



Università degli Studi di Padova

DEPARTMENT OF INFORMATION ENGINEERING
Master's Degree in Automation Engineering
(DM 270/04)

MASTER FINAL PROJECT

**ANALYSIS AND FULL AUTOMATION
OF AN INDUSTRIAL SYSTEM:
FUNCTIONAL OPTIMIZATION AND
VERTICAL INTEGRATION**

Candidate:
Nicolò Lorenzi
1076756-IAM

Supervisor:
Prof. Maria Elena Valcher
(University of Padova)
Correlator:
Prof. Julio Garrido Campos
(University of Vigo)

Contents

1	Introduction	9
1.1	The “Ricardo Marin” Laboratory	9
1.2	Vigo and PSA Peugeot Citroën	11
1.3	Motivation of the project	12
2	An overview of the tools	15
2.1	Automation with PLC and IEC 61131-3	15
2.2	Beckhoff technology for automation	17
2.3	Odette standard for barcodes	19
2.4	Petri nets	21
2.5	Electrical equipment	23
2.6	The AS-i network	25
2.7	Operating modes	27
2.8	Microsoft Visual Studio	28
3	The power & free conveyor	31
3.1	About power & free conveyors	31
3.2	Description of the system	31
3.3	Stations	35
3.4	Electrical equipment	37
3.4.1	Electric motor	37
3.4.2	Sensors	37
3.4.3	Actuators	38
3.4.4	PLC modules	38
3.5	Network configuration	40
3.6	Description of the automation procedure	41
3.6.1	Operation of the loading station	42
3.6.2	Operation of the working station	44
3.6.3	Operation of the unloading station	45
3.7	Programming within TwinCAT	46
3.8	Description of the related HMI	47

4	The double level conveyor belt	53
4.1	About belt conveyor systems	53
4.2	Description of the system	53
4.3	Elevators	56
4.3.1	The elevator of the warehouse area	58
4.3.2	The elevator of the robot area	59
4.4	Electrical equipment	59
4.4.1	Electric motor	59
4.4.2	Sensors	60
4.4.3	Actuators	61
4.4.4	PLC modules	62
4.5	Network configuration	63
4.6	Description of the automation procedure	64
4.6.1	Operation of the warehouse area	64
4.6.2	Operation of the robot area	66
4.7	Programming within TwinCAT	67
4.8	Description of the related HMI	69
4.9	Additional arrangements	74
4.9.1	The stand-alone operating mode	74
4.9.2	Smart management of the motor	75
5	Conclusions	77
5.1	Open Problems & future work	78
	Aknowledgment	81
A	About the ADS protocol	83
B	Technical drawings of the power & free conveyor	85
C	XML	95
	Bibliography	97
	Sitography	99

List of Figures

1.1	A schematic description of the “Ricardo Marin” laboratory. . .	10
1.2	A top view of Vigo with the PSA Peugeot Citroën plant. . . .	12
2.1	A screenshot of TwinCAT <i>System Manager</i>	18
2.2	A screenshot of TwinCAT <i>PLC Control</i>	19
2.3	An example of Code 39 used throughout the project.	20
2.4	The cordless barcode reader used throughout the project. . .	20
2.5	A simple Petri net with 2 places, 2 transitions and 1 token. . .	22
2.6	An inductive sensor of the power & free conveyor.	23
2.7	A photocell of the double level conveyor belt.	24
2.8	A limit switch of the power & free conveyor.	24
2.9	The components of a pneumatic actuator.	24
2.10	The MetraTest AS-i System tester connected to a slave. . . .	26
2.11	Petri net for the evolution of operating mode based on UNE EN 60204-1.	28
3.1	Higher floor (with a path deviation) in the area of power & free conveyor.	32
3.2	Lower floor in the area of power & free conveyor.	32
3.3	Two essential components of the power & free conveyor. . . .	33
3.4	Ends (U-turns) of the power & free conveyor.	33
3.5	Behaviour of pulling hooks in the power & free conveyor. . .	34
3.6	Railroad switches feature the path deviation.	34
3.7	The deviation from the standard path.	35
3.8	Differences between types in the stations of the power & free conveyor.	36
3.9	Scheme of the controller and its modules.	39
3.10	Petri net for the loading station of the power & free conveyor.	42
3.11	Petri net for the working station of the power & free conveyor.	44
3.12	Petri net for the unloading station of the power & free conveyor.	45
3.13	Main page of the HMI for the power & free conveyor.	48
3.14	<i>Manual</i> tab of the HMI for the power & free conveyor.	50
3.15	<i>Stacks</i> tab of the HMI for the power & free conveyor.	50

3.16	<i>Database</i> tab of the HMI for the power & free conveyor. . . .	51
4.1	High views of the belt conveyor system under consideration. . .	54
4.2	Exchange areas of the belt conveyor system.	56
4.3	Platform of the elevator in the warehouse area with its side brackets.	56
4.4	The swinging obstruction that controls the access to an elevator.	57
4.5	Entrances of the elevators.	58
4.6	Mechanism for the movement of the side brackets.	59
4.7	The elevator of the warehouse area with the electric motor and the control cabinet.	60
4.8	Two reed switches used in the elevator of warehouse area. . .	61
4.9	Scheme of the controller Beckhoff BC9050.	62
4.10	Petri net for the warehouse area of the belt conveyor system.	65
4.11	Petri net for robot area of the belt conveyor system.	67
4.12	Main page of the HMI for the conveyor belt.	70
4.13	<i>Configuration</i> tab of the HMI for the conveyor belt.	71
4.14	<i>Manual</i> tab of the HMI for the conveyor belt.	72
4.15	<i>Stacks</i> tab of the HMI for the conveyor belt.	72
4.16	<i>Monitorization</i> tab of the HMI for the conveyor belt.	73
4.17	Evolution of the working modes in absence of an HMI for the belt conveyor system.	74
4.18	Control buttons and indicator lights next to warehouse area.	75
5.1	Final checks on both installations.	78
B.1	The location of “Ricardo Marin” laboratory.	86
B.2	The framework of the power & free conveyor.	87
B.3	Positions of the stations and pulling system.	88
B.4	A drag element of the pulling system.	89
B.5	The hook of a drag element.	90
B.6	A cart for the transportation of goods in boxes.	91
B.7	The framework of a generic station.	92
B.8	The lower bracket of a generic station.	93
B.9	The higher bracket of a generic station.	94
C.1	The XML document used in the configuration of an HMI for the power & free conveyor.	96

List of Tables

2.1	Mapping of a byte (of neither inputs or outputs) within the AS-i process image.	26
3.1	Sensors of the power & free conveyor.	37
3.2	Pneumatic actuators of the power & free conveyor.	38
3.3	Mapping of the input AS-i slaves in the power & free conveyor.	40
3.4	Mapping of the output AS-i slaves in the power & free conveyor.	41
3.5	Alarms of the power & free conveyor.	47
4.1	Relevant sizes of the double level conveyor belt.	54
4.2	Sensors in the warehouse area of the belt conveyor system.	60
4.3	Sensors in the robot area of the belt conveyor system.	60
4.4	Actuators in the warehouse area of the belt conveyor system.	61
4.5	Actuators in the robot area of the belt conveyor system.	61
4.6	Mapping of the input AS-i slaves in the belt conveyor system.	63
4.7	Mapping of the output AS-i slaves in the belt conveyor system.	64
4.8	Alarms of the belt conveyor system.	69

Chapter 1

Introduction

1.1 The “Ricardo Marin” Laboratory

This final project for the Master’s degree in Automation Engineering was developed at the University of Vigo (Galicia, Spain) during my stay of five months thanks to Erasmus+ Programme.

The “Ricardo Marin Martin” laboratory is located in the Industrial Engineering School building at the University Campus (CUVI). The lab is accessed by professors and students who belong to the Department of Automation and Systems Engineering; the research activity developed in the lab primarily covers the areas of motion control (non-conventional kinematic robotics) and industrial informatics (automation, warehousing, transportation, integration).

The lab is provided with an assembly plant prototype that is made up of transportation and warehousing systems which were produced by the local company Esypro Group: it represents the processing to which some raw goods are subjected in order to obtain a final product that has to be stored. The plant, that is described in detail in [1], consists of various installations:

- a “pick to light” system¹;
- an automatic overhead storage and retrieval system;
- a power & free conveyance system;
- a belt conveyor system;
- an automatic stacker crane for boxes;

¹It is an installation with a few sites for manual storage and withdrawal of goods; every site has a digital device that displays light signals and information in order to help the human worker. This feature suggests the origin of the name.

- manipulating ABB and Fanuc robots.

A useful representation of the laboratory and its equipment is given in Figure 1.1: it shows that the plant layout is structured on two levels: a ground level and an overhead one.

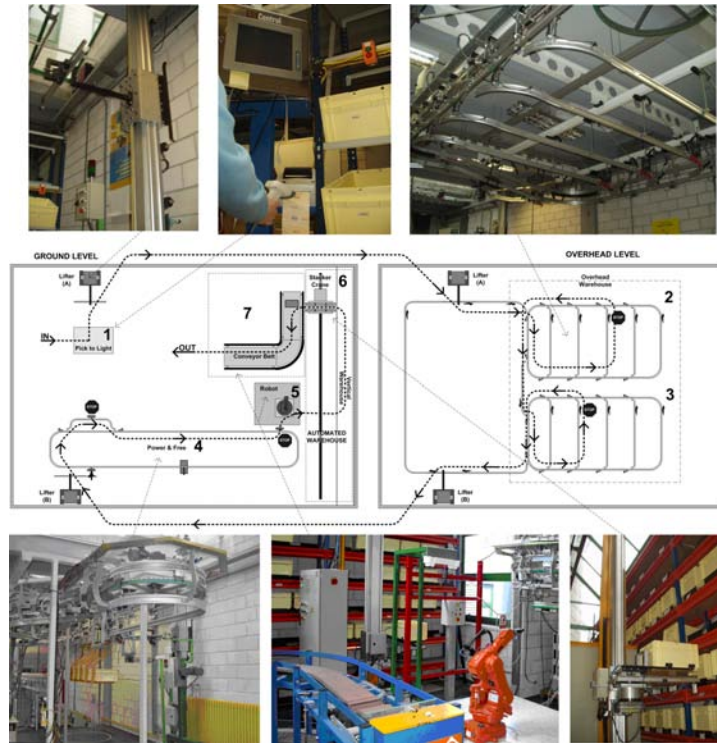


Figure 1.1: A schematic description of the “Ricardo Marin” laboratory.

Modern industrial systems follow CIM², which is a manufacturing strategy that employs computers to control the entire production process; the benefits of this choice are given by cost savings, due to increased efficiency, and a higher product quality. CIM is based on a complex communication network for the exchange of information that works horizontally across departments and vertically across sectors. A kind of pyramidal structure divided in levels develops from the bottom to the top as follows:

- **process** (level 0): it includes sensors and actuators of the plant;
- **machine** (level 1): it contains PLCs that control the lower level by employing AS-i networks;

²Computer-Integrated Manufacturing.

- **cell** or **area** (level 2): it is responsible for the supervision of industrial computers and PLCs within the previous level. Ethernet and wireless networks are used to this end;
- **factory** (level 3): it manages technical and economic aspects that are involved in the manufacturing process.

For what it concerns the automation of the plant in “Ricardo Marin” lab, it is important to remark the isolation of each installation from the others; this means that an **horizontal integration** is needed with the aim to make all manufacturing units collaborating in an integrated production process. For instance, if a product that is being carried by the power & free conveyor is no longer required within the supply chain then the belt conveyor system has to be informed in order to receive that product and move it to the warehouse; otherwise, the box containing that product could keep occupying the unloading station of the power & free conveyor, with a high probability of causing an inappropriate cart jam.

Moreover, for what it regards CIM and relations between different domains of its characteristic pyramid, a **vertical integration** is required in order to optimize resource management and reduce manufacturing time. As a matter of fact, the implementation of human-machine interfaces³ is designed with the intention to transmit information to higher level processes, such as an MES⁴ that generates production orders and keeps track of them; furthermore, traceability systems and maintenance support systems allow to store relevant data concerning the manufacturing process and hence the system management proves to be simpler and more functional.

1.2 Vigo and PSA Peugeot Citroën

Nowadays, Vigo hosts one of the biggest factories of PSA Peugeot Citroën, the French multinational manufacturer of automobiles and motorcycles. The plant counts on more than 6900 workers and produces 8 different car models, with an average of 1700 vehicles per day. There is a huge assembly line, made up of approximately of 1200 robots and 740 automata, that covers the whole production cycle in its four workshops (stamping, shoeing, painting and assembling); many tasks, such as welding and conveyance, are entirely automated.

The importance acquired by the center is doubtless consequent to its position: in fact, it is situated in the free economic zone of Vigo. Moreover, it is quite close to the port (Figure 1.2), which is one of the most important

³They are realized on PCs linked to the the control system via an “automation network”.

⁴Manufacturing Execution System.

in Spain; about 85% of the automobiles produced in Vigo are exported by sea towards all continents.



Figure 1.2: A top view of Vigo with the PSA Peugeot Citroën plant.

There is a close collaboration between PSA Peugeot Citroën and the University of Vigo, in particular with the Engineering School, as witnessed by the several factory tours and employment opportunities offered to the most deserving students.

1.3 Motivation of the project

A **conveyor system** is an ensemble of mechanical equipments that allows the horizontal and vertical movement of goods from one place to another. The advantage of using a conveyor is given by the possibility of transporting heavy and bulky materials in a quick and safe way, whereas it would be more expensive and risky assigning this task to human workers; that is the reason why conveyor systems are widespread, especially in factories where activities like materials handling, packaging and automotive take place.

This project describes the entire automation procedure for two different kinds of conveyance systems, namely a power & free conveyor and a double level conveyor belt, separately. After a detailed description of both systems under mechanical and electrical perspectives, it proposes the design of an effective strategy that involves various automation fields, such as industrial informatics for networks configuration and control systems design for a functional optimization of working processes. Besides the automation of the considered systems, the project carries out the realization of a complete but intuitive human-machine interface (HMI) for each conveyor, in order to check and control its behaviour by means of a computer connected to the network; according to vertical integration up to the second level of the CIM pyramid, also alarm diagnostic and database management are provided, as it will be explained better later.

The main scope of the work is to design an operating mode that complies with safety rules and guarantees power saving as much as possible: this means implementing solutions that avoid any dangerous situation for the user and the plant, while operating in such a way to smartly manage involved resources.

Another aim is to experience the whole study of an industrial system: compared to a simulation, a real application presents mechanical and electrical aspects that have to match for a correct functioning. On the other hand, the occurrence of working faults drastically slows down the manufacturing process and implies a cost in terms of resources and time due to needed repairs.

An outline of the thesis is as follows. In Chapter 2 the technicalities useful in order to better understand the main part of the project are discussed. A brief overview of programmable logic controllers is necessary, moreover theoretical notions and common proceedings which were used in the automation procedure for both systems are presented. The power & free conveyance system is analyzed in depth in Chapter 3, whilst Chapter 4 treats in detail all the topics regarding the belt conveyor system. These two sections have quite similar structures and may be read independently from each other. Finally, concluding remarks and considerations about open issues and future work are stated in Chapter 5, followed by Appendices.

Chapter 2

An overview of the tools

2.1 Automation with PLC and IEC 61131-3

A **PLC**, namely a Programmable Logic Controller, is a device consisting of both hardware and software components that is of widespread use for the automation of industrial systems, especially those involving electromechanical processes.

A PLC runs a program in a cyclic way and elaborates analog and digital signals that come from sensors and are headed to actuators located in the plant; for this reason, the hardware structure of a PLC contains a CPU¹, an I/O section and an interface for other devices. Due to communication needs, various modules such as serial and Profibus interfaces are planned for connecting PLCs to the most used networks within the industrial field.

Before the introduction of PLCs, relay circuits and wired logic connections were the main tools to implement industrial operations (for example, the opening and closing of a valve). This simple kind of automation was really hardy and reliable in plants with a non-stop production cycle, but it lacked of adaptability in case some changes were needed, because a new wiring had to be done. On the other hand, when using a PLC it is sufficient to replace a few logical instructions in the software in order to modify the functioning of the system, which means a relevant saving in terms of resources and time.

A PLC is a microprocessor device characterized by a cycle time on the order of few milliseconds, hence it could be reasonable to replace it with a personal computer for carrying out automation tasks. However, throughout the years PLCs proved to be preferable to PCs because they are entirely built by the same manufacturer, they better tolerate limit conditions inside factories (for example, electromagnetic interferences, dust, vibrations) and

¹Central Processing Unit.

can be programmed by personnel lacking of specific informatics skills.

Every manufacturer provides a development environment for writing and debugging PLC programs: it consists of an application that can be used on most PCs and laptops. The implementation of a PLC program is split into two phases:

1. **offline** operation: the development environment is not linked to the PLC, so the software programmer can write instructions that have to be processed in order to make the system work properly.
2. **online** operation: the program is loaded into the PLC, which is now connected to the personal computer. During the debug of the program, it is possible to employ dedicated tools that allow to read and force variable values, set break points during its execution and so on: in other words, the actual behaviour of the system under consideration can be evaluated in this phase.

The **IEC 61131** is an international standard for programmable controllers developed by the International Electrotechnical Commission (IEC). Although it was introduced when several manufacturers already followed their own business philosophy, the standard tries to give guidelines with the aim to obtain portability among different products in PLC programming.

At present, IEC 61131 is divided into 9 parts and the first of them are:

1. *General information*;
2. *Equipment requirements and tests*;
3. *Programming languages*.

The third part of IEC 61131 was first published in 1993². It deals with programming languages that follow either graphical or textual standards within PLC automation:

- **Ladder Diagram** (LD), graphical: language based on the circuit diagrams of relay logic hardware;
- **Functional Block Diagram** (FBD), graphical: language with blocks representing logic and arithmetic functions;
- **Structured Text** (ST), textual: high-level programming language provided with *IF-THEN-ELSE* and loop structures;

²The current edition was released in 2013.

- **Instruction List (IL)**, textual: low-level programming language (assembler);
- **Sequential Flow Chart (SFC)**, graphical: flow chart describing parallel and serial sequences.

IEC 61131-3 defines three kinds of **POU** (Program Organization Unit), which are the smallest compilable units within a PLC program. It is possible to distinguish between:

- **Function (FUN)**: it is a time invariant block, as to the same input parameter corresponds the same output at any time instant;
- **Function Block (FB)**: it is a time variant block provided of data record, so it is able to store values and use them in following calls;
- **Program (PROG)**: it allows to access the input and output signals of the device in order to make them available to other POU's.

Every POU consists of a declaration area, where variables of different type may be defined, and a programming area, where the actual program of that block can be found; a POU can “*call*” other POU's within its execution.

Due to the vastness of the topic, this thesis purposely does not treat in depth the IEC 61131-3 standard regarding the programming of industrial automation systems. However, [4] provides a complete and detailed guide that turned out to be very useful throughout the development of the project, with in mind the idea of respecting the above mentioned standard.

The PLC programs for the automation of the conveyance systems under consideration were implemented according to IEC 61131-3 standard; more in detail, Structured Text (ST) was chosen as programming language.

2.2 Beckhoff technology for automation

Beckhoff Automation is a German company involved in open automation systems that offers PC-based control technology [6]; it covers a wide range of products from industrial PCs to fieldbus components and dedicated software. The use of Beckhoff technologies guarantees universal and suitable solutions for automation tasks taking into account functional complexity, computing performance and costs; for this reason, in many cases Beckhoff products are a valuable and effective choice.

TwinCAT (The Windows Control and Automation Technology) is a package for automation that turns any compatible PC into a real-time controller with a multi-PLC system and development environments for configuration, programming and diagnostic, also providing connection to all

common fieldbuses and interfaces support.

The **ADS** (Automation Device Specification) communication protocol is a transport layer within Beckhoff TwinCAT system that permits data exchange among different software modules. It is used on top of TCP/IP protocol, namely in a networked environment all the data is accessible from any point. As a matter of fact, the user within the Beckhoff system may use any connecting route in order to connect to other linked devices and to configure them. To this end, more information is presented in Appendix A.

TwinCAT **System Manager** is the application dedicated to the configuration of the system as it carries out the proper connection to I/O devices; it supports most commonly used industrial fieldbuses such as Profibus, Profinet, Ethernet and CANopen. As it is shown in Figure 2.1, *System Manager* manages data mappings between the components allowing bit-based links; moreover, its environment allows the user to read and force the values of task variables and I/O devices.

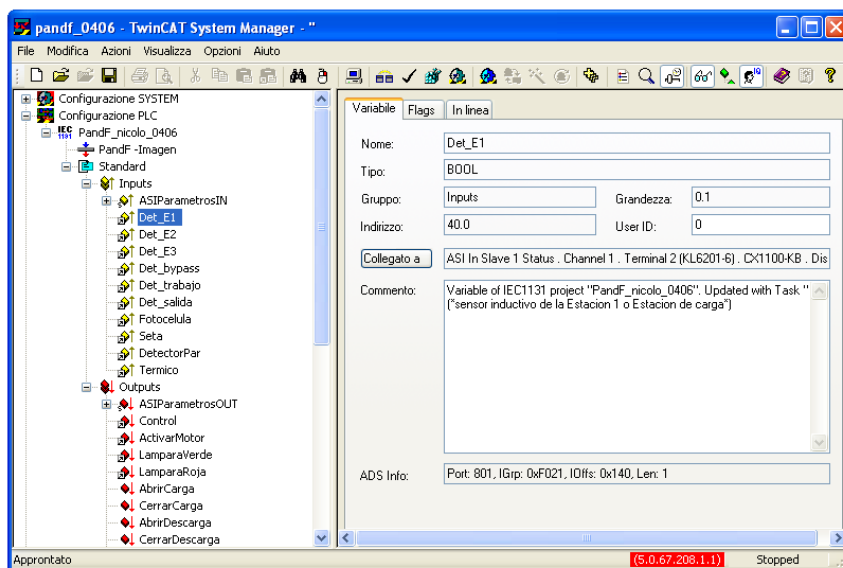


Figure 2.1: A screenshot of TwinCAT *System Manager*.

TwinCAT **PLC Control** is the main automation software, in fact it is the environment in which PLC programs are written and debugged. An example is given in Figure 2.2, which shows the list of POU's on the left part of the page, whereas the right one is dedicated to the definition of variables and instructions for the selected functional block. At the bottom of the page there are two text bars which are coloured to make the user aware of the state of the connection with the PLC; the first bar turns red once the

connection is established and prints on the display the linked device with its IP address and port used, while the second bar is blue within *Config Mode* (“offline”) and becomes green if “online” mode is activated.

