Synthesis of an analytical system prototype using MQTT and AWS in the Industry 4.0 context

Denis Gaiduchek[1,*,†], Liudmyla Kryvoplias-Volodina[1,†], Volodymyr Kostin[1,†], Zinaida Burova[2,†], Oleksandr Zaporozhets[1,†]

[1] *National University of Food Technologies, 68, Volodymyrska str., 01033 Kyiv, Ukraine*
[2] *National University of Life and Environmental Sciences of Ukraine, 15, Heroyiv Oborony str., 03041, Kyiv, Ukraine*

Abstract

The article presents the results of architectural design as well as end-to-end prototype implementation of the system for receiving, transmitting and visualizing technological information for food packaging machines within the Industry 4.0 concept. The focus is on building an intelligent architecture enabling automated processing of production data, which is an important step in increasing production efficiency, reducing operating overheads and increasing the reliability of modern packaging lines. The creation of a prototype serves as a basis for further integration with pneumatic, thermal and other equipment used to obtain data, as well as with MATHLAB / Simulink modeling systems to optimize production processes.

Developed prototype of the automated system covers the full cycle of data processing: collecting data from programmable logic controllers (PLC), transmitting them to the cloud environment via encrypted communication channels and further visualizing them in a web interface. The system architecture is based on the use of the Siemens SIMATIC S7-1200 PLC, which performs the initial collecting of sensor parameters. The collected data is transmitted to the Amazon Web Services (AWS) cloud environment via the MQTT protocol, considering the requirements for security and reliability.

Article examines in detail all components of the built system, including the PLC configuration, methods for communication, and data storage in DynamoDB. Particular attention is focused on minimizing operating costs for supporting the cloud infrastructure and interaction between system components, which allows achieving high adaptability and flexibility of the system at scale.

Developed system is an example of effective integration of production equipment with digital services, demonstrating the practical implementation of the principles of Industry 4.0 in packaging equipment in food industry.

Keywords

Industry 4.0. Node-RED, MQTT, Modbus, AWS IoT Core

1. Introduction

Over the past decade, there has been an intensive growth of interest in the Industrial Internet of Things (IIoT) technologies, which is relevant for enterprises involved in food industry in operation of packaging machines and units [1]. In the context of digital transformation in production processes, many companies are planning or already implementing projects aimed to integrate physical devices into a single digital environment to ensure continuous data collection, transmission and analysis to increase production efficiency, reduce operating costs, implement preventive maintenance mechanisms and ensure flexibility of using data in real time.

To enable accurate and automated data transmission of data the system should be designed to support real-time monitoring and execution of measurement algorithms, metrological validation methods, which can be integrated into hardware-software systems for smart manufacturing, improving quality and energy efficiency across industrial processes [2]. Additionally, mathematical

1*ITTAP'2025: 5th International Workshop on Information Technologies: Theoretical and Applied Problems, October 22-24, 2025, Ternopil, Ukraine, Opole, Poland*

* Corresponding author.
† These authors contributed equally.
gayduchekdv@nuft.edu.ua (D. Gaiduchek); kryvopliasvolodina@camozzi.ua (L. Kryvoplias-Volodina); vkostin1951@gmail.com (V. Kostin); zinaburova@nubip.edu.ua (Z. Burova); zaporozhecov@nuft.edu.ua (O. Zaporozhets)
0009-0009-2780-3827 (D. Gaiduchek); 0000-0001-9906-6381 (L. Kryvoplias-Volodina); 0009-0006-6040-2476 (V. Kostin) 0000-0002-4712-6298 (Z. Burova); 0009-0007-5644-2819 (O. Zaporozhets)

and physical model simulation shall be supported to ensure the calculation and construction of industrial equipment [3,4].

However, practical implementation of the IIoT concept requires solving a set of complex technical and organizational challenges, among which the key one is the problem of ensuring reliable, secure and standardized data exchange between initial data collection level (sensors, programmable logic controllers, SCADA systems) and enterprise's information systems or cloud analytical platforms [4].

Effectively overcoming these challenges is critical for implementation of Industry 4.0 scenarios and synthesis of principles of intelligent manufacturing [5]. Successful integration of physical and digital components in production creates prerequisites for building adaptive, self-managed systems that can quickly respond to changes in production processes and make effective decisions based on real-time data [6].

2. Related Works

IIoT architectures for packaging systems integrate machines, sensors, and software platforms to increase throughput, reduce downtime, and product defects, which is in line with Industry 4.0 principles [7,8]. A typical IIoT architecture is shown in Fig. 1 and typically consists of the following layers.

Physical Layer. This layer directly interfaces with packaging equipment and production environment. Key components include:

- Sensors embedded in packaging equipment to monitor variables such as seal temperature, fill level, pressure, cycle time, and material availability.
- Actuators to control motion, pressure, cutting, or sealing mechanisms.

Data Acquisition and Communication Layer. This layer provides reliable and secure communication between packaging line equipment and central systems. It includes:

- PLCs (e.g., Siemens S7-1200) to control machine operations and record process parameters in real time.
- Edge gateways that perform initial data aggregation, normalization, and data preprocessing.

