



Final Year Project Report

Development of an Integrated Industrial IoT System for Real-Time Data Monitoring and Control

Zhalgas Bolatbayev, Yerkebulan Sozakbay

A thesis submitted in part fulfilment of the degree of

BSc in Robotics and Mechatronics

Supervisor: Dr. Alizadeh

Department of Robotics and Mechatronics
Nazarbayev University

May 7, 2025

Table of Contents

Abstract	3
1 Introduction	5
1.1 Motivation	5
1.2 Problem Definition	6
1.3 Possible Practical Applications	6
2 Background Research	8
2.1 Digital Twin and its significance	8
2.2 Current Challenge	8
2.3 Related Work	9
2.4 Data Collection	11
2.5 Synthetic Data	12
2.6 Predictive Maintenance	12
2.7 Tools	12
2.8 Limitations	13
2.9 Relation to our Project	13
3 Methodology	14
3.1 Setting up Sensors	14
3.2 Data Flow	14
3.3 Data Storage	15
3.4 Security	15
3.5 Real-time Monitoring	15
4 Execution	17
4.1 Overall plan and process in general	17
4.2 Sensors set up and data transmission	18
4.3 Using Node-red	18
4.4 Work with database (data collection)	19

4.5	Cybersecurity measures	19
5	Results(Zhargas)	21
6	Conclusion and Future Work	22

Abstract

This report presents the project about building a working prototype that collects real-time industrial data using actual hardware and sends it securely to the cloud for monitoring, analysis, and future machine learning use. We used the MPS 500 modular production system as our base and set up sensors to measure vibration, temperature, and liquid leaks. These sensors help detect early signs of equipment wear or failure. The data is sent through a mix of modern communication protocols like MQTT and S7 using ESP32 microcontrollers and Siemens PLCs.

To make everything work smoothly, we built a central data flow using Node-RED, which handles the logic, processes the data, and pushes it to both a local InfluxDB (for time-series storage) and Google Cloud (for remote access and analytics). We also made a live dashboard where operators can see what's happening in real time—like live sensor values, machine status, and performance alerts.

Security was a top priority, so we encrypted everything using TLS, used strong credentials, and added certificate-based authentication to prevent unauthorized access. Our setup balances low cost with real-world functionality, making it a solid option for small manufacturers or researchers who want access to real operational data for digital twin or predictive maintenance development. In the future, this system could be extended further with AI-driven analytics and more edge computing features.

Acknowledgments

Dr.Alizadeh, our supervisor, for his expert advice, providing the opportunity, sources and devices for work, constant support and guidance throughout this current process.

Chapter 1: Introduction

The Fourth Industrial Revolution Industry 4.0 is rapidly transforming how manufacturing and industrial systems operate. At the heart of this transformation is the Industrial Internet of Things (IIoT), which connects machines, sensors, controllers, and cloud platforms into unified, intelligent systems. These IIoT-enabled systems are capable of real-time monitoring, remote control, data collection, and even autonomous decision-making, leading to significant improvements in efficiency, reliability, and safety across various industries.

One of the most promising applications of IIoT is the creation of Digital Twins, virtual models of physical assets that update in real time based on sensor data. Digital twins allow operators and engineers to simulate processes, monitor machine performance, detect errors, and predict failures before they occur. However, while the concept of the digital twin is widely embraced, its implementation in real environments still faces a critical barrier: access to real, high-quality, labeled operational data.

In many cases, data scientists are forced to rely on synthetic or simulated datasets to develop predictive maintenance models or test digital twin functionality. This leads to less accurate models that don't fully capture the complex behavior of real systems under varying conditions. The absence of real-world datasets also prevents researchers from benchmarking their work effectively and slows down industrial adoption of data-driven maintenance strategies.

1.1 Motivation

The primary motivation behind this project is to bridge the gap between physical industrial systems and digital twin models by developing a reliable, modular IIoT-based data delivery and monitoring platform. By using the MPS 500 modular production system as a base, our aim was to deploy real sensors, collect live industrial data, and enable secure communication between devices and cloud services. This data could then be used to support not just monitoring, but long-term predictive analytics and machine learning applications.

Our goal was to create a system that is scalable, secure, and easy to integrate into real-world industrial setups. To achieve this, we focused on integrating a number of technologies, including:

1. Sensor setup: Connecting temperature, humidity, and motion sensors to monitor critical environmental and system parameters.
2. Protocol design: Using lightweight communication protocols such as MQTT and OPC UA for transmitting data both locally (within the plant network) and remotely (to the cloud).
3. Cloud integration: Linking the system to Microsoft Azure and Google Cloud for data storage, real-time access, and visualization through Node-RED.
4. Remote control: Creating an interface for remote control and live system monitoring.

