FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Tracking products in production lines

Daniel Filipe da Silva Gonçalves

Master in Electrical and Computers Engineering

Internal Supervisor: Prof. Luis Miguel Pinho de Almeida

External Supervisor: Eng. Elder António de Oliveira Matos

July 11th, 2019

© Daniel Gonçalves, 2019

Abstract

Automatic identification techniques have been one of the many topics that Industry 4.0 has addressed. From common barcodes to the latest and improved radio frequency identification systems, all of them can be found in today's factories. In addition, automatic product localization on production lines for traceability and improvement purposes is a constant concern for companies, which are constantly looking for new ways to lower production costs while maintaining product quality.

Bosch Termotechnology S.A. de Aveiro is one of the companies that uses barcode identification in many of its current production lines, but in preliminary studies, significant improvements in line efficiency and productivity were discovered, which make the upgrade to a type of automatic product identification advantageous, depending on the cost of the technology. In addition, a real-time product tracking platform is seen as an advantage by having the ability of identifying other problems, such as detecting bottlenecks at different workstations.

In this dissertation, we create a system for automatic product identification on one of the various production lines of Bosch Termotechnology S.A. and also developed an application for real-time product tracking.

First, we begin by studying several different product identification technologies, from the most common techniques (e.g. barcodes), to identification through indoor tracking. Next, a concept adopted by the current processes of Industry 4.0, Digital Twin, is analysed in order to better understand the concept and thus be able to plan a better platform for product tracking in real time.

Secondly, we present the problem characterization, where it is defined in more detail and analysed the current state of the production line, its architecture and order of operations in the stations. An economic feasibility analysis of the project is also carried out and finally the proposed solution is presented, both for the identification solution and for the tracking platform.

Next, we address the whole process of implementing the proposed solution in the production line and the graphical interface. At the end we will analysed the results of the project.

Finally, we present a risk and project satisfaction analysis, as well as a look back to the state-of-the-art review, to find out if the proposed and implemented solution was, in fact, the most appropriate. We also briefly mentioned the contributions of the project and future work.

Resumo

Técnicas de identificação automáticas tem sido um dos muitos tópicos que o tema da Indústria 4.0 tem abordado. Desde os comuns códigos de barras aos mais recentes e melhorados sistemas de identificação por rádio frequência, todos eles podem ser encontrados nas fábricas de hoje em dia. Em adição, localização automática de produto nas linhas para efeitos de rastreabilidade e estudos de melhoramento das mesmas é uma preocupação constante das empresas, que estão constantemente em busca de novas formas de baixar os custos de produção, mantendo a qualidade do produto.

A Bosch Termotecnologia SA de Aveiro é uma destas empresas que usa identificação por códigos de barras em muitas das suas linhas de produção atuais, porém, em estudos preliminares, foram descobertas possíveis melhorias significativas em termos de eficiência e produtividade, que fazem com que a mudança para um tipo de identificação de produto automática seja vantajosa, dependendo do custo da tecnologia. Em adição, uma plataforma de seguimento de produto em tempo real é vista como uma vantagem em termos de identificação de outros problemas, tal como a deteção de engarrafamentos em postos de trabalho.

Nesta dissertação, criamos um sistema para identificação automática de produto numa das várias linhas de produção da Bosch Termotecnologia SA e também desenvolver uma aplicação para rastreamento de produto em tempo real.

Primeiro, começamos por estudar várias tecnologias diferentes de identificação de produto, desde técnicas mais comuns como códigos de barras, a identificação através de localização *indoors*. De seguida é analisado um conceito adotado pelos processos correntes de Indústria 4.0, Digital Twin, para melhor entender o conceito e assim poder planear uma melhor plataforma de seguimento de produto em tempo real.

Em segundo lugar, será apresentada a caracterização do problema, onde é definido o mesmo em mais detalhe e analisado o estado atual da linha de produção, a sua arquitetura e ordem de operações nas estações. Também é feita uma análise de viabilidade económica do projeto e por último é apresentada a solução proposta, tanto para a solução de identificação como para a plataforma de rastreamento.

De seguida, abordamos todo o processo de implementação da solução proposta na linha de produção e da *interface* gráfica e no final serão analisados os resultados do projeto.

Finalmente, apresentamos uma análise de risco e de satisfação do projeto, assim como um *look back* no estado da arte para averiguar se a solução proposta e implementada foi, de facto, a mais adequada. Também mencionamos brevemente as contribuições do projeto e trabalho futuro.

Agradecimentos

Esta dissertação não teria sido possível sem a ajuda e suporte de algumas pessoas a quem gostaria de agradecer.

Vou começar por agradecer ao meu supervisor na equipa da Bosch Aveiro, Duarte Almeida, por toda a sua disponibilidade, atenção, esforço e paciência que dedicou comigo nestes meses, assim como toda a sua orientação. Sem ele não teria chegado onde cheguei.

A toda a equipa onde estive integrado, por serem pessoas fenomenais e bons amigos que me suportaram nesta etapa. Um agradecimento especial ao Vitor Silva e ao Filipe Costa pela ajuda prestada com os modelos 3D.

Ao Professor Luís Almeida, por dedicar muito do seu tempo a orientar-me, por todos os comentários e críticas tecidas que ajudaram a criar esta dissertação e também pelo incentivo que esteve sempre presente.

Ao Paulo Correia, por ser o meu companheiro de mesa nos últimos 5 meses, e com quem partilho uma amizade de 5 anos. Sem o seu suporte desde o início, que não era a pessoa que sou hoje.

A todos os meus amigos, de modo especial ao João Afonso e Júnio Parente, por todos os momentos positivos e negativos

À minha namorada ,Inês Caiano, pelo apoio incondicional em todos os momentos e por ser um pilar na minha vida, esta que sempre me apoia em todas as decisões tomadas.

Ao meu irmão, Diogo Gonçalves, por me ter ajudado na escolha de curso que estou a terminar.

E finalmente, aos meus pais e à minha família pelos valores e ensinamentos que me transmitiram e que fizeram de mim a pessoa que sou hoje. A eles que direta e indiretamente sempre me apoiaram em todas as decisões tomadas ao longo do meu percurso académico e pessoal.

Daniel Gonçalves

"Success is getting what you want, happiness is wanting what you get."

W. P. Kinsella

Contents

1 Introduction **1**
- 1.1 Context and motivation . . . 1
- 1.2 Objectives . . . 2
  - 1.2.1 Business objectives . . . 2
  - 1.2.2 Primary objectives . . . 3
  - 1.2.3 Secondary objectives . . . 3
  - 1.2.4 Additional requirements . . . 3
- 1.3 Document organization . . . 3

2 Literature Review **5**
- 2.1 Identification technologies and techniques . . . 5
  - 2.1.1 Barcode . . . 5
  - 2.1.2 Radio frequency identification (RFID) . . . 6
  - 2.1.3 Optical character recognition (OCR) . . . 7
  - 2.1.4 Smart cards . . . 7
  - 2.1.5 Speech recognition . . . 8
  - 2.1.6 Comparison . . . 8
- 2.2 Localization based on Radio Frequency (RF) transmission . . . 8
  - 2.2.1 Received signal strength indication . . . 8
  - 2.2.2 Time of flight (ToF) . . . 11
  - 2.2.3 Time of arrival . . . 11
  - 2.2.4 Time difference of arrival . . . 12
  - 2.2.5 Angle of arrival . . . 13
  - 2.2.6 Phase difference of arrival (PDoA) . . . 13
  - 2.2.7 Radio frequency identification . . . 13
- 2.3 Non-RF localization . . . 13
  - 2.3.1 Magnetic positioning . . . 13
  - 2.3.2 Ultrasound positioning . . . 14
  - 2.3.3 Inertial measurement . . . 14
  - 2.3.4 Camera based positioning . . . 14
- 2.4 Comparison between localization solutions . . . 14
- 2.5 Digital Twin . . . 15
- 2.6 Summary . . . 16

3 Problem characterization **19**
- 3.1 Problem definition . . . 20
- 3.2 Proposed solution . . . 23
  - 3.2.1 UHF Solution - Multiple antennas . . . 25

3.2.2 HF Solution . . . 26
3.2.3 Nexeed Control and Nexeed MES . . . 28
3.2.4 Data Analysis . . . 30
3.3 Summary . . . 30

4 Automatic Identification Implementation **33**
4.1 RFID Programming . . . 33
4.1.1 Serial Communication Handler . . . 34
4.1.2 RFID Serial Handler . . . 38
4.1.3 Library validation and verification . . . 43
4.2 Physical Alterations . . . 45
4.2.1 Trolley Tag Support . . . 48
4.2.2 RFID Reader Support . . . 50
4.2.3 RFID Support on stations without stoppers . . . 51
4.3 Station code alterations . . . 52
4.3.1 Write stations . . . 53
4.3.2 Read stations . . . 55
4.3.3 Clear stations . . . 58
4.3.4 Time analysis . . . 58
4.3.5 MES parameters insertion . . . 58
4.4 Deployed station code . . . 60
4.5 Summary . . . 61

5 Product Tracking Web Based Platform **63**
5.1 ASP.NET MVC Framework . . . 63
5.1.1 Architecture . . . 64
5.2 Web platform . . . 65
5.2.1 Station map . . . 65
5.2.2 Statistics . . . 70
5.3 Summary . . . 70

6 Results and validation **73**
6.1 RFID status results . . . 73
6.2 Time results analysis for stations 10 and 12 . . . 75
6.2.1 Data discussion . . . 80
6.3 Time results average for the line . . . 82
6.4 Risk Assessment . . . 83
6.4.1 Risks incurred . . . 85
6.5 Satisfaction analysis . . . 86
6.6 Summary . . . 87

7 Conclusion and next steps **89**
7.1 Proposed solution look back . . . 89
7.2 Lessons learned . . . 90
7.3 Contributions . . . 91
7.4 Next work . . . 91

A Technical drawings **93**
A.1 Steel frame technical drawing 94
A.2 Aluminium cylinder technical drawing 95
A.3 Plastic spacer technical drawing 96

References **97**

List of Figures

1.1 Simplified layout of the production line. . . . 2

2.1 1D and 2D barcode examples. . . . 6
2.2 LoRa and LoRaWAN layers. . . . 10
2.3 Ultra wideband diagram. . . . 12
2.4 Digital twin concept. . . . 16

3.1 Layout of the testing line under analysis. . . . 19
3.2 Generic order of operations in a station. . . . 20
3.3 PLC report generated at station number 33. . . . 20
3.4 Proposed order of operations. . . . 21
3.5 Station stopper layout. . . . 24
3.6 Hardware architecture used with Nexeed Control. . . . 29
3.7 Nexeed MES architecture 29
3.8 Early web application mock-up 30

4.1 Low level RS232 handler. . . . 34
4.2 Higher level ASCII protocol and SerialComm handler. . . . 34
4.3 Main serial communication state machine 35
4.4 Error and reset serial communication state machines 37
4.5 RFID serial handling state machine 39
4.6 Writing a string chart. . . . 41
4.7 Reading and parsing characters from the tag memory. . . . 43
4.8 PC to RF240R RS232 communication message. . . . 44
4.9 PLC to RF240R RS232 communication message with the KL6001. . . . 44
4.10 PLC to RF240R RS232 communication message with the KL6031. . . . 45
4.11 Generic RFID radiation pattern [10]. . . . 46
4.12 Trolley base. . . . 46
4.13 Station stopper. . . . 46
4.14 Trolley base that rests against the stopper. . . . 47
4.15 Trolley tag positioning. . . . 47
4.16 RFID reader positioning. . . . 47
4.17 Ideal tag and reader positioning. . . . 48
4.18 First iteration tag support. . . . 48
4.19 First iteration tag support. . . . 48
4.20 Steel frame. . . . 49
4.21 Steel frame technical drawing. . . . 49
4.22 Aluminium cylinder. . . . 49
4.23 Aluminium cylinder technical drawing. . . . 49

4.24 Plastic spacer. . . . 49
4.25 Plastic spacer technical drawing. . . . 49
4.26 Mounting plate. . . . 50
4.27 Wafer to space the tag. . . . 50
4.28 Final tag support. . . . 50
4.29 Final trolley implementation. . . . 50
4.30 Final stopper implementation. . . . 51
4.31 Station 12 RFID reader and optical sensor on their fly-by read positions. . . . 52
4.32 Implemented SFC write code. . . . 54
4.33 Flow chart of step N064 code for station 10. . . . 55
4.34 Implemented SFC read code. . . . 56
4.35 Station 41 SFC to check if the tags are matched. . . . 57
4.36 Time analysis for the different steps. . . . 58
4.37 Highlighted stations which will be altered. . . . 60

5.1 Model-View-Controller diagram of interactions. . . . 63
5.2 Web platform architecture diagram. . . . 64
5.3 Current state of the web platform. . . . 65
5.4 Web platform database connection error. . . . 65
5.5 Base line layout for map integration. . . . 66
5.6 Every marker at a station. . . . 66
5.7 Marker text pop-up. . . . 67
5.8 Object created to map each unique product. . . . 68
5.9 Following algorithm activity diagram. . . . 69
5.10 Web platform list statistics. . . . 70

6.1 Station 10 write status results from week 23 of 2019. . . . 73
6.2 Station 12 write status results from week 23 of 2019. . . . 74
6.3 Station 10 change times for June 4th. . . . 76
6.4 Station 10 change times for June 5th. . . . 76
6.5 Station 12 change times for June 4th. . . . 77
6.6 Station 12 change times for June 5th. . . . 77
6.7 Station 10 exit times for June 5th. . . . 78
6.8 Station 10 process times for June 4th. . . . 78
6.9 Station 10 process times for June 5th. . . . 79
6.10 Station 12 process times for June 4th. . . . 79
6.11 Station 12 process times for June 5th. . . . 80
6.12 Process, exit and change times for the stations. . . . 80
6.13 Station 10 cycle time for week 23 without outliers. . . . 81
6.14 Station 12 cycle time for week 23 without outliers. . . . 82

A.1 Technical drawing for the steel frame. . . . 94
A.2 Technical drawing for the aluminium cylinder. . . . 95
A.3 Technical drawing for the plastic spacer. . . . 96

List of Tables

2.1 Example structure of a simple 1D barcode. 6
2.2 RFID frequency bands. 7
2.3 Comparison between different identification solutions. 8
2.4 Comparison between different localization solutions. 15

3.1 Time spent reading the barcode in each station by the operator. 22
3.2 Forecast production figures for 2019 on line 7. 22
3.3 Project savings. 23
3.4 UHF RFID devices. 25
3.5 HF RFID devices. 25
3.6 UHF solutions prices. 26
3.7 Siemens IO-link solution. 27
3.8 Siemens RS232 solution. 27
3.9 Turck Profibus-DP solution. 27

4.1 Serial comm fault error codes. 38
4.2 ASCII table with hexadecimal values 40
4.3 Write command frame designation. 42
4.4 Stations implementation. 60

5.1 *LOCATION_RESULTS* table. 68

6.1 RFID read status results from June 19th. 75
6.2 Week 23 of 2019, station 10 times. 80
6.3 Week 23 of 2019, station 12 times. 80
6.4 Time analysis for line 7 in 10 different stations, on June 19th. 83
6.5 Project risk assessment. 83
6.6 Technical risks for the risk assessment. 84
6.7 Project management risks for the risk assessment. 85

Abbreviations

1D	One-dimension
2D	Two-dimension
3D	Three-Dimension
AIDC	Automatic Identification and Data Capture
AoA	Angle of Arrival
BLE	Bluetooth Low Energy
CD	Check Digit
CI	Country Identifier
DT	Digital Twin
FB	Function Block
GSM	Global System for Mobile Communications
HARQ	Hybrid Automatic Repeat Requests
HF	High Frequency
HMI	Human-Machine Interface
IoT	Internet of Things
KPI	Key Performance Indicators
LF	Low Frequency
LTE	Long-Term Evolution
MAC	Media Access Control
MES	Manufacturing Execution System
NFC	Near Field Communications
NLoS	Non-line-of-sight
OCR	Optical Character Recognition
OEE	Overall Equipment Effectiveness
PDoA	Phase Difference of Arrival
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
RF	Radio Frequency
RFID	Radio Frequency Identification
RLTS	Real Time Localization Systems
RSS	Received Signal Strength
SFC	Sequential Function Chart
SHF	Super High Frequency
ST	Structured Text
TDoA	Time difference of Arrival
ToA	Time of Arrival
ToF	Time of Flight
UHF	Ultra High Frequency
UMTS	Universal Mobile Telecommunications System
UWB	Ultra Wideband

Chapter 1

Introduction

1.1 Context and motivation

Bosch Termotecnologia SA in Aveiro (AvP) is part of the Energy and Buildings Technology division inside the Bosch Group. In this division, it is categorized in the thermotechnology business segment, meaning it offers solutions for indoor climate, hot-water supply and industrial steam generation and energy conversion. It is subdivided in three segments, residential heating, residential water and the commercial/industrial segment, the primary focus of the factory in Aveiro is residential water (e.g. gas water heaters) and a second source for residential heating (e.g. boilers). Here, a specialized team works on the development of new processes of industrialization and industry 4.0.

Naturally, reducing the production costs and optimizing the efficiency and the productivity of industrial operations is of the utmost importance. More specifically, by decreasing the time it takes some processes and procedures to execute, it is possible to save valuable time, representing a lower cost per product and an increase in production.

On the production line that this dissertation project is aimed at, trolleys are used to move the product between stations and barcode readers used to identify the product between them, which is a common practice. However, by studying the time it takes for the barcode to be read (manual operation), it is possible to conclude that around 30 seconds (specific line and product) can be saved during the entire cycle of that product. Moreover, using new industry 4.0 concepts like the digital twin, we can further study and optimize the process by analysing product and line information stored in the manufacturing execution system (MES), retrieving useful information to improve the overall production process.

In addition, knowing where a product is located at all times can bring serious advantages in optimizing that specific line, by having to ability to detect bottlenecks and other problems.

This dissertation falls in this context, and it will be developed in the workshop dedicated to the development of new processes, where it is possible to build prototypes and test setups to validate new concepts.

1.2 Objectives

The main objective of the dissertation is to evaluate and implement the use of new technologies (e.g. RFID technology and low energy microcontrollers) in the development and improvement of assembly processes based on trolleys that travel through different stations over the production lines. In these processes, it is not unusual that the operators waste time by having to identify the product and trolley, time which can be reduced by using automatic localization/identification techniques.

Methods that use these already exist in a few production lines (this project will be implemented in a line where this is already in use), where a barcode reader is manually handled by the respective operator to identify the product, and this procedure can take between 2 and 4 seconds to be completed. On the other hand, by using an automatic localization/identification technique (e.g. using RFID), we can decrease or even remove this identification time.

Figure 1.1 presents a simplified layout of the line in question.

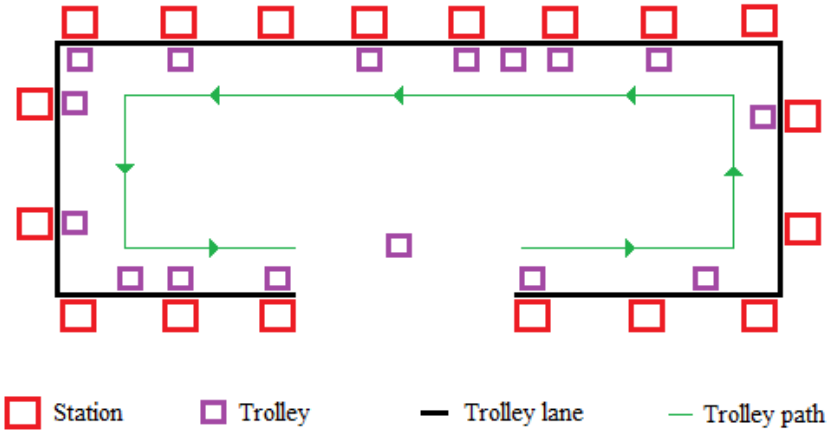


Figure 1.1: Simplified layout of the production line.

Therefore, the global objective is the identification of the product (associating the trolley with the product they carry) and automatic trolley tracking, inside the production line. However, we can divide the objectives in four distinct categories, business objectives, primary project objectives, secondary project objectives and requirements.

1.2.1 Business objectives

- Production cost reduction by removing operator time (VT), identifying the product automatically;
- Improve the efficiency and productivity of the line by reducing the cycle time.

1.2.2 Primary objectives

- Automatic and instantaneous identification system for the product;
- Association system for the trolley and product.

1.2.3 Secondary objectives

- Monitoring of different partial times inside the time cycle of each station (product waiting to be processed and waiting to be moved after the procedure);
- Front-end system to aid in the visualization of the trolleys/products.

1.2.4 Additional requirements

- Trolleys must be identifiable by the MES, allowing the monitoring of their current position and time spent in each station, as well as wait times;
- Replace (or keep as backup) the current barcode reading done by the operator at each station;
- Ensure the scalability of the proposed solution for integration in the various lines with the existing MES.

1.3 Document organization

This document has the following organization. In this section we introduced the context, motivation, objectives and presented the scope of the work and the global and specific objectives. The following section introduces various alternatives for the identification problem, with different technologies and approaches. Then, the next section defines the problem and presents a solution. Consequently, the following two chapters focus on the implementation of the identification technique and the front-end platform for real-time tracking.

Additionally, on chapter 6 we analyse the project results from the deployed system, a risk assessment is presented as well as a satisfaction analysis.

Finally, the last chapter briefly focuses on the topics of lessons learned, contributions, next work and also a look back at the chosen technology.

Chapter 2

Literature Review

This chapter describes some identification and localization techniques and technologies currently used to track indoors objects and correctly identify them in a production line, as well as the concept of a digital twin (DT) in industry 4.0. We start by addressing certain identification techniques currently used in the industry, followed by localization techniques based on wireless communication technologies that are suited for the problem at hand. Then we have an overview of the non-radio technologies and techniques that can be used as solutions to our problem and finally we approach the digital twin concept.

It is also worth to mention that the problems of identification and localization are closely related. In fact, localization without identification is futile, thus we seek methods that support both simultaneously. On the other hand, identification at specific stations in the line already implicitly localizes the trolleys.

2.1 Identification technologies and techniques

Automatic identification and data capture (AIDC) is, generally, an automatic three step process consisting of methods to identify objects, followed by a data collection system and a way to insert that data into computer systems. It has become very popular in many different industries, logistics systems, manufacturing companies and material flow systems. This is being increasingly used in manufacturing processes to track, trace and identify the various different products throughout the assembly line [27].

In 1997, an organization named AIDC 100 was formed, composed of individuals who contributed vastly to the growth of this segments, expanding the businesses knowledge and know-how. Technologies that are normally considered part of AIDC are the following [35].

2.1.1 Barcode

A barcode, in its essence, is a representation of data, and has been used in industries to identify and track objects for decades, being a technology used all around the globe. The traditional ones encode data by varying the width and the spacing of parallel lines (1D, one-dimensional), but

more recently, the use of two-dimensional variants is being increasingly adopted and developed [30]. These are not encoded in the same way, using rectangles, dots, hexagons and other patterns to store more information. Table 2.1 and figure 2.1.

Table 2.1: Example structure of a simple 1D barcode.

7	5	0	1	0	5	4	5	3	0	1	0	7
CI		Company Ident.					Product Ident.					CD



Figure 2.1: 1D and 2D barcode examples.

This technology is widely used today due to the speed and performance it offers, being an accurate, fast and inexpensive method to enter data automatically in a machine [12]. However, it still requires a direct line of sight between the reader and code, which, when the code is in a difficult place, requires a manual operator to handle the reader.

Another method commonly referenced are vision systems [14]. These are used in this context for parts identification and classification. However, they are closely tied with barcodes, with the increase in use of two dimensional barcodes, laser barcode scanners are being replaced with cameras [30], which allows for non specialized hardware to interpret the code.

One of the biggest drawbacks of using barcode identification is the inability to change the code. As such, once a barcode is printed, the encoded code cannot change.

2.1.2 Radio frequency identification (RFID)

RFID systems are quickly being adopted by industries in product identification, as they pose various advantages against many other types of identification technologies, mainly barcodes. In sum, an RFID system is made of three components, the tag, the reader and the host computer. The tag, much like the barcode, holds the product information to be read while the reader has the ability to decode it and send it to a computer to be processed [20].

Compared to barcode systems, RFID systems have the ability to read and write data to tags, have higher data rates and memory sizes, can function in non-line-of-sight (NLoS) scenarios, communications with more than one tag simultaneously, greater security and durability. With this, one of the main advantages of these systems for our purpose is tied the NLoS it offers, meaning that there is no need to manually handle the reader in order read the product data [55].

RFID tags can be defined in two distinct categories, active and passive tags. Active tags require a power supply, normally a battery, and passive ones do not, instead they are powered by the electromagnetic energy transmitted from an RFID reader. The passive tag draws energy from the RF waves through the antenna, powering the internal components and generates a signal back to the system, this is called backscatter (reflection of waves, particles or signals back to the direction from which they came [42]).

Active tags have the advantage of having an extremely long range, where as passive ones have a much bigger lifetime, are typically less expensive and are much smaller.

This technology has various different types of frequencies it can work at, one of the most common is NFC (Near field communications [4]), table 2.2 shows a list of different frequency bands.

Table 2.2: RFID frequency bands.

Band	Range	Data speed
120 - 500 KHz (LF)	Short to medium	Low
10 - 15 MHz (HF)	Short to medium	Medium
850 - 950 MHz (UHF)	Long	High
2.4 - 5.8 GHz (SHF)	Very long	High

2.1.3 Optical character recognition (OCR)

Optical character recognition is a technology used to convert images of text (typed, handwritten or printed) into a digital format [44]. It is currently widely used for scanning printed paper data records, such as passports, invoices, bank statements and receipts and in recent years it has been a good field of research into pattern recognition, artificial intelligence and computer vision. However, OCR systems have not become applicable in industrial identification systems, mainly because of their high price and complex readers in comparison with other technologies [9].

2.1.4 Smart cards

Smart cards are used by practically everyone, most of them without even realizing, in applications such as credit cards, public transport, identification, mobile phones (SIM) and many others. It is, usually, a plastic card with an embedded microprocessor chip and a data storage unit [51].

With means to store, process, manage and perform cryptography algorithms, these cards provide a significant amount of security and the volume of information that they can hold is also surprisingly high, some having up to 256 kilobytes of storage. However, it needs to be inserted into the reader for it to be read, has it needs a physical connection, which is not very useful for the problem at hand.

2.1.5 Speech recognition

Using microphones at designated locations, commands can be made by an operator to signal devices instead of using other methods (e.g. button presses). Systems like this one check speech characteristics of the person speaking and compare them against existing reference models. This method is less used in industries due to the noise interferences that are common in these environments [9].

Voice recognition systems differ from this one by only identifying the person speaking, not the speech in itself, which can simplify the task.

2.1.6 Comparison

The present section compares the identification techniques and technologies shown above [9] in table 2.3.

Table 2.3: Comparison between different identification solutions.

Parameters	Barcode	RFID	OCR	Smart cards	Speech recognition
Typical data storage (bytes)	1 - 100	16 - 64k	1 - 100	16 - 64k	-
Machine readability	Good	Good	Good	Good	Expensive
Human readability	Limited	Impossible	Simple	Impossible	Simple
Influence of dirt/damp	Very high	No influence	Very high	Possible	-
Influence of covering	Failure	Low	Failure	-	-
Influence of direction and position	Low	No influence	Low	Unidirectional	-
Degradation	Limited	No influence	Limited	Contacts	-
Total purchase cost	Very low	Medium	Medium	Low	Very high
Operating costs	Low	None	Low	Medium	None
Reading speed	4s	0.5s	3s	4s	> 5s
Maximum reading distance	0 - 50cm	0 - 10m, UHF	< 1m*	Direct contact	0 - 50cm

*Handheld portable scanners usually need to be very close (<1-2cm) to the text to be effective, but nowadays even a normal camera with the appropriate processing software can capture images with texts from far away and interpret them (within a certain text size limit).

It should be noted that some of these technologies are not applicable in our case, such as smart cards or voice recognition, as they do not bring any benefit to the current solution.

2.2 Localization based on Radio Frequency (RF) transmission

Various different wireless technologies can be implemented in order to locate objects indoors, sometimes taking advantage of already installed infrastructure to achieve this. We generally follow the classification presented in [6], which surveys solutions for this specific problem.

2.2.1 Received signal strength indication

Positioning based on received signal strength (RSS) uses the attenuation of the RF signal as it travels in space to estimate distance, where the greater the distance, the weaker the signal. This way, in an environment with at least 3 nodes that have known positions (called anchors),

we can device where another node is by assessing the RSS from the transmissions of the anchor nodes, assuming they transmit with the same power and the propagation model is similar, doing triangulation. The surroundings where the system is implemented play a big factor in attaining a trustworthy location, since the accuracy is not perfect.

One common triangulation method is named *fingerprinting*. During a first setup phase, or offline phase, a database with known training locations (anchors in predefined places) and the RSS of these locations is built. Comparing the location of our device with this database, it is possible to determine our position.

Various technologies were studied with RSS location in mind, asserting advantages and disadvantages. For example:

1. Bluetooth low energy (BLE), through the recent development of the Internet of Things (IoT), has seen an increased usage in various scenarios and indoor tracking is one of them. It offers a low power and cost efficient solution for all kinds of applications, being a well documented and used technology. For our purpose one concern, which is common to all RSS technologies, is the accuracy of the method in question. In [45], this and a few other technologies were tested in different scenarios and it achieved an average error of 0.753 meters, although this is severely affected by the testing environment.