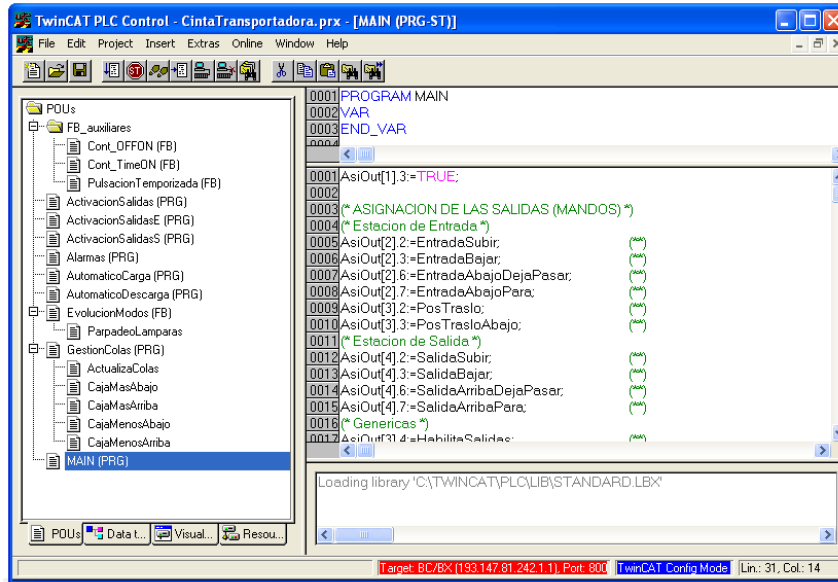


Figure 2.2: A screenshot of TwinCAT *PLC Control*.

PLC Control is based on **Codesys**³, which is a development environment employed in applications for programming controllers that complies with the international industrial standard IEC 61131-3.

2.3 Odette standard for barcodes

Odette is a European standard used within automotive industry and developed by the *Organisation for Data Exchange by Tele-Transmission in Europe*⁴; its introduction was due to the need of manufacturers for reducing the amount of paper documents exchanged every day.

Odette standard allowed a rapid development in information and communication technology (ICT), in fact it merged in a single label the whole data regarding purchases and deliveries of automotive industry: receiver, supplier and product characteristics (serial number, quantity, weight, logistic reference, etc.).

Odette standard employs **Code 39**, which is a discrete barcode symbolology of variable length; the desired text is written between start and stop

³It is the acronym for “controller development system”.

⁴So, ODETTE represents the acronym for the organisation founded in 1984.

delimiters both represented by the same character “*”, as shown in Figure 2.3.

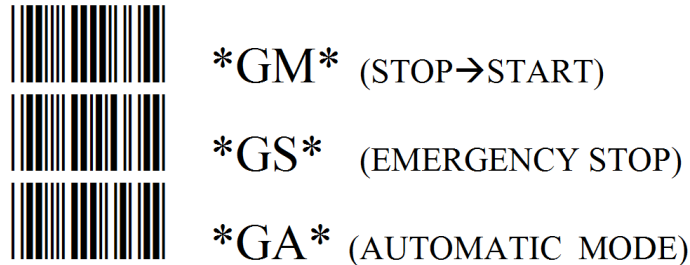


Figure 2.3: An example of Code 39 used throughout the project.

As already mentioned, many fields containing different information are required within a transport label that complies with Odette standard. For this reason, every field is characterized by a barcode with its own prefix character that distinguishes the data and reduces the possibility of errors in the reading.

With the goal of governing the system behaviour according to Odette standard, a cordless barcode scanner Datalogic Gryphon M100 was employed. As shown in Figure 2.4, a command paper with codes for manual operations was printed in order to control the functioning of the system from a remote position.

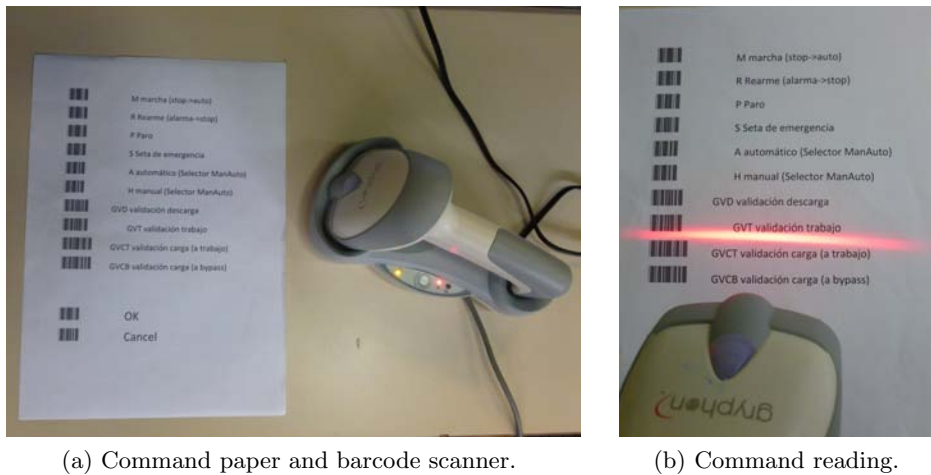


Figure 2.4: The cordless barcode reader used throughout the project.

In other words, this tool allows to carry out tasks such as validating boxes or activating outputs even if the human operator is not close to the computer or the installation, with the only restriction to not exceed 15m of distance

from the base station. For example, using a barcode scanner gives the opportunity to realize a detailed start up while the worker stands within the installation, thus checking by himself/herself the correct functioning of its individual components.

The correct operation of the barcode scanner can be checked with *HyperTerminal*, a software from which it is possible to select a device connected through a port of the computer and then, after choosing the speed rate of the transmission⁵, verify on the display if the read value is the desired one. The barcode scanner was used in order to control both power & free conveyor and double level conveyor belt. The prefix character selected to distinguish the manual commands is “G”, as it was indicated in Figure 2.3, which means *General* according to Odette standard; for the sake of completeness, “P” and “S” indicate *Part number* (reference) and *Serial number*, respectively.

2.4 Petri nets

A **Petri net**⁶ (or **place/transition net**) is a mathematical representation for a discrete distributed system. It consists in a directed bipartite graph defined as a triple $\mathcal{N} := (P, T, \mathcal{E})$, where P, T are disjoint finite sets of nodes and $\mathcal{E} \subseteq (P, T) \cup (T, P)$ is the set of arcs. A Petri net is a state/transition system and for this reason it has two kinds of nodes, namely:

- **places**, which indicate particular conditions or states of the system and are denoted by circles;
- **transitions**, which describe events that could occur (in general, the start and end of various processing steps) and are identified by bars.

Every arc of the graph links a place to a transition (*PT arc*, which leaves an *input place*) or a transition to a place (*TP arc*, which enters an *output place*), whereas connections between nodes of the same type are forbidden; this simple rule derives from previous definitions. Under a graphical point of view, places may be marked by a **token** in order to point out the system state; the number associated to the token in a place shows the overall processing capacity. Any distribution of the tokens over places is a configuration of the net called **marking**. A Petri net can represent an infinite set of discrete states by means of a finite number of places. Tokens can cross a transition according to a rule named **firing**: the transition may fire if it is enabled, meaning that at least one PT arc connected to it is marked with a token. When a transition fires, it consumes the required tokens appearing

⁵In this case, it was equal to 9600 *bit/s*.

⁶The name is due to its inventor, the German mathematician Carl Adam Petri.

in input places and creates tokens in its output places; Figure 2.5 shows an example of Petri net representing a simple system whose behaviour is characterized by two states.

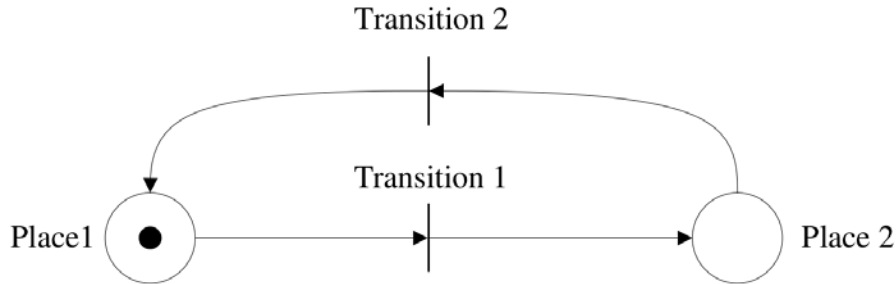


Figure 2.5: A simple Petri net with 2 places, 2 transitions and 1 token.

As implicitly mentioned above, Petri nets are more sophisticated graphs than only alternating sequences of single places and transitions. As a matter of fact, they may be quite complex and present peculiar features that are specified hereinafter:

- **decision** (exclusive executions): a place may present more than one PT arc which enable different transitions, thus the behaviour of the system depends on which branch of the Petri net is followed at a given time;
- **alternative junction**: it corresponds to the place where exclusive executions (namely, different branches) coming from a decision converge, so various TPs arcs may fire to the same place;
- **concurrency** (simultaneous executions): if two or more places receive TP arcs coming from the same transition⁷, then these states make up a parallel structure that arises from a single event;
- **synchronization**: it closes a structure of simultaneous executions, hence when two or more different processes of the system end, they enable a unique transition that can fire in order to let the system continue its operation;
- **mutual exclusion**: within a system there could be a **shared resource**, namely a resource that is accessed by more components or used in different states of the operation. Thus, besides places and transitions, the evolution of states requires the availability of the shared resource, which is requested, used and then given back when needed.

⁷Namely, the TP arc is replicated a few times.

It must be underlined that Petri nets present a non-deterministic behaviour: in other words, if more transitions are enabled at the same time then any of them may fire. Obviously, the previous statement holds only from a theoretical point of view, since a computer algorithm gives priority to the first instruction of the program when various decisions are possible.

In the remainder of the thesis, Petri nets are used with the aim to graphically describe the evolution of operating modes and the automation process of both systems under consideration; from now on, the assumption will be made that if a decision occurs, then only one transition may fire at a time, as all conditions enabling transitions will be exclusive.

2.5 Electrical equipment

A **sensor** is a device that can detect the happening of events within the environment around it: when a change occurs, it changes its output signal in order to transmit such an information.

Three kinds of sensor are used in next chapters:

- **inductive** sensors can detect the proximity of a metal object thanks to the ability to change their own magnetic fields: the proximity of the metal has the effect of reducing the magnetic field of the inductive sensor meanwhile its current raises until it activates a voltage signal by means of a transistor.



Figure 2.6: An inductive sensor of the power & free conveyor.

- **optical** sensors (photocells) are devices made up of a light emitter and a light receiver: when a body interrupts the stream of light from the emitter to the receiver then the photocell changes its output signal in order to communicate that its visual field has been interrupted due to some obstacle.



Figure 2.7: A photocell of the double level conveyor belt.

- **limit switches** are electromechanical devices constituted by an actuator physically linked to a few contacts: if an object touches the actuator then the limit switch creates or breaks an electrical connection by means of its contacts.



Figure 2.8: A limit switch of the power & free conveyor.

An **actuator** is a component responsible for moving or governing a component of a system; a simple example is given by an electric motor.

Pneumatic actuators, which are used in both conveyance systems under consideration, turn energy generated by compressed air at high pressure into either linear or rotational motion: they enable large forces that are used with solenoid valves.



(a) Solenoid valves.



(b) An air cylinder.

Figure 2.9: The components of a pneumatic actuator.

2.6 The AS-i network

Both conveyance systems under consideration employ Actuator Sensor Interface [7], also known as **AS-i**, that is an industrial fieldbus mostly used for the communication with signals related to process level⁸.

An AS-i network is featured by a control system (**master**) which is in charge of consulting and updating data coming from all modules (**slaves**) connected to the network at a certain time. The master starts communicating with a slave to exchange data and requires the slave to respond within a maximum time, thus ensuring a deterministic networking solution. In order to satisfy imposed speeds, only limited pieces of information are exchanged between the master and its slaves; at every PLC cycle, it is possible to swap at most 4 input bits and 4 output bits for each slave.

In its original version an AS-i network could support up to 31 slaves, in other words up to 124 inputs and 124 outputs were controlled; nowadays, thanks to newer versions, it is possible to manage 62 devices meanwhile granting a maximum update time of *10ms*.

Most AS-i networks employ the AS-i flat cable, which realizes fast and safe connections by means of piercing technology: proper needles penetrate the cable jacket and displace the internal copper strands without cutting them. Among all possible kinds of coloured cables available, the yellow flat one is usually chosen to power AS-i modules and enable data communication within the network; in this project a Siemens 3RX9010-0AA00 was used.

In this thesis the considered AS-i master terminal is a Beckhoff KL6201 [8], whilst three different types of Siemens slaves are used:

- a slave with two digital inputs and two pneumatic solenoid valves as outputs: it is the most used slave within the systems under consideration because of the need for actuating air cylinders;
- a control slave with two inputs and two outputs: the inputs are a button and an emergency stop whilst the outputs are two indicating lights;
- a slave with four inputs and four outputs: it is a module that lies in the control cabinet of both systems, it needs its own power supply in order to control the outputs.

The process image⁹ of the KL6201 requires 12 bytes: 6 of them are dedicated to the AS-i parameter interface, while the remaining 6 bytes constitute the

⁸It is the lower field level according to the CIM pyramid.

⁹In this contexts, it consists in a copy of the system state at a given time; it can be used for backup.

AS-i process data. Thus, it is possible to manage only 11 slaves within the network under consideration since the lower half of the first byte (*Byte 1*) presents bits that are reserved to information about the communication status. The ordered way in which slaves are matched to input and output bytes of the process image is depicted in Table 2.1. The higher half of the *Byte 1* are used to save the four bits coming from the slave with address 1; if *Byte 2* is considered, then the four lower bits are related to the slave with address 2 whereas the four higher bits refer to the slave with address 3; and so on for subsequent bytes until *Byte 6*.

	SLAVE $N + 1$				SLAVE N			
BYTE N	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IN	In4	In3	In2	In1	In4	In3	In2	In1
OUT	Out4	Out3	Out2	Out1	Out4	Out3	Out2	Out1

Table 2.1: Mapping of a byte (of neither inputs or outputs) within the AS-i process image.

Before connecting slaves to the AS-i network, it is important to configure each of them with the aim to make it detectable within the network: hence, an unambiguous address consisting of a natural number between 1 and 31 is associated to every slave. To this goal, the MetraTest 36ASi System Tester indicated in Figure 2.10 was used to carry out the correct routing of the network; the connection to slaves employed a flat AS-i cable and a M12 connector.



Figure 2.10: The MetraTest AS-i System tester connected to a slave.

2.7 Operating modes

Any system may work in different conditions based on safety restrictions and desired purposes.

The standard *UNE EN 60204-1 - Safety of Machinery - Electrical Equipment of Machines - Part 1: General Requirements* deals with the state development for industrial machinery in its section 9.2.3. According to it, the following operating modes are taken into account with the aim of designing Petri nets that describe the overall behaviour and help to carry out the automation of the system:

- **0** → **stop** mode: it corresponds to the initial state of the system, from which it is possible to reach all other operating modes if proper conditions are satisfied. The system remains in a state of rest, namely no output signals can be controlled;
- **1** → **automatic** mode: as it is indicated in the above mentioned standard, this mode is selected by setting to 1 the boolean value M/A and it gets activated by a *Start* signal. When the mode is active, the system works in an automatic way that may be totally independent from human users, although a few manual actions may be planned for a better management of the process;
- **2** → **manual** mode: it is enabled under certain conditions if the bit M/A is equal to 0 when the *Start* signal presents a positive edge. In such a case, all outputs are controlled by a human operator and to this end many control buttons are needed: although manual mode counts on safety precautions that are implemented within the PLC program, the user has to be conscious of his actions and hence he needs to know the installation;
- **3** → **alarm** mode: it is necessary in order to check emergencies that have been detected by the controller (after having received a signal from a sensor or after some maximum waiting time has expired) or by human users. The only way to exit this situation consists in solving issues and then restarting the normal operation by an acknowledgment signal.

Any element within a system has to comply with the above mentioned modes, since they are more general and relevant behaviours than the features of specific elements of the system. For instance, if a system has an elevator that performs an ordered sequence of tasks, then the state evolution of the Petri net associated to the elevator depends on the active operating mode; hence, when the system remains in stop mode no actions of the elevator are allowed.

Moreover, to each operating mode one of two indicating lights is associated in the following way:

- stop: steady red light;
- automatic: steady green light;
- manual: blinking green light;
- alarm: blinking red light.

Unless different assumptions are explicitly stated, in the remainder of the thesis every conveyor under consideration will comply with the Petri net depicted in Figure 2.11 for what regards the changes in the modes used during its functioning.

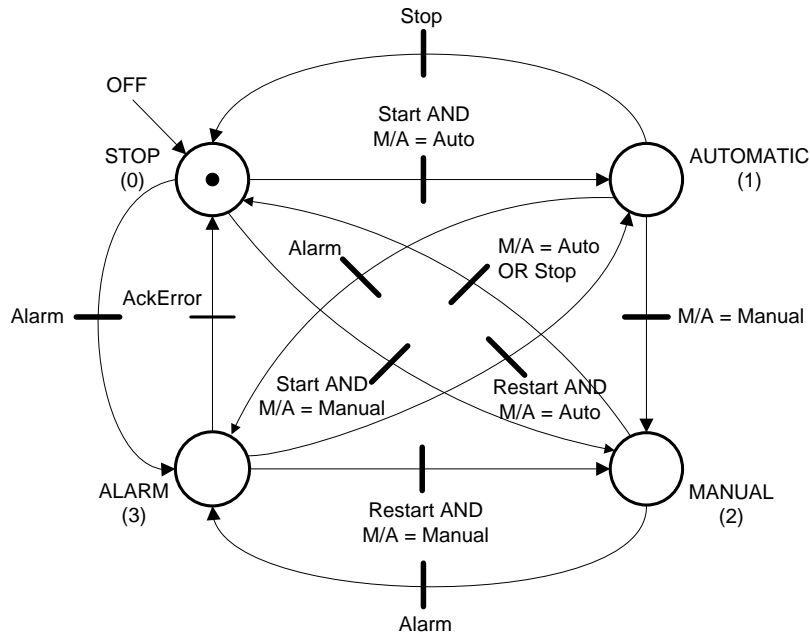


Figure 2.11: Petri net for the evolution of operating mode based on UNE EN 60204-1.

2.8 Microsoft Visual Studio

Visual Studio is an integrated development environment (IDE) that currently supports various programming languages such as *C*, *C++* and *Visual Basic .Net*; it is owned by Microsoft. Due to its nature of multiplatform software, *Visual Studio* allows the realization of programs for browsers, servers,

workstations and smartphones. Its code editor supports *IntelliSense* technology, which can detect syntax and logic errors before compiling the code, in addition to the presence of a code completion component that helps the programmer while writing a program.

Herein, *Visual Studio* 2010 was used in order to realize intuitive but powerful human-machine interfaces for data storage and remote management of the power & free conveyor and the double level conveyor belt; these applications, that do not require any specific software except for JRE¹⁰ and are totally portable, were linked to the relative controller by means of an Ethernet connection.

¹⁰Java Runtime Environment.

Chapter 3

The power & free conveyor

3.1 About power & free conveyors

A **chain conveyor** is a mechanical system that carries out the transportation of several pendant pallets through production lines via a chain arrangement; once the framework is driven by an electric motor, the chain starts moving forward in a continuous loop and the materials contained in pallets are moved along the factory because of its joining to the pulling system.

Chain conveyors are mainly used in environments where there are heavy unit loads, for example in automotive, metal finishing and distribution industries; they can manage both horizontal and vertical changes of direction and require low maintenance, moreover keeping products suspended improve performances in processing activities such as painting and coating.

A **power & free conveyor** is a particular kind of chain conveyor that provides a means for advancing products that are contained in carts or pallets through a manufacturing system: these carriers are *free* to move, stop or accumulate along a continuously moving conveyor (the *power* supplier). Usually, assembly lines require product conveyance to or between “islands of automation” and this is the role of power & free conveyors, whose pulling system keeps running even if one or more carriers stopped at different points for inspection, loading or unloading tasks.

3.2 Description of the system

The power & free conveyor located inside the “Ricardo Marin” laboratory is an overhead double rail conveyor that is structured in two different levels (Figures 3.1 and 3.2) of almost equal length; it has hook carts that can move along the track, thanks to a pulling system ruled by an electric motor.

Since the power & free conveyor has a framework of considerable size, it could be quite hard to understand its mechanical characteristics with a



Figure 3.1: Higher floor (with a path deviation) in the area of power & free conveyor.



Figure 3.2: Lower floor in the area of power & free conveyor.

simple description; luckily, besides the pictures that are herein reported, the reader can find in Appendix B various technical drawings for the whole system and a few components of interest.

Back to the subject, the section of the system is made of two overlying tracks whose mutual distance is 27.5cm : the higher rail is used only by the pulling system while on the lower one hook carts that transport goods can move along the installation. Figure 3.3 (a) shows a hook used in order to

carry carts such as the one represented in (b); it is possible to note the overlying tracks in their steep stretch, too.

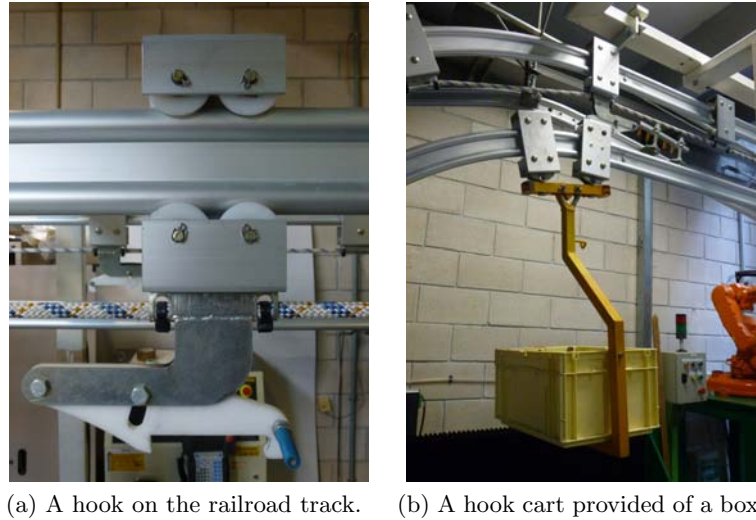


Figure 3.3: Two essential components of the power & free conveyor.

The pulling system is very simple as it consists of an electric motor that controls the drawing of carts which are connected by means of industrial-strength ropes¹. The whole track consists of a circuit closed by the U-turns that are indicated in Figure 3.4: the one at the higher floor accommodates a pulling wheel that is driven by the electric motor in order to generate the motion of the system, whilst the second one is made up of various nylon rollers that allow to keep tension on the ropes.