5. Security layer: Implementing TLS encryption, certificates, and user authentication to ensure secure and authorized access to industrial data.

This project not only solves a technical challenge but also serves as a real, applicable framework that industries in Kazakhstan and beyond can adopt to accelerate their digital transformation efforts.

1.2 Problem Definition

Despite the widespread theoretical adoption of digital twin concepts in industrial settings, there remains a fundamental barrier to their practical implementation: the lack of accessible, high-quality, real-time operational data from physical systems. Digital twins rely on constant data streams from industrial hardware in order to remain accurate, responsive, and useful. However, most small and medium-sized enterprises, and even many research projects, do not have access to such data sources. As a result, researchers and developers frequently resort to using synthetic or simulated data to model system behavior and train predictive maintenance algorithms.

This creates a gap between academic models and real-world deployment. Synthetic data, while useful for initial model validation, often fails to capture the complexity, noise, and variability found in actual industrial environments. Models trained on such data are prone to overfitting or poor generalization, resulting in unreliable or inaccurate performance when deployed in practice. This issue is particularly critical in predictive maintenance, where early fault detection relies on subtle changes in sensor patterns over time.

How can we design and implement a reliable, secure, and real-time IIoT system for collecting and delivering authentic industrial data that supports digital twin development and predictive maintenance modeling?

The question raised above was the main research problem addressed in this capstone project

1.3 Possible Practical Applications

This system can be used in multiple real-world industrial settings where monitoring and predictive control are essential. For example:

- Manufacturing Lines: Monitor temperature and humidity around sensitive equipment and trigger automatic shutdowns or alerts when limits are exceeded.
- Smart Production Labs: Gather long-term sensor data for machine learning projects to predict machine failures or optimize cycle times.
- Remote Industrial Monitoring: View real-time data from multiple production sites via a mobile dashboard and detect anomalies instantly.
- Research Centers: Use collected data to train predictive models using real operational conditions rather than simulations.

These use cases highlight the broad applicability and value of the solution, particularly for companies transitioning to Industry 4.0 and for data scientists seeking to work with clean, real data for industrial AI projects.

Overall, the project shows how IIoT systems can make industrial processes faster, smarter, and more efficient. Through this project, our objective is to demonstrate the transformative potential of cloud integration in industrial automation, offering a scalable, secure and user-friendly solution for modern industrial systems.

Chapter 2: Background Research

The development of cyber-physical systems (CPS) and digital transformation in the manufacturing industry has marked a paradigm shift in how we think about production, system management, and maintenance. With the emergence of Industry 4.0, there has been a significant emphasis on integrating advanced technologies such as IIoT, machine learning, cloud computing, and digital twins to create smarter, interconnected industrial environments.

This section examines the body of research surrounding these technologies, focusing on real-time data collection, predictive maintenance, and the creation of digital twins. Furthermore, it analyzes the tools and frameworks adopted in similar development projects and elaborates on the challenges related to real-world data access, which this project aims to address.

2.1 Digital Twin and its significance

The concept of the digital twin, first proposed by Michael Grieves in 2003, refers to a virtual representation of a physical system that is continuously updated with real-time data from sensors embedded in that system. These virtual models are used for performance analysis, diagnostics, simulation, and predictive maintenance. According to Tao et al. (2019), a digital twin framework typically comprises the physical entity, the virtual model, and the data communication interface between them.

The implementation of digital twins enables organizations to:

1. Monitor system performance in real time.
2. Run simulations under varying parameters.
3. Predict potential failures before they occur.
4. Support decision-making based on historical and live operational data.

As Boschert and Rosen (2016) emphasize, digital twins are not static models but dynamic systems that evolve based on the state of their physical counterparts. This makes them crucial for predictive maintenance, where early detection of anomalies can prevent costly downtime and equipment failure.

2.2 Current Challenge

Despite the theoretical potential of digital twins, one of the most pressing challenges is access to high quality real-time operational data. Most research relies on synthetic or lab-generated datasets that do not accurately represent the chaotic and noisy nature of real industrial

environments (Wang et al., 2020). This leads to digital twins that may work well in simulation but perform poorly in real-world deployment.

For example, predictive maintenance models often depend on patterns that develop over time from actual sensor data. Synthetic datasets can capture broad behaviors but fail to represent subtle degradation signals that only manifest under real-world stressors. This mismatch results in poor generalizability of trained machine learning models and can erode trust in digital systems.

2.3 Related Work

Several research projects have attempted to bridge this gap through testbeds, synthetic twin environments, or limited access to proprietary datasets. Below are some notable examples:

1. The Smart Factory Web (Fraunhofer Institute):

This is a visionary initiative by the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) and the Fraunhofer Institute for Software and Systems Engineering (ISST). It aims to create a network of smart, interconnected factories that can dynamically collaborate, share production resources, and optimize manufacturing capabilities across locations. This concept aligns closely with the goals of Industry 4.0, emphasizing interoperability, flexibility, and real-time responsiveness in modern industrial systems.

