

## COPYRIGHT WARNING

This paper is protected by copyright. You are advised to print or download **ONE COPY** of this paper for your own private reference, study and research purposes. You are prohibited having acts infringing upon copyright as stipulated in Laws and Regulations of Intellectual Property, including, but not limited to, appropriating, impersonating, publishing, distributing, modifying, altering, mutilating, distorting, reproducing, duplicating, displaying, communicating, disseminating, making derivative work, commercializing and converting to other forms the paper and/or any part of the paper. The acts could be done in actual life and/or via communication networks and by digital means without permission of copyright holders.

The users shall acknowledge and strictly respect to the copyright. The recitation must be reasonable and properly. If the users do not agree to all of these terms, do not use this paper. The users shall be responsible for legal issues if they make any copyright infringements. Failure to comply with this warning may expose you to:

- Disciplinary action by the Vietnamese-German University.
- Legal action for copyright infringement.
- Heavy legal penalties and consequences shall be applied by the competent authorities.

The Vietnamese-German University and the authors reserve all their intellectual property rights.



# Assessing the use of edge computing to accelerate Industry 4.0 transformation in manufacturing companies

## BACHELORTHESES



Vietnamese-German University

**Submitted by:** Huy Duc Pham

**Mat.-Nr. 108 015 266 694**

**Study subject:** Mechanical Engineering

**Submission date:** 12.02.2024

**1<sup>st</sup> Supervisor:** Prof. Dr.-Ing. Bernd Kuhlenkötter

**2<sup>nd</sup> Supervisor:** Dr.-Ing. Christopher Prinz

**Advisor:** M.Sc. Maximilian Bega

## Bachelorthesis

Pham, Duc Huy

Matrikelnummer: 108015266694  
Prüfungsordnung: BPO/ MPO 2013

**Thema: Assessing the use of edge computing to accelerate Industry 4.0 transformation in manufacturing companies**

Years after the introduction of the  Vietnamese-German University, companies of various sizes continue to encounter significant obstacles in the incorporation of corresponding technologies and the resulting restructuring processes. In the recent context, where machine learning is becoming increasingly important for businesses, the data situation often hinders the meaningful application of these algorithms. Thus, implementing technology to gather data is becoming even more crucial.

This bachelor's thesis examines whether the integration and usage of edge computing to aggregate, process and provide data can accelerate or facilitate corresponding transformation processes in manufacturing companies. To achieve this goal, the characteristics of using edge computing in manufacturing and in particular on the shopfloor will first be theoretically prepared. This section should identify proper characteristics of edge technology that can simplify and accelerate transformation processes. In addition to the theoretical preparation of the possibilities of edge computing in discrete manufacturing, reasons and hurdles will be determined why the transformation of companies is taking place slowly - theoretical possibilities of edge technology are matched against the specified obstacles.

In a next step, exemplary use cases will be developed and implemented in an industry-oriented experimental environment. Subsequently, a thorough evaluation on the effectiveness of edge computing in addressing the aforementioned issues is conducted. The objective assessments will be employed to determine the extent to which edge computing can mitigate these challenges.

Ausgabedatum: 13.11.2023  
Betreuer: M.Sc. Maximilian Bega

# Acknowledgement

I want to express my deepest appreciation to M.Sc. Maximilian Bega for helping me with valuable instructions and comments throughout the time of internship as well as thesis. I would also like to thank to M.Sc. Furkan Ercan for helping me get familiar with coding and understanding codes.

I would like to thank Lehrstuhl für Produktionssystem for giving me the opportunity to work with great people in the department and providing such a great working environment. My thesis could not have finished without the support from them.

Finally, I would like to express my appreciation to my family, lecturers from both Vietnamese-German University and Ruhr-Universität Bochum, and friends who give me so much motivation during the internship and thesis.



# Abstract

This thesis examines the significant impact of edge computing on contemporary manufacturing, with a specific emphasis on its capacity to fundamentally change data processing and decision-making in the Industry 4.0 era. Firstly, an overview of edge computing in manufacturing is introduced. Secondly, within the realm of Industrial Internet of Things, considering that edge computing offers distinct advantages, thereby generating value for advanced manufacturing. Finally, through the implementation of edge computing devices in manufacturing environment, the study demonstrates substantial improvements in data processing latency, decision-making capabilities, and integration with existing manufacturing systems.



# Table of content

<b>Acknowledgement.....</b>	<b>I</b>
<b>Abstract.....</b>	<b>II</b>
<b>Table of content.....</b>	<b>III</b>
<b>List of abbreviation.....</b>	<b>V</b>
<b>List of code.....</b>	<b>VI</b>
<b>List of figures.....</b>	<b>VII</b>
<b>List of tables .....</b>	<b>VIII</b>
<b>1. Introduction.....</b>	<b>1</b>
<b>2. State of Art.....</b>	<b>2</b>
2.1. Industry 4.0 and IoT in manufacturing .....	2
2.1.1. Industry 4.0 .....	2
2.1.2. IoT Framework in manufacturing .....	3
2.2. Edge computing.....	4
2.2.1. Components of edge computing.....	4
2.2.2. Edge vs. Cloud vs. Fog.....	7
2.3. Edge computing - Catalyzing Data Processing Efficiency .....	9
2.3.1. In Industry 4.0 .....	9
2.3.2. In manufacturing.....	10
2.4. Challenges in manufacturing systems implementation.....	11
2.4.1. In Industry 4.0 implementation .....	11
2.4.2. Edge computing implementation.....	13
2.5. Overall Equipment Effectiveness (OEE) .....	14
<b>3. Methodology .....</b>	<b>15</b>
3.1. Background .....	15
3.2. Approaching .....	15
3.3. Define .....	16
3.3.1. Research Questions 1 and 2.....	16
3.3.2. Research Question 3 .....	17
<b>4. Experimental Setup.....</b>	<b>18</b>
4.1. Experimental Machine and Devices.....	18
4.2. Collecting data from manufacturing machine and storing data .....	20

4.3.	Computing and processing real-time data .....	21
4.3.1.	OEE calculation .....	22
4.3.2.	Live monitoring .....	22
<b>5.</b>	<b>Challenges in Implementing edge devices .....</b>	<b>23</b>
5.1.	Competencies in programming languages .....	23
5.2.	Lack of memory .....	23
<b>6.</b>	<b>Implementation and Result.....</b>	<b>25</b>
6.1.	Network preparation.....	25
6.1.1.	RevPi Core.....	25
6.1.2.	NetFIELD Compact X8M .....	25
6.2.	JAESPA Bandsaw .....	26
6.2.1.	Collecting data.....	26
6.2.2.	Storing and visualizing data .....	27
6.2.3.	Result .....	29
6.3.	Computing with real-time data.....	31
6.3.1.	Phase 1 - Setting input for OEE calculation .....	31
6.3.2.	Phase 2 - Calculate OEE.....	33
6.3.3.	Result .....	38
<b>7.</b>	<b>Evaluation .....</b>	<b>41</b>
<b>8.</b>	<b>Conclusion and Outlook.....</b>	<b>42</b>
<b>9.</b>	<b>Reference.....</b>	<b>43</b>
<b>10.</b>	<b>Affidavit .....</b>	<b>52</b>

# List of abbreviation

AI	Artificial Intelligence
ATMs	Automated Teller Machines
CDNs	Content Delivery Networks
CNC	Computerized Numerical Control
CPU	Central Processing Unit
IIoT	Industrial Internet of Things
I/O	Inputs/Outputs
IoT	Internet of Things
IP	Internet Protocol
LFF	Lern – und Forschungsfabrik
LPS	Lehrstuhl für Produktionssystem
OEE	Overall Equipment Effectiveness
RAM	Random Access Memory
RevPi Core	Revolution Pi Core
RQ	Research Question
RUB	Ruhr-Universität Bochum
SSH	Secure Shell
UTC	Universal Time Coordinated



Vietnamese-German University

# List of code

Code 1: Convert from string to number.....	28
Code 2: Creating table .....	29
Code 3: Selecting data in InfluxDB.....	29
Code 4: "Limit hours" configuration for working hour.....	32
Code 5: Code for getting timestamp and receiving data from "Refresh" button node or from Phase 1 .....	35
Code 6: Define time variables .....	35
Code 7: Calculating for Availability and Performance .....	36
Code 8: Calculating for Quality and OEE.....	36
Code 9: Full script of OEE .....	38



# List of figures

Figure 1: IoT Framework .....	3
Figure 2: IoT framework in manufacturing .....	4
Figure 3: Edge computing: cloud to IoT devices .....	5
Figure 4: Real-time OEE Dashboard at 11:00 (UTC +1).....	11
Figure 5: Overview of data flow in the implementation .....	17
Figure 6: Outline of the implementation .....	17
Figure 7: netFIELD Compact X8M (left) and RevPi Core (right) in LFF .....	18
Figure 8: JAESPA Bandsaw.....	20
Figure 9: ID from Bandsaw and configuration in PiCtory .....	21
Figure 10: Checking storage in RevPi Core through SSH connection.....	24
Figure 11: Storage in RevPi Core after re-install image package .....	24
Figure 12: Memory of the image package.....	24
Figure 13: Data flow for bandsaw .....	26
Figure 14: Config for collecting data from Bandsaw .....	27
Figure 15: Config IP and Input signal of RevPi Core .....	27
Figure 16: Flows for storing data into InfluxDB .....	28
Figure 17: Flow for displaying the dashboard.....	28
Figure 18: Data for Input START (above) and Input STOP in InfluxDB (below) .....	30
Figure 19: Data visualized in Node-RED.....	30
Figure 20: Node-RED flows for calculating and live monitoring OEE .....	31
Figure 21: Config flows for Input.....	32
Figure 22: The whole Node-RED flows for input quantities to OEE .....	33
Figure 23: Config join node to join 4 Inputs into one message.....	33
Figure 24: Gathering data and calculate OEE .....	34
Figure 25: Result when pressing "Refresh" button (above) and result when receiving input from Phase 1 (below) .....	38
Figure 26: Config flow for the gauge nodes.....	39
Figure 27: Config data for the gauge nodes .....	39
Figure 28: OEE dasboard when receiving data from Phase 1 (above) and when pressing "Refresh" button (below).....	40

# List of tables

Table 1: Technical Data of edge devices..... 19

Table 2: The I/O Addresses at the bandsaw and the I/O at the Rev Pi..... 21



## 1. Introduction

Within the realm of Industrial Internet of Things (IIoT), edge computing offers additional advantages such as flexibility, immediate data processing, and self-governance, all of which contribute to the creation of value in intelligent manufacturing. In intelligent manufacturing, the increasing prevalence of terminal network devices has resulted in fresh obstacles regarding the management and maintenance, scalability, and reliability of data centers. Edge computing has shifted the location of computation from centralized data centers to the edge of the network. In addition, edge computing seeks to overcome these issues by establishing a transparent platform that can seamlessly include fundamental functionalities such as networking, processing, storage, and application.<sup>1</sup>

Several research papers have highlighted the benefits of edge devices in Industry 4.0. Verma and Kumar have analyzed the application of edge computing in smart healthcare for early disease detection and remote monitoring of patient.<sup>2</sup> Patel et al. proposed a new method for conducting data analytics Internet of Things using edge devices such as Raspberry Pi.<sup>3</sup> Zhang and Ji applied an edge computing architecture to enable production anomaly detection and energy-efficient production decision approach for discrete manufacturing workshops.<sup>4</sup> This thesis aims to assess the use of edge devices in a manufacturing environment by integrating Overall Equipment Effectiveness (OEE) and tracking binary signal data during machine operation. This should revolutionize maintenance practices, reducing downtime and increasing manufacturing efficiency.

The thesis consists of 9 chapters, including the Introduction and Reference. Chapter 2 introduces Industry 4.0 and provides a theoretical background of edge computing, including a comparison between cloud, fog, and edge computing. It also discusses the advantages and disadvantages of edge computing in Industry 4.0 and manufacturing. Then the advantages are summarized to form the basis for a methodology for implementation in Chapter 3. Chapters 4, 5, and 6 detailed the implementation of the edge device based on the methodology, and Chapter 7 evaluates the results. Chapter 8 serves as the conclusion for this thesis and provides an outlook for further development.

---

<sup>1</sup> Chen et al. 2018

<sup>2</sup> Verma und Kumar 2023

<sup>3</sup> Patel et al. 2017

<sup>4</sup> Zhang und Ji 2020

## 2. State of Art

This chapter presents a theory on edge computing, beginning with a brief overview of Industry 4.0. It then explains the differences between cloud, fog, and edge computing in manufacturing. The following section outlines the advantages and disadvantages of implementing edge devices in manufacturing.

### 2.1. Industry 4.0 and IoT in manufacturing

This section presents an overview of Industry 4.0 and IIoT, highlighting the significance of IIoT in transforming manufacturing processes. Furthermore, it describes a fundamental concept of the Internet of Things (IoT) framework in manufacturing.

#### 2.1.1. Industry 4.0

Industry 4.0, often known as the 4th industrial revolution, envisions the creation of sophisticated cyber-physical systems to build smart factories. It will facilitate the development of industrial ecosystems powered by intelligent systems that possess autonomous self-attributes, such as self-configuration, self-monitoring, and self-healing.<sup>5</sup>

The implementation of Industry 4.0 can lead to exceptional levels of operational efficiency and a significant increase in production speed. Emerging forms of sophisticated manufacturing and industrial processes will arise, focusing on the collaboration between machines and humans and the development of symbiotic products.<sup>6</sup>

Industry 4.0 will embrace a wide range of technologies and their accompanying concepts. Some of the growing trends in the field include the Industrial Internet and the IIoT, as well as new product creation approaches in the 21st century, such as cloud-based design, cloud-based manufacturing, crowd sourcing, and open innovation, among others.<sup>7</sup>

---

<sup>5</sup> Thames und Schaefer 2017

<sup>6</sup> Sun et al. 2022

<sup>7</sup> Ali und Johl 2023

### 2.1.2. IoT Framework in manufacturing

Figure 1 illustrates the IoT framework consists of three essential layers: perception, network, and application.