2. WiFi IEEE 802.11 is in every household and practically in every computing device, so its availability and usage is tremendous, with low risk of having no WiFi within the zone. The utilization of wireless devices with internet connection and low power consumption like the ESP32 and ESP8266 in the IoT area is rapidly increasing ([33] and [38]). The use of a device like this attached to the trolley would be able to determine its position within the line (using the RSS of different wireless anchors), and communicate with the appropriate stations.

   The use of this technology for RSS based triangulation is probably the most used, and various papers have thoroughly studied its performance. In [50] a study in unknown environments was made, with the best result being an error of 5.6 meters (in a 60 m by 43 m stadium), while in [45] the error (known environment, comparable to the bluetooth test) was 0.664 meters.

3. LoRa is a recent wireless data communication technology, developed for low throughput over large distances (10km or more in rural areas), operating in multiple frequency bands (typically 868 MHz in Europe). It is a wideband technology that works point to point or in a mesh topology. LoRaWAN is the upper layer, a media access control (MAC) protocol layer, on which LoRa operates (figure 2.2).

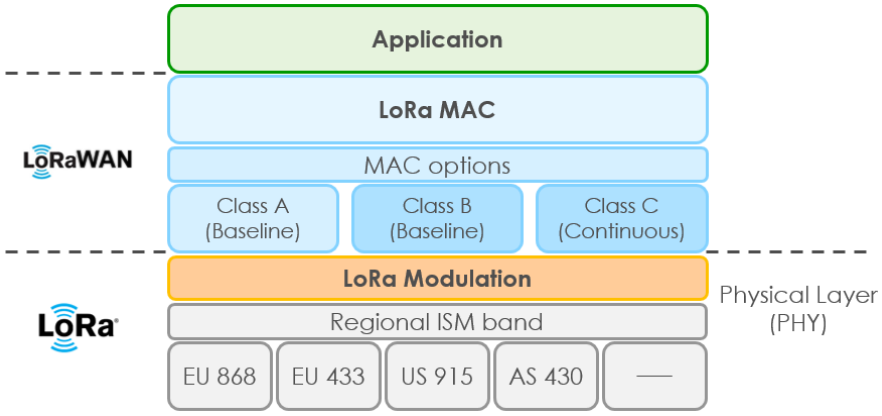


Figure 2.2: LoRa and LoRaWAN layers.

In [39] and [45] we can assess the accuracy of this technology, using the RSS method, in different scenarios. For our purposes, it performs worse than WiFi and BLE in terms of accuracy and precision, although it makes use of a different frequency region compared to those.

4. SigFox is a proprietary technology takes uses a so called "Ultra Narrowband" signal that passes freely through objects (868 MHz in Europe), it has a one hop star topology and it is extremely resilient to interferences. It has roughly the same bandwidth has LoRa and supports multiple devices. Base stations can receive SigFox messages from 10 km to 50 km away, depending on the area.
   In [22], this technology was tested with the fingerprinting method on Belgian postal services cars, commuting throughout the city center and were used 84 SigFox base stations to populate the database and pinpoint the devices. A mean localization error of 340 meters was achieved in the best conditions. For the project in question, this being a proprietary technology and showing no real benefits against LoRa, it is an inappropriate choice.

5. NB-IoT, similar to SigFox, it is a proprietary technology, but has the big difference of being compatible with traditional cellular networks, where a GSM antenna can be re-purposed as an NB-IoT carrier. It is based on the cellular communications LTE, GSM and UMTS. It is reliable (retransmissions based on hybrid automatic repeat requests, or HARQ) and it has less complexity than devices using regular LTE [28]. Similar to SigFox, this being a proprietary technology makes it unusable in our project.

6. ZigBee is based on the specification IEEE 802.15.4, it is a low data rate, low power consumption and low cost wireless networking protocol. It can be implemented in mesh networks and is comparable with bluetooth, albeit with more range but lower bandwidth. It can transmit from 10m-100m apart, node to node, and operates in 2.4GHz (global), 915MHz (Americas) or 868MHz (Europe) [29]. In [16] and [59] we can see that these implementations of ZigBee in indoors localization scenarios have around 1m of error, cor-

roborated by [45]. Keep in mind that these errors are very dependent on the environment where the test was conducted, but these are very similar results in three different articles, which is promising.

2.2.2 Time of flight (ToF)

Time of flight is a principle of distance measurement, where the parameter that is being captured and sequentially analysed (time interval) is converted into distance by multiplying it with the speed at which the signal travels. Two common methods that use time of flight to estimate distance are the time of arrival (ToA) and time difference of arrival (TDoA) [36].

However, this definition is not consensual, meaning that between papers there seem to appear two different versions of what ToF is. Using the principle above, ToA and TDoA require some form of synchronization (ToA between sender and receiver and TDoA between multiple receivers). Another version uses ToF to measure distance, without any synchronization (by measuring the round-trip delay in the same device). Some papers mention ToA and ToF as being the same (see [7]).

2.2.3 Time of arrival

Time of arrival calculates distance between a transmitter and a receiver by knowing the time it takes the signal to propagate between two points in space (two nodes). Because the signal propagation speed is constant for electromagnetic waves (in air, the speed is nearly equal to the speed of light, i.e., $c \simeq 3 \times 10^8$, since it is divided by the index of refraction of the medium, which, for the air, is nearly 1). With the time interval between the device and the anchor, it is now possible to draw a circumference (equation 2.2) around the fixed station (x_{ref}, y_{ref}), with its radius being the distance we calculated (equation 2.1), and by having at least three different stations and using a triangulation technique [25], it is possible to determine the position of our device (see also [37]).

$$radius = (t_{station} - t_{device}) \times c \tag{2.1}$$

$$radius = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2} \tag{2.2}$$

However, using this method we must be wary of our surroundings, because reflections and occlusions by foreign objects can occur and it requires special hardware to calculate with accuracy these travel times, otherwise an error as small as a few nanoseconds will cause the final value to be incorrect.

Clock synchronization, noise, sampling artefacts can all create estimation errors in ToF applications. In article [26], Steven Lanzisera, David Zats, and Kristofer S. J. Pister achieved an accuracy better than 3 meters for 80% of the tests that were made, in a 50 m by 40 m open area.

It is normal to achieve better accuracy compared to the RSS method described above, although it can be combined with other ones for better accuracy. For example, in article [57] ToA and Angle of Arrival (AoA) are merged and in article [41] ToA, RSS and AoA are used together.

Ultra wideband (UWB) for real time localization systems (RLTS) make use of this method, although some systems use the time difference of arrival (TDoA) to determine position coordinates (X,Y,Z) of nodes (also called tags). It uses a set of fixed nodes with known positions (anchors) and calculates the distance, from at least four anchors, to the tag to be localized [23]. The following diagram illustrates the disposition of the system, figure 2.3.

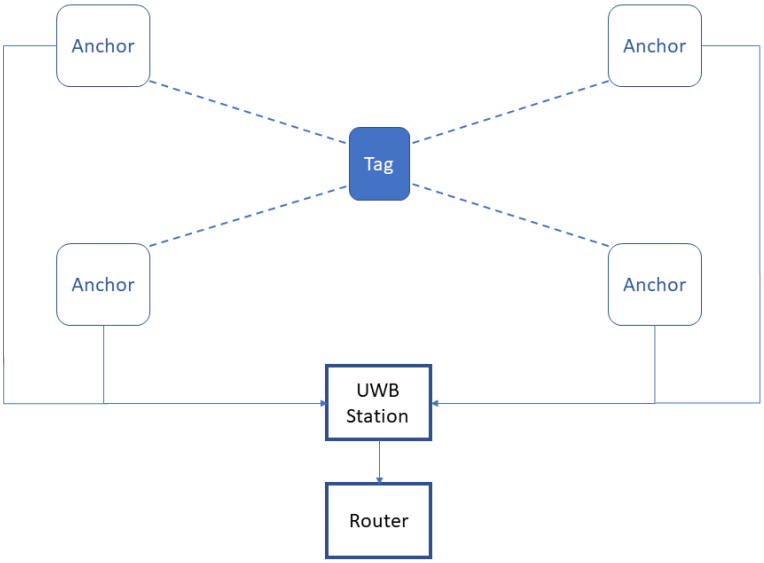


Figure 2.3: Ultra wideband diagram.

2.2.4 Time difference of arrival

Systems that use this method, use one of two schemes to pinpoint a device location. Either a signal is generated from the device and received by several anchors, where the time it took to reach each anchor (TDoA) is then processed, or the opposite happens, where the anchors broadcast synchronized signals and the device measures the TDoA. These time differences can then be used to compute the position of the device, being this method a time of flight distance measurement technique.

As referred in the previous section, some ultra wideband technologies use this method [54]. It requires that the anchors have to be synchronized, meaning that it is more complex to setup than the ToA method, with still some of the same drawbacks.

2.2.5 Angle of arrival

This method makes use of the angle at which a signal arrives at the anchor or receiver. This is possible by having an array of directional sensors and by knowing which sensor received the strongest signal, we can determine where it came from. Another method uses the measure of the TDoA of multiple antennas in an array to determine the AoA of a device. For the project in question, the reliability and accuracy of the method in our environment has to be tested and well documented, so it can satisfy all the requirements [24].

2.2.6 Phase difference of arrival (PDoA)

This method was first introduced in radar systems, but has seen an increased usage in distance estimation of RFID systems [40]. It consists in transmitting continuous wave signals at two different frequencies, measuring the phase difference between both at the receiver. That difference is proportional to the distance and this technique is very accurate. However, phase periodicity (2π), reflections, and everything that can obstruct the signal can cause deviations in the final result.

2.2.7 Radio frequency identification

RFID systems have many industrial platforms where there is a lot of development and research being made to improve this technology [3] [2]. It has multiple advantages such as non-contact, non-line-of-sight (NLoS), low power consumption and simplicity [43]. It is already found in multiple logistics centres, malls, parking systems and so many others. Naturally, tracking with RFID is also a use of this technology [11], where in paper [6] it is mentioned the PDoA method with RFID tags. A device with an RFID reader, going through a predefined path, can have tags along the way, so when it senses one it knows where it is located (RFID antenna power can be customizable so it only reads within centimetres).

2.3 Non-RF localization

Some technologies can be used for localization without the use of wireless communications. We will briefly present the most common ones, namely magnetic, ultrasound, inertial and camera based positioning technologies.

2.3.1 Magnetic positioning

These systems make use of the earth's magnetic field to determine the devices location. Normally used in tracking people through the use of their smartphones, indoor localization methods of this kind have to account for steel reinforced concrete and iron inside buildings that create variations in earth's magnetic field, distorting it [58]. Provided that the building is not changed, those distortions will not change either, so the anomalies will be invariant, allowing for the correct calibration and position measurement.

2.3.2 Ultrasound positioning

Like the majority of car park alert systems, ultrasound sound waves can be used for location estimation [46], the techniques used for this purpose are similar to those used in Radio Frequency, the most commonly used being time of flight. However, because these waves are much slower than those of radio frequencies, the ToF method is much more accurate.

With this technology it is possible to achieve accuracies better than 20 cm, see [19].

2.3.3 Inertial measurement

This method uses dead reckoning to calculate the current position. This means that, with information from previously determined locations, and using sensors that allow to track movement, it is possible to calculate the current device position [60]. In inertial measurement, these sensors are normally a package of inertial sensors like gyroscopes and accelerometers. By capturing a specific force (combination of a inertial linear and gravitational acceleration [6]) and angular velocity of the device, we can track and map the position of the device (given an initial known location).

2.3.4 Camera based positioning

There are many camera based positioning systems, like using reflected ceiling light [48], landmarks [34] or known visual features. The last ones consist of having a database of snapshots from the devices camera with those exact locations attached to it, this way when the device recognizes one of the stored snapshots, it knows its location and can perform corrections. It is often coupled with dead reckoning for moving (e.g. a robot) between locations. Landmarks use mostly the same technique, but instead of snapshots, they save the coordinates with visual markers detectable by the device's camera.

2.4 Comparison between localization solutions

This section presents the solutions described above compared against each other in terms of performance (accuracy and precision), reliability, availability and cost, see table 2.4 [13], [8], [47], [40], [15], [31].

Table 2.4: Comparison between different localization solutions.

Technology/Technique	Performance	Reliability	Availability	Cost
Received Signal Strength	Medium	Medium	High	Low
Time of Arrival	High	High	High	Low
Time Difference of Arrival	High	High	High	Low
Angle of Arrival	Medium	High	Medium	Medium
Phase Difference of Arrival	High	Medium	Medium	Medium
Radio Frequency Identification	Maximum	High	High	High - Low
Magnetic Positioning	Low	Low	Low	High
Ultrasound Positioning	Maximum	Medium	Low	Medium
Inertial Measurement	Low	Low	Low	Medium
Camera Based Positioning	High	Low	Low	Medium

Solutions that reuse existing infrastructure (most of the radio frequency localization) are, generally, less expensive and offer higher performance in our desired environment (high traffic, metal structures), by having lower average error distances and higher accepted position estimations, and have, typically, less complex implementations. Some non-RF methods can be very precise, e.g. ultrasound, but since this requires additional infrastructure and a slight concern for signal reflections and occlusions exists, this can pose several problems (mistaking an abnormal trolley behaviour for an arrival at a station) [15].

2.5 Digital Twin

In the literature, we can find different definitions for Digital Twin (DT), such as the following two: "Digital twin is an integrated multi-physics, multi-scale, and probabilistic simulation of a complex product and uses the best available physical models, sensor updates, etc., to mirror the life of its corresponding twin." [52], "Digital twin is a life management and certification paradigm whereby models and simulations consist of as-built vehicle state, as-experienced loads and environments, and other vehicle-specific history to enable high-fidelity modelling of individual aerospace vehicles throughout their service lives." [18]. These definitions emphasize the connection between a physical entity and its corresponding digital model, being this connection established by providing the model with real time data from different sensors [56].

This concept was adopted by the on-going Industry 4.0 process and allows for the creation of a digital factory environment, which in turn enables new optimization operations and traceability venues. The idea of following a product across its entire life cycle, from concept to disposal, is already a well established and used business activity, known as product life-cycle management (PLM) [49], but associating it with a digital twin has a lot of advantages. With new information technologies like the Internet-of-Things (IoT), acquiring various data from an array of different sensors has never been so easy, accelerating the research and development of various Industry 4.0 aspects (e.g. digital twin). Figure 2.4 shows a simple DT concept.

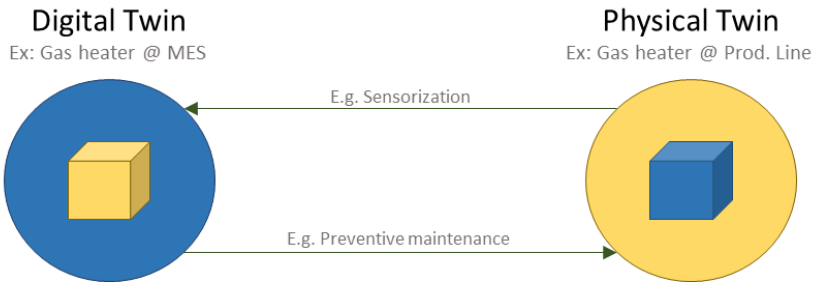


Figure 2.4: Digital twin concept.

In the context of an industrial shop floor, a DT continuously captures and evaluates information retrieved from the physical product during its production process and creates a virtual object with the same characteristics and defects (if present and detected by the sensors), as well as an assortment of data associated with it from all the tests that the product was subjected to. Overall, analysing the real twin or the digital one yields very similar operational results.

2.6 Summary

In this chapter we briefly surveyed different techniques and technologies for localizing devices in indoor spaces and also identifying them, which are fundamental to implement the concept of Digital Twin. We gave particular attention to Radio-Frequency transmission technologies since these seem more adequate to the problem at hand. In fact, given the nature of the environment where the project is going to be implemented, with many moving parts or entities (trolleys, operators, different products), achieving high reliability and accuracy eliminates most of the non-RF methods and identification techniques. For example, camera based positioning and inertial measurement are not robust enough for the time we are trying to save, if the dead reckoning is off even by a couple centimetres, the identification process may not trigger. Overall, these two methods are more error-prone in terms of precision, and they do not transmit information between the station and trolley, the design of another system that makes use of this tracking data and makes the trolley/product association is required.

Additionally, since the entire factory floor can affect the magnetic positioning technique, this seems like a poor choice, as well as ultrasound, given the amount of objects that can pass around the trolley at any given time.

For the identification solutions, speech recognition would not be very reliable in such a noisy ambient, OCR and barcodes would not improve the current solution (both have to use handheld devices and barcodes are currently in use) and smart cards also don't completely automatize the process.

Therefore, a Radio-Frequency transmission technology is, to our understanding, the best choice for this scenario, and in particular, RFID seems to be the most reliable method to assure that there are no errors and delays in the product identification phase.

Chapter 3

Problem characterization

This chapter presents the problem that this dissertation proposes to solve. Figure 3.1 illustrates the current layout of the line where the desired system is going to be deployed.

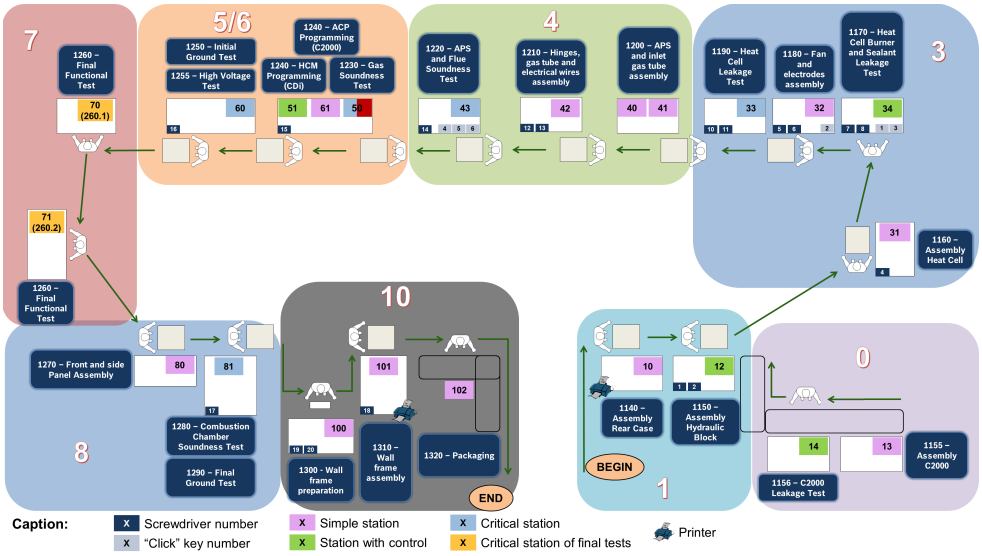


Figure 3.1: Layout of the testing line under analysis.

There is a considerable variety of different stations in the line, all with particular procedures to be done to the product. Its association with the trolley is made at station 10 and proceeds throughout all the different stations (except the station block marked as 0 on the figure), finishing on 102.

Every coloured block (marked from 0 to 10) represents a different automation system composed by a central PLC (NEXEED Control, Rexroth Indralogic L40) and a Bus Coupler (Beckhoff BK3120) for every station, all interconnected by the fieldbus communication protocol Profibus-DP and independent between each other. Every PLC communicates directly with the recipe and MES server, either to receive the order of operations or to send data relative to the product being worked on.

3.1 Problem definition

The problem that this dissertation aims to solve is closely related with the in-doors tracking problem, given that, knowing where the trolleys are at all times is very valuable. Figure 3.2 provides a generic overview of the operations done at a station in this line.

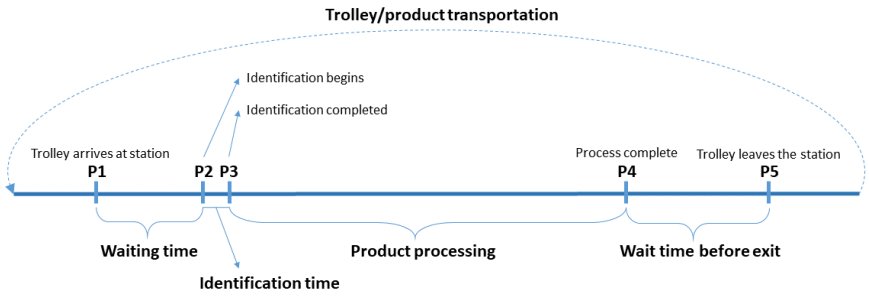


Figure 3.2: Generic order of operations in a station.