Data Processing Layer (Cloud Computing). At this layer, data is collected for centralized analysis and storage:

- Cloud platforms such as AWS hosted data sets and analytics services.
- DynamoDB or other scalable databases are used to store data.

Application Layer. This is the interface for operators, engineers, and analysts. Applications can include:

- Real-time dashboards that display performance metrics such as overall equipment efficiency, cycle time, defect rates, and machine health.

- Predictive maintenance alerts that identify wear patterns in motors, belts, or sealing elements.
- Lot tracking and quality assurance reports based on packaging parameters and production logs.
- Integration with ERP and PMS (product management systems) to plan production, drive new product development, and update inventory in a timely manner.

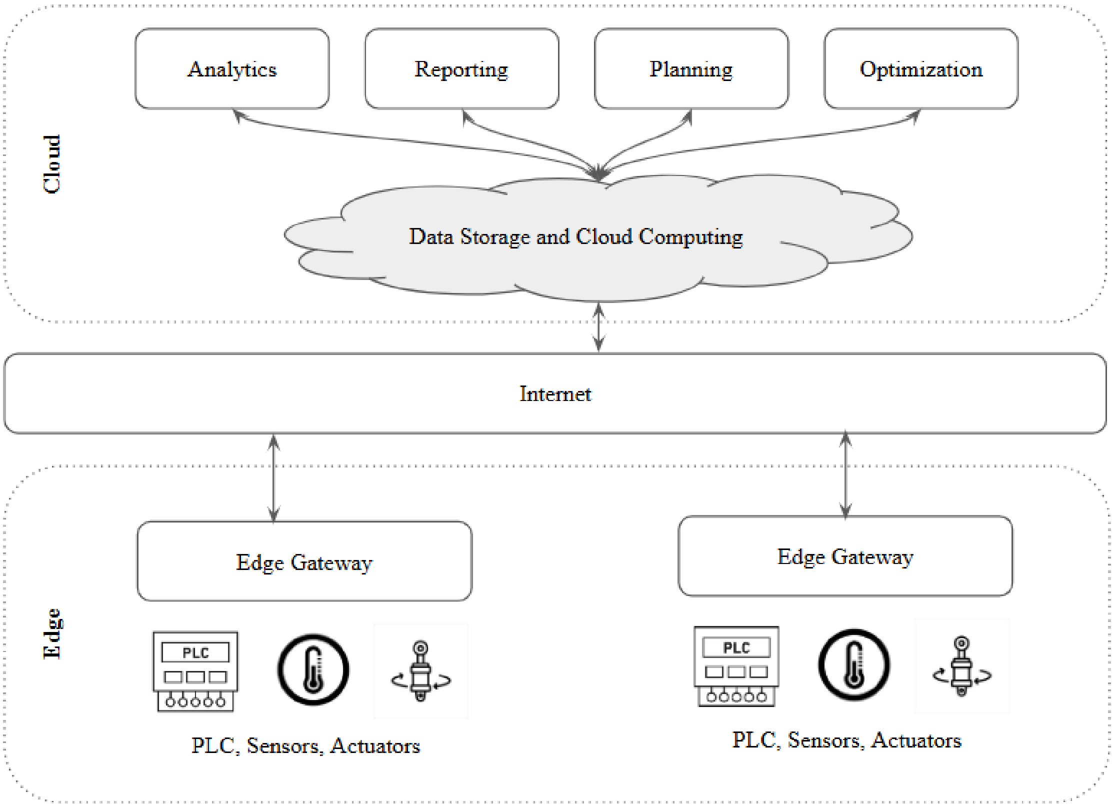

Figure 1: Typical IIoT system architecture

3. Proposed model

High-level architecture of the analytical system prototype

The diagram in Fig. 2 illustrates architecture of the developed analytical system prototype, which integrates industrial PLCs, edge computing and cloud services.

At physical level of the system there are industrial sensors that are connected to the Siemens Simatic S1200 PLC [8,9], configured as Modbus slaves. The sensors are responsible for measuring the physical parameters of the technological process (for example, temperature, pressure, material availability, etc.).

Simatic S1200 controller is connected to sensors directly, without a separate communication module. The PLC configuration is performed using the TIA Portal environment - an integrated engineering platform from Siemens.

Data transfer from PLC to edge computing level is carried out using the Modbus TCP protocol, which is a commonly used standard in industrial communications.

Node-RED gateway [10,11], which is installed on the Raspberry Pi [12], acts as a Modbus master and serves as an intermediate computing element of the system [13]. This gateway polls the PLC and performs basic data processing or formatting. Node-RED is a low-code environment that allows to quickly deploy and integrate industrial data streams. In addition, the gateway acts as a component for transmitting to the cloud infrastructure using resource-efficient protocols.

The processed data is transmitted from gateway to cloud using the MQTT protocol [14,15], a protocol built on a publish/subscribe architecture that is ideal for IIoT systems due to its low traffic consumption.

The data is received by AWS IoT Core [16], an Amazon managed service that provides secure communication between connected devices and cloud services. Data is stored in DynamoDB [17]. Collected and processed data is then visualized in a web application, implemented using software based on the React framework for the client side (frontend) and the Node.js runtime for server logic (backend). This application provides interactive graphical interface for viewing, analyzing, and interacting with data in real time.

