

Article

Advancing Small and Medium-Sized Enterprise Manufacturing: Framework for IoT-Based Data Collection in Industry 4.0 Concept

Martin Barton ^{1,*}, Roman Budjac ^{2,*}, Pavol Tanuska ¹, Ivan Sladek ² and Martin Nemeth ¹

¹ Faculty of Materials Science and Technology, Slovak University of Technology, Jana Bottu 2781/25, 917 24 Trnava, Slovakia; pavol.tanuska@stuba.sk (P.T.); martin.nemeth@stuba.sk (M.N.)

² Research Centre, University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia; ivan.sladek@uniza.sk

* Correspondence: martin.barton@stuba.sk (M.B.); roman.budjac@uniza.sk (R.B.)

Abstract: Currently, industrial engineers are witnessing a rapid development of innovation in modern industry and the integration of critical elements of Industry 4.0 into production processes in order to remain competitive. Such changes are characterized by a large amount of effort and financial resources. To successfully deploy these changes requires not only the purchase of hardware and software but also the training of employees and the modification of the company's organizational structure. The main objective of the article is to propose a framework for the modernization of SMEs to a level close to the Smart Factory by using the necessary attributes of Industry 4.0. The framework design is based on the initial state of a standard SME and consists of the design of fitting new IoT devices for efficient data collection, the design of a data warehouse for storing process data using Hadoop, and the integration of process- and operational-level data into the prepared data warehouse. The resulting design is developed in the form of a methodology and is generalized for use in manufacturing enterprises. The universal design is independent of the initial state of the enterprise.

Keywords: IoT; sensors; microcontrollers; Industry 4.0; SMEs; embedded devices



Citation: Barton, M.; Budjac, R.; Tanuska, P.; Sladek I.; Nemeth M. Advancing Small and Medium-Sized Enterprise Manufacturing: Framework for IoT-Based Data Collection in Industry 4.0 Concept. *Electronics* **2024**, *13*, 2485. <https://doi.org/10.3390/electronics13132485>

Academic Editor: Andrea Bonci

Received: 30 May 2024

Revised: 17 June 2024

Accepted: 24 June 2024

Published: 25 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the current digital era, it is more important than ever to investigate digitization possibilities for manufacturers and find suitable investing possibilities to improve existing processes and deploy new strategies based on the principles of the Industry 4.0 concept. In the contemporary discourse of global economic progression, Industry 4.0, which emphasizes digital interconnectivity, intelligent automation, and data-driven decision making, emerges as a pivotal paradigm shift. Predominantly, discussions have focused on its implications for expansive multinational entities. However, a meticulous examination reveals the Fourth Industrial Revolution's profound and multifaceted impact on small and medium-sized enterprises (SMEs) [1]. Within Industry 4.0, SMEs are positioned not merely as passive recipients but as active participants that can utilize their abilities for strategic advantage. The real-time analytics capability enabled by this industry transformation can provide SMEs with exceptional agility. This vital characteristic allows them to exhibit a skill associated with larger conglomerates, negating potential marginalization. Operational efficiency and cost reduction are critical factors in the effectiveness of an organization. In [2], the authors concluded that the number-one barrier to digitalization identified in the study is the lack of sufficient investment in complementary resources. Increasing this is one of the basic expectations after the implementation of changes, which requires the deployment of Industry 4.0 attributes. It is not confined to mere cost reduction. It encompasses a holistic framework: refined processes, strategic inventory management, and the anticipatory advantages proffered by predictive maintenance. Integrating the Internet of Things (IoT) and advanced analytics further reinforces this operational paradigm, facilitating SMEs in

achieving production optimization, which is crucial for sustainable business trajectories. The inherent adaptability of SMEs means they are well placed to harness the innovations of Industry 4.0. Their lean structures are suited for the easy adoption of new technologies and practices. The interconnected digital environment encourages collaborative efforts, opening doors for partnerships and joint innovation ventures. In the contemporary landscape of global commerce, adopting Industry 4.0 technologies emerges as a fundamental imperative for SMEs [3]. Such integration is predominant not solely for operational enhancement but as a requisite strategy to ensure viability in an increasingly digitized and interconnected market. By assimilating these technological advancements, SMEs can gain enhanced operational efficiency, refined customer-centric analytics, and the requisite scalability. Oversight to engage with this technological evolution might impede SMEs' ability to maintain competitive parity, whereas proactive adoption could facilitate their enduring resilience and growth within the context of the contemporary digital economy [4]. Finally, the proposed research builds continuously on previous results. This manuscript is based on a previous research work where we focused on the identification of necessary attributes for the implementation of the Industry 4.0 concept based on a literature review [5]. This previous study is also a detailed analysis of SMEs in relation to Industry 4.0, including the benefits of SME digitalization. We identified these core technologies as pillars that are foundational in deploying the Industry 4.0 concept in SMEs. Namely, these are mainly the following technologies:

- Internet of Things;
- Big Data;
- Cloud Computing;
- Artificial Intelligence (AI);
- Simulation;
- Cybersecurity.

These technologies are the foundation of the Industry 4.0 concept, although it is optional for all these technologies to be implemented simultaneously. It is not desirable even in certain situations, limited by the company's size. On the contrary, using these technologies alone will not achieve the optimal result and the expected benefits of digitization. As these technologies act synergistically during implementation, it is necessary to use these individual technologies to the benefit of a deeper integration of these technologies into industrial processes. Particularly in the case of small and medium-sized enterprises, these technologies need to be implemented in a systematic and targeted manner. Their integration needs to be staggered over time appropriately, considering financial resources, human resources, the market situation, and other essential aspects that may negatively impact the company's sustainability and development. In particular, SMEs are more sensitive to fluctuations caused by external influences, which can more quickly lead to slowing down or stopping the company's growth. As a result, integrating the Industry 4.0 concept in SMEs is a complex process that needs to be implemented gradually and methodically [6]. On this basis, our aim in the following article is to offer a way of implementing some technologies that are neither financially nor technically demanding. Nonetheless, it still follows all the principles of implementing Industry 4.0 principles and processes for the advancement of the company. This approach is mainly chosen since small and medium-sized enterprises prefer a more cautious iterative approach when implementing new technologies in their production processes. At the same time, the challenge of implementing the pillars of Industry 4.0 must have clear objectives and expected outcomes, as the negative impacts of external circumstances on their business can affect the company more quickly than in the case of corporate companies [7]. The first section will analyze the hardware and software tools used to implement the identified attributes of Industry 4.0, focusing on low-cost solutions for small and medium-sized manufacturing enterprises. Subsequently, in the next section, based on the previous review, we will propose our framework for SME modernization to support SMEs' efforts to digitize their production systems and implement Smart Factory principles concerning their affordability [8]. The article's main contribution

is the proposed design and verification of a framework for the modernization of small and medium-sized enterprises up to the level of Smart Factory in the concept of Industry 4.0. The framework has been deployed and verified on the Agro, Food, and Beverage (AFB) production system. It is an automated production line, controlled by a PLC, but without Industry 4.0 elements. In the initial state, no data collection takes place on the line. The line simulates the production of processing, filling, storing, and recycling granulate and liquid [9].

2. Materials and Methods

This section will analyze the hardware and software tools used to implement the identified attributes of Industry 4.0, focusing on low-cost solutions for small and medium-sized manufacturing enterprises. In the rapidly evolving Industry 4.0 environment, the integration of sensor data is increasingly dependent on systems that manage and analyze data from a number of sensors.

Sensors are the initial part of the Field Level layer in the automation pyramid [10], which is the primary source of data for the other systems above (the control level and then MES, ERP). From the Field Level layer and the control system, the data need to reach the higher layers of the automation pyramid. These sensor data are passed on to the processing and integration layer. These data pass from the sensors and devices through middleware technologies before they are part of the Manufacturing Execution System (MES) system integration and Enterprise Resource Planning (ERP). These technologies facilitate the communication and processing of data from the sensors to the data warehouse. Node-RED, in this case, is part of the middleware layer and a tool for connecting hardware and services. Message Queuing Telemetry Transport (MQTT) protocol is used for reliable communication, provides a reliable way of messaging with low bandwidth, and is suitable for real-time processes.

As a low-cost solution for IoT, microcontroller-based devices are used for various applications, as found in several articles and studies. In the Smart Village project [8], traffic sounds were analyzed using 50 such devices, with an annual cost of about USD 850 for data analysis and processing using the ThingSpeak framework. Microcontroller solutions can also be deployed to monitor gas concentrations and achieve a cost-effective solution for collecting and evaluating such data [11]. Another possible application is creating a weather monitoring station using sensors connected to a microcontroller board, and the cost per station created is minimal in terms of both hardware and operation [12]. Evidence of the widespread use of microcontrollers is also provided by a study [13] using such devices for real-time monitoring and storage of electricity consumption data. The device measured voltage and current on one phase, with planned expansion to three-phase measurement. Microcontroller devices are also used in small and medium-sized manufacturing companies, for instance, to measure multiple dimensions of 3D objects using multiple cameras connected to a Raspberry Pi, presented as a low-cost solution [14]. A similar application has also been deployed in the food industry, using Arduino to collect data affecting the final product's quality to determine the product's correct quality, such as temperature, pH, or weight. The authors argue that if SMEs want to reach Industry 4.0 level, they must deploy similar solutions for sensing and evaluating the manufacturing process [15].

In recent years, microcontrollers and open-source solutions have received increased interest from both research and the commercial community. Microcontrollers are often used in places where commercial solutions have gaps. In the last five years, there has been a growing interest in single-board computers with microcontrollers such as Raspberry Pi and Arduino [16]. However, their usability is limited mainly by the need for programming knowledge, so enterprises often opt mainly for classic commercial and reliable solutions, even at the expense of the gaps that commercial solutions have. In the future, however, there is an expectation of even more interest in microcontrollers and open-source solutions

that can save enterprises money [16]. Microcontrollers are also increasingly receiving attention in the literature, as found in the above study.