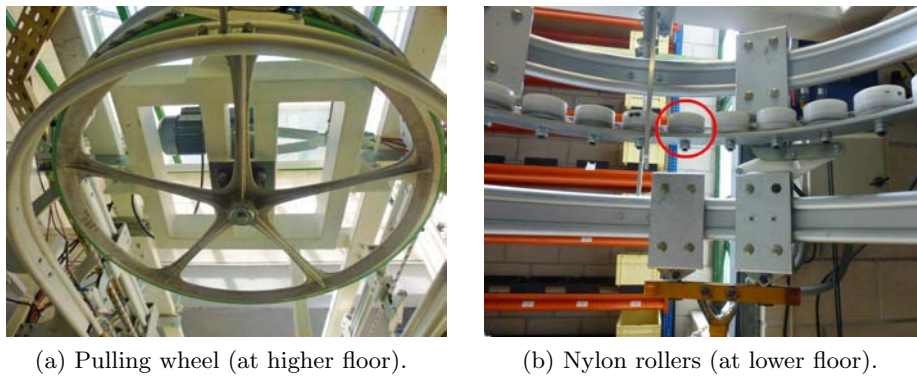
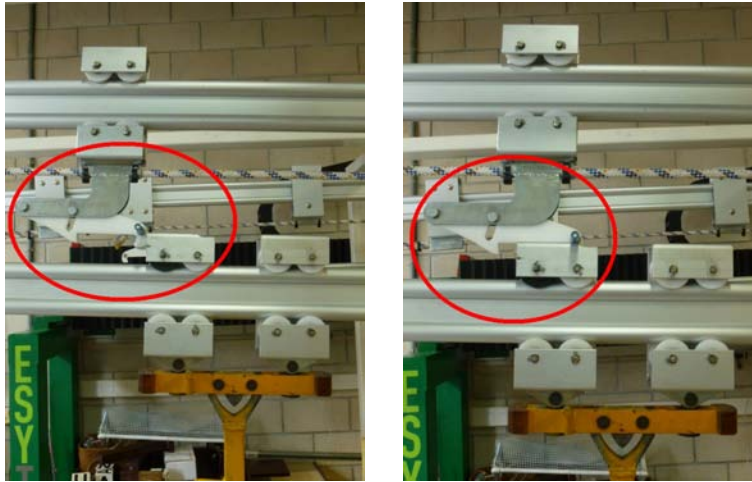


Figure 3.4: Ends (U-turns) of the power & free conveyor.

To allow a better comprehension of the pulling system, Figure 3.5 represents how carts get hooked to the pulling system once it starts running.

¹Although more often this connection is made by chains.



(a) Cart free from the pulling system.

(b) Cart hooked by the pulling system.

Figure 3.5: Behaviour of pulling hooks in the power & free conveyor.

Note: The framework of mechanical components within the power & free conveyor has a relevant role in the functioning of the system: indeed, smart mechanisms are used to realize the matching between the pulling system and carts and to free carts when they are stopped at the stations, too. This mechanical concepts are worth to be described; however such a description would be quite lengthy and out of the scope of the thesis, so related information can be found in the recent projects [2] and [3].

The hook carts are not constrained to follow a unique track on their rail; as a matter of fact, there are two railroad switches (Figure 3.6) that generate a deviation from the standard path, as it will be explained later.

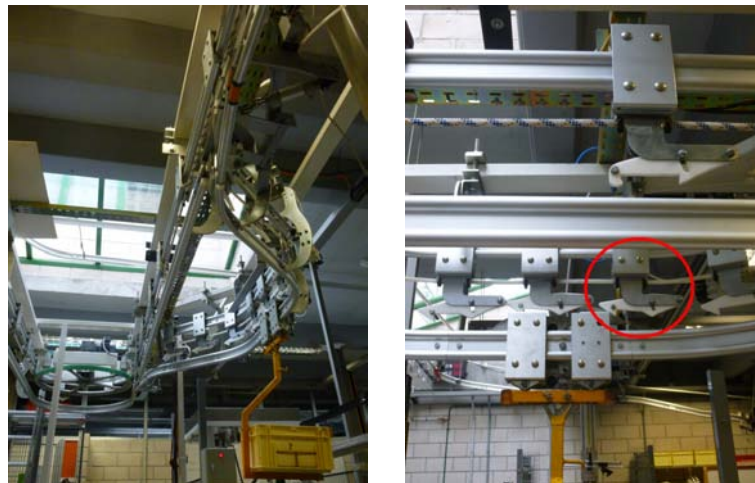


(a) 1st switch (Entrance of the deviation).

(b) 2nd switch (Exit of the deviation).

Figure 3.6: Railroad switches feature the path deviation.

The area that is crossed by the deviation from the usual track, that from now on will be called **working area** (Figure 3.7(a)), needs its own pulling system as it is impossible to extend the above mentioned durable ropes. Thus, within the working area an air cylinder is responsible for controlling the motion of carts along the track, which is provided with many hooks on its upper rail (as shown in Figure 3.7(b)); these hooks are able to move back and forth with the aim to hook carts and drag them forward. However, because of physical constraints, at most 3 carts should stay inside this area at the same time in order to avoid undesired consequences.



(a) The working area and its rail-road switches.

(b) Differences between pulling systems.

Figure 3.7: The deviation from the standard path.

3.3 Stations

The power & free conveyor under consideration consists of three main areas that are stop points for carts. They are called **stations** and each of them has different characteristics:

- **loading station:** it is located at the higher floor and it manages the load of raw goods of various kinds (coming from another conveyor) that need to be classified and then sent to the following stop points, depending on the position of the 1st railroad switch;
- **working station:** it lies in the working area between the railroad switches and it consists of a deviation that extends the path crossed by carts. It is a point where products are subjected to some processing or even quality control by a human worker;

- **unloading station:** it is placed at the lower floor and, as the name suggests, it is the destination of products being carried inside carts; they are picked up manually or by a robot in order to be transferred to the warehouse via the belt conveyor system.

The stations of the conveyor are classified with respect to their behaviour as follows:

- **type 1:** the station features a simple pneumatic system of closure and opening that allows it to stop incoming carts or let one of them pass;
- **type 2:** the station, besides the opening mechanism, is equipped with tongs that block the cart once it arrives and hold it until the start of the opening command.

The loading and working stations are of *type 1*, instead the unloading one is a *type 2* station (Figure 3.8); tongs are used to ensure that goods which are waiting to be picked up from carts will not be damaged by subsequent carts entering the unloading area.

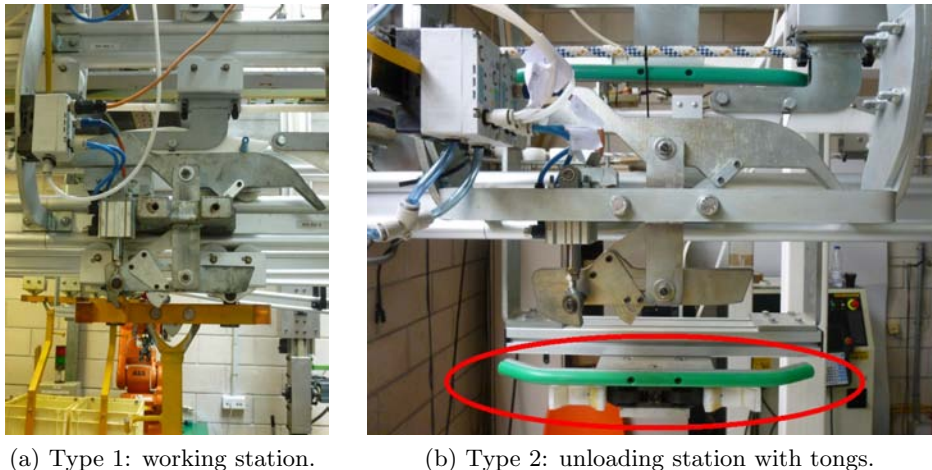


Figure 3.8: Differences between types in the stations of the power & free conveyor.

According to the previous classification of station types, these states distinguish the behaviour of the stop points during a work cycle:

1. *Waiting for cart arrival:* the station is closed and no carts are in it;
2. *Locking the cart* (only for the unloading station): the station stops a cart and closes its tongs around the hook;
3. *Waiting for cart validation:* the station has stopped a cart and is waiting for a validation before freeing the cart;

4. *Unlocking the cart* (only for the unloading station): after the validation, the station unlocks the cart in it by releasing tongs;
5. *Waiting for cart exit*: the station is open and the cart can leave it and proceed along the installation.

3.4 Electrical equipment

In this section the electrical components that feature the power & free conveyor are described. They allow the control of its operation according to the needed specifications; firstly the electric motor and sensors and actuators are presented, then a description of the controller and its modules is given.

3.4.1 Electric motor

The pulling system of the power & free conveyor is driven by means of a SEW-EURODRIVE three-phase alternating current motor that supplies an electric power equal to $0.15kW$. As already shown in Figure 3.4, the electric motor lies on the top of the conveyor U-turn at the higher floor; its axis is matched to a horizontal wheel that produces the sliding of the rope to which tow hooks are fixed.

3.4.2 Sensors

The power & free conveyor has various inductive sensors to reveal the passing or arrival of carts in certain points of interest. These and other sensors are indicated in Table 3.1 along with a brief description that identifies the position of each of them.

Description	Task
Inductive sensor	Detection of a cart stuck at the loading station
Inductive sensor	Detection of a cart crossing the bypass path over the 1 st railroad switch
Inductive sensor	Detection of a cart entering the working area over the 1 st railroad switch
Inductive sensor	Detection of a cart stuck at the working station
Inductive sensor	Detection of a cart over the 2 nd railroad switch
Inductive sensor	Detection of a cart stuck at the unloading station
Optical sensor	Detection of a jam after the loading station
Limit switch	Detection of an excessive torque in the electric motor

Table 3.1: Sensors of the power & free conveyor.

3.4.3 Actuators

The system under consideration needs various pneumatic actuators in order to work properly: they are air cylinders (which are matched to solenoid valves) with different characteristics that are used for controlling actions within stations and setting path configurations on the railroad switches. Table 3.2 gathers all them: it should be remarked that they have different stroke lengths due to the specific goal of each air cylinder.

Description	Task
Air cylinder (stroke of 25mm)	Opening of the loading station
Air cylinder (stroke of 25mm)	Opening of the working station
Air cylinder (stroke of 25mm)	Opening of the unloading station
Air cylinder (stroke of 100mm)	Cart locking in the unloading station
Air cylinder (stroke of 150mm)	Control of the 1 st railroad switch
Air cylinder (stroke of 150mm)	Control of the 2 nd railroad switch
Air cylinder (stroke of 250mm)	Pulling system of the working area

Table 3.2: Pneumatic actuators of the power & free conveyor.

3.4.4 PLC modules

The power & free conveyor was automated by means of a Beckhoff CX1000 Embedded PC [10], namely a modular control device that satisfies tasks requiring features and computing capacity of industrial PCs but at a lower cost; it can be operated without display or keyboard (“headless” mode) and it is equipped with Ethernet and RS-232 built-in interfaces.

CX1000 shows to be a powerful PLC complying with IEC 61131-3 standard once it is combined with the TwinCAT automation software: within Windows CE², programs written in high-level languages can be executed in real-time in parallel with TwinCAT. A CX1000 unit is programmed via a laptop or a desktop PC, which is connected to the controller via Ethernet network, and then loaded into the desired system.

PLCs can be integrated with modules for both digital and analog I/O signals or even for communication with the plant (e.g. Profibus, AS-i, Ethernet); the PLC and other devices, such as the AS-i network, need to be connected to power supply in order to work. The following modules, which are also depicted in Figure 3.9, constitute the smart equipment of the considered system:

- Siemens SITOP power 2 (6EP1331-1SL11): power source of at most 24V and 5A;

²It is an operating system subfamily provided with real-time capability.

- Siemens AS-i POWER (3RX9 502-0BA00): power supply for the AS-i interface of at most 30V and 5A;
- Beckhoff CX1000-N002: RS-232 communication module;
- Beckhoff CX1000-N001: communication module for programming PC (2 USB ports, 1 VGA input);
- Beckhoff CX1000-N000: access to network and RS-232 communication module;
- Beckhoff CX1000-0112: CPU of the programmable logic controller;
- Beckhoff CX1000-COOL: cooling system;
- Beckhoff CX1100-0002: K-BUS, As-i communication module;
- Beckhoff KL6201: master terminal of the AS-i bus;
- Beckhoff KL9010: bus end terminal.

Both the power sources, SITOP power 2 and AS-i POWER, are supplied by the single phase line of 230V; all the above mentioned modules and the controller lie together on a rack located upon a table close to the installation.

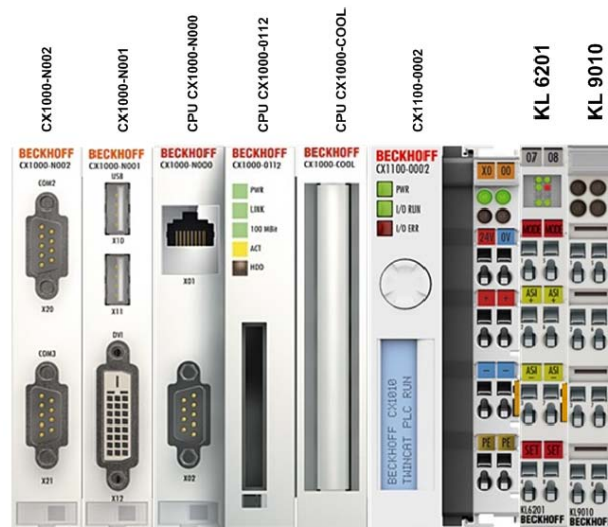


Figure 3.9: Scheme of the controller and its modules.

The TwinCAT environments in the PC and the controller of the system are connected through an Ethernet cable; the address of the CX1000 was set to 193.147.81.217, while its identifier according to the ADS protocol was 5.0.67.208.1.1.

3.5 Network configuration

In Section 2.6 it is described a procedure that allow to carry out the correct addressing of slaves within an AS-i network; it was used for mapping both input and output signals that characterize the operation of the system under consideration.

For the sake of completeness, the resulting maps are indicated in Tables 3.3 and 3.4.

Addr.	Location	Identification of the INPUTS
0	System data	-
1	Loading station	IN1 - Not connected IN2 - Detector of loading station
2	Unloading robot	IN1 - Not used IN2 - Not used
3	1 st railroad switch	IN1 - Detector entrance in bypass area IN2 - Detector entrance in working area
4	Working station	IN1 - Photocell IN2 - Detector of working station
5	2 nd railroad switch	IN1 - Not connected IN2 - Detector entrance in unloading area
6	Unloading station	IN1 - Not connected IN2 - Detector of unloading station
7	Unloading station	IN1 - Not connected IN2 - Not connected
8	Control cabinet	IN1 - Thermal switch IN2 - Not connected IN3 - Not connected IN4 - Torque detector
9	Beacon	IN1 - Emergency stop IN2 - Not connected IN3 - Button
10	Air cylinder	IN1 - Not connected IN2 - Not connected
11	Overhead conveyor	IN1 - Not used IN2 - Not used
12	Loading robot	IN1 - Not used IN2 - Not used

Table 3.3: Mapping of the input AS-i slaves in the power & free conveyor.

Addr.	Location	Identification of the OUTPUTS
0	System data	-
1	Loading station	OUT3 - Opening loading station OUT4 - Closing loading station
2	Unloading robot	OUT3 - Not used OUT4 - Not used
3	1 st railroad switch	OUT3 - Switch on working path OUT4 - Switch on bypass path
4	Working station	OUT3 - Opening working station OUT4 - Closing working station
5	2 nd railroad switch	OUT3 - Switch on bypass path OUT4 - Switch on working path
6	Unloading station	OUT3 - Opening unloading station OUT4 - Closing unloading station
7	Unloading station	OUT3 - Locking hook OUT4 - Unlocking hook
8	Control cabinet	OUT1 - Motor OUT2 - Not connected OUT3 - Not connected OUT4 - Not connected
9	Beacon	OUT1 - Not connected OUT2 - Red light OUT3 - Green light
10	Air cylinder	OUT3 - Forward motion OUT4 - Backward motion
11	Overhead conveyor	OUT3 - Not used OUT4 - Not used
12	Loading robot	OUT3 - Not used OUT4 - Not used

Table 3.4: Mapping of the output AS-i slaves in the power & free conveyor.

3.6 Description of the automation procedure

This section addresses the main topic of the project, namely the procedure to make the power & free conveyor automated, in order to make it operate at its best: hence, all the strategies described in the following descend from considerations regarding issues such as the safety of the plant and power saving.

The Petri nets that represent the operations taking place in every station are indicated in Figures 3.10, 3.11 and 3.12; the automation procedure is based on these diagrams, which are explained separately.

For a better understanding of what follows, it is convenient to point out that the two railroad switches of the power & free conveyor represent shared resources which are requested from both the loading and the working station; thus, an efficient strategy for the automation of the system needs to optimize the management of these components. Herein, a railroad switch results *available* if it has no requests for changing its position; conversely, it becomes *busy* after a station requests the transit of one cart across it.

3.6.1 Operation of the loading station

The loading station, or Station 1, carries out the task of loading pieces or products into carts that will be carried along the installation. From a programming perspective, it is the most complex station because it is responsible for managing the path that every cart has to cross. The destination of a cart depends on which validation is selected: indeed, carts may be sent either to the working station (long path) or directly to the unloading station (short, or *bypass* path).

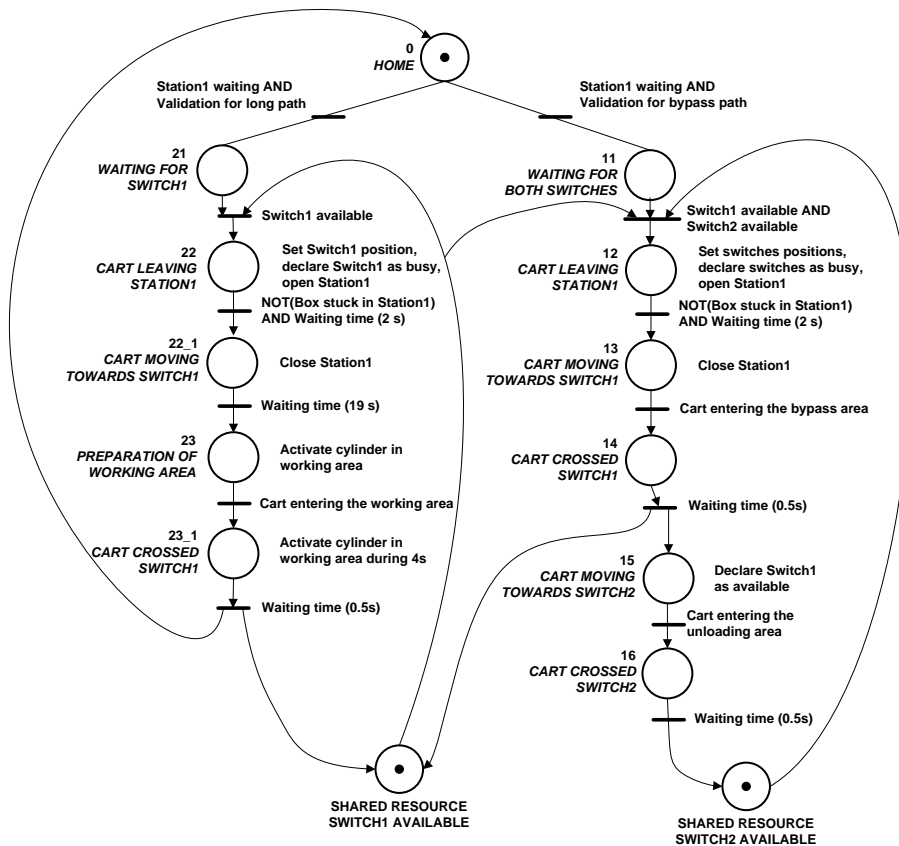


Figure 3.10: Petri net for the loading station of the power & free conveyor.

As it is shown in Figure 3.10, when Station 1 is closed and there is a cart that waits for being validated in it, then two different evolutions are possible depending on which path is chosen.

- **Long path:** if the cart stuck in Station 1 receives a validation signal for being sent to the working area, then the system begins to follow the left branch of the related Petri net by waiting till the first railroad switch is available, since later the working station will be in charge of managing the second one. When the first railroad switch is available, the system sets it in order to deviate the path and declares it busy so that nobody else can take it until it becomes available again. Station 1 activates its opening mechanism, hence the cart in it can start moving by means of the pulling system to which it is hooked. If the cart is no more detected by the inductive sensor of Station 1, it takes an additional time of $0.5s$ before closing Station 1, to let a possible subsequent cart enter it. The cart proceeds and after $19s$, when it is expected to be close to the first railroad switch, the working area starts operations to welcome the cart: in detail, the pulling system of such an area begins to move forward and backward during brief time intervals with the aim to hook carts along that stretch of path and carry them to the working station. When the cart enters the working area, the air cylinder keeps functioning for additional $4s$ thus all carts in the trail can be piled up just in front of the working station (in fact, only one cart at a time can enter it). After the sensor situated just over the first railroad switch detects the incoming cart, a waiting time of $0.5s$ is required before declaring available the shared resource in such a way that another cart can leave Station 1 in order to be sent to the working area³.
- **Short path:** if the cart stuck in Station 1 receives a validation signal for being sent to the unloading station, then the system begins to follow the right branch of the related Petri net by waiting till both railroad switches are available, as the cart needs both of them along the track to reach the unloading station. If the railroad switches are available, then the system sets their positions on the bypass path and sets their status as busy so that nobody can use them until they will become available again. Station 1 opens, so that the cart can leave it. When the cart is no more detected at the stop point, a countdown of $2s$ starts at the end of which Station 1 can accommodate again possible subsequent carts. So, the cart is carried by the pulling system till the first railroad switch; when the inductive sensor over it reveals the passage of the cart through the entrance of the bypass area, then a

³On the other hand, a validation signal for a short path would require also the second railroad switch, which now may be busy because of a request from the working station.

0.5s time delay is sufficient to make the first railroad switch available again (namely, that shared resource is given back). This arrangement allows a cart heading to the working station to start its path even if the previous cart has not reached the unloading station yet, because the second railroad switch is not needed in this case. Once the cart crosses the second railroad switch and enters the unloading area, then after a waiting time of 0.5s the second railroad switch becomes available (this shared resource is returned, too): the cart can proceed to the unloading station and a new instance of the short path may be activated from the initial state of the Petri net if there is another cart in Station 1.

3.6.2 Operation of the working station

The working station, also known as Station 2, is responsible for holding carts that come from the loading station and contain products which need to be somehow processed or subjected to quality control procedures by a human worker. Station 2 behaves in a simpler way than the loading station, even though they need to be collaborative in order to share the second railroad switch: indeed, carts coming out of both the working area and the bypass area require it along the track to reach the unloading area.