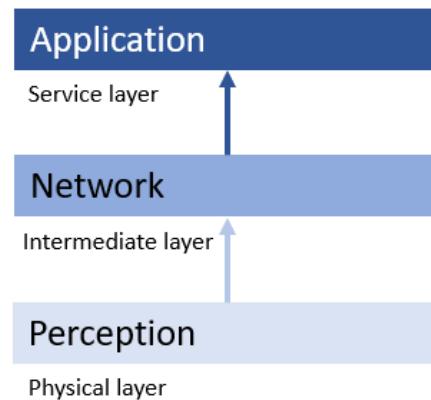


Figure 1: IoT Framework<sup>8</sup>

In manufacturing, the perception layer entails physical devices, such as sensor-equipped machines, responsible for gathering crucial process data pertaining to machine health and performance metrics. Figure 2 is an example of IoT framework, which provides more details for manufacturing. The network layer connects these physical devices with edge devices or cloud servers using suitable networking protocols. Data collected from machines undergo transmission to these edge devices, including computers, Arduino boards, or Raspberry Pis, before being relayed to cloud servers for comprehensive storage and analysis.<sup>9</sup>

Cloud servers serve as pivotal hubs managing and storing this data, acting as computational engines hosting applications that derive actionable insights. For instance, sensors continuously monitor tool conditions on machines, providing real-time insights crucial for informed decision-making.<sup>10</sup>

This established IoT framework empowers industries with real-time monitoring capabilities, which enables them to make data-driven decisions and facilitate predictive maintenance strategies. It forms a foundation for smart factories and industries, catalyzing innovation, fostering growth, and bolstering global competitiveness through intelligent, interconnected systems.<sup>11</sup>

<sup>8</sup> Surjya Kanta Pal 2022

<sup>9</sup> Surjya Kanta Pal 2022, p. 293–335

<sup>10</sup> Surjya Kanta Pal 2022, p. 293–335

<sup>11</sup> Surjya Kanta Pal 2022, p. 293–335

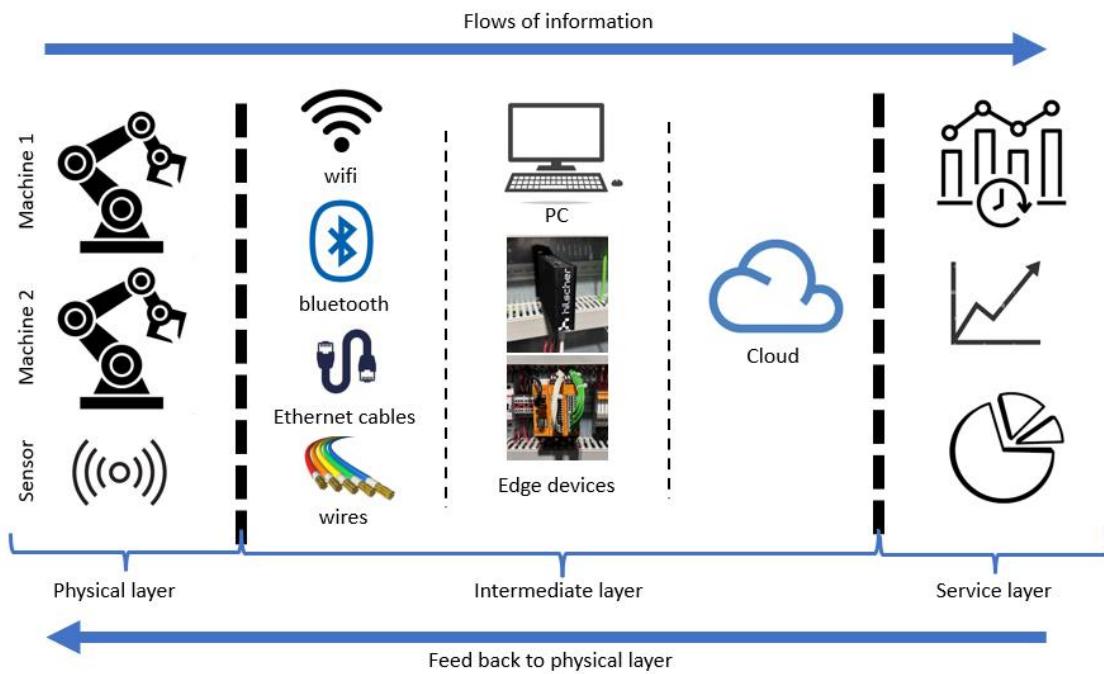


Figure 2: IoT framework in manufacturing<sup>12</sup>

## 2.2. Edge computing

Edge computing is a distributed computational paradigm that situates computational capabilities and data storage near data origination points. This configuration responds adeptly to the imperative requirements for minimizing latency in data processing and optimizing bandwidth utilization.<sup>13</sup>

The antecedents of Edge Computing can be traced back to the inception of Content Delivery Networks (CDNs) during the latter part of the 1990s.<sup>14</sup> These networks were initially developed to distribute web and video content from servers located in the immediate vicinity of end-users. As time passed, this infrastructure evolved significantly, adapting to accommodate the storage of applications and their integral components at edge servers. This critical evolution marked the beginning of early-stage edge computing services, which notably facilitated real-time data aggregation.<sup>15</sup>

### 2.2.1. Components of edge computing

Figure 3 illustrates how cloud computing concentrates critical services, such as servers, storage, databases, and applications are centralized in the cloud infrastructure,

<sup>12</sup> Surjya Kanta Pal 2022

<sup>13</sup> Cao et al. 2020, p. 85714–85728

<sup>14</sup> Ray Fernandez 2022

<sup>15</sup> Cao et al. 2020, p. 85717–85728

accessible across diverse devices via the Internet. This accessibility offers unprecedented flexibility and convenience, eliminating the necessity for individual server setups or reliance on local data centers, consequently leading to substantial cost reductions.<sup>16</sup>

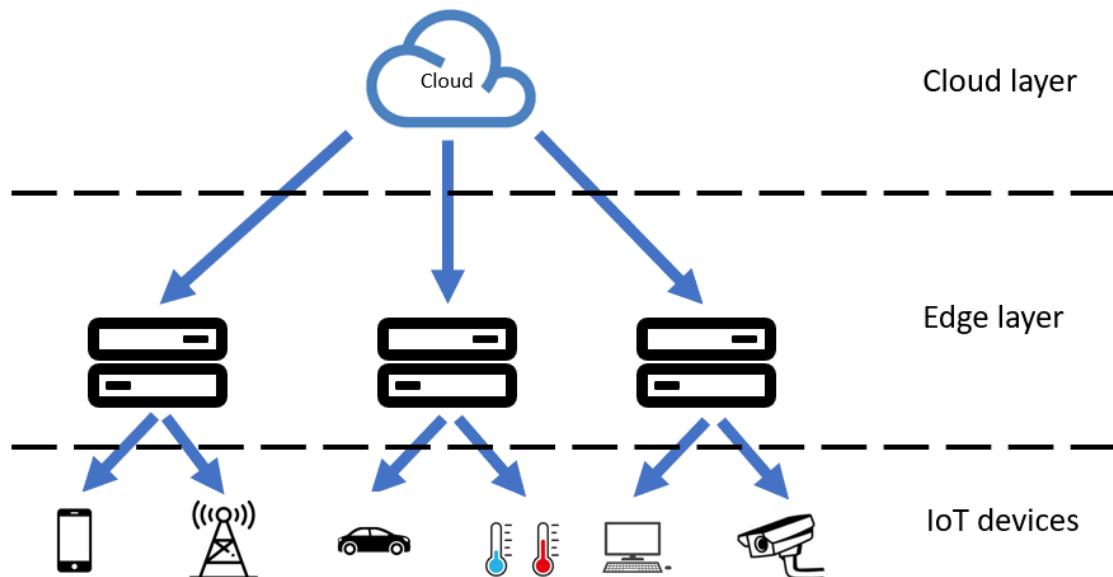


Figure 3: Edge computing: cloud to IoT devices<sup>17</sup>

In contemporary computing infrastructures, a hierarchical framework operates in which cloud servers function as central entities that manage applications and data. These servers are stationed in both public and private cloud services, as well as physical data centers, act as repositories for applications and orchestrate the management of multiple edge nodes across the network.<sup>18</sup>

Integrated computational entities known as edge devices, such as Automated Teller Machines (ATMs), surveillance cameras, and vehicles, are the primary focus of this framework. Despite their limited computational capabilities, these devices are optimized for immediate, low-latency tasks. They predominantly handle real-time data processing, functioning as primary data sources by performing localized computations before relaying crucial information to higher-tier computing layers.<sup>19</sup>

The classification of "edge nodes" encompasses various devices, including edge devices, edge servers, and edge gateways, which are intended to execute computational

<sup>16</sup> Verma und Kumar 2023, p. 121–155

<sup>17</sup> Avasalcai und Dustdar 2023

<sup>18</sup> Nguyen et al. 2020

<sup>19</sup> Cao et al. 2020

tasks within the realm of edge computing.<sup>20</sup> These nodes are located near data sources and conduct localized computations, which significantly diminish latency and enhancing response times in the decentralized architecture characteristic of edge computing.<sup>21</sup>

Specialized computational units known as edge servers are strategically positioned in operational settings like manufacturing plants, retail outlets, or distribution centers.<sup>22</sup> These servers are fitted with sturdy industrial-grade components, featuring powerful Central Processing Units (CPUs) equipped with 8 - 16 cores or more, substantial memory capacity exceeding 16GB, and considerable local storage. Their pivotal role involves adeptly managing enterprise application workloads, delivering shared services, and catering to specific localized processing needs.<sup>23</sup>

Edge gateways in edge computing environments serve a dual function by performing networking tasks and managing enterprise application workloads. Their responsibilities encompass protocol translation, firewall protection, wireless connectivity establishment, and management of fundamental networking operations.<sup>24</sup> The primary objective of edge gateways revolves around ensuring efficient data transmission, strong security measures, and seamless communication within the intricate fabric of edge computing environments.<sup>25</sup>

Cloud servers play a crucial role in managing applications and data by coordinating interactions with diverse components such as edge devices, nodes, servers, and gateways. This orchestrated collaboration optimizes data processing, reduces latency, and facilitates seamless communication by utilizing localized computational capabilities near the data sources.<sup>26</sup> This configuration effectively addresses the dynamic demands inherent in contemporary computing architectures.<sup>27</sup>

---

<sup>20</sup> Jazaeri et al. 2021, p. 3187–3228

<sup>21</sup> Dong et al. 2020, p. 313–320

<sup>22</sup> Dong et al. 2020, p. 313–320

<sup>23</sup> Carvalho et al. 2021, p. 993–1023

<sup>24</sup> Alabadi et al. 2022, p. 66374–66400

<sup>25</sup> Zhao et al. 2023, p. 431–437

<sup>26</sup> Ding 2022, p. 283–297

<sup>27</sup> Avasalcai und Dustdar 2023

### 2.2.2. Edge vs. Cloud vs. Fog

Cloud computing has transformed data storage practices, displacing traditional methodologies with a wide range of services such as: "pCloud", "OneDrive", "Box", "Dropbox", and "Google Drive". This transformative shift, facilitated by digitalization, enables secure and ubiquitous access to data regardless of time or location. Fundamental attributes which characterize these cloud services include secured accessibility via web-based platforms, adaptable scalability of resources, facile data sharing to foster collaborative efforts, and nuanced insights into resource utilization and management. In the field of manufacturing, cloud servers serve as custodians of data and applications, thereby facilitating remote process monitoring while leveraging the centralized computing power inherent in cloud infrastructure to achieve operational improvements.<sup>28</sup>

Fog computing represents a computational model, in which network switches and routers carry out analytics on network packets without sensor or application dependencies. This model represents the notion of edge computing, underscoring the proximity of computation to data generation sources. Fog computing, which synonymous with edge computing, signifies the execution of data analytics in immediate to the sources of data generation, thus constraining data to the local network.<sup>29</sup>

The implementation of fog computing requires the creation of a virtualized distributed computing infrastructure spanning across both edge devices and the cloud. In this framework, each network device hosts a software agent capable of dynamically downloading and executing analytics code.<sup>30</sup> However, the practical deployment of a fog computing framework encounters significant challenges arising from the processing power, memory, and battery power of edge devices, along with their unpredictable availability.<sup>31</sup> Furthermore, the sporadic availability of these devices further complicates the establishment of a universally deployable fog computing system. These complexities pose substantial impediments to realizing an extensively applicable fog computing architecture.<sup>32</sup>

---

<sup>28</sup> Vaibhav Sharma 2023; Kolekar und Sakhare 2023

<sup>29</sup> Foko Sindjoung et al. 2023; Surjya Kanta Pal 2022

<sup>30</sup> Elmansy et al. 2023

<sup>31</sup> Sohail et al. 2023

<sup>32</sup> Lin et al. 2023

Edge computing involves the execution of computational processes within a server located closer to specific machines, commonly referred to as the "edge server." The decentralized architecture of the edge server allows it to work in tandem with centralized cloud servers rather than replacing them.<sup>33</sup> This infrastructure consists of multiple nodes strategically positioned closer to physical devices, enabling immediate and transient data analysis near the sources. This proximity optimizes data transmission to monitoring platforms while reducing congestion within the cloud infrastructure. Fog servers act as intermediaries between physical devices and cloud servers, expediting data reception and alleviating network congestion.<sup>34</sup>

Furthermore, edge computing enables the provision of computing, storage, and networking services closer to the network's edge. The installation of an extra layer of nodes acts as a mediator between the cloud infrastructure and the IoT devices, comprising sensors and actuators. This edge layer encompasses a diverse array of distributed edge devices, including cloudlets, portable edge computers, and edge-cloud configurations, allowing for the deployment of applications in remote locations. Essential traits characterizing an edge device involve heterogeneity, mobility, and constraints in computational resources.<sup>35</sup>



In the context of Figure 2, edge servers or devices serve the purpose of conducting preliminary data processing tasks, such as data filtering or prompt predictive analysis. Moreover, network routers, switches, sensor nodes, and gateways have the potential to be used as computational nodes for analytics, effectively addressing scalability concerns inherent in IoT systems.<sup>36</sup> This approach, termed device edge computing, involves executing application-specific analytics on sensor data through sensor nodes and gateway devices for edge computing. By using edge devices for computation, IoT system scalability challenges are addressed, thereby enhancing both network and computational capabilities in a sophisticated manner.<sup>37</sup>

---

<sup>33</sup> Lin et al. 2023, p. 86–97; Zhang et al. 2023

<sup>34</sup> Zhang et al. 2023

<sup>35</sup> Lo und Niang 2023, p. 605–611

<sup>36</sup> Surjya Kanta Pal 2022

<sup>37</sup> Surjya Kanta Pal 2022

## 2.3. Edge computing - Catalyzing Data Processing Efficiency

This section provides an insight into the capabilities of edge computing in manufacturing. Edge computing represents a paradigm shift in data processing methodologies by strategically positioning cloud services closer to the source of data. This proximity ensures expedited processing and instantaneous responses, thereby optimizing system efficiency, particularly in time-critical scenarios.<sup>38</sup>

### 2.3.1. In Industry 4.0

The hallmark of edge computing is its scalability and adaptability, effectively managing substantial volumes of data across diverse applications, including supply chain tracking, point-of-sale systems, and distributed artificial intelligence. The versatility of edge computing highlights its relevance in constantly changing operational landscapes. In practical scenarios, especially in retail environments, edge computing proves to be a potent tool for optimizing computing resources. For instance, security cameras autonomously process local data, accurately categorizing and processing relevant information. This decentralized approach significantly reduces the load on central systems, resulting in significant improvements in operational efficiency.<sup>39</sup>

Beyond the realm of retail, the pervasive influence of edge computing extends across various industrial sectors, including manufacturing, energy, transportation, healthcare, and media.<sup>40</sup> With its capacity for localized data processing, it offers remarkable operational efficiencies that transcend conventional industry limitations. Combined with the transformative capabilities of 5G technology, edge computing plays a key role in reinforcing wireless networks. The establishment of local data centers adjacent to 5G towers facilitates rapid data transfer, amplifies device connectivity, and reduces latency, thus substantially amplifying the potential of 5G technology.<sup>41</sup>

In an era characterized by the proliferation of sensor-enabled smart environments, edge computing is emerging as an indispensable technology. When integrated with advanced technologies, such as 5G, it adeptly manages the growing

---

<sup>38</sup> Lin et al. 2023

<sup>39</sup> Sugumaran et al. 2023; Lin et al. 2023

<sup>40</sup> Zhang et al. 2021

<sup>41</sup> Baldoni et al. 2023, p. 51–61

volumes of data generated by smart cities, residences, and vehicle networks. This results in significant changes in lifestyle and interactions with technological systems.<sup>42</sup>

### 2.3.2. In manufacturing

Edge computing possesses inherent characteristics that significantly benefit manufacturing operations. The foremost attribute is its low latency, enabling real-time data processing in close proximity to the data source. In manufacturing contexts, instant processing is crucial, particularly in tasks like predictive maintenance, where instantaneous insights into machinery performance could prevent costly downtimes. The capacity for real-time processing facilitates swift decision-making on the production floor, optimizing operational efficiency and enabling prompt responses to dynamic operational demands. Additionally, the aspect of bandwidth efficiency differentiates edge devices by processing data locally and transmitting only essential information to centralized systems. This minimizes network congestion, alleviating data flow impediments within the manufacturing ecosystem.<sup>43</sup>

