software. This browser-based access reduces setup friction and allows instructors to monitor or demonstrate from their own machines.
2. **Integrated smart-home board that minimizes wiring complexity.** The use of a pre-integrated ESP32 smart home kit (with LEDs, relays, sensors, and input devices on a single PCB) greatly simplifies the learning experience. Students are not required to spend extensive time debugging breadboard wiring or loose jumper connections; instead, they can focus on mapping sensors and actuators to MQTT topics and tags, and on designing HMI elements and simple control logic. This aligns well with the educational goal of emphasizing **system-level thinking** over low-level assembly.
3. **Stepwise lab progression from connectivity to supervisory control.** The accompanying labs (e.g., basic setup, MQTT communication, tag configuration, and smart-home operation) are structured so that each step visibly “unlocks” new capabilities:
 - Early labs confirm connectivity and data paths by having students observe sensors updating on the dashboard and toggling actuators.
 - Later labs layer on supervisory control concepts, such as grouping devices, simulating occupancy, and creating simple automation behaviors.

This progression helps students build confidence: HMI controls offer immediate visual and physical feedback (e.g., a light turning on, or a value changing), which reinforces the connection between abstract tags and tangible device behavior. From an instructor’s perspective, the ability to quickly verify correct operation (e.g.,



If I press this HMI button, does that LED change state?") also makes troubleshooting and grading more straightforward.

Although formal usability studies (e.g., SUS scores or time-on-task metrics) have not yet been conducted, the architecture is deliberately aligned with established HMI usability principles: clear mapping between controls and plant objects, real-time feedback, and a consistent visual metaphor (rooms, devices, and states) that matches students' mental model of a home.

2) Scalability: Hardware, Software, and Curriculum

The platform is also designed to scale along several dimensions.

- **Hardware scalability.** Additional ESP32-based nodes (for example, a second smart home board, a mock "garage" or "garden" node, or even simple standalone sensor boxes) can be added to the same MQTT broker with minimal configuration. Each new device simply publishes and subscribes to its own topic hierarchy. Because the Raspberry Pi can host many MQTT clients simultaneously and the smart home kit is based on a widely available ESP32 module, expanding the "neighborhood" is primarily a matter of purchasing additional low-cost boards and copying or slightly modifying existing firmware.
- **Software and functional scalability.** On the software side, the tag structure and HMI project can be extended by:
 - Creating additional pages or views (e.g., per room or per floor),
 - Adding new trends, alarm conditions, or derived/calculated tags,
 - Integrating simple scripts to implement higher-level logic.

As course needs evolve, instructors can introduce more sophisticated supervisory behaviors (e.g., demand-response scenarios, scheduled scenes, occupancy-based control) without changing the underlying hardware. The MQTT-based architecture also makes it straightforward to connect the platform to external services such as cloud dashboards, databases, or analytics scripts.

- **Pedagogical scalability.** Because the station is self-contained and relatively low-cost, entire lab sections can be built around multiple identical stations, enabling:
 - Introductory classes to focus on basic networking, MQTT, and HMI building.
 - Intermediate courses to explore alarm design, tag naming conventions, and data logging.
 - Advanced topics (e.g., security, edge analytics, or integration with actual building energy data) to be layered on top, using the same physical platform.

Instructors can also assign small groups to customize their smart home station (e.g., redesign the dashboard, implement a specific automation scenario, or prototype an energy-saving strategy), leveraging the platform as a project-based learning scaffold that remains consistent across course levels.

In summary, the evaluation indicates that the proposed smart home SCADA platform is:

- **Cost-effective**, enabling multiple stations to be deployed for less than the price of a single industrial trainer while still supporting rich system-level learning.
- **Usable**, thanks to a browser-based HMI, simplified hardware integration, and labs that guide students from basic connectivity to supervisory control.
- **Scalable**, both technically and pedagogically, allowing additional devices, more complex control algorithms, and advanced course content to be added without redesigning the core architecture.

A natural next step—beyond the scope of the current implementation—is to pair this design-oriented evaluation with formal empirical studies (e.g., pre/post concept inventories, usability surveys, and longitudinal tracking of student outcomes) to quantitatively measure learning gains and refine the lab sequence accordingly.



Conclusion

Summary of Findings

This work presented the design and implementation of a low-cost smart home IIoT/SCADA platform tailored for IoT and supervisory control education. By combining a Raspberry Pi edge node, an ESP32-based smart home kit, a compact router, and an MQTT-centric communication architecture, the platform achieves a complete three-tier design—field, middleware, and supervisory layers—at a hardware cost on the order of \$200 per station. This meets the initial objective of enabling multiple, independently operable stations to be deployed within the budget of a typical teaching laboratory, rather than relying on a small number of high-cost industrial trainers. Additional details about the implementation can be found at: <https://inductiveautomation.com/resources/casestudy/clovis-community-college>

Beyond cost, the platform’s main contribution lies in its integration of technical fidelity and pedagogical structure. Technically, the system mirrors modern IIoT/SCADA deployments: field devices publish and subscribe via MQTT, an edge node hosts the broker and SCADA gateway, and students interact through a browser-based HMI that exposes tags, control widgets, and real-time feedback. Pedagogically, this architecture is embedded in a sequence of labs that move from prototype construction and network bring-up, through topic/tag mapping and functional testing, to higher-level supervisory operation in a “smart home neighborhood.” This progression provides students with hands-on experience across the full stack—from embedded code to network messaging to supervisory control—using a familiar and engaging smart home context.