Point P1 marks the arrival of a trolley at the station, where it waits until the product identification begins (P2). P2 to P3 corresponds to the time it takes for the identification to take place and when it does the product code is sent to the MES followed by the response with the recipe. Afterwards, from P3 to P4, the process/testing takes place (and when completed, a report is generated, figure 3.3) and from P4 to P5 the trolley is waiting before departing for the next station, and finally on P5 the product leaves and is transported to the next workbench. Figure 3.3 shows a XML report that is sent to the MES at the end of every product process.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header version="V2.x" eventId="200" eventName="partProcessed" contentType="3" timeStamp="2019-01-25T12:10:33.3805530+00:00">
    <location lineNo="7" statNo="33" statIdx="1" fuNo="3" workPos="1" toolPos="1" processNo="1190" processName="Heat Cell Leakage Tests" application="PLC" />
  </header>
  <event>
    <partProcessed identifier="837091700079477336000055P" />
  </event>
  <body>
    <structs>
      <resHead result="1" typeNo="7733600055" nioBits="0" cycleTimePrev="112870" />
    </structs>
    <structArrays>
      <array name="Stat033ScanningData">
        <structDef>
          <item name="class" dataType="8" />
          <item name="compIdentifier" dataType="8" />
          <item name="batch" dataType="8" />
          <item name="state" dataType="8" />
          <item name="manufacturer" dataType="8" />
          <item name="typeNo" dataType="8" />
```

Figure 3.3: PLC report generated at station number 33.

The current report has a considerable amount of information, where some of it stands out for the problem at hand, specifically the timestamp of the generation of the report, the station and line identification, product code id, the processing time ($t_{P4} - t_{P3}$, figure 3.2) and some test specific values.

Currently, when a trolley arrives at a station (fixed stopping place), it waits for a few moments ($t_{P2} - t_{P1}$) and then an operator needs to manually grab a barcode reader and scan the product code,

which takes $t_{P3} - t_{P2}$ seconds (figure 3.2). This reader is connected to the PLC embedded on the station and it asks the MES for that specific product recipe when the operator scans it. After the MES responds with the recipe, work can be done at that station.

After careful evaluation, we conclude that the removal of barcode manual scanning times is crucial in improving the line effectiveness and efficiency, automating the identification and association process. To achieve this we need to know, with precision, when a trolley arrives (and leaves) the stations, to effectively trigger the message sent from the PLC to the MES when the trolley reaches a station, bypassing the current barcode reader.

The figure 3.4 provides the proposed order of operations that will take place once the system is installed.

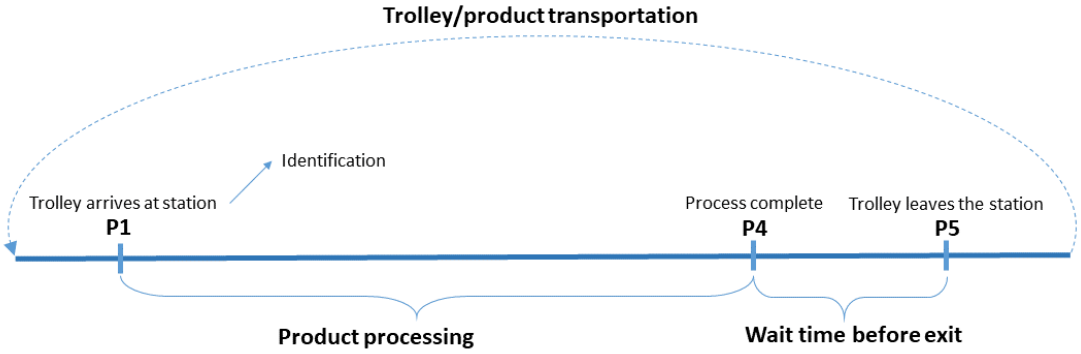


Figure 3.4: Proposed order of operations.

As we can see, on P1 when the trolley arrives at the station, the identification will automatically take place, starting the product process. This system will effectively remove the time between P2 and P3 (see figure 3.2) and also reduce the time between P1 and P2, by starting the product processing earlier (some processes have a start-up time that can start immediately without any operator input).

Given the time it takes for the barcode read to take place, with the price per minute for the given product, we can calculate, roughly, how much money we can save by automating this process.

The current time spent at each station by the product and the production forecast can be seen in tables 3.1 and 3.2:

Table 3.1: Time spent reading the barcode in each station by the operator.

Station	Barcode read time (s)	VT/100 units (min)
P10	0	0
P12	3.7	6.2
P34	2.1	3.5
P33	2.1	3.5
P40	2.1	3.5
P42	2.1	3.5
P43	3.7	6.2
P50	2.1	3.5
P51	2.1	3.5
P60	3.7	6.2
P61	2.1	3.5
P70	2.1	3.5
P71	2.1	3.5
P81	2.3	3.8
Total	**32.3**	**53.9**

Table 3.2: Forecast production figures for 2019 on line 7.

Month	Equipment: CDi	Equipment: C2000i
January	5,712	432
February	4,140	756
March	2,688	1,596
April	2,184	1,032
May	2,988	2,112
June	1,556	1,308
July	2,952	1,908
August	2,952	1,056
September	2,264	3,288
October	3,260	3,948
November	2,624	3,516
December	312	4,680
Total	**33,632**	**25,632**

With the above data, we can evaluate the economic viability of the project, calculating the amount of money that can be saved yearly by reducing the time the operator is handling each product (VT time). On table 3.3 we can verify the savings of the project.

Table 3.3: Project savings.

Equipment	Price/min (€)	Time saved (s)	Savings/year (€)
CDi	0.19	32.3	3,439.99
C2000i	0.19	32.3	2,621.73
		One year total	**6,061.72**
		Two year total	**12,123.44**

The project savings have a total sum of 6,061.72€ per year, by just bypassing the barcode reader. In addition to this value, this system can help in the identification of other problems, inefficiencies and bottlenecks in the line (with a more accurate knowledge of the trolley location), allowing for improvements as well as a slight ramp up in production from the fact that we are actively saving time with each product. Moreover, this project can be applied to other lines (currently 4), increasing the direct value that it provides.

Another aspect of the problem is as follows, what to do with the amount of data stored in the MES. With detailed trolley arrival and departure timestamps, processing times and change over times, it is possible to optimize the overall production process, as well as to create a tool for online tracking of the current status of the line, among other possible potential benefits.

Therefore, we extended our objectives to include the development of a graphical tool to represent the state of all trolley DTs in real-time. Moreover, we wish to integrate the new functionality with the current on-site technology for a scalable and all-in-one solution.

3.2 Proposed solution

Given the objectives and nature of the project, reliability, accuracy, precision and robustness is of the utmost importance, errors in identifying the trolley station are not tolerated, otherwise the wrong recipe will be given to the wrong station by the MES and that can cause the total loss of the product.

With these requirements in mind, the solution that we propose is an RFID system that physically enforces a positioning that favours a correct reading. In particular, we place the RFID tags on the trolleys and readers on the stations in such a way that when a trolley stops at a station (figure 3.5), the tag and the reader are centimetres apart. By checking an optical sensor that is embedded in the stopper, we can make sure that the trolley is safely in place during the read and write operations.

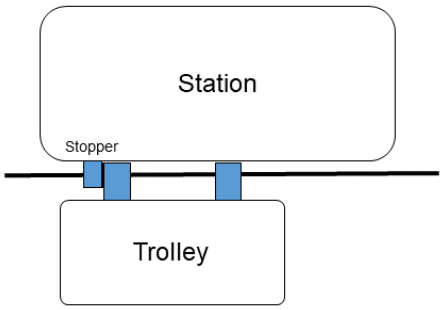


Figure 3.5: Station stopper layout.

In the first station, an association of the trolley to that product must be made (in order to remove subsequent barcode readers), writing the tag that is on the trolley with the product code (up to 25 character code) that it is carrying, identified by a barcode, and cleaning it at the end of the line. This code can be read at every station, although the manual barcode reader must remain at the first station in order to write the correct code into the tag.

When a trolley reaches a station and stops the reader will detect the product code and send it to the PLC integrated on the station. That PLC will proceed to ask the MES for the recipe with the code from the tag and work at that station can be started.

When the work is complete, the station PLC can attach to the report the RFID read/write status and also the product process time, exit time and change time. We can effectively remove the time between P2 and P3 (figure 3.2) and with the timestamps at P4 and P5, we are able to study various different times, such as, how long it takes for a trolley to leave the station after the work is complete.

This procedure is repeated in the rest of the stations and the RFID tag can be reused for a new product when it passes the final station. This way we have accurate knowledge of where the trolley is at all times, relevant information (time wise) is extracted from the line and the manual barcode readers are successfully replaced by an automatic method.

Table 3.4 shows a non-exhaustive list of existing UHF RFID technology that suits our system requirements, note that in this table there are readers with integrated antennas, as well as reader gateways, where the antennas must be bought separately.

Table 3.4: UHF RFID devices.

Ultra High Frequency Solutions			
Equipment	**Price (€/un)**	**Description**	**Manufacturer**
M-ARU-ETH-E6	598,41	Integrated antenna	Kathrein
RRU 4400	893,86	Reader with 4 antenna ports	Kathrein
MIRA-100-Circular	90,10	Mid Range antenna, up to 2 meters	Kathrein
LORA	110,84	Low Range antenna, up to 20 centimetres	Kathrein
IUH-F190-V1-FR1-01	1.300,55	Integrated antenna	Pepperl+Fuchs
BIS U-6020-053-104-00-ST26	2.750,00	Reader with 4 antenna ports, RS232	Balluf
BIS U-301-C0-TNCB	334,00	Integrated antenna	Balluf
BIS U-6027-054-104-06-ST27	2.900,00	Reader with 4 antenna ports, Ethernet TCP/IP	Balluf
DTE810	2.043,93	Reader with 4 antenna ports, Ethernet TCP/IP	ifm electronic

Alternatively, we can also opt for a cheaper technology using HF RFID as shown in Table 3.5.

Table 3.5: HF RFID devices.

High Frequency Solutions			
Equipment	**Price (€/un)**	**Description**	**Manufacturer**
RF210R	170,00	Small integrated reader, up to 20 millimetres	Siemens
RF220R	200,00	Small integrated reader, up to 35 millimetres	Siemens
RF240R	220,00	Small integrated reader, up to 65 millimetres, RS232 ASCII	Siemens
ASM 456	186,00	Siemens communication module	Siemens
BIS M-400-007-001-00-S115	417,60	Similar to RF220R	Balluff
BIS M-400-045-401-07-S4	318,00	Similar to RF210R	Balluff
BIS M-402-007-002-00-S115	617,00	Reader with separated antenna	Balluff
BIS M-401-007-001-00-S115	457,00	Similar to RF240R	Balluff
TBEN-S2-2RFID-4DXP	325,00	Turck communication module	Turck
TB-M18-H1147	154,38	Small integrated reader, up to 30 millimetres	Turck
TB-M30-H1147	154,38	Small integrated reader, up to 45 millimetres	Turck
TN-M30-H1147	154,38	Small integrated reader, up to 77 millimetres	Turck
TN-M30-IOL-H1147	190,00	Small integrated reader, IO-Link, up to 77 millimetres	Turck

For the information storage, the tag must have, at least, the capacity to store the product code (25 characters) in user data storage (for UHF, see EPCglobal Gen 2 tags [21]).

With the information above, we can observe that a UHF solution is substantially more expensive, compared to HF, and the main advantages it offers, namely higher read distance and number of tags read per second, are not relevant to the problem at hand. However, it is possible to group readers together, due to the many different antennas one gateway can support, allowing for two distinct solutions, e.g., using UHF and multiple antennas or RFID HF.

3.2.1 UHF Solution - Multiple antennas

Grouping different stations to the same reader (with various antennas) can decrease the number of readers that exist in the line. By analysing the figure 3.1, it is possible to group sets 1/3 and 7/8 (stations 100, 101, 102, 80, 31, 32 do not need a reader), effectively decreasing the number of readers while maintaining the same antennas, we can analyse the different solutions in table 3.6.

Table 3.6: UHF solutions prices.

Grouping stations	RFID Readers	Antennas	Price (€)
NO	6*	14	6,624.56***
YES	4**	14	4,836.84***

*Using 6 RRU 4400 Kathrein equipments and 14 MIRA-100-Circular antennas.

**Using 4 RRU 4400 Kathrein equipments and 14 MIRA-100-Circular antennas, by sharing equipments between PLCs to take advantage of the cheaper cost per antenna.

***These prices do not include any cables (antenna cables, ethernet/serial communication cables) or other accessories (PLC bus terminals and power transformers for the readers)

Grouping RFID readers, and communicating with different PLCs with the same reader has a clear price advantage over the other solution, with minor changes.

3.2.2 HF Solution

Concerning HF RFID, the average price of these devices is usually lower, being this a good route to follow for the proposed solution. After careful evaluation of the state-of-the-art in terms of HF RFID and some meetings with a few distributors, namely Bresimar Automação S.A. for the Turck GmbH products and Siemens Portugal S.A., three main solutions emerged as follows:

1. RFID readers with integrated antennas that communicate through the protocol IO-Link (industrial communications networking standard IEC 61131-9 [17]) to the Bus Coupler (specific IO-Link bus terminal needed);
2. RFID readers with integrated antennas that communicate through serial communications (RS232);
3. RFID readers with integrated antennas that communicate with a gateway device, which uses Profibus-DP to connect to the PLC, effectively using only one gateway per PLC and connecting up to 4 antennas to this device.

3.2.2.1 Solution 1 - IO-Link

The IO-Link protocol is an open interface standard described in IEC 61131-9 [5] that consists of an IO-Link master and one or more devices (e.g. sensors, RFID readers, valves, I/O modules, motor starters), where the master establishes the connection between the devices and the automation system. The main benefit of this protocol is the tool-supported parameter assignment, where all device configurations are saved in the master, simplifying the replacement of devices that are hard to configure.

This solution with Siemens AG hardware can be seen in table 3.7.

Table 3.7: Siemens IO-link solution.

Equipment	Description	Quantity	Price per unit (€)	Total
RF240R - IO Link	Reader with integrated antenna and up to 65mm read range	14	264.29	3,700.06
Tag MDS D424	Standard HF tag 2k bytes memory	20	14.86	297.20
Spacer	Tag spacer for mounting in metal surfaces	20	2.86	57.20
RF200 Cable	IO-Link Connecting cable	14	27.86	390.04

The solution amounts to 4,444.50€ excluding the Beckhoff Automation GmbH IO-Link bus terminals which should be around 150-250€ for each of the 14 stations.

3.2.2.2 Solution 2 - RS232

In this case, the solution (table 3.8) uses the serial communication RS232 with an ASCII based protocol to communicate between the PLC and the reader.

Table 3.8: Siemens RS232 solution.

Equipment	Description	Quantity	Price per unit (€)	Total
RF240R - RS232	Reader with integrated antenna and up to 65mm read range	14	221.43	3,764.31
Tag MDS D424	Standard HF tag 2k bytes memory	20	14.86	297.20
Spacer	Tag spacer for mounting in metal surfaces	20	2.86	57.20
RF200 Cable	Connecting cable with external power and ground supply (optional)	14	10.00	140.00*
KL6001	Beckhoff Serial bus terminal	14	189.45	2,652.3

*These cables can will be made here on-site, as the RFID connectors are standard M12 8pin.

This RS232 solution, amounts to 6,911.01€ with all the necessary equipment to implement it.

3.2.2.3 Solution 3 - Profibus-DP

Lastly, the third solution, by Turck GmbH (table 3.9), is composed by a Profibus-DP gateway and RFID readers, it bypasses the bus couplers on the different stations and connects directly to the PLC (by direct access through the field network Profibus-DP).

Table 3.9: Turck Profibus-DP solution.

Equipment	Description	Quantity	Price per unit (€)	Total
BL67-GW-DPV1	PROFIBUS Gateway Hub where the RFID expansion modules connect	7	174.45	1,221.15
BL67-2RFID-S	RFID expansion module	9	198.75	1,788.75
BL67-B-2M12	Bus terminal for the gateway in which the expansion module connects	9	26.40	237.60
TW-R50-B128	HF tag with 5cm diameter	25	4.35	108.75
RK4.5T-2-RS4.5T/S2503	Connection cable between the antenna and the bus terminal	17	22.90	389.30
TN-M30-H1147	RFID reader with range up to 77mm	17	118.75	2,018.75

The price for this solution is 5,764.30€ .

3.2.2.4 Solutions comparison

Between HF and UHF RFID, we chose the HF option because it is a cheaper solution than UHF, and the problem at hand does not need the advantages of UHF RFID, mainly the longer reading distances and the number of tags that can be read per second.

Among HF solutions, the first one is discarded for being about the same price as solution 2 (section 3.2.2.2), because to the calculated price of 4,444.50 € , we still have to add the 14 IO Link bus terminals, and there is no practical use of the advantages of IO-Link in our problem, so the RS232 solution takes priority over this one. Also there is more know-how inside the factory on the second solution (some production lines use HF RFID in other applications), where as IO Link is still not used here.

Between 2 and 3, they are about 1,100€ apart, favouring solution 3, however, technologically they differ substantially, either in hardware or software. The main advantage of the RS232 solution is the lack of gateway and extra hardware, making the communication more direct and streamlined. Additionally, the hardware for solution 3 is quite bulky and some electrical switchboards (where the PLCs are housed) do not have the space to accommodate that hardware, which implies an extra cost, by having to change the switchboard, that the company does not want to support.

Nonetheless, test material was ordered for both solutions and tests were conducted to determine which one was the best. Both confirmed their advertised read ranges and worked flawlessly and instantly, so the deciding factor rested upon company maintenance difficulty and simplicity, where the RS232 solution gained to the detriment of the other one.

3.2.3 Nexeed Control and Nexeed MES

As stated earlier, the automation controlling the different stations is called Nexeed Control, and it is a control engineering system for Bosch and has a vast range of functions, such as: Soft-PLC (programmable according to IEC 61131-3), data administration, data concept with master computer linkage, production and machine data acquisition, scalable control engineering by reusing software modules, object oriented programming, MES linkage and much more.

The standard hardware architecture of the stations controlled with Nexeed Control is shown in figure 3.6.

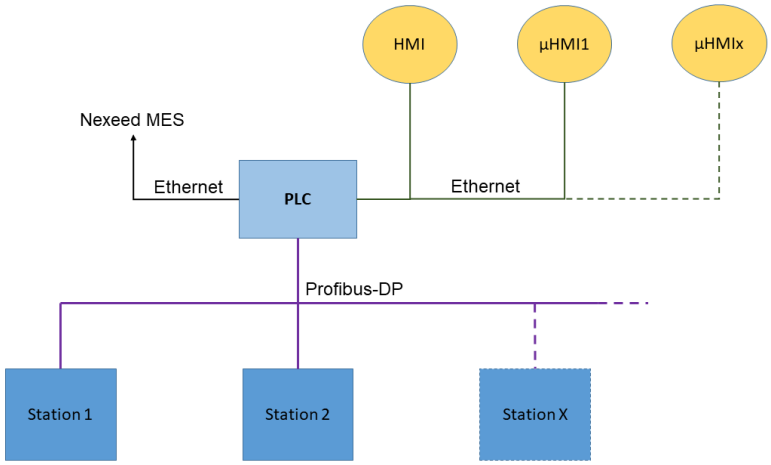


Figure 3.6: Hardware architecture used with Nexeed Control.

The proposed solution (HF RFID - RS232) adds serial communication bus terminals to the bus couplers on the different stations. In terms of software, a function block library shall be developed to communicate through to the RFID reader (a lower level function block to operate the RS232 and a higher level handler that implements the Siemens AG ASCII protocol and sends data to the lower level).

Furthermore, every station shall communicate with the MES to indicate if the product was identified via RFID or barcode, as well as the timestamps mentioned previously (change over time, exit time and processing time).

All the data already collected and the new data to be added shall be accessible through the web portal to the MES server and visualized there. That data is stored in an ORACLE database for access. A diagram of the services and servers can be seen below (figure 3.7).

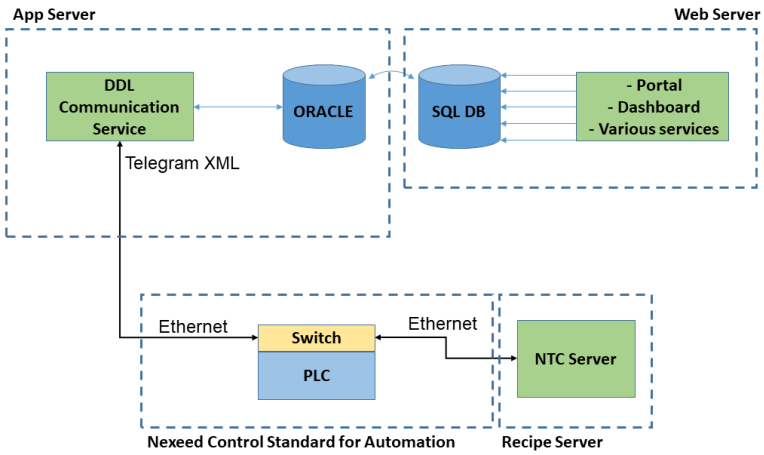


Figure 3.7: Nexeed MES architecture

As stated previously, the reports are sent through XML telegrams, then they are processed by the DDL service and stored in the main database. After that they are copied into another SQL database, to add data redundancy for higher reliability.

3.2.4 Data Analysis

With the identification system implemented and the new data added to the MES, we will have at our disposal a wide range of product and trolley data, enabling the creation of digital twins of the products. It will be possible to extract the following information regarding each product.

- Real time positioning over the entire production cycle (linkage between product - station, there is enough accuracy to tell that the product is between stations, but not where exactly);
- Test results that were conducted at every station (including defective products);
- Detailed cycle, arrival, processing and departure times.

This data treatment shall be made by a web-based application, we propose the use of the Microsoft framework ASP.NET MVC for that purpose. It will communicate with the MES database, treat all the data into useful statistics and also display a real time map of the line in question, we can see an early mock-up of the website below (figure 3.8).

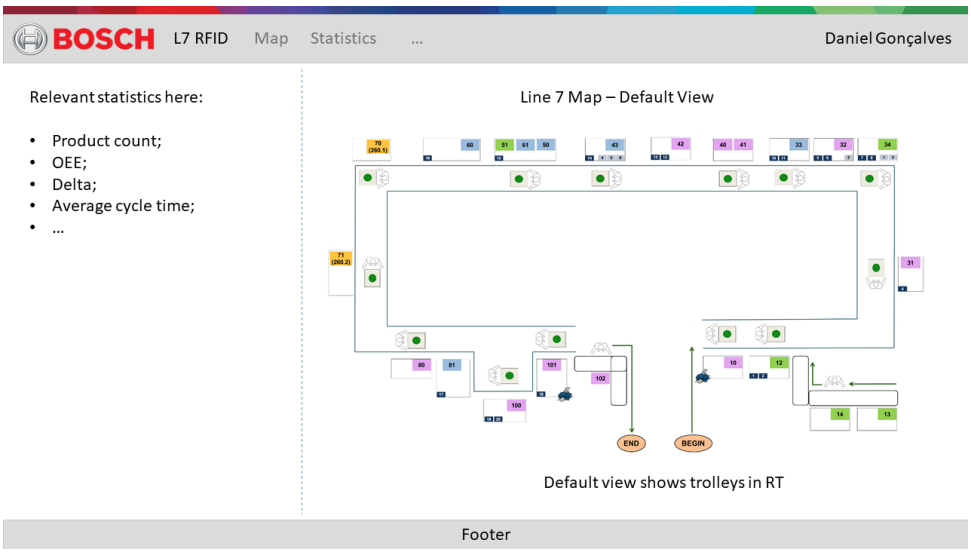


Figure 3.8: Early web application mock-up

The map on the right can support several different applications and functionality, such as being able to click on the products and access their process history. On the left a list of the most useful statistics can be checked, by analysing the MES data that we are introducing with the RFID system, it is possible to compare average times (departure, arrival, processing, cycle), among other data.

3.3 Summary

In this chapter we addressed the details of the problem at hand, with quantitative information on the benefits that can be potentially achieved. Based on this information, we defined the exact

problem as well as a desired solution and characterized its potential main components.

Chapter 4

Automatic Identification Implementation

This chapter describes the RFID implementation, from the ASCII protocol programming to the physical trolley additions that were made, as well as all the problems that arose from it.

The stations are generally featured with a rail to hold the trolley, a stopper that extends and keeps the trolley in its working position, a manual barcode reader that communicates through RS232 with the PLC and all the process-specific tools to work on the product. Adding to this, there is usually a switchboard that contains the bus coupler and the PLC, depending on the station, right next to the it.

4.1 RFID Programming

The selected Siemens AG RFID reader (RF240R) has two main modes of operation, scan mode and normal mode. Once the reader has been switched to scan mode, it will operate autonomously. This means that the data is transferred automatically to the host via the serial interface when the reader is activated. The type of data acquisition and transfer is preset in the reader using parameters. It is normally used to execute read commands autonomously based on an internal timer or the presence of tags.

In contrast the normal mode responds to commands sent by the host. The following mode variants are with and without presence check, this means that on the first one it sends the host a message saying that there is a tag in its antenna range, where as the seconds one just waits for the host to operate the reader completely.

For our purposes, we are going to use the reader in normal mode without presence check, as we do not want unexpected messages in any stage of the communication process, since we are going to use optical sensors (already embedded on the stations) to detect the presence of trolleys.

In order to use the RFID reader with the Nexeed Control L40 PLCs, a library with two distinct function blocks was created, one that controls the lower level of the serial communication (*SerialComm*) and the other that handles the higher level (*RFIDSerialHandling*), to choose which

message to send, the construction of ASCII messages and the parsing of the answers, coupled with the error handling and redundancy to create a solution as robust as possible. Figures 4.1 and 4.2 show the function blocks.

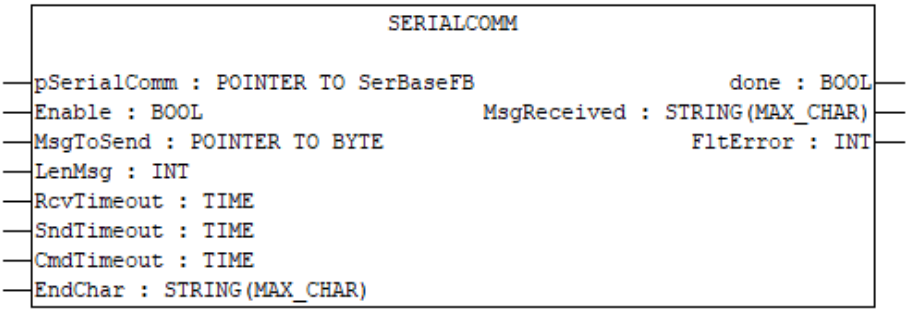


Figure 4.1: Low level RS232 handler.

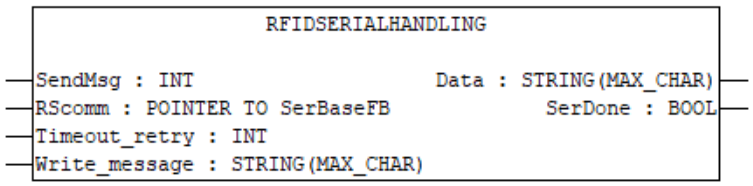


Figure 4.2: Higher level ASCII protocol and SerialComm handler.

The programming languages utilized were Structured Text (ST) and Sequential function chart (SFC), from the open international standard IEC 61131-3 for PLCs [53], for the construction of these libraries and also to edit the main code sequence for the different stations.

4.1.1 Serial Communication Handler

This function block (FB) is called by the *RFIDSerialHandling* and handles the RS232 communication for the selected serial bus terminal. With this, this block will be instanced at every station that has an RFID reader. It receives the following inputs.

- pSerialComm: A pointer to the base serial function block, which is typically an array that we instantiate when adding the bus terminal and it holds the communication parameters;
- Enable: Activate or deactivate the function block;
- MsgToSend: Pointer to an array of bytes that hold the message to be sent;
- RcvTimeout: Receive timeout time (ACK timeout);
- SndTimeout: Sending timeout time (Bus terminal not found);
- CmdTimeout: Command timeout time (Reader not found);
- EndChar: Which character ends the received data frame.

It outputs the data received from the device, a boolean that signals the state of the communication and an integer that shows the error state.

This function block work flow is governed by a state machine which works as follows (figure 4.3).

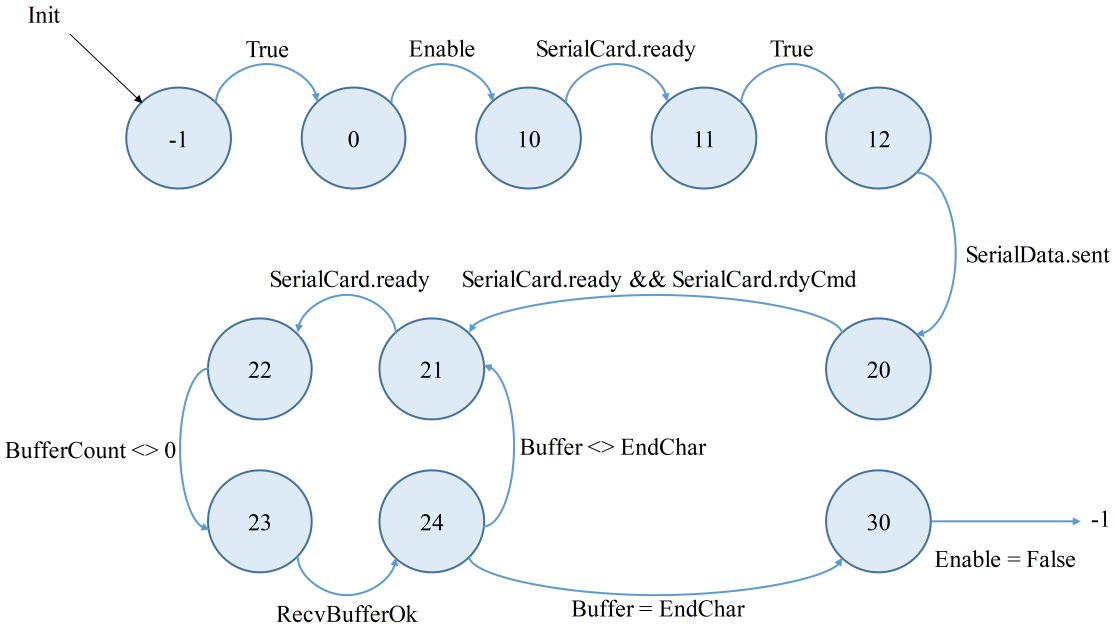


Figure 4.3: Main serial communication state machine

The initial state *-1* does some initializations, such as clearing buffers and variables used in some states. The state *0* is where the program stays, until it receives the signal to send the command and a pointer to the array of bytes to be sent.

Next, it goes to state *10* where it checks if the serial bus terminal (e.g. KL6001) is ready and advances to state *11* if so, otherwise there is a timeout timer that activates and sends the state machine to an error state (*200*, see figure 4.4 and table 4.1) with the appropriate error message (*"Serial send not ready"*).

State *11* sends the data and advances to state *12* where it awaits until the string is sent to deactivate the send flag and proceed to the next set of steps (receive loop).

On state *20*, the serial card buffer is cleaned, the receive data buffer is set to the pointer of an auxiliary variable and some variables are cleaned. After this, the state moves to *21* and it begins the receive cycle (*21-22-23-24*) which will be explained in four different algorithms, one for each state in pseudocode (algorithm 1, 2, 3 and 4).

State *21* waits for the card to be ready to go to the next step or else it starts a timer, which outputs a command timeout error and goes to the error state.

Algorithm 1 State 21 of the receive loop which verifies if the card is ready.

```
1: State 21:
2: if Serial bus terminal ready then
3:     Set buffer deletion to false;  ▷ When the previous state is 20, this variable is set to true (after sending the message)
4:     Set a flag that activates the reading of buffer length to true;
5:     Go to the next state;
6: else
7:     if If a timeout timer output is activated then
8:         Deactivate timer;
9:         Send error type through output variable;
10:        Go to error state 200;
11:    else
12:        Start timer
13:    end if
14: end if
```

On the next one (*22*), it checks if the card buffer has something, updates the message length and the received string with the buffer data and goes to state *23*. If the card does not receive data, eventually the output of the receive timeout timer is activated and it goes to the error state.

Algorithm 2 State 22 of the receive loop which updates the string with the new characters.

```
1: State 22:
2: if Serial bus terminal was able to read the buffer characters then
3:     Stop reading the number of characters;  ▷ This operates the flag in our IF condition above
4:     if The buffer size is different than zero then
5:         Update a variable that holds the length of the receive string with the length of the buffer;
6:         Copy the contents in the buffer onto the end of the receiving string;
7:         Update a received length variable saved on the bus terminal with the buffer count;
8:         Activate the receive buffer bus terminal flag;
9:         Go to the next state;
10:    else
11:        if If a timeout timer output is activated then
12:            Deactivate timer;
13:            Send error type through output variable;
14:            Go to error state 200;
15:        else
16:            Start timer
17:        end if
18:    end if
19: end if
```

The following state waits until the buffer has new data to be parsed into our string and clears a control byte that appears after each receive cycle.

Algorithm 3 State 23 of the receive loop which waits until new data ia available.

```
1: State 23:
2: if The received buffer flag is true then                ▷ The next data cycle is received
3:     Clear the last byte of the received string;         ▷ After each cycle a control byte is removed
4:     Set the receive buffer flag to false;
5:     Go to the next state;
6: end if
```

The last state in our loop checks the last byte of the string and compares it with our control byte, if they match the message is received (control byte is '0x03', in ASCII $03) and the state machine goes to state *30*.

Algorithm 4 State *24* of the receive loop which either ends the communication or restarts the loop to receive new data.