For the aforementioned practical applications and ever-increasing interest, we consider microcontroller solutions as sufficiently suitable and reliable solutions for IoT attributes, even with sensor equipment that can be connected to them. The study's authors [17] identified open-source Hadoop and Cloud technology as cost-effective solutions for small and medium-sized enterprises for data storage and evaluation due to the hardware and software deployment costs for processing and analyzing both structured and unstructured data.

In [18], Node-RED was used to integrate data from OMRON PLC devices into the ThingsBoard database. The system was successfully deployed and tested on a water distribution system. The authors consider Node-RED a flexible and easy-to-learn tool up to implementation. In the study [19], Node-RED was used as an integration tool in a Cyber-Physical System in a simulated environment. The system was validated on a production line, where data from production system operations and client behavior could be collected. In the article [20], Node-RED was deployed and tested in a distributed environment with PLCs, single-board Raspberry Pi computers, and solutions based on the ESP8266 microcontroller. The study resulted in the high reliability of such solutions regarding data transfer in the network and recommendations to work with Node-RED in industrial environments. The above results contribute to the decision to utilize Node-RED as an integration framework for working with data from microcontroller-based devices. The paper [21] used Apache MiNiFi to collect data from manufacturing processes and the Apache NiFi for the Extract, Transform, Load (ETL) process in an industrial architecture. The deployment resulted in an architecture that improves final decision making in a manufacturing enterprise based on the collected and evaluated data. The results confirm the suitability of deploying Apache MiNiFi and NiFi in an industrial environment.

Apache MiNiFi, NiFi, and Hadoop [22] were deployed for data collection, evaluation, and prediction in a simulated trucking environment. Apache MiNiFi was deployed on a Raspberry Pi microcomputer, and the entire architecture was focused on speed and weather monitoring and subsequent violation prediction. The architecture worked with real-time data. The work results confirm the suitability of using Apache MiNiFi and NiFi to work with real-time data. The mentioned technologies meet the attributes of Industry 4.0 for SME microcontrollers and IoT sensors, Hadoop—Big Data, Node-RED, and Apache NiFi integration. Based on these technologies, the next chapter will propose a framework for upgrading SMEs to a level close to that of a Smart Factory, including Cloud Computing.

2.1. The Design of a Framework for SME Modernization to a Level Close to That of a Smart Factory

Based on the identified key attributes of Industry 4.0, this chapter will propose a framework for upgrading SMEs to the Smart Factory level by implementing these identified attributes. A Smart Factory is an interconnected network of machines, communication protocols, and computers using advanced technologies such as AI for data analysis, process control, and learning [23]. A Smart Factory extracts data from data sources, analyzes them, and sends requests to smart machines based on the analyzed data [24]. This subsection will outline a detailed step-by-step framework designed to facilitate the digital transformation of SMEs. In designing the framework, it is crucial to prioritize cost-effectiveness, considering that SMEs often lack the extensive capital necessary for adopting complex solutions. This approach ensures that the framework is accessible and feasible for SMEs with limited financial resources. Furthermore, the framework must be deployable independently of the initial state of the manufacturing company and designed with cybersecurity in mind. All systems used must be able to communicate with each other based on communication protocols. The framework must be designed based on the identified attributes of Industry 4.0, namely, IoT, Big Data, horizontal and vertical integration, cybersecurity, and Cloud Computing. The cybersecurity concern was resolved on the AFB line where our data

collection system is to be deployed, as discussed in the article [9]. We decided to omit the simulation attribute in this part of the work to reduce the cost of deploying the framework.

The PLC level is directly connected to the sensors and actuators on the production line. It is also connected via Profinet or Profibus to the Human–Machine Interface (HMI) and can be used to control the production line and, via Open Platform Communication (OPC), to control the higher-level SCADA and MES systems. At the same time, the PLC level is connected to a relational database where available process data from the production are collected. This state will be considered a standard in this article and used as an initial state in the architecture design. The typical initial state diagram is shown in Figure 1.

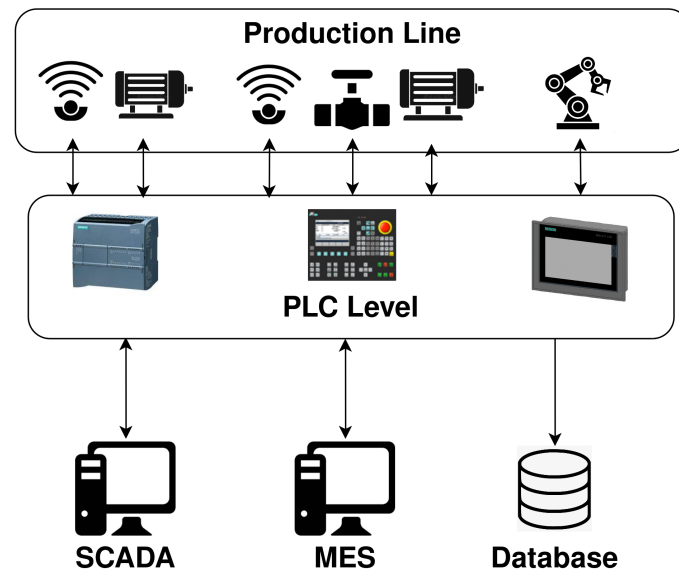


Figure 1. Standard initial state diagram for framework design.

In terms of the identified attributes, it will first be necessary to connect all the machines, production lines, actuators, and existing available electronic equipment on the IoT level so that enterprises can collect process data. Once the new IoT devices are in place, a thorough analysis will need to be performed, and a horizontal and vertical integration process will need to be designed and implemented. To ensure a seamless transition to the Smart Factory model, enterprises must focus on robust data connectivity and integration. This involves establishing a reliable network infrastructure that supports the continuous flow of data from IoT-enabled machines and sensors to central data systems. A critical step in this process is the development of a comprehensive data management strategy that includes data validation, cleansing, and transformation to ensure accuracy and consistency. These data then feed into advanced analytics platforms. Furthermore, the integration with existing SCADA, MES, and ERP systems ensures that the entire organization has access to real-time data, facilitating better decision making and enhancing overall operational efficiency. This holistic approach not only streamlines the transition but also maximizes the potential benefits of digitization, helping SMEs to achieve greater agility and resilience in their operations. Additionally, continuous monitoring and iterative improvements to the digital infrastructure are critical to adapting to emerging technologies and market demands. This proactive approach ensures that the enterprise remains competitive and can fully leverage the benefits of a Smart Factory environment, leading to improved operational efficiency and reduced downtime. Lastly, continuous innovation and adaptation are key to the long-term success of a Smart Factory. The digital landscape is rapidly evolving, with new technologies emerging that can further enhance operational efficiency and data utilization. Enterprises should remain open to adopting emerging technologies such as blockchain for secure data transactions, augmented reality for maintenance and training, and edge computing for faster data processing. Staying ahead of these trends ensures that the enterprise remains

agile and can respond effectively to changing market conditions and customer demands. The design can be seen in Figure 2 as a conceptual design of the digitization solution, and it is preceded by the original state in Figure 1. The following subsections detail the steps outlined in Figure 2 required to move SMEs towards Smart Factories.

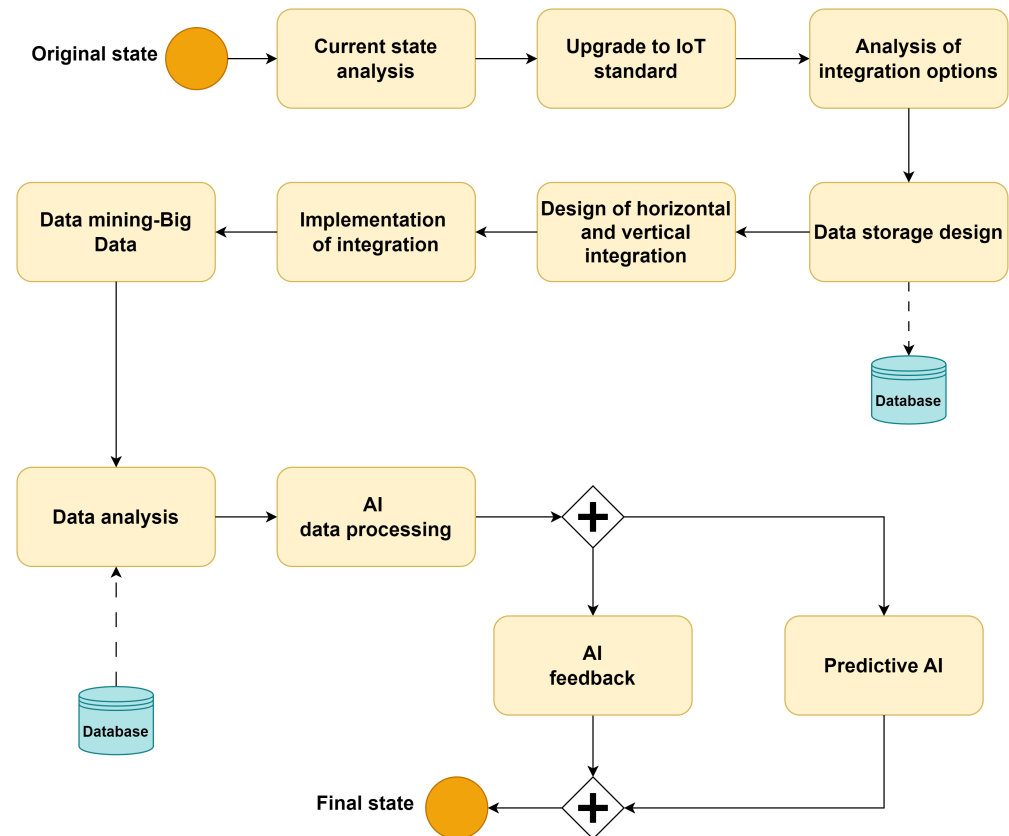


Figure 2. The conceptual design of a solution for the digitization of small and medium-sized enterprises (SMEs).

2.2. Design and Deployment of IoT Devices

The IoT devices in the industry perform an essential function where companies can collect process data from these devices and then analyze them, whether to optimize production, to create simulations, or for predictive maintenance. Therefore, companies must have such devices and cover the entire production, as more is needed to have only a part of IoT devices for a comprehensive analysis. At the same time, it is necessary to deploy IoT devices/sensors enabling data collection for predictive maintenance, such as thermometers, dust, vibration, and humidity sensors. These sensors mainly allow the monitoring of external factors but have a significant impact on production. The choice of sensors depends on the production line. It is necessary to analyze the usability of the sensors in a given production environment before deploying them and to select sensors that will collect valuable data at the workplace. For this reason, an analysis will be necessary for each company before purchasing and positioning sensors to ensure that the sensors are placed as effectively as possible in the production line environment. For example, different sensors will be needed in the paint shop, and different sensors will be required in production. It depends on the characteristics of the machine and the production environment. Also, when selecting sensors, consideration must be given to the coverage, the sensitivity of the sensors, and the way the sensors communicate. Companies have two options for connecting the new sensors:

- Connect new sensors to existing PLCs;
- Connect the sensors to microcontroller boards.

In the case of connecting sensors to existing PLCs, several problems arise. The sensors would need to be connected to PLC modules. PLC architectures are custom-designed, and most companies will not have spare modules to connect new sensors, which necessarily incurs the cost of buying PLC modules. If a company wants to connect analog and digital sensors, it will need modules for both types of signals, which will increase the cost. In addition, such an intervention will require the modification of the PLC control program, which may harm companies for which an external company created the program and they would have to pay additionally to upgrade the program. For the above reasons, connecting new sensors to the microcontroller board becomes a more advantageous solution. This solution avoids the problems mentioned above, while the above solution with sensors can be added to the existing architecture without the need to modify it. The microcontroller solution and the connection method will also influence the subsequent choice of sensors.

The advantages of connecting IoT devices via a microcontroller are as follows:

- A device with a microcontroller is an intelligent device capable of working with the end device and performing intelligent control.
- Single-board microcomputers are a cost-effective solution [25].

The authors of the paper [25] deployed an architecture with microcontrollers in two companies, and the architecture is being used daily. The architecture consists of IoT devices connected to microcontroller-based solutions. The connected sensors send data to the boards, which control actuators at the output according to the given inputs. The entire architecture is further connected to the Cloud, where process data are sent. This paper confirms the efficiency and low cost of IoT solutions with microcontrollers and validates the scalability of existing manufacturing architectures with such a solution.

Again, several criteria are used to select a suitable microcontroller solution. In the article [26], the criteria that influence the selection of microcontrollers were analyzed. The analysis resulted in the five most important criteria, which include the following:

- Reliability;
- Programming flexibility;
- Electronic functionality;
- Power consumption [26].

Price was also included in the analysis, but due to the low prices of microcontrollers on the market, price plays a minor role in the selection process. In the article [26], different microcontrollers are compared regarding the most important criteria identified. The Raspberry Pi 4 B (RPi 4), BeagleBone Black (BBB), Arduino Nano 33 BLE (AN 33 BLE), and Blue Pill (BP) achieved the best results. The experimental results show that BBB, AN 33 BLE, and Blue Pill are the best in terms of reliability. Their error rates were within 0.01 percent. RPi 4 and Blue Pill offer the highest programming flexibility, allowing many programming languages and libraries. RPi 4 and AN 33 BLE performed the best in supportability due to the large developer communities of these frameworks. BBB and Blue Pill have the highest electronic functionality because of their outstanding capabilities and project development features. AN 33 BLE and Blue Pill achieved the lowest power consumption. AN 33 BLE achieved around 700 mW, and Blue Pill achieved around 730 mW. RPi 4 achieved a power consumption of around 2.5 W and BBB 1.7 W [26]. All of the mentioned devices are suitable for use in IoT architecture. The suitability of their use must be based on the analysis performed on a given workstation and machine. Table 1 describes the selected technical parameters of the mentioned solutions and the current prices on the European market.

Table 1 shows that RPi 4 and BBB are several times more powerful than AN 33 BLE and Blue Pill.

The Arduino Nano board contains the lowest number of General Purpose Input/Outputs (GPIOs), but it is equipped with a gyroscope, accelerometer, magnetometer, digital microphone, pressure gauge, thermometer, and humidity sensor from the factory. The RPi 4 has optional versions that vary in RAM sizes: 1 GB, 2 GB, 4 GB, and 8 GB. The RPi 4 has a built-in display and camera port, making it suitable for image capture. All the solutions

mentioned work with a unified 3.3 V signal. Microcontrollers will work as an intermediate link in the proposed architecture for data acquisition. The data will be read either cyclically or when the value of a given sensor changes. Again, it depends on the characteristics of the sensing site and sensor. The read data will be sent to the proposed storage via communication protocols and programming tools. The framework design after embedding IoT devices and single-board microcomputers is shown in Figure 3. IoT Gateway can also be used to connect the new IoT devices.

Table 1. Technical parameters of selected single-board computers.

	Raspberry Pi 4B [27]	Beagle Bone Black [28]	Arduino Nano 33 BLE [29]	Blue Pill [18]
Processor	ARM Cortex A72, 1.5 GHz	AM335X ARM Cortex A8, 1 GHz	ARM Cortex-M4F, 64 MHz	ARM Cortex-M3, 72 MHz
RAM	1–8 GB	512 MB	256 kB	20 kB
Power source	USB C 5 V 3 A	5V 1.2–2A	Regulation up to 21 V	5 V
Unified signal	3.3 V	3.3 V	3.3 V	3.3 V
GPIO	28 digital	65 digital, 7 analog	14 digital, 8 analog	37 digital, 10 analog
Network interface	Ethernet, Wi-Fi, Bluetooth	Ethernet, Bluetooth, external Wi-Fi via USB port	Bluetooth	-
Connectivity	2× USB 2.0, 2× USB 3.0, microHDMI, audio, video RGB, DSI display port, CSI camera port	2× USB 2.0, HDMI, audio, LCD, three serial ports	1× USB	1× microUSB, 2× A/D converter, 2× SPI
Price on the EU market	EUR 38–95	EUR 56	EUR 22	EUR 8

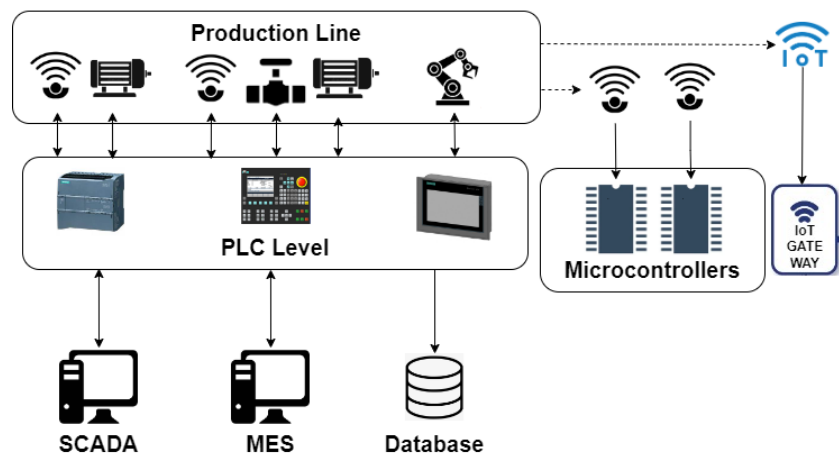


Figure 3. Designing a solution after deploying IoT devices and microcontrollers on the production line.

As can be seen in Figure 3, the proposed framework will be entirely independent of the original solution. This is possible just by using microcontrollers instead of a PLC solution. According to our proposal, IoT devices will be fitted to the existing production line at locations determined by the result of the analysis. The embedded IoT devices will be connected to the microcontroller boards, collecting data from them.

2.3. Data Storage Design

In the proposed architecture, the repository will be designed to store data and process them using a Hadoop-based system so that the data, once processed, are ready for visualization, ad hoc reports, or use for a digital twin. According to IBM, Hadoop is suitable for small and medium-sized enterprises because it can significantly improve enterprise performance even with a small number of clusters [30]. The advantages of using Hadoop are data protection in case of hardware failure, scalability from using a single server to

using thousands of clusters, and real-time analytics for decision making. A similar system is Apache Spark, which is faster than Hadoop but more expensive because it relies on in-memory computation for real-time data processing, which requires high performance and large RAM, unlike Hadoop. Although the Hadoop ecosystem is open-source and free, data mining can cost around EUR 5000 per month in an average enterprise [30]. These costs are related to maintaining active servers and cooling the infrastructure, not to mention the cost of building such a system. However, an investment can earn the enterprise more resources than it consumes. Compared to traditional data warehouses, Hadoop provides greater scalability in supporting relational and unstructured data, taking this ecosystem beyond a single vertically integrated system [31]. The authors of this study further state that Hadoop is a suitable tool for most automotive application workloads.

The authors in [32] applied Hadoop to a predictive maintenance system on a welding robot. Through analysis, they determined that the welding robot is a critical point in production and adequately deployed predictive maintenance can significantly help reduce downtime at this point in production. The deployed system helped in real-time prediction and could alert personnel in advance of a faulty spot on the welding robot, thereby significantly increasing maintenance efficiency and improving equipment. The deployment of the system increased the quality of production, reduced maintenance time, and thus increased overall productivity. Hence, the data warehouse design can identify the critical points in the production from the analysis and deploy Hadoop exactly on these vital points, increasing overall productivity. Deploying to critical locations saves the finances of building infrastructure to analyze the data, and it is possible to work with fewer clusters as such a system will require less performance than deploying Hadoop to the entire production. The paper [33] used Hadoop in Big Data architecture on a paint shop at Farplas Automotive. Data were collected using Apache NiFi into the Hadoop system from critical locations in plastic injection molding. The deployment of Hadoop resulted in high efficiency, as the system could prevent many problems through prediction and warn the operator before a problem occurred, thus increasing the productivity of the production in question. The study's authors describe Hadoop as a capable, scalable, reliable, and low-cost ecosystem [33,34]. As mentioned earlier, the advantages contribute to the decision to design a Hadoop-based storage solution for SMEs. From the above advantages, disadvantages, and research articles, it is concluded that Hadoop deployment suits SMEs. When storing and analyzing data, it is better to build the infrastructure on site and use another storage located at a remote location for backup or Cloud capabilities. In this case, using the Cloud can save the finances required to build and cool the infrastructure. Hence, it is possible to deploy Hadoop in designated critical locations in production and analyze data only from those critical locations, reducing the number of clusters required to process the data and thus reducing the cost of building and maintaining the infrastructure. Figure 4 shows the framework design after adding Hadoop.

Alternative data storage solutions

A Hadoop storage solution may be too costly to deploy and operate for capital-starved SMEs, especially if the business would need to collect all process data from production. Therefore, this subsection will describe alternative data storage solutions applicable to SMEs.

Apache Cassandra is an open-source NoSQL distributed database [35]. This system is easily scalable even if the application is under high load. The distribution also prevents data loss, e.g., when the hardware of a given data center fails. Deploying Cassandra on multiple machines is recommended to obtain all the benefits of a distributed database. Each computer is then referred to as a single data node. These nodes communicate with each other using the gossip protocol, which provides peer-to-peer communication. Unlike the master–slave architecture, each node has the same functionality as any other node, contributing to the overall system's resilience and robustness.

In Cassandra, data are automatically distributed among nodes, achieved by partitioning. Each node owns a particular set of tokens, and Cassandra distributes data within the

cluster based on the ranges of these tokens. The partition key is essential in determining the locality of the data and is responsible for distributing the data among the nodes. Once the data are inserted into the cluster, a hash function is applied to the partition key. The output is then used to determine the node where the data are stored based on the range of tokens [35]. Cassandra requires a minimum of 8 GB of RAM to function, but the recommended RAM size is at least 32 GB, and an 8-core processor is on each data node. The use of Cassandra is free, which is a great advantage for small and medium-sized enterprises that will only bear the cost of purchasing the necessary hardware to deploy this system.

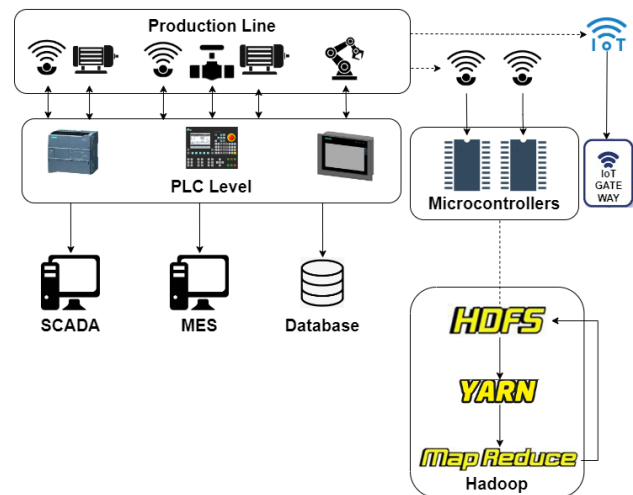


Figure 4. A block diagram of the framework after adding Hadoop.

ClickHouse is a column-oriented SQL database management system (DBMS) for online analytical processing (OLAP). It is offered as an open-source or Cloud solution [36]. ClickHouse can process data in a fraction of a second and return real-time results. It is an excellent tool for applications working with massive structured data sources. ClickHouse uses SQL syntax very similar to ANSI SQL and was created to work with significant data sources. The system uses data compression to achieve higher performance. It is not just universal compression but also several specialized codecs targeted at different data types stored in columns. ClickHouse is relatively unique and requires adherence to a strict architecture and data structure to achieve the best results.

ClickHouse requires 32 GB of RAM to function, and the manufacturer indicates that the system is not rated for use with less than 16 GB of RAM. The manufacturer recommends using up to 128 GB for high workloads [36]. ClickHouse is offered as a free, open-source solution but can also be purchased as a Cloud solution. However, since ClickHouse is suitable for working with large data sets, it is better suited for enterprises that want to handle all available process data, not just critical data. It is also more expensive and requires more computing power than Apache Cassandra.

2.4. Data Integration Design

The microcontrollers in the architecture designed this way must use communication protocols and tools to send data to the designed repository. In this case, the data must be divided into data to be processed in real time and non-real time. Real-time data are more complex to work with, but at the same time, there is less of such data than non-real-time data, and it is for these reasons that such a division is justified. Using different tools to work with the data, one that will work better with real-time and the other with non-real-time data, is advisable.

If IoT devices are connected to single-board microcomputers and are ready to collect process data, these data need to be integrated in the next step. A large number of commercial as well as open-source solutions are available on the market. It is up to each enterprise to consider which alternatives to use for its solution regarding the finances required to

use the software or the functionality a given integration tool can provide. The choice of the tool must be conditional on a prior analysis of the business, which will also indicate the suitability of the use of the tool. In this subchapter, some integration tools suitable for SME integration are mentioned. For selected integration tools, the way of communication between the IoT device and the database and the use of communication protocols for data transfer are also mentioned. Several criteria must be considered when selecting a suitable integration tool. In general, the following areas should be focused on:

- The number of supported data sources—it is necessary to consider not only the current number of data sources but also the possible growth of the company and, therefore, the increase in the number of data sources.
- Data security—especially when working with sensitive data, it is necessary to focus on how the integration tool works with the data.
- Scalability—as the amount of data increases, the system must continue to work efficiently and maximize its usability.
- Available data transformations—mapping data from different sources into a unified temporal view is time-consuming, and data transformations can provide the desired data format regardless of the source.
- The frequency of data updates—the need to process data in real time or batches.
- Learning curve—related to staff training. The less time staff has to spend learning new software, the better.
- Enterprise size—one type of tool is appropriate for large enterprises, and the other for small or medium-sized enterprises [37].

For SMEs, the price of the integration tool will also be an essential criterion. Analyzing the enterprise based on these criteria will significantly help in selecting the appropriate integration tool. According to Gartner, the most commercially used integration tools and best-rated customer reviews are Skyvia, Informatica Powercenter, Pentaho, Alteryx Designer, and Microsoft SQL Integration Services [38].

Each commercially available framework is presented as suitable for use in small and medium-sized enterprises. However, to achieve lower costs, it is better to use free solutions. Such a solution, especially when using microcontrollers, can be data integration using the Node-RED and Apache NiFi frameworks, as they offer solutions that can be deployed directly on selected types of microcontroller solutions. At the same time, they also include libraries for working with PLC devices. Commercially available frameworks can integrate all the data available in the enterprise, including data from social networks. However, using Node-RED and Apache NiFi to integrate the lower process level is better because of low cost, ease of use, and deployment. The Node-RED framework can be used to integrate process data retrieved periodically from IoT devices and PLCs.

Regarding the number of supported data sources criterion, Node-RED can work with virtually unlimited sources; it is only necessary to always supplement the existing program with new streams to collect new process data. Regarding data security, Node-RED does not need an active internet connection. It just needs to work on the local network, protecting data from misuse in the external environment. Since Node-RED is deployed on selected microcontrollers, in terms of scalability, the system can be expanded as needed. Regarding the data update frequency criterion, Node-RED can read data in batches and real time. It exclusively depends on the use of communication protocols. Working with Node-RED is easy, especially for people who have programming experience. However, with a large community of supporters, learning how to work with the framework quickly is possible. Overall, Node-RED is well suited for integration in small and medium-sized businesses.

Node-RED has an implementation tailored for deployment with microcontrollers, and its homepage provides links and files for installation on Raspberry Pi, Arduino, Beagle Bone Black devices, and more. It also offers several libraries to work with microcontrollers, allowing, for example, the direct querying of pins. Node-RED can retrieve data from sensors connected to the microcontroller, visualize these data using a graphical environment, and store them directly in a database. The available libraries are on the most commonly used

database systems, including the Hadoop Distributing File System (HDFS), so if enterprises use different database systems using Node-RED, there will be no problem integrating new data into an existing database.

Using MQTT as the communication protocol between Node-RED and the microcontroller is convenient. In this case, Node-RED will be a tool to retrieve data from sensors connected to the microcontroller using MQTT. Node-RED will send a request to retrieve data, the current data on the connected sensors will be retrieved, and the data will be sent to the variables in Node-RED using MQTT. The retrieved variables can be directly stored in the HDFS or database, and the whole process can be repeated using the Timestamp element periodically according to the selected time in the order of milliseconds. Since Node-RED can be directly deployed on the microcontroller, it uses its power and does not overload the local computer. If the enterprise has unintegrated data from the PLC, the tool can also be used for such integration. In this case, different communication protocols have to be used. Since Node-RED cannot be deployed directly on the PLC, as in the case of microcontrollers, it is necessary to use the power of the local PC or server when integrating the data. In this case, it will be essential to use OPC Unified Architecture (OPC UA) as the communication protocol to connect from Node-RED to the PLC and retrieve the relevant data, which can then be written into variables and sent to the database.

Figure 5 shows the communication scheme between the IoT device, PLC device, and database using the MQTT broker and OPC UA protocol. The arrow shows the data flow from sensors towards data storage or the Cloud periodically according to a selected time in milliseconds.

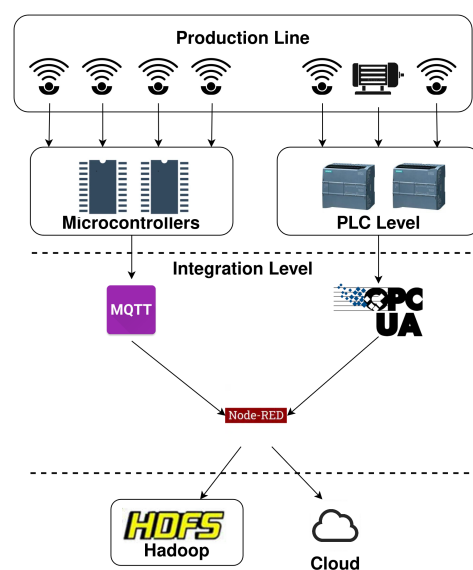


Figure 5. Block diagram of communication between IoT device, PLC, and database using MQTT and OPC UA and Node-RED.

2.5. Design of Process Data Integration Required to Process in Real Time

The open-source tool Apache NiFi can integrate the process data required for real-time processing. This tool is suitable for streaming process data from sensors and actuators to databases. In the case of Apache NiFi, a GUI and a non-GUI version are available. The version with a Graphical User Interface (GUI) is licensed and paid for. However, the unlicensed version, provided free of charge, can also be used for integration.

The Apache Foundation has developed a NiFi version called Apache MiNiFi to work with microcontrollers and power-limited devices. It can be installed on a microcontroller to create data streams that will be sent to Apache NiFi for further processing or stored in a database system. Like NiFi, MiNiFi is offered as a licensed paid version with a GUI or an unlicensed free version. The unlicensed version of both tools is sufficient to create streams for real-time data processing. There are two ways to create data streams for MiNiFi. The

first option is to manually create a Yet Ain't Markup Language (YAML) file, which is the configuration of the streams and the configuration of the MiNiFi instance used to start. However, this option is demanding and, therefore, not often used. The second way is to create flows in NiFi and save them as templates in an Extensible Markup Language (XML) file. This file can then be converted to a YAML file using the MiNiFi tool, which can be deployed to the MiNiFi instance.

Regarding the criterion of the number of supported data sources, NiFi can collect data from many sources. The number depends on the performance of the server on which NiFi is running. In terms of data security, various secure communication protocols can be used, and similar to Node-RED, the entire architecture can be designed without an external connection to the internet, making the security solution more straightforward. Regarding available data transformation, NiFi offers options to preprocess data into different forms. It uses so-called processors, which can process the data or store it in a database system. In terms of the frequency of data updates, NiFi is a suitable tool for processing process data in real time and is a suitable tool for use by small, medium, and large enterprises. In terms of scalability, NiFi runs on a local server and is thus affected by the performance of that server. If necessary, the architecture must be extended with a new server. Like Node-RED, NiFi is also used by many users, so in terms of a learning curve, learning how to work with NiFi is relatively easy as there are a large number of open and freely available resources. NiFi is a codeless tool that works with a drag-and-drop system, and you need to mainly learn how to connect individual data streams and configure nodes instead of working with code. NiFi is highly scalable, capable of handling from a few devices to thousands, making it suitable for businesses as they grow. Additionally, its ability to handle both batch and real-time data processing ensures that enterprises can adapt their data strategies to their specific operational needs. The support for a variety of database systems and the ease of extending its functionality through custom processors mean that NiFi can be a cornerstone of a business's data architecture, facilitating seamless integration and efficient data management.

The MQTT broker can be used as a communication protocol to communicate between MiNiFi and the microcontroller, on which data will be generated in JavaScript Object Notation (JSON) format. The second communication protocol will be used for communication between MiNiFi and NiFi. Since this communication will be used for real-time data transfer, it is possible to use REST API protocol, which is suitable for this type of communication due to its specification. The architecture design can be based on the diagram shown in Figure 6.

2.6. Design of Data Integration from MES Control System

After integrating the data from the production line and microcontrollers, it is possible to proceed to the design and integration of data from the MES control system. The data from the MES system also need to be collected and analyzed because the MES system contains unique data that can be used to improve and enhance the control process and, hence, the entire production process. Also, in the case of this integration, there are many different tools and options to integrate the data from the control system into the prepared database. It is possible to use commercial solutions, and solutions are offered free of charge. For small and medium-sized enterprises, however, it is more advantageous to focus on the options and tools provided free of charge, also in this case, from the point of view of financial requirements. A tool that has been used to integrate process data from microcontrollers, the Node-RED tool, can also be used to integrate data from the control system. If a company has used Node-RED to integrate process data, there will be an advantage in using this tool as there is no need to learn a new tool or introduce another new tool into the process.

The Node-RED tool offers a library called `node-red-contrib-nupmes` that provides tools for integrating data from MES systems. This library was created in 2022, so it is a relatively new tool [39]. The library offers options to connect to an existing MES system using the `NuPMesServer` and `NuPMesOPCUAServer` configuration nodes. The library

allows direct connection to the MES system via a Uniform Resource Locator (URL) or an OPC UA server, which can be used for the connection. A third option is to query any data from the MES using the REST API protocol, where individual data requests can be specified directly in the configuration of a given node. Another option is to use the NuPMesGetIO node and the NuPMesSetIO node, which allow values to be obtained or set directly at the I/O addresses, and these nodes can be triggered at set times periodically, using the “Tick” parameter. When obtaining or setting values, the REST API protocol is again used for communication. The library also offers Pareto analysis or Overall Equipment Effectiveness (OEE) analysis, which can be used if the MES system does not perform them and to obtain results, e.g., on the quality of production or the usability of machines in production. The NuPMesPareto and NuPMesOEE nodes are used for these purposes. The analyses can be used in several ways, either by specifying the start time of the analysis and the end time or by specifying the date and time and defining an attribute with a flag, e.g., “today” or “last hour”.

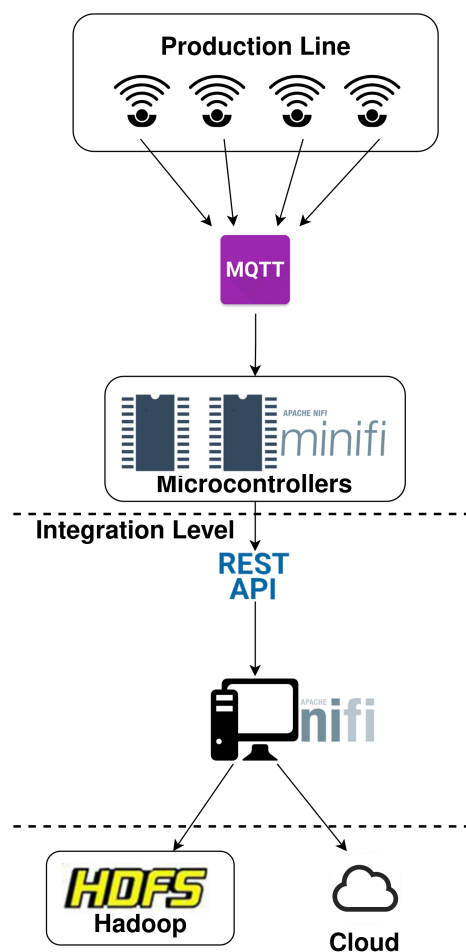


Figure 6. Block diagram of communication between IoT device and database with real-time data collection using Apache NiFi.

The data obtained from the MES system can be preprocessed into the desired format using Node-RED and stored in the proposed storage. This procedure is identical to the method for storing process data. It is possible to either store all the data that the MES system can provide or focus on the critical data identified by the analysis and store them in the database system. Critical data could include information about orders placed and executed, the number of pieces produced, machine performance, or alarms that the MES system records. This library can also integrate data from the SCADA system, like the MES

system. Figure 7 shows a proposed scheme for integrating data from the MES and SCADA systems into the created data repository.

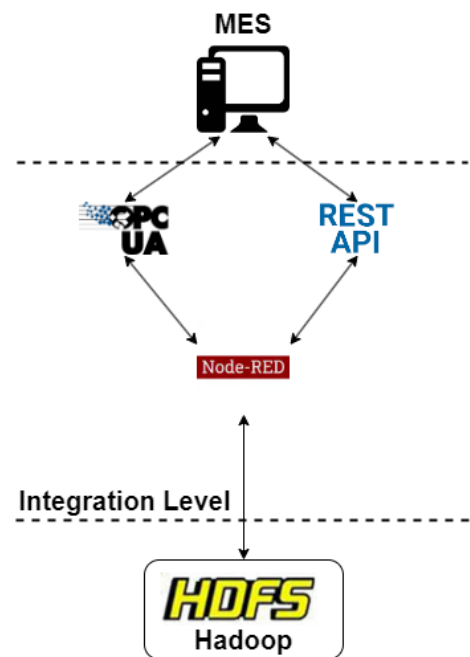


Figure 7. Block diagram of communication between MES system and database with data collection using Node-RED.

3. Results and Discussion

3.1. Summary of SME Framework Proposal

Based on the identified Industry 4.0 pillars in Section 1 and the design of the different parts of the framework, this chapter summarizes the process of upgrading SMEs to Industry 4.0 in the form of the Business Process Model and Notation (BPMN) diagram in Figure 8.

The first necessary step towards digitization is an analysis of the enterprise, which is required to go through the process and operational management levels in particular. For small and medium-sized enterprises, the focus should be on the process-level analysis in particular, primarily to identify critical points in the production process, to identify places to fit solutions with microcontrollers and connectible sensors that will collect usable data to help. For example, they can better control the environment affecting production. It is also necessary to analyze the other levels of the enterprise and identify the data that will need to be collected and stored from each level. When analyzing the enterprise and operational levels, it is necessary to focus on the existing infrastructure and how it can be used for data analysis, possibly its improvement, and the systems currently used by the enterprise. Once a thorough analysis has been carried out, the results need to be summarized, and based on the results, a digitization plan needs to be proposed, providing the basis for building the framework. The plan should propose the locations for the microcontrollers, focusing in particular on critical locations in production, the locations of the sensors, the plan for the use or completion of the infrastructure for data collection and analysis, the usability of the systems used, and their possible integration into a higher-level system. This section should clarify whether the business plans to back up the data to its built or existing storage or the Cloud. All plans mainly depend on the available resources and finances of the enterprise. However, a thorough analysis is the most critical part of the entire digitization process and, therefore, needs to be given sufficient time, effort, and resources.

Once the analysis is complete, it is possible to start a gradual digitization process by building a data repository and, at the same time, selecting solutions with a microcontroller. These two processes can run in parallel as they do not affect each other. When selecting

microcontrollers, it is necessary to consider their performance compared to power consumption since the prices of microcontrollers are low. Another critical parameter is the number of GPIOs, which affects the number of connectible sensors and their types.

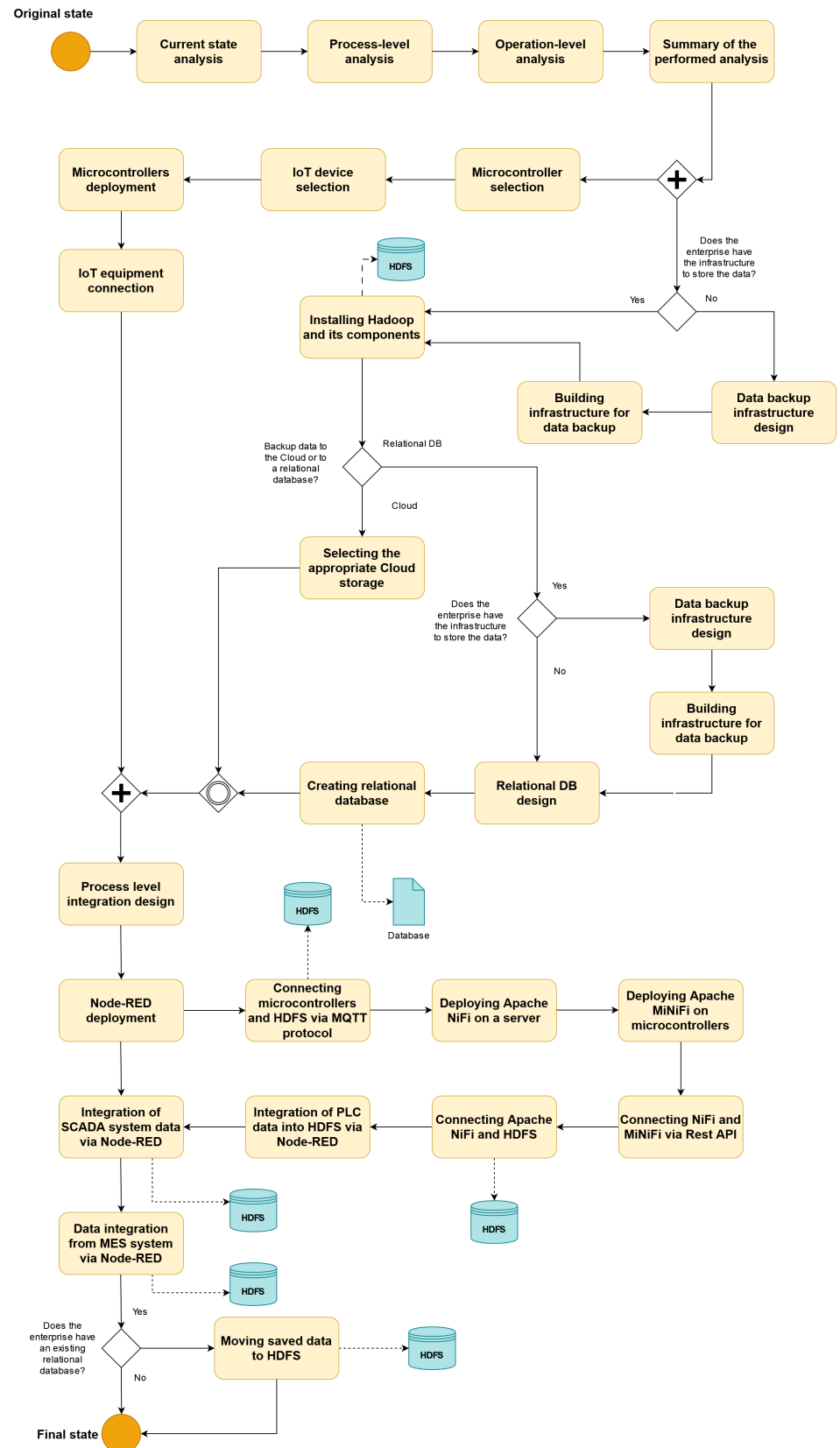


Figure 8. Designing the process of upgrading SMEs to the Smart Factory level.

If analog sensors are fitted on the line, it is better to select microcontrollers with Analog–Digital Converters (ADC) and the like; if only digital sensors are used, selecting a controller with digital pins is sufficient. Once the microcontrollers are selected, IoT devices can be chosen to be connected to the microcontrollers through multiple types of communication interfaces. From the process-level analysis, it must be clear which types of sensors need to be selected or which data will need to be collected and evaluated. The next step is to embed the microcontrollers on the production line while embedding the IoT devices in predetermined locations. IoT devices can then be connected to the microcontrollers via GPIO. The hardware is thus ready for data collection.

Parallel to the selection of microcontrollers, work can be performed on the design and creation of the data storage. If the enterprise has an established data storage infrastructure, it can proceed directly to installing Hadoop or another alternative storage system on the existing infrastructure. If the enterprise does not have an infrastructure in place, the infrastructure must first be designed based on the requirements and the analysis results. Once the infrastructure has been designed, it can be built. Building the infrastructure is the most expensive part of the entire transformation to the Smart Factory level. However, at the same time, it is necessary to select high-quality and robust components and build storage space for multiple data nodes for security reasons. Once the infrastructure is built, we can install Hadoop and all the necessary elements—HDFS, MapReduce, and Yet Another Resource Negotiator (YARN)—to process all the available data; since storing the data outside of HDFS as a backup is essential, deciding whether the data will be stored in a relational database or the Cloud is necessary. If the enterprise backs up the data to the Cloud, the next step is to select suitable Cloud storage that can reasonably store a high volume of data.

Another critical factor when choosing a storage device is a stable and fast internet connection, as a high transfer speed is also essential in the Cloud due to the volume of data transferred to prevent data loss. Suppose the enterprise decides to collect data in its database. In that case, the next step is to design and build the infrastructure for data storage, which does not have to be as powerful as the infrastructure for Hadoop, which will also reduce its cost. Using the same infrastructure for both the data backup and the Hadoop system is not advisable, as the enterprise could lose all the data if the hardware fails. It is therefore recommended to build two different infrastructures; if possible, they should not be located in the same place. It depends on the size of the company and its capabilities. However, it is better to locate the storage as far away from the data source and the Hadoop storage as possible, especially for security reasons. Once the repository is built, the design and creation of a data repository to back up the data follow, completing the parallel processes. An alternative in this section may be to use another suitable data store of choice, e.g., Apache Cassandra or ClickHouse, described in Section 2.3. If the microcontrollers and IoT devices are populated and the repositories built and established, the enterprise can move on to the process-level integration design.

The entire process-level integration design must be based on the results of the process-level analysis. In the course of the solution, before this step, the enterprise should already have microcontrollers and sensors in place and a Hadoop system ready to collect and process data. In this step of the solution, the enterprise should completely design the interconnections between the IoT devices and the Hadoop system. This will be gradually implemented into the solution process, focusing on the communication protocols and software. Based on the previous analysis, it is necessary to distinguish between real-time and non-real-time data processing, where real-time data are considered critical production data that must be sensed in real time. Different communication protocols should handle these two data types, as described in Section 2.4. At the same time, in this step, it is necessary to design data integration from existing PLC devices, HMIs, and existing higher-level systems and link them to the Hadoop system. If the enterprise also has an existing database providing data collection, it is necessary to propose moving the data to Hadoop.

After the process-level integration design step, the integration can be implemented in incremental steps.

Implementation begins with deploying the Node-RED system and microcontrollers into an industrial process. Such a system will work as a software intermediary between IoT devices and the HDFS. The Node-RED technology will be applied to non-critical or non-real-time data. It will be possible to periodically read data from sensors connected to the microcontrollers and write the read data to the HDFS using the created nodes. The final step in the implementation for non-critical data is to create the actual connection between the IoT devices and the HDFS using the MQTT protocol. The next step is to deploy Apache NiFi on the server and Apache MiNiFi on the microcontrollers. Both systems, like Node-RED, will work as an intermediary between the microcontrollers and the HDFS but will be used to process real-time data. Once NiFi and MiNiFi are deployed, interconnections using communication protocols need to be established. The interconnection between NiFi and MiNiFi is possible using the REST API communication protocol. By its specification, it is suitable for real-time data transfer, thus ensuring data transfer from the microcontrollers to the server. A data stream needs to be created on the server to monitor the data coming from the MiNiFi. NiFi can preprocess the data, which can be used to ensure that the data arrive in the HDFS in an acceptable format. It depends on the performance of the microcontroller and the server used, and accordingly, the data can be preprocessed either in the microcontroller or directly on the server. The data can then be sent directly to the HDFS storage, which is the next step in the implementation procedure and thus completes the data integration process from the microcontrollers.

In the next step, it is necessary to integrate data from PLC devices into the HDFS, which can be performed using Node-RED and OPC UA communication protocol. The data can be read periodically or when the value changes, preprocessed into an acceptable format using Node-RED, and sent directly to the HDFS, thus ensuring data integration from the original part of the PLC-controlled production line. The last part of the integration is integrating data from the MES and SCADA systems, and Node-RED with the `node-red-contrib-numpes` library can be used for both systems. The library offers two communication protocols for integration, either OPC UA or REST API. The data can then be preprocessed into the desired format using Node-RED and stored directly in the database system.

The last remaining step is to move the data from the relational database to the HDFS, ensuring that even legacy historized data from the production line can be processed. Figure 9 shows the final design of the framework for upgrading SMEs to the Smart Factory level with both hardware and software parts. The final design can be compared with Figure 1, which shows the initial production state before the proposed changes.

The original scheme is complemented by new IoT devices connected to microcontrollers for data mining. The microcontrollers are connected using communication protocols through MQTT protocol to the Node-RED system for non-real-time data processing or through REST API to the Apache NiFi server for real-time data processing. Both Node-RED and Apache NiFi send data directly to the Hadoop system, specifically to the HDFS, for data processing, which is implemented using YARN and MapReduce. The result of the processing is again stored in the HDFS. From the original scheme, data from the PLC level are stored in the HDFS for processing using Node-RED and OPC UA communication protocol, whereas, in the original scheme, these data were stored in the database. The MES and SCADA system data are stored in HDFS using the Node-RED framework, OPC UA, and REST API protocols. All data can be sent for backup to either a relational database or to the Cloud. In the design, all data from the original database are also stored in the HDFS for processing the historized data. The design also includes the option of alternative data storage solutions to Hadoop, namely, Apache Cassandra or ClickHouse.

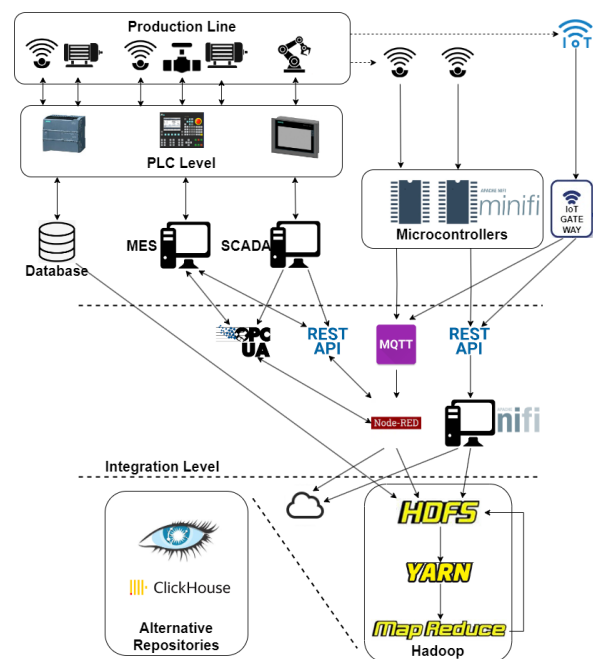


Figure 9. Final proposal for framework for upgrading SMEs to Smart Factory level.

3.2. The Deployment of the Framework on the Test Production Line

The framework was deployed under laboratory conditions on an AFB production system. This production line is automated and PLC-controlled but without Industry 4.0 features. New sensors, selected based on a system analysis, were fitted first. Three Raspberry Pi 4 microcontrollers were also fitted. Sparkfun ICM 20948 accelerometers and SPH0645 microphones were connected to the microcontrollers, serving as IoT devices for sensing the vibrations and sounds accompanying the occasional jamming of the line. The AFB production system utilizing Raspberry Pi and sensors is shown.

Node-RED was used to integrate the data from the microphones connected to the Raspberry Pi 4 using the Node-RED contrib ads1x15 i2c library, which is used to obtain the analog signal from the device connected to the I2C and also serves to convert the analog signal to digital. The Node-RED Waveshare library was used to integrate the accelerometer data, which work with multiple connected components, as well as the ICM20948. The data were processed and stored in a data store using Node-RED.

Hadoop was chosen as the data store. Data from the PLC and MES were integrated using Wonderware and stored in the Historian database, which runs on a separate server and also serves as a backup. The data were transferred from the Historian database to the HDFS. The resulting framework used on the AFB production system is shown in Figure 10. This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

3.3. The AFB Production Plant Use Case

One of the workstations on the AFB production plant is the “Bottle Filling Station” presented in Figure 11. The bottle filling station is equipped to handle empty bottles delivered via a conveyor belt. This station incorporates a carousel with seven distinct positions, including entry and exit points. The carousel operates sequentially across four functional positions: filling bottles with granules, filling bottles with liquid, applying caps using a pneumatic arm, and sealing the bottles with a rotating arm. Upon completion, the filled and sealed bottles are moved along the conveyor belt for further processing. A Siemens S7-300 Programmable Logic Controller (PLC) controls (Siemens, Munich, Germany) the station’s operation. Observation and data explanatory analysis revealed the filling station as a bottleneck in the manufacturing process, with occasional bottle jams at the carousel

entry. To address this problem, a Raspberry Pi 4 microcontroller was installed at the filling station's entry. This microcontroller was connected to Sparkfun ICM 20,948 accelerometers and SPH0645 microphones to detect specific vibrations indicative of bottle jams. Data from these sensors were transmitted via MQTT to Node-RED for processing and storage in the Hadoop Distributed File System (HDFS). Furthermore, all process data from the PLC were seamlessly integrated using the Wonderware system, particularly the ArchestrA IDE, into a Historian database (MySQL). This database also acts as a backup for production system data. Notably, data were collected every 300 milliseconds and promptly sent to HDFS. Alerts were instantly generated based on the microcontroller data, notifying operators of any bottle jam at the station entry. This feature enables swift intervention, thereby reducing production downtime. Similar measures were implemented at other problematic points along the line, with additional microcontrollers and sensors installed. Data from all PLCs on the line were integrated using the same principle.

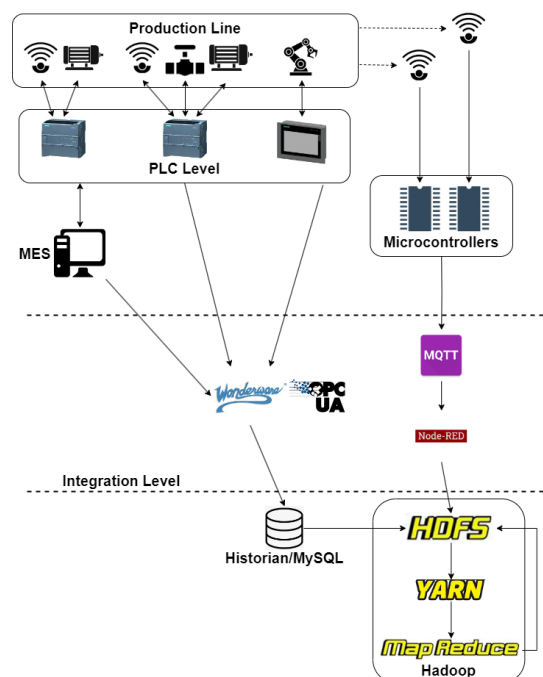


Figure 10. Final deployment of framework for upgrading SMEs to Smart Factory level.

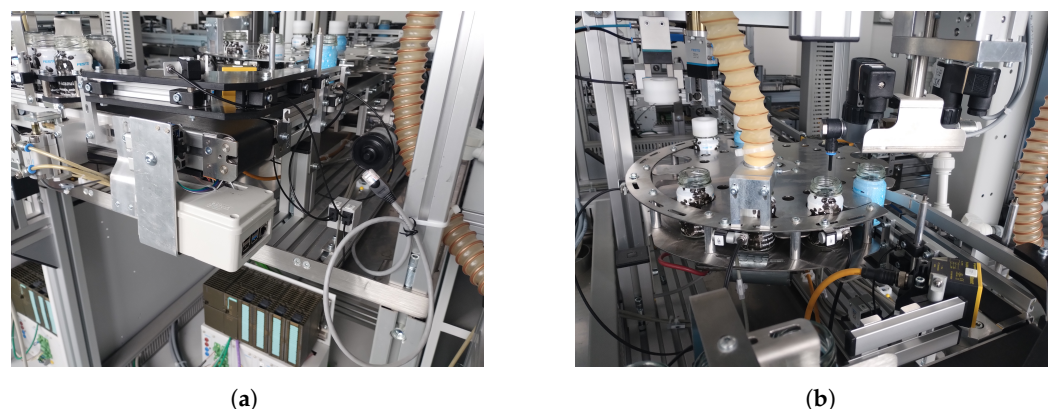


Figure 11. Bottle filling carousel as part of the AFB plant. (a) Utilizing a Raspberry Pi to log vibration data on the carousel, (b) different perspective on the carousel at the bottle filling workstation.

4. Conclusions

Small and medium-sized enterprises are currently facing pressure to increase investments in the digitization of the production process. The digital transformation process has

become an essential part of SMEs and directly impacts their competitiveness and sustainability. Digital transformation is a crucial process for SMEs, enabling them to better respond to changing market demands, optimize production processes in real time, and increase overall productivity and competitiveness. By incorporating selected critical elements into their manufacturing operations, SMEs can go beyond traditional production methods and reach a level close to a Smart Factory. The detailed research showed clear benefits of incorporating Industry 4.0 attributes, e.g., increased production quality, productivity, reduced waste, reduced downtime, higher profits, increased business competitiveness, and similar results [5]. These benefits can serve as an incentive for enterprises to increase their level of digitization. The framework was designed on a model solution of a manufacturing company, independent of the initial state of production. In this article, we discussed how to implement each change. SMEs need to be systematic and strategic when introducing changes towards a Smart Factory transformation. First, it was proposed to fit new IoT devices by connecting them to the peripherals of microcontrollers, thus ensuring the sensing of physical variables that can indirectly affect production and cause errors or downtime. Subsequently, a Hadoop-based data repository was designed to store and process all process data from the production. Integrating and storing data using Hadoop can be problematic for some smaller enterprises as it is a complex ecosystem. Therefore, alternative data stores suitable for small and medium-sized enterprises were proposed. In the next step, the integration of process data from microcontrollers, PLCs, and higher-level SCADA and MES systems into the created data repository via Node-RED and Apache NiFi was proposed. The contribution of this paper is the generalized framework design for SMEs shown in Figure 9. This conceptual design can be deployed in any manufacturing enterprise because it is independent of the initial production state. The resulting framework is one possible solution for incorporating the necessary attributes of Industry 4.0.

For enterprises, the selection of the same components is not necessary in the vital framework, but the incorporation of all necessary attributes into production is. Given that every manufacturing enterprise is unique, the framework cannot be considered the only possible solution for the digitization of SMEs. SMEs must incorporate each of the identified necessary attributes of Industry 4.0 into production, but the specific way they are incorporated can vary. The choice of IoT devices, data storage, integration tools, communication protocols, and hardware must be based on a detailed business analysis. It will primarily depend on the requirements of the business in question. In the case of SMEs, we emphasized low upfront investment costs when designing the framework. Therefore, alternative data storage solutions were also proposed during the solution, which represents the most expensive item of the entire investment. The main objective of this paper was to design a framework for upgrading SMEs to the Smart Factory level by implementing the necessary attributes of Industry 4.0.

Future studies will closely monitor and diagnose the advanced data obtained by the deployed framework on the AFB production line. This initiative aims to preemptively identify potential failures on the production line before they occur in the production system and is part of a strategy to leverage predictive maintenance. Consequently, a significant amount of data need to be acquired. Finally, following the data acquisition phase, the data will be analyzed using advanced data analytics and mining techniques. This process aims to identify patterns and trends, enabling the prediction of potential future failures in the production process.

Author Contributions: Conceptualization, M.B. and R.B.; methodology, M.B. and P.T.; software, R.B.; validation, M.B. and R.B.; formal analysis, R.B.; investigation, M.B.; resources, R.B. and I.S.; data curation, R.B.; writing—original draft preparation, M.B.; writing—review and editing, R.B.; visualization, R.B. and I.S.; supervision, P.T.; project administration, M.N.; funding acquisition, P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the modernization and new possibilities of online education in the field of logical control systems and process visualization under Grant KEGA-1782 (2022–2024).

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Türkeş, M.C.; Oncioiu, I.; Aslam, H.D.; Marin-Pantelescu, A.; Topor, D.I.; Căpuşneanu, S. Drivers and Barriers in Using Industry 4.0: A Perspective of SMEs in Romania. *Processes* **2019**, *7*, 153. [\[CrossRef\]](#)
2. Szalavetz, A. Digitalisation-induced performance improvement: Don't take it for granted! *Acta Oecon.* **2022**, *72*, 457–475. [\[CrossRef\]](#)
3. Amaral, A.; Peças, P. A Framework for Assessing Manufacturing SMEs Industry 4.0 Maturity. *Appl. Sci.* **2021**, *11*, 6127. [\[CrossRef\]](#)
4. Abdel-Aty, T.A.; McFarlane, D.; Brooks, S.; Salter, L.; Abubakar, A.S.; Evans, S.; Hawkridge, G.; Price Thomas, K.; Negri, E.; Mukherjee, A.; et al. The Role of Low-Cost Digital Solutions in Supporting Industrial Sustainability. *Sustainability* **2024**, *16*, 1301. [\[CrossRef\]](#)
5. Barton, M.; Budjac, R.; Tanuska, P.; Gaspar, G.; Schreiber, P. Identification Overview of Industry 4.0 Essential Attributes and Resource-Limited Embedded Artificial-Intelligence-of-Things Devices for Small and Medium-Sized Enterprises. *Appl. Sci.* **2022**, *12*, 5672. [\[CrossRef\]](#)
6. Chang, S.C.; Chang, H.H.; Lu, M.T. Evaluating Industry 4.0 Technology Application in SMEs: Using a Hybrid MCDM Approach. *Mathematics* **2021**, *9*, 414. [\[CrossRef\]](#)
7. Häring, K.; Pimentel, C.; Teixeira, L. Industry 4.0 Implementation in Small- and Medium-Sized Enterprises: Recommendations Extracted from a Systematic Literature Review with a Focus on Maturity Models. *Logistics* **2023**, *7*, 99. [\[CrossRef\]](#)
8. Ciuffoletti, A. Low-Cost IoT: A Holistic Approach. *J. Sens. Actuator Netw.* **2018**, *7*, 19. [\[CrossRef\]](#)
9. Horak, T.; Strelec, P.; Huraj, L.; Tanuska, P.; Vaclavova, A.; Kebisek, M. The Vulnerability of the Production Line Using Industrial IoT Systems under DDoS Attack. *Electronics* **2021**, *10*, 381. [\[CrossRef\]](#)
10. Martinez, E.M.; Ponce, P.; Macias, I.; Molina, A. Automation Pyramid as Constructor for a Complete Digital Twin, Case Study: A Didactic Manufacturing System. *Sensors* **2021**, *21*, 4656. [\[CrossRef\]](#)
11. Montoro-Lendínez, A.; López-Ruiz, J.L.; Charte, F.; Espinilla-Estévez, M.; Medina-Quero, J. Low-cost IoT gas concentrator system prototype. In Proceedings of the 2022 Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (XV Technologies Applied to Electronics Teaching Conference), Teruel, Spain, 29 June–1 July 2022; pp. 1–7.
12. Johnston, S.J.; Apetroaie-Cristea, M.; Scott, M.; Cox, S.J. Applicability of commodity, low cost, single board computers for Internet of Things devices. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; pp. 141–146.
13. Molangi, G.M.; Tamasala, M.M.; Sadi, S.Y.; Lusala, A.K. Design of a low-cost IoT-based platform for real-time monitoring of power quality and energy consumption. In Proceedings of the 2022 IEEE PES/IAS PowerAfrica, Kigali, Rwanda, 22–26 August 2022; pp. 1–5.
14. Ullah, U.; Bhatti, F.A.; Maud, A.R.; Asim, M.I.; Khurshid, K.; Maqsood, M. IoT-enabled computer vision-based parts inspection system for SME 4.0. *Microprocess. Microsyst.* **2021**, *87*, 104354. [\[CrossRef\]](#)
15. Adesta, E.Y.T.; Hilmy, I.; Avicenna; Agusman, D. Toward Real Time IoT Based Paste Monitoring System for Small to Medium Enterprise (SME). *J. Phys. Conf. Ser.* **2019**, *1167*, 012006. [\[CrossRef\]](#)
16. Martikkala, A.; David, J.; Lobov, A.; Lanz, M.; Ituarte, I.F. Trends for Low-Cost and Open-Source IoT Solutions Development for Industry 4.0. *Procedia Manuf.* **2021**, *55*, 298–305. [\[CrossRef\]](#)
17. Sen, D.; Ozturk, M.; Vayvay, O. An Overview of Big Data for Growth in SMEs. *Procedia-Soc. Behav. Sci.* **2016**, *235*, 159–167. [\[CrossRef\]](#)
18. Lab, M. STM32F103C8T6 Blue Pill Development Board, 2021.
19. Lukoki, V.; Varela, L.; Machado, J. Simulation of Vertical and Horizontal Integration of Cyber-Physical Systems. In Proceedings of the 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT), Prague, Czech Republic, 29 June–2 July 2020; Volume 1, pp. 282–287.
20. Torres Ventura, J.; Ruelas Puente, A.H.; Herrera García, J.R. Rendimiento para la interoperabilidad entre Raspberry pi, ESP8266 y PLC con Node-RED para el IIoT. *Ingenius* **2023**, 90–97. [\[CrossRef\]](#)
21. Para, J.; Ser, J.D.; Nebro, A.J.; Zurutuza, U.; Herrera, F. Analyze, Sense, Preprocess, Predict, Implement, and Deploy (ASPPID): An incremental methodology based on data analytics for cost-efficiently monitoring the industry 4.0. *Eng. Appl. Artif. Intell.* **2019**, *82*, 30–43. [\[CrossRef\]](#)
22. Young, R.; Fallon, S.; Jacob, P. An Architecture for Intelligent Data Processing on IoT Edge Devices. In Proceedings of the 2017 UKSim-AMSS 19th International Conference on Computer Modelling and Simulation (UKSim), Cambridge, UK, 5–7 April 2017; pp. 227–232.

23. Illa, P.K.; Padhi, N. Practical Guide to Smart Factory Transition Using IoT, Big Data and Edge Analytics. *IEEE Access* **2018**, *6*, 55162–55170. [[CrossRef](#)]
24. Chen, B.; Wan, J.; Shu, L.; Li, P.; Mukherjee, M.; Yin, B. Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. *IEEE Access* **2018**, *6*, 6505–6519. [[CrossRef](#)]
25. Wu, Z.; Qiu, K.; Zhang, J. A Smart Microcontroller Architecture for the Internet of Things. *Sensors* **2020**, *20*, 1821. [[CrossRef](#)]
26. Álvarez, J.L.; Mozo, J.D.; Durán, E. Analysis of Single Board Architectures Integrating Sensors Technologies. *Sensors* **2021**, *21*, 6303. [[CrossRef](#)]
27. Raspberry Pi 4 Model B. *OFFICIAL RASPBERRY PI HANDBOOK 2024: Astounding Projects with Raspberry Pi Computers*; Raspberry Pi: London, UK, 2019.
28. Coley, G. BeagleBone Black System Reference Manual. Available online: https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf (accessed on 24 February 2024).
29. Arduino Nano 33 BLE Product Reference Manual, 2024. Available online: <https://docs.arduino.cc/resources/datasheets/ABX0030-datasheet.pdf> (accessed on 23 June 2024).
30. IBM. *Hadoop vs. Spark: What's the Difference?*; IBM: Armonk, NY, USA, 2021.
31. Luckow, A.; Kennedy, K.; Manhardt, F.; Djerekarov, E.; Vorster, B.; Apon, A. Automotive big data: Applications, workloads and infrastructures. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015.
32. Guo, D.; Du, W.; Peng, H.; Zhao, C.; Wang, P.; Chen, X. Research on prognostics technology of spot-welding system in automotive manufacturing based on statistical process control. In Proceedings of the 2021 China Automation Congress (CAC), Beijing, China, 22–24 October 2021; pp. 4025–4029.
33. Akın, O.; Deniz, H.F.; Nefis, D.; Koziltan, A.; Cakir, A. Enabling Big Data Analytics at Manufacturing Fields of Farplas Automotive. In *Intelligent and Fuzzy Techniques: Smart and Innovative Solutions, Proceedings of the INFUS 2020 Conference, Istanbul, Turkey, 24–26 August 2021*; Kahraman, C., Cevik Onar, S., Oztaysi, B., Sari, I.U., Cebi, S., Tolga, A.C., Eds.; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2021; pp. 817–824.
34. Xu, L.D.; Duan, L. Big data for cyber physical systems in industry 4.0: A survey. *Enterp. Inf. Syst.* **2019**, *13*, 148–169. [[CrossRef](#)]
35. Apache Cassandra Documentation. Technical Report. Available online: <https://cassandra.apache.org/index.html> (accessed on 1 April 2024).
36. Clickhouse, C. What Is ClickHouse? 2023. Available online: <https://clickhouse.com/docs/en/intro> (accessed on 3 March 2024).
37. Dancuk, M. 12 Data Integration Tools Reviewed, 2022. Available online: <https://phoenixnap.com/kb/data-integration-tools> (accessed on 23 March 2024).
38. Gartner Peer Insights. Best Data Integration Tools Reviews 2024 | Gartner Peer Insights. Manual.
39. Node-RED Community. Node-RED Library: Node-Red-Contrib-Nupmes. Version 3.0.7, 2024. Available online: <http://flows.nodered.org/node/node-red-contrib-nupmes> (accessed on 14 January 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.