Limitations

The current platform and study have several limitations that frame the scope of the reported results. First, although we now include entry–exit survey data that show measurable gains in knowledge, familiarity, and interest, the evaluation remains “exploratory and small-scale”. The survey was administered to a single cohort without a control group, and we did not collect variance measures or perform statistical significance testing, which limits the strength of causal claims about learning gains.

Second, while the hardware is sufficient for the intended educational scenarios, it remains constrained by the capabilities of low-cost components. The Raspberry Pi has finite processing and memory resources, which may limit the complexity of concurrent analytics, large-scale data logging, or heavyweight security mechanisms. Similarly, the ESP32 smart home kit offers a fixed and relatively small set of I/O points, which constrains the scale and diversity of physical scenarios that can be represented without adding additional modules.

Third, the current implementation still emphasizes “functionality and accessibility over security and robustness”. MQTT communication is configured for a trusted lab environment, and advanced topics such as encrypted transport, authentication, and role-based access control in the HMI are not yet central to the lab sequence. Finally, the platform relies on a commercial SCADA environment (Ignition), albeit under educational licensing; long-term sustainability and portability may be affected if licensing terms change or if institutions seek fully open-source alternatives.

Future Work

Future work will pursue three complementary directions. On the technical side, the platform can be extended to integrate real-world cloud services and data pipelines, for example, by forwarding selected MQTT topics to cloud dashboards, databases, or analytics platforms. This would allow students to explore edge–cloud architectures and topics such as latency, bandwidth, and data lifecycle management. In parallel, security-focused enhancements—such as enabling TLS for MQTT, implementing authenticated user roles in the HMI, and introducing basic intrusion or anomaly detection scenarios—would provide a natural bridge to cybersecurity and critical-infrastructure protection topics.



On the curriculum side, there is an opportunity to formalize and disseminate a standardized instructional package built around the platform. This would include detailed lesson plans, assessment rubrics, concept inventories for IoT and SCADA topics, and modular project ideas aligned with different course levels (from introductory IoT to advanced industrial automation). Such a curriculum would make it easier for other institutions to adopt and adapt the platform, and would support multi-site studies of its impact.

Finally, on the research and evaluation side, we have taken a first step by collecting entry–exit survey data on students’ self-reported knowledge, confidence, and career interest. Building on these preliminary results, future work will focus on more systematic empirical studies of student learning and user experience. This includes refining and expanding the pre/post instruments to target key concepts (MQTT architecture, tag-based SCADA models, HMI design principles), administering standardized usability and engagement surveys, and collecting qualitative feedback from both students and instructors. Longitudinal use across multiple semesters will also make it possible to study how the platform supports progression from basic familiarity with IoT devices to more sophisticated system-level reasoning, and to compare outcomes against alternative instructional approaches. Taken together, these extensions will deepen the platform’s value as both a practical teaching tool and a research testbed for low-cost, scalable IIoT/SCADA education.

Acknowledgements. This work was supported by the National Science Foundation’s Advanced Technology Education Program under award 2202201. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. The following students from Colvis Community College participated in the Smart Home project: Neuro Cabrera, Meagan Eggert, Julian Perry Laxamana, Gurkaran Singh, Mohammad Abumaali, and Benjamin Hallaway.

Disclosures. The authors declare no conflicts of interest.

References

- [1] S. Deshpande and S. Hsieh, “Cyber-physical system for smart traffic light control,” *Sensors*, vol. 23, no. 11, p. 5028, Jun. 2023, doi: 10.3390/s23115028.
- [2] A. Al-Kharusi, M. Al-Kalbani, N. A. Al-Majrafi, and J. Al-Habsi, “Revolutionizing supply chain education: Integrating Industry 4.0 and IoT into university curriculum,” in *Proc. IEOM Soc.*, Tokyo, Japan, 2024, pp. 1–5. [Online]. Available: <https://ieomsociety.org/proceedings/2024tokyo/194.pdf>
- [3] Ednex, “Industry 4.0 & IoT curriculum for universities and colleges,” *Ednex*, n.d. [Online]. Available: <https://ednex.me/higher-education/industry-4-0-curriculum/>
- [4] F. Qiu, A. Kumar, J. Hu, P. Sharma, Y. B. Tang, Y. Xu Xiang, and J. Hong, “A review on integrating IoT, IIoT, and Industry 4.0: A pathway to smart manufacturing and digital transformation,” *IET Inf. Secur.*, vol. 2025, no. 1, p. 9275962, 2025.
- [5] Cranes Varsity Pvt. Ltd., “Why closing the skill gap is critical for Industry 4.0 readiness,” *LinkedIn*, Jul. 2025. [Online]. Available: <https://www.linkedin.com/pulse/why-closing-skill-gap-critical-industry-40-readiness-kz9jc>
- [6] T. Ion, “The importance of SCADA systems educational programs in higher education from a cybersecurity perspective,” in *Proc. BASIQ Int. Conf.*, 2021, pp. 835–843. [Online]. Available: <https://basiq.ro/papers/2021/21105%20-%20Ion%20T.pdf>
- [7] M. Dehbozorgi and D. Contestabile, “Bridging the skill gap in engineering education: Designing learning factories for Industry 4.0 and 5.0,” in *Proc. 2023 ASEE Annu. Conf. Expo.*, Baltimore, MD, USA, Jun. 2023.