```
1: State 24:
2: if Check the 3 characters of the received message are equal to our control characters to end
   the communication then                                  ▷ The last characters should always be $03
3:     Copy our received string to our output message string;
4:     Set out output boolean as true;
5:     Reset all the different timeout timers;
6:     Go to state 30;
7: else                                                    ▷ The message is not yet fully received
8:     Set the output boolean as false;
9:     Go to state 21;
10: end if
```

On state *30*, it waits until the handler acknowledges the exit output variable and deactivates the enable flag, in which the state machine jumps to state *-1*, like shown in Fig. 4.4.

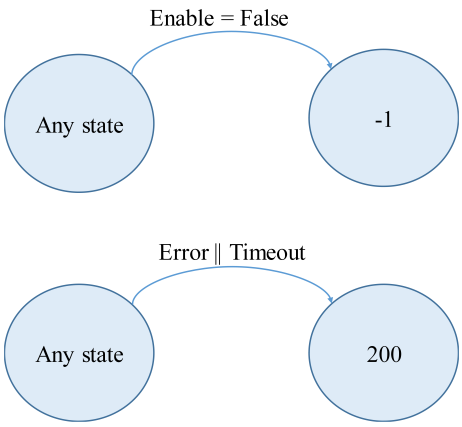


Figure 4.4: Error and reset serial communication state machines

The state machines above are mainly operated by the higher level function block, which is always looking at the exit variables and operating accordingly (checking if the communication is done, retrieving the message and disabling or enabling the function block and handling all the faults that occur in the serial communication).

On the table below we can check every type of error possible for this communication.

Table 4.1: Serial comm fault error codes.

Fault Error	Integer	Description
FLT_ERR_00001	-1	Error building send message
FLT_ERR_00002	-2	Serial Send not ready
FLT_ERR_00003	-3	Error sending serial data
FLT_ERR_00004	-4	Error receive serial data not ready
FLT_ERR_00005	-5	Error receiving serial data
FLT_ERR_00006	-6	Error receiving serial data
FLT_ERR_00007	-7	Error parsing receiving data
FLT_ERR_00008	-8	Error receive timeout exceeded

4.1.2 RFID Serial Handler

This function block is called by a service program created in the main PLC code that is called in every run cycle, this way the main code can alter the inputs and watch the output variables of this function block, which are the following.

- SendMsg: Integer that selects which message to send (reset, read or write);
- RScomm: Pointer to the bus terminal serial array;
- Timeout_retry: Integer that indicates how many times a message should be retried before it fails;
- Write_message: String which will be parsed and written onto the tag;
- Data: Output string that returns the read data;
- SerDone: Output boolean that indicates when the communication is done.

The way this FB is called in the main code sequence is by creating a program in ST with the same inputs and outputs has the RFID serial handler FB, and in that program calling that FB and giving it those inputs and outputs. Now by changing those program variables, we are effectively operating the FB. This is the standard way other FBs are conducted in the code.

This FB contains five different functions:

1. Reset - Send a reset command to the RF240R reader, which acts as an error acknowledgement and also as an initiator so the reader can receive other commands;
2. Read - Send a reset followed by a read command that reads the first 50 bytes of memory from the tag, starting in the address 0x0000;
3. Write - Resets the reader and then writes whichever message the Write_message string has onto the tag, starting in the first memory position;
4. Read without reset - Same as function 2 but without the reset;
5. Write without reset - Same as function 3 but without the reset.

The state machine that operates the entire library can be seen in Fig 4.5.

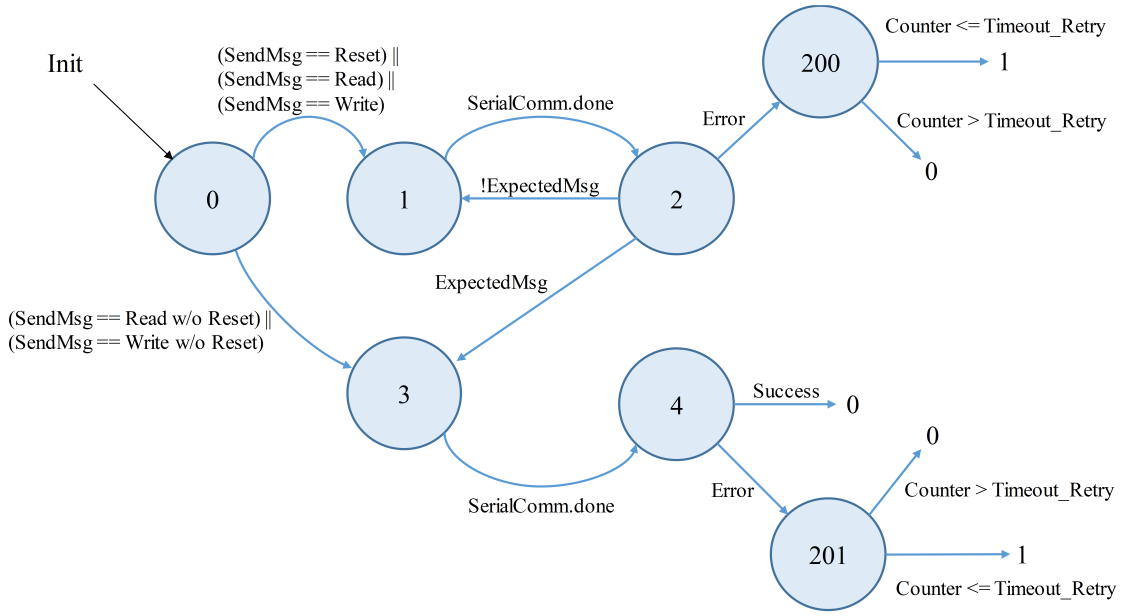


Figure 4.5: RFID serial handling state machine

When there is no operation taking place, the rest state is *0*, in which it is waiting for the main code to send a command that indicates the function to be executed. Depending on the function the state machine can evolve into state *1* or *3*. State *1* sends a reset command and when the *SerialComm* FB signals the *"done"* boolean, it advances to state *2* where it checks the received message. If the message is not expected for that function (bad reset) it goes back to state *1*, if no message was received but it triggered an error (e.g. timeout error, reader damaged), go to state *200* which is an error state, and finally if the message received is as expected, advance to state *3*.

State *3* is analogous to state *1*, but instead of sending a reset command, it sends whichever function the user requested (read/write without reset jump from state *0* to this one, bypassing states *1-2*). State *4* analyses the received message, if it is an error it goes to state *201* otherwise it succeeds, waits for an acknowledge and goes to state *0*.

The error states *200-201* differ in terms of what error triggered them but both use a retry flag (timeout) that is set by the user and indicates how many times the command is sent before it triggers an error. State *200* and *201* are used to indicate an error finding the RFID reader and the RFID tag, respectively.

4.1.2.1 Siemens specific RS232 ASCII protocol

The protocol (see [1]) is a simple communications procedure for a point-to-point connection, the default parameters of the serial interface are as follows:

- Data frame with 8 data bits;
- 1 stop bit;
- Odd parity;
- Transmission speed: 19.2 kBaud.

Full duplex data transfer is possible since no handshake is defined, allowing for the passage of information in both directions at the same time.

The net data is transferred with the ASCII characters ’0’ to ’9’ and ’A’ to ’F’, with two uniquely recognizable control characters at the start and end of the frame ("STX [02]" and "ETX [03]", respectively). With this, a frame is uniquely identified by the control characters for the start and end, with no other security mechanisms or frame length information, the amount of data to be transferred is then restricted by the monitoring time and by the length byte in the frame.

In the eventual case that the host takes longer than 300 ms to send the message, the reader will ignore the current frame. This is the monitoring time of the RF240R reader.

There are no data consistency or line monitoring mechanisms in this ASCII protocol. These must be taken into account by the developed application.

The ASCII table with the corresponding hexadecimal values is the following (Fig. 4.2).

Table 4.2: ASCII table with hexadecimal values

ASCII Value	’0’	’1’	’2’	’3’	’4’	’5’	’6’	’7’	’8’	’9’	’A’	’B’	’C’	’D’	’E’	’F’
HEX Value	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46

In order to write the net byte value of "0x34", which is mapped by the ASCII values "3" (0x33) and "4" (0x34), we need to write "0x33 0x34".

The data transfer must always meet the specification that in each byte, first the high nibble is transferred and then the low nibble. Also only one command can ever be active no the reader, meaning that a command is always followed by the corresponding acknowledgement that must arrive before new command can be sent, with the exception of the "RESET" and "SLG-STATUS" commands. This is why the library was designed with many reset states to handle errors and achieve more redundancy.

4.1.2.2 Tag Write

Starting with the writing procedure, we want to write the product code (23-25 character string, e.g. "123456789101112131415161P") into the tag. In order to be able to write any character (0-9 and a-Z) and code it to the tag, we cannot simply convert the bytes directly and write them because we will lose the ability to write characters above ’F’ (0x46). As such, we need to transform, for example, an ’8’ into its HEX equivalent, ’0x38’ and then write ’0x33’ and ’0x38’ to code the 3 and 8 effectively.

The writing algorithm can be explained in the steps below.

1. For every character in the string do the following;
2. Transform that character in its byte form;
3. Calculate the remainder of that byte by 16;
4. Convert to ASCII (sum 48 or 55 depending on number value in base 10);
5. Save that number in an array and divide the number by 16 (removing nibbles from the initial number);

6. Repeat steps 3-5 until the division returns 0;
7. Invert the numbers inserted in the array above (e.g. [1;2] to [2;1]);
8. Create an array with the correct write message headers for the RFID and concatenate with the array created in step 5;
9. Complete the array with '0x5F' (underscore character) in every memory position until it reaches the max length defined for the net data and add the footer;
10. Send this array and its length to the *SerialComm* FB to be sent.

An example of treating a string can be seen below in Fig. 4.6.

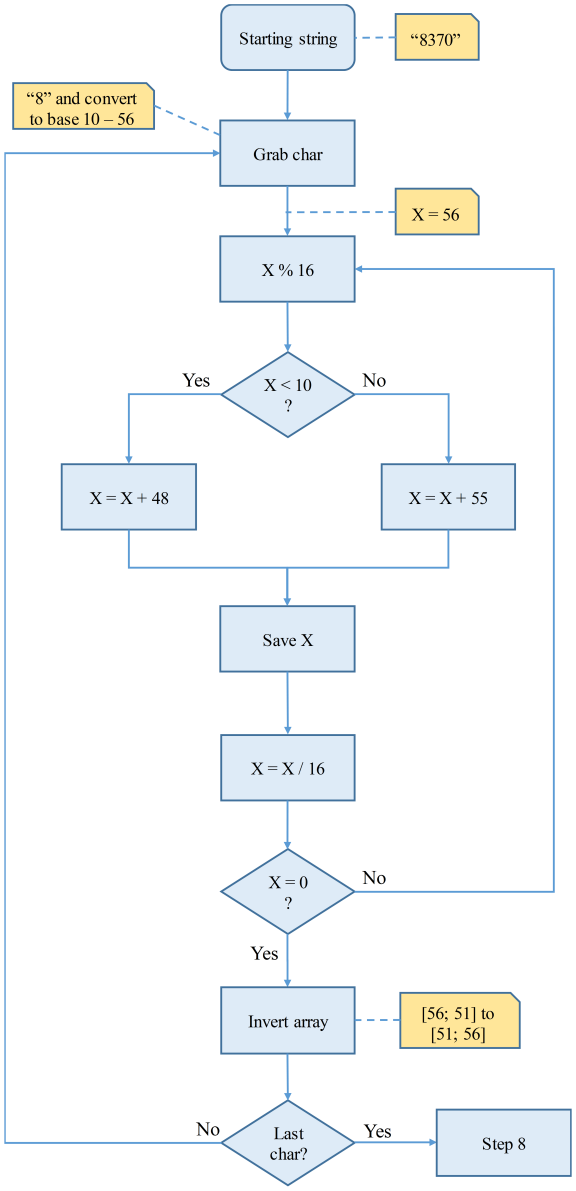


Figure 4.6: Writing a string chart.

By having an array with [51;56] in decimal, in hexadecimal it translates to [33;38] which maps with the ASCII protocol above into [3;8]. By joining both numbers, we get 38 which is 8 in ASCII.

This algorithm is needed to write capitalized characters that the product code can have.

The steps in the enumeration above successfully create an array that has all the appropriate write headers and footers, which can be seen below (table 4.3) as well as the command that is sent (excluding data).

Table 4.3: Write command frame designation.

Address	Designation	Value	Description
Byte 0	Frame Length	0x32	Net data length of frame minus 1 byte (depends on data length)
Byte 1	Frame Length	0x44	Net data length of frame minus 1 byte (depends on data length)
Byte 2	Command	0x30	Write command
Byte 3	Command	0x31	Write command
Byte 4	Status	0x30	Fixed status
Byte 5	Status	0x30	Fixed status
Byte 6	User data address	0x30	Address of the user data to be written
Byte 7	User data address	0x30	Address of the user data to be written
Byte 8	User data address	0x30	Address of the user data to be written
Byte 9	User data address	0x30	Address of the user data to be written
Byte 10	User data length	0x32	Length of user data to be written
Byte 11	User data length	0x38	Length of user data to be written
Byte 12	User data	Lowest byte, High nibble	Data
Byte 13	User data	Lowest byte, Low nibble	Data
Byte n-1	User data	Highest byte, High nibble	Data
Byte n	User data	Highest byte, Low nibble	Data

On the final steps, the array is written with underscores ('0x5F') until it reaches the max length for the net data, this was made in order to simplify the reading mechanism, which needs the data length to be read. Because the product code can vary between a few lengths, we opted to write always 50 bytes and populate the spaces after the product code with underscores. This way on the reading procedure, we can always read 50 bytes from the tag, and grab the message until it reaches an underscore.

4.1.2.3 Tag Read

The reading algorithm is, in its essence, the inverse procedure of the writing steps, as follows:

1. Detect the first underscore on the message and remove from there on;
2. Copy the received string into an array of bytes;
3. For every byte either subtract 48, 55 or 87 if the byte value is between 48 and 57, 65 and 70 and 97 and 102, respectively;
4. After the subtraction, join in another array numbers two by two from the original array, being the left nibble the four least significant bits from the first number and the right nibble the four least significant bits from the second number.

5. Convert the new array into a string.

An example of the reading procedure can be seen in Fig. 4.7 for the string "38".

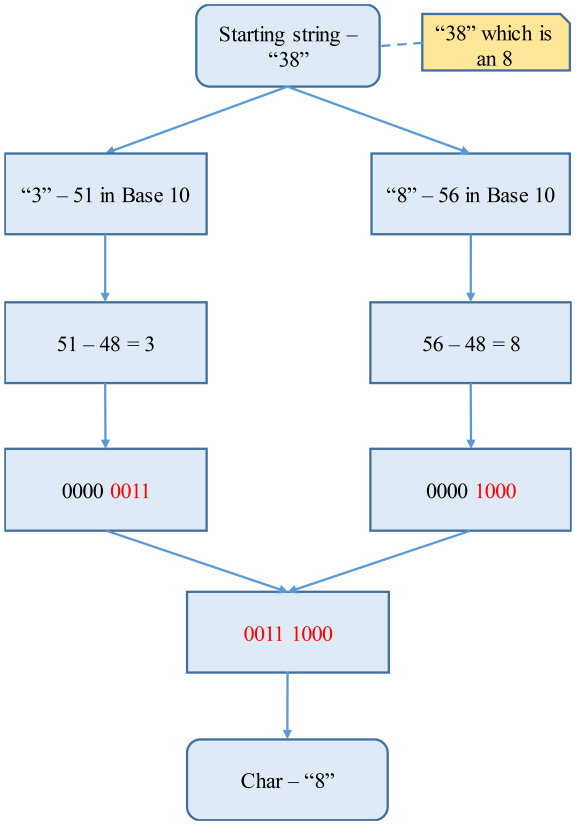


Figure 4.7: Reading and parsing characters from the tag memory.

This writing/reading procedure is the standard for storing data in RFID tags, there being many devices available on the market that read tags and do this exact reading procedure. Such a device was used to validate the writing algorithm.

4.1.3 Library validation and verification

On the first library versions and to test the ASCII commands, the reader was connected directly to the PC via USB (with a RS232 to USB adapter) and commands were sent to the reader with the help of a serial port terminal. For example, executing a RESET command consisted of sending the following message, "$020A000000050200000010001$03", the same way the library does.

The validation was made using the Nexeed Control training module, which already had a serial communication card (KL6001) and a few extra accessories, making it perfect for testing before deploying the code on the production line.

Once the code was incorporated and ready to be tested on the training module, the RFID reader was not responding to any command that was sent by the PLC. Even though the message sent was exactly the same. The reader was acknowledging the message sent by the PC and not the PLC.

After testing other things (i.e. changing the RS232 baudrate), the PC to reader communication was sniffed using an oscilloscope, as well as the PLC to reader, with the intent to compare the messages bit by bit to find any deviations. Below we can see both messages (Fig. 4.8 and 4.9).

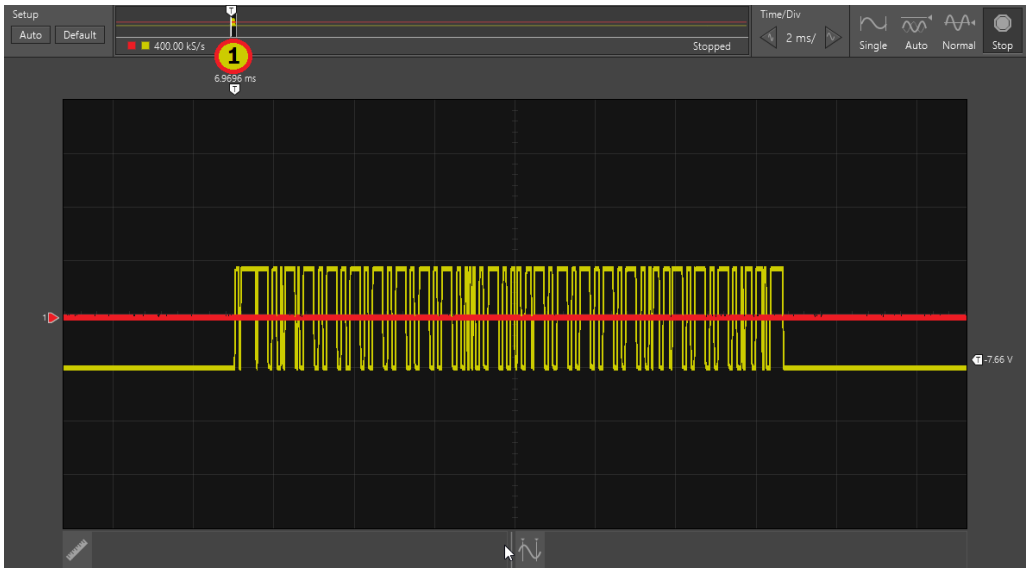


Figure 4.8: PC to RF240R RS232 communication message.



Figure 4.9: PLC to RF240R RS232 communication message with the KL6001.

Comparing both messages, we can verify that the PLC message is sent in five different bursts instead of a single one (see the top of figure 4.9). These bursts are the size of the bus terminal data transmit buffer, which is 16 bytes, and from the beginning of the message until the last burst, there is a difference bigger than 300ms, that being the communication timeout for the RFID reader.

In order to solve this, a bus terminal with a bigger internal buffer was tested (KL6031 with 128 bytes) with the results being the following (Fig. 4.10).

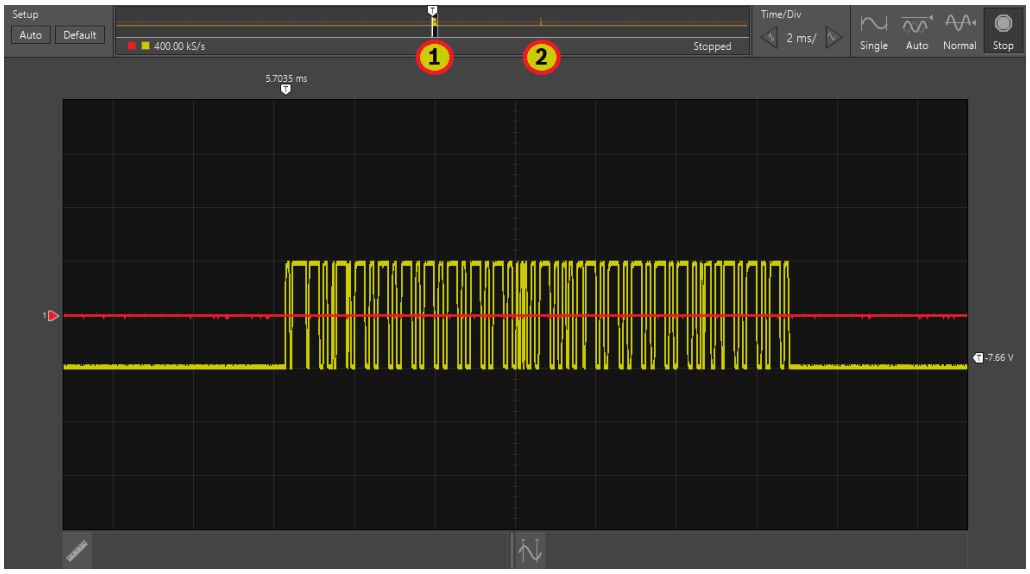


Figure 4.10: PLC to RF240R RS232 communication message with the KL6031.

Even with more internal buffer size, the message was still segmented in two bursts, resulting in it being discarded by the reader. This solution, even if it had worked, is not very robust because if the command we are trying to send was bigger than the buffer, it would not work.

Therefore, the solution is to increase the transmission rate of the field network Profibus-DP to 6 MBits/s. Although it still fragments the message in various different segments, it transmits then fast enough that the 300 ms timeout is not a concern any more.

In conclusion, all the different library functions were tested and validated on the training module, such as writing and reading tags, reset commands, errors in case of no tags found or reader communication failures and how it could affect the production line.

4.2 Physical Alterations

For the selected RFID reader, the read ranges are up to 65mm depending on the tag. Two types were tested, the MDS D423 which has a maximum range of 4cm paired with this reader and the smaller MDS 424 which has a read range of 6cm with the RF240R. Both tags are very similar in technical specifications, they only differ in physical characteristics (size and mounting hole). The higher range tag was selected, because having the higher read range is very beneficial in our application, since the expected distance between reader and tag is 4cm, this means that if we intend to do fly-by readings, we want to keep the tag in the antenna range as long as possible (figure 4.11).

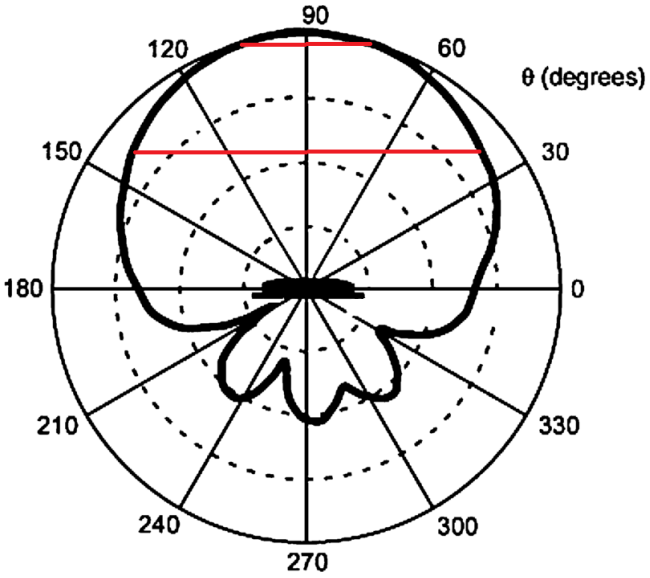


Figure 4.11: Generic RFID radiation pattern [10].

Taking a look at figure 4.11, to increase our read chances, we want to keep the tag in the antenna range as long as possible. The best place to do that is by cutting the range where it is wider, that is why the MDS 424 tag was chose, because it offers a wider range, meaning it stays in the antenna radiation range longer.

Nevertheless, we now know we have a few centimetres to work with and we need to find a way to place both tag and reader to accurately and reliably save and read the product code. For this, we took advantage of the fact that the trolley moves in tracks and some stations have stoppers.

Taking a look at the trolley and stopper, they can be viewed in the figures below (Fig. 4.12 and 4.13).



Figure 4.12: Trolley base.

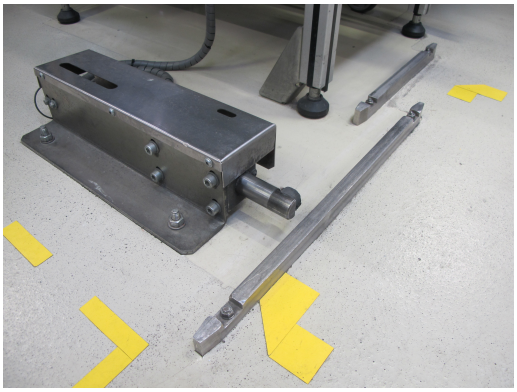


Figure 4.13: Station stopper.

When pushed against the stopper, on its regular working position, an optical sensor right beside the stopper cylinder detects the trolley base. This coincides with the forward most leg, we can verify this below (Fig. 4.14).

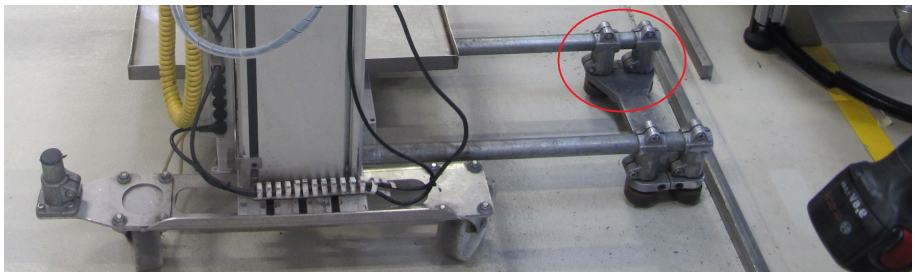


Figure 4.14: Trolley base that rests against the stopper.