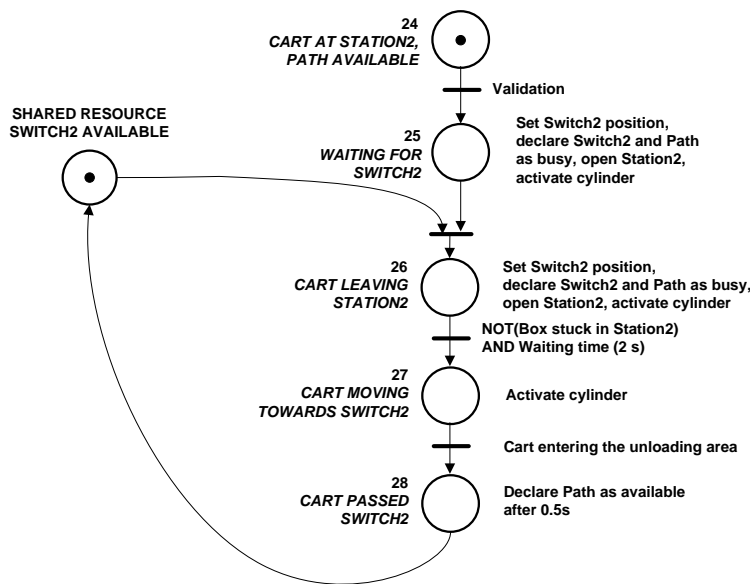


Figure 3.11: Petri net for the working station of the power & free conveyor.

If there is a cart stuck in Station 2 and the stretch of path over the stop point is not occupied, then a validation signal let that cart proceed to the unloading station. When the second railroad switch is available, namely the loading station is not keeping it busy, Station 2 makes it physically con-

nected with the deviation trail, sets as busy both the second railroad switch and the last half of trail within the working area, activates the pulling system ruled by the air cylinder and starts to open; hence, the cart can leave it and once it is no longer detected by the inductive sensor, Station 2 is closed after a time interval of 2s. The movement of the air cylinder makes the cart move until it crosses the second railroad switch and gets hooked by the main pulling system of the power & free conveyor: when the cart enters the unloading area, then a waiting time of 0.5s ensures that the second railroad switch is available (so the shared resource can be returned back) and the second half of the path within the working area becomes available again.

Note: The use of a boolean indicator for the availability of the second half path within the working area is needed to prevent that a subsequent cart enters Station 2 and receives a validation signal meanwhile the previous one is still in that stretch of path: this situation is very unlikely to happen, however it was analyzed to prevent any possible accident.

3.6.3 Operation of the unloading station

The unloading station, also called Station 3, accommodates carts whose content may be picked up or exchanged with products carried in boxes within the belt conveyor system that lies close to it. It is the station with the simplest behaviour, due to the fact that the industrial manipulating robot Fanuc Arc Mate 100 iB is not considered herein; as already mentioned, a human worker carries out the task of swapping contents at this stop point.

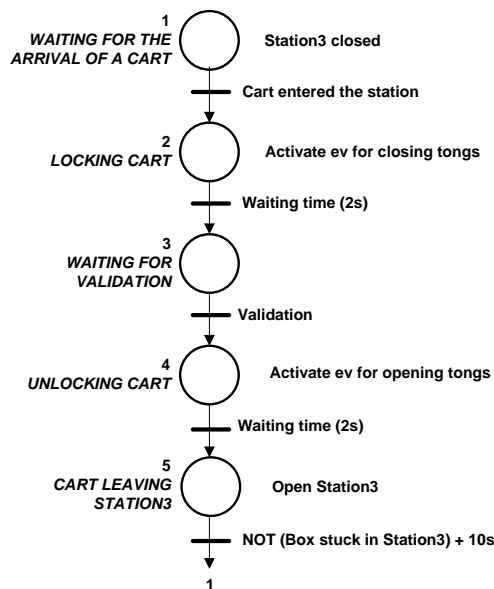


Figure 3.12: Petri net for the unloading station of the power & free conveyor.

It is worth remembering that Station 3 is the only one provided with tongs controlled by pneumatic components to lock a cart, while the operator removes pieces from it or adds in it products coming from a box of the belt conveyor system; except for this peculiarity, the functioning of this stop point is the same as that of the other stations and it is summarized in Figure 3.12.

Station 3 remains closed, but with open tongs, until a cart enters it, then tongs sequentially close to lock the hook and fasten it to the framework of the stop point; after $2s$, the cart starts waiting for a validation signal that lets it move to the loading station. Once the cart gets validated, tongs release it and additional $2s$ pass before Station 3 activates its opening mechanism, to avoid the risk that the cart gets hooked by the pulling system while it is still locked, causing a possible damage in the installation. If the cart leaves Station 3 within $10s^4$, then the stop point closes again and a subsequent cart may enter it.

3.7 Programming within TwinCAT

This section shows the POUs that were used in order to realize the automation procedure of the power & free conveyor.

The PLC Control program includes many blocks that comply with the statements of Section 2.1:

1. *Main* (PROG): as the name suggests, it is the principal program and it calls all the other POUs;
2. *Alarms* (FB): this block manages faults occurred in the operation of the pulling system and errors that may feature any station;
3. *ModesEvolution* (FB): it is a generic module for controlling the evolution of operating modes in an industrial system that complies with the standard UNE EN 60204-1. Moreover it governs green and red indicating lights that may be turned on (with either a steady or blinking behaviour) or off;
4. *Pull* (FB): it is responsible for the activation of the electric motor that drives the pulling system;
5. *Station* (FB): it elaborates all features concerning stations of the system and their behaviour;
 - *CartStackManagement*: this action is enabled after a cart is validated and it updates the stack of the station to which that cart is headed;

⁴Otherwise, an alarm would be generated due to a possible fault in the station.

- *UpdateStacks*: it is an action involved in the update of a stack after the opening of a station, as a cart moves from a stack to another one;
6. *SwitchesManagement* (FB): it is the smartest module of the program, in fact it controls the sequential operations carried out by the loading station and the working stations; to this goal, it manages the settings to which both the bypass path and the deviation to the working area are subjected;
 - *Enabling*: it is the action in charge of setting the positions of both railroad switches, which may head carts along either the long path or the short path;
 7. *Cont-OFFON* (FB): it is a generic module that counts how many times a variable (which is a parameter assigned the block) presents a rising edge. Thus, it implements a counter for a quantity of interest, for example the number of opening cycles carried out by a station;
 8. *Cont-TimeON* (FB): this generic block measures the duration of the time interval during which a signal remains activated;
 9. *Cylinder* (PROG): it manages the activation of the air cylinder located in the working area to ensure the motion of carts.

In Table 3.5 possible and undesired events that may occur during the system operation are indicated; in order to prevent damages to the framework or to the staff, the generation of specific alarms was planned.

ID	Alarm description
0	Absence of alarms within the installation
1	Jam of the pulling system
2	Emergency stop
3	Excessive torque of the electric motor
4	Thermal switch opened circuit
5	At least one station is in an error condition
6	Virtual emergency stop of the HMI

Table 3.5: Alarms of the power & free conveyor.

3.8 Description of the related HMI

This section deals with the realization of a human-machine interface to control and check the operation of the power & free conveyor and to govern the tasks that are realized by the stations. An HMI represents a powerful and reliable tool to control the system without the need to stay in the same

place where it is installed, assumed that there is some kind of communication between the computer and the plant (for example, via an Ethernet cable). With the goal of portability in mind, the configuration of the connection is carried out by means of an XML file that may be modified and updated according to specific requirements.

The main page of the HMI is depicted in Figure 3.13; it is possible to distinguish a primary part (the left half), which is always in the foreground, from a secondary one (the right half), which collects navigation tabs that are accessible to the user.

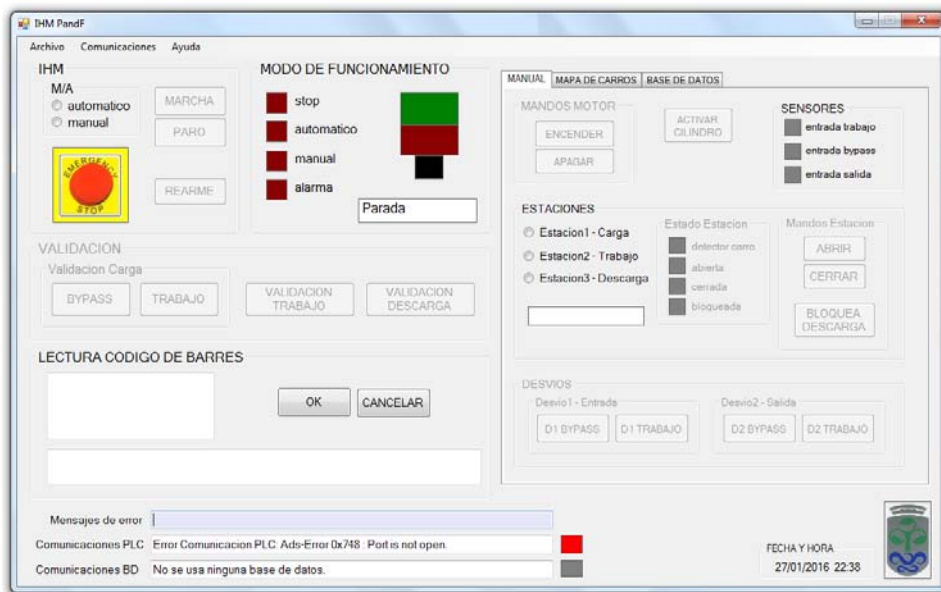


Figure 3.13: Main page of the HMI for the power & free conveyer.

At the top of the page three menus appear:

- *Archive*: herein it is possible to select the desired XML configuration file, alternatively the default one is loaded;
- *Communication*: it is used in order to establish a connection between the HMI and a target device which is detected thanks to its IP address;
- *Help*: it opens a pop-up page that contains information about the program.

Moreover, at the bottom of the page there are the logo of Vigo University and a date/time indicator. In addition, three text boxes have been created to display error messages and information about communication with PLC and the database; at their sides, related graphical indicators turn green once

the connection is established, otherwise they are red and a description of the issues is displayed.

If the communication with the PLC representing the target system is missing, then no commands can be executed within the HMI environment; in this way, no change of settings is possible in the absence of connection, something that could affect the following work session once the link is activated.

The *Functioning Modes* area is made up of light indicators, which graphically alert the user about the current operating mode of the system, and a text box; two bigger lights simulate the real operating lights, whose behaviour (fixed, blinking or turned off) depends on which mode is in force.

The *HMI* area contains the virtual buttons that allow to control the evolution of the modes in the considered system: it is possible to choose between manual and automatic mode and also to command signals for turning on/off the installation, generate a general alarm and send an acknowledgment in order to reset the alarm and restart the system.

The *Validation* area gathers four buttons that allow the validation of carts which are queued in the stations of the power & free conveyor; as already mentioned, in the loading station two different kinds of validation are possible (to proceed along either the long path or the short one), whereas in the working station and in the unloading station it is not possible to decide the destination of a cart that is validated (since the path is unique).

Last but not least, as far as the general part of the main page is concerned, the *Barcode Reading* area performs the conversion of codes read by the barcode scanner into a readable text and it displays that text on the screen with a brief description of the selected action: before executing the command, the program asks to confirm the action or to cancel it in case it is not the desired one. The request a confirmation is a programming choice that prevents the realization of undesired tasks due to the worker distraction.

The *Manual* tab (Figure 3.14) includes all the commands that are required to intuitively control the manual operation of the power & free conveyor: the operator can make the electric motor run and enable the pulling system of the working area, too. Once he selects a station he is able to manage its evolution by closing and opening it or even setting the positions of both railroad switches; various indicators inform him about the state in which the station is and the signals coming from sensors and outputs, so the operator is aware of the current condition within the installation.

It is important to point out that all virtual buttons were programmed following a foolproof strategy, for example:

- if Station 3 is selected, then buttons to set railroad switch positions are not available because that stop point does not employ them;



Figure 3.14: *Manual* tab of the HMI for the power & free conveyor.

- if Station 1 or Station 2 is selected, it is not possible to push the switching button to lock and unlock carts in fact both them have no tongs;
- if Station 3 is selected, then the operator is allowed to open it if and only if tongs are open, otherwise a locked cart or an incoming cart carried by the pulling system could damage the station.

The *Stacks* tab (Figure 3.15) deals with the three FIFO stacks of carts that are associated to each station of the system under consideration. Since

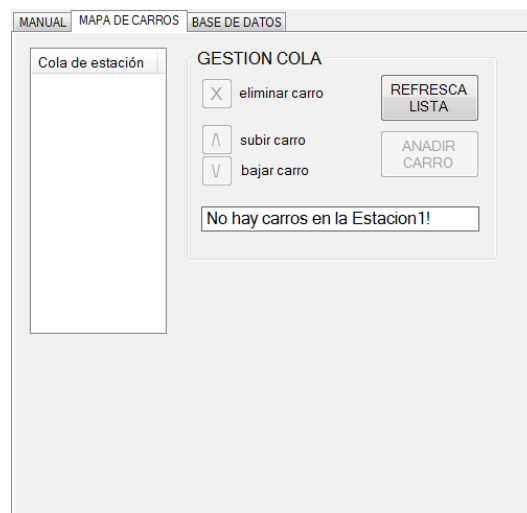


Figure 3.15: *Stacks* tab of the HMI for the power & free conveyor.

the processing of the PLC program manages these stacks and saves their properties as retain variables, the HMI can visualize them and modify their content; note that there are virtual buttons that allow to delete, shift up, shift down or even add a cart within the stack of the selected (if one) station, besides a button for updating the visualization of its content.

Every non-empty stack is displayed as a list of numbers that represent the identification number of each cart (or box), with the top value being the cart stuck at that station whilst the value at the bottom indicates the last of subsequent carts.

The implementation of stacks allows to track the flow of goods within the installation in order to improve performances and to be aware of the utilization of products.

The *Database* tab (Figure 3.16) contains 4 subtabs that are designed to manage a database implemented within the application *Microsoft SQL Server Express* on the personal computer; they store information regarding historical production of the manufacturing process that will be analyzed at higher levels of the CIM pyramid with the aim to update the related MES.

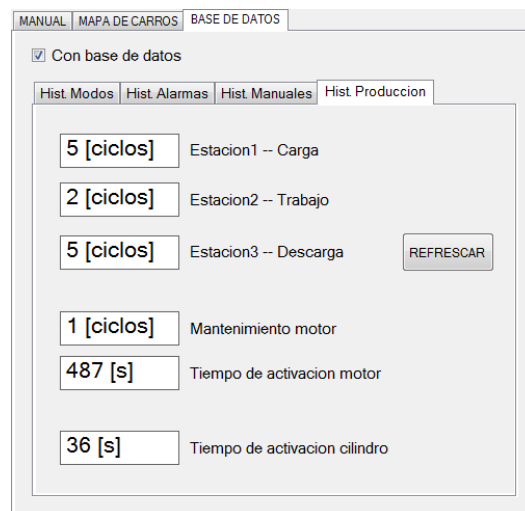


Figure 3.16: *Database* tab of the HMI for the power & free conveyor.

Since the communication with the database requires a considerable consumption of memory and reduces the performance of the HMI, it is given the opportunity to the user to choose if he/she wants to activate the updating of the database during a work session of the power & free conveyor; if this is the case, the HMI environment is responsible for tracking the operation of the system and particular events that characterize the following parameters:

- *Operating modes*: the inspection of the ways in which the system works provides a useful information about its utilization;

- *Alarms*: within a work session many alarms due to faults or errors may be generated. It is important to know all them in order to be aware of the system conditions;
- *Manual actions*: when the manual operating mode is active, all the enabled outputs need to be saved to check if the installation was controlled in a proper way during its operation;
- *Maintenance*: a regular monitoring of the tasks carried out by the stations and frequent checks on the operating time of both pulling systems allow to know the actual use of mechanical components and to estimate the consumption of electric power.

Every time that the occurrence of an event causes a change in one of the previous values, its new value is saved in the database as an entry with various fields: date and time when the event occurred, a unique identification number for that event and a brief description of the event. The identification number and the description of every event are listed in the XML file used for the configuration of the system, as it is explained in Appendix C.

Chapter 4

The double level conveyor belt

4.1 About belt conveyor systems

A **belt conveyor system** is a mechanism of conveyance composed by two or more pulleys that rotate generating the loop motion of its carrying medium, also known as **conveyor belt**.

In general, only **drive pulleys** are powered by an electric motor: they produce the movement of the belt, whilst **idler pulleys** rotate passively in order to allow a continuous flow of goods on the belt. The belt is made up of one or more layers that can be distinguished in:

- **carcass**: the lower layer is a woven fabric with weft and warp in polyester or nylon that provides strength and shape to the belt;
- **cover**: the upper layer has various rubber and plastic frames, whose shapes depend on the kind of good that will be carried by the system.

Depending on the scope of the plant, there are two main types of belt conveyors: the ones taking care of the pallets or boxes moving along the assembly line of a factory, and the ones handling large amounts of bulk material (e.g. grain or coal). Nowadays conveyor belt systems are widely diffused even in various processes not belonging to industrial environment; moving sidewalks, escalators and check-out counters are actual examples.

The main advantage of using a conveyor belt system is its versatility; in fact, such a plant is labor saving and allows to transport rapidly large volumes of material in any direction, reducing production costs.

4.2 Description of the system

In the “Ricardo Marin” laboratory there is a double level conveyor belt, namely a belt conveyor system made up of two overlapped belts that are

connected to each other by means of two elevators; this structure allows box transportation between the warehouse and an area next to the unloading station of the power & free conveyor.

The double level conveyor belt is shown in Figure 4.1, where it is possible to notice that belts present a curvature of ninety degrees; the possibility of abrupt changes of direction is an essential topic from a space saving point of view.



(a) View to the power & free conveyor.



(b) View to the warehouse.

Figure 4.1: High views of the belt conveyor system under consideration.

In Table 4.1 are indicated the length measures concerning of the whole system; in addition, boxes have standard sizes equal to $395 \times 295 \times 215$ mm.

Measure	Value [mm]
External width	515
Internal width	435
Short straight portion	1380
Long straight portion	2180
External radius of curvature	610
Internal radius of curvature	175
Total height	1035
Low level height	340
High level height	940

Table 4.1: Relevant sizes of the double level conveyor belt.

The system seems to be quite simple, but it actually has to be structured in a very smart way; in fact, under the horizontal integration point

of view, it is a medium of physical connection between the warehouse and the power & free conveyor that has to handle requests coming from both these systems. Furthermore, many functioning options are possible and so its behaviour changes according to which settings are defined, as it will be explained in the remainder of the chapter.

The conveyor belt is lay out on two floors, which can be thought as two FIFO stacks and are responsible for boxes moving in opposite directions: indeed, boxes on the higher level are carried towards the power & free conveyor, meanwhile the ones on the lower level move to the warehouse. Depending on the number of boxes that populate the system at a certain time, the two stacks may vary their occupancy; more specifically, due to belts length and boxes size, each belt can accommodate a maximum of 9 boxes. Furthermore, there are two elevators and each of them can host at most one box at a time in order to transfer a box from one floor to the other; they are associated to stacks¹, too. Hence, the capacity of the belt conveyor system reaches the value of 20 boxes.

As mentioned above, the system lies in a distribution area because it transfers products of limited size between the warehouse, where they are handled manually or by means of the automatic stacker crane, and the power & free conveyor, which will carry them along the supply chain; there is a steady flow of boxes in a closed loop that need to be managed.

Before proceeding, it is necessary to clarify certain operating aspects which will be assumed in the following. For the sake of simplicity, from now on the elevator next to the warehouse will represent the “**warehouse area**”, whilst the other one will denote the “**robot area**” of the system; in fact, although it was not used in this project, an industrial manipulating robot Fanuc Arc Mate 100 iB lies between the double level conveyor belt and the power & free conveyor. Figure 4.2 graphically describes the previous subdivision: in (a) it is possible to see the warehouse with the automatic stacker crane, whereas (b) shows the robot which acts on the unloading station of the power & free conveyor.

Because of the nature of the belt conveyor system and its role as intermediary, various functionalities are possible; the main principle to remember is that filled boxes move from the warehouse area to the robot area on the upper floor and then they are carried in the opposite direction on the lower one.

¹To be precise, stacks of unitary size.

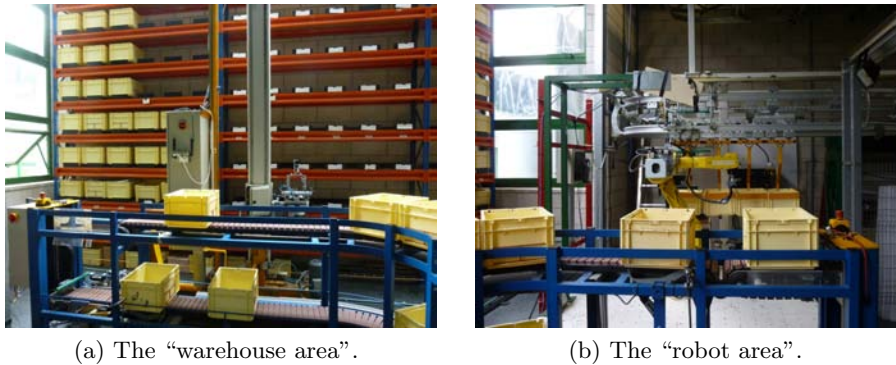


Figure 4.2: Exchange areas of the belt conveyor system.

4.3 Elevators

Each elevator consists of a particular platform that is able to move vertically to transport boxes from one floor to the other. Such a platform consists of a belt, that is properly fixed to a column controlled by pneumatic elements, with a swinging mechanical component (that is pointed out in Figure 4.3(a)) at its side: one end is connected to a coiled spring whereas its opposite has two idler pulleys that can generate a physical contact with the idler pulleys that are located on both floors of the belt conveyor system. Thus, the motion of the two conveyor belts is transmitted to the platform² when the elevator reaches a certain floor of the system: the platform belt assumes the same direction of motion of the belt at that level. In Figure 4.3 the platform of the elevator that lies within the warehouse area is represented: differently from the other one, it has two side brackets whose benefits will be pointed out in following sections.