Practical examples demonstrate the effectiveness of edge computing in augmenting manufacturing processes. For instance, in scenarios where predictive maintenance is required, sensors embedded in machinery capture real-time data on key parameters. Edge devices process this data locally, enabling the immediate detection of anomalies or potential equipment failures.<sup>44</sup> Proactive identification of such issues averts disruptive downtimes, enabling optimized maintenance schedules and enhancing OEE. Figure 4 presents OEE data, including availability, performance, and quality. Another significant application is in quality control, where cameras and sensors stationed across production stages capture quality-relevant data. Edge computing enables immediate analysis, allowing quick adjustments to maintain stringent quality standards, minimize defects, and ensure product consistency.<sup>45</sup>

---

<sup>42</sup> Kong et al. 2022; Patrikar und Parate 2022; Dong et al. 2020

<sup>43</sup> Dong et al. 2020; Lin et al. 2023; Surjya Kanta Pal 2022; Verma und Kumar 2023

<sup>44</sup> Ringler et al. 2023; Mourtzis und Balkamos 2023

<sup>45</sup> Peniak et al. 2023

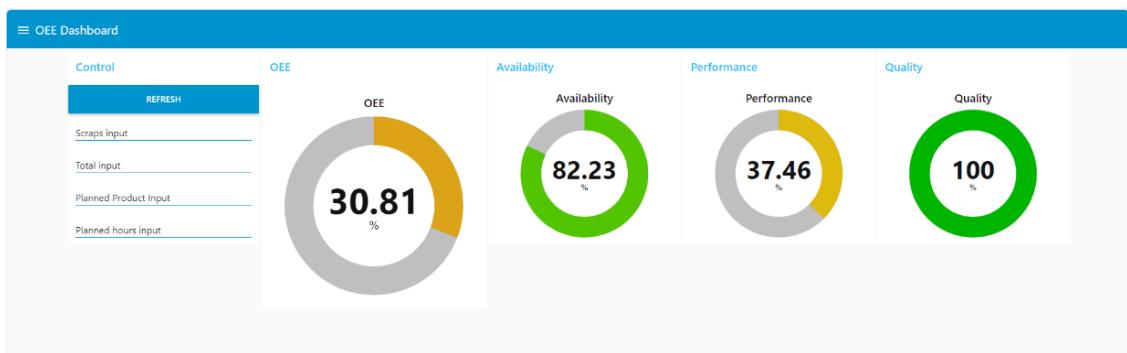


Figure 4: Real-time OEE Dashboard at 11:00 (UTC +1)

Furthermore, in logistics and supply chain management fields, edge devices integrated into inventory systems or vehicles track inventory, monitor stock levels, and optimize routes.<sup>46</sup> This local data processing by edge computing enhances logistics operations, refining delivery efficiency, curtailing costs, and ensuring timely execution of supply chain logistics. Moreover, edge devices drive adaptive production processes by adjusting manufacturing parameters in real-time, responding to variable demand or changing conditions.<sup>47</sup> This adaptability fosters flexible and agile manufacturing operations, optimizing resource allocation and production efficacy.<sup>48</sup>

These examples demonstrate how edge computing supports manufacturing by providing real-time insights, fortifying operational agility, and refining resource utilization. The strategic application of edge computing in these contexts emphasizes its pivotal role in transforming manufacturing operations, amplifying efficiency, and increasing the realization of Industry 4.0 principles.<sup>49</sup>

## 2.4. Challenges in manufacturing systems implementation

Although edge computing offers significant benefits in manufacturing, it still has limitations when implemented in this industry.

### 2.4.1. In Industry 4.0 implementation

The integration of Industry 4.0 in the manufacturing sector gives rise to numerous challenges across data management, machine learning utilization, and organizational adaptation. Data challenges encompass the complexity of acquiring and managing diverse

<sup>46</sup> Zhang et al. 2023

<sup>47</sup> Chai und Zeng 2023, p. 9448–9458

<sup>48</sup> Zhang et al. 2023

<sup>49</sup> Ringler et al. 2023, p. 1–6

data sources within manufacturing environments. These intricacies stem from disparate systems, incompatible formats, and diverse origins, including sensor-generated, machinery-related, and supply chain data. Maintaining data quality could become complicated due to data silos, inconsistencies, and incomplete entries. This could have an impact on overall accuracy and reliability.<sup>50</sup> Robust data security remains critical yet challenging, requiring comprehensive security measures and access controls to protect sensitive manufacturing data from cyber threats without impeding operational efficiency. Furthermore, integrating data from older systems into contemporary frameworks presents issues with data compatibility, affecting seamless data flow and interoperability which in turn weakens efficient data utilization.<sup>51</sup>

The use of machine learning in manufacturing confronts obstacles rooted in data scarcity, quality, and relevance. The scarcity of high-quality datasets, which suitable for training machine learning models, restricts their effectiveness in optimizing manufacturing processes. Moreover, the accuracy and effectiveness of these algorithms are impeded by data inaccuracies and biases. Therefore, it is necessary to thoroughly validate and curate data to ensure precise insights. Additionally, identifying and accessing relevant datasets that align with manufacturing objectives is challenging, which limits the application of advanced machine learning techniques in manufacturing environments.<sup>52</sup>

Furthermore, the adoption of Industry 4.0 technologies in manufacturing is hindered by organizational barriers. Resistance to technological change and cultural inertia pose significant challenges, impeding the seamless integration of contemporary technologies and methodologies.<sup>53</sup> Moreover, the shortage of skilled professionals proficient in managing advanced technologies, such as IoT, Artificial Intelligence (AI), and data analytics hampers successful integration efforts. Operational inefficiencies can undermine the potential optimization of manufacturing operations when workflows and processes are not adequately restructured to align with Industry 4.0 standards.<sup>54</sup>

---

<sup>50</sup> Nimawat und Das Gidwani 2022

<sup>51</sup> Mourtzis und Balkamos 2023

<sup>52</sup> Ringler et al. 2023; Mourtzis und Balkamos 2023

<sup>53</sup> Baque-Cantos et al. 2023

<sup>54</sup> Ali und Johl 2023, p. 838–871

### 2.4.2. Edge computing implementation

The integration of edge devices within the context of Industry 4.0 manufacturing poses a myriad of complex challenges necessitating meticulous examination. Foremost among these challenges are issues related to interoperability and integration complexity. Diverse edge devices frequently operate on disparate protocols or standards, leading to interoperability conundrums.<sup>55</sup> The optimization of manufacturing processes hinges crucially on efficient data exchange. Nevertheless, the task of facilitating seamless communication and integration between these devices, sensors, and existing legacy systems is marked by intricacy. The absence of standardized protocols stands as a significant impediment, impeding the establishment of streamlined connectivity and, consequently, hindering the enhancement of manufacturing operations.<sup>56</sup>

Another critical challenge is data security inherent to edge devices deployed in manufacturing environments. These devices located closer to operational areas become potential targets for cyber threats, which elevates concerns regarding data security. It is essential to protect sensitive manufacturing data while maintaining data processing efficiency.<sup>57</sup> Establishing robust security measures encompassing encryption, stringent access controls, and proactive threat detection mechanisms is vital to fortify critical data against breaches or unauthorized access.<sup>58</sup>

Ensuring the reliability and durability of edge devices is a significant challenge in the manufacturing context. These devices operate in rugged industrial settings that are characterized by adverse conditions such as temperature fluctuations, humidity, vibrations, and electromagnetic interference. Therefore, it is paramount to sustain reliability and durability under such harsh environments. The devices must demonstrate durability in these conditions while ensuring dependable performance, ensuring uninterrupted data processing and system functionality that are crucial for manufacturing operations.<sup>59</sup> Moreover, scalability and management complexity present significant challenges when managing a range of edge devices in manufacturing facilities. As the

---

<sup>55</sup> Ringler et al. 2023

<sup>56</sup> Peniak et al. 2023

<sup>57</sup> Foko Sindjoung et al. 2023

<sup>58</sup> Kolekar und Sakhare 2023

<sup>59</sup> Peniak et al. 2023

number of devices increases, efficient device management, monitoring, and maintenance become more intricate.<sup>60</sup>

## 2.5. Overall Equipment Effectiveness (OEE)

OEE has been extensively utilized in the industrial sector, particularly in the realm of factory maintenance. The gold standard is used to assess the level of utilization of a manufacturing operation in relation to its maximum capacity.<sup>61</sup>

The primary aim of OEE is to ascertain the efficiency of a certain asset or process line, and OEE has three key components: Availability refers to the amount of time that a machine is operational and accessible. Performance measures the actual production rate of the machine in comparison to the ideal or theoretical pace. Quality refers to the process of assessing and ensuring the control of the products' quality.<sup>62</sup> OEE classifies the notable deficiencies resulting from subpar performance and investigates the concealed capabilities of the production process. The implementation and enhancement of OEE also include the cooperation of machine operations, maintenance, management, engineering, and planning. The primary constituents of OEE are computed using the subsequent equation:<sup>63</sup>



Vietnamese-German University

$$\text{Availability} = \frac{\text{Run time}}{\text{Total planned time}}$$

$$\text{Performance} = \frac{\text{Current performance}}{\text{Theoretically performance}}$$

$$\text{Quality} = \frac{\text{Good products}}{\text{Total products}}$$

These measurements facilitate decision-making about equipment management and production planning. The OEE is determined using the following mathematical formula:<sup>64</sup>  $\text{OEE} = \text{Availability} \times \text{Performance} \times \text{Quality}$

---

<sup>60</sup> Foko Sindjoung et al. 2023; Ma et al. 2023

<sup>61</sup> Li et al. 2022

<sup>62</sup> S. Kalpande 2014

<sup>63</sup> S. Kalpande 2014

<sup>64</sup> Li et al. 2022

### 3. Methodology

The methodology for exploring the use of edge devices in manufacturing is divided into three parts. Firstly, the advantages of edge devices are highlighted. Secondly, assumptions are made based on the aforementioned advantages. Finally, research questions (RQs) are defined, and an overview of their implementation is provided.

#### 3.1. Background

In Peniak et al., industrial devices tend to produce a considerably larger amount of data compared to other IoT devices, which leads to delays and higher expenses when transferring data to the cloud. Minimizing reaction times to key events and maintaining unique security needs are essential in the industrial setting. Hence, transferring calculations to edge devices within industrial facilities can aid in mitigating these problems and enhancing the speed of reaction and efficiency of bandwidth. This implies that certain data processing and storage operations are transitioning from the cloud to edge.<sup>65</sup>

S. K. Pal et al. mentioned that edge device plays a crucial role in achieving the deployment of real-time monitoring and control of manufacturing processes.<sup>66</sup>

In terms of security, Keyan et al. outlined that conventional cloud computing necessitates the uploading of all data to the cloud for consolidated processing, which follows a centralized processing approach. During this procedure, there are potential hazards such as data loss and data leakage, which cannot ensure the security and confidentiality of information. Instances such as account passwords, history search information, and even commercial secrets have the potential to be divulged. Edge computing exclusively handles tasks inside its own domain, processing data locally without the need for cloud uploads. This eliminates the hazards associated with network transmission, ensuring data security. When data is subjected to an assault, it simply impacts the local data, rather than affecting all data.

#### 3.2. Approaching

To understand the advantages of the edge device for Industry 4.0 transformation in manufacturing, in this study, two edge devices are implemented in the manufacturing

---

<sup>65</sup> Peniak et al. 2023

<sup>66</sup> Surjya Kanta Pal 2022

environment. This implementation is aimed to stretch out the ability of edge devices to enhance machine control operations by facilitating faster and more accurate decision-making processes, leading to improved production efficiency. Moreover, these devices reduce data processing latency due to their computational capabilities and connecting directly to data sources. Lastly, their capacity to handle real-time data is going to increase production processes, allowing for immediate adjustments and enhanced quality control, therefore aligning closely with Industry 4.0 objectives.

A scoping study was carried out to delineate the main features of this project, offering valuable insights into the production process. Three RQs are outlined for the purpose of the project as follows:

How can edge device take part in monitoring manufacturing machine operation?

How does local data processing in edge computing mitigate the risks associated with network transmission?

In what ways does the shift to edge devices for data processing improve the effectiveness of real-time monitoring and control in manufacturing processes?

### 3.3. Define



This section will analyze and define three RQs mentioned in Section 3.2. Subsequently, an overview of two implementations will be presented to answer the following questions.

#### 3.3.1. Research Questions 1 and 2

In this project, it is aimed to express the ability of edge devices are integrated seamlessly into manufacturing infrastructure, gathering information directly from a manufacturing machine through wires. This integration is crucial for real-time monitoring since it tracks operational parameters, thereby detecting anomalies and preventing equipment failure and optimizing production efficiency. With the support of Node-RED, which is installed on the edge device, the data is then stored locally. Figure 5 illustrates the flows of data transferring from the manufacturing machine to the edge device.

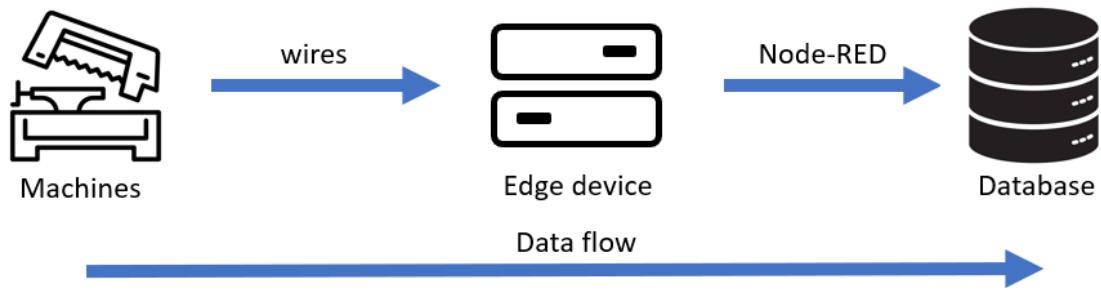


Figure 5: Overview of data flow in the implementation

This local processing should enhance data security significantly, mitigating risks like data loss and leakage that are prevalent in cloud computing models.

### 3.3.2. Research Question 3

Automation and data interchange have significantly enhanced the performance and efficiency of production through the implementation of Industry 4.0. An obstacle faced in Industry 4.0 is the ability to anticipate and avert anomalous actions inside the production procedure. Real-time data monitoring is increasingly crucial for factory control in terms of ensuring quality and performance.<sup>67</sup> The capability of edge devices to process data in real-time and calculate OEE is evident in the implementation, highlighting the ability to visualize performance, quality, and availability as any modifications occur.

During the implementation, there are some limitations in tracking real total of production and scraps and measuring the quality in the OEE. In that case, the total of production and scraps will be inputted by hand and the quality of the production is assumed to be random numbers which are not less than 90%. However, the total and scraps are still calculated with real-time. Figure 6 illustrates the overall view of the implementation.