With this, the optimal placements for the reader and the tag is to attach the tag on that highlighted part of the trolley (figure 4.12), facing outwards towards the stopper and placing the reader above the optical sensor, so it faces directly the tag. This setup allows for both devices to be only 3-4cm apart. For better understanding of the tag-reader positions, we can check the figures 4.15 and 4.16.

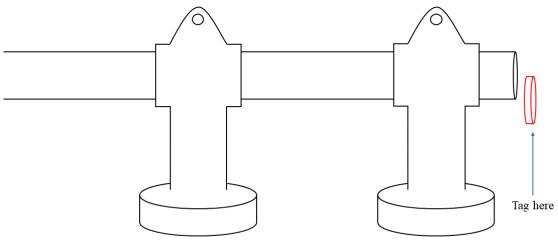


Figure 4.15: Trolley tag positioning.

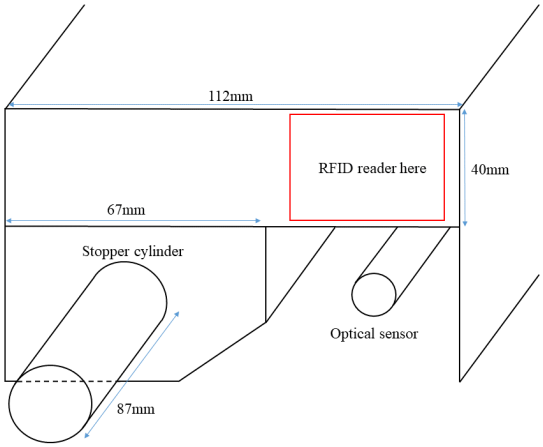


Figure 4.16: RFID reader positioning.

The cylinders at the base of the trolley are hollow and face the *"to-be"* position of the reader directly like shown in figure 4.17. This way, tag and reader are only 4cm apart in the worst case scenario.

The distance between the face of the reader and the place of the tag is going to be roughly 48mm, as the stopper placement slightly varies from station to station.

Therefore, we need to take into account the slightly lower tag position relative to the trolley hollow cylinder and the tag mounting plate must not be large enough that it hits the stopper or any other line aluminium profile, sliding without problems in the predefined tracks.

Figure 4.17 joins both 4.16 and 4.15 views and shows what the system is supposed to be once implemented, in terms of clearances.

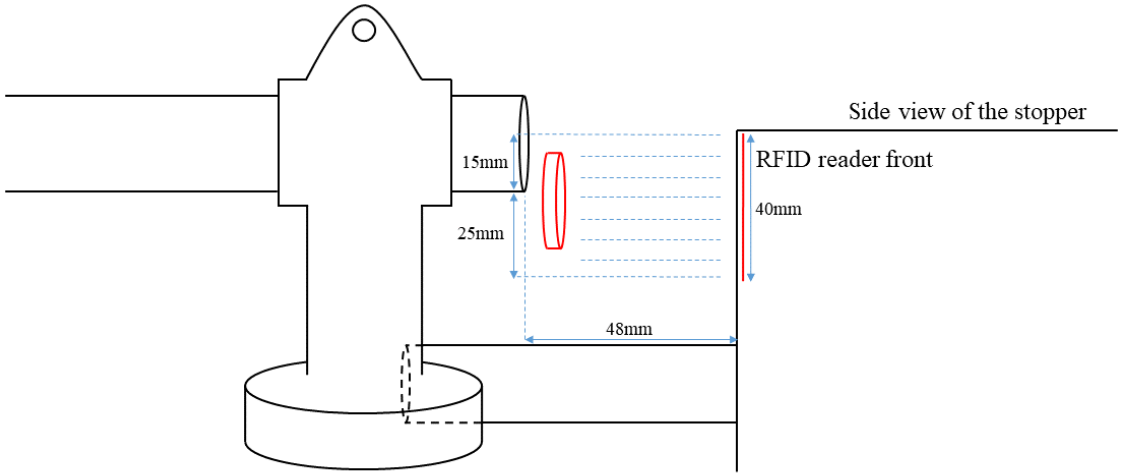


Figure 4.17: Ideal tag and reader positioning.

4.2.1 Trolley Tag Support

For the prototypes, we used a 3D printer to test and validate the concept. The first iteration featured a semi-oval shape on the end to align the tag with the reader and a cylinder long enough to be inserted and secured on the trolley base. The tag mounted to the prototype using a mounting cover (that is why figure 4.18 and 4.19 have two holes on either side of the support).

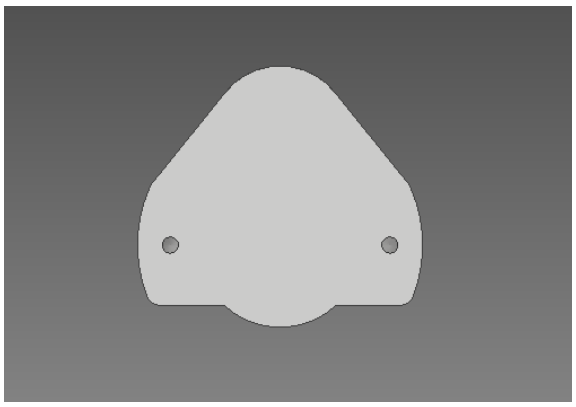


Figure 4.18: First iteration tag support.

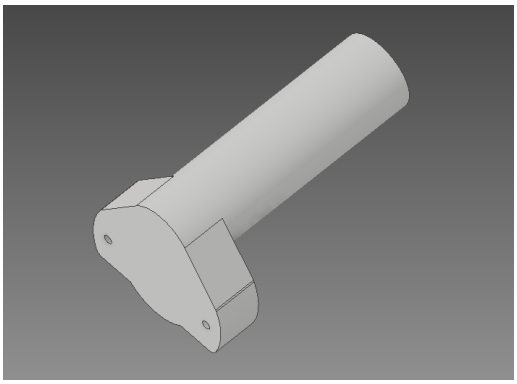


Figure 4.19: First iteration tag support.

This concept worked as intended, but the support could not be made of PLA material, as it would not robust enough to handle the line conditions and the screw threads. So it was decided that the support should be made of aluminium, but this brings another concern. Radio-Frequency noise, metal surfaces, liquids a few other conditions all block radio performance, which can lower our effective range and render the system useless [32]. To combat this, in addition to the aluminium support, a 1cm PLA spacer between the tag and the metal was created.

Furthermore, to facilitate the machining work, the cylinder block and the oval shape metal frames were separated and the latter was changed from aluminium to steel and cut by laser, while

the cylinder is made by a lathe. The spacer and mounting covers are still made by 3D printing, to not interfere with the RFID.

The following figures (Fig. 4.20 to 4.27) show the final support, disassembled and assembled, as well as the technical drawings (full drawings in the appendix).

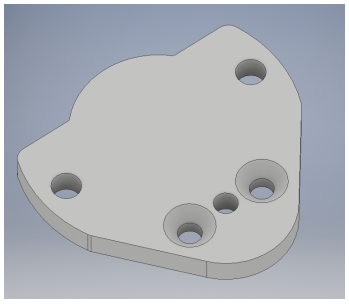


Figure 4.20: Steel frame.

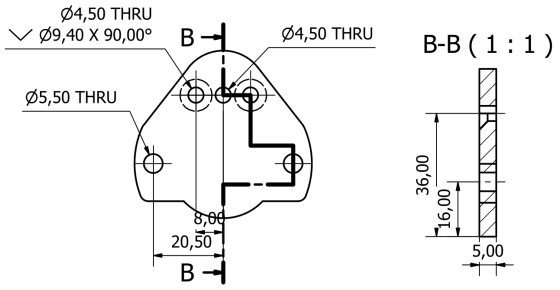


Figure 4.21: Steel frame technical drawing.

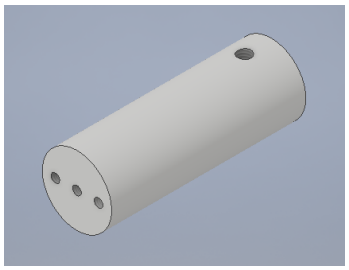


Figure 4.22: Aluminium cylinder.

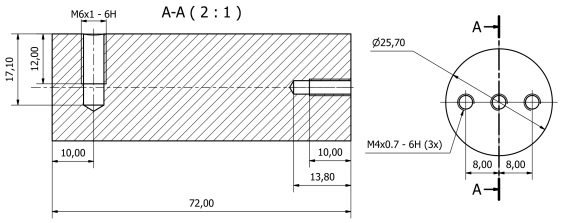


Figure 4.23: Aluminium cylinder technical drawing.

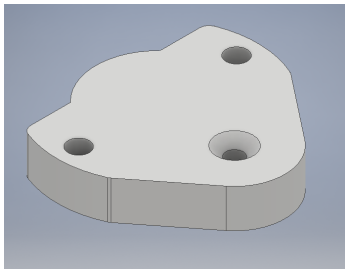


Figure 4.24: Plastic spacer.

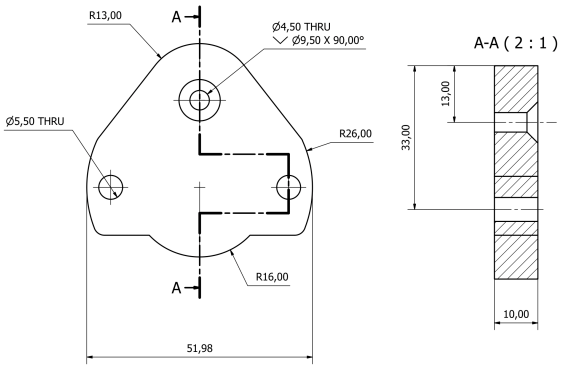


Figure 4.25: Plastic spacer technical drawing.

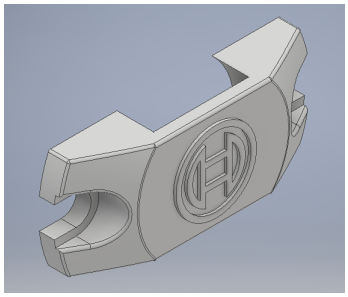


Figure 4.26: Mounting plate.

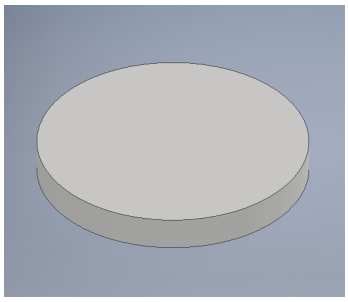


Figure 4.27: Wafer to space the tag.

The assembled tag support can be seen below (Fig. 4.28).

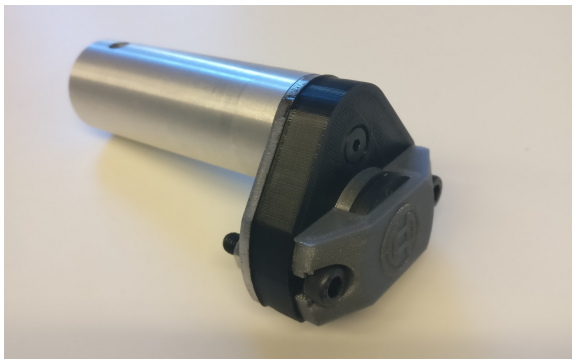


Figure 4.28: Final tag support.

On the trolleys, a small hole was drilled and the support was mounted on every line trolley (20 in total are with this solution, Fig. 4.29).



Figure 4.29: Final trolley implementation.

4.2.2 RFID Reader Support

Following the concept in section 4.2, in order to mount the reader the stopper was disassembled and a metal plate was added to support the optical sensor and the RFID reader, the stopper needed to have its sealing raised and shifted 1cm to the right so the reader could stay inside the casing and a 4 by 4 cm plate was cut in order for the reader to sit aligned with the stopper's front.

The final stopper can be seen below (Fig. 4.30).

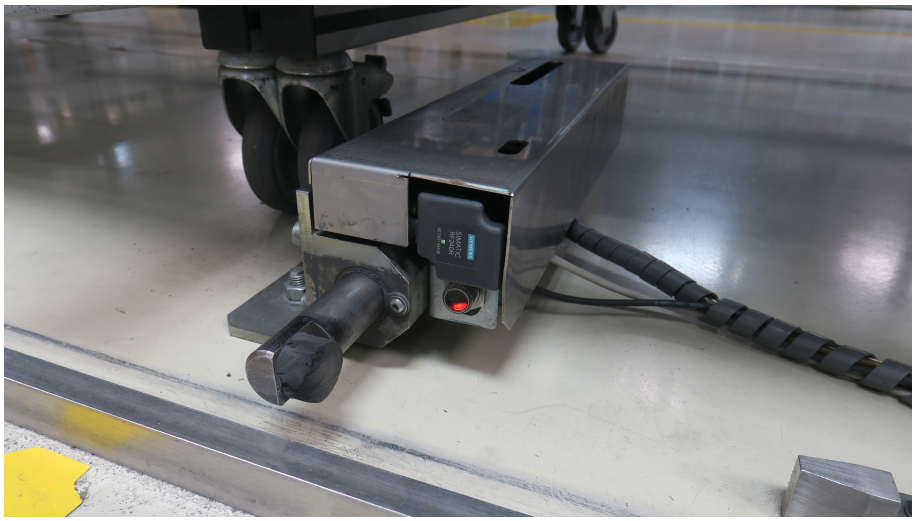


Figure 4.30: Final stopper implementation.

4.2.3 RFID Support on stations without stoppers

In the line there are only 9 stations that have stoppers implemented, the other 8 do not. In order to prove the concept that it is possible to implement a robust system that can detect the product code even if the trolley travels with some speed, one station (station 12) is going to have a RFID reader with an optical sensor that can read the product in a trolley fly-by.

For this, we placed the sensor and reader in the same height as they are in the stopper, but the optical sensor further back. This was made so that we can send the read command to the reader before the tag is face to face with it, so that when the tag reaches the reader range, it can send the product code straight away to the PLC, gaining maximum trolley speed.

The following figure (Fig. 4.31) shows the setup created for the fly-by read, like stated before.



Figure 4.31: Station 12 RFID reader and optical sensor on their fly-by read positions.

4.3 Station code alterations

To implement the proposed solution on the line, we must take into account the three different purposed stations.

1. Product write stations - Stations where the product is only going to be written into the tag memory.
2. Product read stations - Stations where the product is only going to be read from tag memory.
3. Product clear station - Stations where the product is going to be read from the tag memory, followed by a write to clean the memory.

The code between the different purposed differs greatly, but in the same scope, it is almost exactly the same, depending on the station.

By importing the created RFID library and creating a program that calls the RFID handler FB every run cycle like referenced in section 4.1.2, the station code is ready to work with the RFID device.

The following enumeration (steps) explains the general order of operations that takes place in every station:

1. Initializations;
2. Start scanner and wait for manual read;
3. Check barcode type (there are two types, normal products and KanBan codes, which change the type of product being produced);
4. Send request parameters NTC for message P1 (recipe server);
5. Wait for server response;

6. Send request parameters NTC for message P4;
7. Wait for server response;
8. Verify NTC sequence;
9. Station specific process (this is where the product alterations and process take place);
10. Send MES parameters;
11. Wait for MES response;
12. Wait for trolley;
13. Unblock stopper;
14. Block stopper and repeat the process.

4.3.1 Write stations

In the line where the project at hand is being implemented, only one station is a write station (station 10), where the product code is printed and manually associated with the physical product.

Our implementation goal is to create a step between 9 and 10 that effectively writes the product code on the tag memory and validates it, with redundancy methods to avoid system failure and assure the production does not stop.

The implemented procedure follows Bosch GmbH industry standards, such as using a return value variable in every transition with four types of values, OK, NOK, JP and RUNNING, that symbolize, respectively, good to proceed, step failed, jump to another step and step is currently running, so no transition has *RetVal = RUNNING*, it is just an internal state to avoid it to jump. Another standard and good practice is the use of steps for different actions, if we need to, for example, wait for a sensor and do something afterwards, those are two separate steps because they have different purposes and scopes.

The following SFC chart shows the added steps and transitions, which will be explained below (Fig. 4.32).

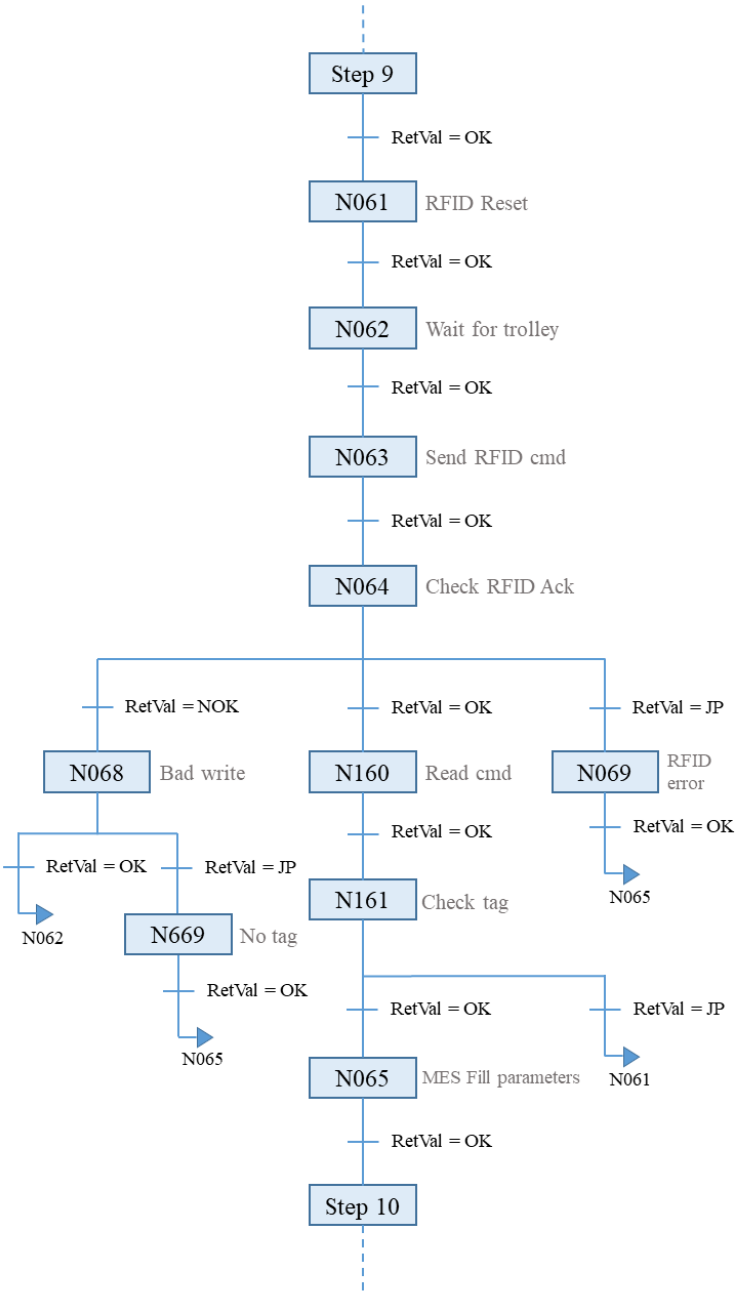


Figure 4.32: Implemented SFC write code.

After the process is completed (step 9) and before we send the status to the MES, we write the code to the tag and assure that it is there, for this we immediately begin by sending a reset command (N061) and wait for the trolley to arrive at the stopper (it should be there already, unless the operator processes a new product without a trolley, in which he needs one to proceed, so we wait in step N062 for that presence). Step N063 sends the write command and step N064 checks the RFID library response, which can lead to 3 different steps.

If the RFID library response is *'No_reader'*, that means that there is no response from the RFID reader and it goes to step N069, which alerts the operator with a message and prompts

him to manually click the HMI to advance the process. This is a very important step because it means that the tag was not written, which could lead it to be read with another product code in the following stations and that can have serious issues with already packaged products (more on this in the risk assessment on chapter 6 section 6.4). After the operator manually acknowledges the error, it jumps to state N065.

Another library response is *'Cmd_error'* which means that no tag was found within the time-out, this sends the program to step N068, where a RFID reset is performed and there is a counter of how many times this error was triggered. If the counter is less than three, it jumps to step N062 to restart the write process, otherwise it sends a message to the HMI that requires an operator response, stating that no tag was found. This counter exists because the sensor can be triggered by things other than the correct trolley base (like the trolley wheels), so it resends the write command up to three times, after that we assume that the tag can be broken or another thing is blocking the sensor and it jumps to step N065.

Finally, the last library response is *'Success'* which means that the tag was written. It then goes to step N160 where it waits for the trolley and sends a read command, to verify and assure that the correct product code was written. N161 checks the response and if the read product code equals the written one, it proceeds to N065, if not it restarts the process on step N061.

One of the most important steps is N064, in which we access the state of the RFID reader and act accordingly. Figure 4.33 shows a flow chart of the actions step N064 takes.

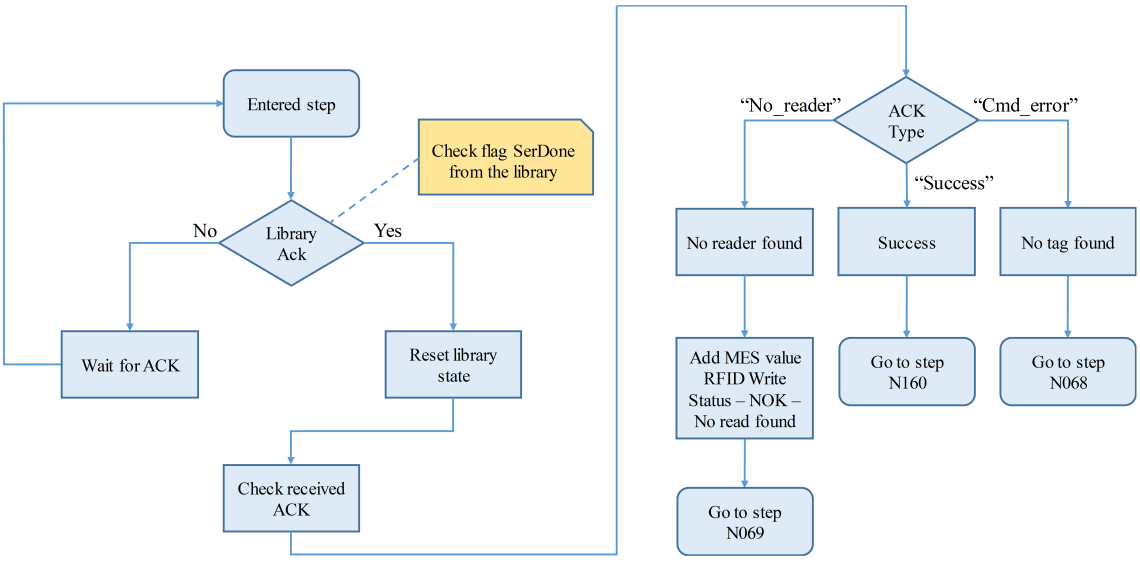


Figure 4.33: Flow chart of step N064 code for station 10.

Step N065 is the one that adds the MES parameters (see section 4.3.5).

4.3.2 Read stations

The read stations are the majority (all the stations without stoppers and 7 of the 9 with stoppers), where the current order of operations is stated above, but instead of adding steps between

9 and 10, similar to the write stations, the steps where the manual barcode executes the read are edited, namely step N041, to add the automatic reading procedure and some are added to send data to the MES.

The added and edited code can be observed in the figure 4.34.

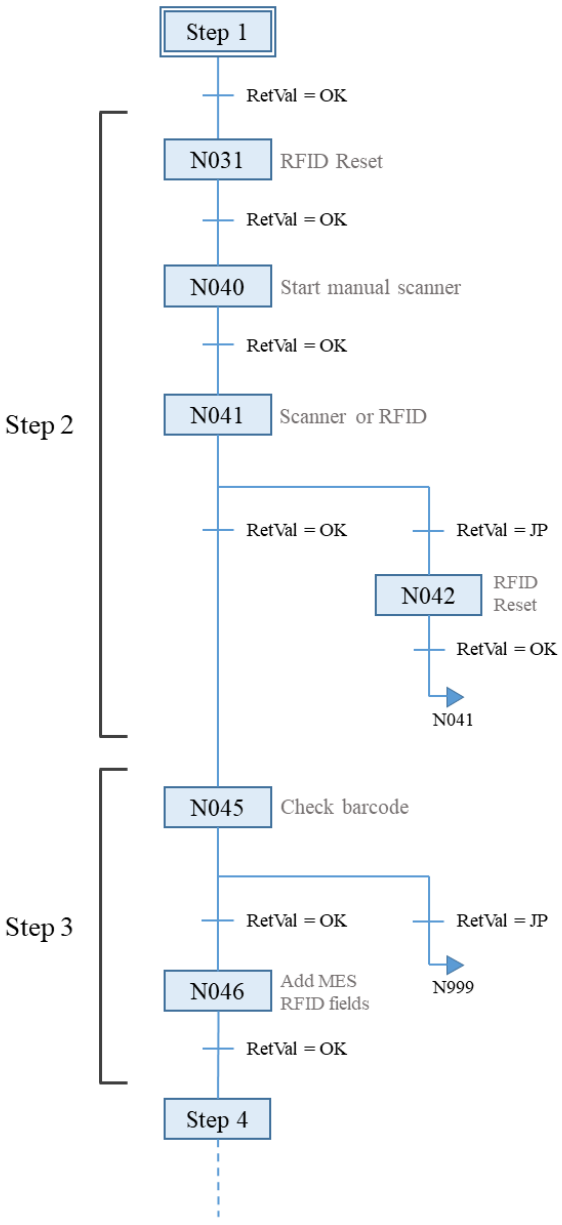


Figure 4.34: Implemented SFC read code.

Step N031 was added and sends the RFID a reset command. Step 40 starts the manual scanner and no changes were made, followed by step N041, where it waits for the barcode to be read. This is the most important step in the read operation, where the code stays, waiting for a *ReadDataOk*

from the manual barcode FB. Parallel to that waiting condition, if the optical sensor is tripped and a flag activating the reader is active, the read command is sent and then the response analysed.

If the response is *'No_reader'* then it deactivates the RFID flag so that only the manual barcode is active and it alerts the operator. Furthermore, if the response is *'Cmd_Error'* it means no tag was found and it jumps to step N042 where it resets the reader and jumps back to N041 to read again. Finally the last response type is the barcode itself, where it is then verified in the next step, N045. Here a verification function had to be created similar to the one that already exists, because that one has barcode specific structure verification, which we do not have, so we needed to create one that has the same functional product code verifications. If it fails, it jumps to the final step on the sequence and restarts the whole process.

Finally on step N046, the information relative to the RFID read status is prepared to be sent to the MES.

4.3.2.1 Check station

In order to add robustness, a check station that compares the tag code with the product code was added in the middle of the line. Station 41 requires the operator to manually read the barcode, and it advances if the tag code is equal to the barcode, otherwise it alerts the operator that they do not match. Also if the trolley has no tag or the tag is broken, after the first RFID read retry, a button is added to advance the process if the barcode has been read manually.

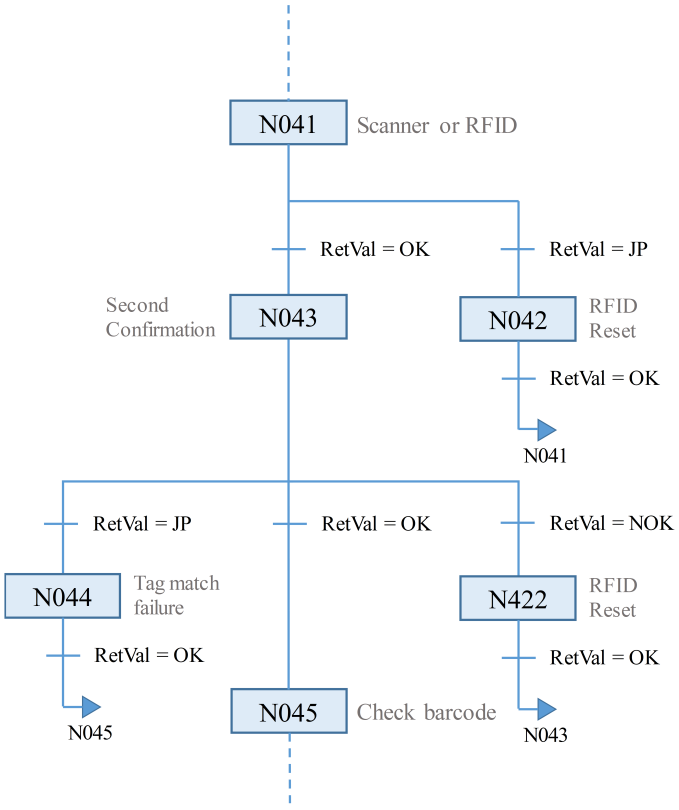


Figure 4.35: Station 41 SFC to check if the tags are matched.

Figure 4.35 shows the SFC code of this station, on step N041, either the code is read through the barcode or the RFID, then on step N043, the other method must be used. If the RFID cannot read the tag (e.g. no tag), it jumps to step N422 where a button appears that allows the operator to bypass this process and read the barcode directly. Furthermore, if the codes do not match, it goes to step N044 where a message to the operator is displayed and that the manual identification will be used.

We chose to add this check in order to mitigate risk 1.3 (see section 6.4), in detriment of losing the gains of that station, which are less important.

4.3.3 Clear stations

Clear stations join both systems, they are effectively a read station for the products but also act as a tag memory cleaner because they are the last read station in the line. This means that step 2 and 3 are edited with the read code explained above, and between step 9 and 10 we add the write procedure but instead of writing the product code, we write 25 zeros, this way if a tag is mistakenly not written on station 10, it will not cause problems on the following read stations.

4.3.4 Time analysis

In the proposed solution, one major analysis point is the study of the different times it takes the current order of operations to take place. Namely the process, exit and change time.

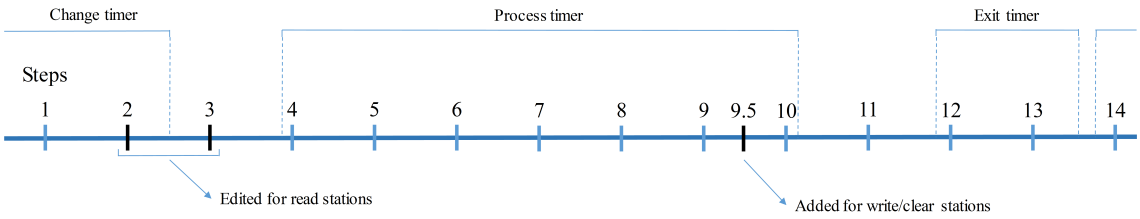


Figure 4.36: Time analysis for the different steps.

These timers were implemented at the same time the stations were edited with the RFID code and the results are sent with the RFID status to the MES for further analysis in step 10.

4.3.5 MES parameters insertion

The PLC sends an XML file with the structure shown in section 3.1. It is composed by a set of structures with the names and values of all the variables that are part of that section. In our case we want to insert the data into the *'Test Info'* section, so we add the name of the variable and its value to the test info structure and update its size.

Depending on the type of station (write, read or clear) we add a variable named *'RFID Write Status'* or *'RFID Read Status'* with the values *'OK'*, *'NOK - No Tag Found'*, *'NOK - No Reader'* or *'NOK - Manual Read'* for read/clear stations after we validate the RFID status. That means that we edit the test info structure after the read takes place or after the write is successful.

Just before step 10 (*'Send MES parameters'*) we added a step that fills the test info structure with three mores values, the process, exit and change timers in seconds (see listing 4.1), where the process is the current product processing time, the change timer is the time it took the current trolley to arrive at the present station from the last trolley that left. And finally the exit time is the time of the last trolley.

An example of the read status MES insertion is as follows:

Listing 4.1: RFID read status MES insertion in the station code.