- [8] S. Hsieh, M. Barger, S. Marzano, and J. Song, "Preparing the manufacturing workforce for Industry 4.0 technology implementation," in *Proc. 2023 ASEE Annu. Conf. Expo.*, Baltimore, MD, USA, Jun. 2023.
- [9] A. Alcaide, I. Robalo, F. G. Montoya, and F. Manzano-Agugliaro, "SCADA system for online electrical engineering education," *Inventions*, vol. 7, no. 4, p. 115, Dec. 2022, doi: 10.3390/inventions7040115.
- [10] D. Mourtzis, E. Vlachou, G. Dimitrakopoulos, and V. Zogopoulos, "Cyber-physical systems and education 4.0—The teaching factory 4.0 concept," *Procedia Manuf.*, vol. 23, pp. 129–134, 2018, doi: 10.1016/j.promfg.2018.04.005.
- [11] PLC Training, "SCADA - HMI training," *PLC-Training.org*, n.d. [Online]. Available: <https://plc-training.org/scada-hmi-training.html>
- [12] Tatsoft, "The 8 best SCADA platforms in the world: A practical comparison for modern industrial applications," *Tatsoft*, n.d. [Online]. Available: <https://tatsoft.com/top-8-scada-platforms-compared/>
- [13] M. Al-Yaman, D. Alswaiti, A. Alsharkawi, and M. Al-Tae, "A cost-effective modular laboratory solution for industrial automation and applied engineering education," *MethodsX*, vol. 14, p. 103388, May 2025, doi: 10.1016/j.mex.2025.103388.
- [14] X. Zhang, "A comprehensive guide to at-home SCADA learning for beginners," *LinkedIn*, Jul. 2025. [Online]. Available: <https://www.linkedin.com/pulse/comprehensive-guide-at-home-scada-learning-beginners-xuetian-zhang-ejfme>
- [15] L. Ahsan, M. J. A. Baig, and M. T. Iqbal, "Low-cost, open-source, Emoncms-based SCADA system for a large grid-connected PV system," *Sensors*, vol. 22, no. 18, p. 6733, Sep. 2022, doi: 10.3390/s22186733.
- [16] B. Bajracharya, V. Gondi, and D. Hua, "IoT education using learning kits of IoT devices," *Inf. Syst. Educ. J. (ISEDJ)*, vol. 19, no. 6, pp. 40–44, Dec. 2021. [Online]. Available: <https://files.eric.ed.gov/fulltext/EJ1329491.pdf>
- [17] Arduino, "Explore IoT kit," *Arduino Education*, n.d. [Online]. Available: <https://www.arduino.cc/education/explore-iot-kit>
- [18] ACEBOTT, "QE023 smart home edu kit," *ACEBOTT*, n.d. [Online]. Available: <https://acebott.com/product/acebott-qe023-esp32-smart-home-education-kit-level-1/>
- [19] J. Siddiqi, A. S. Gandy, and S. Hsieh, "Technical training for Industry 4.0 technologies: Low-cost gantry candy sorting system for education and outreach," in *Proc. 2024 ASEE Annu. Conf. Expo.*, Portland, OR, USA, Jun. 2024.
- [20] S. Shinde, "How to use MQTT in Node-RED," *FlowFuse*, Jun. 2024. [Online]. Available: <https://flowfuse.com/blog/2024/06/how-to-use-mqtt-in-node-red/>
- [21] A. Zare and M. T. Iqbal, "Low-cost ESP32, Raspberry Pi, Node-Red, and MQTT protocol based SCADA system," in *Proc. 2020 IEEE Int. IOT, Electron. Mechatron. Conf. (IEMTRONICS)*, Vancouver, BC, Canada, 2020, pp. 1–5, doi: 10.1109/IEMTRONICS51293.2020.9216432.
- [22] K. Walz, K. Cooper, B. Reid, C. Baechle, C. Akelian, and K. Alfano, "Supervisory controls and data acquisition instructional materials and resources for energy education programs," in *Proc. 2022 ASEE Annu. Conf. & Expo.*, Minneapolis, MN, USA, Aug. 23–26, 2022, doi: 10.18260/1-2--40599.
- [23] N.A. Mamaril, E.L. Usher, C.R. Li, D.R. Economy and M.S. Kennedy, "Measuring undergraduate students' engineering self-efficacy: A validation study," *Journal of Engineering Education*, 105(2), pp.366-395, 2016.