Users can access the application from any device — smartphone, tablet, or desktop computer, which provides flexible remote monitoring and decision-making based on current production data.

This architecture demonstrates a scalable and modular IIoT system that combines industrial equipment (e.g. Siemens PLCs), edge computing layer (via Raspberry Pi and Node-RED), cloud services (AWS IoT Core), and user-friendly user interface.

Deployed Node.js backend web application acts as analytical module within the designed IIoT architecture. It provides lightweight scalable environment for processing sensor data stored in DynamoDB database.

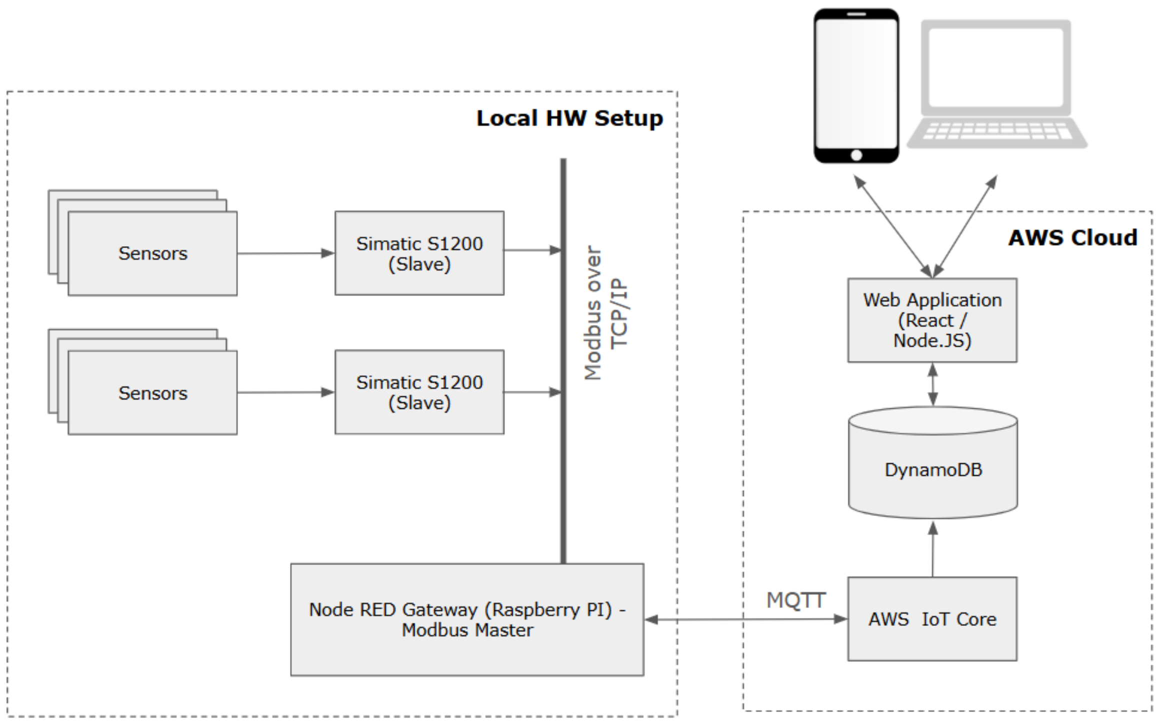

Figure 2: Architecture of the implemented analytical system prototype

Rationale for choosing the Modbus protocol

Despite the rapid development of alternative industrial protocols, Modbus TCP remains effective solution, holding a significant share of the market. Modbus is one of the oldest and most widely used protocols for two reasons. First, it is an open protocol, which has led to widespread adoption in manufacturing. Many companies offer Modbus TCP-compatible devices, software libraries and tools that provide support when creating and managing industrial automation systems. Second, it is a standard protocol for many existing devices, which enables wide range of possibilities when designing systems based on legacy or existing equipment and devices.

Compared to other industrial protocols, Modbus TCP is quite lightweight and easy to use. For this reason, Modbus TCP is widely used and integrated into various types of devices from different vendors. Modbus TCP operates over standard Ethernet networks and uses standard TCP/IP, which makes it compatible with most Ethernet equipment and infrastructure.

Node-RED as an Edge Level Gateway in IIoT Architecture

In modern IIoT architectures, an important component is the Edge Level Gateway, which provides local data processing, reduces the load on cloud services, and reduces delays in information transmission. One of the tools that has gained widespread use at this level is Node-RED, a visual data flow development environment focused on the integration of IoT components [10].

Node-RED is distinguished by a low entry threshold for developers due to use of a graphical interface based on visual programming model. This approach allows to create integration solutions without the need for in-depth knowledge of programming languages, which is relevant in the context of rapid development and prototyping in engineering and manufacturing environments.

Tool supports a wide range of industrial automation protocols, including Modbus TCP, MQTT, OPC-UA, HTTP, etc. This ensures compatibility of Node-RED with physical devices (sensors, controllers) and cloud platforms (e.g. AWS IoT Core, Microsoft Azure IoT Hub), making it effective middleware in heterogeneous data transfer environment.