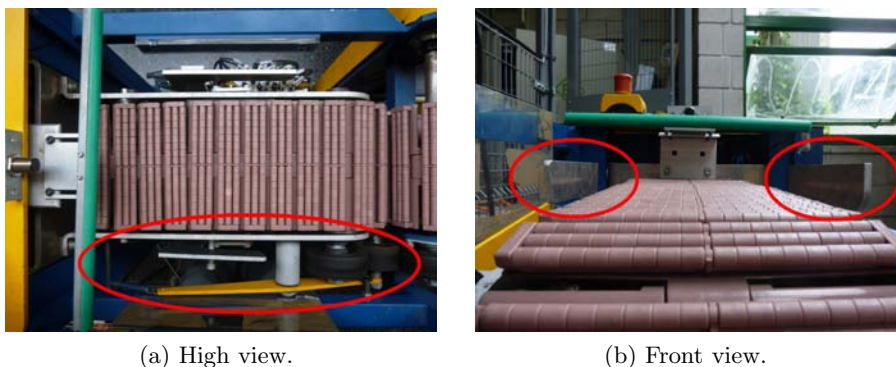


Figure 4.3: Platform of the elevator in the warehouse area with its side brackets.

²Of course, assuming that the engine is on.

The elevators of the belt conveyor system are almost identical under a physical perspective, however they present different features. Similarities in their behaviours are resumed as follows:

1. There is an elevator entry that is closed most of the time, anyway it can be opened under special conditions; more specifically, the access is controlled by means of a mechanical obstruction that can stop boxes or let one of them enter if the elevator is *available*, namely if the platform lies at the corresponding floor and there are no boxes on it. This strategy prevents the opening when there is already a box on the platform or when the platform is positioned at the lower level: a box would fall in both cases, thus causing possible damages to products and the framework.
2. If the elevator is available and there is at least one box stuck at its entrance, then the access is unlocked and only that box can enter the elevator. In fact, the obstruction is a swinging hooked arm positioned under the entrance (Figure 4.4): its front end results lifted up and locks the box when the elevator access is closed, whereas it sets down during the opening; this movement causes the back end to rise in order to stuck possible subsequent boxes. The motor starts to run and activates the motion of the platform belt thanks to idler pulleys; the elevator collects the incoming box, which leans on a wall at the end of the platform.



Figure 4.4: The swinging obstruction that controls the access to an elevator.

3. The platform transfers the box to the other floor.
4. Once the idler pulleys of the platform match to the idler pulleys of the conveyor belt located on that floor, the box leaves the elevator and is

carried along that level.

Figure 4.5 shows accesses to the elevators of the double level conveyor belt. Note that the entry of the warehouse area lies at the lower floor, while the robot area is accessible from the higher floor; related optical sensors for the detection of stuck boxes are pointed out with red circles.

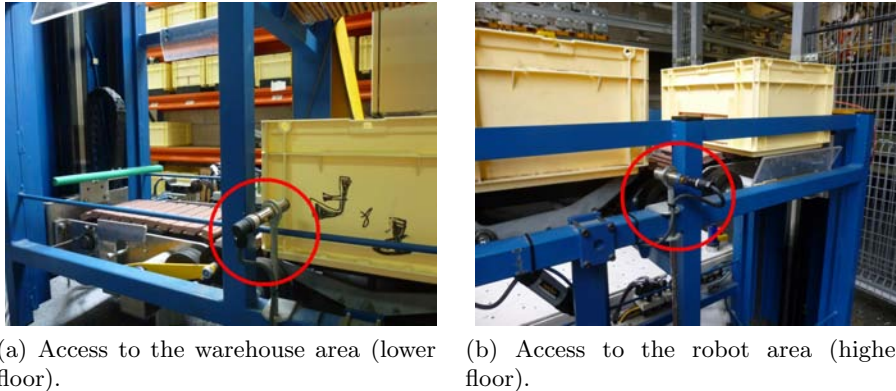


Figure 4.5: Entrances of the elevators.

What distinguishes the elevators is their connection to the belts and in particular the direction of the motion; they work in an analogous way, which is described in detail in next subsections.

4.3.1 The elevator of the warehouse area

The behaviour of the warehouse area is quite complex to be managed because of its possible functioning options. As a matter of fact, whereas the robot area is merely used for the transit of boxes, the elevator herein considered needs to face one issue more: boxes may be taken away or new ones may be put in circulation because of exchange operations between the belt conveyor system and the warehouse.

For this reason the elevator platform is provided with two side brackets, as shown in Figure 4.6: they are necessary in order to stop a box at the higher floor, otherwise the sliding belts would carry it to the robot area.

In fact, when the filled platform raises to the higher floor the side brackets lift to interrupt the physical contact between the belt and the box over it; in this situation, depending on which mode the system is working in, the human worker or the warehouse intelligence (which is represented by the automatic stacker crane) decides if the remaining box has to keep staying in the belt conveyor system or it is preferable to remove it and put it again in the warehouse. On the other hand, when the platform lies at the lower floor there may be a request from the warehouse to put a new box on the conveyor belt: thus, the elevator reaches the higher floor without loading

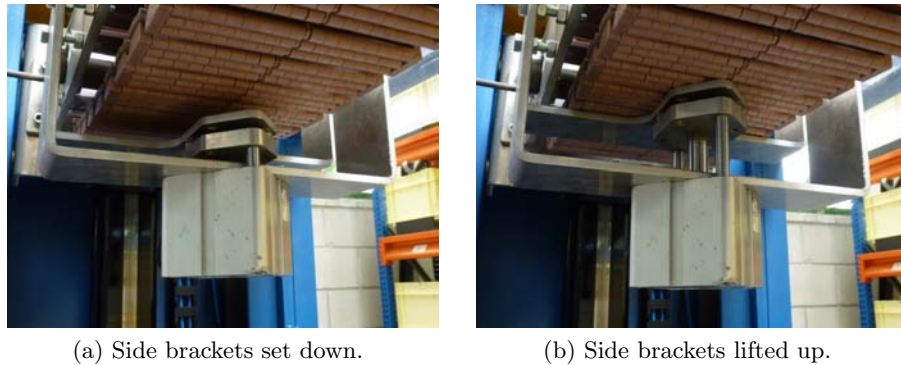


Figure 4.6: Mechanism for the movement of the side brackets.

any of the boxes at its entrance and once its side brackets lift up the platform can accommodate a box coming from the warehouse.

4.3.2 The elevator of the robot area

The entry of this elevator is located at the higher floor, thus only boxes proceeding from the warehouse area can enter it. If the platform is available and at least one box is stuck at its entrance, then that box can be carried inside the elevator: here, the manipulating robot or a human worker may pick up pieces from the box or put in it new ones coming from the unloading station of the power & free conveyor. When this operation ends, or even when there is no swap of products, a validation allows the elevator to go down at the lower floor, where the box will come back to the warehouse area. As already stated, this project does not deal with the industrial manipulating robot Fanuc Arc Mate 100 iB, although it would not require a huge effort to consider its presence in the plant; anyway, from now on all the actions that concern the elevator of the robot area are assumed to be done manually.

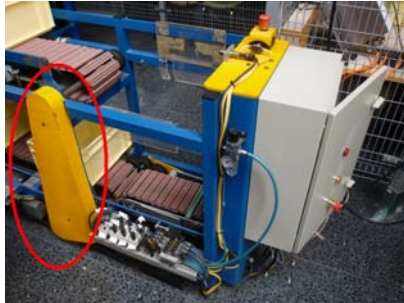
4.4 Electrical equipment

This section describes the electrical components that characterize the system under consideration; they allow to control its operation with respect to the desired goal of the designer. The electric motor and then sensors and actuators are presented, later a review of the used PLC and its modules is provided.

4.4.1 Electric motor

As mentioned above, the double level conveyor belt counts on four belts, more specifically one for each elevator and those at the higher and lower floors of the system: they are all driven by means of a SEW-EURODRIVE

three-phase alternating current motor powered with 400V that supplies an electric power equal to 0.25kW.



(a) Motor and transmission of motion.



(b) Idler pulleys of an elevator.

Figure 4.7: The elevator of the warehouse area with the electric motor and the control cabinet.

The motor, which is positioned next to the warehouse area, operates a drive pulley that transmits motion to idler pulleys physically connected to it.

4.4.2 Sensors

The double level conveyor belt is equipped with various sensors to reveal the passage or arrival of boxes in certain points of interest for any of the elevators; brief descriptions of these devices are contained in Tables 4.2 and 4.3 for the elevators of the warehouse and robot areas, respectively.

Description	Task
Photocell	Detection of a stuck box at the entrance of the elevator
Photocell	Detection of a box in the elevator at lower ground
Reed switch	Detection of the elevator platform at lower ground
Reed switch	Detection of the elevator platform at higher ground
Photocell	Detection of a box in the elevator at higher ground
Photocell	Detection of a box getting off the elevator
Reed switch	Detection of the side brackets at lower ground
Reed switch	Detection of the side brackets at higher ground

Table 4.2: Sensors in the warehouse area of the belt conveyor system.

Description	Task
Photocell	Detection of a stuck box at the entrance of the elevator
Photocell	Detection of a box in the elevator at higher ground
Reed switch	Detection of the elevator platform at higher ground
Reed switch	Detection of the elevator platform at lower ground
Photocell	Detection of a box in the elevator at lower ground
Photocell	Detection of a box getting off the elevator

Table 4.3: Sensors in the robot area of the belt conveyor system.

As it can be observed, the warehouse area is constituted by the same equipment except for two reed switches that detect the position of the side brackets; reed switches with different goals are indicated in Figure 4.8.

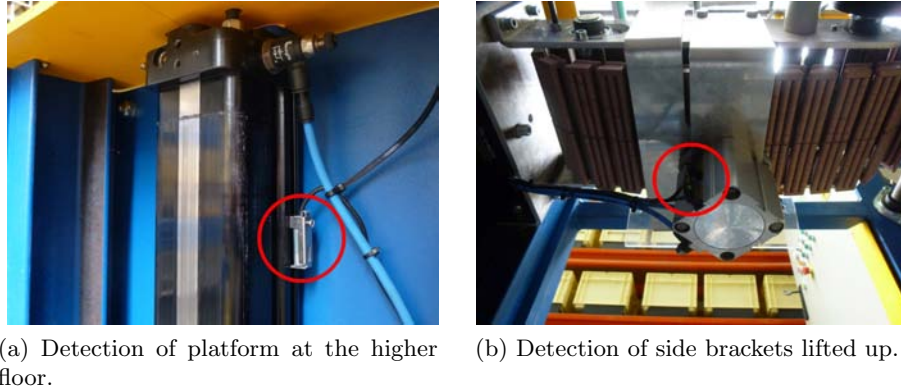


Figure 4.8: Two reed switches used in the elevator of warehouse area.

4.4.3 Actuators

The system under consideration needs five pneumatic actuators to work properly: they are air cylinders (which are matched to solenoid valves) used for managing the access to platforms and control vertical movements of the elevators and the side brackets. Tables 4.4 and 4.5 collect them: it is useful to recall that only the elevator within the warehouse area is provided with side brackets.

Description	Task
Air cylinder	Locking mechanism at the elevator entrance
Air cylinder	Ascent and descent of the elevator
Air cylinder	Ascent and descent of the side brackets

Table 4.4: Actuators in the warehouse area of the belt conveyor system.

Description	Task
Air cylinder	Ascent and descent of the elevator
Air cylinder	Ascent and descent of the side brackets

Table 4.5: Actuators in the robot area of the belt conveyor system.

4.4.4 PLC modules

The device used to control the belt conveyor system is a Ethernet TCP/IP Bus Terminal Controller Beckhoff BC9050 [11], which is a **bus coupler**³ provided with integrated PLC functionality and a fieldbus interface for Ethernet.

The BC9050, which is represented in Figure 4.9, is a non-central intelligence constituted by a bus terminal, with at most 64 terminals, and a bus end terminal that is necessary for data exchange between the bus coupler and the bus terminals; it could be programmed within the TwinCAT programming environment according to IEC 61131-3 standard and the PLC program can be loaded via Ethernet. In addition to the BC9050, the following devices

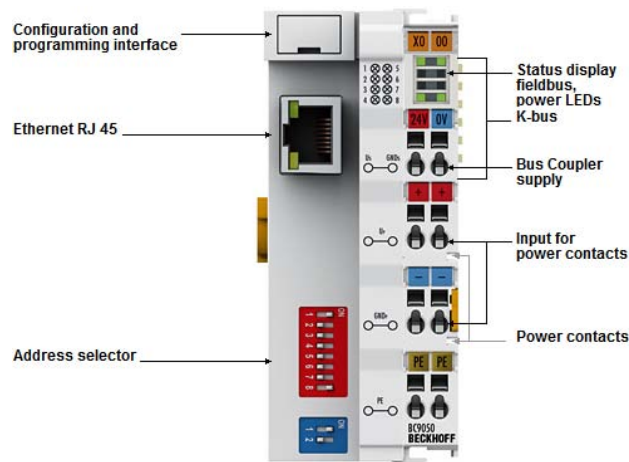


Figure 4.9: Scheme of the controller Beckhoff BC9050.

constitute the core of the automation for the system under consideration:

- Siemens SITOP power 2 (6EP1331-1SL11): power source of at most 24V and 5A;
- Siemens AS-i POWER (3RX9 502-0BA00): power supply for the AS-i interface of at most 30V and 5A;
- Telemecanique Altivar 31: frequency inverter for three-phase asynchronous motors;
- Beckhoff KL6201: master terminal of the AS-i bus;
- Beckhoff KL9010: bus end terminal.

The power sources, SITOP power 2 and AS-i POWER, are the same used for the power & free conveyor. The connection between TwinCAT

³Namely, a device that couples two buses without any interruption in power supply.

applications in the PC and the controller is made via Ethernet cable; the address of the BC9050 was fixed to 193.147.81.242 and the configuration procedure was based on [12].

Note: PLC programs for the Beckhoff BC9050 need to be saved with the extension “.prx”; it is different from the format “.pro” which is normally used with embedded PCs, as the BC9050 is a low range product.

4.5 Network configuration

The procedure explained in Section 2.6 for the addressing of As-i slaves allowed to map input and output signals, such as photocells and pneumatic actuators, that characterize the whole system; for the sake of completeness, the result of this mapping is reported separately in Tables 4.6 and 4.7.

Address	Location	Identification of the INPUT signal
0	System data	-
1	Warehouse area	IN1 - Emergency stop IN2 - Not connected
2	Warehouse area	IN1 - Detector box leaving lift IN2 - Detector box stuck at lift entrance
3	Warehouse area	IN1 - Reed side brackets set down IN2 - Reed side brackets lift up
4	Warehouse area	IN1 - Detector box in platform (low level) IN2 - Detector box in platform (high level)
5	Warehouse area	IN1 - Button IN2 - Emergency stop IN3 - Not connected IN4 - Inverter Drive
6	Robot area	IN1 - Detector box in platform (low level) IN2 - Detector box leaving lift
7	Robot area	IN1 - Detector box in platform (high level) IN2 - Detector box stuck at lift entrance
8	Robot area	IN1 - Emergency stop IN2 - Not connected
9	Robot area	IN1 - Reed platform at high level IN2 - Reed platform at low level
10	Warehouse area	IN1 - Reed platform at high level IN2 - Reed platform at low level

Table 4.6: Mapping of the input AS-i slaves in the belt conveyor system.

Address	Location	Identification of the OUTPUT signal
0	System data	-
1	Warehouse area	OUT3 - Not connected OUT4 - Not connected
2	Warehouse area	OUT3 - Coming up of platform OUT4 - Coming down of platform
3	Warehouse area	OUT3 - Opening of lift access OUT4 - Closure of lift access
4	Warehouse area	OUT3 - Coming up of side brackets OUT4 - Coming down of side brackets
5	Warehouse area	OUT1 - Outputs activation OUT2 - Control cabinet light OUT3 - Light of control cabinet button OUT4 - Motor
6	Robot area	OUT3 - Coming up of platform OUT4 - Coming down of platform
7	Robot area	OUT3 - Opening of lift access OUT4 - Closure of lift access
8	Robot area	OUT3 - Not connected OUT4 - Not connected
9	Robot area	OUT3 - Not connected OUT4 - Not connected
10	Warehouse area	OUT3 - Not connected OUT4 - Not connected

Table 4.7: Mapping of the output AS-i slaves in the belt conveyor system.

4.6 Description of the automation procedure

This section deals with the most interesting part of the project, in fact it analyzes the strategies used to control the operation of the double level conveyor belt; the work described in the following aimed at improving the performances of the system, meanwhile saving power and ensuring the plant safety.

Figures 4.10 and 4.11 indicate the Petri nets used for describing the desired functioning of the elevators in the warehouse and robot areas, respectively; the automation procedure follows these diagrams.

4.6.1 Operation of the warehouse area

The elevator representing the warehouse area behaves in a sophisticated way for what it concerns programming. As a matter of fact, three different events

may occur in it:

- a new box proceeding from the warehouse is put on the platform, so the system counts on an additional box;
- a box coming from the robot area is picked up to be stored in the warehouse, hence the system “loses” a box;
- a box coming from the robot area is sent again to the other elevator, thus it remains within the system and there is no variation in the number of moving boxes.

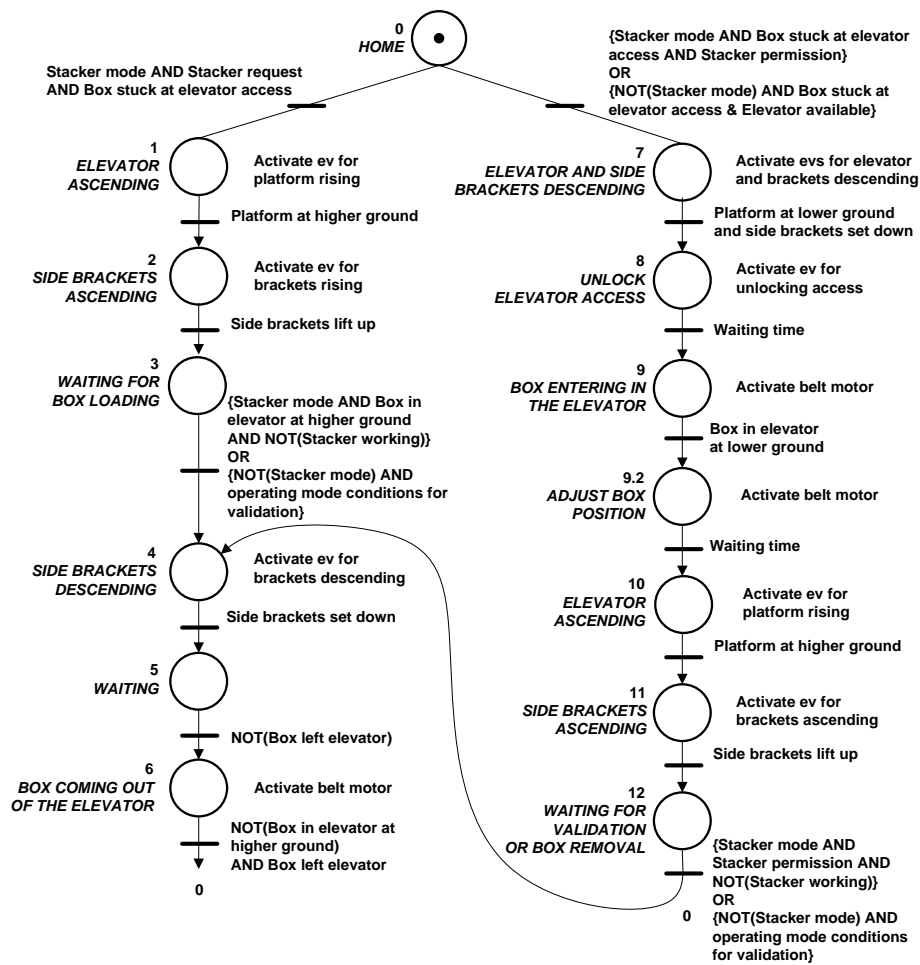


Figure 4.10: Petri net for the warehouse area of the belt conveyor system.

Due to the complexity of implementation, the Petri net depicted in Figure 4.10 present two parallel branches; they mainly depend on the utilization of the stacker crane, which could change the number of boxes within the conveyance system.

- Assumed that the stacker crane does not present any request for adding a new box or it is not even used, if there is a box stuck at the elevator access and the platform is available then the warehouse area behaves as described by the right branch of the related Petri net. The elevator descends to the lower floor and the side brackets set down, so that the platform can accommodate the box which was stuck at the entrance once the access gets open; after a brief waiting time, which is necessary to let the box adjust its position, the elevator ascends to the higher level and when it reaches the proper height then the side brackets lift up. In this situation the system waits for a validation, perhaps subsequent to a box removal performed by the stacker crane or manually, and then the system evolves from the state 4 of the left branch of considered Petri net.
- Otherwise, if the stacker crane sends a request with the aim to insert a new box within the conveyor belt, then the left branch of the Petri net is followed. The empty platform rises to the higher floor and the side brackets lift up to accommodate the box coming from the warehouse; once the tasks carried out by the stacker crane end, a validation signal allows the side brackets to set down in order to carry the new box towards the robot area, in such a way that the elevator becomes available again.

4.6.2 Operation of the robot area

The working of this elevator is much simpler than the previous one, especially for the absence of the industrial robot that simplifies the corresponding Petri net. Assumed that at the first start of the double level conveyor belt the platform lies at the lower floor, if the elevator is available and there is a box stuck at its access then the platform ascends to the higher floor and once there it waits 2s in order to ensure its stable position; if the platform remains free from boxes, then the elevator access gets unlocked, thus allowing the box to enter the elevator after the activation of the electric motor. When the photocell at the access does not detect anymore the presence of the moving box whereas the photocell situated inside the elevator is activated by the same box, then the access is locked again and the platform after a waiting time carries out the exchange operations (load or unload of products) with the unloading station of the power & free conveyor; once these tasks are completed, a validation signal allows the platform to descend to the lower floor. Here, the running motor transmits motion to the belts, hence the box gets off the elevator; the transit of the box over the photocell at the exit of the elevator produces the ascent of the platform to the higher floor, so that it is ready in case there would be a box arriving from the warehouse area.