```
(*e Send MES values *)
(*p Enviar valores para MES *)

(* OpCon MES RFID Values *)
(* RFID Value 1 - Result *)
Stat012TestInfo [ iCounterStat012TestInfo + 1].name := 'RFID Read Status';

CASE RFID_MES OF
  (* NO RFID *)
  0:
    Stat012TestInfo [ iCounterStat012TestInfo + 1].value := 'NOK - Manual read';

  (* No reader *)
  1:
    Stat012TestInfo [ iCounterStat012TestInfo + 1].value := 'NOK - No reader found';

  (* No tag *)
  2:
    Stat012TestInfo [ iCounterStat012TestInfo + 1].value := 'NOK - No tag found';

  (* Success *)
  3:
    Stat012TestInfo [ iCounterStat012TestInfo + 1].value := 'OK';
END_CASE

iCounterStat012TestInfo := iCounterStat012TestInfo + 1;
RetVal := OK;
```

Stat012TestInfo is a structure (from station 12) with the test information results for that station (which can be seen on the XML file 3.1), with variable size and can have customizable information. The process to insert the timers differs because it is not saved in the test information but instead in the test results, which has more parameters for different types of results (see listing 4.2).

Listing 4.2: MES timers insertion on the station code.

```
(* RESULTS *)
(* Result 1 - Process Time *)
```

```
Stat011TestResults [ iCounterStat011TestResults + 1].name := 'Process Time';
Stat011TestResults [ iCounterStat011TestResults + 1].value := ProcessTimerTemp;
Stat011TestResults [ iCounterStat011TestResults + 1].upLim := 0;
Stat011TestResults [ iCounterStat011TestResults + 1].loLim := 0;
Stat011TestResults [ iCounterStat011TestResults + 1].setValue := 0; (* no predefined processing
    time value *)
Stat011TestResults [ iCounterStat011TestResults + 1].checkType := 3; (* no check upper or lower
    limit check *)
Stat011TestResults [ iCounterStat011TestResults + 1].unit := 's';
Stat011TestResults [ iCounterStat011TestResults + 1].result := 1; (* OK *)
```

4.4 Deployed station code

The station code deployment schedule can be seen in table 4.4, where "Week 24" refers to 2019.

Table 4.4: Stations implementation.

Time Frame	Stations
Pilot stations	10, 12
Week 24	34, 33, 40, 42, 43, 50, 60, 81
Long-term	51, 61, 70, 71

The stations with the time frame of "Long-term" will not be altered to the RFID system during this dissertation because the time it would take to design a custom support for each station would be unaffordable, but they are still on track to be changed afterwards. All the stations that will suffer alterations are highlighted below (the ones that do not, are hidden behind the respective squares).

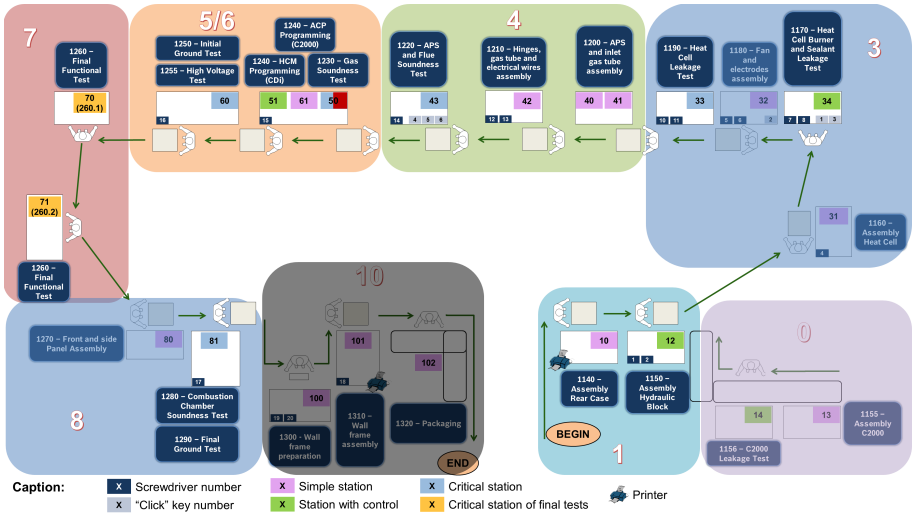


Figure 4.37: Highlighted stations which will be altered.

On June 12th, we have implemented station 10 (write station), 12 (read station) and 81 (clear station). All the code is ready to be deployed, but due to the line being in production on the early and afternoon shifts, it is only possible to test and deploy the code late at night.

The code, being deployed in a function block library, is highly scalable and can be used in other lines (there is currently another line being constructed near line 7 which will produce the same products and have the same setup, so naturally this system is being considered for deployment there). This was also a goal of the project at hand.

On June 18th, every station up-to the long term ones are working with the RFID system, this represents a major milestone to the scope of this dissertation. After implementing the system, some unexpected gains (time wise) manifested on station 60. Particularly, when the product is identified at station 60, a protective door has to slowly close in order to perform a dielectric strength test, which starts immediately after the product is identified. Automatically closing the door as soon as the product arrives stands as a big time improvement.

The non-stopper stations are a long-term goal because these require more studying (both of the station and how the operator manages the trolley, which can be off standard) for mounting positions of the RFID reader. Nonetheless, station 12 was chosen for this exact reason, to provide a proof of concept that it is possible to achieve high automatic identification percentages even in these cases (which will be verified in chapter 6).

4.5 Summary

In this chapter we addressed the entire RFID implementation process, from the architecture and design of the function block libraries to the physical changes and additions. More concretely, in this chapter:

- We designed two function blocks which are a part of our RFID library according to the protocol set by Siemens AG, more specifically:
  1. A lower level RS232 handler for receiving and sending data through a serial bus terminal;
  2. A higher level handler that parses the messages, either to convert the write message into something the reader understands or to parse the received message into a readable string;
- We created, by 3D printing at first and then by machining aluminium and steel, a support to hold the RFID tag safely in the trolley base;
- We altered the stoppers (and stations without stoppers) to accommodate the new hardware (RF240R);
- We took the current stations code and altered in such a way that it left the manual barcode functionality, as well as the RFID one and added MES parameters for traceability of the RFID status, as well as different time measurements.
- Finally, the current project implementation status was analysed.

With this, we fulfilled the two primary objectives, automatic and instantaneous identification for the product and an association system for the trolley and product. Additionally we accomplished the first secondary objective, a monitoring system for the different partial times inside the time cycle of each station. All of this while complying with the requirements set by Bosch GmbH, which state that the trolley must be identifiable by the MES, the current barcode reading should be replaced or kept as backup and scalability for the proposed solution must be ensured.

Additionally, we hit the project implementation milestone to have all the stations with stoppers (and one without) with the RFID system deployed on the line.

Chapter 5

Product Tracking Web Based Platform

This chapter aims to introduce and explain the web based platform that tracks all the products in the line. It will have access to the product data by having direct access to the MES database.

5.1 ASP.NET MVC Framework

The chosen framework is an open-source server-side web application designed for web development and it was developed by Microsoft Corporation. It allows programmers to write ASP.NET with any supported .NET language and it supports a various array of extensions to achieved the desired outcomes.

The MVC is an architectural pattern, and it stands for *'Model-View-Controller'* with the following diagram of interactions (Fig. 5.1).

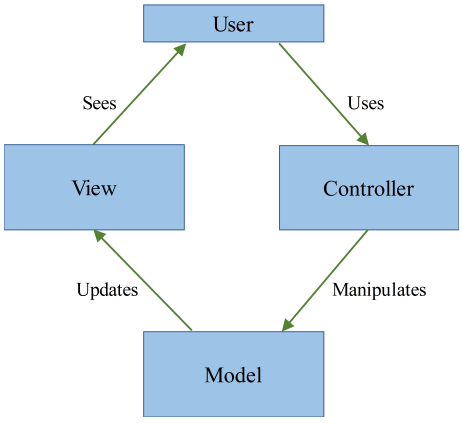


Figure 5.1: Model-View-Controller diagram of interactions.

The controller is the central component which handles all the business logic of the application, such as retrieving data based on user input and is written in C#. It manipulates the model, which is the application's dynamic data structure.

The view is the representation to the viewer, written in HTML, CSS with bootstrap and razor (from the server-side).

In our use case, we are going to use JavaScript (using some JQuery, Ajax requests and JSON) to handle the real-time operations.

This application can be accessed anywhere, as long as the Bosch MES database is accessible, and that is one of the main advantages of having chosen this route to implement the tracking interface.

5.1.1 Architecture

When the user inserts the website URL on a browser, it will prompt the controller for the appropriate route, create the model and display the view to the user.

Now in the client-side, a JavaScript routine runs every 10 seconds and makes an Ajax request to another server-side controller (*OracleController*). This request runs a function that queries the database about the latest results by location on the MES. It then returns the data serialized as JSON and a display algorithm is ran to show the products in their respective stations.

The architecture overview is the following (Fig. 5.2).

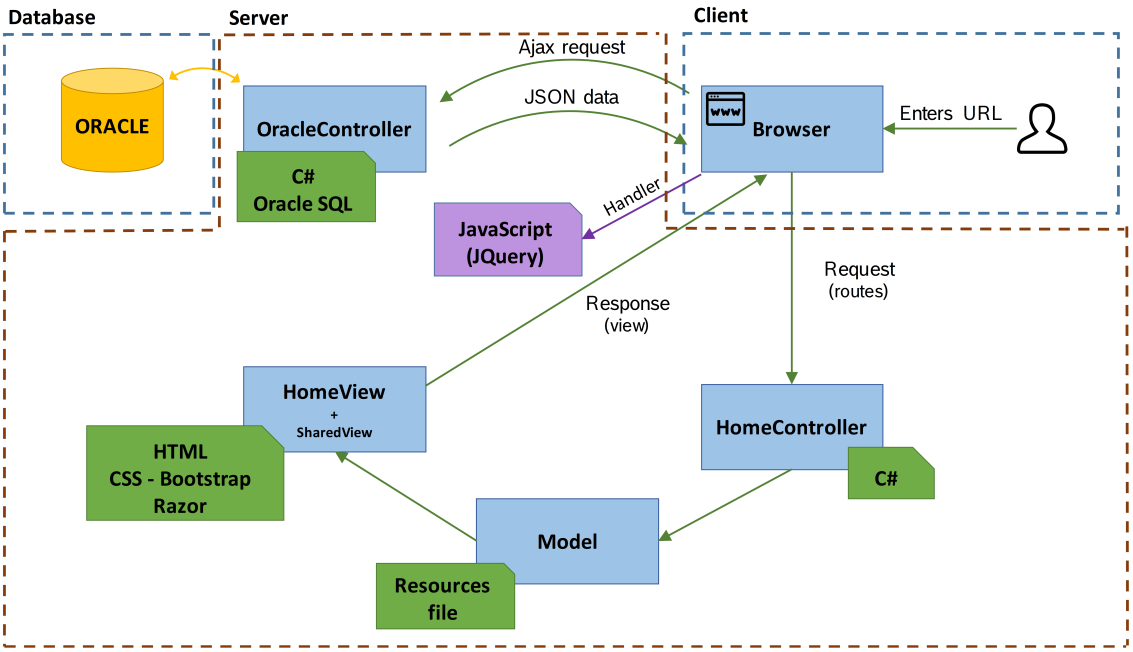


Figure 5.2: Web platform architecture diagram.

5.2 Web platform

The state of the web platform is shown below (Fig. 5.3). The right section of the page has a map of the line, where it is possible to see all the different markers on the stations and these correspond to their real position. On the left there are two different tables, one of the last available shift data, with the beginning timestamp, production number target, parts processed, delta (target minus processed), number of OK parts and NOK products that were re-worked.

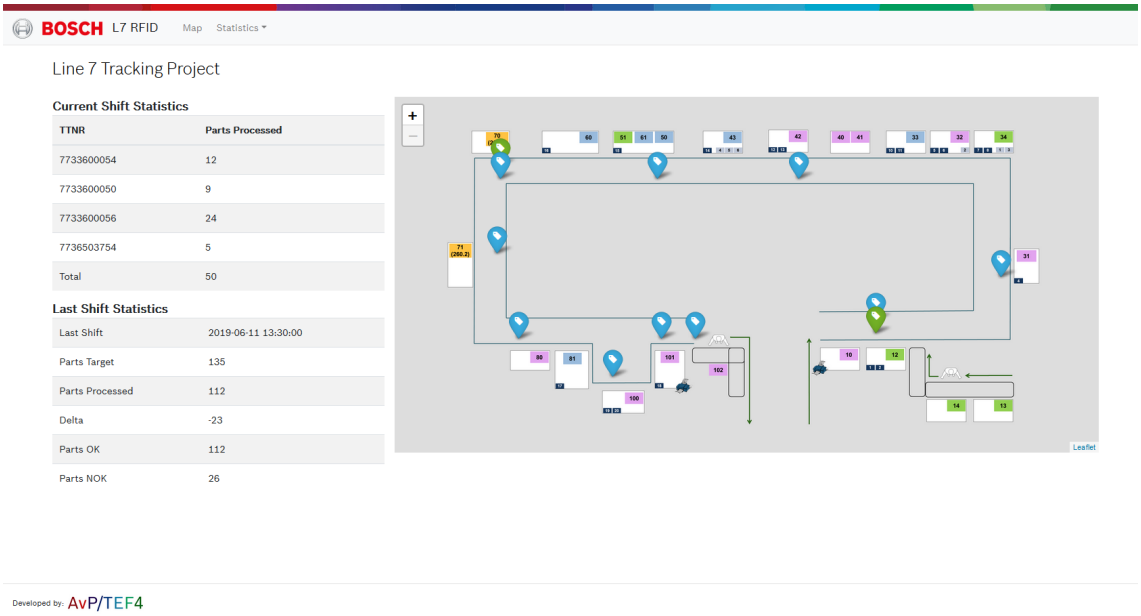


Figure 5.3: Current state of the web platform.

When the connection to the server is dropped, a message alerting the user automatically appears, and it disappears once it is established again. All this happens automatically and does not need user input, this is because this web application is oriented to be used in a screen turned on 24/7 to monitor the line, without any need of user intervention (see figure 5.4).

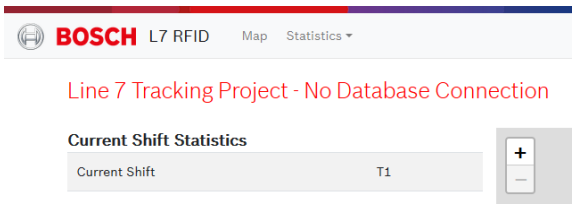


Figure 5.4: Web platform database connection error.

5.2.1 Station map

For the map representation (Fig. 5.5), an open source JavaScript library for interactive maps was used, Leaflet, as well as a plug-in for multicoloured markers. A simplified layout of the one

showed in chapter 3 was made to simplify the identification of where each individual marker is at all times.

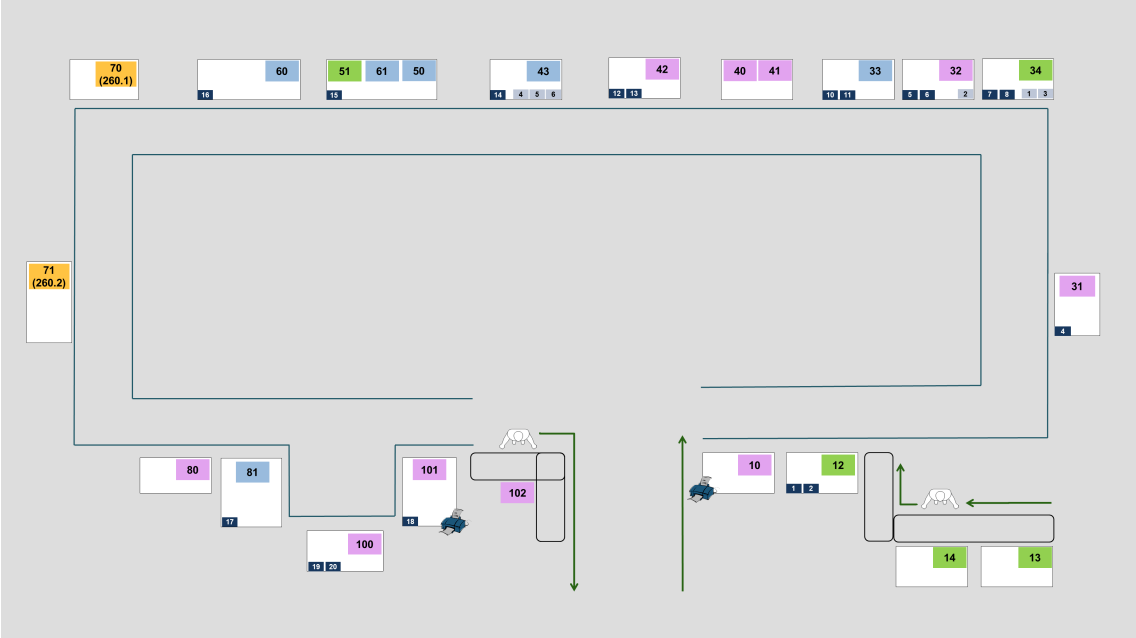


Figure 5.5: Base line layout for map integration.

At each station there are five different markers that indicate the amount of trolleys that are travelling (explained below in section 5.2.1.1) or at that specific station (see figure 5.6).

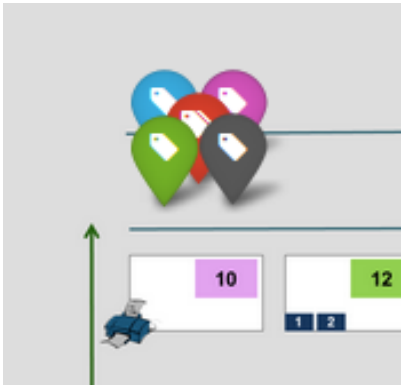


Figure 5.6: Every marker at a station.

The markers appear in order, first the blue then the green, purple and grey, this way there can be up to four trolleys at that station. Finally the red indicates a bottleneck. Hovering the mouse above the markers brings them to the front for a more user friendly interface and when the red marker appears, the others disappear to clearly indicate the bottleneck situation.

All the markers for all the stations are programmed in a matrix, where the rows are all the different stations and the columns the different coloured markers, with position 0 being the red

marker, and blue, green, purple and grey the rest, respectively. The following matrix (equation 5.1) shows the JavaScript configuration for the markers.

$$\begin{bmatrix} 10_0 & 10_1 & 10_2 & 10_3 & 10_4 \\ 12_0 & 12_1 & 12_2 & 12_3 & 12_4 \\ 31_0 & 31_1 & 31_2 & 31_3 & 31_4 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 102_0 & 102_1 & 102_2 & 102_3 & 102_4 \end{bmatrix} \tag{5.1}$$

Every marker is also clickable, it has information about the product that it holds, such as the product identification number, the last station results and the time it left the last station. At the request of the client, more information can be added to this pop-up (see figure 5.7).

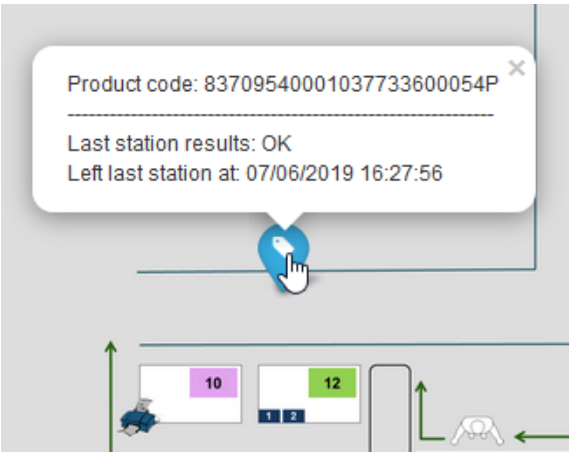


Figure 5.7: Marker text pop-up.

5.2.1.1 Following algorithm

The NEXEED MES has a vast variety of different databases and tables that hold all sorts of information regarding the whole shop floor and much more. Regardless, there is a table that holds the information sent by all the different line PLCs at the end of each product cycle. That table is inside the *QUAL* database and is called *LOCATION_RESULTS*, which as the name implies, has the station results for every location. Its rows are in table 5.1:

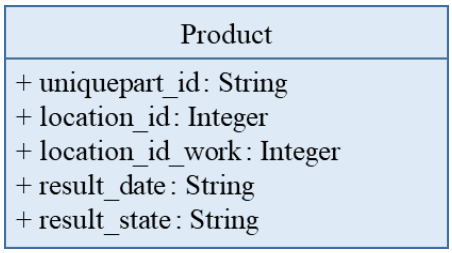


Figure 5.8: Object created to map each unique product.

Table 5.1: *LOCATION_RESULTS* table.

LOCATION_RESULTS
LOCATION_RESULT_UID
UNIQUEPART_ID
LOCATION_ID
RESULT_DATE
PROCESS_NUMBER
PSTATINTERVAL
WORKCYCLE_COUNTER
RESULT_STATE
PART_ATTRIBUTE
TYPE_NUMBER
TYPE_VARIANT
TYPE_VERSION
MACHINE_ID
SHIFT_CODE
ARCHIVE_FLAG
BATCH
DELETE_DATE
TEST_TYPE
TIME_STAMP
V_RESULT_DATE

The relevant information retrieved is the *UNIQUEPART_ID* (it has the product code), the *RESULT_DATE* (XML telegram timestamp), *RESULT_STATE* (OK vs NOK) and most importantly the *LOCATION_ID* which has the location the XML telegram was sent from.

With this, the process that runs every 10 seconds is as follows. By requesting data from the server by the client, the server queries this database for the last 20 entries in the table *LOCATION_RESULTS*, with these entries a list of objects is created and each object (see figure 5.2.1.1) corresponds to a specific product (which has the product code, station location, result state and timestamp, see figure 5.2.1.1). This list is then serialized, converted to JSON and sent to the client, answering his request.

Furthermore, on the client side, there is a control matrix that holds all the product codes at whichever stations they are at. The rows are all the different stations, with the first column being an integer for the station code and the following columns are all the unique products codes, ordered by arrival time. With this, it is not a control matrix but more of a vector of vectors with varying sizes (see equation 5.2).

$$\begin{bmatrix} (10 & 10_0 & 10_1 & \dots & 10_X), \\ (12 & 12_0 & 12_1 & \dots & 12_Y), \\ (31 & 31_0 & 31_1 & \dots & 31_Z), \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (102 & 102_0 & 102_1 & \dots & 102_W) \end{bmatrix} \tag{5.2}$$

The purpose of this vector of vectors is to keep track of which products are at which stations,

for when a new database entry arrives with a product code that already exists somewhere on the vector, we are able to remove that marker.

Therefore, when the client receives that server JSON data, it then parses it into a list and for every object in that list there are two main procedures. Firstly, the client will verify if that product code is already associated with a marker on the map, if it is, it then removes that marker from the map and reorganizes those station markers (to keep the blue, green, purple and grey hierarchy).

Finally, in the last procedure it adds that marker to the map, using the objects location number and the vectors first position to find the right station and marker. It checks how many markers are already deployed at that station and adds one on the end, be it a normal marker or the bottleneck one. An activity diagram with this algorithm, from the moment the client received the serialized JSON data can be see in figure 5.9

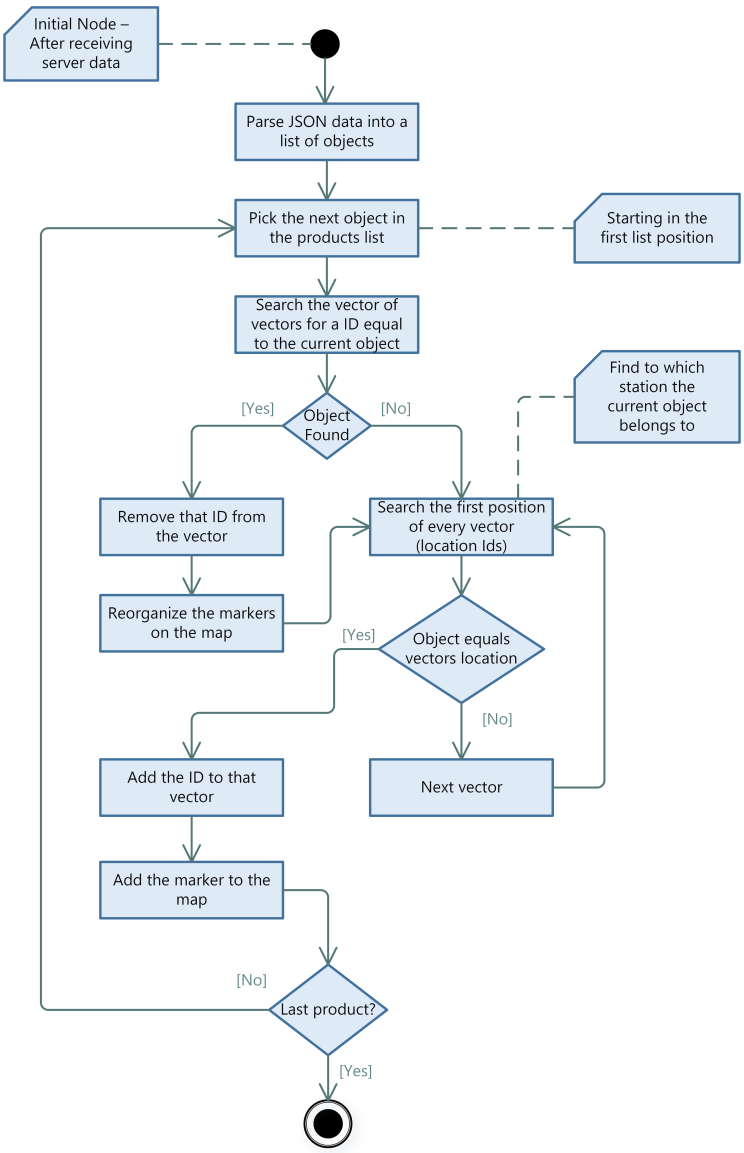
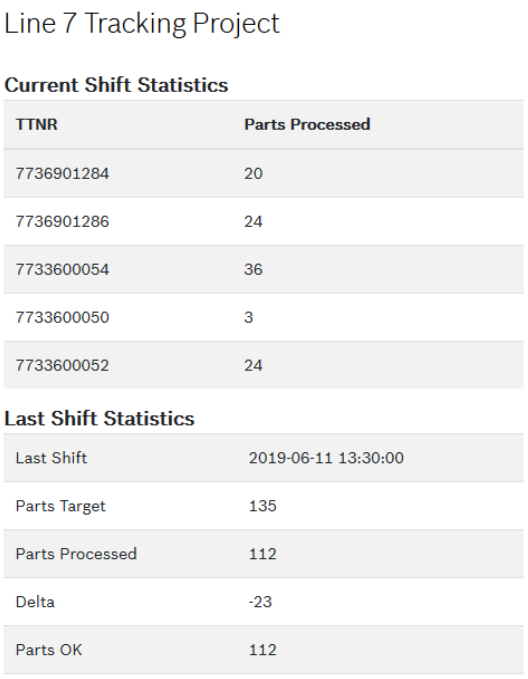


Figure 5.9: Following algorithm activity diagram.

5.2.2 Statistics

The left side of the application has the stats of the current and last shift (see figure 5.10). The last shift statistics are stored in another database, and it is updated once that specific shift ends, with parts target number, parts processed, parts OK and NOK and a few other line stats. This table is also updated through a function that runs every 10 seconds (the query is very lightweight) and updates the shift data. It was made this way so that the application can be open 24/7 with accurate data.


Line 7 Tracking Project

Current Shift Statistics

TTNR	Parts Processed
7736901284	20
7736901286	24
7733600054	36
7733600050	3
7733600052	24

Last Shift Statistics

Last Shift	2019-06-11 13:30:00
Parts Target	135
Parts Processed	112
Delta	-23
Parts OK	112

Figure 5.10: Web platform list statistics.

The current shift statistics are not stored in a database, because the production targets, overall equipment efficiency and key performance indicators (KPI) are all calculated on the fly by the system that needs them (e.g. shop floor Andon systems) with data from the actual producing product cycle time, the number of operators currently working, the shift size and the programmed stops.

Luckily, using the data from the *LOCATION_RESULTS* table, we can extract the number of produced parts in real-time for the current shift, organized by product type (TTNR), this data is very useful to keep track of how well the line is working.

5.3 Summary

In this chapter we addressed how the framework ASP.NET MVC works, as well as showing how the application will be devised. We proposed and executed an algorithm to track and follow the trolleys/products using database entries from the XML telegrams coming from all the stations

PLCs. Finally we developed two tables with statistics from the current and last shift to be shown in real-time alongside the line map.

We fulfilled and exceeded the secondary objective of having a front-end system to aid in the visualization of the trolleys/products, by having statistics to the left of the map that track the performance of the current and last shift of that line.

Chapter 6

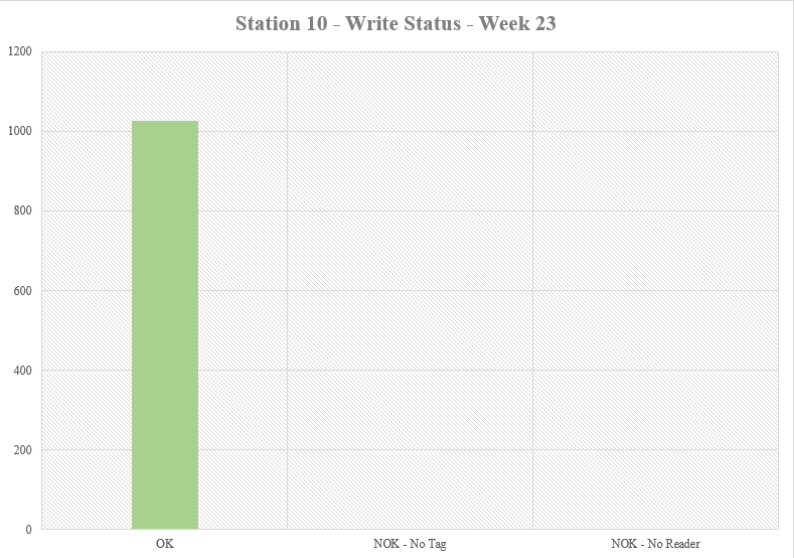
Results and validation

This chapter will approach the station results, such as how well the system is working with different process time analysis. The chapter also presents a risk assessment, because the deployment of new software and hardware on a working production line can be very detrimental if something goes wrong and also show a satisfaction analysis.

More specifically, two stations (10 and 12) are going to be analysed in more detail since they are fitted with the system since May, and the rest of the implemented stations (see section 4.4) will be evaluated as a group, since their final deployment only occurred in the middle of June.

6.1 RFID status results

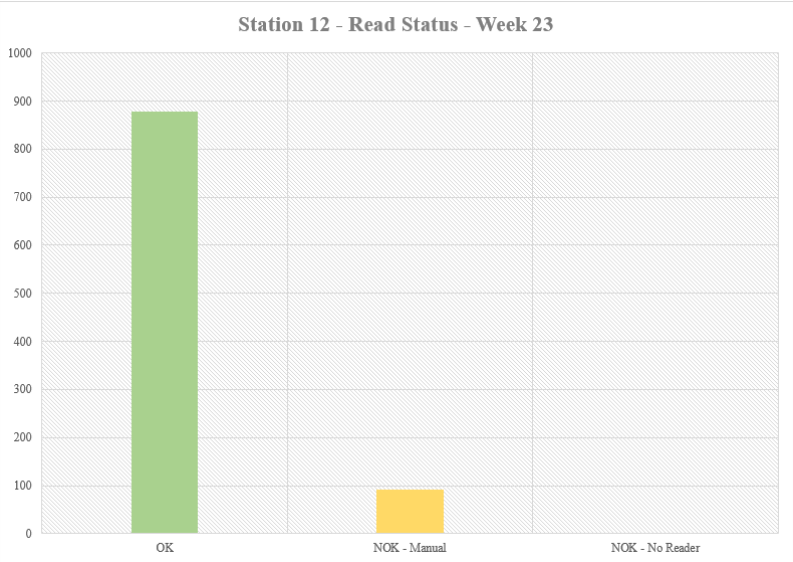
The following figures (see figure 6.1) show the RFID status from stations 10 and 12 on week 23 of 2019 (3 to 7 of June).



OK	1025
NOK - No Tag	0
NOK - No Reader	0
TOTAL	1025
OK Percentage	100%

Figure 6.1: Station 10 write status results from week 23 of 2019.

As we can see, station 10 (write station) achieved an OK rating of 100% with 1025 products worked on, which is a very good result as that means that every trolley that started the process and passed through this station to begin the product construction had its tag written into, and validated through the processed explained in section 4.3.1. Figure 6.2 shows these same results but for the read station 12, without the stopper.



OK	879
NOK - Manual	91
NOK - No Reader	0
TOTAL	970
OK Percentage	90.62%

Figure 6.2: Station 12 write status results from week 23 of 2019.

As expected, these results are not perfect, but very good nonetheless for a station without stopper. After talking with the station operators, most of the NOK occurrences are derived from either other non-related errors that occur in the NTC process and they scan the code again (its easier to scan the barcode than it is to backup the trolley from its working position) or it happens because the RFID scanner is too far forward and they stop the trolley right before it (only in station 12), which can be easily fixed by moving the reader backwards a few centimetres.

As of week 25 of 2019 (June 17th to June 21st), the short-term stations (see section 4.4) all have the implemented system. The next table (table 6.1) will show the RFID status results from the read stations on June 19th.

Table 6.1: RFID read status results from June 19th.

Station	Products processed	OK	NOK - Manual	NOK - No Reader	OK Percentage
33	73*	73	0	0	100%
34	73*	73	0	0	100%
40	184	184	0	0	100%
42	183	183	0	0	100%
43	189	187	2	0	98.94%
50	190	178	12	0	93.68%
60	186	186	0	0	100%
81	216	213	3	0	98.61%

*Lower number of parts processed than the rest of the line because one of the product families made on the line do not stop here.

These results are very satisfactory, we can verify that it is being well received by the operators (this being the third day since it was implemented on these stations) and we expect to reach 100% read status on every station. The small number of products that are not read by RFID can be explained by the trolley working position. Some stations have procedures where the operators do not push the trolley against the stopper, and these are not in compliance to Bosch standards, so one solution can be to move the stopper back a little to facilitate the whole procedure.

The write process to clear the tag on station 81 achieve an OK percentage of 100% with 185 tags cleaned out of 185.

6.2 Time results analysis for stations 10 and 12

On the stations, like stated before in section 4.3.4, we have three different times from the stations order of operations. Starting with the change over time, we can verify the results for station 10 on June 4th and 5th. Note that the spikes that appear on the figures correspond usually to shift breaks, and the times of 0 seconds to the scanning of KanBan codes (for printing more product codes, tied to station 10) or turning off the station and back on (first change over is always zero in this case). Then we will observe the exit times for station 10 and finally the process times for both stations (figures 6.3 and 6.4).

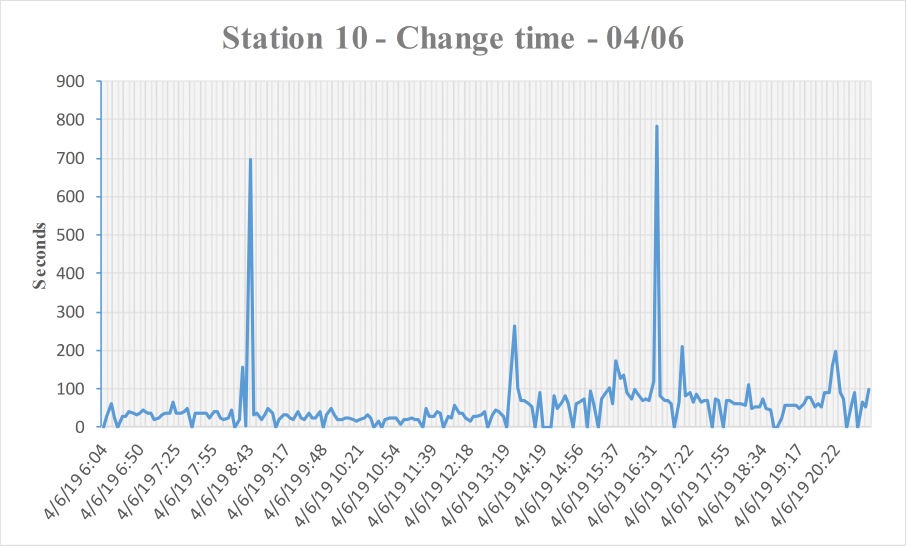


Figure 6.3: Station 10 change times for June 4th.

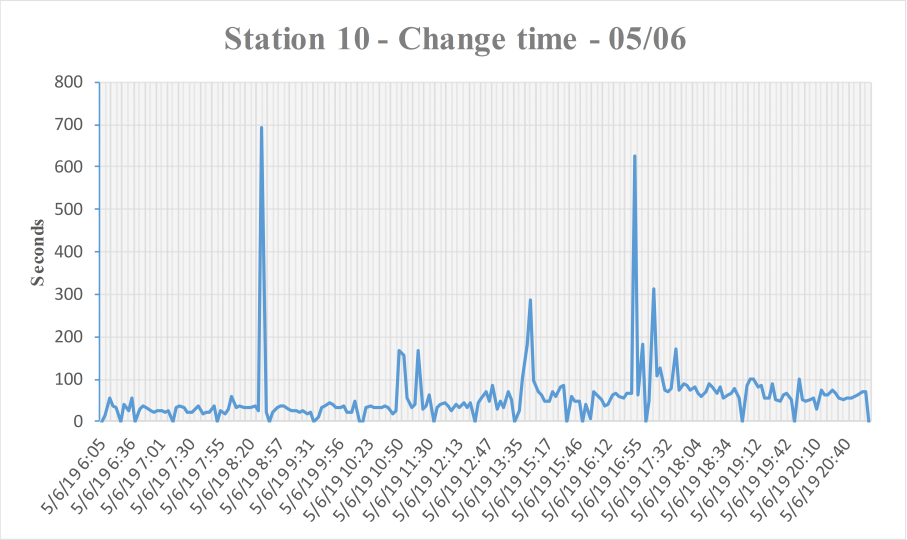


Figure 6.4: Station 10 change times for June 5th.

The average change over time for the whole week without the break spikes is about 50 seconds. For station 12 we have the following change over time graphs (figures 6.5 and 6.6).

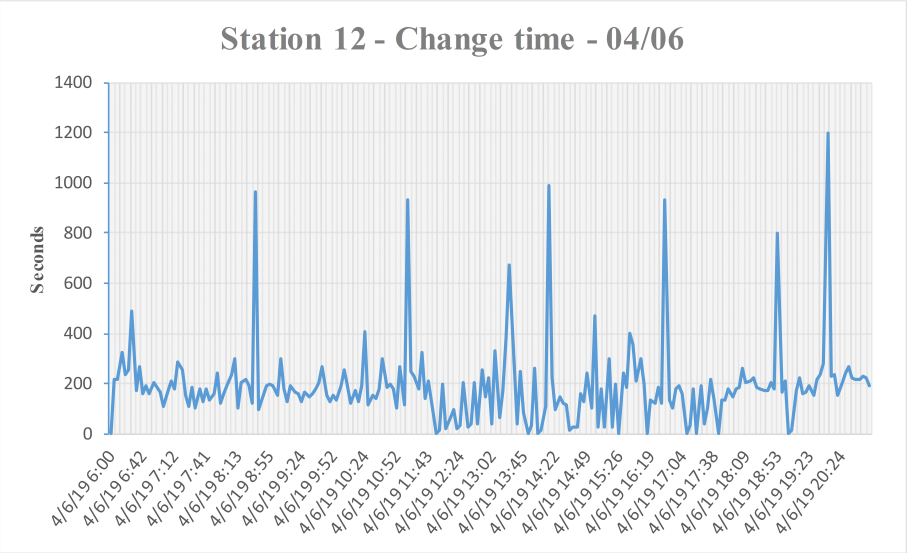


Figure 6.5: Station 12 change times for June 4th.

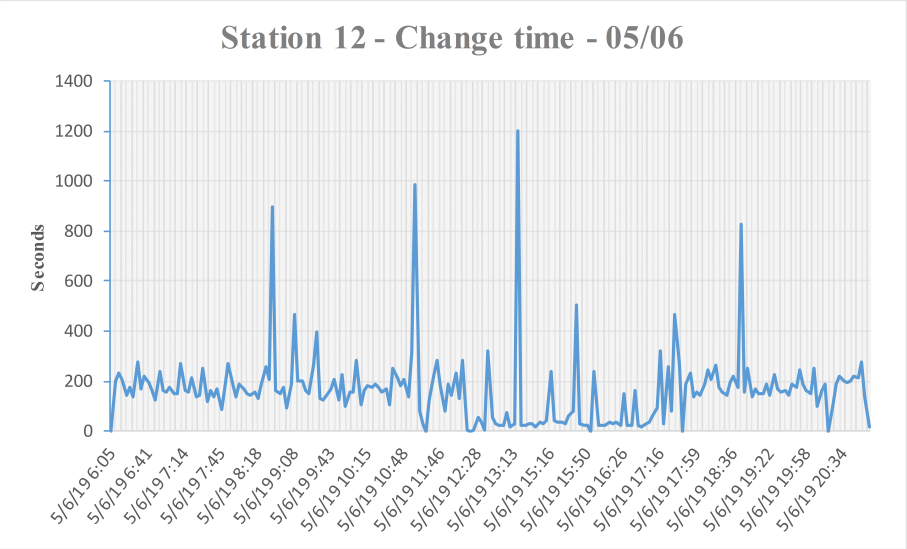


Figure 6.6: Station 12 change times for June 5th.

The average station 12 change over time is around 158 seconds for that same week.

The exit time does not exist on station 12, since it has no stopper, so when the process ends it starts the change over timer. The station 10 exit times for June 5th is the following (figure 6.7).

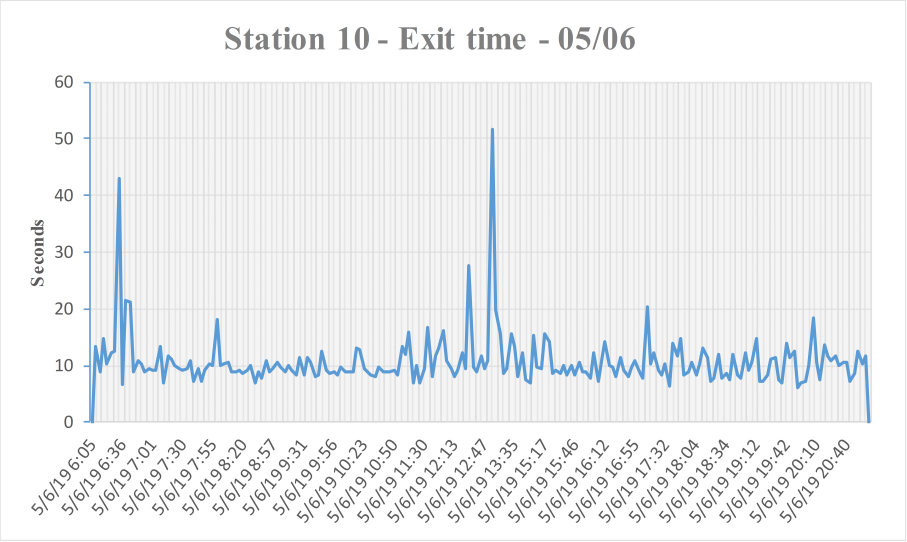


Figure 6.7: Station 10 exit times for June 5th.

The average exit time for week 23 is around 11 seconds.

Finally, the processes times are the following for station 10 (figures 6.8 and 6.9).

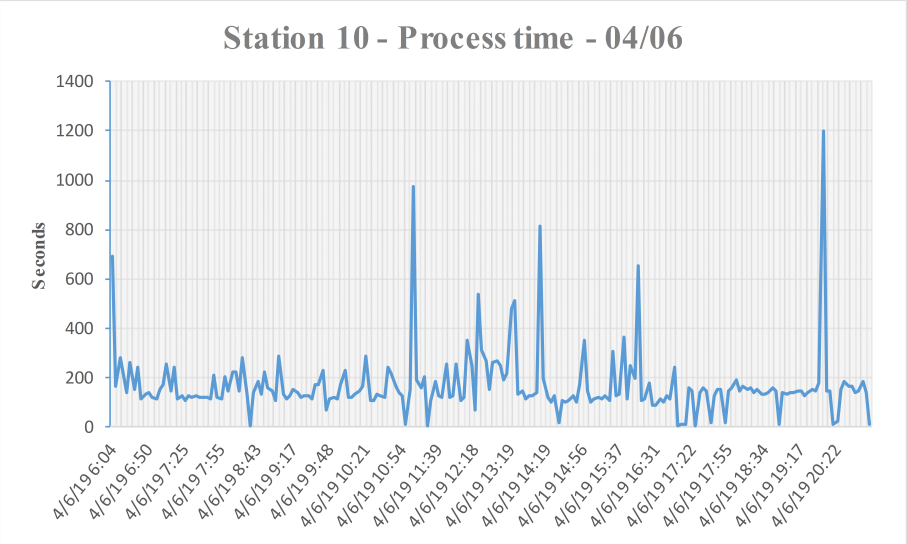


Figure 6.8: Station 10 process times for June 4th.

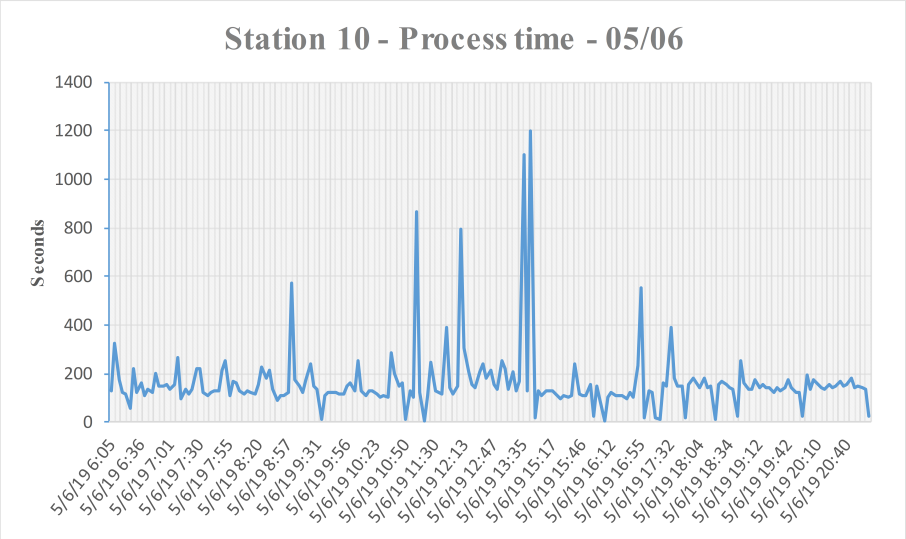


Figure 6.9: Station 10 process times for June 5th.

The average process time for this station, on week 23, without the outliers is 151 seconds. For station 12 we have the 46 seconds for the process time average (figures 6.10 and 6.11).

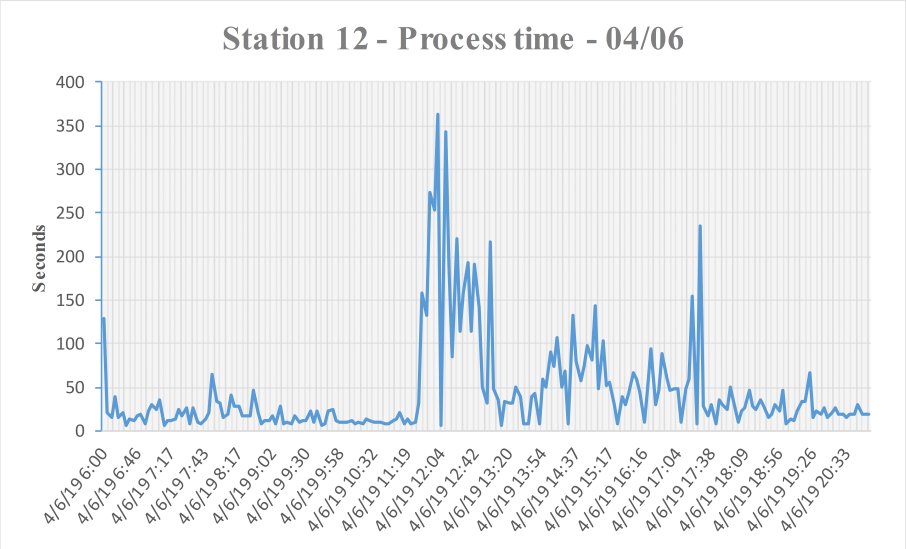


Figure 6.10: Station 12 process times for June 4th.

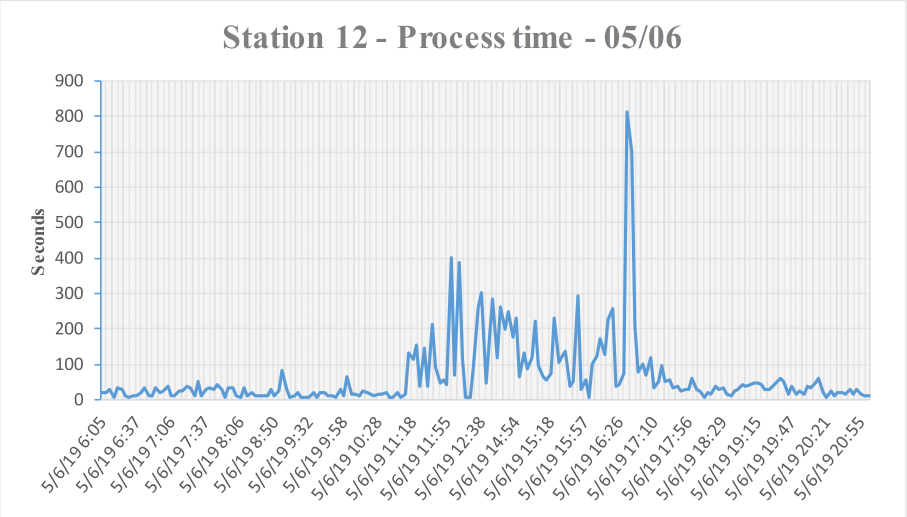


Figure 6.11: Station 12 process times for June 5th.

The big increase we see on the start of the second shift is derived from the change in product, where the process for the latter takes longer.

6.2.1 Data discussion

To recall the different processes and their execution times in the stations, figure 6.12 shows a chart with the station operations and their respective timers.

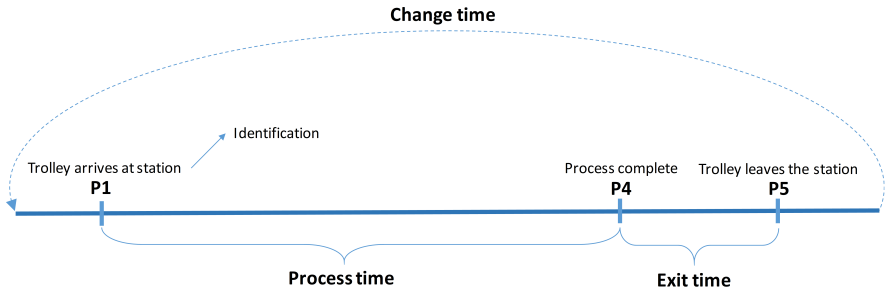


Figure 6.12: Process, exit and change times for the stations.

For the outliers, we removed every time that took over 400 seconds to complete, those being exceedingly high to be considered part of the normal procedure.

With this, below in tables 6.2 and 6.3 we have a summary of the times listed above.

Table 6.2: Week 23 of 2019, station 10 times.

Operation	Average time (s)
Changeover	11
Trolley exit	50
Product process	151
Total	212

Table 6.3: Week 23 of 2019, station 12 times.

Operation	Average time (s)
Changeover	158
Trolley exit	0
Product process	46
Total	204

As we can see, by adding all the times from station 10, we have a cycle time of around 212 seconds for each product, where as station 12 has 204 seconds, this is roughly the same, which is expected as both stations have identical production numbers.

To achieve a better line productivity and being station 10 the first one, this one should never have to stop working and subsequent stations should have a cycle time equal or lower as to not bottleneck the line. This is the ideal scenario, but problems arise and reworks must often be made to products, which cause delays. Station 10 has an average exit time of 11 seconds, which is a time loss that must be addressed, the only process that occurs in that time is the trolley exit and manual identification of the next code.

The process on 10 takes thrice as long, but the change over time on 12 is tripled, this is also normal as the process on station 12 is normally very short. The following figures show the average cycle time of station 10 and 12 (Week 23 of 2019, figures 6.13 and 6.14).

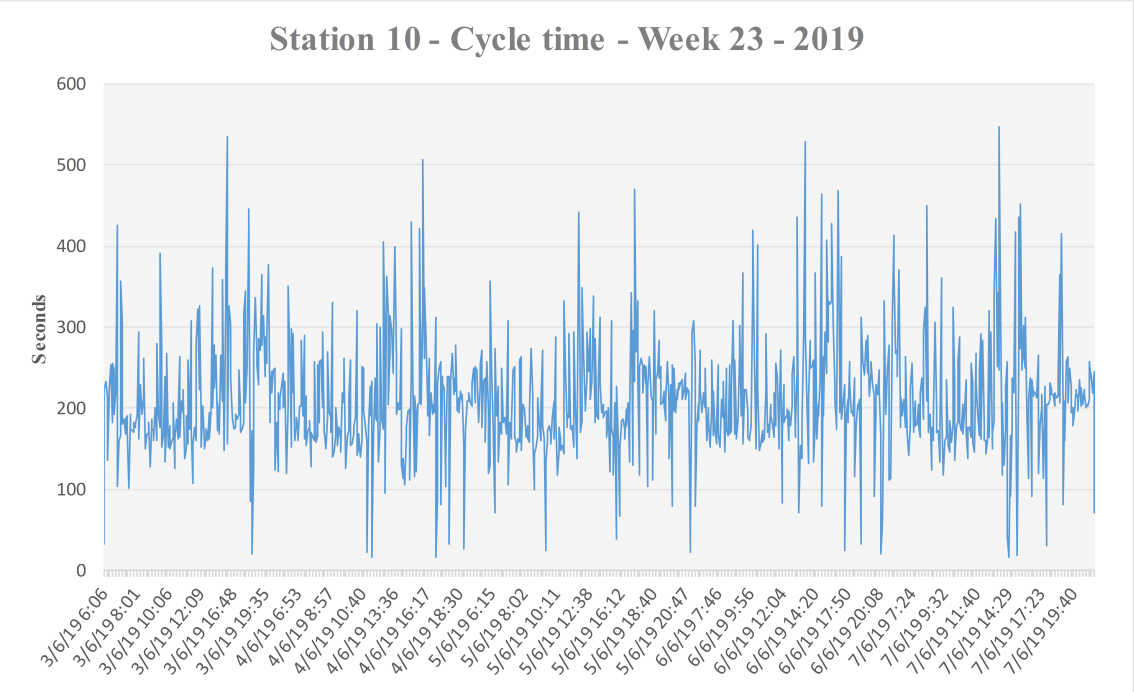


Figure 6.13: Station 10 cycle time for week 23 without outliers.

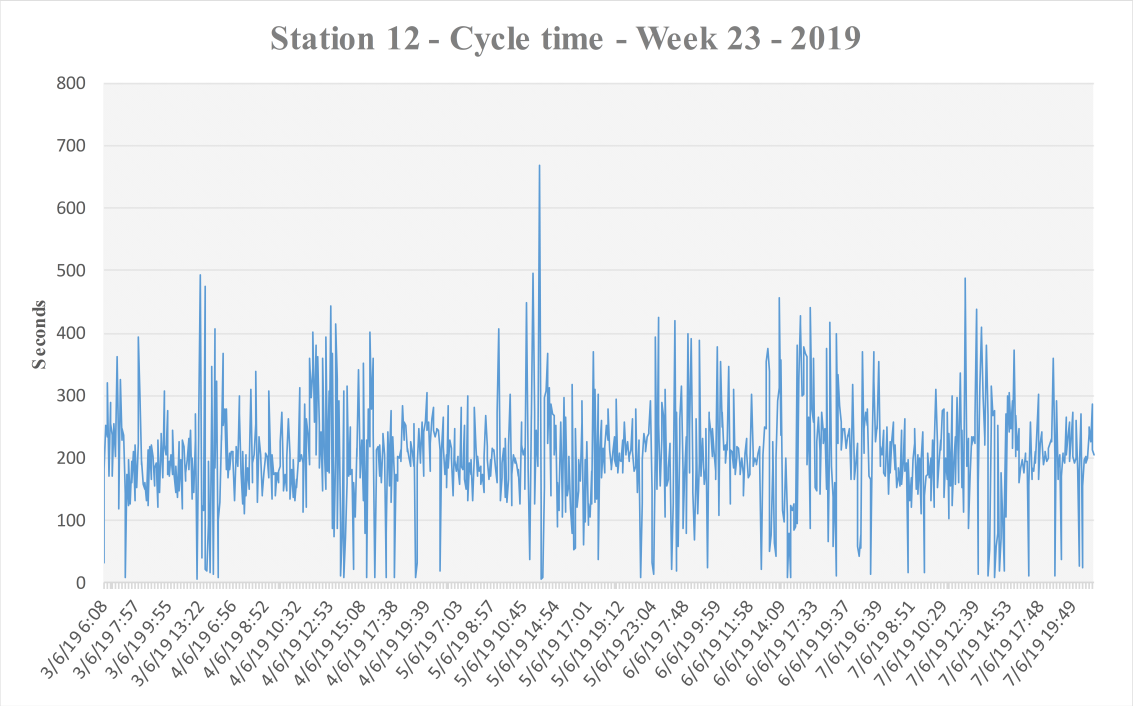


Figure 6.14: Station 12 cycle time for week 23 without outliers.

With this data, it is not possible to state whether the current system is faster than the older one, because these stations were not the bottleneck of the process (station 10 still has to manually identify to write the tag). But it is a step in the right direction for the line managers and relevant engineering teams to evaluate and do further analysis on all the stations time scheduling and also we can establish that the cycle time by product has been reduced (by removing VT time from the operator, previously determined by the Methods and Times engineering team TEF6).

6.3 Time results average for the line

As of week 25 of 2019, like stated before, we have 10 different stations that are collecting data, and the average process, exit and change times are the stated in table 6.4 for each station (times above 400 seconds removed to make the data accurate as possible, due to them being shift changes and breaks).

Table 6.4: Time analysis for line 7 in 10 different stations, on June 19th.

Station	Avg. process time(s)	Avg. exit time(s)	Avg. change time(s)	Avg. cycle time(s)
33	100.35	18.47	78.68	197.50
34	66.55	25.06	100.34	191.95
40	35.17	49.08	97.15	181.40
42	10.91	36.51	163.72	211.14
43	139.60	32.98	46.68	219.25
50	141.41	23.72	48.42	213.55
60	143.15	51.56	36.13	230.84
81	118.59	9.16	85.08	212.83

The stations where the average cycle time is higher need to be addressed to improve line productivity, namely station 60 where the exit time is currently 51.56 seconds and needs to be much lower in order to reduce the 230.84 seconds of average cycle time.

6.4 Risk Assessment

During the implementation of any project, it is necessary to consider the possible risks that may occur and identify their likelihood. In this sense, risk management arises, a process through which the risks of the project are recorded in order to reduce or completely avoid the consequences of an unforeseen development event.

A risk can be mapped through a qualitative risk matrix, evaluating it in two parameters: probability and consequence (see table 6.5).

Table 6.5: Project risk assessment.

Risk Assessment		Consequence				
		Insignificant	Minor	Moderate	Critical	Catastrophic
Likelihood	Unlikely	Very Low	Very Low	Low	Medium	High
	Seldom	Very Low	Low	Medium	Medium	High
	Occasional	Low	Medium	Medium	High	Extreme
	Likely	Low	Medium	High	High	Extreme
	Definite	Medium	High	High	Extreme	Extreme

Within the scope of the project at hand, there are some risks that have already been outlined and duly mapped into a qualitative risk matrix. These are presented below, together with the respective qualitative values of probability and consequence, as well as measures for prevention and mitigation of consequences. They are divided in two different tables and risk types, the technical and management risks (see tables 6.6 and 6.7).

Table 6.6: Technical risks for the risk assessment.

Description	Risk classification	Preventive measures	Consequence mitigation
1.1) Technical failure of the RFID reader		Assure that the reader is secure and protected by the stopper and its connections are well established	Manual read by the barcode scanner to bypass the RFID reader
1.2) Technical failure of the RFID tag		Design protective mounting covers for the tag and have a good and durable support	Replace the tag as soon as possible
1.3) Wrong product - trolley association (placing a trolley with a different product code than the one being worked on in the write station)		Training operators on how the process should take place	Correction via adjustment of the traceability database
1.4) Tag writing error		Execute a read procedure to ensure that tag was written into	Retry the writing process
1.5) Tag reading error		Verify the read code and check it against known parameters like code length and factory identification inside the code	Retry the reading process
1.6) Trolley general failure (needing replacement midway through the line)		None	Alert the line operators that the replacement trolley will not have the correct product code in its tag, manual barcode scanning is needing until it reaches the last station

Table 6.7: Project management risks for the risk assessment.

Description	Risk classification	Preventive measures	Consequence mitigation
2.1) Change in requirements for some tasks		None	Revaluate the task effort and schedule accordingly
2.2) Wrong electrical schematic for third party modifications		Edit the schematic when the definite changes to the stations electric side are complete	Notify the relevant persons that changes are in progress for those stations
2.3) Task taking more effort than expected to accomplish		When scheduling the task, always add extra time to account for these time losses	Re-schedule the time plan if it does not interfere with the deadline, otherwise compensate by working more hours
2.4) Late arrival of material		Order in advance, allowing delays to have no effect on the development of the project	None
2.5) Delays in tasks of other teams that can affect our project (either directly or indirectly)		Check the availability and schedule of these teams and their planned work to act accordingly	None
2.6) Production disruption (caused directly or indirectly by our involvement)		Always double-check everything that is being done, and validate the changed stations before leaving (run a full production cycle to validate)	Reset changes made

6.4.1 Risks incurred

During the course of the project, some of the expected risks occurred, as well as unexpected ones. Despite the preventive measures taken, the reality of these projects is that delays and unexpected behaviours happen and have to be dealt with, that is one of the points of a risk assessment, to have a contingency plan even in the worst cases.

One of the main concerns with the project implementation is risk 1.3. In the event that a trolley arrives at a reading station with a product other than the one written in the tag, it means that the product in the tag has already been packaged. The operator may not notice that the number is different (it is not checked and it is inconceivable to be checked, a lot of time would be lost in that process) and go ahead with the work. This results in a very serious error, because in the system, for traceability purposes, there are associations of components to the product and when this product is reworked at the station (because the tag is written with a code that has already passed that station), there will be disassociation of internal components to a product that is already on its way to the warehouse. If there is a complaint in the final consumer, it will be very difficult to trace the component, and in addition the new product that is being produced will not have the correct association. This causes loss of profit unnecessarily (the product is either thrown away or it has to be reopened and reworked with the correct component). This is why even though the write station rewrites the old product code, a clear station was introduced in case of failure on that write station (memory is now empty instead of having a valid product code).

This risk (1.3) happened twice on the first week of the project implementation, but it was noticed by the line operators when they arrived at a manual barcode station (e.g. station 31) and the automation gave out the error that the product did not OK in the last station (which it did, but with the wrong barcode). Both of these occurrences happened on shift changes or breaks, which led to confusion in which trolley should go where/was already written on station 10, but since then the operators were instructed on the correct write procedure (always have the trolley being worked on against the stopper) and the clear station was added, so this risk is mitigated and should not happen any more.

Risk 2.3 also happened on the proof of concept task, arising from the Profibus-DP baudrate problem. Risk 2.5 also affected the project, due to the tag supports being delivered 1 week later.

6.5 Satisfaction analysis

As the project in question was integrated at the industrial level, several people in different departments that are aware of the project were asked to give feedback about it. This is of extreme relevance to assess how the project is affecting production and whether it is having the intended results. Thus, in this way, a satisfaction analysis was performed to determine if the work carried out is in accordance with the expectations and objectives of the company. Analyses were requested from a process development engineer, the group leader from this engineering team and several operators. The entire comments are in the appendix, but some of the remarks are as follows:

- According to a process development engineer, "This project represented the first very significant step towards what is expected to be the future of our division in terms of product identification", "For the particular line where it was implemented, the gains in 8 different stations are evident in the reduction of operator time dedicated to identification operations, and a return on investment is expected in about one year", "(...) anywhere in the world today it is simple (...) to consult in real-time, information about which products are being processed in which stations, with results and statistical data of the shift that is currently taking place".
- The group leader of Industry 4.0 and development of new processes also commented, "The system was designed and successfully developed and industrialized on schedule. The industrialization highlighted the reduction of time, above the initial forecasts (...) in the various stations of the line", "(...) because this implementation is being successful, it works as a proof of concept that we intend to implement in other lines", "(...) its frontend is elegant and offers transparency about the production flow (...) thus identifying the continuous improvement teams where they should act to increase the production capacity of the line".
- Some operators in the line also expressed how the new system helps them, namely by not having to reach for the manual barcode reader, this has helped them get started on their procedures much faster. The operator from station 60 even stated that not only did he not lose time identifying the product, as soon as the trolley reached the station some automatic procedures begun automatically and it saved him more time than previously thought.

6.6 Summary

In this chapter we analysed thoroughly all the data from the two pilot stations as well as the other 8 stations where the system was implemented. We checked RFID read/write status and process, change and exit times, and also made a risk assessment and performed a satisfaction analysis. More concretely, in this chapter:

- We analysed all the RFID read and write results for a whole week, and determined the percentage of correct writes and reads on stations 10 and 12.
- We also analysed all the RFID read status from stations 34, 33, 40, 42, 43, 50, 60 and 81, and the clear status from station 81.
- We checked the change times for two different days for stations 10 and 12, exit times for station 10 and process times from both.
- We analysed the overall times from the other implemented stations.
- A risk analysis has been created to better understand all the risks and associated problems that can happen, and avoid them or at least prepare for them.
- Lastly, a satisfaction report was asked of the line operators and clients of the project in order to receive feedback and improve the project further.

Chapter 7

Conclusion and next steps

This chapter will focus on a retrospective work, with special emphasis on the evaluation of the technologies studied previously, specifically if the proposed solution made sense, after having worked with it. A lessons learned section for the project at hand, taking into account the industrial environment and finally the future work that this project entails.

In the initial steps of this dissertation, to arrive at the proposed solution, meetings were held with various suppliers and companies to ascertain the state of the art in terms of product identification that they had. This part is of extreme relevance because an initial study that is not carried out properly can lead to catastrophic results during the project development. Thus, with the objectives and requirements of the work at hand defined, the implementation of the system began, always with the utmost focus on robustness.

The RFID reader communication brought some challenges, such as the programming of the serial bus terminal communication and the problems with message segmentation due to the low internal buffer. A good part of the effort spent in the RFID library was to make it as robust as possible to contemplate potential avenues of failure.

For the front-end product tracking application, there was no previous experience with the chosen framework, nonetheless, it was the selected solution because it would speed up the entire development process once some skill with it developed and it is a standard framework for this type of applications at Bosch.

The results obtained from the project are very satisfactory, and will allow for future line optimizations, either by balancing operators because their order of operations changed (by not having to identify the product), or by the new time data that is available, allowing for further identification of problems such as bottlenecks.

7.1 Proposed solution look back

Looking at the identification technologies, we have the 1D and 2D codes that are still widely used in the industry, the radio frequency identification that was the adopted solution, optical character recognition, smart cards and speech recognition. Analysing all these techniques of product

identification, for our specific case the proposed solution was the more appropriate choice, having a cost/benefit and performance superior to others while being relatively simple in terms of the changes required in the stations.

Additionally, by examining RF transmission based localization techniques, while offering the possibility to track the products between the lines, which would greatly improve the current real-time line map, there is the robustness and accuracy concern that comes associated with it. Anything less than 100 percent accuracy for the product-station association can be a very serious problem (see risk 1.3 in 6.4). With this and with the current results, we think that the proposed and implemented solution were the better choice.

Furthermore, on the web application front, the designed mock-up (see section 3.2.4) looks very similar to the created application, with the addition of having shift stats on the left. With this, the concept of digital twin mentioned was also used, with the markers clickable and information about the products displayed, namely the unique product code, which can be searched on the MES and a complete description of every work that has been done to it available there, at every station with results and times.

7.2 Lessons learned

There are very important lessons to learn from my dissertation internship here at Bosch Termotechnology SA, namely the differences between projects in academic and business environments. Some of these differences and also other points are described below:

- During the project there were several guidelines and standards that had to be followed to ensure that we were always operating within the company norms. Because of this, more often than not the project development is usually slower.
- There are numerous procedures for all types of requests, such as requesting material, intervening on production lines or simply asking for support from a team member on a specific task.
- Every week there were project reviews, where all last week's tasks were evaluated, if they were fulfilled and if there were no delays. A weekly plan of tasks amounting to 40 hours of work was then made, with estimated time on tasks, milestones evaluation and also a global rundown of the project. All of this helped to achieve a good project, following all its life-cycle steps.
- The approval of changes in the production line must pass through a wide range of people for approval, which are responsible for several different areas of the factory.
- Training had to be given to the line managers and operators in order to use the implemented project effectively. Furthermore, because of this, the station alterations must be made in such a way that every type of failure is accounted for and it does not affect their order of operations much, that is why their feedback is of the utmost importance, as the project at hand is essential to boost their productivity.

- Industrial shop floor version control, in addition to have standard version control features, not only allows for the storage of the current and previous code versions, but in our case (deployed PLC code), it compares the latest uploaded code version with the one running on the PLC in real-time to check if they match. This is done to prevent deployment of code without saving it first.

7.3 Contributions

This dissertation main contributions are the following:

- Introduces a new identification system for Bosch Termotechnology S.A. in Aveiro that brings remarkable improvements and can be applied in other lines;
- Greater knowledge of station times, so that continuous improvements teams can further refine the efficiency and productivity of the line;
- Increased know-how in HF RFID solutions for the company;
- Provides a way to know the state of the line anywhere, without having to physically see it.

7.4 Next work

The work that will be carried on after this mainly involves replicating the system on lines that can benefit from it. Studies will be conducted on the other trolley-based lines to determine if a system like this has returns in a 2 year time-frame. If so, this project will be rolled out to those lines to. Also on line 7, the remaining stations will be retro-fitted with the system already present on station 12.

Appendix A

Technical drawings

In this appendix, we present the technical drawings of the tag support.

A.1 Steel frame technical drawing

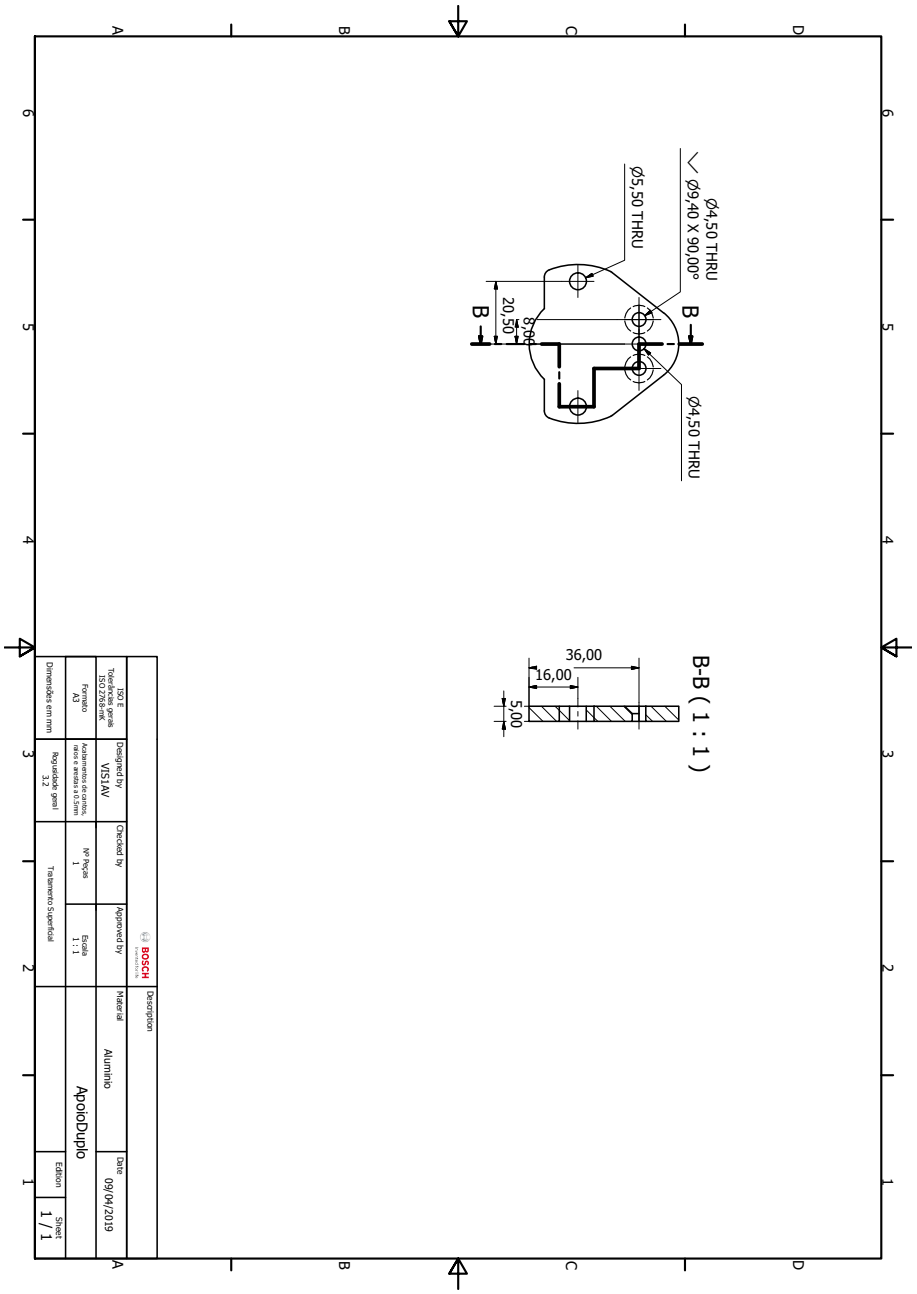


Figure A.1: Technical drawing for the steel frame.

A.2 Aluminium cylinder technical drawing

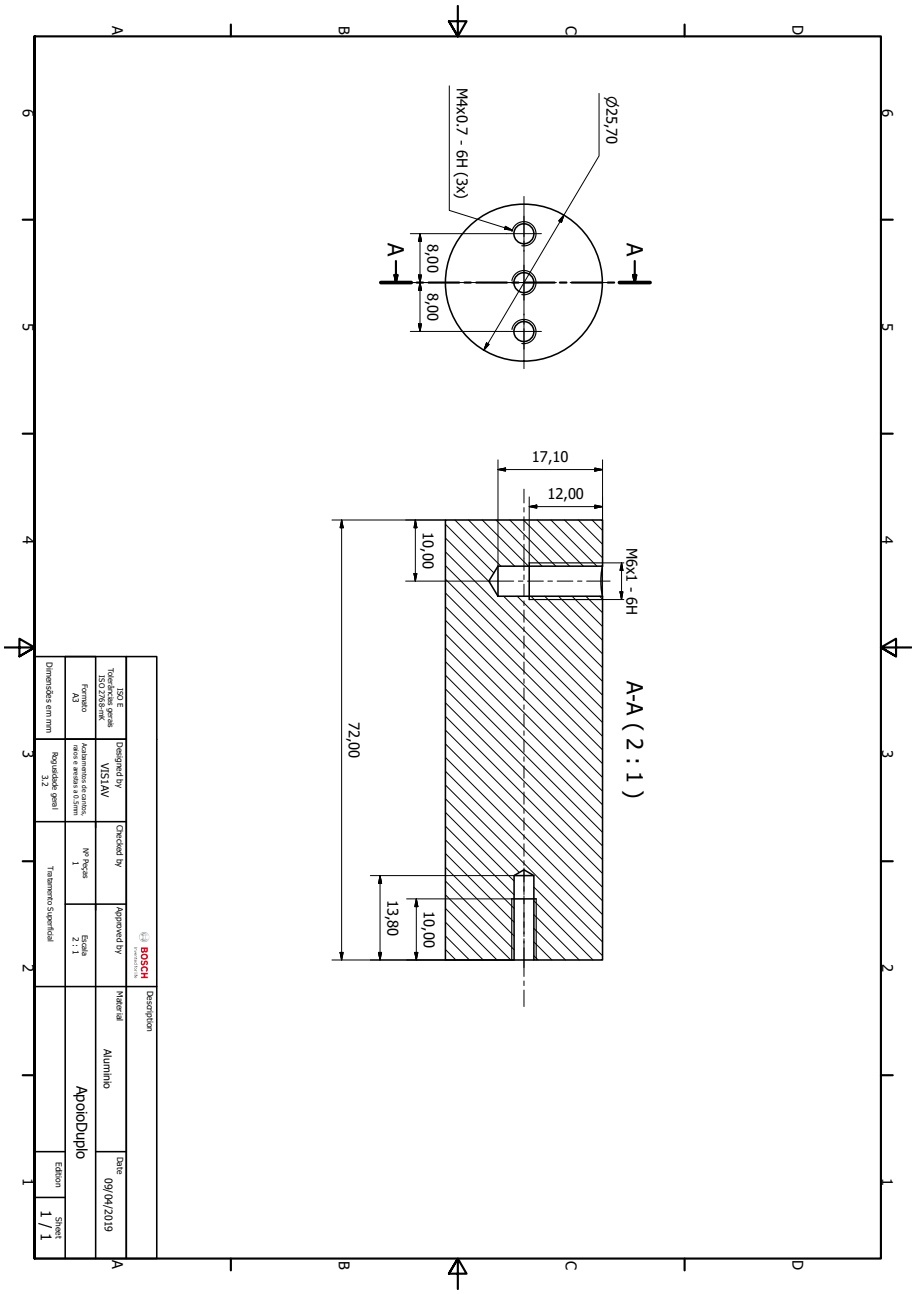


Figure A.2: Technical drawing for the aluminium cylinder.

A.3 Plastic spacer technical drawing

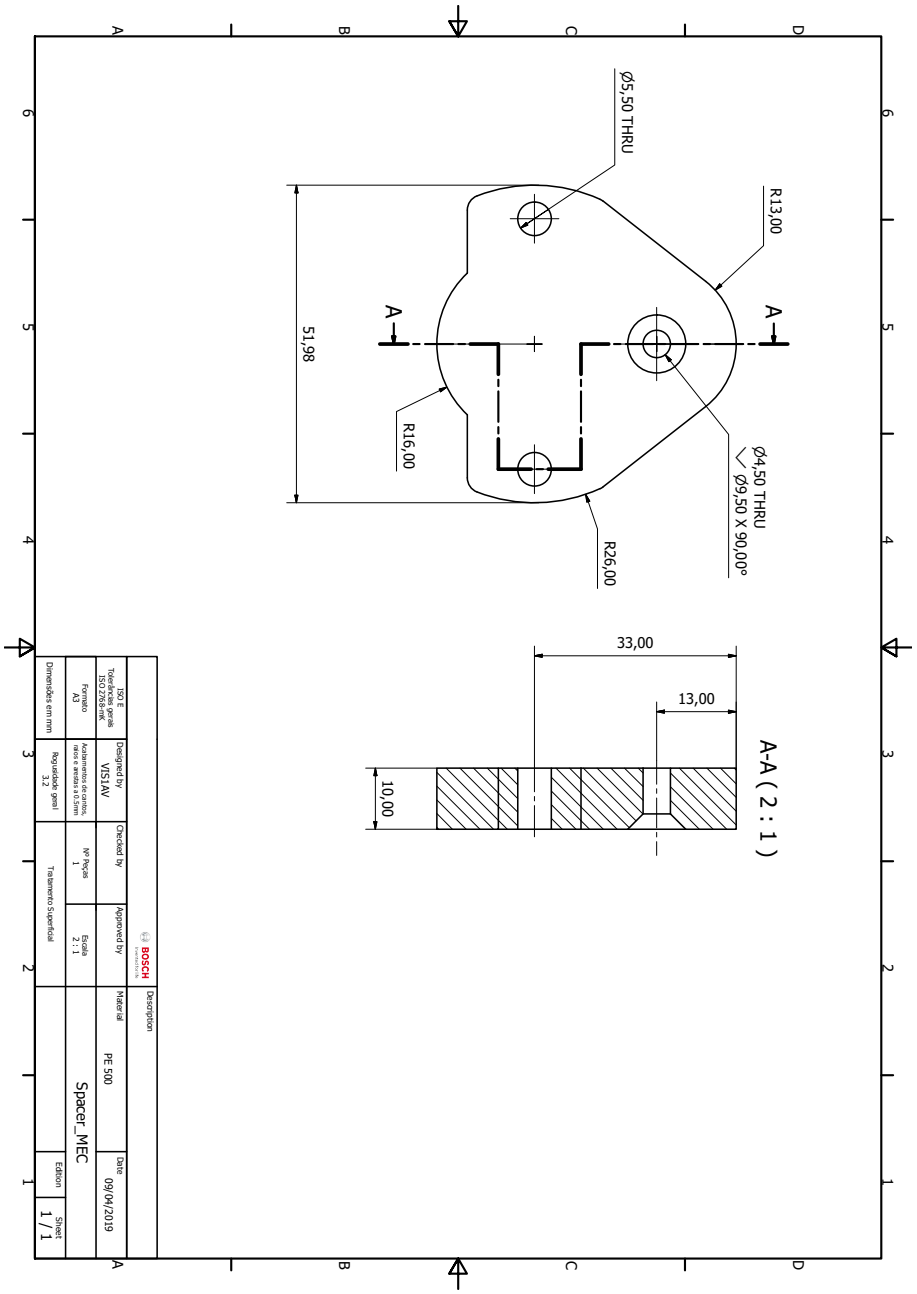


Figure A.3: Technical drawing for the plastic spacer.

References

[1] Siemens AG. Rfid systems ascii protocol for rf200, 2016.

[2] Vipul Chawla and Dong Sam Ha. An overview of passive rfid. *IEEE Communications Magazine*, 45(9), 2007.

[3] Jiang Chengkun, He Yuan, Zheng Xiaolong, and Liu Yunhao. Orientation-aware rfid tracking with centimeter-level accuracy. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 11-13 April 2018*, 2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). Proceedings, pages 290–301. IEEE Computer Society.

[4] Nak-Gwon Choi, Hyuek-Jae Lee, Sang-Hoon Lee, and Seong-jeen Kim. Design of a 13.56 mhz rfid system. In *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, volume 1, pages 4–pp. IEEE, 2006.

[5] IO-Link Community. Io-link system description, 2016.

[6] D. Dardari, P. Closas, and P. M. Djurić. Indoor tracking: Theory, methods, and technologies. *IEEE Transactions on Vehicular Technology*, 64(4):1263–1278, 2015.

[7] Waltenegus Dargie and Christian Poellabauer. Fundamentals of wireless sensor networks: Theory and practice - chapter 10.

[8] Reinhard Exel and Thilo Sauter. A comparison of time-and rss-based radio localization for the factory floor. In *Factory Communication Systems (WFCS), 2012 9th IEEE International Workshop on*, pages 13–20. IEEE, 2012.

[9] Klaus Finkenzeller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons, 2010.

[10] Christian Floerkemeier and Sanjay Sarma. Rfidsim - a physical and logical layer simulation engine for passive rfid. *Automation Science and Engineering, IEEE Transactions on*, 6:33 – 43, 02 2009.

[11] Feng Gang, Li Jian-qiang, Luo Chengwen, and Ming Zhong. Ptrack: a rfid-based tracking algorithm for indoor randomly moving targets. In *Smart Computing and Communication. First International Conference, SmartCom 2016, 17-19 Dec. 2016*, Smart Computing and Communication. First International Conference, SmartCom 2016. Proceedings: LNCS 10135, pages 144–53. Springer International Publishing.

[12] J. Z. Gao, L. Prakash, and R. Jagatesan. Understanding 2d-barcode technology and applications in m-commerce - design and implementation of a 2d barcode processing solution. In *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, volume 2, pages 49–56, July 2007.

[13] Mehdi Golestanian, Joshua Siva, and Christian Poellabauer. Radio frequency-based indoor localization in ad-hoc networks. In Jesus Hamilton Ortiz and Alvaro Pachon de la Cruz, editors, *Ad Hoc Networks*, chapter 6. IntechOpen, Rijeka, 2017.

[14] H Golnabi and A Asadpour. Design and application of industrial machine vision systems. *Robotics and Computer-Integrated Manufacturing*, 23(6):630–637, 2007.

[15] Yanying Gu, Anthony Lo, and Ignas Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications surveys & tutorials*, 11(1):13–32, 2009.

[16] M. H. Habaebi, R. O. Khamis, A. Zyoud, and M. R. Islam. Rss based localization techniques for zigbee wireless sensor network. In *2014 International Conference on Computer and Communication Engineering (ICCCE), 23-25 Sept. 2014*, 2014 International Conference on Computer and Communication Engineering (ICCCE), pages 72–5. IEEE.

[17] R Heynicke, D Krush, G Scholl, B Kaercher, J Ritter, P Gaggero, and M Rentschler. Io-link wireless enhanced sensors and actuators for industry 4.0 networks. In *AMA Conferences*, pages 134–138, 2017.

[18] Jacob D. Hochhalter. Coupling damage-sensing particles to the digital twin concept. pages 258–262. Langley Research Center, April 2014.

[19] S. Holm. Ultrasound positioning based on time-of-flight and signal strength. In *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN 2012), 13-15 Nov. 2012*, 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN 2012), page 6 pp. IEEE.

[20] V Daniel Hunt, Albert Puglia, and Mike Puglia. *RFID: a guide to radio frequency identification*. John Wiley & Sons, 2007.

[21] EPCglobal Inc. Epc radio-frequency identity protocols generation-2 uhf rfid, 2013.

[22] T. Janssen, M. Aernouts, R. Berkvens, and M. Weyn. Outdoor fingerprinting localization using sigfox. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 24-27 Sept. 2018*, 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), page 6 pp. IEEE.

[23] J. Jerabek, L. Zaplatilek, and M. Pola. A proposal of radio ultrawideband systems for precision indoor localization. In *2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA), 21-22 April 2015*, 2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA). Proceedings, pages 355–8. IEEE.

[24] Zhong Jubin, Lin Zhiyun, Chen Zhiyong, and Xu Wenyuan. Cooperative localization using angle-of-arrival information. In *2014 11th IEEE International Conference on Control; Automation (ICCA), 18-20 June 2014*, 11th IEEE International Conference on Control Automation (ICCA), pages 19–24. IEEE.

[25] M. Lamot and B. Zalik. A contribution to triangulation algorithms for simple polygons. *Journal of Computing and Information Technology - CIT*, 8(4):319–31, 2000.

[26] S. Lanzisera, D. Zats, and K. S. J. Pister. Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization. *IEEE Sensors Journal*, 11(3):837–45, 2011.

[27] Roger Lindau and Kenth Lumsden. The use of automatic data capture systems in inventory management. *International Journal of Production Economics*, 59(1):159 – 167, 1999.

[28] N. Mangalvedhe, R. Ratasuk, and A. Ghosh. Nb-iot deployment study for low power wide area cellular iot. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 4-8 Sept. 2016*, 2016 IEEE 27th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), page 6 pp. IEEE.

[29] Manpreet and J. Malhotra. Zigbee technology: current status and future scope. In *2015 International Conference on Computer and Computational Sciences (ICCCS), 27-29 Jan. 2015*, 2015 International Conference on Computer and Computational Sciences (ICCCS), pages 163–9. IEEE.

[30] L. Mao, Y. Zhang, and J. Wan. Application of 2d barcode technology based on image processing. In *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, pages 2613–2616, May 2018.

[31] Rainer Mautz. Overview of current indoor positioning systems. *Geodezija ir Kartografija*, 35(1):18–22, 2009.

[32] A. J. Mercer, R. K. James, G. Bennett, P. Patel, C. Johnston, and J. Cai. Performance testing of rfid systems with rf-harsh materials. In *2011 IEEE International Conference on RFID-Technologies and Applications*, pages 537–543, Sep. 2011.

[33] J. Mesquita, D. Guimaraes, C. Pereira, F. Santos, and L. Almeida. Assessing the esp8266 wifi module for the internet of things. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), 4-7 Sept. 2018*, volume vol.1 of *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 784–91. IEEE.

[34] Y. Mizuchil, T. Ogura, Kim Young-Bok, Y. Hagiwara, and Choi Yongwoon. Camera-based measurement for close-distance relative vessel positioning. *Journal of Mechanical Science and Technology*, 31(4):1899–907, 2017.

[35] Stephanie Neil. Track parts and materials on the fly, 2014.

[36] Bryan O'Keefe. Finding location with time of arrival and time difference of arrival techniques, 2017.

[37] V. Osa, J. Matamales, J. F. Monserrat, and J. Lopez. Localization in wireless networks: the potential of triangulation techniques. *Wireless Personal Communications*, 68(4):1525–38, 2013.

[38] C. Pereir, D. Guimaraes, J. Mesquita, F. Santos, L. Almeida, and A. Aguiar. Feasibility of gateway-less iot e-health applications. In *2018 European Conference on Networks and Communications (EuCNC), 18-21 June 2018*, 2018 European Conference on Networks and Communications (EuCNC), pages 324–8. IEEE.

[39] D. Plets, N. Podevijn, J. Trogh, L. Martens, and W. Joseph. Experimental performance evaluation of outdoor tdoa and rss positioning in a public lora network. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 24-27 Sept. 2018*, 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), page 8 pp. IEEE.

[40] A. Povalac and J. Sebesta. Phase difference of arrival distance estimation for rfid tags in frequency domain. In *2011 IEEE International Conference on RFID-Technologies and Applications (RFID-TA 2011), 15-16 Sept. 2011*, 2011 IEEE International Conference on RFID-Technologies and Applications (RFID-TA), pages 188–93. IEEE.

[41] Jiang Qing, Qiu Feng, Zhou Mu, and Tian Zengshan. Benefits and impact of joint metric of aoa/rss/tof on indoor localization error. *Applied Sciences*, 6(10):296 (18 pp.), 2016.

[42] K. V. S. Rao, P. V. Nikitin, K. V. S. Rao, and P. V. Nikitin. Theory and measurement of backscattering from rfid tags. *IEEE Antennas and Propagation Magazine*, 48(6):212–218, Dec 2006.

[43] Chris M Roberts. Radio frequency identification (rfid). *Computers & security*, 25(1):18–26, 2006.

[44] A. M. Sabu and A. S. Das. A survey on various optical character recognition techniques. In *2018 Conference on Emerging Devices and Smart Systems (ICEDSS)*, pages 152–155, March 2018.

[45] S. Sadowski and P. Spachos. Rssi-based indoor localization with the internet of things. *IEEE Access*, 6:30149–30161, 2018.

[46] H. Schweinzer and M. Syafrudin. Losnus: An ultrasonic system enabling high accuracy and secure tdoa locating of numerous devices. In *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN 2010), 15-17 Sept. 2010*, 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN 2010), page 8 pp. IEEE.

[47] Guowei Shen, Rudolf Zetik, and Reiner S Thoma. Performance comparison of toa and tdoa based location estimation algorithms in los environment. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pages 71–78. IEEE, 2008.

[48] S. Shimada, H. Hashizume, and M. Sugimoto. Indoor positioning using reflected light and a video camera. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 24-27 Sept. 2018*, 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), page 8 pp. IEEE.

[49] John Stark. Product lifecycle management. In *Product lifecycle management (Volume 1)*, pages 1–29. Springer, 2015.

[50] R. Suksawang and W. Suwansantisuk. Performance of rss-based localization in unknown environments. In *2018 IEEE 14th International Colloquium on Signal Processing; Its Applications (CSPA), 9-10 March 2018*, 2018 IEEE 14th International Colloquium on Signal Processing Its Applications (CSPA), pages 25–30. IEEE.

[51] H. Taherdoost, M. Zamani, and M. Namayandeh. Study of smart card technology and probe user awareness about it: A case study of middle eastern students. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 334–338, Aug 2009.

[52] Fei Tao, Jiangfeng Cheng, Qinglin Qi, Meng Zhang, He Zhang, and Fangyuan Sui. Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 94(9):3563–3576, Feb 2018.

[53] Michael Tiegelkamp and Karl-Heinz John. *IEC 61131-3: Programming industrial automation systems*. Springer, 1995.

[54] J. Tiemann, F. Eckermann, and C. Wietfeld. Atlas - an open-source tdoa-based ultra-wideband localization system. In *2016 7th International Conference on Indoor Positioning and Indoor Navigation (IPIN), 4-7 Oct. 2016*, 2016 7th International Conference on Indoor Positioning and Indoor Navigation (IPIN), page 6 pp. IEEE.

[55] J. R. Tuttle. Traditional and emerging technologies and applications in the radio frequency identification (rfid) industry. In *1997 IEEE Radio Frequency Integrated Circuits (RFIC) Symposium. Digest of Technical Papers*, pages 5–8, June 1997.

[56] J. Vachálek, L. Bartalský, O. Rovný, D. Šišmišová, M. Morháč, and M. Lokšík. The digital twin of an industrial production line within the industry 4.0 concept. In *2017 21st International Conference on Process Control (PC)*, pages 258–262, June 2017.

[57] W. A. P. van Kleunen, K. C. H. Blom, N. Meratnia, A. B. J. Kokkeler, P. J. M. Havinga, and G. J. M. Smit. Underwater localization by combining time-of-flight and direction-of-arrival. In *OCEANS 2014 - TAIPEI, 7-10 April 2014*, OCEANS 2014 - TAIPEI, page 6 pp. IEEE.

[58] Shao Wenhua, Zhao Fang, Wang Cong, Luo Haiyong, T. Muhammad Zahid, Wang Qu, and Li Dongmeng. Location fingerprint extraction for magnetic field magnitude based indoor positioning. *Journal of Sensors*, 2016:1945695 (16 pp.), 2016.

[59] Hu Xin, Cheng Lianglun, and Zhang Guangchi. A zigbee-based localization algorithm for indoor environments. In *2011 International Conference on Computer Science and Network Technology (ICCSNT), 24-26 Dec. 2011*, volume vol.3 of *Proceedings of the 2011 International Conference on Computer Science and Network Technology (ICCSNT)*, pages 1776–81. IEEE.

[60] Chiou Yih Shyh, Tsai Fuan, and Yeh Sheng Cheng. A dead-reckoning positioning scheme using inertial sensing for location estimation. In *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 27-29 Aug. 2014*, 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pages 377–80. IEEE Computer Society.