It is important to highlight low hardware requirements that allow Node-RED to be deployed on embedded devices such as Raspberry Pi or other ARM platforms. This allows to reduce the cost of implementing edge infrastructure, which is a critical factor for small and medium-sized businesses.

Node-RED also provides high extensibility through a system of open libraries, which enables the possibility to implement pre-processing, local analytics, machine learning, storage and interaction with REST APIs.

Thus, using Node-RED as a gateway at the edge computing level in IIoT systems is a viable technical solution that combines functional flexibility, low implementation costs and compatibility with modern industrial data transfer standards.

Cloud environment

In the context of the IIoT, cloud environments are a key infrastructure layer that ensures scalability, availability, and integration of distributed industrial systems. Enterprises can collect data from a large number of sensors and controllers in centralized way, process it in real time, apply analytics and machine learning algorithms for predictive maintenance, process optimization, and ensure continuous monitoring.

Cloud services provide ready-made tools for connecting devices, managing digital twins, routing messages, and integrating with ERP and MES systems. Flexible scaling solutions and high level of security, cloud environments allow implementing full-fledged IIoT architectures even in complex industrial environments with large number of nodes and unstable networks.

Amazon Web Services (AWS) is the most common cloud platform for IIoT solutions. The platform is actively used in industrial systems, including food industry, energy, and logistics. AWS offers a wide range of software tools for developers and official documentation for almost all major programming languages and embedded system architectures. In addition, open protocols (MQTT, HTTPS) are supported, which greatly simplifies a wide range of devices. To perform analytical calculations, run test services and process data in real time, the Node.JS backend web application is deployed to the EC2 t3.micro component [18] with the following parameters - 2 vCPUs, 1 GB of RAM, SSD EBS storage. The cost of use within AWS Free Tier is 750 hours of use per month for the first 12 months from the moment of account registration. Outside the Free Tier, the cost is about

$0.0104/hour (≈ $7.50/month, taking into account continuous operation. This component is sufficient for low-complexity computing processes, deploying test servers, and basic data flow management without significant financial costs.

AWS IoT Core is used as a message broker for exchanging data using the MQTT protocol between the edge gateway and other cloud services. It provides support for MQTT 5.0 protocol, connects via TLS 1.2, and has built-in authentication and authorization mechanisms. The cost within the AWS Free Tier includes 2.25 million message publications per month for 12 months, and $1.00 - $1.25 per 1 million messages in the Free Tier package. This allows to avoid the costs of the own broker infrastructure, providing scalability and security without additional administration.

MQTT protocol integration

MQTT [14] is a lightweight protocol based on the publish/subscribe concept designed for networks with limited bandwidth, high latency, or unreliable connections. The main technical features of MQTT include:

- Minimal network load: due to the small message size.
- Asynchronous data exchange model: clients publish or subscribe to messages without direct interaction with each other.
- QoS (Quality of Service) levels: provide flexible management of message delivery reliability.
- Support for low-power connections: relevant for wireless sensors and battery-powered devices
- Support for TLS/SSL: provides secure data exchange over the Internet.

Modern packaging lines implement a large number of data collection points: strain gauge sensors, vibration sensors, actuators, and controllers. MQTT allows for flexible integration of these components into the overall IIoT architecture. Due to flexibility of scaling new nodes can be easily added to the system without changing overall structure, due to minimal delays, MQTT broker provides instant message routing, which is critical for adaptive control of packaging processes; support by cloud solution providers, as MQTT easily integrates with modern cloud platforms including AWS IoT, Azure IoT, HiveMQ.

Among practical advantages of integrating MQTT protocol in packaging industry, the following factors should be considered. Reducing the total amount of downtime in the architecture due to real-time data exchange via MQTT, it is possible to quickly detect deviations in the operation of mechanisms (for example, excessive vibration, overheating, film shortage). This allows to move from reactive to preventive maintenance. MQTT provides a stable transmission channel for images and quality control sensor readings, which allows accurate calibration of equipment and reduction of defective products, resulting in improved product quality. In multi-section packaging lines, MQTT facilitates synchronization between packaging, printing, labeling, and palletizing stations. Considering mentioned above aspects - optimal data volume, ease of integration, scalability, and QoS - the MQTT protocol is the optimal choice for use in IIoT packaging equipment systems. It enables reliable interaction between devices and data collection and processing platforms even in complex production environments, where response time, stability, and network exchange efficiency play a key role.