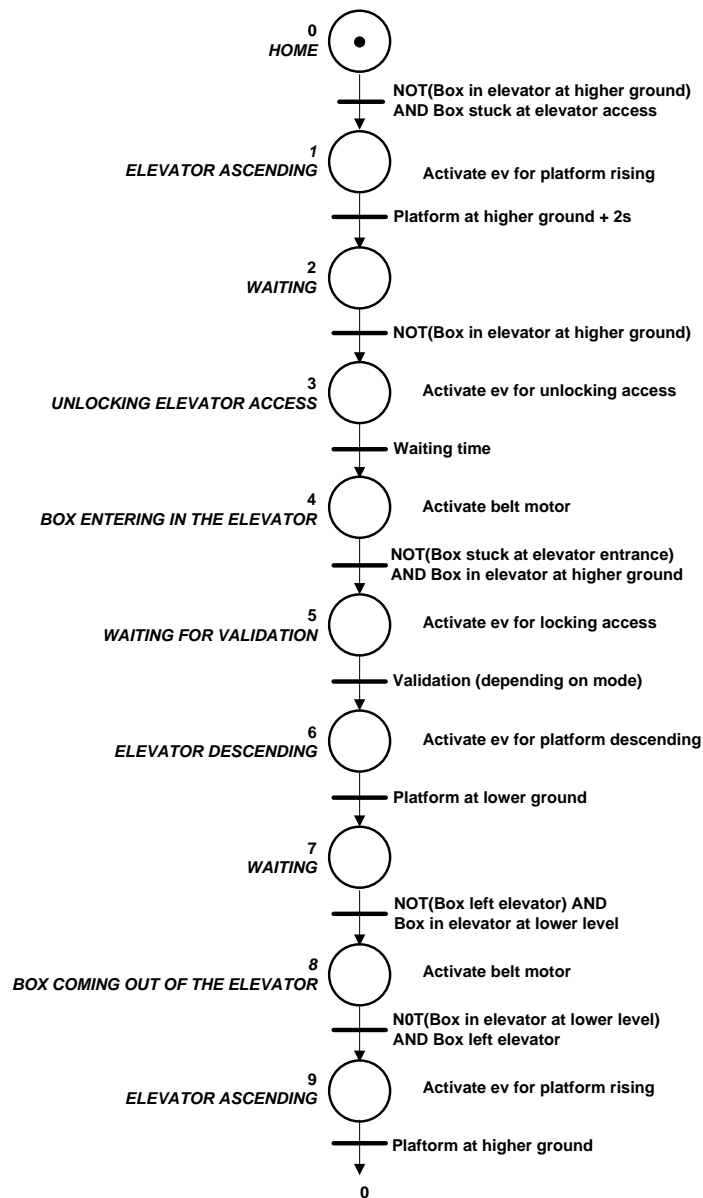


Figure 4.11: Petri net for robot area of the belt conveyor system.

4.7 Programming within TwinCAT

This section presents the list of all POU's that were realized for the automation of the double level conveyor belt.

The PLC Control program is made up of various blocks that comply with the detailed description of Section 2.1:

1. *Main* (PROG): as the name suggests, it is the principal program and

- all other POU's are called from it;
2. *Alarms* (PROG): this program manages errors generated when pushing at least one of the three emergency stops or resulting from faults in elevators, when a box gets stuck along the path;
 3. *ModesEvolution* (FB): it is a generic module for controlling the evolution of the operating modes in an industrial system that complies with the standard UNE EN 60204-1;
 - *FlashingLights*: it is an action that carries out the control of green and red indicating lights, which may be turned off or on and when they are on they can be steady or blinking;
 4. *TimedPulsation* (FB): this generic module recognizes if the pressure on a physical button has been shorter or longer than a certain duration (which is a parameter assigned to the block);
 5. *ActivationOutputs* (PROG): it is responsible for the activation of generic output signals of the installation, namely the electric motor and the indicating lights;
 6. *ActivationOutputsWarehouseSide* (PROG): this program enables all outputs that involve the warehouse area of the belt conveyor system;
 7. *ActivationOutputsRobotSide* (PROG): this program activates all outputs within the robot area of the belt conveyor system;
 8. *AutomaticModeWarehouseSide* (PROG): it implements the sequential operations of the elevator within the warehouse area;
 9. *AutomaticModeRobotSide* (PROG): it manages the functioning of the elevator next to the robot area;
 10. *Stack Management* (PROG): this program initializes and controls the four stacks of boxes that are virtually associated to the whole system;
 - *BoxMoreHigh*: it updates the higher floor stack when a new box enters it;
 - *BoxLessHigh*: it updates the higher floor stack when a new box leaves it;
 - *BoxMoreLow*: it updates the lower floor stack when a new box enters it;
 - *BoxLessLow*: it updates the lower floor stack when a new box leaves it;
 - *UpdateStacks*: this action updates all stacks when it is necessary.

Table 4.8 collects the main danger situations that may occur in the installation and hence are associated to the generation of alarms.

ID	Alarm description
0	Absence of alarms within the installation
1	Emergency stop of the control cabinet
2	Emergency stop of the elevator in warehouse area
3	Emergency stop of the elevator in robot area
4	Jam at the elevator in warehouse area
5	Jam at the elevator in robot area
6	Virtual emergency stops of the HMI

Table 4.8: Alarms of the belt conveyor system.

4.8 Description of the related HMI

This section shortly describes the human-machine interface that was realized to control the functioning of the double level belt conveyor and to choose a few working options, too. The advantage of an HMI is that it represents a powerful tool that allows to control an installation even from a distance, assuming that there exists a cable connection (for example, within an Ethernet network). Due to the need for portability, the configuration of the connection is supported by the use of an XML document that may be modified and updated according to specific requirements.

The main page of the HMI is presented in Figure 4.12; it is made up of a primary part (the left half), which is always in the foreground, and a secondary part (the right one) which is divided in navigation tabs that are accessible to the user.

In the top part of the page there are three menus:

- *Archive*: it offers the opportunity to select the desired XML configuration document, otherwise the default one is loaded;
- *Communication*: it is used in order to connect or disconnect the HMI to a system that is identified by an IP address;
- *Help*: it opens a pop-up page with information about the program.

At the bottom of the page there are the logo of Vigo University and a date/time indicator. In addition, two text boxes are used to display error messages and information about communication with PLC; a light indicator turns green once the connection is established, otherwise it is red.

In absence of communication with the PLC, no command is executable within the HMI environment; this precaution avoids that settings which are

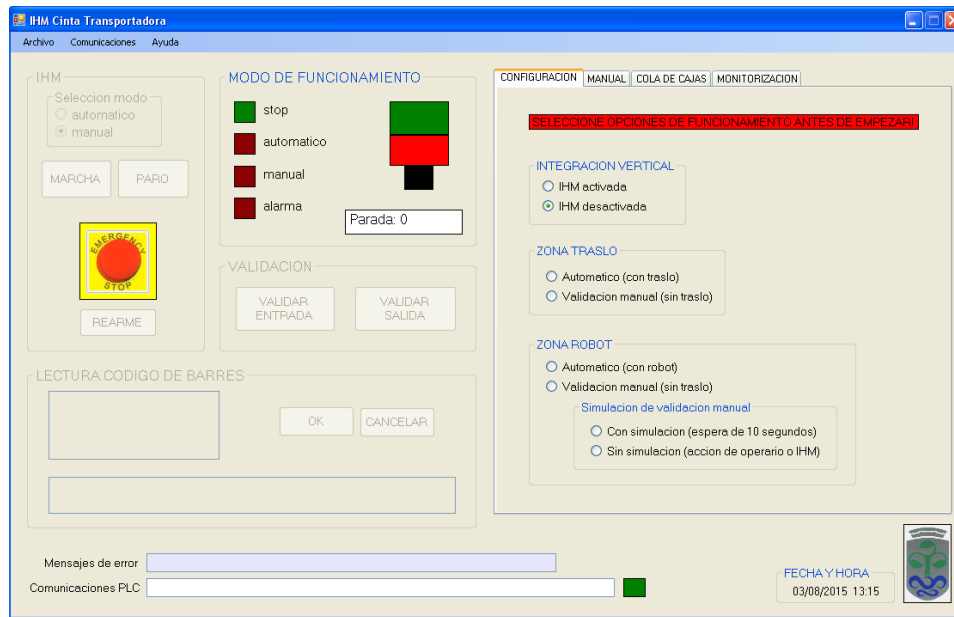


Figure 4.12: Main page of the HMI for the conveyor belt.

changed before getting connected affect the following work session once the communication is activated.

The *Functioning Modes* area is made of light indicators that indicate the current operating mode of the system; next to them, a text box shows that mode. Two bigger lights simulate the real operating lights that behave in different ways (fixed, blinking or turned off) depending on which mode is in force.

The *HMI* area contains the virtual buttons that allow to control the evolution of operating modes in the system; from here it is possible to select either the manual or the automatic mode, moreover it is possible to control signals to turn on/off the installation, to generate an alarm and to send an acknowledgment in order to reset it and restart the system.

The first step to take when starting an HMI session is to open the *Configuration* tab (Figure 4.13), because it allows to set the desired options for the operation of the conveyor belt system. More specifically, firstly it is possible to choose if the system has to behave in collaboration with the HMI or in stand-alone mode⁴, and then which settings feature the areas of interest in the system:

- warehouse area: this side of the double level belt conveyor may collaborate with the automatic stacker crane in order to exchange boxes with the warehouse. If the stacker crane is used, then it is responsible

⁴This particular mode is described in Section 4.9.1.

for the validation after it ends to remove or add a box on the platform; conversely, if the stacker crane is not considered the validation can be done only manually.

- robot area: this other side of the conveyance system lies near a manipulating robot whose task consists of exchanging products between the power & free conveyor and the belt conveyor system. Although it is not the case herein considered, if the robot acts on the platform it also has to validate boxes once it completes its tasks; in fact, the robot takes control of the elevator and the conveyor belt can only wait for a validation from it. On the other hand, when the robot is not used boxes need to be validated manually.

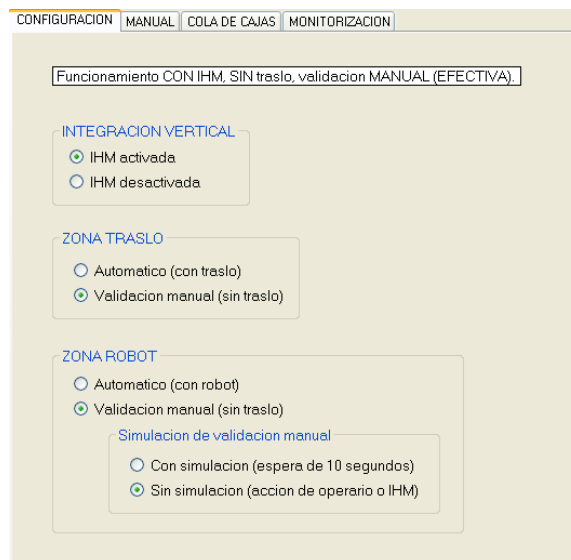


Figure 4.13: *Configuration* tab of the HMI for the conveyor belt.

As neither the automatic stacker crane or the manipulating robot were used in this project, two different kinds of manual validation were implemented: the first one consists of a command coming from a human worker that pushes a button in the door of the control cabinet or from an HMI signal, whereas the second one is given by a kind of *demo* operating mode that validates boxes on both elevators after a waiting time of 10s.

The *Manual* tab (Figure 4.14) is very simple and allows to command in an intuitive way the manual functioning of the whole system: the user is let to activate the belts motion and he/she can lift up or set down the side brackets, but more than this he/she is able to control the transfer of boxes from one floor to the other one by governing the operation of both elevators. For the sake of completeness, this tab is provided with state indicators to

inform about the evolution of the system and virtual emergency stops similar to the real ones positioned over both elevators.

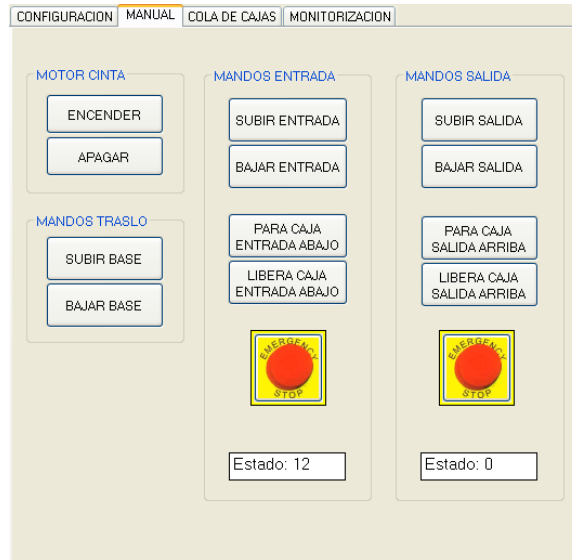


Figure 4.14: *Manual* tab of the HMI for the conveyor belt.

The *Box Stacks* tab (Figure 4.15) manages the four stacks that characterize the system; as already mentioned, both floors may be thought as FIFO stacks of maximum size equal to 9 boxes whereas both platforms represent stacks of unitary size.

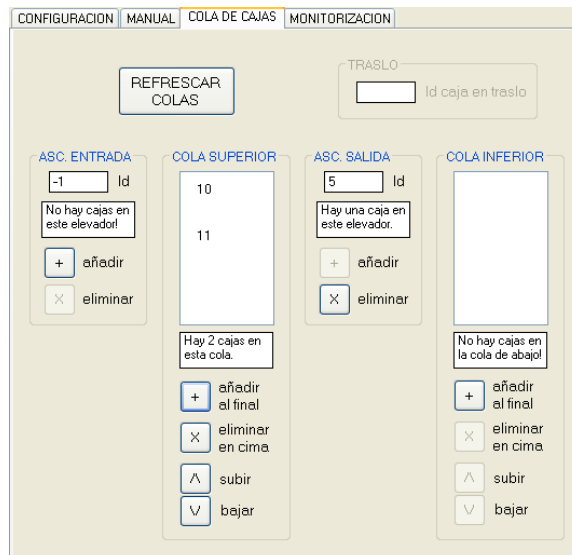


Figure 4.15: *Stacks* tab of the HMI for the conveyor belt.

Since the program run by the controller stores an identification number for each code as a retain variable and manages stacks, the HMI is able to visualize them and modify their content; as a matter of fact, virtual buttons are designed to delete, add or even shift boxes within a certain stack.

The above mentioned figure shows how actions on stacks are possible only under specific conditions; for example, one is not allowed to delete a box in an empty stack or to shift up the first element; for these reason, help texts make the user aware of his/her errors if he/she tries to do something forbidden.

Finally, the *Monitoring* tab (Figure 4.16) shows various indicators if the monitoring of the system is enabled; this choice is due to the fact that the visualization of indicating lights associated to photocell states is unnecessary and at the same time it reduces the performances of the system, as it scans signals coming from the AS-i network. However, when the monitoring of input and output signals is required, this tab displays also the number of cycles carried out by both elevators and the duration of motor activation; this information may be useful in order to estimate the work load of the double level conveyor belt and the consumption of electric power and mechanical components in it. There are buttons to update and reset the above mentioned counters from the HMI, without the need to open the file that contains the PLC program in the case it is not available.



Figure 4.16: *Monitoring* tab of the HMI for the conveyor belt.

4.9 Additional arrangements

4.9.1 The stand-alone operating mode

The double level conveyor belt has the peculiarity of being able to work without the need of forcing variable values within the programming environment *TwinCAT PLC Program* or even by means of a human-machine interface. In fact, in the door of the control cabinet located next to the elevator of the warehouse area there are two buttons that allow to control the whole system in a simple way, once it is properly programmed; the associated Petri net is depicted in Figure 4.17 and it points out how only one command button and an emergency stop are needed in order to make the conveyor working.

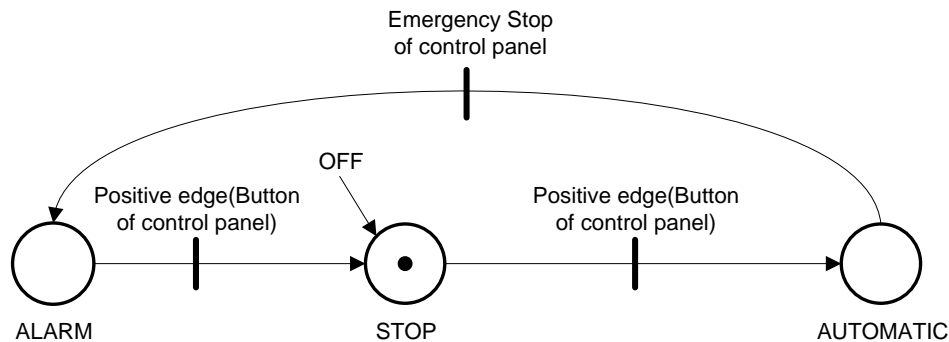


Figure 4.17: Evolution of the working modes in absence of an HMI for the belt conveyor system.

Obviously, the manual mode is not planned because it would require more buttons with the aim to manage every output signal of the system; hence, a human worker that stands in front of the control panel can command the automatic functioning of the system⁵. As a matter of fact, the first pressure on the command button starts a working cycle: after 4s of waiting, the system enters the stop mode⁶. A rising edge of the signal associated to the command button initiates the automatic operation of the system; from now on, the human worker can only validate boxes being kept in elevators at the higher floor or end the work session. For the first goal, he acts as follows:

- a short pressure of the command button (less than 2s) validates the box which is waiting on the platform of the warehouse area;
- a long pressure of the command button (more than 2s) is the validation for the box which is waiting in the elevator of the robot area.

⁵This corresponds to an automatic operating mode, in fact a manual mode would give to the user the possibility to control individually all the outputs but this is not the case.

⁶This arrangement ensures the safety of the system before a working cycle begins.

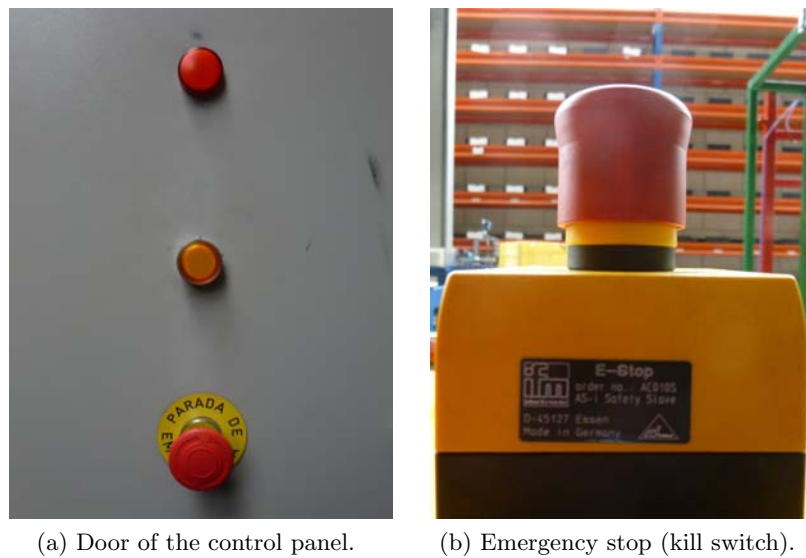


Figure 4.18: Control buttons and indicator lights next to warehouse area.

At every time instant, the user can push one of the emergency stops situated in the door of the control cabinet and over the elevator next to him (Figure 4.18), respectively, in order to arrest the normal operation of the installation and switch it to the alarm mode. Once in alarm mode, the system can be turned off by manually resetting any emergency stop that has been pushed and then pushing one time more the command button.

4.9.2 Smart management of the motor

Under the power saving point of view, it is not hard to notice that a continuous sliding of the belts is inappropriate: as a matter of fact, if almost 20 boxes were being carried on the belt conveyor system then a box leaving an elevator would move for a very short path before bumping into the following box, causing an unjustified use of the motor and an unnecessary deterioration of belt covers due to friction with boxes.

The design of box stacks within the system was already mentioned within the description of the PLC program in Section 4.7; at every time instant, boxes are tracked along the installation because they are identified with a serial number.

The information obtained from the current size of these stacks is used to determine the interval duration during which the belt motor has to be kept running; it is important to remark that the motor is activated only when any elevator releases a box it was accommodating or lets a new one enter. In other words, greater values in the stack sizes cause a reduction in the

activation time of the motor, as the occupancy of the system is greater; indeed, there are various boxes stuck at the entry of the other elevator, thus for the outgoing box a shorter path along the belt is sufficient to reach the queue in front of it.

According to the settings to which the inverter is subjected, it takes approximately 81s for a box to cross the belt from the exit of one elevator to the closed entry of the other one, when that floor is totally empty. Every level can accommodate at most 9 boxes in addition to the one that lies on the platform; this means that when the platform releases a box, it is possible to save 9s of motor running for every box that is on the belt at that time. For example, if there are 3 boxes at the entrance of the elevator in the warehouse area and the platform in the opposite side releases a box, then the electric motor needs to drive idler pulleys only for 54s in order to let the outgoing box reach the other ones.

However, since there are two belts at different levels, it seemed reasonable to let them independently activate the motor; in other words, if there are two boxes leaving different elevators at the same time, then the idler pulleys of both belts will be driven for a duration sufficient to let the box on the less occupied floor reach the box that is closest to it.

Chapter 5

Conclusions

This thesis presented a detailed description of the procedures which were carried out in order to obtain the full automation of an industrial system made up of a power & free conveyor and a double level conveyor belt that are installed inside the laboratory “Ricardo Marin” in the School of Industrial Engineering of the University of Vigo.

Since the goal of the project was the automation procedure for two conveyance systems within the same plant, focused on the optimization of the functioning and on the vertical integration, considerations regarding mechanical and electrical aspects of both installations have not been treated in depth, although it would be interesting to analyze them because they match in a very complex way that surely required collaboration among technicians from different branches of Engineering.

Anyway, it is not even so easy to describe such an experimental project, in fact there is a huge amount of program lines that would be worth to be explained, but they would be boring to the reader for the complexity of the arguments involved in it. For this reason, the thesis tried to balance the mere description of the systems with the implementation of the strategies for controlling their operations: this choice is motivated by the desire to describe the installations in as much detail as possible and to understand why certain decisions were taken even if other ones could be valuable too.

At the beginning of the project both power & free conveyor and double level belt conveyor were already installed inside “Ricardo Marin” laboratory with electrical and pneumatic components. Anyway, to me it has been preferable to carry out the maintenance and review of equipment connections; as a matter of fact, a few sensors were improperly positioned and the addressing of AS-i slaves in the belt conveyor system was not the expected one. In addition, a mechanical issue arose in both railroad switches of the power & free conveyor causing a partial closure of the paths, and

so a technician replaced the air cylinders that were quite worn out. After being repaired, the cylinders did not behave anymore as bistable components, instead they started behaving in a monostable way: by default, they closed both railroad switches on the long path and so it became necessary to control them by a steady electrical signal to allow a cart to move along the bypass path. Motivated by such a drawback, the automation process needed to be revised.



(a) The power & free conveyor.



(b) The double level conveyor belt.

Figure 5.1: Final checks on both installations.

During the development of the new control strategy many videos of working sessions were recorded and various pictures of the components of both systems were taken; this precious material, together with the PLC programs and HMIs that were realized to control the conveyors, is now kept by Professor Julio Garrido Campos and it is be available for students and visitors of the Systems & Automation Division of the School of Industrial Engineering of Vigo.

5.1 Open Problems & future work

One of the most relevant goal in an industrial system, in addition to the production yield, is to ensure the safety of workers and products; hence, many dangerous situations have been considered while programming in order to cause alarms that stop the normal operation of the machinery and force an emergency mode.

Anyway, there is a situation in the conveyor belt that has not been treated with sufficient care and should be fixed by next students. At the moment, if one elevator¹ which is ascending with a box on its platform does not reach the higher floor within 10s, then an alarm indicates that a box is stuck along the vertical path. Here, the worker may do a potentially unsafe

¹Especially the one in the warehouse area, because of the system framework.

action; indeed, the emergency mode of the system does not contemplate the disabling of the solenoid valve that feeds the elevator, therefore a worker who manually tries to release the box could get seriously injured. It is fundamental to prevent this situation by driving a signal to let the platform move down when the alarm corresponding to a box stuck in the elevator is generated.

On the other hand, in the power & free conveyor, there could happen a deplorable although quite unlikely, event. It was already mentioned that the working area is able to accommodate at most three carts, since the fourth one would get stuck on the first railroad switch. This means that it could be impossible to change path settings because of a jam or, even worse, a cart coming from the loading station could fall because of a missing track at the first railroad switch. However, the implementation of stacks allows the supervision of the flow of carts between stations and therefore at any time instant it is possible to be aware of the number of carts that are waiting for in a certain station. Hence, it would be advisable to add a constraint that, at the loading station, allows to validate a cart for the long path if and only if the stack of the working station has less than three carts, otherwise the validation should be forbidden until carts start leaving the working area. Thanks to the PLC program structure, a modification of the code to address this issue would be very easy to perform.

Acknowledgment

At the end of this university career, longer than I expected it to be, it is the time for me to draw conclusions and thank people who believed in me.

One year ago I was leaving for Vigo, where I spent five months of my life within the Erasmus programme: certainly, it has been the most intense experience I ever lived and today I am proud to say that I feel a *ciudadano vigues*. Anyway, when I came back home I was a different person from the one who left with doubts and fears his small world: I improved few sides of my behaviour and maybe I worsened other ones, but surely I became more aware of myself and I realized that what really matters to me is my unchanged passion for engineering topics and issues, so I will keep facing them in the next future.

First of all, I want to mention my lovely mother Carla for being a shiny North Star especially during the darkest nights of my journey: it is her merit if I found the right way without losing the orientation. My father Gastone and my elder brother Filippo love me too, even though in their own way.

I am very grateful to my supervisors, Professor Maria Elena Valcher and Professor Julio Garrido Campos: they are cordial people as well as outstanding professors, both able to transmit their huge passion for Engineering. Professor Valcher was my supervisor also for the bachelor's degree and I chose her as Erasmus responsible even before choosing Vigo as destination: working with her was a pleasure due to her reliability and knowledge, moreover she is the only one who responds to mails at midnight and even later. Instead Professor Garrido hosted me within his laboratory and showed me his availability when I needed help in order to clear doubts or fix issues even though giving me freedom of action for the development of the project.

In Vigo I met a girl that turned out to be very special to me; despite of an intimate confidence between us, just as suddenly she appeared to me so she decided to follow her own way. However, she gave me love and motivation during various months and for this reason I will always carry Yaiza in the depth of my heart.

I want to thank Giancarlo Gaspani, whom I consider my mentor and source of inspiration: his intelligence is exceeded only by his cordiality.

Last but not least, I thank all the friends of mine who shared with me tiring but amazing mountain trails, tiny dorm rooms, lessons with ups and downs or poor meals: they grew by my side contributing to make me the person I am.

Appendix A

About the ADS protocol

A link with a socket network is needed for establishing the communication (namely, reading and writing of variables of the PLC program) among the IHM and the target controller.

Remembering that the ADS protocol is used, it is necessary to provide the ADS address (*AdsAmsNetId*) and port (*AdsPortNr*) of the PLC under consideration in order to get connected to it; these identifiers allow to detect any ADS device in the network. By default, the TwinCAT installation constructs the *AdsAmsNetId* from the TCP/IP address of the device with an extension of “.1.1” (2 additional fixed bytes) whereas the *AdsPortNr* differentiates sub elements of the connected device: for instance, port 800 indicates local process data (for the BC9050 only) and port 801 represents the first of four PLC runtime systems.

According to the ADS protocol, every variable of the PLC may be accessed of two parameters:

- *Index Group*: it indicates the group to which a variable belongs (for example, input signals or HMI variables);
- *Index Offset*: it reveals the byte offset of the target variable within a *Group* structure.

In order to get aware of *Group* and *Offset* indices, it is convenient to open to the file *ProjectName.tpy* using *Notepad* or *Wordpad* and then look for desired variables (according to their declaration within the file *ProjectName.prx*. Additional information can be found in [5].

Appendix B

Technical drawings of the power & free conveyor

This appendix collects technical drawings that were presented by two students from the Industrial Engineering School of the University of Vigo in their final projects [2] and [3]; they are herein reported with the aim of providing a detailed information about the power & free conveyor.

Figure B.1 shows the position of “Ricardo Marin” laboratory within the building that accommodates the Industrial Engineering School, whereas in Figures B.2 and B.3 the framework of the power & free conveyor under consideration is depicted.

Figures B.4 and B.5 show the mechanical components that feature the pulling system; they are designed in order to hook carts like the one which is indicated in Figure B.6.

Finally, the framework that characterizes stations and ensures them to stop carts is presented in Figures B.7, B.8 and B.9.

86 APPENDIX B. TECHNICAL DRAWINGS OF THE POWER & FREE CONVEYOR

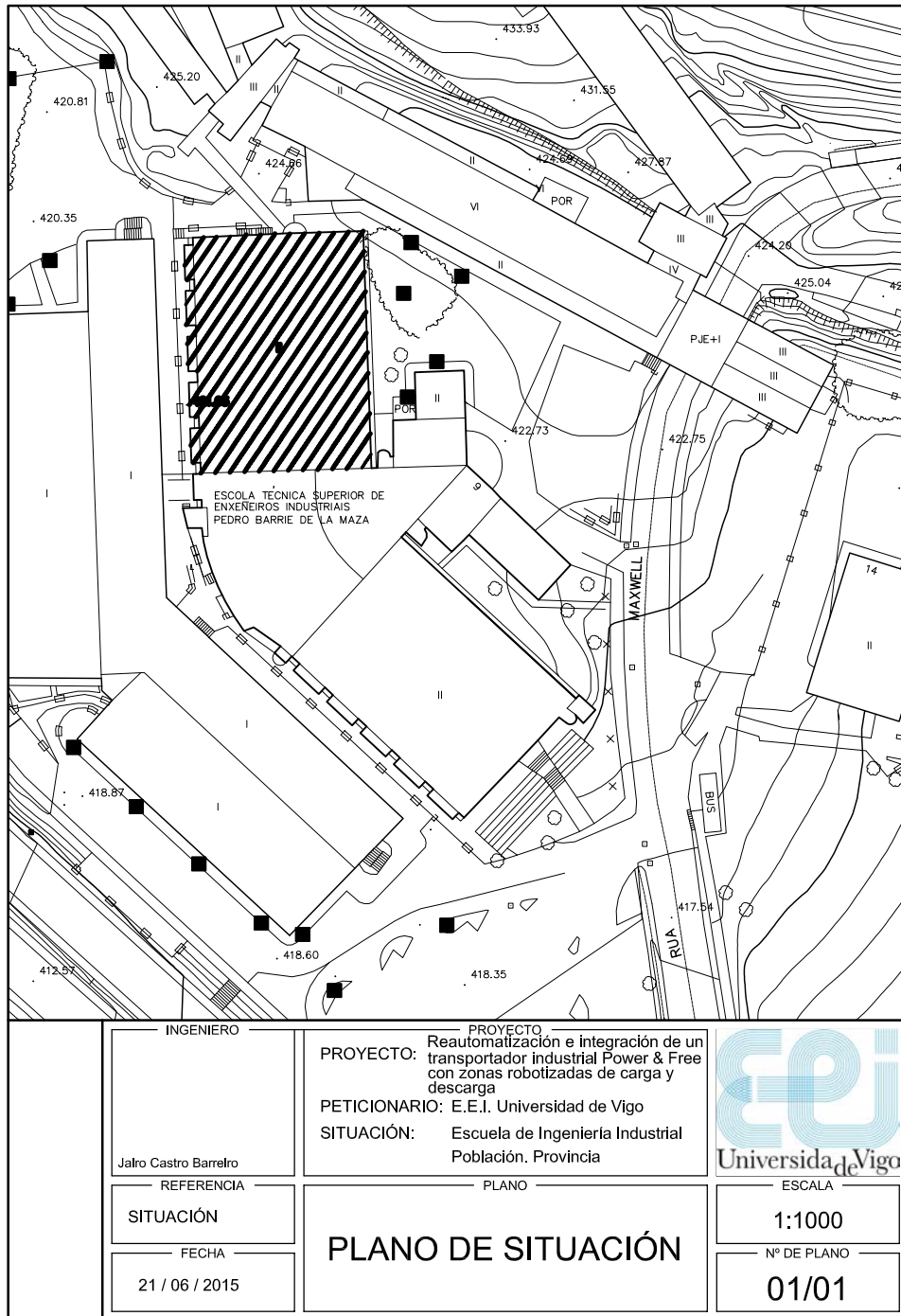


Figure B.1: The location of “Ricardo Marin” laboratory.

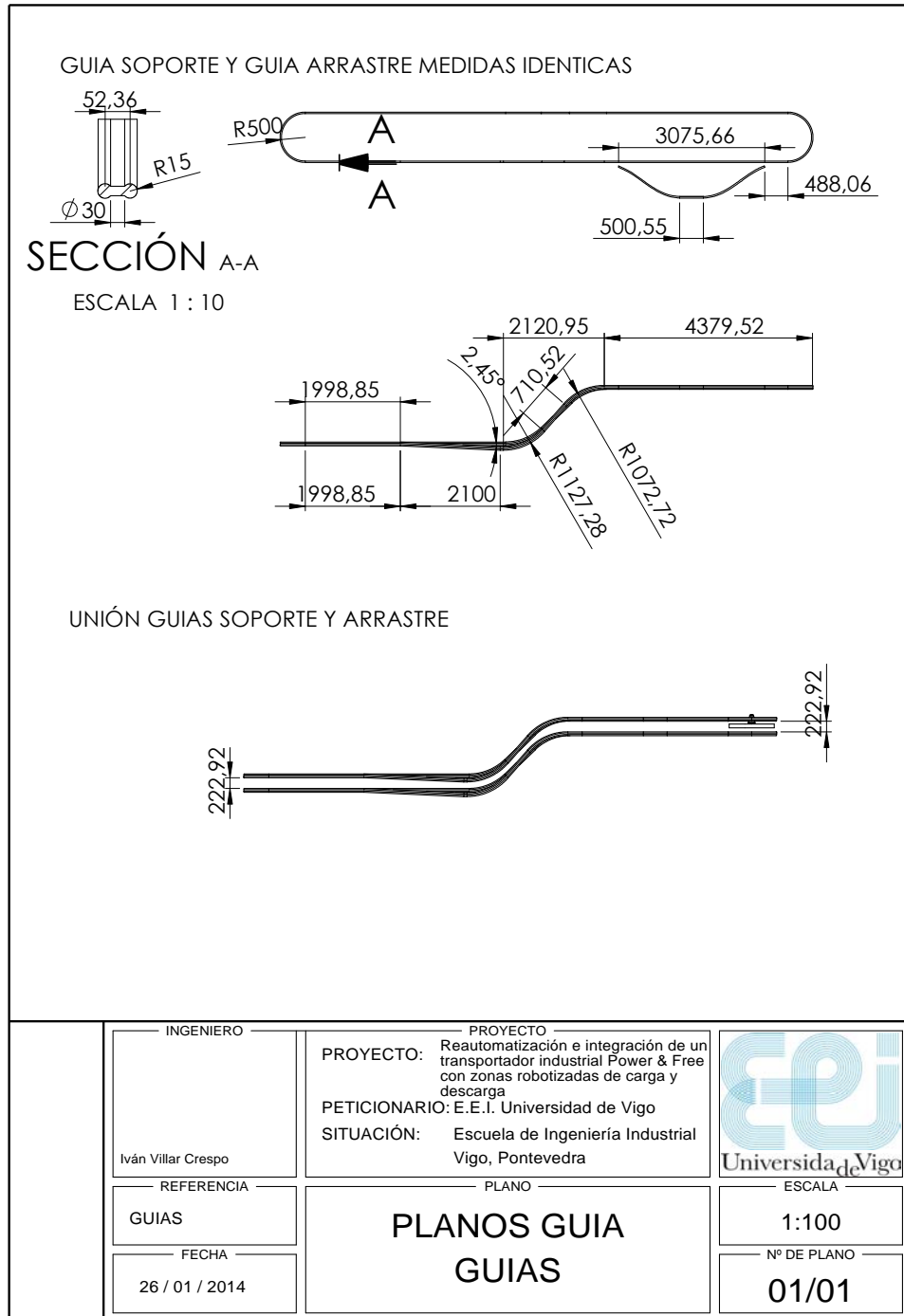


Figure B.2: The framework of the power & free conveyor.

88APPENDIX B. TECHNICAL DRAWINGS OF THE POWER & FREE CONVEYOR

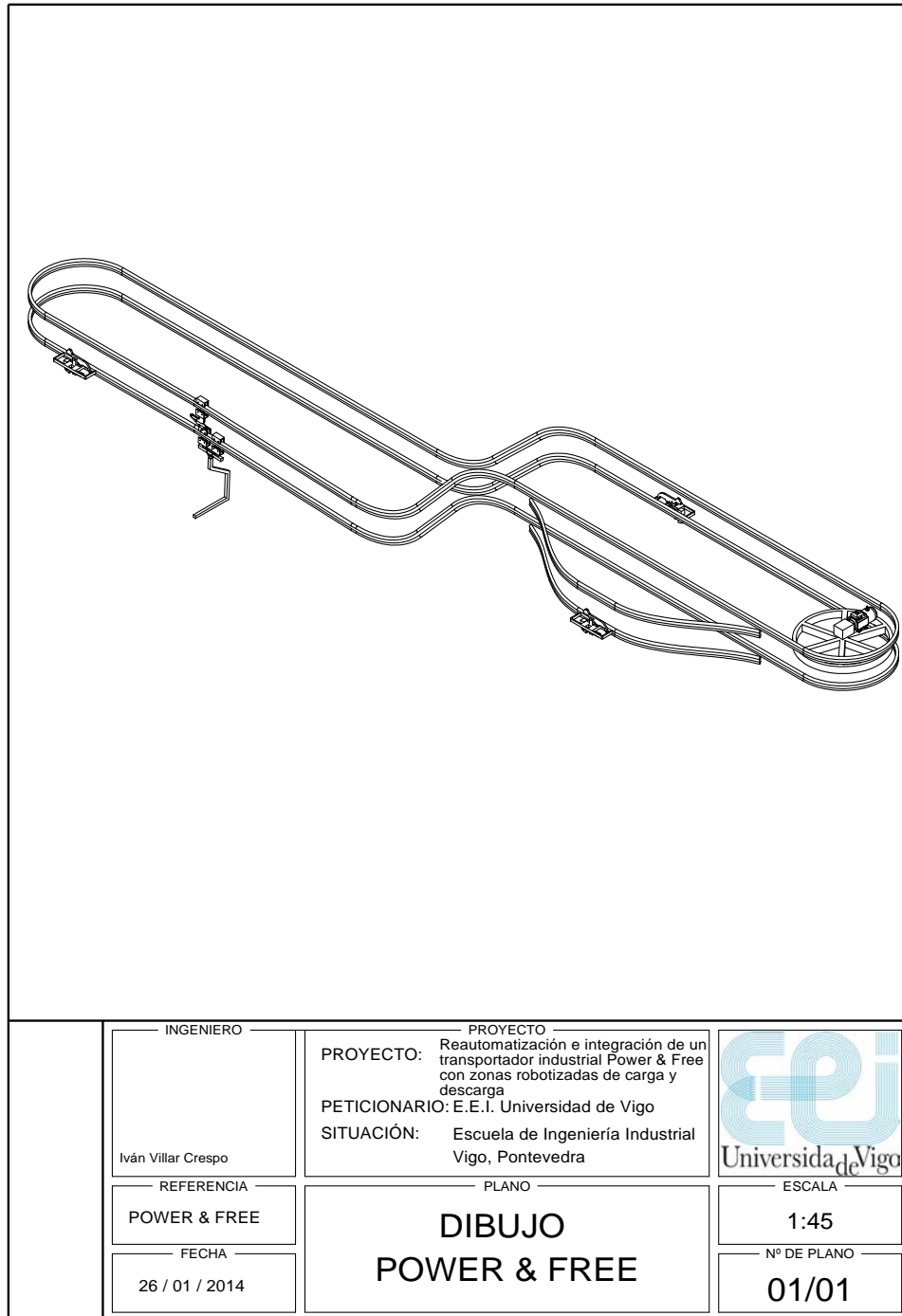


Figure B.3: Positions of the stations and pulling system.

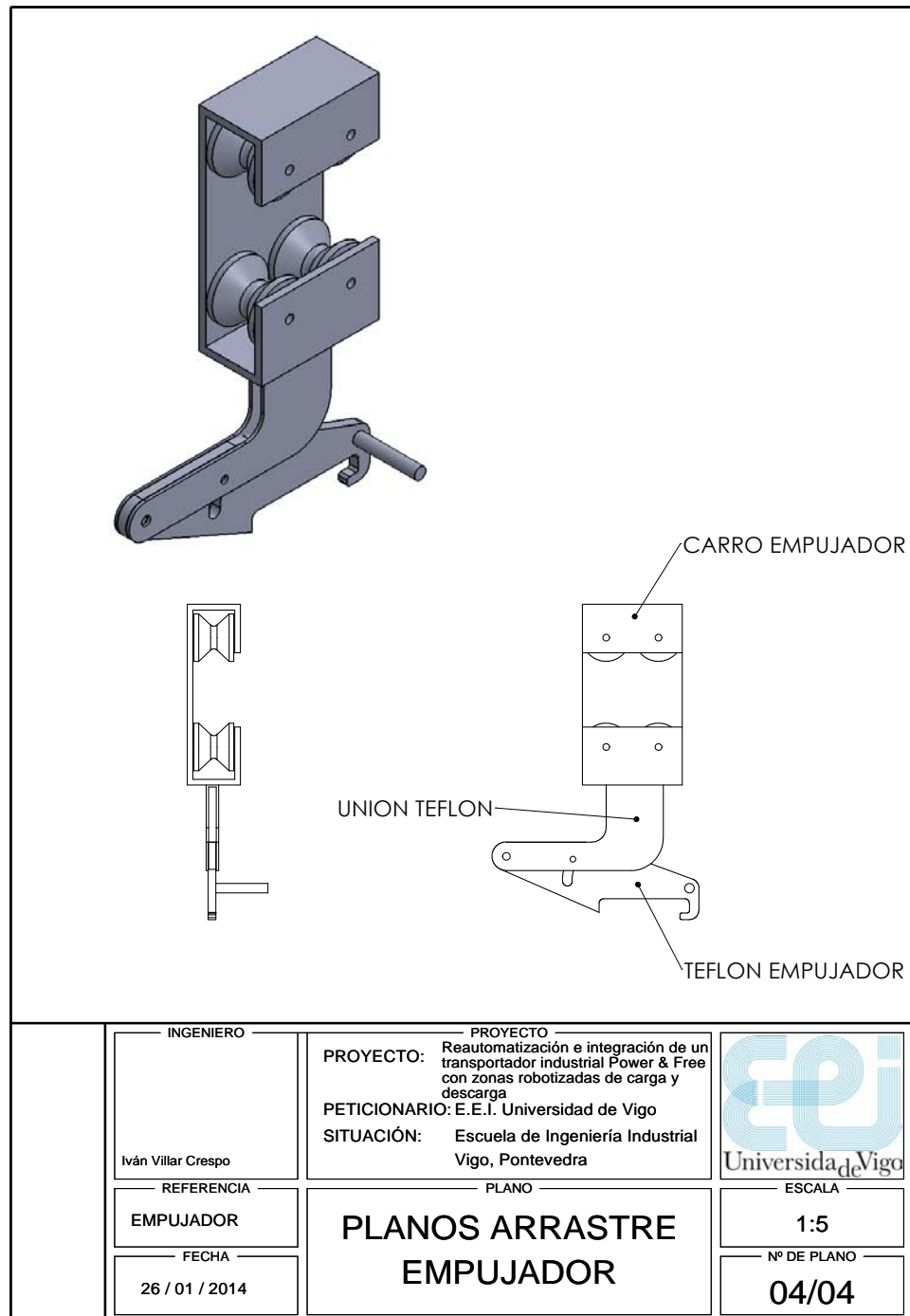


Figure B.4: A drag element of the pulling system.

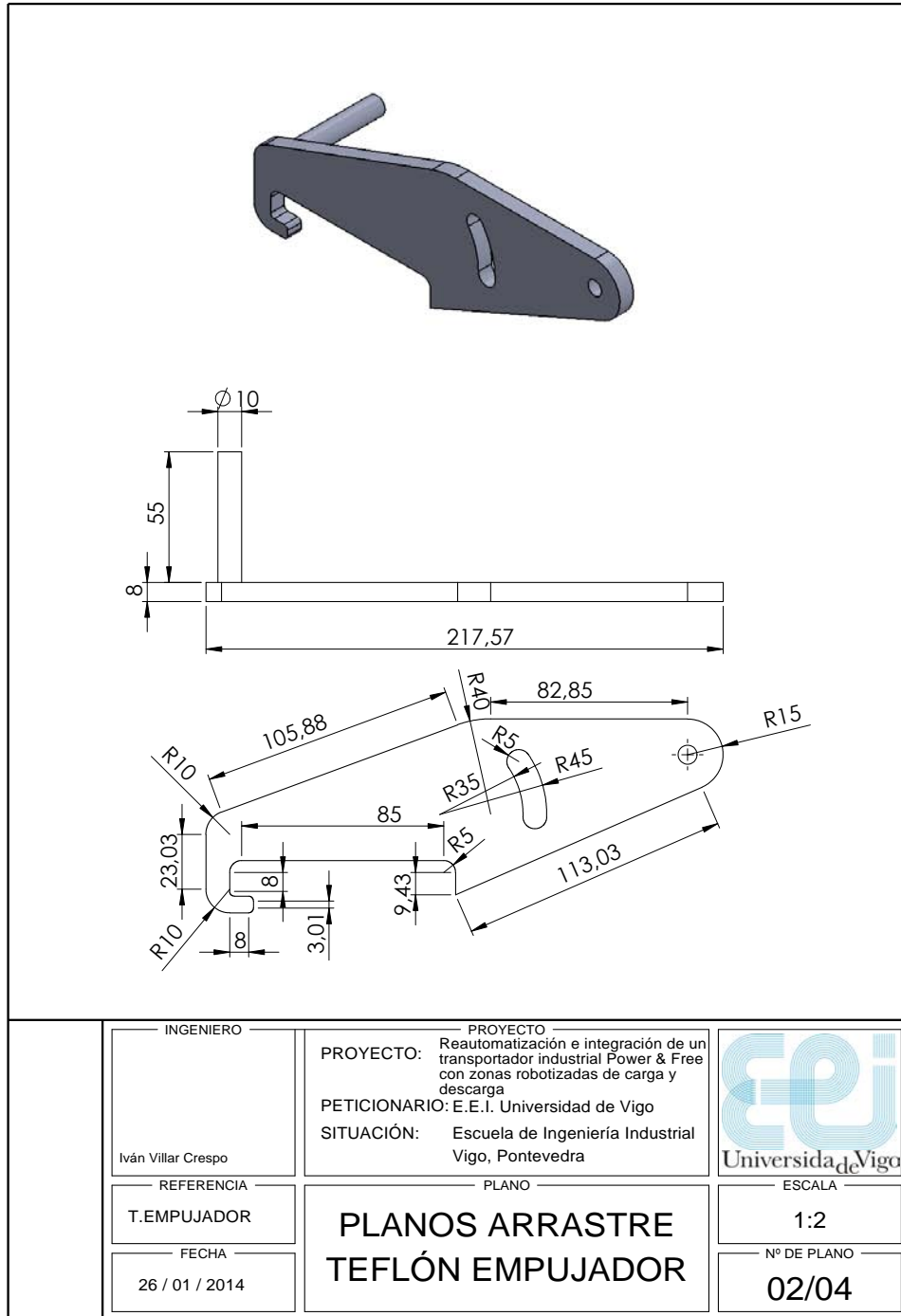


Figure B.5: The hook of a drag element.

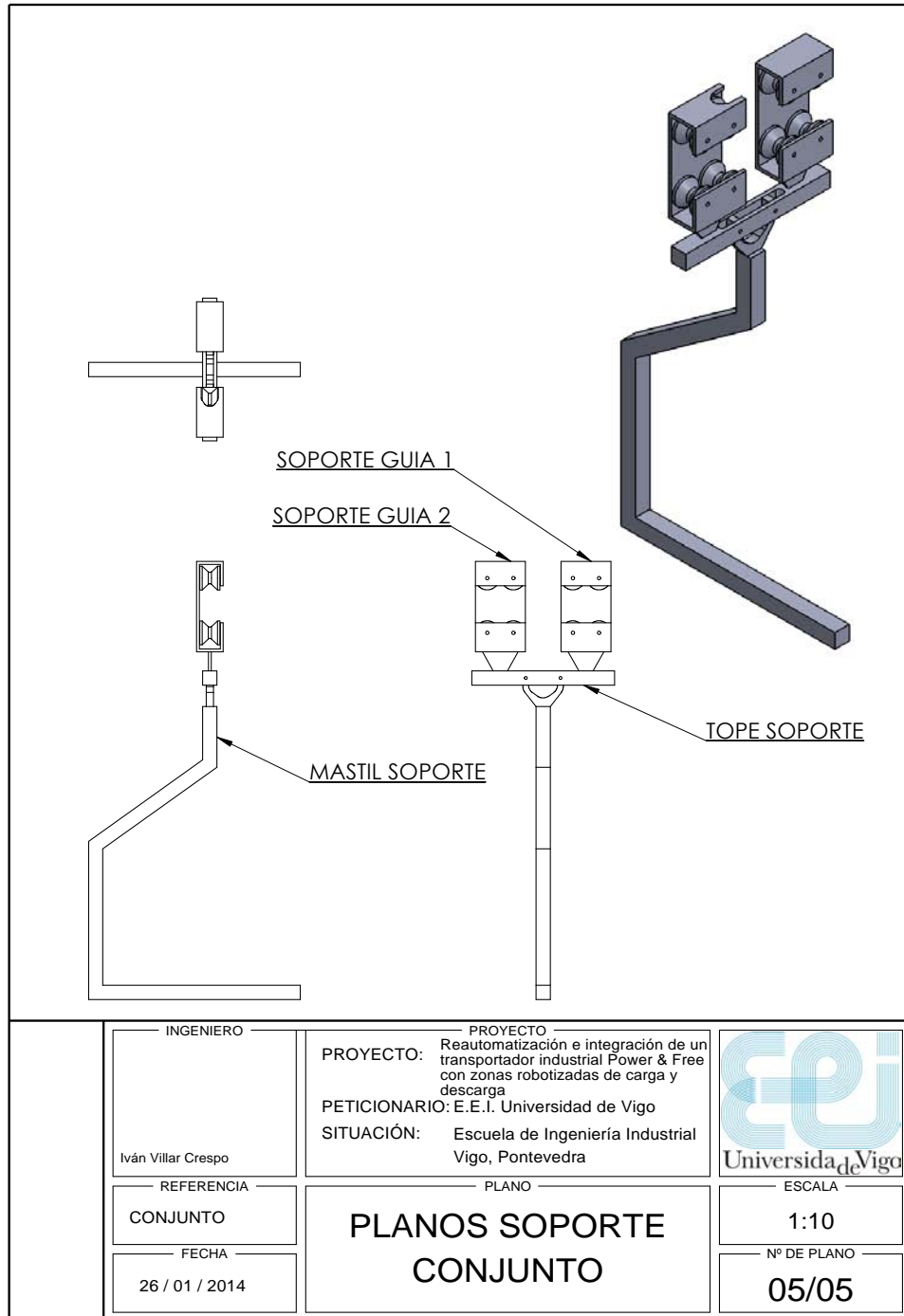


Figure B.6: A cart for the transportation of goods in boxes.

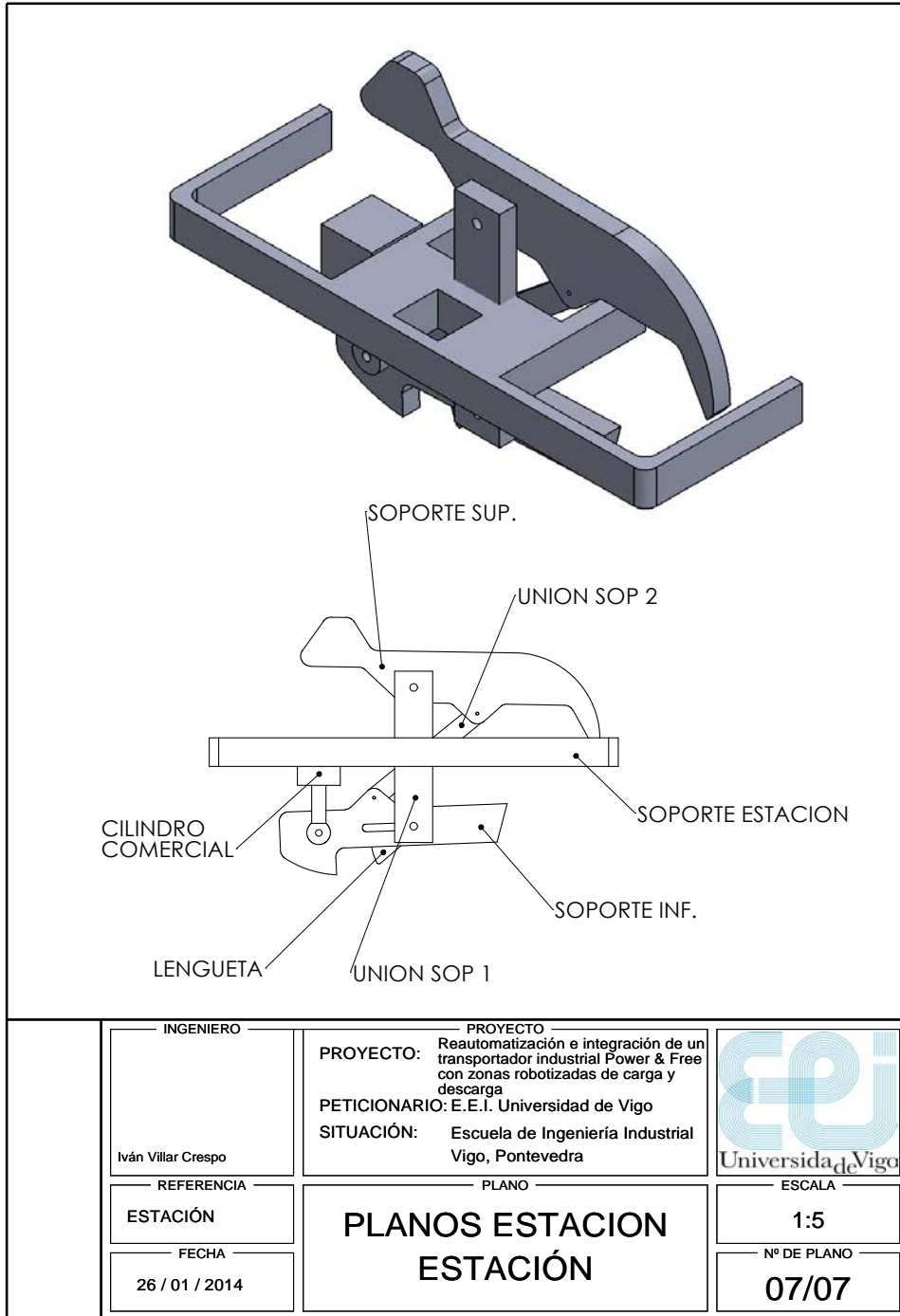


Figure B.7: The framework of a generic station.

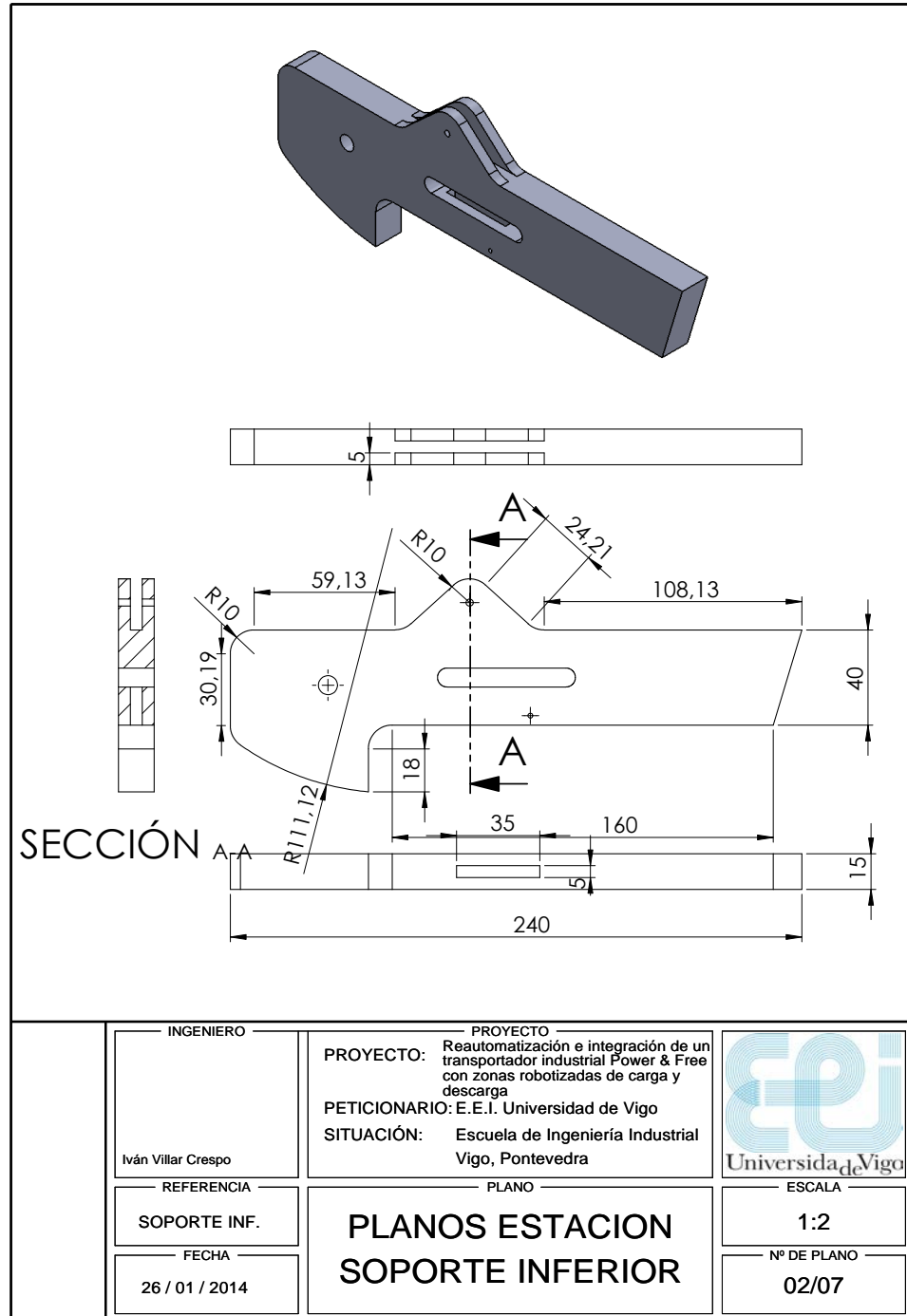


Figure B.8: The lower bracket of a generic station.

94APPENDIX B. TECHNICAL DRAWINGS OF THE POWER & FREE CONVEYOR

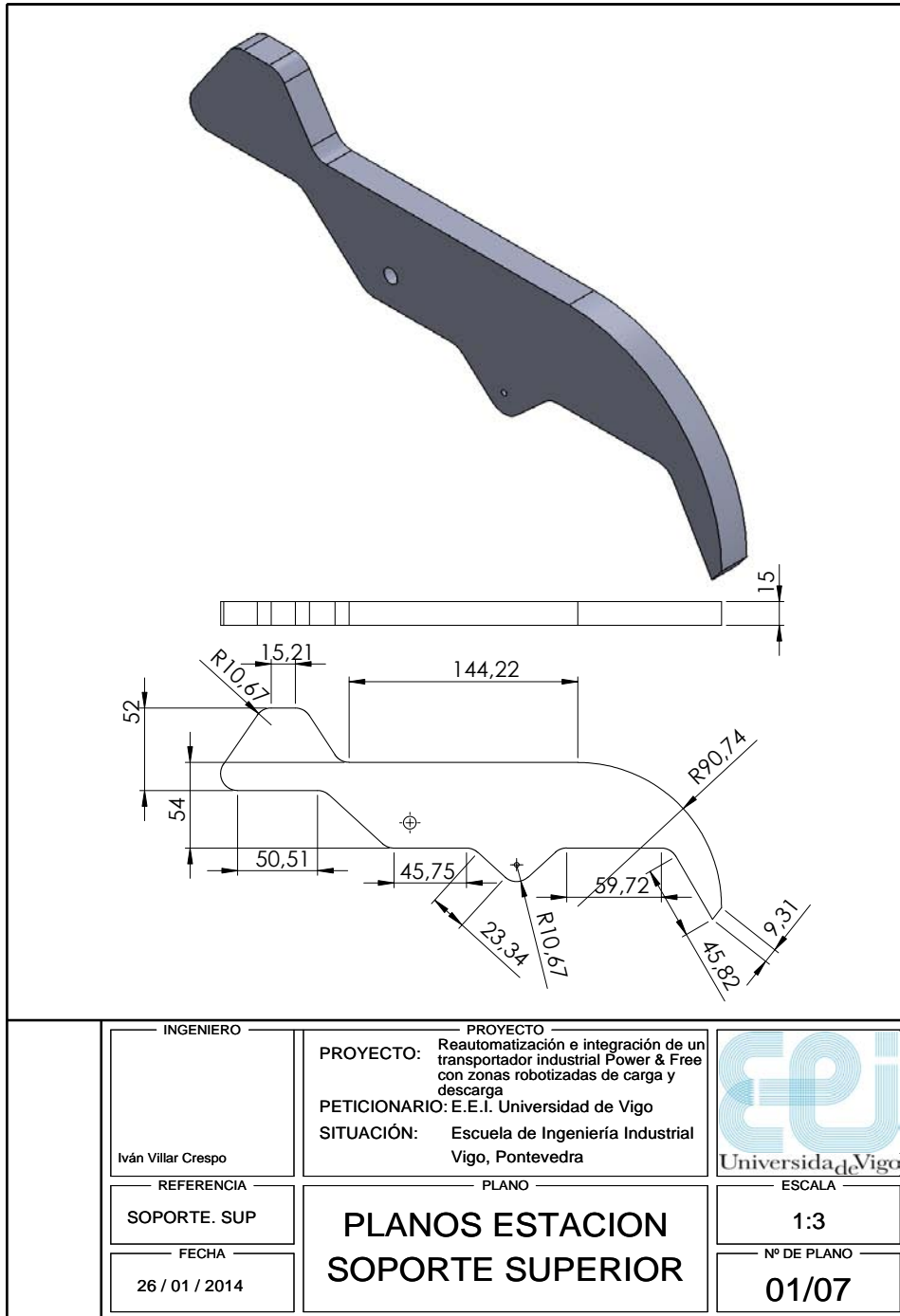


Figure B.9: The higher bracket of a generic station.

Appendix C

XML

Nowadays, the exchange of big amounts of data and information among computer applications may be managed in a structured way thanks to the **XML**¹ language, which defines a set of rules for encoding documents. The main advantage given by XML is the portability of textual data across the Internet; moreover, due to its simplicity of reading for both humans and machines, it is widespread in the configuration of applications for the Microsoft .NET Framework.

The characters that constitute an XML document are divided into **markups** and **contents**, depending on their assignment: usually, a markup is a single character identified with either “<” or “>” whereas a content consists of any string that is not a markup. A **tag** is defined as a markup construct that begins with “<” and ends with “>”; there exist start-tags (e.g. <content>) and end-tags (e.g. </content>) in order to define the body of a structured element that may be featured by children (nodes) and attributes.

Figure C.1 shows the XML file that configures the HMI that was realized with the aim to check and rule the operation of the power & free conveyor; it is saved in the path *bin/debug* from the directory that contains the Visual Studio project file.

The first two nodes regard the database related to the system under consideration: more specifically, they set the connection to the database thanks to few attributes and choose by default to disable its update². Other nodes are in charge of automatically establishing the link to the desired device, which is detected with the IP address 193.147.81.217, by means of Ethernet network.

Finally, with in mind the aim of realizing a detailed database, the XML document contains nodes regarding alarms, operating modes and manual

¹It is the acronym for eXtensible Markup Language.

²This is a strategy of the programmer that reduces the workload of the HMI in order to ensure better performances for what it concerns the navigation within the application.

commands; all them are individually classified in child nodes with an identificative number and a brief description as attributes.

```

<configuracion>
  <conexionBD>Data Source=AULA30\SQLEXPRESS;Initial Catalog=PowerAndFreeAula3;
    Integrated Security=True</conexionBD>
  <conBD>0</conBD>
  <proyecto_defecto>00001AA.xml</proyecto_defecto>
  <ip ip_defecto_activo="True" conexion_automatica="True">193.147.81.217</ip>
  <numeromaquina>1</numeromaquina>
  <alarmas>
    <alarma numero="1">fotocelula</alarma>
    <alarma numero="2">seta</alarma>
    <alarma numero="3">detector_par</alarma>
    <alarma numero="4">termico</alarma>
    <alarma numero="5">estacion</alarma>
    <alarma numero="6">seta_ihm</alarma>
  </alarmas>
  <modos>
    <modo numero="0">Stop</modo>
    <modo numero="1">Automatico</modo>
    <modo numero="2">Manual</modo>
    <modo numero="3">Alarma</modo>
    <modo numero="5">Arrancando</modo>
  </modos>
  <actuacionesmanuales>
    <actuacion numero="0">abrir_e1</actuacion>
    <actuacion numero="1">cerrar_e1</actuacion>
    <actuacion numero="2">abrir_e2</actuacion>
    <actuacion numero="3">cerrar_e2</actuacion>
    <actuacion numero="4">abrir_e3</actuacion>
    <actuacion numero="5">cerrar_e3</actuacion>
    <actuacion numero="6">liberar_e3</actuacion>
    <actuacion numero="7">bloquear_e3</actuacion>
    <actuacion numero="8">motor_on</actuacion>
    <actuacion numero="9">motor_off</actuacion>
    <actuacion numero="10">des1_bypass</actuacion>
    <actuacion numero="11">des1_trabajo</actuacion>
    <actuacion numero="12">des2_bypass</actuacion>
    <actuacion numero="13">des2_trabajo</actuacion>
    <actuacion numero="14">activar_cilindro</actuacion>
    <actuacion numero="15">parar_cilindro</actuacion>
    <actuacion numero="16">marcha</actuacion>
    <actuacion numero="17">paro</actuacion>
    <actuacion numero="18">seta_alarma_ihm</actuacion>
    <actuacion numero="19">rearme</actuacion>
    <actuacion numero="20">ack_error</actuacion>
  </actuacionesmanuales>
</configuracion>

```

Figure C.1: The XML document used in the configuration of an HMI for the power & free conveyor.

Another XML document configures the double level conveyor belt, although it is quite different from the above one since there is no historical production management; in addition, that system is automated with a low performance controller and so its configuration is focused on the correct addressing of input and output signals, whose procedure was treated in Section 2.6.

Bibliography

- [1] J. Garrido Campos, “Assembly Plant Prototype Based on Transportation and Warehousing Systems: Ricardo Marin Lab”, <http://gruposesa.webs.uvigo.es/RicardoMarinLab.html>, 2014.
- [2] I. Villar Crespo, “Actualización de Transportador Power & Free”, *Master’s degree thesis*, 2014.
- [3] J. Castro Barreiro, “Reautomatización e Integración de un Transportador Industrial Power & Free con zonas robotizadas de carga y descarga”, *Master’s degree thesis*, July 2015.
- [4] K. H. John and M. Tiegelkamp, “IEC 61131-3: Programming Industrial Automation Systems”, *Springer*, 2nd Edition, October 2010.
- [5] P. González González, “Automatización Conforme a la Normativa de Seguridad de Máquinas: Implementación según Modelo GEMMA”, *Master’s degree thesis*, June 2014.

Sitography

- [6] <http://www.beckhoff.com/english/default.htm>
- [7] <http://www.as-interface.net/>
- [8] https://www.beckhoff.com/english.asp?bus_terminal/k16201_k16211.htm
- [9] http://www.beckhoff.com/english.asp?embedded_pc/embedded_pc_series_cx1000.htm
- [10] https://www.beckhoff.com/english.asp?bus_terminal/bc9000_bc9050.htm
- [11] http://infosys.beckhoff.com/content/1033/BC9xx0/HTML/Bt_BC9xx0_FirstSteps.htm