At its core, the Smart Factory Web provides a web-based platform that connects multiple factory sites through a common digital infrastructure. Each connected factory exposes its production capabilities, resource availability, and scheduling flexibility via standardized interfaces. This allows factories to form temporary partnerships or production alliances, enabling dynamic order fulfillment based on real-time capacity and capabilities.

The architecture leverages Industrial Internet of Things (IIoT) technologies, semantic data models, and digital twins to create a virtual representation of each factory and its assets. Using OPC UA, REST APIs, and MQTT protocols, it ensures seamless communication between devices and systems. Additionally, the platform supports cloud-based analytics, edge computing, and machine learning for predictive maintenance, quality control, and production optimization.

One of the key demonstrations of the Smart Factory Web is the international production network prototype, where factories in Germany and South Korea dynamically collaborated to handle manufacturing tasks. This proves the feasibility of cross-border industrial cooperation powered by digital technologies.

The Smart Factory Web supports a wide range of use cases including:

- Dynamic production network optimization
- Real-time factory monitoring
- Decentralized decision-making
- Flexible supply chain integration

Overall, the Smart Factory Web represents a major step toward adaptive, collaborative manufacturing ecosystems, showcasing how distributed smart factories can work together to achieve higher efficiency, agility, and resilience in global production networks. However, due to IP restrictions, it lacks open access to data for external researchers.

2. **PHM Society Data Challenges:** The Prognostics and Health Management (PHM) Society Data Challenges are internationally recognized competitions that focus on advancing the field of predictive maintenance, fault detection, and diagnostics through the use of real or simulated industrial datasets. Organized annually by the PHM Society, these challenges serve as an essential platform for researchers, students, and professionals to develop and benchmark their algorithms in realistic conditions.

The core objective of the PHM Data Challenges is to foster innovation in data-driven maintenance strategies by providing high-quality datasets that replicate real-world industrial systems. These datasets typically include sensor readings, operational conditions, maintenance logs, and failure events from domains such as aerospace, manufacturing, energy systems, and transportation. Participants are tasked with using this data to predict system degradation, detect faults, or estimate the remaining useful life (RUL) of components.

Unlike many open datasets, PHM Challenge data is often rich, time-series-based, and labeled, offering a closer approximation to what engineers encounter in operational settings. Past challenges have included NASA's C-MAPSS jet engine degradation data, Prognostics datasets for rotating machinery (bearings, motors, gearboxes), Wind turbine and power electronics health data, Industrial robotics failure prediction datasets.

These challenges have played a crucial role in standardizing evaluation metrics, such as scoring functions for prediction accuracy and uncertainty, allowing consistent comparison between models. Moreover, they have encouraged the use of machine learning, deep learning, Bayesian modeling, and physics-informed hybrid models in prognostics research.

The PHM Society also publishes the results of these challenges in associated technical reports and invites top-performing teams to present their methodologies at the Annual Conference of the PHM Society. This creates a valuable feedback loop between academia and industry, where practical needs and theoretical advancements intersect.

In the context of IIoT and digital twins, the PHM Society Data Challenges highlight the importance of realistic, complex datasets for developing reliable predictive maintenance solutions. They also underscore the ongoing need for high-quality, real-time operational data—a challenge this project directly addresses by building a real-world IIoT platform capable of producing such data for future research and application. While useful for benchmarking, these datasets are still mostly artificially generated, lacking the unpredictability and complexity of real data.

3. **ADAMOS IIoT Platform:** ADAMOS (ADaptive Manufacturing Open Solutions) is an Industrial Internet of Things (IIoT) platform designed specifically for the mechanical and plant engineering sector, offering manufacturers a modular and vendor-neutral solution for advancing digital transformation in Industry 4.0 environments. Initiated by a consortium of leading industrial companies including DMG MORI, Software AG, ZEISS, Dürer, and ASM PT, ADAMOS combines deep domain knowledge with state-of-the-art digital technologies to support smart factory implementations.

The ADAMOS platform provides a centralized infrastructure for machine connectivity, data collection, analytics, and app development, all tailored to the needs of manufacturing enterprises. One of its core strengths lies in enabling interoperability across diverse machine types and vendors, solving a major challenge in heterogeneous production environments. It supports standardized communication protocols such as OPC UA and MQTT, ensuring seamless integration between machines, edge devices, and cloud-based applications.