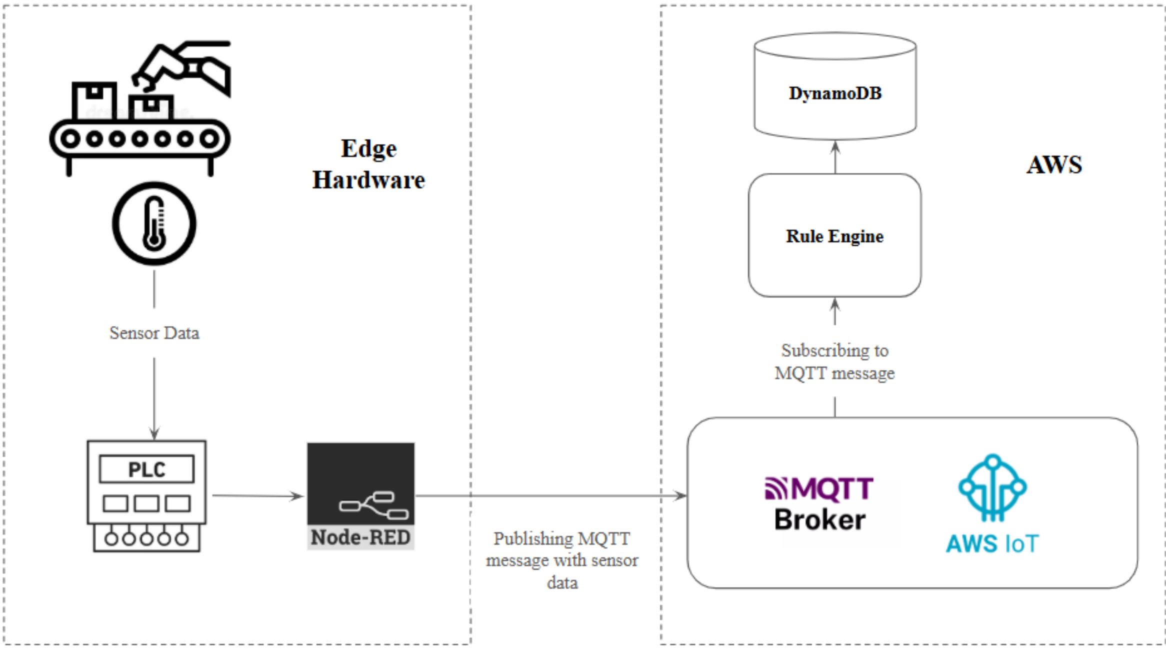

Figure 3: MQTT protocol integration

Figure 3 demonstrates practical integration of MQTT protocol, specially designed for devices with limited resources and unstable network connections. It is based on the "publisher-broker-subscriber" model: The publisher sends messages to a specific topic, the MQTT broker receives these messages and sends them to all subscribers who are subscribed to the corresponding topic; Subscribers receive only those messages that correspond to their interests (topics). In the designed system Node-RED gateway serves as publisher of MQTT message with sensor data while AWS Rule Engine acts as a subscriber.

DynamoDB in IIoT Architecture

DynamoDB [17] is a serverless NoSQL database that automatically scales with workload. In the context of IIoT systems, amount of data from sensors transmitted via MQTT to AWS IoT Core can be very large and uneven throughout the day or production cycle. DynamoDB automatically adapts to workload volumes without the need for database administration. Almost 100% availability of the service in real-time across AWS regions is guaranteed.

Since the system uses AWS IoT Core to process MQTT messages, it is important that the database provides low latency for storing and reading data. DynamoDB can process millions of queries per second with latency down to milliseconds, which is critical for real-time monitoring of parameters and displaying the current state of equipment.

DynamoDB is tightly integrated with other AWS services. For example, data from AWS IoT Core is transmitted directly to DynamoDB via the IoT Rules Engine. Since MQTT messages can be different for different types of sensors (temperature, pressure, vibration, etc.), DynamoDB as a NoSQL database allows storing data with a flexible structure where a fixed table schema is not required. This is especially convenient when changing the sensor configuration or adding new types.

AWS Free Tier provides 25 GB of storage, 2.5 million requests per month for 12 months. Outside the Free Tier, payment is made for the used read/write units and storage volume (≈ $0.25/GB per month). Provides reliable real-time data storage with the ability to scale horizontally without changing the architecture.

Equipment and Components

Table 1 lists the peripheral equipment, software, and infrastructure components used to create the prototype, and lists their main purpose and basic characteristics.

Table 1
Peripheral equipment, software, and infrastructure components

Name	Purpose and characteristics
PLC SIMATIC S7-1200 CPU 1214C DC/DC/DC 6E27 214-1AE30-0X80	The PLC is equipped with 14 discrete inputs 24 VDC, 10 discrete outputs 24 VDC (0.5 A) and 2 analog inputs 10-bit 0–10 VDC
SIEMENS DC-24V / 5A 6EP1333-2BA20	The power supply provides a stabilized 24 VDC voltage with an output current of up to 5 A for operating industrial equipment
TIA Portal V18	Siemens software environment for designing, programming and configuring automated systems and PLCs.
Asus RT-AC58U Dual Band WiFi Router	Dual-band AC1300 Wi-Fi router provides speeds of up to 867 Mbps on 5 GHz and up to 400 Mbps on 2.4 GHz for Internet access
Raspberry PI 4 Model 8	Used as a peripheral gateway - a compact and energy-efficient solution for data processing and deployment of the Node-RED environment.
Node-RED version 4.0.9	Visual programming environment for creating data processing flows, automation, and IoT device integration
Node.js version 22.14.0	JavaScript platform for server-side code execution
Amazon Web Services account	Provides access to cloud services to deploy components for collecting, processing, and analyzing data from industrial devices and systems.