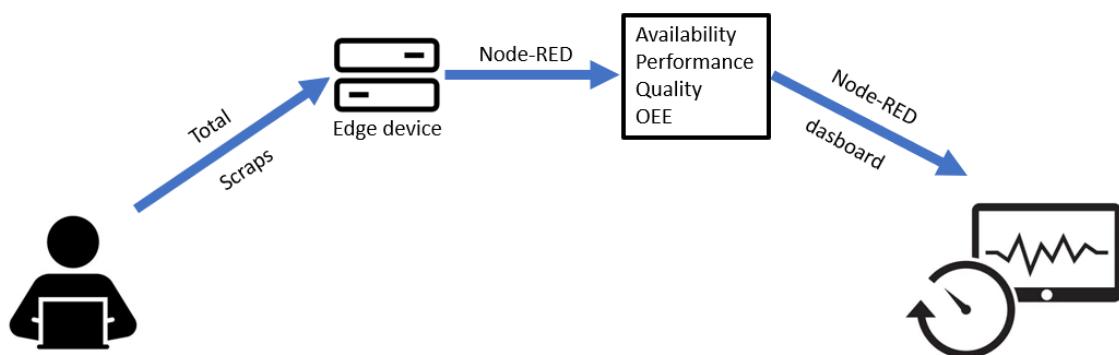


Figure 6: Outline of the implementation

<sup>67</sup> Li et al. 2022

## 4. Experimental Setup

The Lern – und Forschungsfabrik (LFF) is utilized for research, teaching, qualification, and industry collaboration purposes. It has equipped various manufacturing machines such as Computerized Numerical Control (CNC) machines, turning machines, milling machines as well as many automations, robotics, and digital manufacturing in an advanced industrial environment. In addition, LFF provides an opportunity to apply theoretical knowledge in a practical setting, working with machinery, tools, and technologies that are used in the manufacturing industry. Therefore, LFF is an appropriate place for this research to test and implementing new technologies from the IIoT.

### 4.1. Experimental Machine and Devices

Figure 7 illustrates two edge devices which are implemented in the LFF for research purposes. The netFIELD Compact X8M is an industrial edge computing device designed to fulfill the demanding requirements of Industry 4.0 applications. The device is constructed using the high-performance NXP i.MX 8M Mini Quad-core ARM Cortex-A53 processor, which offers significant computational capabilities for efficiently handling intricate tasks in real-time. The gadget is packed with 2 GB DDR4 RAM (Random Access Memory) and 32 GB eMMC storage, making it suitable for data-intensive applications.<sup>68</sup>



Figure 7: netFIELD Compact X8M (left) and RevPi Core (right) in LFF

<sup>68</sup> Hilscher Gesellschaft für Systemautomation mbH 2023

On the other hand, the Revolution Pi Core from KUNBUS is a series of industrial-grade computers based on the Raspberry Pi. These devices are specifically designed for use in industrial settings, where they modify the well-known and flexible Raspberry Pi platform for more demanding applications. The Revolution Pi Core models commonly have ARM processors, such as the Broadcom BCM2835, and are outfitted with standard Raspberry Pi interfaces, along with a set of GPIO pins for interfacing with industrial equipment.<sup>69</sup>

Both devices are equipped with Node-RED, a tool for implementing simple automation tasks and quickly creating prototypes. The Table 1 provides technical data of two edge devices.

*Table 1: Technical Data of edge devices*

	<b>NetFIELD Compact X8M<sup>70</sup></b>	<b>Revolution Pi Core<sup>71</sup></b>
<b>Processor</b>	NXP i.MX 8M Mini Quad-core ARM Cortex-A53	Broadcom BCM2835
<b>Clock speed</b>	1.8 GHz  Vietnamese-German University	250 MHz
<b>RAM</b>	2 GB DDR4	512 MB
<b>Storage</b>	32 GB eMMC	4 GB
<b>Operating system</b>	netFIELD OS on Linux aarch64 (ARMv8-A)	Raspbian image on Linux ARMv6
<b>Connectivity</b>	Ethernet	Ethernet
<b>Power supply</b>	24 V DC	12-24 V DC
<b>Dimension</b>	112 mm x 84 mm x 25 mm (without plugs and mounting bracket)  167 x 118 x 27 mm (with plugs and mounting bracket)	96 x 70 x 22 mm

<sup>69</sup> Revolution Pi n.d.

<sup>70</sup> Hilscher Gesellschaft für Systemautomation mbH 2023

<sup>71</sup> Broadcom Corporation 2012

## 4.2. Collecting data from manufacturing machine and storing data

Several machines at the LFF already have built-in IIoT capabilities, and communication issues may be resolved through software solutions. For collecting and storing data, the edge device, Revolution Pi Core (RevPi Core), a compact industrial computer, has been chosen. It serves as a flexible and robust solution for various industrial and automation applications. Revolution Pi Core is connected to RevPi DIO and RevPi DI devices, as shown in Figure 7, providing customizable Inputs/Outputs (I/O) options. These devices contribute to the collection of binary signals (0 and 1) from the bandsaw system. Figure 8 shows the JAESPA Bandsaw, which is implemented in LFF.



Figure 8: JAESPA Bandsaw

By integrating RevPi Core directly into the bandsaw's signals, the binary data is accurately tracked with no latency. The Node-RED function, installed on the RevPi Core, allows for real-time visual representation of the binary data. Table 2 displays the location of relevant bandsaw signals used for research and linked to RevPi DIO and RevPi DI.

Table 2: The I/O Addresses at the bandsaw and the I/O at the Rev Pi

ID	Name	Description	Bandsaw Address	I/O RevPi
2	Start	Start the bandsaw	X411	Input_DI_3
3	Stop	Stop the bandsaw	X412	Input_DI_4
5	NOT-AUS	Emergency stop button	X551	Input_DIO_5
21	Sägebügel senken	Cutting process starts. Cutting head moves down.	Y536	Input_DI_15
22	Sägebügel heben	Cutting process ended. Cutting head moves up.	Y537	Input_DI_16

Figure 9 illustrates the positions of the button to control the bandsaw. In this implementation, only three buttons are used, which are signed by numbers 2, 3, 5 in the Figure 9. The “Sägebügel senken” and the “Sägebügle heben” are based on the movement of the bandsaw. After connecting wires from the bandsaw to the Revpi Core to receive signal, they are configured in the RevPi Core web, which is able to access via the Internet Protocol (IP) Address 134.147.229.144, in order to ping in the interface via SSH (Secure Shell) connection of the Revpi Core.



Figure 9: ID from Bandsaw and configuration in PiCtory

#### 4.3. Computing and processing real-time data

The implementation of Industry 4.0 has significantly enhanced the productivity and effectiveness of the industrial sector via the utilization of automation and data interchange. Edge device has been crucial in facilitating the transition to Industry 4.0,

where data processing is interconnected via the IIoT.<sup>72</sup> Real-time OEE allows manufacturers to access a live analytics dashboard, an advanced alarm system, and ensures transparency in the production line.<sup>73</sup> In this implementation, with netFIELD Compact X8M, OEE demonstrates the capability of processing data in real-time on edge devices, hence enabling real-time computerization.

#### 4.3.1. OEE calculation

Availability: measures the actual runtime of the equipment compared to the planned production time. It considers the Availability Loss, which is downtime (including equipment failures, maintenance, unplanned stops, and changeover time). The formula of Availability is:

$$\text{Availability (\%)} = (\text{Actual runtime} / \text{Planned Production time}) \times 100\%$$

Actual runtime is also calculated as: Planned Production time – Downtime

In LFF – a factory in Lehrstuhl für Produktionssystem (LPS) for research and testing industrial application scenarios, it has been empirically determined that the average Availability of equipment is 82.235%.



Performance: is calculated based on the ability of a manufacturing process to shift products during specific time intervals, usually measured within an 8-hour working day. It gauges the attainment of planned production targets and gradually decreases when no products are produced.

Quality rate: considers the number of good-quality products produced in relation to the total number of products produced.

$$\text{Quality rate (\%)} = (\text{Good Products} / \text{Total Products}) \times 100\%$$

#### 4.3.2. Live monitoring

With the features that Node-RED has provided, the Node-RED-powered live OEE Monitoring Dashboard represents a new model in the continuous quest for manufacturing excellence. It gives an immediate window into production processes, offering critical insights into the optimization of efficiency, the minimization of disruptions, and the fostering of continuous improvement.

---

<sup>72</sup> Zhang et al. 2021

<sup>73</sup> Li et al. 2022

## 5. Challenges in Implementing edge devices

Before discussing the implementation, this chapter will highlight several challenges that arose during the process. The mentioned problems took a significant amount of time to resolve during installation.

### 5.1. Competencies in programming languages

To serve and reach the purpose of the research, general knowledge about coding languages is necessary. Two main challenges in coding are pointed out during the implementation.

Coding often involves abstract concepts that can be difficult to grasp, such as algorithms, data structures, and design patterns. It is important to maintain a clear and logical structure when writing about coding concepts to ensure comprehension.<sup>74</sup>

Moreover, one of the most significant challenges in learning to code is debugging. Identifying and fixing errors can be time-consuming and frustrating, especially when errors are cryptic, or the source of the problem is unclear. It may take time to read through the code to track where errors are occurring, especially in tracking time for OEE in this research.<sup>75</sup>



### 5.2. Lack of memory

Initially, there was no information available regarding the username and password from previous testing and implementation. As a result, the device was reset to its factory settings and updated with the latest image available on the KUNBUS website. After several experiences with RevPi Core, including getting familiar with the interface, working with the Linux operating system, accessing via SSH connection, and downloading IoT applications, some issues arose when using the edge device. However, it was discovered that the RevPi core had run out of memory, despite being flashed previously. Figure 10 illustrates the status of memory of the RevPi Core.

---

<sup>74</sup> Sakshi Gupta 2023

<sup>75</sup> Sakshi Gupta 2023

```
pi@RevPi12880:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       3.3G  3.1G    0 100% /
devtmpfs        183M    0  183M   0% /dev
tmpfs          215M    0  215M   0% /dev/shm
tmpfs          215M   25M 191M  12% /run
tmpfs          5.0M  4.0K 5.0M   1% /run/lock
tmpfs          215M    0  215M   0% /sys/fs/cgroup
/dev/mmcblk0p1   253M   43M 210M  17% /boot
tmpfs          43M    0   43M   0% /run/user/1000
```

Figure 10: Checking storage in RevPi Core through SSH connection

To find the root of the problem, the image package has been checked. The previously installed image package has nearly filled the storage capacity of RevPi Core. In order to reduce the amount of memory, the lite version of that image package was installed. As shown in Figure 11, the storage in /dev/root has been greatly reduced.

```
pi@RevPi12880:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       3.4G  1.8G  1.4G  58% /
devtmpfs        183M    0  183M   0% /dev
tmpfs          215M    0  215M   0% /dev/shm
tmpfs          215M   25M 191M  12% /run
tmpfs          5.0M  4.0K 5.0M   1% /run/lock
tmpfs          215M    0  215M   0% /sys/fs/cgroup
/dev/mmcblk0p1   253M   43M 210M  17% /boot
tmpfs          43M    0   43M   0% /run/user/1000
pi@RevPi12880:~ $
```

Figure 11: Storage in RevPi Core after re-install image package

For other applications that are missing in the lite version and required for implementation, they can be installed manually later. Figure 12 illustrates the memory of the full version image package and its lite version.

2023-01-12-revpi-buster	06.02.2024 11:45	Datenträgerimagedatei	3.318.568 KB
2023-01-12-revpi-buster-lite	19.09.2023 16:04	Datenträgerimagedatei	1.563.064 KB

Figure 12: Memory of the image package

## 6. Implementation and Result

Following the experimental set up, this chapter will present the details of the implementation using two edge devices in LFF. The first implementation demonstrates how the edge device can track signals from a manufacturing machine. The second implementation showcases the edge device's ability to process real-time data and provide live monitoring.

### 6.1. Network preparation

This section explains how to access an edge device using an IP address and discusses the safety of storing data locally on the device.

#### 6.1.1. RevPi Core

The Rev Pi is equipped with an operating system that runs on a Raspbian OS. Hence, all communication originating from the RevPi Core and transmitted to the network is comparable to that of a typical computer or Raspberry Pi. The MAC address of the RevPi Core is associated with the static IP address 134.147.229.144 within the network. Since the connections are restricted to the Local Network, there is no requirement to configure data security. Access to the operating system can be achieved using an SSH connection, which is protected by a login and password. In addition, the user interface for configuring the input and output settings through web browser is protected by a username and password.

#### 6.1.2. NetFIELD Compact X8M

NetFIELD Compact is an edge gateway device, which allow user to deploy workloads in close proximity to industrial machinery, allowing for the dissemination of IIoT logic and intelligence.

When netFIELD Compact is connected to the netFIELD Cloud through the internet, user is able to administer device remotely through the netFIELD.io Portal. The netFIELD Portal serves as the web-based user interface for the netFIELD Cloud. This connectivity enables the remote management of IIoT applications and facilitates the control and distribution of these applications over the internet. The box's MAC address is linked to the static IP address 134.147.229.182 on the network. Access to the operating system can be attained through the netFIELD.io, which required login and password from

both netFIELD account which is created in netFIELD.io and another account from the device itself.

## 6.2. JAESPA Bandsaw

The Figure 13 depicts the transmission of binary data for the bandsaw, as presented in Table 2. The RevPi Core gathers all the data and then transmits it to Node-RED, a flow-based programming tool used to integrate these separate data streams. Node-RED processes and potentially improves the data prior to transmitting it to InfluxDB, a specialized database designed for efficient storage and retrieval of time-stamped data. InfluxDB facilitates instantaneous analysis and monitoring of the bandsaw's operational and performance metrics.

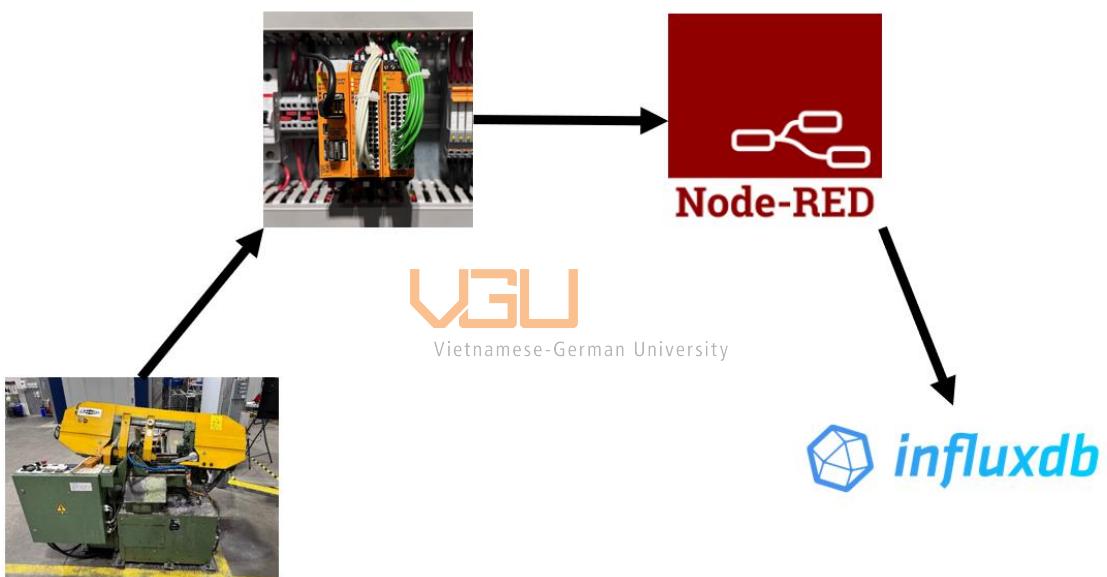


Figure 13: Data flow for bandsaw

### 6.2.1. Collecting data

The Node-RED visual editor, which has been installed in RevPi Core, is able to access via the IP Address 134.147.229.144:1880. In order to create a working configuration for a RevPi system using Node-RED, it is necessary to install two essential node packages: “node-red-contrib-revpi-nodes” and “node-red-contrib-influxdb”. After installation, the “node-red-contrib-revpi-nodes” is set up to establish communication with the RevPi Core by configuring the accurate IP address, hence enabling the transmission of signal data from the device. In Figure 14, a “inject timestamp” node is set up to activate every 5 seconds, commencing the process of collecting data at consistent intervals.

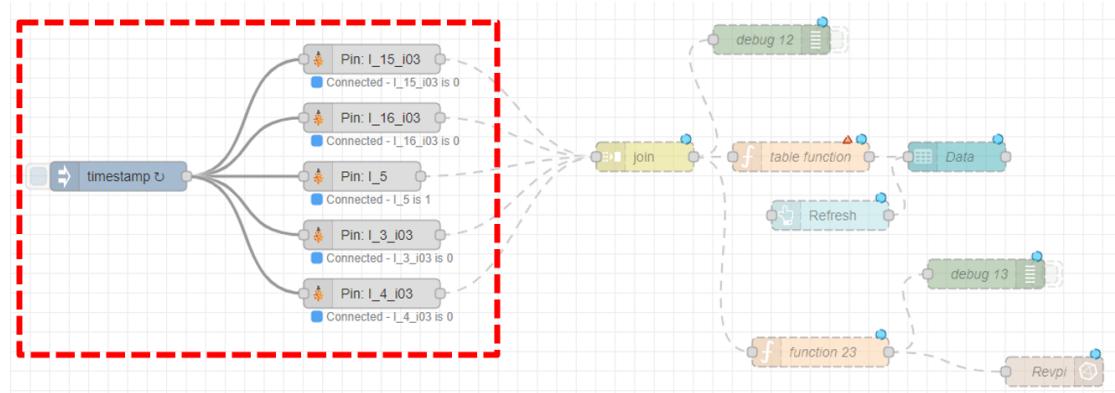


Figure 14: Config for collecting data from Bandsaw

The “revpi-getpin” node is configured to the corresponding input pin as shown in Figure 15. The server is set as 134.147.229.144:8000. Once access to the RevPi Core is obtained, select the “Input Pin” corresponding to the input in Table 2.

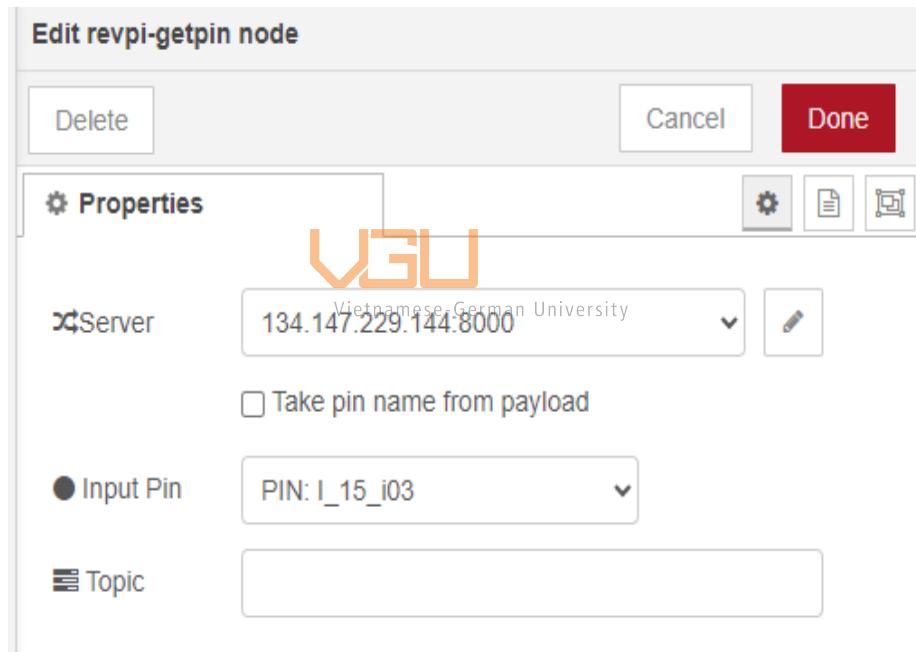


Figure 15: Config IP and Input signal of RevPi Core

### 6.2.2. Storing and visualizing data

The data will be stored locally in InfluxDB, which exclusively accepts and stores data in numerical format. Once the data has been consolidated into a unified message and formatted correctly, it can be further processed or stored. The “influxdb out” is set up with a suitable server for data storage as shown in Figure 16.

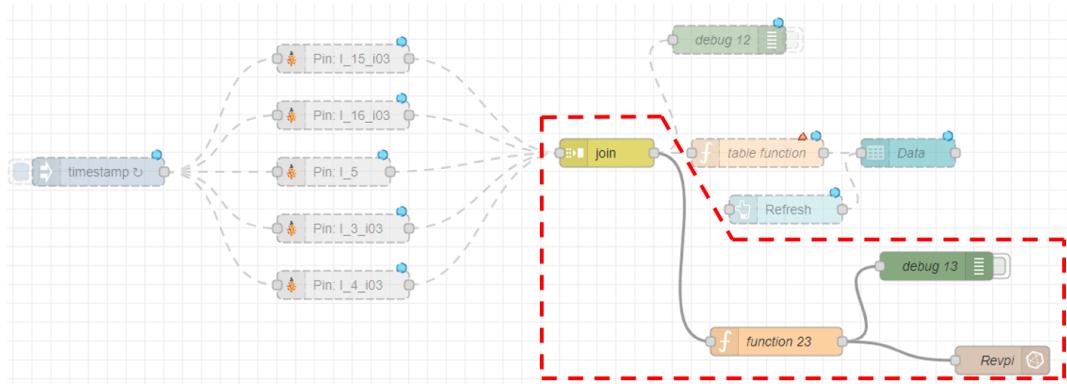


Figure 16: Flows for storing data into InfluxDB

To convert the data from “revpi-getpin” from text format to numeric format, it is necessary to include a “function 23” node in the flow. Code 1 illustrates the code that use to transform the data from “string” type to numeric.

```
msg.payload = {
    "notaus" : parseFloat(msg.payload["revpi/single/I_5"]),
    "start" : parseFloat(msg.payload["revpi/single/I_3_i03"]),
    "stop": parseFloat(msg.payload["revpi/single/I_4_i03"]),
    "ssenken": parseFloat(msg.payload["revpi/single/I_15_i03"]),
    "sheben": parseFloat(msg.payload["revpi/single/I_16_i03"])
}
return msg;
```

Vietnamese-German University

Code 1: Convert from string to number

Furthermore, the data is presented through a Node-RED dashboard, which offers immediate feedback on production processes, facilitating prompt decision-making and efficiency improvements. In Figure 17, the “table function” node contains code which has been shown in Code 2, creates a table for displaying the data to the “ui\_table” node after joining all five signals from the RevPi Core. Additionally, the “refresh” button node updates the data when clicked.

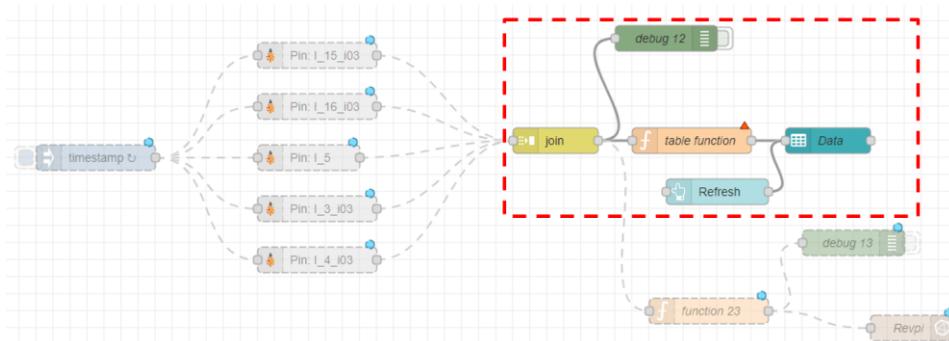


Figure 17: Flow for displaying the dashboard

In Code 2, the code begins by retrieving any previously stored data from the flow's context under the key savedData. If no data is found, it defaults to an empty array. Subsequently, it creates a new object with a timestamp and several other properties that correspond to the RevPi's I/O pin states. This new object is then added to the beginning of the tableData array using unshift(). The updated tableData array is then reassigned for future use in the flow. Finally, the table has the updated tableData, prepared for the next node in the flow.

```
tableData = flow.get("savedData") || [];
tableData.unshift({
  "timestamp" : new Date().toLocaleString(),
  "NOTAUS" : msg.payload["revpi/single/I_5"],
  "Start" : msg.payload["revpi/single/I_3_i03"],
  "Stop" : msg.payload["revpi/single/I_4_i03"],
  "Sägebügel senken" : msg.payload["revpi/single/I_15_i03"],
  "Sägebügel heben" : msg.payload["revpi/single/I_16_i03"],
});
msg.payload = tableData;
flow.set("savedData", tableData);
return msg;
```



Vietnamese-German University

Code 2: Creating table

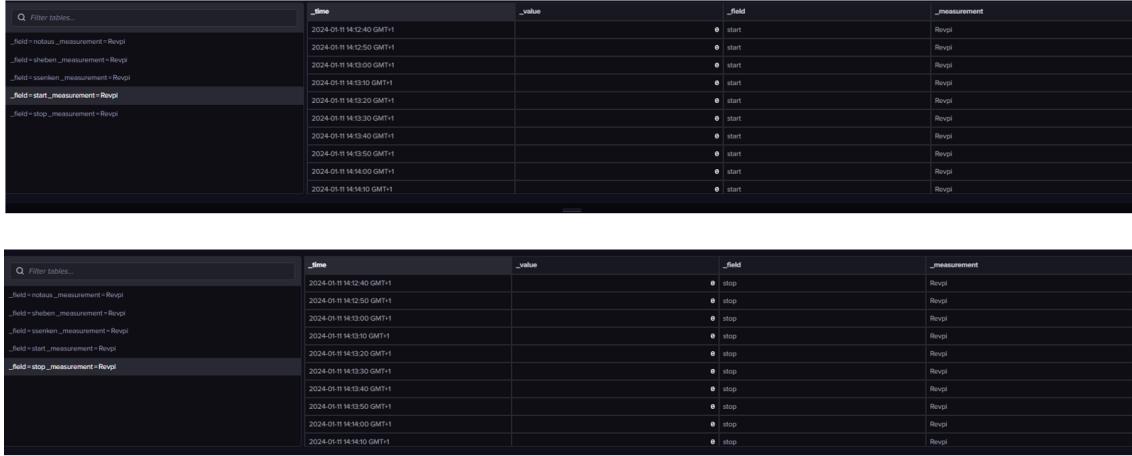
### 6.2.3. Result

The InfluxDB, which has been installed in RevPi Core, is able to access via the IP Address 134.147.229.144:8086. In addition, to access the database in InfluxDB, username and password is required. To visualize the data, InfluxDB requires some commands for selecting and sorting in database, which is shown in Code 3. Figure 18 shows data that has been retrieve from the RevPI Core, which is automatically updated.

```
from(bucket: "RUB")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "Revpi")
  |> filter(fn: (r) => r["_field"] == "notaus" or r["_field"] ==
  "sheben" or r["_field"] == "ssenken" or r["_field"] == "start" or
  r["_field"] == "stop")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty:
  false)
  |> yield(name: "mean")
```

Code 3: Selecting data in InfluxDB

In Code 3, the InfluxDB code is specifically intended for retrieving and processing data from the bucket "RUB". The query filters the data to a specific time range which are "v.timeRangeStart" and "v.timeRangeStop". The "Revpi" measurement then filters for records whose field has one of five predefined values: "notaus", "sheben", "stop", "start", or "ssenken". After that, it uses an aggregate window to average (mean) the data points for each window period specified by "v.windowPeriod". This aggregate does not produce empty windows if there are no data points (createEmpty: false).



The image shows two screenshots of an InfluxDB interface. The top screenshot displays the 'Input START' table with the following data:

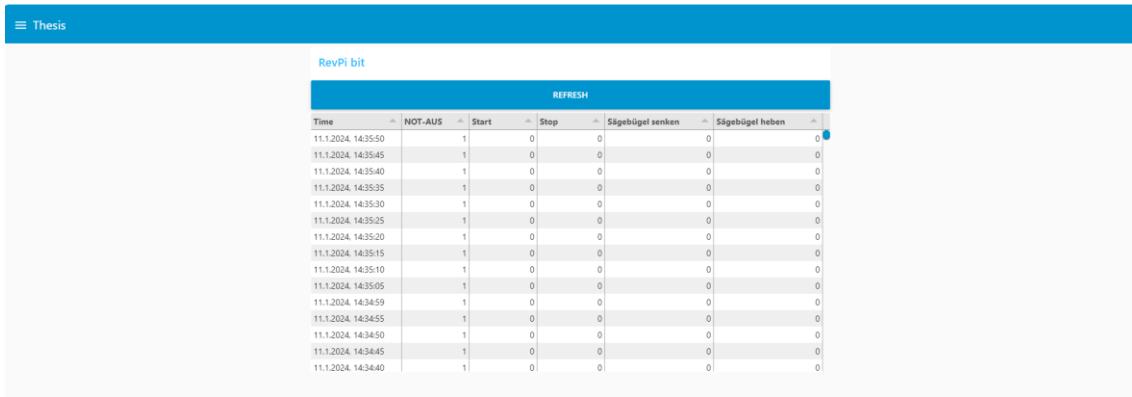
time	_value	_field	_measurement
2024-01-11T14:12:40 GMT+1	0	start	Revpi
2024-01-11T14:12:50 GMT+1	0	start	Revpi
2024-01-11T14:13:00 GMT+1	0	start	Revpi
2024-01-11T14:13:10 GMT+1	0	start	Revpi
2024-01-11T14:13:20 GMT+1	0	start	Revpi
2024-01-11T14:13:30 GMT+1	0	start	Revpi
2024-01-11T14:13:40 GMT+1	0	start	Revpi
2024-01-11T14:13:50 GMT+1	0	start	Revpi
2024-01-11T14:14:00 GMT+1	0	start	Revpi
2024-01-11T14:14:10 GMT+1	0	start	Revpi

The bottom screenshot displays the 'Input STOP' table with the following data:

time	_value	_field	_measurement
2024-01-11T14:12:40 GMT+1	0	stop	Revpi
2024-01-11T14:12:50 GMT+1	0	stop	Revpi
2024-01-11T14:13:00 GMT+1	0	stop	Revpi
2024-01-11T14:13:10 GMT+1	0	stop	Revpi
2024-01-11T14:13:20 GMT+1	0	stop	Revpi
2024-01-11T14:13:30 GMT+1	0	stop	Revpi
2024-01-11T14:13:40 GMT+1	0	stop	Revpi
2024-01-11T14:13:50 GMT+1	0	stop	Revpi
2024-01-11T14:14:00 GMT+1	0	stop	Revpi
2024-01-11T14:14:10 GMT+1	0	stop	Revpi

Figure 18: Data for Input START (above) and Input STOP in InfluxDB (below)

Figure 19 illustrates couples of data from Revpi Core that are displaying in the Node-RED dashboard. Compares to the data in InfluxDB, although the data that display with Node-RED backend is not required couple lines of code to visualize it, it needs to have the "refresh" button to update the data.



The image shows a screenshot of a Node-RED dashboard. The top navigation bar has a 'Thesis' tab. The main area features a table titled 'RevPi bit' with the following data:

Time	NOT-AUS	Start	Stop	Sägebügel senken	Sägebügel heben
11.1.2024, 14:35:50	1	0	0	0	0
11.1.2024, 14:35:45	1	0	0	0	0
11.1.2024, 14:35:40	1	0	0	0	0
11.1.2024, 14:35:35	1	0	0	0	0
11.1.2024, 14:35:30	1	0	0	0	0
11.1.2024, 14:35:25	1	0	0	0	0
11.1.2024, 14:35:20	1	0	0	0	0
11.1.2024, 14:35:15	1	0	0	0	0
11.1.2024, 14:35:10	1	0	0	0	0
11.1.2024, 14:35:05	1	0	0	0	0
11.1.2024, 14:34:59	1	0	0	0	0
11.1.2024, 14:34:55	1	0	0	0	0
11.1.2024, 14:34:50	1	0	0	0	0
11.1.2024, 14:34:45	1	0	0	0	0
11.1.2024, 14:34:40	1	0	0	0	0

Figure 19: Data visualized in Node-RED

### 6.3. Computing with real-time data

The backend of Node-RED on netFIELD Compact X8M can be reached by using the IP Address 134.147.229.182:1880. To establish an OEE dashboard using Node-RED, it is imperative to install the "node-red-dashboard" node package. This package facilitates the creation of input data for OEE and enables live monitoring of the OEE dashboard. Figure 20 show an overview of Node-RED flows for calculating and live monitoring OEE dashboard.

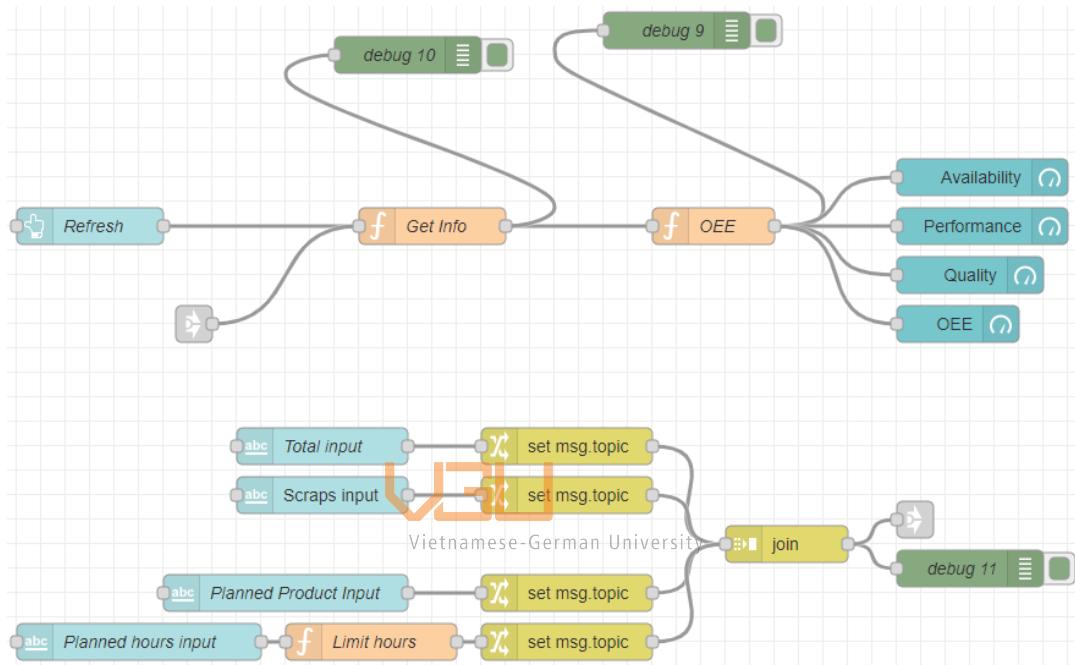


Figure 20: Node-RED flows for calculating and live monitoring OEE dashboard

#### 6.3.1. Phase 1 - Setting input for OEE calculation

Figure 21 illustrates the inputs required for an Overall Equipment Efficiency (OEE) calculation. The "Total input" and "Scrap input" nodes presumably gather data on the overall production and scrap quantities. The "Planned Product Input" and "Planned hours input" nodes can be utilized to enter the expected production output and the planned production time respectively. The "Limit hours" node is set up to limit the input of the "Planned hours input" to a maximum of 8 hours per day, ensuring that the working hours do not exceed this restriction.

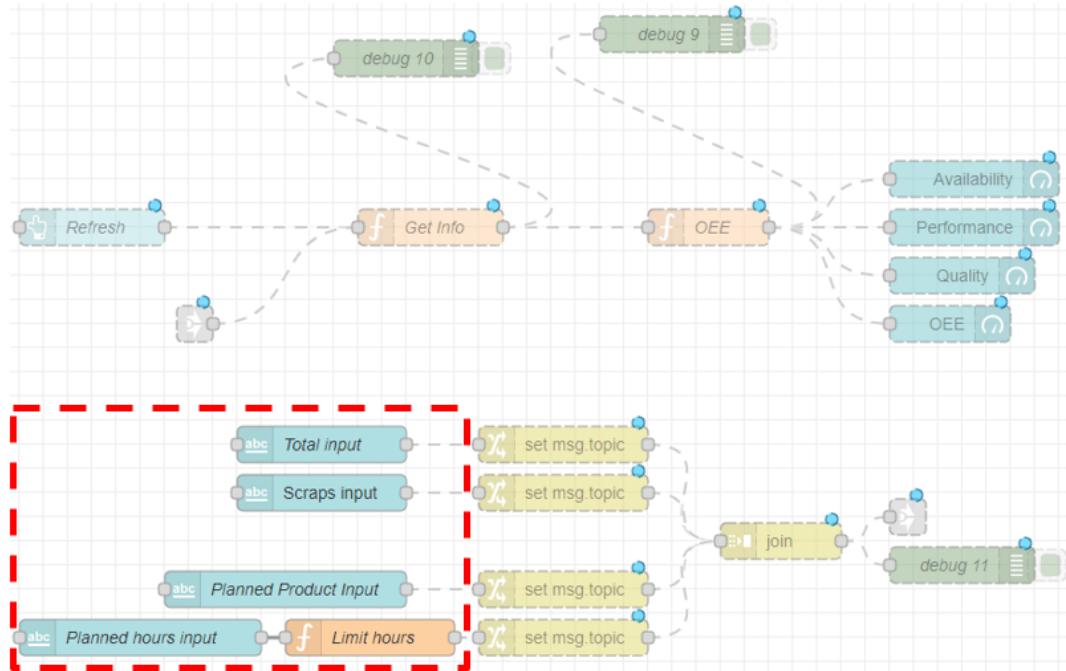


Figure 21: Config flows for Input

As shown in Code 4, the code begins by converting the output from “Planned hours input” node to a numeric value and assigning it as “input”. If there is no output from the “Planned hours input”, the default value of 8 is used. The code thereafter defines a valid range for the “input”, bounded by the minimum value of 0 and the maximum value of 8. If the “input” is a numerical value and is within the required range, the code proceeds by returning the message object with the payload attribute assigned to the input value. If the input falls beyond the permissible range, it will generate a warning in the Node-RED environment, highlighting the problem and terminating the flow by returning null, therefore rejecting the “input”.

```

var input = parseFloat(msg.payload) || 8;
var min = 0;
var max = 8;
if (!isNaN(input) && input >= min && input <= max) {
    msg.payload = input;
    return msg;
} else {
    node.warn("Input is outside the allowed range (" + min + " - " + max
+ ")");
    return null;
}

```

Code 4: "Limit hours" configuration for working hour

Figure 22 illustrates the inputs are further handled by “set msg.topic” nodes, which may format the incoming data with certain subjects for further downstream processing.

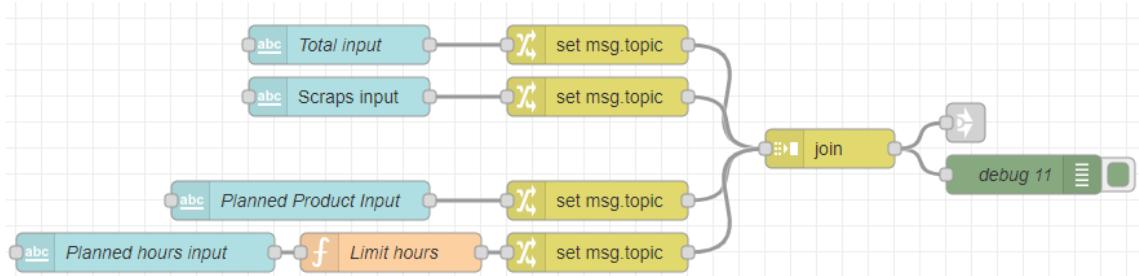


Figure 22: The whole Node-RED flows for input quantities to OEE

The "join" node is set to manually combine four message parts, each identified by a unique “msg.topic”, into a single key/value object as shown in Figure 23, and it will output a message when all parts have been received for a comprehensive OEE computation. The message will be sent to “Get info” node through the “link out” .

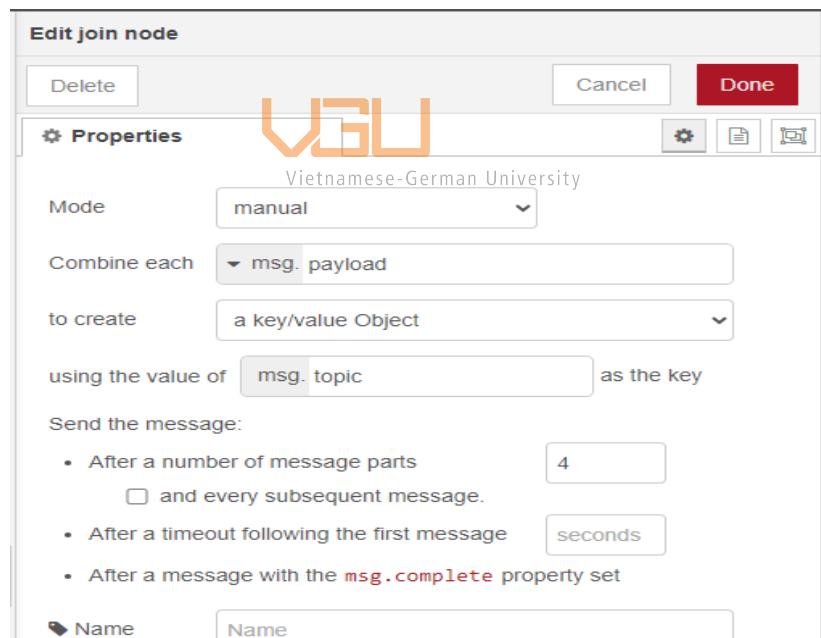


Figure 23: Config join node to join 4 Inputs into one message

### 6.3.2. Phase 2 - Calculate OEE

Figure 24 illustrates the overall view of phase 2 flow to calculate OEE. There are two separate ways of data that the node “Get Info” will receive in this phase. The first one is “Refresh” node, which will let the “Get Info” send the default data to calculate OEE. The second method involves the “Get Info” function gathering data from the specified inputs in Phase 1 in order to compute OEE.

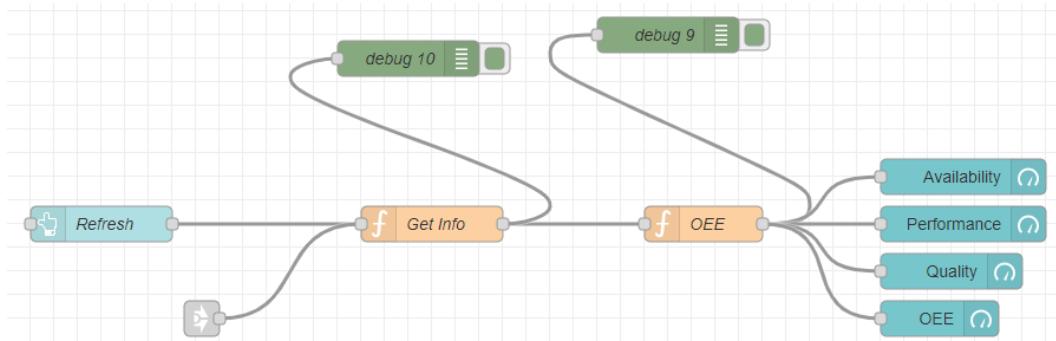


Figure 24: Gathering data and calculate OEE

Code 5 shows the code retrieves the timestamp from the system by the time it receives data from the “Refresh” button node or acquires data from the Phase 1 and then it establishes the current\_date as the beginning of the day (midnight) by resetting the hours, minutes, seconds, and milliseconds to zero. The work\_start\_time is calculated by adding a fixed offset of 21600000 milliseconds (equivalent to 6 hours) to the midnight timestamp, which reflects the usual start time for a shift. The code also establishes predetermined values of 7200 for planned\_production and 8 for planned\_hours, which may be replaced by values from incoming message in Phase 1. In addition, if there is no available data for the total and scraps, they will be automatically set to 0.

The script then calculates the current\_worktime by determining the number of hours that have passed since the start of the workday. The program incorporates a provision to account for shifts that commence after 3 pm (equivalent to 54000000 milliseconds after midnight), by adjusting the values of work\_start\_time and current\_worktime accordingly. After getting the time in the “Get Info” node, the data then transfers to “OEE” node to calculate.

```

var current_timestamp = (new Date()).getTime();
var current_date = new Date(current_timestamp);
current_date.setHours(0, 0, 0, 0);
var starttime = current_date.getTime();
var work_start_time = starttime + 21600000;
var planned_production = parseFloat(msg.payload.product) || 7200;
var planned_hours = parseFloat(msg.payload.hours) || 8;
var total = parseFloat(msg.payload.total) || 0;
var scraps = parseFloat(msg.payload.scaps) || 0;
var current_worktime = (current_timestamp - work_start_time) / 3600000;
if (current_timestamp > starttime + 54000000) {
    work_start_time = starttime + 54000000;
    current_worktime = (current_timestamp - work_start_time) / 3600000;
};

```

```

msg.payload =
{
  "resettime": starttime,
  "currenttime": current_timestamp,
  "starttime": work_start_time,
  "currentworktime": current_worktime,
  "total": total,
  "scraps": scraps,
  "product": planned_production,
  "hours": planned_hours
}
return msg;

```

*Code 5: Code for getting timestamp and receiving data from "Refresh" button node or from Phase 1*

The process of calculating OEE is started by defining variables for work\_start\_time, current\_time, and the time elapsed since a reset point as shown in Code 6. These variables are taken from the message payload obtained from the “Get Info” node. The current\_worktime is used along with planned\_hours and planned\_production to determine the current\_plan\_production rate.



```

//Plan:
var work_start_time = msg.payload.starttime;
var current_time = msg.payload.currenttime;
var difftime = (current_time - msg.payload.resettime) / 3600000;
var current_worktime = msg.payload.currentworktime;
var planned_production = parseFloat(msg.payload.product);
var planned_hours = parseFloat(msg.payload.hours);
var current_plan_production = ((current_worktime) / planned_hours) *
planned_production;

```

*Code 6: Define time variables*

Furthermore, in this OEE calculation, Available is assumed a fixed number of 82.235 and Quality is a variable that is not affected by the time according to Section 4.3.1, the calculation of the difftime is then used to set the Performance to 0 from midnight until a new shift is started at 6 o'clock in the morning. In Code 7, the code calculates performance based on the ratio of actual total production (total) to the planned production rate (current\_plan\_production). If there has been no production yet, performance is determined by calculating the percentage deviation from the projected production. Consequently, the performance will gradually decline over time in the absence of output during a shift.

```

//AVAILABILITY:
var availability = 82.235;
//PERFORMANCE:
var total = parseFloat(msg.payload.total);

if (total === 0) {
    var performance = (1 - current_plan_production / planned_production)
* 100;
} else {
    performance = (total / current_plan_production) * 100;
};
if (difftime >= 0 && difftime < 6) {
    performance = 0;
};

```

Code 7: Calculating for Availability and Performance

Because there is no real data for the shift, the code simulates a quality assurance process in which a random mechanism determines if a product is good in this research as shown in Code 8. The quality percentage is determined by dividing the number of good products by the total number of products, taking into account any defective items. Furthermore, the quality is assumed to be 100 if there is no input of both total and scraps from the Phase 1.

Vietnamese-German University

```

//QUALITY:
var goods = 0;
var scraps = parseFloat(msg.payload.scraps);
var quality = 100;
if (total !== 0 && scraps === 0) {
    for (var i = 0; i <= total; i++) {
        var randomNumber = Math.round(Math.random() * 5);
        if (randomNumber !== 0) {
            goods++;
        }
    }
    quality = (goods / total) * 100;
} else if (total !== 0 && scraps !== 0) {
    quality = ((total - scraps) / total) * 100;
};

//OEE:
var oee = (availability * performance * quality) / 10000;

```

Code 8: Calculating for Quality and OEE

After calculating all three factors of OEE, the script computes the final OEE by multiplying the three percentages and then dividing by 10,000 to re-scale to a percentage format. Code 9 below illustrates a full script of OEE calculation.

```
//Plan:
var work_start_time = msg.payloadstarttime;
var current_time = msg.payload.currenttime;
var difftime = (current_time - msg.payload.resettime) / 3600000;
var current_worktime = msg.payload.currentworktime;
var planned_production = parseFloat(msg.payload.product);
var planned_hours = parseFloat(msg.payload.hours);
var current_plan_production = ((current_worktime) / planned_hours) *
planned_production;

//Calculation OEE:

//AVAILABILITY:
var availability = 82.235;

//PERFORMANCE:
var total = parseFloat(msg.payload.total);
if (total === 0) {
    var performance = (1 - current_plan_production / planned_production) *
100;
} else {
    performance = (total / current_plan_production) * 100;
};
if (difftime >= 0 && difftime < 6) {
    performance = 0;
};

//QUALITY:
var goods = 0;
var scraps = parseFloat(msg.payload.scraps);
var quality = 100;
if (total !== 0 && scraps === 0) {
    for (var i = 0; i <= total; i++) {
        var randomNumber = Math.round(Math.random() * 5);
        if (randomNumber !== 0) {
            goods++;
        }
    }
    quality = (goods / total) * 100;
} else if (total !== 0 && scraps !== 0) {
    quality = ((total - scraps) / total) * 100;
};
```

```

//OEE:
var oee = (availability * performance * quality) / 10000;

msg.payload =
{
  "availability": parseFloat(availability.toFixed(2)),
  "performance": parseFloat(performance.toFixed(2)),
  "quality": parseFloat(quality.toFixed(2)),
  "oee": parseFloat(oee.toFixed(2)),
  "total": total,
  "scraps": scraps,
  "plannedproduction": planned_production,
  "plannedworktime": planned_hours
}
return msg;

```

Code 9: Full script of OEE

### 6.3.3. Result

Figure 25 illustrate the results from the edge device that demonstrate its capability to process real-time data to calculate OEE. The variables resettetime, starttime, and currenttime indicate that the system is capable of processing and delivering data with precision down to the second. The edge device promptly calculates time-sensitive metrics. The OEE can be computed either when the "Refresh" button is clicked or when the input from Phase 1 is received.

18.1.2024, 10:28:44 node: debug 10 msg.payload : Object `object resettetime: 18.1.2024, 00:00:00 [UTC+1] currenttime: 18.1.2024, 10:28:44 [UTC+1] starttime: 18.1.2024, 06:00:00 [UTC+1] currentworktime: 4.4789930555555556 total: 0 scraps: 0 product: 7200 hours: 8	18.1.2024, 10:28:45 node: debug 9 msg.payload : Object `object availability: 82.23 performance: 44.01 quality: 100 oee: 36.19 total: 0 scraps: 0 plannedproduction: 7200 plannedworktime: 8	
18.1.2024, 10:58:09 node: debug 11 total : msg.payload : Object `object scraps: 100 product: 5000 hours: 8 total: 1000	18.1.2024, 10:58:10 node: debug 10 total : msg.payload : Object `object resettetime: 18.1.2024, 00:00:00 [UTC+1] currenttime: 18.1.2024, 10:58:09 [UTC+1] starttime: 18.1.2024, 06:00:00 [UTC+1] currentworktime: 4.969414722222222 total: 1000 scraps: 100 product: 5000 hours: 8	18.1.2024, 10:58:11 node: debug 9 total : msg.payload : Object `object availability: 82.23 performance: 32.2 quality: 90 oee: 23.83 total: 1000 scraps: 100 plannedproduction: 5000 plannedworktime: 8

Figure 25: Result when pressing "Refresh" button (above) and result when receiving input from Phase 1 (below)

With the Node-RED backend, the edge device also provides the ability to real-time monitor the OEE by immediately visualizing the performance, quality, and availability as any modifications occur. Figure 26 shows the gauge nodes of three components of OEE. The OEE gauge nodes are connected to the OEE function node, in order to receive data and display it on the dashboard.

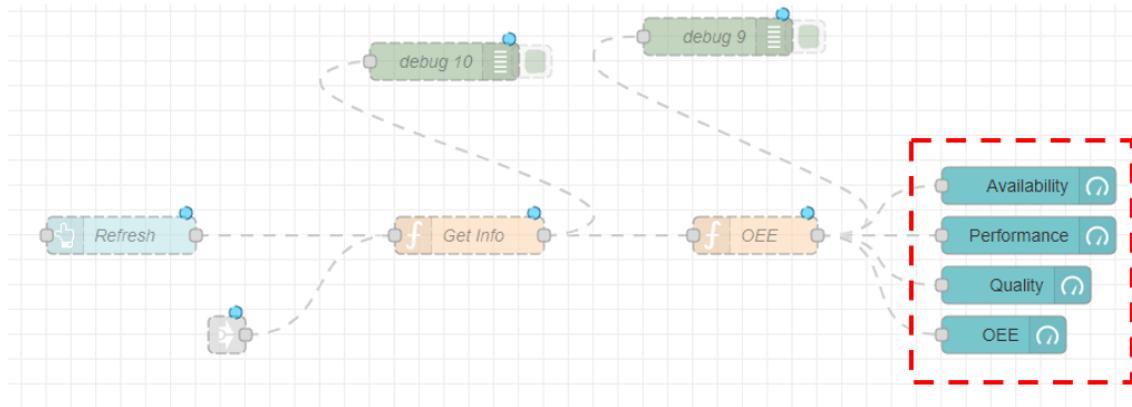


Figure 26: Config flow for the gauge nodes

Then four gauges nodes are configured the output corresponding to the output of “OEE” node as shown in Figure 27.

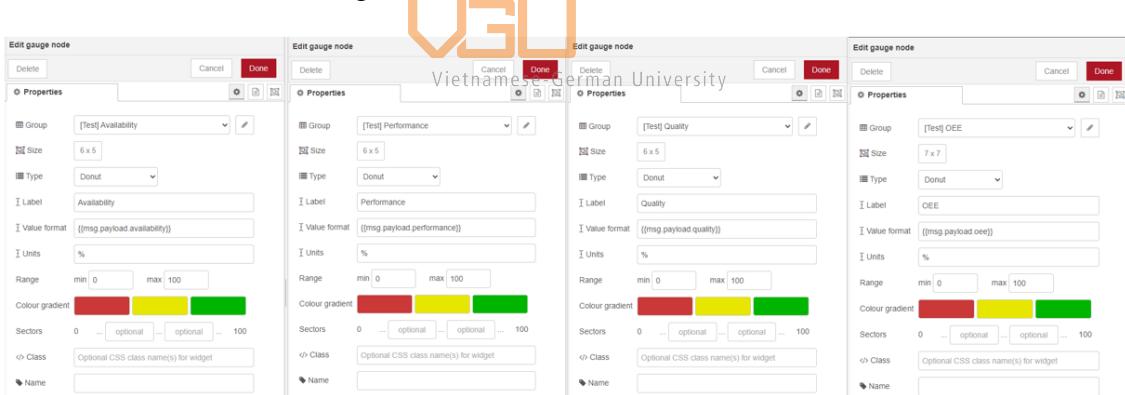


Figure 27: Config data for the gauge nodes

After configuring every element for the gauge nodes, the interface of the OEE dashboard is illustrated as shown in Figure 28. When the 'Refresh' button is pressed or input is received from Phase 1, the dashboard data will update automatically based on results of the calculations explained in Sections 6.3.1 and 6.3.2.

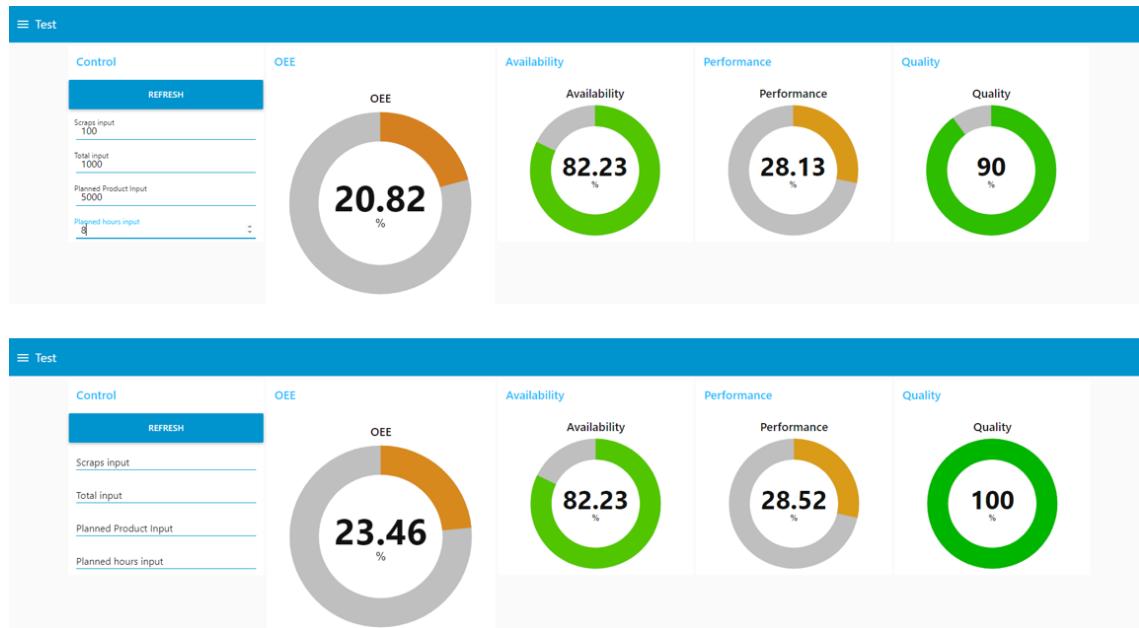


Figure 28: OEE dashboard when receiving data from Phase 1 (above) and when pressing "Refresh" button (below)

## 7. Evaluation

In the work with RevPi Core, its strong computational capabilities allowed for accurate and efficient processing of real-time data. It resulted in processing and visualizing data directly from the bandsaw with Node-RED backend are expected to lead to more accurate and timely decision-making. The implementation of storing binary values of the bandsaw to local storage using RevPi Core aids in long-term data analysis and trend identification, which can be crucial for predictive maintenance and optimizing the bandsaw's performance (RQ 1).

Moreover, local data storage provides a significant security, as it reduces the risks associated with remote data breaches and unauthorized external access. Both edge device ensure security by using login and password authentication. Similarly, the Influxdb backend also employs this security measure to safeguard the data which are stored locally on the edge device (RQ 2). It is not necessary to upload data to the cloud to avoid cyber-attack during the uploading.

A limitation was noticed during the implementation of the edge device. It is not possible to measure the delay when transferring data from the bandsaw to the Revpi Core due to the lack of appropriate facilities.



The implementation of netFIELD Compact offers an efficient method for processing and analyzing real-time data of OEE metrics, leading to improved planning, and decision-making processes (RQ 3). The successful use of these technologies highlights the capacity of edge computing to improve manufacturing productivity and efficiency.

Edge computing plays a significant role in modern manufacturing by potentially revolutionizing data processing and decision-making processes in the era of Industry 4.0. While the introduction of edge computing devices such as the RevPi Core and netFIELD Compact X8M has shown notable reductions in data processing delay, enhanced decision-making abilities, and seamless integration with existing production systems, there is still potential of incorporating AI and machine learning in such as detecting scraps for calculating quality to make OEE more accurate.

## 8. Conclusion and Outlook

This thesis has examined the transformative role of edge computing in modern manufacturing, demonstrating its potential to revolutionize data processing and decision-making processes in the era of Industry 4.0. Through the implementation of edge computing devices such as RevPi Core and netFIELD Compact X8M, it has illustrated substantial improvements in data processing latency, enhanced decision-making capabilities, and seamless integration with existing manufacturing systems. Despite the limitation in the input data that has been mentioned in the Sections 3.3.2 and 7, the results emphasize both the improvements in productivity as well as the obstacles and remedies involved in using edge computing in manufacturing settings. Furthermore, the research underscores the importance of edge computing in enabling more sustainable, flexible, and secure manufacturing operations.

This work shows the transformative impact of edge computing and its utilization in manufacturing. Despite the limitations in implementing, the use of edge devices can demonstrate substantial improvements in data processing latency and decision-making. Further implementation is possible that could illustrate a significant reduction in time of data processing compared to cloud computing or fog computing. By implementing machine learning, the binary signal data is analyzed identify subtle changes in machine operation that precede failures. This allows for timely maintenance actions to be taken before breakdowns occur, improving overall efficiency and reducing downtime. Therefore, edge device could predict machinery failure or required maintenance more accurately. Moreover, edge device could greatly enhance the precision of OEE calculations by enabling predictive analytics for equipment failure, quality defect forecasting, and optimization of performance parameters in real-time.

## 9. Reference

### Alabadi et al. 2022

ALABADI, Montdher; HABBAL, Adib; WEI, Xian, 2022. *Industrial Internet of Things: Requirements, Architecture, Challenges, and Future Research Directions*. In: IEEE Access, 10, S. 66374-66400. DOI 10.1109/ACCESS.2022.3185049

### Ali und Johl 2023

ALI, Kashif; JOHL, Satirenjit K., 2023. *Driving forces for industry 4.0 readiness, sustainable manufacturing practices and circular economy capabilities: does firm size matter?* In: Journal of Manufacturing Technology Management, 34(5), S. 838-871. DOI 10.1108/jmtm-07-2022-0254

### Avasalcai und Dustdar 2023

AVASALCAI, Cosmin; DUSTDAR, Schahram. *Edge Computing: Use Cases and Research Challenges*. In: VOGEL-HEUSER, Birgit und WIMMER, Manuel (Hrsg.) 2023. *Digital Transformation*. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 125-142. ISBN 978-3-662-65003-5.



### Baldoni et al. 2023

BALDONI, Gabriele; COMINARDI, Luca; GROSHEV, Milan; LA OLIVA, Antonio de; CORSARO, Angelo, 2023. *Managing the Far-Edge: Are Today's Centralized Solutions a Good Fit?* In: IEEE Consumer Electronics Magazine, 12(3), S. 51-61. DOI 10.1109/mce.2021.3082503

### Baque-Cantos et al. 2023

BAQUE-CANTOS, Miguel A.; MOREIRA-CAÑARTE, Cristhian Y.; ULTRERAS-RODRÍGUEZ, Andrés; NIEVES-LIZÁRRAGA, Daniel O.; GONZÁLEZ-RODRÍGUEZ, Felipe D.J.; MOREIRA-CHOEZ, Jenniffer S.; CAMPOS-SÁNCHEZ, Shirley T.; CANTOS-FIGUEROA, Mariana D.L.;

RINCÓN-GUIO, Cristian, 2023. *Technological Enablers and Prospects of Project Management in Industry 4.0: A Literature Review*. In: Academic Journal of Interdisciplinary Studies, 12(4), S. 53. DOI 10.36941/ajis-2023-0094

### **Broadcom Corporation 2012**

BROADCOM CORPORATION, 2012. *BCM2835 ARM Peripherals*. In: United Kingdom [online], 2012. Verfügbar unter:

<https://www.raspberrypi.org/app/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>

### **Cao et al. 2020**

CAO, Keyan; LIU, Yefan; MENG, Gongjie; SUN, Qimeng, 2020. *An Overview on Edge Computing Research*. In: IEEE Access, 8, S. 85714-85728. DOI 10.1109/ACCESS.2020.2991734

### **Carvalho et al. 2021**

  
CARVALHO, Gonçalo; CABRAL, Bruno; PEREIRA, Vasco; BERNARDINO, Jorge, 2021. *Edge computing: current trends, research challenges and future directions*. In: Computing, 103(5), S. 993-1023. DOI 10.1007/s00607-020-00896-5

### **Chai und Zeng 2023**

CHAI, Yuan; ZENG, Xiao-Jun, 2023. *Shapley Value-Based Computation Offloading for Edge Computing*. In: IEEE Transactions on Vehicular Technology, 72(7), S. 9448-9458. DOI 10.1109/TVT.2023.3250486

### **Chen et al. 2018**

CHEN, Baotong; WAN, Jiafu; CELESTI, Antonio; DI LI; ABBAS, Haider; ZHANG, Qin, 2018. *Edge Computing in IoT-Based Manufacturing*. In: IEEE Communications Magazine, 56(9), S. 103-109. DOI 10.1109/MCOM.2018.1701231

## Ding 2022

DING, Shiya. *Multi-agent Reinforcement Learning for Task Allocation in Cooperative Edge Cloud Computing*. In: HACID, Hakim, ALDWAIRI, Monther, BOUADJENEK, Mohamed R., PETROCCHI, Marinella, FACI, Noura, OUTAY, Fatma, BEHESHTI, Amin, THAMSEN, Lauritz und DONG, Hai (Hrsg.) 2022. *Service-Oriented Computing – ICSOC 2021 Workshops*. Cham: Springer International Publishing, S. 283-297. ISBN 978-3-031-14134-8.

## Dong et al. 2020

DONG, Yunqi; BAI, Jiujun; CHEN, Xuebo. *A Review of Edge Computing Nodes based on the Internet of Things*. In: 2020. *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security*: SCITEPRESS - Science and Technology Publications, S. 313-320. ISBN 978-989-758-426-8.

## Elmansy et al. 2023



ELMANSY, Hossam; METWALLY, Khaled; BADRAN, Khaled, 2023. *Learning agent-based security schema mitigating man-in-the-middle attacks in fog computing*. In: International Journal of Electrical and Computer Engineering (IJECE), 13(5), S. 5908. DOI 10.11591/ijece.v13i5.pp5908-5921

## Foko Sindjoung et al. 2023

FOKO SINDJOUNG, Miguel L.; VELEMPINI, Mthulisi; TAYOU DJAMEGNI, Clémentin, 2023. *A data security and privacy scheme for user quality of experience in a Mobile Edge Computing-based network*. In: Array, 19, S. 100304. DOI 10.1016/j.array.2023.100304

## Hilscher Gesellschaft für Systemautomation mbH 2023

HILSCHER GESELLSCHAFT FÜR SYSTEMAUTOMATION MBH, 2023. *User manual*

*netFIELD Compact X8M*. In: n.p. [online], 2023. Verfügbar unter:

<https://hilscher.atlassian.net/wiki/spaces/DL/pages/176947201/netFIELD+Compact+X8M+Revision+3+en>

### **Jazaeri et al. 2021**

JAZAERI, Seyedeh S.; JABBEHDARI, Sam; ASGHARI, Parvaneh; HAJ SEYYED JAVADI, Hamid, 2021. *Edge computing in SDN-IoT networks: a systematic review of issues, challenges and solutions*. In: Cluster Computing, 24(4), S. 3187-3228. DOI 10.1007/s10586-021-03311-6

### **Kolekar und Sakhare 2023**

KOLEKAR, Vikas K.; SAKHARE, Sachin R., 2023. *A Research Perspective on Data Management Techniques for Federated Cloud Environment*. In: International Journal on Recent and Innovation Trends in Computing and Communication, 11(5), S. 338-346. DOI 10.17762/ijritcc.v11i5.6622



### **Kong et al. 2022**

KONG, Xiangjie; WU, Yuhan; WANG, Hui; XIA, Feng, 2022. *Edge Computing for Internet of Everything: A Survey*. In: IEEE Internet of Things Journal, 9(23), S. 23472-23485. DOI 10.1109/JIOT.2022.3200431

### **Li et al. 2022**

LI, Yuan H.; INOUE, Luiz C.G.V.; SINHA, Roopak. *Real-time OEE visualisation for downtime detection*. In: 2022. 2022 IEEE 20th International Conference on Industrial Informatics (INDIN): IEEE, S. 729-734. ISBN 978-1-7281-7568-3.

### **Lin et al. 2023**

LIN, Rongping; GUO, Xuhui; LUO, Shan; XIAO, Yong; MORAN, Bill; ZUKERMAN, Moshe, 2023. *Application-aware computation offloading in edge computing networks*.

In: Future Generation Computer Systems, 146, S. 86-97. DOI  
10.1016/j.future.2023.04.009

### **Lo und Niang 2023**

LO, Nogaye; NIANG, Ibrahima. *SDN-based QoS architectures in Edge-IoT Systems: A Comprehensive Analysis*. In: 2023. 2023 IEEE World AI IoT Congress (AIoT): IEEE, S. 605-611. ISBN 979-8-3503-3761-7.

### **Ma et al. 2023**

MA, Yehan; CHEN, Cailian; ZENG, Shen; GUAN, Xinping; LU, Chenyang, 2023. *Data-Driven Edge Offloading for Wireless Control Systems*. In: IEEE Internet of Things Journal, 10(12), S. 10802-10816. DOI 10.1109/JIOT.2023.3242770

### **Mourtzis und Balkamos 2023**

MOURTZIS, Dimitris; BALKAMOS, Nikos, 2023. *Design of Manufacturing Systems Based on Digital Shadow and Robust Engineering*. In: Applied Sciences, 13(8), S. 5184. DOI 10.3390/app13085184

### **Nguyen et al. 2020**

NGUYEN, Tuan A.; MIN, Dugki; CHOI, Eunmi, 2020. *A Hierarchical Modeling and Analysis Framework for Availability and Security Quantification of IoT Infrastructures*. In: Electronics, 9(1), S. 155. DOI 10.3390/electronics9010155

### **Nimawat und Das Gidwani 2022**

NIMAWAT, Dheeraj; DAS GIDWANI, Bhagwan, 2022. *Challenges facing by manufacturing industries towards implementation of industry 4.0: an empirical research*. In: International Journal on Interactive Design and Manufacturing (IJIDeM), 16(4), S. 1371-1383. DOI 10.1007/s12008-022-00961-7

### **Patel et al. 2017**

PATEL, Pankesh; INTIZAR ALI, Muhammad; SHETH, Amit, 2017. *On Using the Intelligent Edge for IoT Analytics*. In: IEEE Intelligent Systems, 32(5), S. 64-69. DOI 10.1109/MIS.2017.3711653

### **Patrikar und Parate 2022**

PATRIKAR, Devashree R.; PARATE, Mayur R., 2022. *Anomaly detection using edge computing in video surveillance system: review*. In: International journal of multimedia information retrieval, 11(2), S. 85-110. DOI 10.1007/s13735-022-00227-8

### **Peniak et al. 2023**

PENIAK, Peter; BUBENÍKOVÁ, Emília; KANÁLIKOVÁ, Alžbeta, 2023. *Validation of High-Availability Model for Edge Devices and IIoT*. In: Sensors (Basel, Switzerland), 23(10). DOI 10.3390/s23104871



### **Ray Fernandez 2022**

RAY FERNANDEZ, 2022. *A brief history of edge computing*. In: n.p. [online], 2022. Verfügbar unter: <https://www.techrepublic.com/article/edge-computing-history/>

### **Revolution Pi n.d.**

REVOLUTION PI, n.d. *RevPi Core*. In: Germany [online], n.d. Verfügbar unter: <https://revolutionpi.com/revpi-core>

### **Ringler et al. 2023**

RINGLER, Nicolas; KNITTEL, Dominique; PONSART, Jean-Christophe; NOUARI, Mohammed; YAKOB, Abderrahmane; ROMANI, Daniel. *Machine Learning based Real Time Predictive Maintenance at the Edge for Manufacturing Systems: A Practical Example*. In: 2023. 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET): IEEE, S. 1-6. ISBN 979-8-3503-3179-0.

### **S. Kalpande 2014**

S. KALPANDE, 2014. *Oee an effective tool for tpm implementation-A case study*. In:

Serbia [online], 2014. Verfügbar unter:

[https://www.cqm.rs/2014/cd1/pdf/papers/focus\\_3/058.pdf](https://www.cqm.rs/2014/cd1/pdf/papers/focus_3/058.pdf)

### **Sakshi Gupta 2023**

SAKSHI GUPTA, 2023. *CODING Is Programming Hard? A Guide To Getting Started in*

2024. In: n.p. [online], 2023. Verfügbar unter:

<https://www.springboard.com/blog/software-engineering/is-programming-hard/#:~:text=The%20major%20reason%20why%20programming,or%20any%20other%20human%20language.>

### **Sohail et al. 2023**

SOHAIL, Fizzah; HAIDER, Hira; ISMAT, Najma. *The Impact of Fog computing in the IoT*

  
 Vietnamese-German University  
 World. In: 2023. *2023 Global Conference on Wireless and Optical Technologies (GCWOT)*: IEEE, S. 1-7. ISBN 979-8-3503-3371-8.

### **Sugumaran et al. 2023**

SUGUMARAN, Vijayan; GOEL, Neha; YADAV, Ravindra K. (Hrsg.), 2023. *Handbook of*

*Research on Machine Learning-Enabled IoT for Smart Applications Across Industries*:

IGI Global. ISBN 9781668487853.

### **Sun et al. 2022**

SUN, Xu; YU, Hao; SOLVANG, Wei D.; WANG, Yi; WANG, Kesheng, 2022. *The*

*application of Industry 4.0 technologies in sustainable logistics: a systematic literature*

*review (2012-2020) to explore future research opportunities*. In: Environmental science

and pollution research international, 29(7), S. 9560-9591. DOI 10.1007/s11356-021-

17693-y

### **Surjya Kanta Pal 2022**

SURJYA KANTA PAL, 2022. *Digital Twin – Fundamental Concepts to Applications in Advanced Manufacturing*. In: Springer Series in Advanced Manufacturing, S. 293-335.

### **Thames und Schaefer 2017**

THAMES, Lane; SCHAEFER, Dirk (Hrsg.), 2017. *Cybersecurity for Industry 4.0*. Cham: Springer International Publishing. ISBN 978-3-319-50659-3.

### **Vaibhav Sharma 2023**

VAIBHAV SHARMA, 2023. *THE IMPLEMENTATION OF CLOUD COMPUTING AS STRATEGIC TECHNOLOGY FOR SUSTAINABLE DEVELOPMENT USING REGRESSION ANALYSIS*. In: Journal of Contemporary Issues in Business and Government, 29(2), S. 162-169. DOI 10.48047/cibg.2023.29.02.016

### **Verma und Kumar 2023**

VERMA, Parul; KUMAR, Umesh.  *Analyzing the Application of Edge Computing in Smart Healthcare*. In: RAWAT, Danda B., AWASTHI, Lalit K., BALAS, Valentina E., KUMAR, Mohit und SAMRIYA, Jitendra K. (Hrsg.) 2023. *Convergence of Cloud with AI for Big Data Analytics*: Wiley, S. 121-155. ISBN 9781119904885.

### **Zhang et al. 2021**

ZHANG, Caiming; CHEN, Yong; CHEN, Hong; CHONG, Dazhi, 2021. *Industry 4.0 and its Implementation: a Review*. In: Information Systems Frontiers. DOI 10.1007/s10796-021-10153-5

### **Zhang et al. 2023**

ZHANG, Junna; CHEN, Jiawei; BAO, Xiang; LIU, Chunhong; YUAN, Peiyan; ZHANG, Xinglin; WANG, Shangguang, 2023. *Dependent task offloading mechanism for cloud–*

*edge-device collaboration.* In: Journal of Network and Computer Applications, 216, S. 103656. DOI 10.1016/j.jnca.2023.103656

### **Zhang und Ji 2020**

ZHANG, Chaoyang; JI, Weixi, 2020. *Edge Computing Enabled Production Anomalies Detection and Energy-Efficient Production Decision Approach for Discrete Manufacturing Workshops.* In: IEEE Access, 8, S. 158197-158207. DOI 10.1109/ACCESS.2020.3020136

### **Zhao et al. 2023**

ZHAO, Xiaoyan; CHEN, Ruiguang; LI, Jianwei; LI, Chunlei; CHEN, Yan; ZHANG, Tianyao; ZHANG, Zhaoxi, 2023. *Framework Design of an Edge Gateway System Supporting Multi-Protocol Standardized Access Detection.* In: Journal of Advanced Computational Intelligence and Intelligent Informatics, 27(3), S. 431-437. DOI 10.20965/jaciii.2023.p0431



## 10. Affidavit

I hereby affirm in lieu of oath

- that I have written this academic thesis independently and have used only the resources listed in the attached list.
- that this academic paper has not been submitted to any other examination authority in the same or a similar form.
- that I have clearly identified any passages taken verbatim or in spirit from published or unpublished sources.

Bochum, 12.02.2024

[Place, date]



[Signature]

Mechanical Engineering Examination Regulations 2013 (Bachelor & Master)

### § Section 14 Failure, cheating and breach of regulations

- (1) If the candidate attempts to influence the result of an examination by cheating, the examination in question is to be assessed as 5.0 ("fail") or "fail". The respective examiner or invigilator will record the assessment. The assessment is carried out by the examination board. In the event of multiple or other serious attempts at cheating, the candidate may be excluded from taking further examinations or exmatriculated.
- (4) The submission of plagiarized material for project work, course-related assignments or the Master's thesis is considered cheating in accordance with para. 1.

### § Section 63 Higher Education Act - HG

- (5) The universities and the state examination offices can demand and accept an affirmation in lieu of an oath from the examination candidates that the examination work has been carried out by them independently and without unauthorized external assistance. Anyone who wilfully
  - (2) 1. violates a regulation of a university examination regulation concerning cheating in examinations or
  - (3) 2. against a corresponding regulation of a state or ecclesiastical examination regulation
  - (4) is in breach of an administrative offence. The administrative offence may be punished with a fine of up to 50,000 euros. The universities may regulate the details in regulations.

The competent administrative authority for the prosecution and punishment of administrative offences according to sentence 2 number 1 is the Chancellor and according to sentence 2 number 2 the State Examination Office. In the event of multiple or other serious attempts at cheating, the candidate may also be exmatriculated.

§ Section 156 StGB False affirmation in lieu of an oath

Anyone who makes a false affidavit before an authority responsible for taking an affirmation in lieu of an oath or makes a false statement in reliance on such an affidavit is liable to a custodial sentence not exceeding three years or a monetary penalty.

§ Section 161 StGB Negligent false statement;

Negligent false affirmation in lieu of an oath

(1) If one of the offences specified in sections 154 to 156 has been committed through negligence, a custodial sentence not exceeding one year or a monetary penalty shall be imposed.

(2) The offence shall not be punishable if the offender corrects the false statement in good time. The provisions of section 158 (2) and (3) shall apply mutatis mutandis.



Ruhr-Universität Bochum  
Fakultät Maschinenbau  
Lehrstuhl für Verfahrenstechnische  
Transportprozesse Universitätsstraße 150  
44780 Bochum