The platform is designed with a strong emphasis on usability and flexibility. Through its App Store and App Factory, companies can deploy, develop, and manage a wide range of industrial apps for functions like:

- Condition monitoring
- Predictive maintenance
- OEE (Overall Equipment Effectiveness) analysis
- Workflow automation
- Remote machine service and diagnostics

Moreover, ADAMOS offers built-in data governance, security features, and role-based access control, making it suitable for secure industrial deployment. It supports edge computing capabilities, allowing data to be pre-processed near the machine to reduce latency and improve responsiveness—crucial for real-time applications.

The platform also promotes collaborative innovation by enabling OEMs and end-users to co-create solutions on a shared digital foundation. This ecosystem-oriented approach enhances scalability and accelerates the development of intelligent services that are aligned with real-world shopfloor requirements.

In the context of IIoT research and smart production, ADAMOS exemplifies a practical implementation of how industrial connectivity, data analytics, and cloud integration can be orchestrated to achieve operational excellence. It aligns closely with this project's goal of building modular, real-time IIoT architectures for data-driven applications such as digital twins and predictive maintenance, demonstrating the effectiveness of open, vendor-neutral platforms in modern manufacturing.

In contrast, our project aims to open the black box of real-world industrial data by creating a publicly usable IIoT data platform based on the MPS 500 system.

2.4 Data Collection

A central component of any digital twin or industrial AI system is data. The data pipeline includes:

- Sensor setup – Choosing and installing sensors to collect physical measurements like temperature, pressure, humidity, vibration, etc.
- Communication protocols – Transmitting the data from edge devices to servers or the cloud. Protocols include:
 - MQTT – Lightweight, publish-subscribe model, ideal for constrained devices.
 - OPC UA – A platform-independent standard used in industrial automation.
 - Modbus/TCP, HTTP, CoAP – Other protocols with varying degrees of reliability, latency, and overhead.
- Data storage – Historical data is essential for training and validation of ML models. It is often stored in time-series databases like InfluxDB or on cloud platforms such as Microsoft Azure IoT Hub or Google Cloud IoT Core.
- Security – Encryption (TLS), certificate-based authentication, and role-based access control are critical to protect sensitive industrial data.

Our system implements this full pipeline with secure communication using TLS encryption, flexible protocol selection (MQTT and OPC UA), and scalable cloud data handling.

2.5 Synthetic Data

Due to the scarcity of real data, many researchers rely on synthetic data for predictive modeling. For example, Zhang et al. (2021) created synthetic failure datasets using physics-based simulations to train a convolutional neural network (CNN) for fault prediction in pumps. While the model performed well in controlled tests, it showed reduced accuracy on real-world deployment due to unmodeled noise and unexpected behaviors.

This limitation is echoed by Liu et al. (2020), who stress the importance of hybrid modeling, combining physics-based and data-driven models, but caution against over-reliance on non-empirical data. Without ground-truth failure data, it is difficult to validate model assumptions and measure performance under stress conditions.

This project directly addresses the issue by collecting live sensor data in an industrial environment, enabling the development and benchmarking of more realistic models.

2.6 Predictive Maintenance

Predictive maintenance involves using analytics and machine learning to anticipate failures before they happen. Common approaches include:

- Statistical techniques: Linear regression, ARIMA, and moving averages.
- Machine learning: Random forests, support vector machines (SVM), and decision trees.
- Deep learning: Recurrent neural networks (RNN), long short-term memory (LSTM), and CNNs for time-series analysis.

All of these methods require labeled datasets that contain both normal and faulty behavior. The lack of fault labels in real data is a persistent issue. Some researchers tackle this with anomaly detection, but it still requires a well-distributed dataset with sufficient variability, which our platform intends to provide.

2.7 Tools

Our development approach uses a modern stack of tools tailored to IIoT applications:

- PLC Programming (TIA Portal, Siemens S7-1200): Configuring the physical control logic that runs on industrial-grade hardware.
- Node-RED: A flow-based development tool for integrating hardware devices, APIs, and online services. It allows intuitive visualization and manipulation of live data streams.
- MQTT Broker (Mosquitto): Lightweight server to handle message delivery in a publish-subscribe model.

- OPC UA Server: Facilitates structured data communication and interoperability between different devices.
- Cloud Platforms (Azure, Google Cloud): Used for data storage, dashboard creation, and ML integration.
- Security Modules: TLS encryption, X.509 certificates, and hashed credentials for authentication and secure access.

These technologies are chosen based on real-world compatibility, extensibility, and wide support in the industrial community.

2.8 Limitations

While previous research demonstrates the viability of digital twins and predictive maintenance, there is a clear gap in the availability of systems that offer:

- Real-time, real-world data acquisition.
- Open access to datasets for ML and analytics research.
- Flexible and secure protocol integration.
- A modular and low-cost platform for SME adoption.

This project directly addresses these gaps by creating a working IIoT prototype with a fully integrated data pipeline using actual hardware (MPS 500), real sensor data, and live transmission over secure channels.

2.9 Relation to our Project

This background review has provided us insight into the research landscape surrounding digital twins, IIoT data acquisition, predictive maintenance, and real-time monitoring. It has highlighted the technological advancements made, the challenges faced by researchers, particularly regarding data access and the solutions that have been explored.

Our project builds upon this foundation by implementing a modular, cloud-connected IIoT system that delivers real operational data for use in digital twin development. It bridges a crucial gap between theoretical models and real-world industrial needs, offering both practical applicability and academic value.

By contributing a real-time data collection platform, we enable future researchers and industries to test their models in authentic conditions, advancing the vision of a truly connected and intelligent manufacturing ecosystem.

Chapter 3: Methodology

The main goal of our project is to build a system that can collect real industrial data from sensors in real time, send that data securely using modern protocols, and store it for future analysis, like machine learning or digital twin development. Below is a breakdown of how we approached this.

3.1 Setting up Sensors

We started by inspecting the physical setup of the MPS 500 modular production system, particularly focusing on the handling and processing stations. The goal was to figure out where we could place sensors to get the most useful data.

- **Handling Station:** This station uses a gripper that slides on rails and needs lubrication to reduce friction. Over time, friction increases if it is not maintained well. So, we placed vibration sensors near the moving parts to catch early signs of wear or improper lubrication.
- **Processing Station:** Here, the key component is a motor. To monitor its health, we used temperature sensors and oil leakage sensors. These help us track if the motor is overheating or leaking, both of which could indicate a problem.

We also took note of how long each station takes to complete its actions under normal conditions. By recording this timing data, we can monitor station performance over time and catch inefficiencies or malfunctions.

3.2 Data Flow

Once the sensors were in place, we needed a way to collect and transmit that data. For this, we used Node-RED, a flow-based tool that acts like a middleman between the sensors and the places where we store and visualize the data. The data flow looks like this:

Sensors → Node-RED → Storage and User Interface

We used a few different communication protocols depending on the type of sensor and what it connects to:

1. **S7 Protocol (Ethernet ISO-on-TCP):** This is used to get data directly from Siemens PLCs (Programmable Logic Controllers).
2. **MQTT Protocol:** This is what we used for wireless sensors, like the ones connected to ESP32 microcontrollers. MQTT is lightweight and supports a topic-based pub-

lish/subscribe system, which makes it great for IIoT setups. We tested ports 1883 (unencrypted) and 8883 (TLS encrypted) for secure communication.

3.3 Data Storage

We needed a way to store the sensor data for both short-term and long-term use. For this, we chose two systems:

- InfluxDB: This is a local time-series database. It's good for storing sensor data because it handles timestamps well and supports secure access through user authentication.
- Google Cloud (Google Sheets): For remote access and experimental purposes, we also sent data to Google Sheets through Node-RED. This allows remote users to view live updates without accessing the internal network.

Both storage systems let us analyze historical trends and patterns, which is useful for data scientists or engineers working on predictive maintenance or digital twins.

3.4 Security

Since we're dealing with industrial data, security is a must. We took a few steps to make sure the system is safe:

- TLS Encryption: Especially for MQTT on port 8883, we used TLS to encrypt messages and protect data from being intercepted.
- Authentication: InfluxDB supports user authentication, so only authorized users can view or modify the data.
- Certificates: For secure communication between devices, we implemented digital certificates to verify device identity.

3.5 Real-time Monitoring

To make the system user-friendly, we created a dashboard using Node-RED's built-in UI features. This interface allows anyone connected to the network (or with access to the cloud dashboard) to:

- See live sensor readings
- Monitor the status of machines

- Check for alerts if something goes wrong (like high vibration or abnormal temperature)
- Track timing and cycle performance of stations

This part is especially useful for operators and maintenance teams, because it gives them an instant overview of how the system is doing, removing need to dig through logs or raw data.

Chapter 4: Execution

4.1 Overall plan and process in general

This project presents an end-to-end monitoring solution for an industrial handling and processing station that seamlessly combines measurements from Arduino-based sensors (vibration, gas concentration, leak detection and temperature) with logical and operational data from a Siemens S7 PLC to deliver both real-time insights and historical analysis.

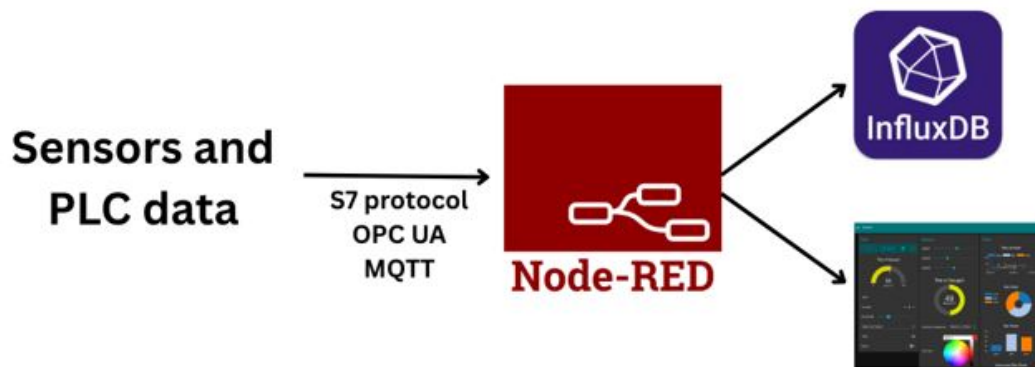


Figure 4.1: Fig.1 Data Flow

Sensor readings are collected by ESP32 microcontrollers and published over MQTT on the local Ethernet/Wi-Fi network, while PLC tag values (motor states, fault codes, cycle counts, etc.) are retrieved via native S7 and OPC UA protocols using Siemens TIA Portal. All incoming data streams are ingested into a central Node-RED instance running on a Raspberry Pi.

Within Node-RED, protocol translation is handled by dedicated nodes for MQTT, S7 and OPC UA, and inline JavaScript functions perform calculations such as vibration RMS computation or gas threshold evaluation. The enriched data is then routed in parallel to both storage backends and the human-machine interface.

High-frequency time-series data (for example, one-second temperature and vibration samples) are stored in an on-premises InfluxDB instance with tailored retention and downsampling policies, ensuring detailed recent history and longer-term aggregated views. Aggregated events and daily summaries are forwarded to Google Cloud Firestore and BigQuery for scalable analytics and long-term archiving.

A web-based HMI dashboard, built with node-red dashboard panels embedded via Node-RED “ui” nodes, queries InfluxDB to render live charts, alarm indicators and historical trends. Critical alerts—such as gas concentration exceeding safety thresholds—trigger email or SMS notifications directly from Node-RED.

Development and deployment tools include Siemens TIA Portal v16 for PLC programming,

Arduino IDE 2.0 for ESP32 firmware, Node.js v14 with node-red-contrib-s7, node-red-contrib-opcua and node-red-contrib-mqtt nodes for middleware logic, InfluxDB 2.0 for time-series storage and visualization, and Node-RED's Google Cloud nodes for data export and offline batch processing.

By integrating low-cost Arduino sensors, industrial PLC data and a protocol-agnostic middleware layer, this solution achieves comprehensive station monitoring, fast local querying and scalable cloud-based analytics. Future work will focus on deploying predictive maintenance algorithms against the historical dataset in Google Cloud and expanding edge-compute capabilities within the Node-RED environment.

4.2 Sensors set up and data transmission

In this project, MPS 500 tags were employed for detailed process monitoring—specifically, to calculate elapsed times for performance evaluation—while maintenance-oriented measurements were captured by Arduino-based temperature, vibration, and liquid-leak sensors.

Sensor data were aggregated on an ESP32 microcontroller and initially published via MQTT to public brokers (e.g., test.mosquitto.org, himemq.com, broker.emqx.io), whose simplicity and topic-based architecture facilitated rapid deployment but lacked confidentiality safeguards.

To mitigate unauthorized access, the MQTT layer was migrated to a private EMQX.io cloud service on port 8883, leveraging TLS encryption, obscure server endpoints, password-protected authorization, and client certificates to restrict data access.

Concurrently, PLC tag values were acquired over the Siemens S7 protocol; overcoming administrative and cybersecurity barriers required procuring permission certificates and enabling PUT/GET permissions on the PLCs, thus authorizing controlled read–write access to process parameters.

This combined architecture ensures secure, end-to-end visibility of both sensor and PLC data for comprehensive station monitoring.

4.3 Using Node-red

Node-RED can serve as a transparent bridge for transmitting information and data between disparate system components. By encapsulating each data source (sensors, PLCs, cloud services) into modular input and output nodes, it enables seamless flow of process values—such as vibration levels, temperature readings, and cycle counts—from the edge all the way through to storage and visualization layers.

Within this bridge, Node-RED provides powerful logic interactions: inline JavaScript or function nodes can perform arithmetic calculations, threshold evaluations, moving averages and other “magazine-style” computations, as well as arbitrary data-format transformations (for example, converting raw sensor payloads into standardized JSON, CSV or InfluxDB line protocol).

Protocol flexibility is a core strength. The platform natively supports MQTT for light-weight publish/subscribe from ESP32-hosted sensors; OPC UA and S7 for industrial PLC communications; direct writes to time-series stores such as InfluxDB; and even simple integrations with Google Sheets for lightweight archival or reporting. New protocols can be added via community-contributed nodes, ensuring extensibility.

To aid operations teams, Node-RED can host a built-in monitoring interface—dashboard nodes allow rapid construction of live charts, gauges, tables and alarm panels without external web-server configuration. All elements respond in real time to incoming data and can be secured behind existing network authentication.

Finally, its visual, flow-based programming model makes developing and maintaining these integrations exceptionally straightforward. Rather than writing extensive boilerplate code, developers simply drag nodes onto a canvas, wire them together, and configure their properties; custom logic can be added with minimal JavaScript snippets. This ease of programming accelerates deployment and simplifies ongoing maintenance.

4.4 Work with database (data collection)

IIoT forms a vital component of our digital twin initiative, enabling reliable delivery and collection of real-time data streams. This connectivity layer is indispensable for close collaboration with data scientists as they build accurate digital replicas of our handling station and develop predictive maintenance models.

In the early stages, we relied on Microsoft Azure’s IoT services alongside Power BI for data ingestion and visualization. Although feature-rich, this combination proved both inefficient in terms of performance and costly to operate at scale.

Subsequently, we transitioned to Google Cloud—stripping out its IoT-specific offerings—to provide a streamlined environment focused on data scientists’ needs and machine-learning toolchains. By delegating raw data handling elsewhere, our Google Cloud setup became a dedicated analytics and model-training platform.

For lightweight historical record-keeping, we employed Google Sheets to log key metrics such as per-cycle swing times and overall process durations. Its ubiquity and ease of use made it a convenient stopgap for initial data archiving.

Today, we leverage InfluxDB as our primary time-series database. Its protocol-agnostic flexibility allows us to ingest any data type, perform local computations, and integrate seamlessly with Node-RED for downstream visualization and analytics—delivering a performant, cost-effective backbone for both live monitoring and retrospective analysis.

4.5 Cybersecurity measures

All communications occur over a secured local network with strictly limited access, enforced via VLAN segmentation and firewall rules that permit only registered device IPs and service ports. Access control lists restrict traffic flow such that only the Node-RED middleware, the EMQX.io broker, and authorized PLC endpoints can exchange messages, preventing lateral

movement by unauthorized actors.

Sensor and PLC data are transmitted via MQTT over port 8883, where TLS v1.2+ encryption protects every payload in transit. Each ESP32 client and PLC interface must authenticate using a unique username and strong password—stored in a hardware-backed keystore on-device—before any PUBLISH or SUBSCRIBE operation is allowed. MQTT topics themselves are guarded by granular ACLs: for example, “sensors/+/vibration” may be readable by the monitoring dashboard but not writable, whereas “actuators/+/setpoint” is writable only by privileged control consoles.

For OPC UA integration, mutual X.509 certificate authentication is enforced end-to-end. The server (PLC) presents a device certificate signed by our internal Certificate Authority, which the client validates against a trusted root store. Likewise, each client holds its own certificate and private key, issued with limited lifetimes and subject to periodic rotation. OPC UA endpoints are bound to specific network interfaces and ports, and all communications require encryption via AES-256 GCM with RSA-2048 key exchange.

In the cloud tier, our EMQX.io broker issues per-client certificates and private keys during device provisioning. These certificates are tied to unique device identifiers and stored securely in EMQX’s credential vault. Mutual TLS ensures that only devices presenting a valid certificate—along with correct login credentials—can establish a session. The broker’s certificate chain is regularly audited, and certificate revocation lists are propagated automatically to both edge and cloud nodes to revoke access for decommissioned or compromised devices.

Finally, all authentication events and connection attempts are logged centrally, with intrusion-detection rules alerting on unexpected login failures or certificate errors. Automated scripts enforce quarterly rotation of ACLs, passwords, and certificates, ensuring that our entire IIoT communication fabric remains resilient against evolving cybersecurity threats.

Chapter 5: Results(Zhalgas)

The acquisition of both sensor and PLC data was completed successfully using the Siemens S7 and MQTT protocols. PLC tag values—such as cycle start and end timestamps for the handling and processing station—were retrieved reliably over the S7 interface, while temperature, vibration, and liquid-leak measurements from the ESP32-connected Arduino sensors were published via MQTT.

All defined cybersecurity measures were fully implemented. MQTT communication was confined to port 8883 with TLS encryption, client authentication via unique username/password credentials, and tightly scoped ACLs. The Node-RED middleware processed and routed all data streams without error, demonstrating robust protocol translation and logic handling under the secured network policies.