PLC configuration

SIMATIC S7-1200 CPU 1214C DC/DC/DC [19] was chosen due to the combination of built-in discrete inputs, which allow sensors and switches to be connected without additional modules, and the integrated Ethernet network interface providing fast data exchange with other devices and systems, simplifying integration into the existing automation infrastructure.

The PLC configuration is performed in the TIA Portal environment, which provides convenient interface for configuration and integration of the equipment.

The PLC model used in prototype does not have built-in support for Modbus server functions, so to ensure data exchange using this protocol, it is necessary to create a corresponding software block in the TIA Portal environment. This enables PLC to operate as a Modbus TCP server, specify the required communication parameters, configure the address space, and ensure interaction with client devices operating using the Modbus protocol (Figure 4).

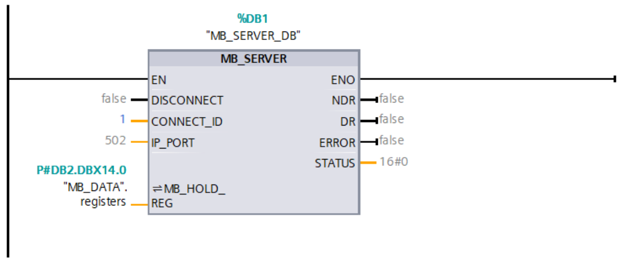

Figure 4: Embedded Modbus Server software block in the TIA portal environment

Implementing a Node-RED-based edge gateway

The Node-RED based peripheral gateway installed on Raspberry Pi 4 Model B (8GB) is a compact peripheral data processing node that collects telemetry from sensors/PLCs, performs local data filtering and aggregation, manages events (alerts, rules), and then securely transmits data to the cloud environment via the MQTT protocol. With a quad-core ARM Cortex-A72 (≈1.5 GHz), 8 GB of RAM, and Gigabit Ethernet, the RPi 4 provides a balance of computing power, I/O, and network interfaces in a small package. Node-RED provides visual flow environment with ready-to-use nodes for transmitting data from a Modbus server via the MQTT protocol (Figure 5). A practical implementation diagram of the primary data processing received from the PLC via the Modbus TCP protocol, generation of MQTT messages in JSON format, and transmission of the message to AWS IoT Core is presented.

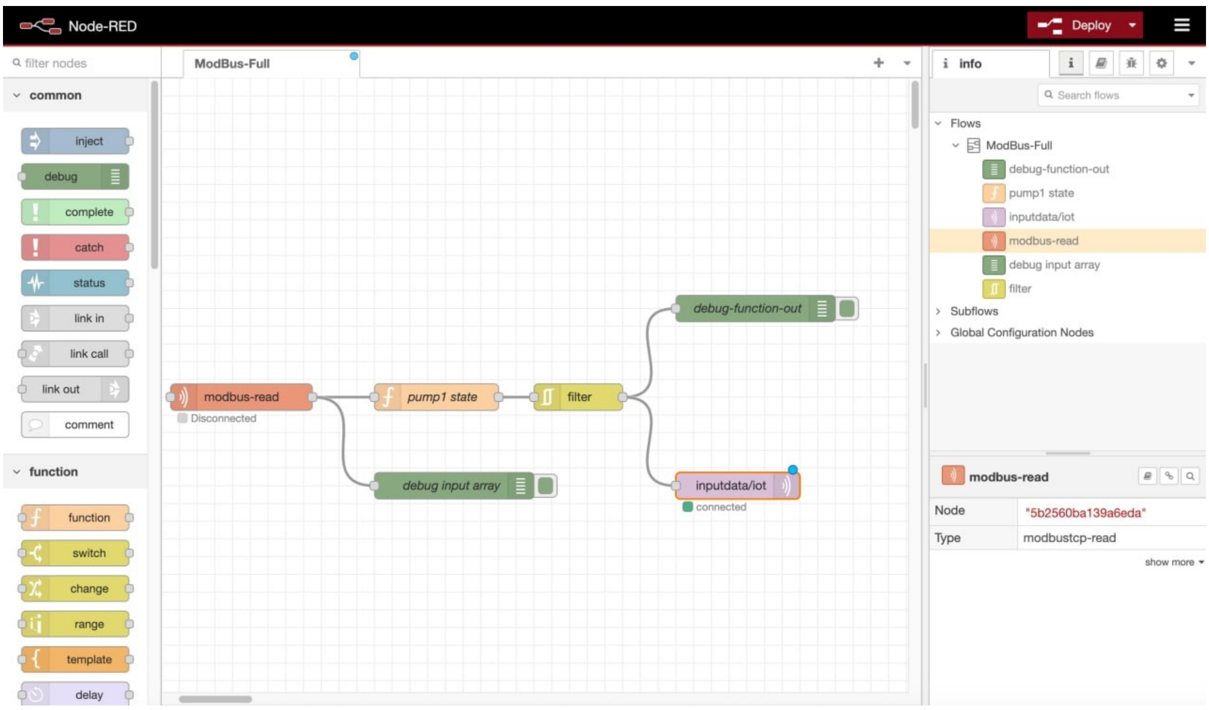

Figure 5: Data transfer scheme from PLC to AWS IoT Core via MQTT protocol

The main components are shown on Figure 5. include:

- Module modbus-read, which provides data reading via the Modbus protocol from the PLC. When configuring the module, it is necessary to determine the settings of the server from which data is read, specifying its network address and TCP port (default 502), the data reading method (Read Holding Registers) and the polling interval. The component is not included in the standard Node-RED components and requires additional installation.

- Module for sending MQTT messages to the cloud environment, including encryption algorithms. For the module to work correctly, it is necessary to specify the required MQTT topic (inputdata/iot) and configure the MQTT broker deployed in the AWS cloud environment.

- Function and filtering modules perform additional data processing. For example, to reduce the data flow for storage in cloud environment, the MQTT message is sent only if the state of discrete sensor has changed.

Data storage

When a device or gateway publishes a message in JSON format via the MQTT protocol to a specified topic, AWS IoT Core receives it through its MQTT broker. Then, a pre-configured IoT rule (Rule) is triggered, which contains a query to select the required fields from the incoming JSON message and a specified action (Action) — a record in Amazon DynamoDB. After the rule is triggered, AWS IoT Core calls an API request to DynamoDB, creating a new item or updating an existing one in the specified table. As a result, messages from the device are stored in DynamoDB as a structured record, ready for further analysis or processing in the cloud. Figure 8 illustrates a fragment of a DynamoDB table using the AWS console.

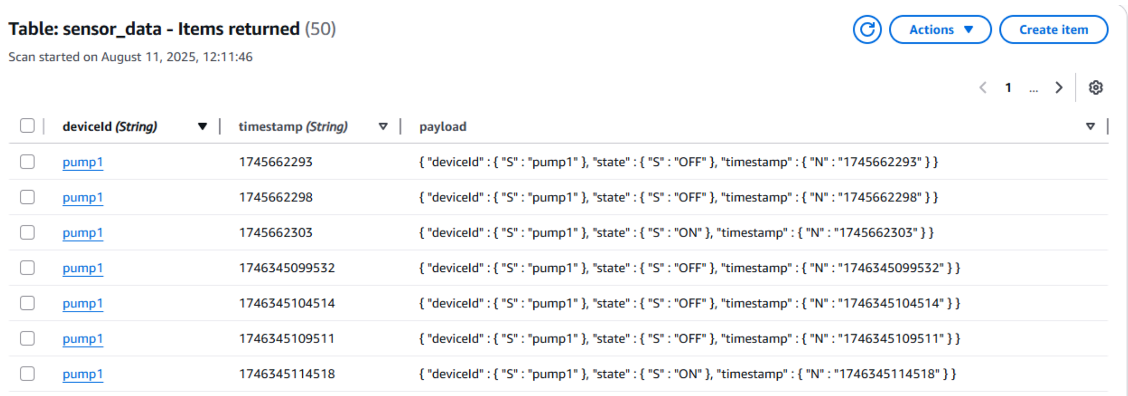
Table: sensor_data - Items returned (50)
Scan started on August 11, 2025, 12:11:46

deviceId (String)	timestamp (String)	payload
pump1	1745662293	{ "deviceId" : { "S" : "pump1" }, "state" : { "S" : "OFF" }, "timestamp" : { "N" : "1745662293" } }
pump1	1745662298	{ "deviceId" : { "S" : "pump1" }, "state" : { "S" : "OFF" }, "timestamp" : { "N" : "1745662298" } }
pump1	1745662303	{ "deviceId" : { "S" : "pump1" }, "state" : { "S" : "ON" }, "timestamp" : { "N" : "1745662303" } }
pump1	1746345099532	{ "deviceId" : { "S" : "pump1" }, "state" : { "S" : "OFF" }, "timestamp" : { "N" : "1746345099532" } }
pump1	1746345104514	{ "deviceId" : { "S" : "pump1" }, "state" : { "S" : "OFF" }, "timestamp" : { "N" : "1746345104514" } }
pump1	1746345109511	{ "deviceId" : { "S" : "pump1" }, "state" : { "S" : "OFF" }, "timestamp" : { "N" : "1746345109511" } }
pump1	1746345114518	{ "deviceId" : { "S" : "pump1" }, "state" : { "S" : "ON" }, "timestamp" : { "N" : "1746345114518" } }

Figure 6: Fragment of DynamoDB table

A detailed description and configuration of components deployed in the cloud environment is beyond the scope of this article. This work provides a general overview of the services used with an explanation of their practical feasibility in the system architecture. During the design, special attention was focused on optimizing the cost of cloud infrastructure by prioritizing the use of free tariffs (Free Tier) and services with no subscription fee under moderate load conditions [18]. This approach allowed minimizing operating costs of supporting the prototype, while maintaining functionality necessary for transmission, primary processing and visualization of technological information in real time. The selection of components was carried out considering the "cost-performance" ratio, which ensures the possibility of further scaling the system without a significant increase in financial costs.

3. Results

As a result of the research, a prototype of the analytical system for food packaging machines was designed and implemented within the Industry 4.0 concept. The proposed architecture provides a full cycle of processing technological data - from collecting data from the Siemens SIMATIC S7-1200 PLC via the Modbus TCP protocol, their initial processing on the edge gateway based on Raspberry Pi running Node-RED, to transmission to AWS cloud environment using the MQTT protocol and subsequent visualization in the web interface.