Time-series data from the PLC—specifically, elapsed times for key process steps—were ingested into an on-premises InfluxDB instance protected by network segmentation and authorization controls. Concurrently, sensor payloads from the ESP32 microcontrollers were forwarded to the Google Cloud environment, where password-authenticated endpoints ensured confidentiality during transit and storage.

A web-based HMI was successfully developed to visualize real-time and historical metrics, providing operators with immediate insight into station performance. All raw and aggregated datasets, including detailed time-stamp logs and sensor records, have been archived and will be submitted alongside this report for further review and analysis.

Chapter 6: Conclusion and Future Work

In conclusion, this project has successfully achieved its primary objectives by establishing a robust data-collection framework, deploying Arduino-based temperature, vibration, and liquid-leak sensors at critical points on both the handling and processing stations, and capturing PLC tag values via the Siemens S7 protocol. All sensor and PLC streams were ingested through Node-RED, secured end-to-end with TLS encryption on MQTT port 8883, password-protected authentication, OPC UA mutual certificates, and strict local-network access controls. Time-series records of cycle durations and sensor measurements were reliably stored in an on-premises InfluxDB instance—safeguarded by VLAN segmentation and authorization rules—and selectively forwarded to Google Cloud for broader analytics needs. The real-time HMI interface now provides operators with immediate visibility into station performance, while archived datasets support both manual inspection and downstream machine-learning workflows.

Looking ahead, our focus will shift toward realizing a full digital-twin solution underpinned by these live and historical data streams. By collaborating with data-science teams, we will develop predictive-maintenance models that leverage actual sensor and PLC measurements, eliminating reliance on synthetic inputs. Stages of this work include (1) curating and labeling the archived dataset, (2) training anomaly-detection and remaining-useful-life algorithms, (3) validating model outputs against observed station behavior, and (4) integrating inference nodes directly into the Node-RED flows for on-edge alerts. This progression—from raw data acquisition to real-time model deployment—will close the loop between operations and analytics.

Future enhancements will also encompass expanded sensor coverage (e.g., acoustic or current-draw monitoring), automated certificate rotation and ACL management to further strengthen cybersecurity, and exploratory implementation of edge-compute nodes for localized preprocessing. We will refine retention and downsampling policies in InfluxDB, migrate long-term archival from Google Sheets into managed cloud databases, and expose standardized APIs for third-party dashboards. Collectively, these efforts will evolve our prototype into a scalable, secure, and intelligent digital-twin platform capable of delivering actionable insights and prescriptive maintenance recommendations.

Bibliography

- [1] S. Boschert and R. Rosen, “Digital twin—the simulation aspect,” in *Mechatronic Futures*, pp. 59–74, Springer, 2016.
- [2] M. Grieves, “Digital twin: Manufacturing excellence through virtual factory replication.” White Paper, 2014. Available: <https://www.researchgate.net/publication/307509727>.
- [3] Z. Liu, X. Zhang, and Y. Liu, “Data-driven hybrid method for remaining useful life prediction with limited sensor data,” *Reliability Engineering & System Safety*, vol. 193, p. 106675, 2020.
- [4] F. Tao, M. Zhang, Y. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.
- [5] L. Wang, M. Törngren, and M. Onori, “Current status and advancement of cyber-physical systems in manufacturing,” *Journal of Manufacturing Systems*, vol. 56, pp. 11–25, 2020.
- [6] C. Zhang, Z. Liu, and D. Wu, “Deep learning-based anomaly detection for mechanical systems: A survey,” *Sensors*, vol. 21, no. 3, p. 972, 2021.
- [7] T. Becker, H. Stern, and D. Witsch, “A concept for automated production data acquisition in cyber-physical production systems,” in *Procedia CIRP*, vol. 72, pp. 396–401, 2018.
- [8] A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas, “Intelligent predictive maintenance for customer support in manufacturing using big data and iot: The icam framework,” *Computers in Industry*, vol. 111, pp. 59–75, 2019.
- [9] C. Brecher, M. Emonts, and G. Schuh, “Smart factory web: A semantic web platform for plug-and-produce,” in *Digital Transformation of the Design, Construction and Management Processes of the Built Environment*, pp. 295–306, Springer, 2021.
- [10] J. Lee, E. Lapira, B. Bagheri, and H.-A. Kao, “Recent advances and trends in predictive manufacturing systems in big data environment,” *Manufacturing Letters*, vol. 1, no. 1, pp. 38–41, 2013.
- [11] PHM Society, “Phm data challenge.” Accessed: 2025-05-05, 2025. <https://www.phmsociety.org/events/data-challenges>.
- [12] Fraunhofer IOSB, “Smart factory web.” Accessed: 2025-05-05, 2025. <https://www.smartfactoryweb.de/>.
- [13]