The use of Node-RED as an edge gateway has proven its effectiveness due to low requirements for hardware resources, flexible integration and support for a wide range of industrial protocols. MQTT protocol turned out to be the optimal choice for this scenario due to its lightweight nature, support for QoS levels and the ability to work in networks with limited bandwidth.

AWS-based cloud infrastructure, including IoT Core, DynamoDB, and EC2 t3.micro services, provided scalability, reliability, and security of the system at minimal costs.

5. Conclusions

The designed system architecture and developed end-to-end prototype serves as a solid baseline for further system evolving and core functionality extension. The proposed approach is suitable for integration of the IIoT scenarios and applicable for variety of domains including packaging equipment capable to reduce operating costs at the development and testing stage, while maintaining the ability to scale system for industrial volumes.

The proposed solution demonstrates practical implementation of Industry 4.0 principles in the field of packaging equipment, contributes to increasing production efficiency, and also creates a basis for the implementation of adaptive mechanisms, preventive maintenance, and advanced real-time analytics.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] F. J. Folgado, D. Calderón, I. González, and A. J. Calderón, “Review of industry 4.0 from the perspective of automation and supervision systems: Definitions, architectures and recent trends,” (2024), Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/electronics13040782.

[2] V. Babak, O. Dekusha, Z. Burova, Hardware–software system for measuring thermophysical characteristics of the materials and products, CEUR Workshop Proceedings 3039 (2021) 255–266. URL: https://www.scopus.com/pages/publications/85121266020.

[3] L. Kryvoplias-Volodina, O. Gavva, M. Yakymchuk, A. Drenivska, T. Hnativ, H. Valiulin, Practical aspects in modeling the air conveying modes of small–piece food products, Eastern-European Journal of Enterprise Technologies 5 (11–107) (2020) 6–15. doi:10.15587/1729-4061.2020.213176.

[4] O. Gavva, L. Kryvoplias-Volodina, Methodology of the quantitative approach to the selection of optimal structures of adaptronic functional modules of packaging machines, Processes and Equipment, Food Industry (2023) 116–125. doi:10.24263/2225-2916-2023-33-34-15.

[5] S. H. K. Tin, S. Thu, K. K. Maung, IoT and Industrial Automation: A Review of Current Research and Emerging Trends, FMDB Transactions on Sustainable Technoprise Letters 2 (3) (2024) 151–160. doi:10.69888/FTSTPL.2024.000330.

[6] F. Qiu, A review on integrating IoT, IIoT, and Industry 4.0: A pathway to smart manufacturing, IET Industrial Systems Engineering (2025). doi:10.1049/ise2/9275962.

[7] Hussein, F. A., Yaseen, M. T., & Mustafa, M. O., The Future of Intelligent Industrial Systems: PLC, Node RED, and IoT/IIoT, AUIQ Technical Engineering Science. doi:10.70645/3078-3437.1034.

[8] N. W. Nugraha, F. Suryatini, N. Lilansa, F. A. M. Farhan, Implementation of Industrial IoT Integration Using Node RED and PLC on Cascade Control Level and Flow Plant, Jurnal Penelitian Pendidikan IPA 11 (6) (2025) 191–205. doi:10.29303/jppipa.v11i6.11623.

[9] A. H. Embong, L. Asbollah, S. B. Abdul Hamid, Empowering industrial automation labs with IoT: A case study on real-time monitoring and control of induction motors using Siemens PLC and Node-RED, Journal of Mechanical Engineering and Sciences, 2024. URL: https://journal.ump.edu.my/jmes/article/view/9600.

[10] H. Nugraha, A. D. Hermawan, M. A. J. Mulya, I. Firmansyah, Temperature sensor integration into the Node-RED platform for transformer monitoring, Journal of Physics: Conference Series 2673 (1) (2023). doi:10.1088/1742-6596/2673/1/012037.

[11] Node-RED, Node-RED user guide, 2025. URL: https://nodered.org/docs/.

[12] Raspberry Pi Foundation, Raspberry Pi 4 Model B specifications, 2024. URL: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[13] International Electrotechnical Commission, Industrial communication networks – Fieldbus specifications – Modbus TCP. IEC 61158, 2010.

[14] HiveMQ, MQTT Essentials – The Ultimate Guide to MQTT for Beginners and Experts, 2025. URL: https://www.hivemq.com/mqtt/.

[15] OASIS Open, MQTT version 5.0 specification, 2019. URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.

[16] Amazon Web Services, AWS IoT Core documentation, Amazon, 2025. URL: https://docs.aws.amazon.com/iot/.

[17] Amazon Web Services, Amazon DynamoDB documentation, Amazon, 2025. URL: https://docs.aws.amazon.com/dynamodb/.

[18] Amazon Web Services, Amazon EC2 instance types, Amazon, 2025. URL: https://aws.amazon.com/ec2/instance-types/.

[19] Siemens AG, SIMATIC S7-1200 programmable controller, Siemens, 2022. URL: https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf.