TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING
Department's title

Development of a Palletizing & Depalletizing Station

Kaubaaluste peale- ja mahalaadimisjaama kavandamine

MASTER THESIS

Üliõpilane: Mesut Urcan

Üliõpilaskood: 184517MAHM

Juhendaja: Mart Tamre, Professor, TTU
Kristo Piir, Senior project engineer, ABB

Tallinn 2020

(*On the reverse side of title page*)

AUTHOR’S DECLARATION

Hereby I declare, that I have written this thesis independently.
No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

“.......” .... 202… .

Author: ....
/signature /

Thesis is in accordance with terms and requirements

“.......” .... 202… .

Supervisor: …........................
/signature/

Accepted for defence

“.......”....................202… .

Chairman of theses defence commission: ....
/name and signature/

Non-exclusive Licence for Publication and Reproduction of Graduation Thesis[1]

I, Mesut URCAN__________ (name of the author) (date of birth: 26/09/1992) hereby

1. grant Tallinn University of Technology (TalTech) a non-exclusive license for my thesis Development of a Palletizing & Depalletizing Station _____________________________,

(*title of the graduation thesis*)

supervised by

Mart Tamre____,

(*Supervisor's name*)

1.1 reproduced for the purposes of preservation and electronic publication, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright;

1.2 published via the web of TalTech, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright.

1.3 I am aware that the author also retains the rights specified in clause 1 of this license.

2. I confirm that granting the non-exclusive license does not infringe third persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

---

[1] *Non-exclusive Licence for Publication and Reproduction of Graduation Thesis is not valid during the validity period of restriction on access, except the university`s right to reproduce the thesis only for preservation purposes.*

_______________ (*signature*)

_______________ (*date*)

TalTech Department`s title

THESIS TASK

Student: Mesut Urcan, 184517..................................(name, student code)

Study programme, main speciality: Mechatronics ....(code and title)

Supervisor(s): Professor, Mart Tamre...........................(position, name, phone)

Consultants: Kristo Piir, Senior project engineer.......(name, position)

ABB...(company, phone, e-mail)

Thesis topic:

(in English) *Development of a Palletizing & Depalletizing Station*

(in Estonian) *Kaubaaluste peale- ja mahalaadimisjaama kavandamine*

Thesis main objectives:

1. Feeding a gluing & thermal processing system with HDF boards by depalletizing pallets with an industrial robot
2. Palletizing processed HDF boards by using another industrial robot

Thesis tasks and time schedule:

No	Task description	Deadline
1.	Depalletizing HDF boards	June 2020
2.	Palletizing ready products	June 2020
3.		

Language: English **Deadline for submission of thesis:** "23" Dec 2020 .a

Student: Mesut Urcan .... "22" Dec 2020 .a

/signature/

Supervisor: Mart Tamre .... "22" Dec 2020 .a

/signature/

Consultant: Kristo Piir .... "22" Dec 2020 .a

/signature/

Head of study programme: Mart Tamre .... "22" Dec 2020 .a

/signature/

Terms of thesis closed defence and/or restricted access conditions to be formulated on the reverse side

CONTENTS

List of abbreviations and symbols

CS	Capacitive Sensor
DI	Digital Input
DO	Digital Output
DOF	Degree of Freedom
GI	Group Input
GO	Group Output
HDF	High Density Fibreboard
HMI	Human Machine Interface
ID	Identification
IO	Input Output
IP	Ingress Protection
IRB	Industrial Robot
IRC	Industrial Robot Controller
ISO	International Standards Organization
ISV	Vacuum Security Valve
LED	Light Emitting Diode
NC	Normally Closed
NO	Normally Open
OS	Optical Sensor
PLC	Programmable Logic Controller
PP	Program Pointer
Prod	Product
PVC	Poly Vinyl Chloride
SCARA	Selective Compliance Articulated Robot Arm
TCP	Tool Centre Point
TROB1	Task Robot-1
Vac	Vacuum

1. INTRODUCTION

Pick and place applications are widely used in industry for collecting or splitting various kind of products. Robotic solutions in the industry are becoming more popular every day and those solutions help to make a process faster and more efficient compared to manual labour.

The project is done for a company called Harviker. Harviker manufactures solid wood furniture. The company was established in 1995 and started production in 1997. Mainly used materials are pine and birch. The main products are home furniture - various tables, cabinets and drawers. Its partners are based in the Netherlands, Belgium, England and Sweden.

The main target of the project is building two partially autonomous robotic cells which can depalletize wood boards to a gluing & heating machine and, palletize the final products which are fed to the second robotic cell by the same machine after the products are processed. The purpose of the project from ABB side is mitigating human interaction to accelerate the processing time.

The worker in the area provides full pallets to the first station for the depalletizing process. The robotic cell must pause working when there is any human interaction. When human interaction is done, the robotic cell can continue operating automatically. Furthermore, the station is expected to understand existence/absence of pallets (for instance only the left pallet is present) and also to check if there is any human on or around the pallets. This operation is controlled by area scanners. The first and the second stations are able to understand the status changes in pallet slots.

Moreover, if there is a new pallet in the depalletizing station, the robot searches the height of the pallet at the beginning of the process. This process is maintained by an optical sensor. The product thickness is provided by an operator at the beginning of a process. The product type information is delivered by the heating-gluing machine for both stations. To prevent feeding several layers of products at once to the depalletizing station's conveyor, the capacitive proximity sensors were installed to the gripper. In addition to this, a pneumatic by-pass line is installed to make picking wood board process easier with different thickness values. In addition to this, additional optical sensors were installed to check the presence-absence of boards in case of any failure during the picking/transferring process.

What is more, the robotic cells should indicate important parameters like human interaction, pallet status, internal errors etc. by both audial and visual alarms to improve safety conditions and productivity. Additional light beam application was built to control the human interaction where the area scanning is insufficient. An HMI with two tabs on robot controllers was designed to improve manipulation capability and awareness. The first tab is used to control manual actions. The second tab contains process information saved by the robot. A pneumatic control system was installed to control vacuum generators. It enables handling of different sized products with activating different suction pad groups on the grippers. The robot communicates with the thermal processing machine to understand when it shall start or stop feeding the panels.

The working principle of the second station is almost the same as the first station. The main difference is the second station is a palletizing station. There is no height search in this station because the pallets provided to the station are empty. There is no additional control for the product thickness because the final product thickness is the same for all variations. The HMI designed for the second station has also some different functions in both control and process information tabs. Additional features like controlling pallet level, an indication of pallet height etc. are added to increase the manipulation capability.

The demand for an automated robot station is requested to accelerate the processing time, to mitigate the human interaction (error) and to maintain a standard productivity rate. It is expected that the station does not stop functioning completely when there is interaction with the station.

2. LITERATURE REVIEW

2.1 Scope of the work

It is possible to see different approaches to the solution expected from the customer since the pick and place application is a widely used method in industry. It is important to review previous similar solutions to make progress in the project. Before starting the literature review, the main problems can be summarized as:

- A gripper design to pick the product with different sizes
- Decision of what kind of grabbing mechanism will be used
- Separation of multi-layers (thin boards stick to each other)
- Detection of material thickness
- Detection of stack height
- Human interaction to the station
- Detection of picking multi-layers or a missing layer

The layout and process flow were given given by the customer. ABB was asked to provide the tasks given above. The existing solutions shall be discussed in the following sections.

2.2 Gripper design solutions

There are eighteen product types with different length, width and thickness values. Number of the products in one layer varies depending on the length and width of the products. The minimum number of products in one layer is two, and the maximum number of products in one layer is six. This does not seem like a big issue when depalletizing process is performed, but it may become a problem when the products are picked from the second conveyor line. Two products are fed to the second conveyor line each time. Depending on the size of the product, the area that product covers change. That means, the gripper must customize itself to pick different combination of products.

There are many solutions in industry with different ready-made designs of grippers. The companies like Schmalz, OnRobot and Destaco offers pneumatic end effectors to handle with wood products [1], [2], [3].

A research run by a group of academicians is related with the problem that the project refers. The gripper mentioned is able to grip different products thanks to its valves which are automatically activated when the gripping surface and the product are physically contacted. Namely, the valve is autonomously switchable between open and close states. The principle depends on avoiding air leakage on unused suction pads. As the valve becomes small, the gripper can cope with higher step and higher curvature. The absorbed weights of the large-scale and miniature grippers were 19.6 N (2 kgf) and 120 mN (12 gf), respectively. This gripper is limited to use in less dust environment, because the dust fills in the valve [4].

Another team of academicians made a research about a gripper which is capable of changing its gripping positions. The article refers that it is hard to pick different shaped and sized products with fixed positions of suction cups. To solve this problem, a novel vacuum-suction gripper has been designed to adjust the positions of suction cups as requested. Also, it is stated that, a novel algorithm was developed to find optimal positions of gripping points. The paper refers that, when the gripping points are fixed, the resultant of suction forces of multiple suction cups will be probably deviated from the centre of gravity of the workpiece with some specific patterns. In addition, it will reduce the gripping stability. That is why, it was decided developing a system capable of changing its gripping points. It was seen that, gripper performed a good performance on given patterns, but the algorithm was not successful in adapting itself to some patterns [5].

Palletizing and depalletizing processes are mostly maintained by the pneumatic systems. Pneumatic systems provide low cost solution comparing to electrical installations. There is another method called electro adhesive gripping which is working by the astrictive force produced in the electrostatic fields and is suitable for electrical conductors. As for electrical insulators, it is difficult to making precise calculation of the astrictive force of electro-adhesion [5],[6]. Electro adhesive handling provides stable grasp while avoiding the application of excessive gripping force. The adhesives sustain large shear forces with very low pressure. They also release objects without residual adhesion when the grip is relaxed. [7]. The electro-adhesive grippers provide grabbing products piece by piece, it could be very important separating very thin materials (like a paper sheet) [6]. The adhesive allows the manipulator to pick up and release flat objects and offers several advantages over traditional vacuum-based grippers. These include no external tubing or support equipment, as required by suction cup grippers commonly found in factories, and the ability to be used for space applications where vacuum gripper usage is precluded. Moreover, while the adhesive does require power to engage, the power consumption is extremely small relative to a vacuum gripper [8].

The rubber vacuum sucker is an effective tool for gripping a sheet workpiece. However, it is limited to gripping flat and smooth workpieces. With a coarse workpiece, the pressure difference between the negative pressure inside the rubber sucker and the external atmospheric pressure causes reverse flow of air into the sucker through the gaps on the workpiece surface, causing the negative pressure inside the sucker to be lost and the vacuum absorption to become ineffective. Furthermore, dust, fragments, residues, and food juices can easily be absorbed into the rubber sucker, causing the blocking of the vacuum tube and the failure of the vacuum source [9].

2.3 Detection of product thickness

Measurement of product thickness is one of the challenges in the project. It is requested for the depalletizing station. The panels sent to the thermal process machine must be sorted to prevent damaging the gluing roller. Different measuring methods are available in industry. There are important parameters to be considered such as precision, cost, covered area etc.

A contact type optical sensor is used to measure wooden block thickness in a research made in India. The traditional contact type thickness measuring sensors are inductive type, capacitive type and ultrasonic type. This measurement technique has some advantages and some drawbacks according to their measuring principles and environmental effects. In case of inductive type transducer sensitivity is truly affected by stray magnetic fields and severely dirt and other contaminants may produce difficulty for using capacitive type transducer. Beside this, ultrasonic vibration transducer avoids those physical phenomena, but its dynamic reading is not suitable for moving materials. For this shortcoming, a new type of opto-imaging thickness gauge has been designed & fabricated. Working principle of the sensor is relying on a grayscale image plate and a shaft which is centrally connected to this plate. Shaft is covered with a closed PVC pipe and the shaft is able to move inside of the PVC pipe. There is a roller connected to bottom of the shaft. The roller is the touch point of the system which is able to roll over on the wooden block. Shaft is moving inside of the pipe because of the change in product thickness and there is a spring attached on top of the shaft to compensate the movement. To sense the shaft movement, an optical trans-receiver is placed in front of the imaging plate. When a wooden block sample move under the roller, the shaft connected imaging plate is displaced towards upward direction in front of the optical trans-receiver. Due to which variation of light absorption by imaging plate, the reflected light intensity would be varied accordingly & converting it into voltage level utilizing signal processing circuit. After getting this voltage level it is mapped into a reference voltage to displacement table. Then the thickness of the material could be measured accurately [10].

Industrial application of optical sensors can be shown as an example. Two optical sensors (MiniOD, SICK) can be positioned on a vertical axis in a opposite way. The wood panel thickness can be identified precisely using the difference of measurement data of optical sensors. Sensors have the range of 1000 mm [11], [12] The range of the sensors are suitable for the application since the thickest wood panel is 32 mm.

2.4 Separation of multi-layers

Handling with thin wood panels could be harder than it is expected since the wood veneers have porous structure. Porous structure may allow picking two or even more panels at once. Furthermore, unstable material density could cause the same issue. There are some industrial applications to prevent such situations.

One industrial application can be shown as a good example. The company Schmalz invented a suction cup that permit particularly careful handling of the workpiece. The floating suction cups float on an air cup above the workpiece and facilitate low-contact handling. In addition, the floating suction cups prevent through-suction of the workpiece so that thin, porous workpieces can also be separated with no problem. The specially arranged suction points prevent damage to the material and ensure careful handling [13].

2.5 Detection of stack height

Stacking height control is needed to prevent possible crashes and time waste during the process. Different stack heights could be observed depending on the previous production numbers. To avoid this situation, an application which is able to detect the distance between the robot gripper and the stack is needed.

A study is done by a group of researchers to track and grab objects with a 4DOF robot elbow. In this study, an investigation of mobile robot manipulation applied stereo vision system, tracking and gripping the object with different disciplines to produce a robotics system with abilities to perform a complex real task is conducted. At first, by using eye-to-hand camera, robot finds the position of the target and move straight ahead; using signal from stereo cameras robot will know whenever it is near the object enough, which means that target is in the workspace of the manipulator. After finish moving, mobile robot begins tracking the target. It also measures the distance from object to the cameras. After that, functions transform from world to the manipulator coordinate through camera coordinate which helps controller know the target is inside the workspace of the arm or not. Finally, it is solved the inverse kinematic functions, robot will move and grip the target. The system consists of the following main parts: detection of the target, moving by using eye-to-hand camera, tracking the target and measurement distance by using stereo visions and solving the inverse kinematics and gripping. Colour detection by applying threshold is used to detect parts. After detection, robot moves against the part. Computer decides the optimal gripping position by using the stereo camera and stops the robot. Then picking process starts [14].

2.6 Human interaction to the station

Safety is an important consideration in human-robot interactions. Robots can perform powerful movements that can cause hazards to humans surrounding them. To prevent accidents, it is important to identify sources of potential harm, to determine which of the persons in the robot's vicinity may be in greatest peril and to assess the type of injuries the robot may cause to this person [15].

Four by Three project is aiming to measure the trust of workers on fenceless human–robot collaboration in industrial robotic applications as well as to gauge the acceptance of different interaction mechanisms between robots and human beings. Since December 2014, the FourByThree2 Project ('highly customizable robotic solutions for effective and safe human–robot collaboration in manufacturing applications') is developing a new generation of modular industrial robotic solutions that are suitable for efficient task execution in collaboration with humans in a safe way and are easy to use and program by the factory worker. The system also includes space monitoring using a projection and a vision system, which provides the information needed to modify the velocity of the robot according to its relative distance with respect to the worker. The safety approach is centred on the design of the actuators with the capability to monitor the force and torque in each one, which provides the opportunity to implement variable stiffness strategies and reactive behaviour in case of contact/collision. [16].

Laser scanners are extremely popular in industry to detect human interaction. Human-robot coexistence and interaction is examined in the following article. Usage of the laser scanners can be shown as an example. Two laser scanners (KEYENCE SZ-V32n) placed on two opposite corners of the robot cell, about 50 cm from to the floor (at the calf height). With the proposed placement, the laser scanners define invisible planes that detect the presence of moving or standing humans inside the cell. Through a dedicated interface, it is possible to program which are the elements in the cell that should not be identified as human intruders (e.g., the robot, the sliding plate on the track, and the working table) [17].

2.7 Detection of picking multi-layers or a missing layer

Panel handling is one of the important cases in the project. Picking more than one layers at once or missing one or more layers may affect the height of the panel stack and it could lead bigger problems as the process proceed. Also, using more than two layers will damage the thermal heating machine’s gluing roller.

A research is made to investigate increasing effectiveness of an end-effector by integrating multiple sensors. The present work deals with the development of a multiple sensor integrated robot end-effector which can be gainfully used for product assembly in industries. An experimental set- up was made to carry out mechanical assembly using a SCARA robot. A force/torque sensor, capacitive proximity sensor, inductive proximity sensors, Ultrasonic sensors and tactile sensors were mounted on the wrist of the robot to facilitate the identification of correct part and to perform the desired joining operation either through pushing a part on to another or through screwing. Two proximity sensors both capacitive (Model: CR30-15DP) and inductive (Model: E2A-S08KS02-WP-B1-2M) are mounted on the robot gripper to detect the presence or absence of any object [18].

The UD18-2 checks for double layers and splices using ultrasonic technology. The sensor is able to determine whether one, two or no material layers are present between its sender and receiver. The UD18-2 can reliably detect objects regardless of material, including paper, cardboard, shiny metal, transparent plastic, and more. The UD18-2 also features the capability to teach-in up to four sensitivity levels, and the sensor can switch between sensitivity levels during operation. This allows the sensor to tackle complex applications and ensure permanent system availability with consistent production quality [19].

2.8 Conclusion

Different technologies and systems are used to solve the problems stated above. All the systems have their advantages and disadvantages. Price, spacing, accuracy, weight and so many other factors must be considered before applying the solution. The applications stated above could be used as a guideline, but also new approaches can be developed during the installation of the line.

As it can be seen from the many industrial projects, timing is an important parameter. Delivery time is an important factor but also engineering time is as important as the delivery time since its cost is extremely high. Timing is one of the biggest challenges in the project and it can be expressed as the first objective of the project.

There are eighteen product types and according to their dimensions, desired productivity rate changes. The productivity rate is affected by how many pickings or placings are made during picking or placing one layer. Product weight is also an important factor since the carrying speed of the robots depends on this parameter. According to the optimized speed and pickings/placings, a certain amount of productivity rate has to be maintained. System halt is an important factor what affects the productivity. Intended or unintended halts has to be minimized to enhance the productivity.

Gripper design and decision of gripping system should be made according to the material type. As it can be seen from the quoted references, pneumatic systems are widely used in wood industry. Depending on the product dimension and weight, surface area and suction power parameters shall have the highest priority and design and decision shall be made according to these parameters.

Product thickness, stack height detection, multi/missing layer detection can be maintained with the optical and capacitive sensors since they have enough precision and, they are cost effective. Number of these sensors have to be at least twelve since the six of them shall be used for multi-layer detection, six of them shall be used for presence detection and one of the six optical sensors shall be used for height detection. Calculation was made according to the maximum (six) product number on one layer.

Human interaction can be detected with area and laser scanners since they are highly used in industry for this purpose. Comparing with camera systems, they have lower prices and easier to use. The type of the scanners shall be decided according to their response time and range.

3. GENERAL DESIGN OF THE STATIONS

3.1 System layout

Initially, taking the system as a whole would give a better understanding of the processes. The robotic stations take place at the beginning and at the end of the process as an initiator and a terminator. Since the tasks of the robotic stations were described in detail so far, the rest of the system would be briefly described in this section.

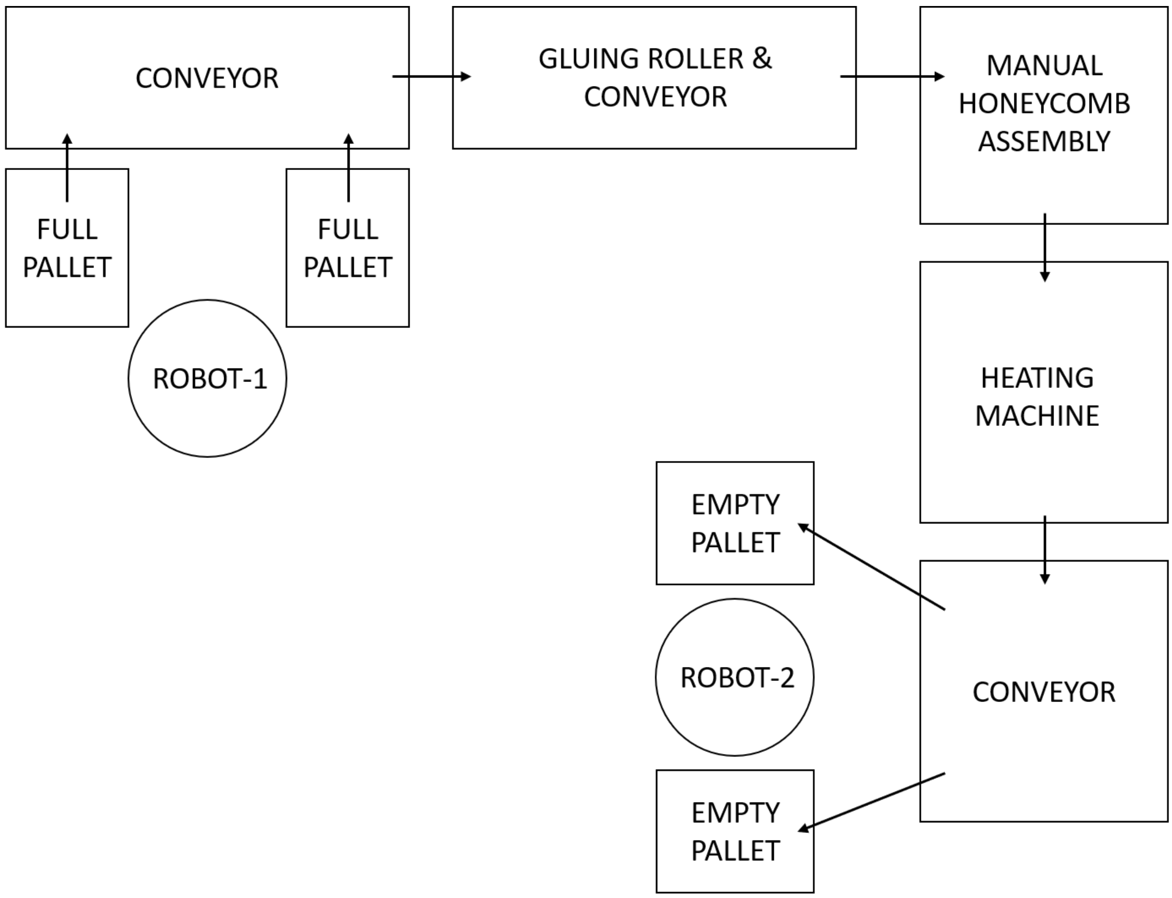

Figure 1. System layout

After first process, which is depalletizing of wood boards, is done, the boards go into the gluing process. There is a rolling cylinder right after the first conveyor which applies the glue to the board surface. The cylinder height is adjusted manually according to the boards thickness. Two boards are glued at each turn. After the cylinder, boards are transferred to the manual assembly area over another conveyor. This conveyor has sensors which sense the boards and their positions to avoid product collision in assembly area. In assembly area, a conveyor with a tilting table delivers the boards to the operators for the assembly of honeycomb structure in between the glued boards. After the assembly process, the ready products are placed on a belt conveyor which transfers the products to the oven. The products are sandwiched and heated in two floors of an oven for a certain amount of time. The time can vary up to product type and dimensions. After heating process, products are transferred into a buffer area to be stocked and transferred to the palletizing station. The buffer area again has two floors to increase stocking capacity and avoid additional waiting time. Each floor sequentially serves two ready products in one turn to the palletizing station. The final roller conveyor has belts in between the rollers to adjust the product position precisely for the robot. Belt system repositions the final products at the corners of the conveyor and the robot picks the products and palletizes them.

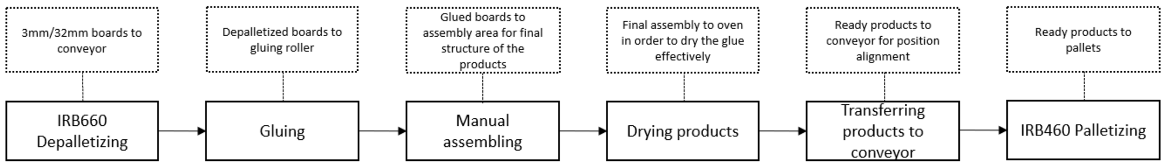

Figure 2. Process flow

3.2 Product specification

3.2.1 Product description

Products are HDF wood products. Their density varies between 500 – 1040 kg/m3. It has a strong structure because it consists of highly compressed wood fibres. It is dry processed, so it has a smooth surface on both sides. It is used in manufacturing furniture. In this particular application, both side of a honeycomb is overlaid with glued veneer by hot press.

3.2.2 Product dimensions

There are eighteen different product types in total. The products can be gathered in two different groups. The only difference between these groups is the product thickness. First group has 3 mm thickness and the second group has 32 mm thickness. Product length and width values are as in the following table.

Table 1. Product groups and dimensions

	Prod 1 (mm)	Prod 2 (mm)	Prod 3 (mm)	Prod 4 (mm)	Prod 5 (mm)	Prod 6 (mm)	Prod 7 (mm)	Prod 8 (mm)	Prod 9 (mm)
3 mm	450x337	605x337	905x537	450x437	605x437	905x437	450x537	605x537	905x337
32 mm	450x337	605x337	905x537	450x437	605x437	905x437	450x537	605x537	905x337

3.2.3 Product alignment in stations

Products are brought into the line on pallets by the system operator. Each product type is stacked and aligned before it is brought into the line to make the depalletizing process as stable as possible. Since there are a few key factors during the picking process in order to provide optimal conditions for the gripper, product alignment is vital and mandatory. The picking process shall be elaborated on following sections.

There is a single pallet type for both stations which is Euro Pallet 1000 mm x 1200 mm. Position alignment of products on pallets is made according to the mentioned pallet in Figure 3.

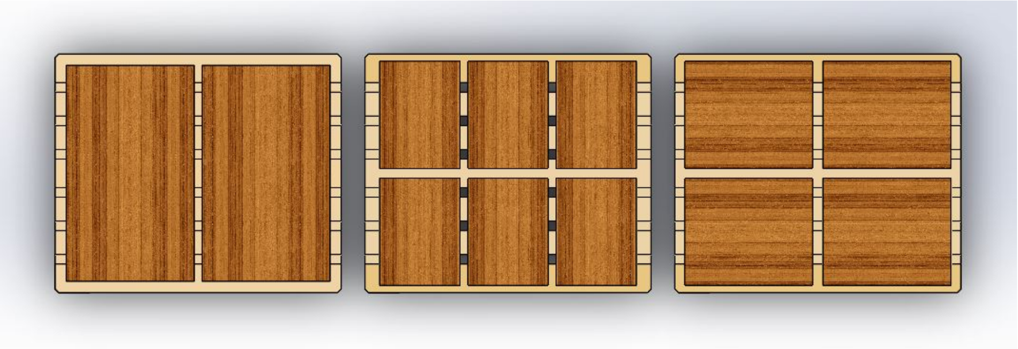

Figure 3. Product alignments on pallets

There are three main alignment methods among the product types, six products per layer, four products per layer and two products per layer. Alignment is made according to the product dimensions and gripper requirements. A full list of product number of products in one layer is given in Table 2:

Table 2. Number of products in one layer

	Prod 1	Prod 2	Prod 3	Prod 4	Prod 5	Prod 6	Prod 7	Prod 8	Prod 9
Number	6 pcs	4 pcs	2 pcs	4 pcs	4 pcs	2 pcs	4 pcs	4 pcs	2 pcs

There are three main rules which are followed for the alignment. First rule is arraying the layers by getting the edges as references. The layers must be arrayed on the pallet to maintain a stable picking process. The arraying is also important for the layer separation mechanism. The mechanism was designed by calculating the distance between panels and considering the centre axis as an empty area. The alignment does not have to have a high precision since the suction cups are not close to the edge of the products.

The distance between the panels is approximately 30 mm. The distance arrangement is made by using thick (32 mm) panels. The centre axis distance is important for the layer separation mechanism. If the distance is too high or too low, it can affect the separation efficiency. Too high distances between panels may hinder the separation since the air which is blown between panels may lose its power because it spreads in the ambient. On the other hand, too low distances may result in a collision. The mechanism may hit the products during its linear action.

3.3 Robot selection

Two robots are needed in the station because of the different processes. One robot is used for depalletizing process and the second robot is used for palletizing process. Robots were selected from ABB's four axis palletizing family. There is one chief reason for this selection. The reason is the axis orientation of the processes. There is no process which requires movement on vertical axis. So, horizontal orientation of the final axis is sufficient for the process. Therefore, palletizing robots are cheaper comparing to six axis robots since they require less equipment.

Robot types are different from each other. The depalletizing robot is selected as IRB660 to obtain a sufficient working envelope for the process. Existence of a gluing machine in the depalletizing station requires more free space to maintain a secure pick and place application. On the other hand, the palletizing robot is selected as IRB460 which is the smallest robot of the palletizing family since there is no obstacle or reach issues for the palletizing process. Both IRB460 and IRB660 robots have sufficient lifting capacity for both palletizing and depalletizing processes thanks to the low weight of the wood boards.

Table 3. Technical data of the robots

	Handling capacity (kg)	Reach (m)	Axes	Protection
IRB460	110	2.4	4	IP67
IRB660	180	3.15	4	IP67

3.4 Robot pedestals

A pedestal is an apparatus that helps to mount and elevate the manipulator to/from the ground with required height level. It consists of a base plate, an intermediary body, and a ground mounting plate. A pedestal must be durable and rigid considering the manipulator weight. It helps keeping stability and ensuring manoeuvrability of the manipulator in high speeds. Both robots' levels are higher than the ground level in order to contribute providing sufficient reach for the process and to prevent possible collision risks with the other machines. IRB660 pedestal height is less than IRB460 pedestal height because IRB660 already has the advantage of a higher working envelope. Increasing the base height helps the robots to reach the furthest points of the station which is necessary for picking and placing targets and maintaining a secure process. There is no height increment made for the pallet positions since the robots' working envelopes also cover the levels under the robot base.

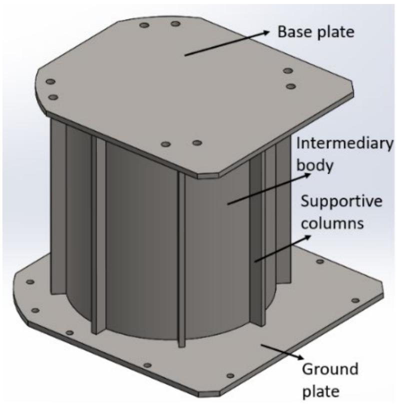

Figure 4. Robot pedestal part description

The base plate is the part that manipulator is connected to the pedestal. Generally, it has the mirrored design of the manipulator base in order to provide a firm mounting and suitable connection surface. The intermediary body is the part that elevates the manipulator from the ground. Its height depends on the process requirements. It consists of a circular or a rectangular body column and several supportive columns. The body column carries the main weight, while the additional columns supports the body column by covering the body columns all around. The body column consists of a circular entity and its inner volume is empty. The main purpose of the intermediary section is to reduce the pedestal weight and cost by using less material for elevation while providing a rigid standing. The ground mounting plate is the last in the pedestal. It provides connection surface for the ground mounting and helps adjusting parallelism of the manipulator to the ground.

Ground plate, base plate, intermediary body, and support columns are welded to each other. Structural steel is used as material. The design of the pedestal was sent to a provider and cutting, drilling and welding was done by the provider.

There are two pedestals in the layout. The design of the pedestals is identical since IRB460 and IRB660 have the same type of ground connection. The base plate thickness is 40 mm and, the ground plate thickness is 50 mm. The support column thickness is 50 mm and, its width is 100 mm. The only difference between the pedestals is the height difference. In order to provide sufficient reach to the conveyor for IRB460, the height of the pedestal was set as 800 mm. Since IRB660 has longer reach compared to IRB460, the pedestal height of the manipulator was set as 300 mm. With the given height levels, both robots are able to perform the picking and placing tasks without any collision.

The manipulators were attached to the base plate from eight connection points. M24*140 8.8 screws with 4 mm flat washers and M24 nuts were used to perform the connection. The torque value is given in the product manual as 725 Nm. The screws were tightened with the help of a torque meter.

The pedestals were attached to the ground from twelve connection points. M22*140 8,8 screws with 4 mm flat washers were used to perform the connection. The same torque value applied for the screw connections.

3.5 Gripper functionality

The wood industry mainly uses pneumatic and mechanical systems for picking applications. The underlying reasons of the preference can be exemplified as, the cheap and a large scale of ready solution availability, cost-effective consumption rates and possibility of solution implementation for multi-product groups. Pneumatic components are highly employed to fulfil the gripper functions in both stations considering the easiness and applicability of these systems.

Initially, identical grippers were intended to use for both robots resulting from the application similarity but, it had been noticed that the depalletizing station requires more detailed control over the products since the depalletizing robot has to deal the products with different thickness values while the palletizing robot deals with a single thickness value. Thin product groups require additional applications such as air regulation, layer separation etc. As a result of this situation, same base structure was used for both robots but the depalletizing gripper was modified in order to fulfil the additional requirements.

There are two main factors considered during the gripper designs; the product diversity and the process requirements. The product diversity is the first key element, because the gripper has to maintain the same picking performance for each product type. It means that the gripper should be able to handle the products with different sizes and thickness values. The second key element is carrying out several process requirements. Although the sole purpose of the project may seem like a simple picking process, it brings along several common and different subtasks with it. Besides the vacuum generation, the sub-applications are as follows;

- individual and overall control of picking groups,
- product absence-existence control,
- multi-layer control,
- layer separation,
- mechanical barriers,
- air regulation,
- height compensation,
- pallet level measurement,
- air dissipation control.
- product thickness detection

There are eight separately controlled suction groups on both grippers. Each group is connected to a vacuum generator. They were designed considering the different variations of picking processes. Group 1, 4, 5 and 8 consist of six suction cups and, group 2, 3, 6 and 7 consist of three suction cups.

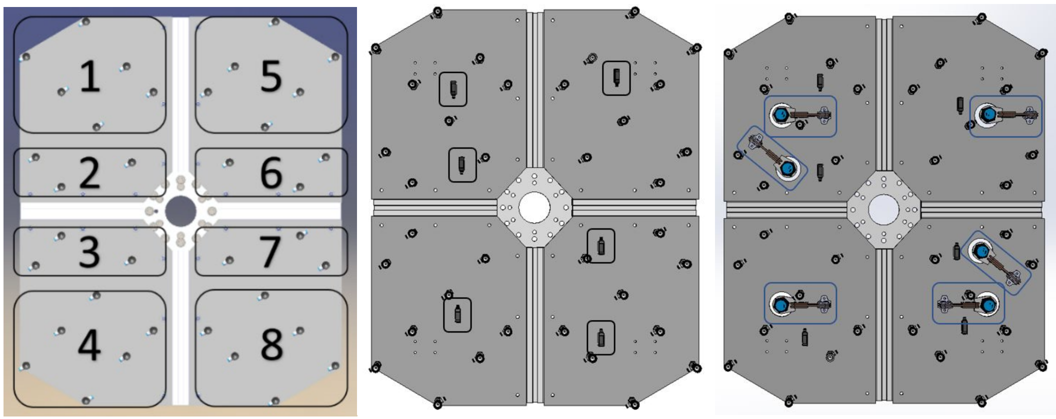

Figure 5. Bottom view of the gripper (Group division, OS and CS illustrated)

There are different picking combinations according to the product dimensions. All groups are used in one picking cycle regardless the product type. While the group 1, 4, 5 and 8 are able to grip not all but some types by themselves, group 2, 3, 6 and 7 are considered as support groups and they are not able to pick any product type by themselves. The picking combinations are listed in Table 4. First column consists of the product types and first row consists of the picking groups. Second column indicates different product numbers of specified types in first column. As it is shown in the Table 4, it is possible to handle with different product types with three combinations in total.

Table 4. Product picking combinations by groups

	Number in a layer	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8
Prod 1	6 pcs	1st	2nd	2nd	3rd	4th	5th	5th	6th
Prod 2,4,5,7,8	4 pcs	1st	1st	2nd	2nd	3rd	3rd	4th	4th
Prod 3,6,9	2 pcs	1st	1st	1st	1st	2nd	2nd	2nd	2nd

The picking process requires a full surface contact between the suction cups and the products to avoid a possible air leakage. Curvature of the product surface (especially for 3 mm boards), arbitrarily arranged stack heights, loosened mounting elements, springiness deformation of the rubber material of suction cups and any small deviation in level measurement can easily cause a contact loss between the products and the suction elements. Length compensators are used to eliminate the small deviations between these two surfaces.

The length compensators are located as a bridge between the gripper plate and the suction cups. After gripper reaches the product surface, it keeps getting close to the product surface with the help of the length compensators to ensure that there is no gap between the suction surface and the products. Furthermore, the length compensators help preventing possible damages to the boards and the vacuuming equipment. The sensors used in level measurement, pallet position displacement, external interference to the pallets by operators or any complication during picking process can make small changes in the pallet level. The robot can get close to the picking surface more than needed since it is not informed about the changes. In this case, length compensators can protect both the gripper equipment and the products from possible damages.

Some suction cups are not used in all picking processes because of the board dimensions. The vacuum elements are connected in parallel in each group. Idle suction elements may weaken the total suction force when the all vacuum generators are enabled since there is no physical element underneath them. The control of suction elements in product wise could be expensive since there are 36 elements in total. In order to minimize air dissipation and maximize the suction force, ISV (vacuum efficiency) valves are used between the length compensators and the suction cups. During vacuum generation, if a suction cup is uncovered or only partly covered, the ISV automatically stops the influx of air. 36 pieces of FESTO ESH-HDL-4-QS 20mm length compensators and ISV M6 valves are used in total (Figure 6).

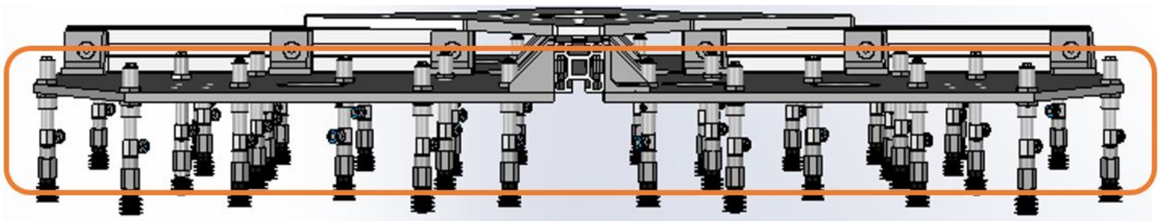

Figure 6. Level compensator and ISV demonstration

Product absence-existence control is required to increase control over the process. Unintended results may arise in case of a product drop during the transfer between work objects. Sudden air pressure drops, dustiness of the product surface, leakage due to a tear or air blockage on tubing or suction cups, increased weight by stuck products or ejector error may result in a drop in suction surface area. Any fallen product may cause a disorder in the process cycle. In order to prevent the disorder, gripper suction surface is controlled by six optical sensors (SOOE-BS-R-PNLK-T) in every cycle during product transfer only (Figure 5, second image). Sensors are activated after picking processes and deactivated after placing process. Otherwise it shall cause a continuous halt in the system because of lack of products.

There is a picking process difference between IRB460 and IRB600. While IRB660 is picking all the products at once, IRB460 is picking products one by one because of the conveyor logic. Sensor logic was made according to the picking process order in IRB460. Position of the sensors were selected considering the approximate product positions on gripper.

Table 5. Optical sensor reading order for product types

	Amount	OS 1	OS 2	OS 3	OS 4	OS 5	OS 6
Prod 1	6 pcs	1st	2nd	3rd	4th	5th	6th
Prod 2,4,5,7,8	4 pcs	1st	1st	2nd	3rd	4th	4h
Prod 3,6,9	2 pcs	1st	1st	1st	2nd	2nd	2nd

A necessary measure was taken in IRB660 station in order to prevent multi-layer picking possibility of the 3 mm boards. The porous material structure of the 3 mm boards may cause picking double, triple or even more layers at once. Since the all process was designed considering the input of one layer at each time, picking several layers has unintended consequences. Uneven product group heights on a pallet may cause a damage on suction cups or length compensators. On the other side, feeding the conveyor line with several layers possibly damages the gluing cylinder. Since the gluing cylinder height level is controlled by a manual lever, unexpected height input shall cause a damage on either the product or the cylinder. A two-step preventive measure was taken in order to avoid possible consequences. Capacitive proximity sensors are used to detect extra layers during a picking process. It can be understood by its name, a capacitive sensor reads and detects the changes in the capacitance. A capacitive sensor follows the basic theory of a capacitor working principle. It involves two conductive components. One of the components is the capacitive element, a plate, inside of the sensor, and the second component is the object to be sensed. The first element is joined to an oscillator which initiates the electric field. The air gap between these two elements is used as an insulator. If any object appears in front of the sensor, it causes a change in the capacitance value and the sensor registers the difference. SICK CM30-25NPP-EC1 model six pieces of capacitive sensors are used in order to perform the task (Figure 5, third image). Sensors are mounted close to the suction surface. Mounting element acts as a spring to mimic the compensation action of suction elements. Sensitivity adjustment was made after several tests with multi-layers. The sensor logic is the same with the optical sensor logic.

After detection of the additional undesired layers, several tests were performed to find out the best solution for separating the stuck layers. The most promising and effective method is selected as blowing air from the internal side of the products. In order to perform the method, a mechanism was designed. The tests performed shall be elaborated in chapter 6.

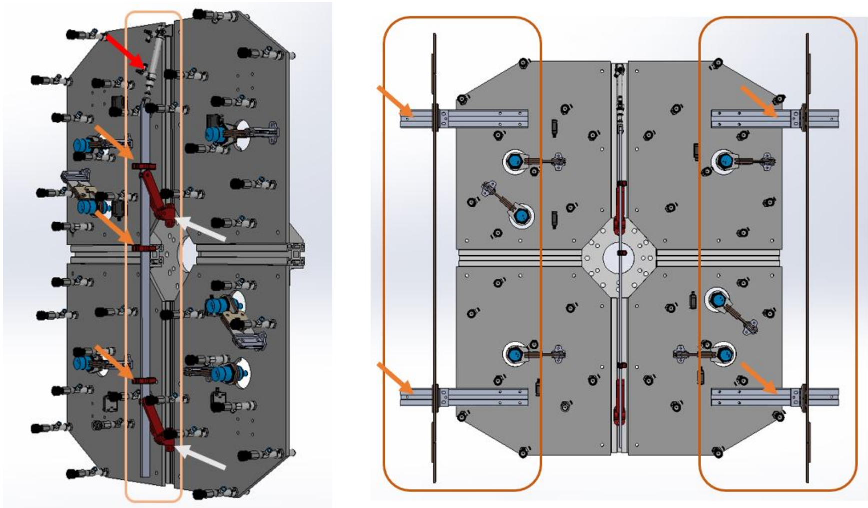

Figure 7. Multi-layer separation and barrier mechanisms

An 800 mm aluminium bar (Figure 7, first image) is connected to a double stroke pneumatic cylinder (red arrow) to make a simple crank-shaft motion. The cylinder initiates a linear motion against the gripper surface for the aluminium bar with the help of the mounting elements (white arrows). There are three blowing elements (orange arrows) attached to the aluminium bar. Blowing elements were designed to blow air in parallel to the gripper surface. While the bar is going down against the gripper surface, blowing elements blow air in between stuck layers. Air pressure is controlled by a check valve to avoid dragging the separated parts far away. A 3/2 solenoid valve is used to control air flow for the blowing elements. A 5/2 solenoid valve is used to control cylinder actions.

Layer separation process could cause the panels fly away from their initial positions. Blowing force could be higher than expected in some cases. One or more panels may lose their initial positions because of dragging by the air blown by the separation mechanism. Displaced panels may cause difficulties in the following picking process. Blocking other panels during picking or not having enough suction surface because of displacement can be given as examples. In order to prevent the displacement, two mechanical barriers were designed (Figure 7, second image). The barriers are attached on rails which enable relocating the barriers according to product dimensions. Small deviations in product position are neglected since the gripping surface was designed to mitigate these errors.

Pallet level measurement is required due to preventing manual data entrance. Small variations in product thickness may cause difficulties in estimating the stack height. Even though it is possible to estimate thick boards’ stack height, estimating the thin boards' stack height could take very long time because of difficulties in counting hundreds of wood boards. Besides, variations in product thickness make the height measuring process compulsory. After some certain amount of picking, the stack height has to be remeasured in order to avoid level errors caused by deviations. Level measurement is done by an optical sensor. The process is only necessary for the depalletizing station since the palletizing station's pallets are initially empty. One of the optical sensors, which is used for product detection on gripper, is used in order to avoid extra cost. Pallet level is controlled by the sensor in every 12 picking to correct the level data in controller.

The capacitive sensors move upwards and downwards during the picking process to position itself according to plate positions. Capacitive sensor precision adjustment was made according to full contact of the plate and the sensor surfaces. So, the contact has to be maintained until the placing process. The suction surface level from the main plates of the gripper can be roughly estimated but it is hard to keep all the products on the same level through each process because of several factors such as changes in the air pressure, product shape etc. To mitigate the small deviations in the product and the sensor levels, mounting element of the capacitive sensors were designed as a vertical hinge which can move freely when there is a contact with the panels. Initially, the sensors are approximately at the same level with the suction cups' surfaces. After the pneumatic actuation (picking), the suction cup bellows raises a little bit due to air compression. Thanks to capability of the mounting element's movement, sensors raise exactly the same amount with the panels. The sensor precision adjustment is made using the screwdriver on top of it. Rotation of the screw head changes sensitivity by changing the potentiometer distance. It has 4 mm to 25 mm sensing range. Its range is set approximately between 3mm to 4mm for this particular application.

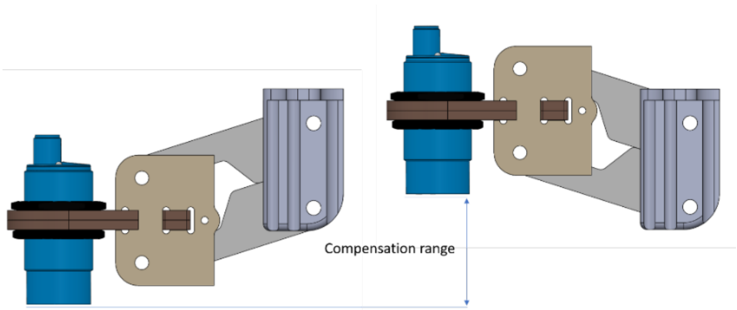

Figure 8. Compensation range of the mounting element

Initially, product thickness detection was one of the tasks required by the customer. It was planned to use the optical sensors on the gripper to detect the product thickness.

When a new pallet is placed to the depalletizing station, robot performs the height search. After the height search, products are picked by the gripper and placed to the conveyor. Product thickness detection was taking place after this step. The gripper was going to make another height search from the surface of the product and define the product thickness according to the height difference between gripper initial position height and the conveyor surface height. After the problem with multi-layer picking occurred, it was understood that different thickness values require different air pressure treatment. Since the air pressure regulation is done according to the product thickness, there was no way to acquire product thickness without it is given to the controller manually. The situation was discussed with the customer and it was decided that the product thickness value shall be provided by an operator at the beginning of a process. Thus, the product thickness search was removed from the gripper tasks.

3.6 Decision of pallet slot locations

Pallet positioning decision was made according to the robot working envelope and product dimensions. Safe transfer of the products from pallet to conveyor or from conveyor to pallet, safety distance from fences, robot movements for an efficient and fast transfer, easiness of replacement of a full or empty pallet, pallet position accuracy, safety factors between conveyor and pallet positions were considered during the position adjustment of the pallets.

A safe transfer of a product has a key role in the process. The path that the robot follows must be as clear as possible to simplify the trajectory and accelerate the process. Complicated actions done by robot may slow down the robot movements to prevent wear on mechanical equipment. A clear path provides a smooth movement trajectory for the robot. The distance between the pallet and other equipment such as the conveyor, the fences and the robot were decided considering the safest and the fastest options.

On the other hand, the pallet location in the station is another important factor that must be considered since the pallet placement is done by the operator. Operator must have a controlled, safe and clear way until the operator reaches to the pallet area to avoid an extra halt time.

Position accuracy of the pallets must be provided for an accurate and precise placing or picking by the robots. Considering the highly stacked pallets, taking the pallets out from stations may be extremely critical and hazardous for the operator, if the pallet positioning is poorly decided.

To maintain a continuous and accurate positioning, L shaped steel bar and sensors were used in the station. L shaped steel bar helps the operators to avoid any misplacement while the sensors provide an accurate placement by measuring the distance of the pallet edges. Two optical sensors (SOOE-BS-R-PNLK-T) were attached on each bar to stabilize the pallet positioning.

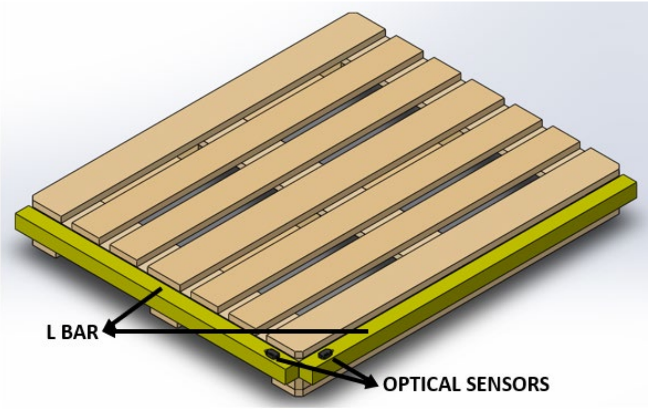

Figure 9. L bar and sensor positions

3.7 System safety

Safety is the most important case in robotic applications. In order to prevent damage to the human being and avoid foreseeable and unforeseeable incidents, all the necessary preventive measures are taken during the layout design stage. There are several safety factors considered during design of the layout. These factors can be summarized as:

- Surrounding of the cells
- Interaction from outside of the cells
- The risk of activating the line while someone is in the station
- Robot action restriction

Surrounding safety is maintained by the fences. Any robot action and required space limits during operation is estimated by RobotStudio software and the distance calculations are made according to the safety regulations. Robot system and safety equipment response times are considered in these calculations.

Interaction from outside is controlled by area scanners (Data Logic Laser Sentinel) which are located at the beginning of the line. The area scanners scan the area between the line entrance and the robot. Any interaction caused by an unidentified external source is restricted and causes a halt in robot system. In addition to this, area scanners scan the pallet areas. They interact with the pallet sensors and in case of absence of a pallet in palletizing/depalletizing areas, the area scanners change their conditions and start scanning pallet area as well.

Since the area scanner is not able to scan behind the objects, area between the robot and gluing machine remains as a dead spot. This area is controlled by a light curtain in both stations. Any violation in this area causes a system halt. Unless the light curtain is activated by the light button located at outside of the station, system stays in halt status.

During programming the trajectory or any modification on robot targets may cause unintended consequences as a result of a mistake done by engineers/operators. In order to prevent the damage and the unexpected incidents, robot action areas are predefined in case of an unexpected robot movement. Using the SafeMove function of IRB systems, all the unnecessary areas are blinded and robot actions to these areas are restricted. Robots shall stop their movement if the defined areas are violated.

More detailed information about safety equipment and system logic shall be provided in safety section.

4. PROCESS FLOW AND PROGRAMMING LOGIC

4.1 Process Flow

Process flow mainly describes the steps that the robot follows during a cycle. One cycle starts with delivering a pallet to the station and it is completed when the robot unloads the gripper. In other words, when the robot cursor moves along all the necessary steps and goes back to its initial position without encountering any error, one cycle is completed. Programming logic directs the robots between tasks and modules to complete a cycle and helps both user and robots to resolve the errors.

In general, both IRB460 and IRB660 robots in the system follow almost the same pattern during a cycle time. Both controllers have three tasks and eight modules. While the tasks are classified according to main system components, the modules are classified according to main steps that the robots follow. There are several minor changes because of the reversed process logic and additional tasks in the controllers which shall be elaborated in this chapter.

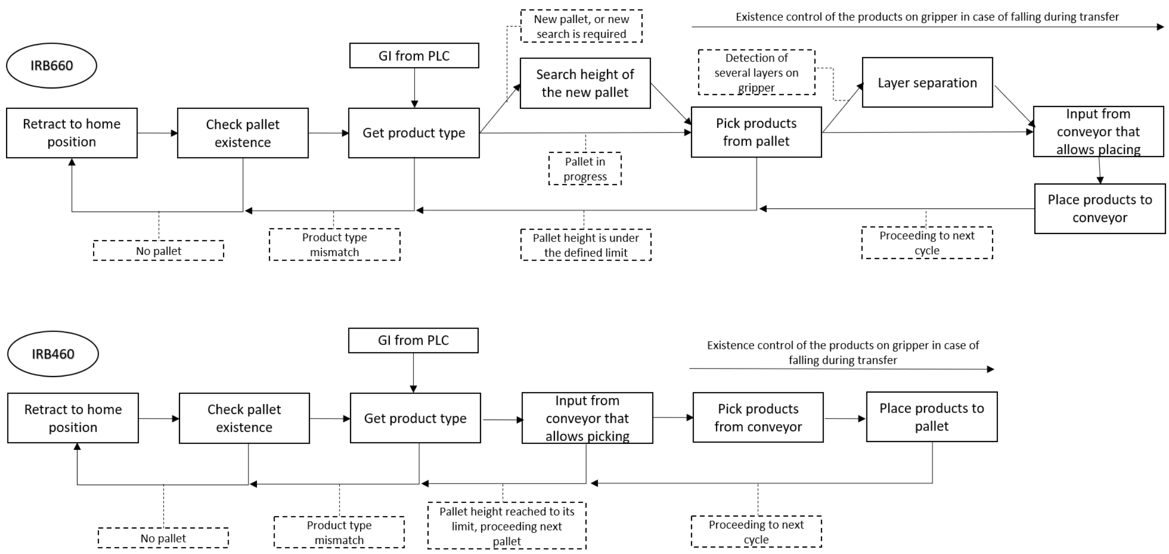

Figure 10. Process flow of the controllers

The process flow mostly controlled by the controller signals, but there are also a few inputs come from the system PLC. These inputs help the system to work without any conflict during a cycle. If the station with gluing machine, heating system, conveyors and robots are considered as a whole, robot controllers are there as a slave while rest of the system controlled by the PLC is there as a master. The tasks in the controllers shall be described according to the process flow in the following sections.

4.2 Definitions of the RAPID elements

A RAPID application is named as task. A task consists of a set of modules. A set of type definitions (like variables or constants), data and routine declarations form a module. The task buffer is used to host modules currently in use (execution, development) on a system.

There are four types of data objects: constant, variables, persistent and parameter. A constant represents a static value and is introduced by a constant declaration. The value of a constant cannot be modified. A persistent (data object) can be described as a "persistent" variable. It keeps its value between sessions. A variable value is lost (re-initialized) at the beginning of each new session. That is, when a module is loaded (module variable) or a routine is called (routine variable).

A routine is called a carrier of executable code. A user routine is defined by a RAPID routine declaration. There are three sorts of routines: functions, procedures and traps. A function returns a value of a specific type and is used in expression context. A procedure does not return any value and is used in statement context. Trap routines provide a means to respond to interrupts and is then later automatically executed if that particular interrupt occurs. A trap routine can never be explicitly called from RAPID code. Routine declarations cannot be nested. It is not possible to declare a routine inside a routine declaration.

The if and test statements are used for selection. The if statement allows the selection of a statement list based on the value of a condition. The test statement selects one (or none) of a set of statement lists, depending on the value of an expression. The for and while statements are used for iteration. The for statement repeats the evaluation of a statement list as long as the value of a loop variable is within a specified value range. The loop variable is updated (with selectable increment) at the end of each iteration. The while statement repeats the evaluation of a statement list as long as a condition is met. The condition is evaluated and checked at the beginning of each iteration.

Execution errors are detected during the execution of a task. Arithmetic errors, for example division by zero, I/O errors, for example no such file or device, fatal (system resource) errors, for example execution stack overflow. The error handler concept of RAPID makes it possible to recover from non-fatal execution errors.

A record data type is a composite type with named, ordered components. The value of a record type is a composite value consisting of the values of its components. A component can have atomic type or record type.

A cross connection is a logical connection between I/O signals of type digital (DO, DI) or group (GO, GI), that allow one or several I/O signals to automatically affect the state of other I/O signals [22].

4.3 Robot tasks

Both IRB460 and IRB660 controllers contain three tasks in total. These tasks were named as 'Line communication life bit', 'ROB1', and 'Signals', respectively. While the ROB1 task is a motion task, other two tasks were created to set and control signals both between the PLC and the robot, and the signals in the robot station. These two semi static tasks keep running even the ROB1 task stops. The semi static tasks normally are not stopped by emergency stops or stop button on the flex pendant.

4.3.1 TROB1 task

ROB1 task forms the bulk of the code. It consists of six modules namely calibration data module, main module, picking module, placing module, targets module, and error module. The task was separated in modules in order to make the task easy to follow and more understandable. Each module in the task contains a different process or data as it can be understood by their names.

Calibration data module contains information about work objects, speed of the robot, and the tool. It helps to define all the necessary data beforehand in the motion task and then be used in move instructions. Work object data helps to describe the work object that the robots interacts with. When a work object is defined at a certain coordinate, the motion instructions that the robots follow based on that work object can be easily shifted or readjusted by only modifying the work object coordinates. Three work objects were defined in the calibration data to ease the position modifications of the robot around first and second pallet, and the conveyor.

A tool data contains information about the characteristic of a tool. In this project, it is the vacuum gripper. These characteristics are orientation and position of the tool center point and the physical characteristics of the tool load.

A speed data is used to define the velocity of the robot and the external axes move. Generally, it saves the time of the user by avoiding speed specification in each move instruction and increases awareness by enabling the naming. Different speed data are used to adjust the robot velocity under variable circumstances such as the movement without load or with load.

Error module is used to raise an error in case of any unintended action during a cycle. The predefined errors may occur during a cycle are listed as: A product fall during picking or transfer (common both in IRB460 and IRB660), pallet misposition (common both in IRB460 and IRB660), multiple layer picking (only in IRB660), multiple layer carrying (only in IRB660).

Three procedures were created for each error type. Different error IDs were assigned to each type. Error IDs were used in user manual to describe the error. Error strings were used to compile error messages. Each error string was named and numbered according to error type. The error strings are called with the help of an error log. An error log is used to display error message on the flex pendant screen, which is formed by an error ID, a title, and up to four error strings. Logic of the error triggering and pallet misposition error shall be explained in the signal task.

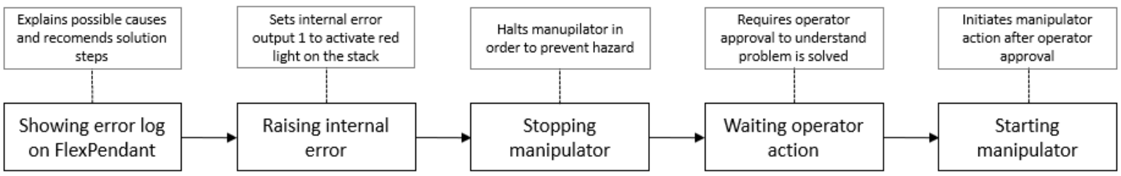

Figure 11. Error process flow

A user defined error procedure consists of several elements. Error log command takes place the first line of the procedure. It displays the error message on the flex pendant screen. A user defined internal error output is set to 1 to activate the red light on signal tower. A persistent data is raised to 1 to activate the buzzer. Robot motion is stopped by using the 'StopMove' command in the next line. At that moment, the robot shall wait until it is restarted by an operator again. After the operator takes control of the situation, it clicks the 'Play' button on the flex pendant and the robot cursor moves to the next line. A 'StartMove' command activates the robot motion in the next line. As a final step, the internal error output is set to 0 to disactivate the red light in the light stack. The error procedure that controls the product fall has an additional function. If the fall happens during picking process, it raises the robot 30 cm on Z-axis to ease the replacement of the product by the operator. Furthermore, it analyses which section is missing and deactivates the related vacuum generator because, it is not always possible to regrip the product when the generator is active due to air leakage of the gripping surface. When the product is detected, it activates the generator after 1.5 seconds.

Figure 12. A custom defined error message example

Target module is the module that contains coordinate data of the positions which are used in the motion task. Each coordinate is stored in a robot target. A robot target is needed to define the position of the robot and additional axes. If the robot can go to the same position with a different axis configuration, the axis configuration is also specified. A robot target is generally used to identify the name of the target. The user can easily spot the needed robot target in a bunch of movement command in a procedure with the help of the robot target tags. This helps to save time and avoid crashes most of the time. In general, it is not necessary to create a target module if the number of the targets is not so many. The targets can be stored in the related task too. The target module was created in order to clean up the other modules. A robot target can be called using the same name tag by other modules in the same motion task. Related robot targets were grouped together under command lines to ease the detection.

Main module is the module that forms the structure of the ROB1 task, and it contains all the initiator procedures and general definitions as its name refers. It is composed of procedures and traps.

Each module which directly contributes to robot action contains a main procedure. When the 'PP to main' button is clicked on the flex-pendant, the program cursor moves to the main procedures in the modules. This step is necessary to start the robot movement after the controller is turned on.

The procedure named 'main' in the main module calls each procedure which is necessary to complete a cycle. It is the procedure that the program cursor roams during a cycle. Each step that the robot is aimed to reach must be written inside of the main procedure. The main procedure starts with a 'StartMove' command to initiate the robot movement in case of existence of a prior 'StopMove' command from the previous cycle. It can be thought as a reset function to initialize the previous commands. Using the same logic, internal error digital output is set to 0. (Internal error digital output is set to 1 if any error occurs during a cycle. It is used to activate the red light on signal tower.)

It contains interrupt conditions which are necessary for the control of motion pause, control of the gripped boards, and control of the possible multi-layer board groups. The input signals which come from the area scanner, the OS, and the CS trigger the interrupts. If an interrupt condition comes true, it calls the related trap routine to inform the operator to take necessary measurements. Multi-layer control trap only exists in IRB660 controller.

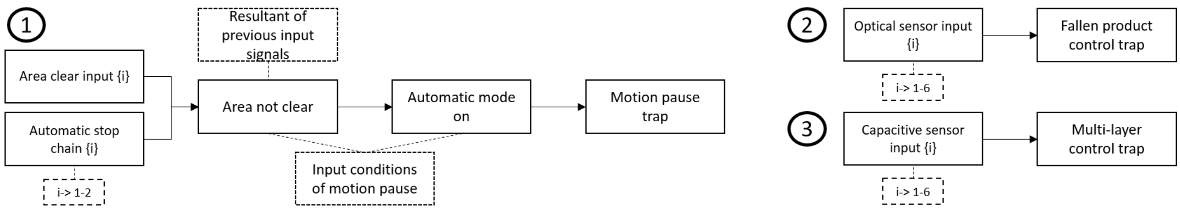

Figure 13. Trap routine triggering conditions

Thereafter, a procedure namely 'pInitialize' is called to initialize robot acceleration and to retract the robot to its home position safely. After all the initialization processes are done, the program cursor enters a while loop. The condition of the while loop is 'TRUE', which leads a continuous cycle until the controller is stopped by the operator or an error. The main processes (such as picking, placing etc.) which are necessary to complete a full cycle are embedded inside of the while loop.

The while loop starts with an if condition in IRB660 that controls the pallet availability to proceed. The pallet slot which is stated as in progress has the priority. Otherwise, the controller choses the first available slot. The if condition also checks the vacuum generators in case there is any active component or gripped product from the previous cycle. After the pallet decision is made, the controller gets the product type from the PLC with the help of a group input.

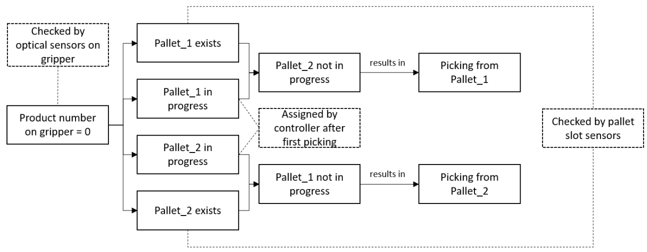

Figure 14. Control of pallet availability condition

The procedure named ‘pGetProductType’ starts with naming the process by overwriting a string for the HMI screen. Right after, it stores the group input value (which varies from 1 to 9) into a numeric data to compare the value with its defined test-case scenarios. Before getting into the test instruction, it compares the current product type and its thickness, and the previous product type and its thickness to decide either getting in or passing the test instruction (The thickness values are only controlled in IRB660 controller since there is only one thickness value in IRB460 process). If there is no change in the data stated above, there is no need to read and overwrite the product information in each cycle. Otherwise, the program cursor goes into the if condition to initialize several important parameters. There are nine cases in the test instruction to compare the stored group input data. Each case represents different product type. After the stored group input data and the case number is matched, all the necessary information about the product type (including product dimensions, product thickness (only in IRB660), how many products are available in one layer, gripper configuration, and pallet picking position) is overwritten in related predefined data. The product dimensions are stored in a string to depict it on HMI screen. Initially, the product thickness is defined as 3 mm, but according to the information selected by operator on HMI screen, it is overwritten (only in IRB660). Available product number in one layer is needed to manage the gripper vacuum sections. According to given product number, the gripper configuration is adjusted. The gripper configuration defines the activation sequence of the vacuum generators. After the test instruction defines the related product data, the previous product type and the product thickness (only in IRB660) are defined as current data to avoid getting into the if condition that initializes the product data. The stopper slider boolean data is defined as false as a reminder for the operator to adjust the slider position (only in IRB660). If operator forgets adjusting the slider position, sliders may hit the products and get damaged. Another control for the selected pallet status is made to confirm that the pallet is not in progress since the initializing is made for a new pallet and a new product type. Otherwise the related pallet status is marked as done. A final initializing is done for the picked panel number according to the information comes from the group output value of the vacuum generators. If the group output is zero, it means the gripper is empty and ready for picking. Otherwise, it might be holding one or more products and these products must be released before the picking process. The final controls are made in order to avoid unintended consequences, but generally these are not expected cases. After the final controls, the program cursor goes out of the if condition that initializes the product, the gripper, and the pallet parameters. As a last step of the getting product type procedure, the program cursor goes into the procedure of robot speed adjustment for the thin and thick plates (only in IRB660). The boards with different thickness values cannot be transferred to the conveyor with the same speed. The thicker products have to be transferred with a lower speed to guarantee that there is no product fall during the transfer. The procedure adjusts the speed parameters according to the thickness data selected by the operator.

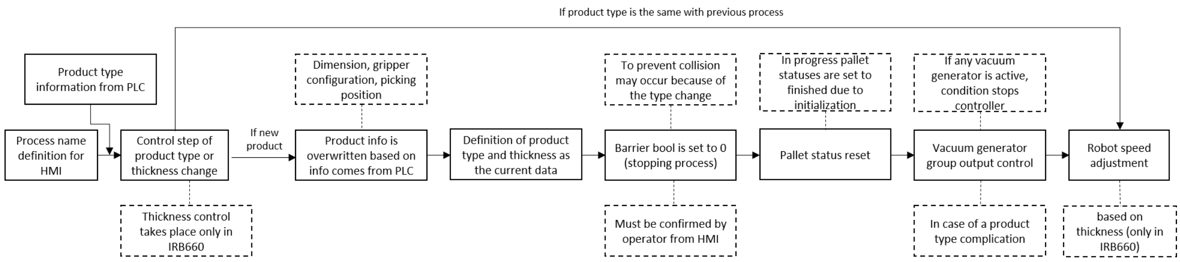

Figure 15. Process flow of the getting product type routine

After acquiring the product type, the program cursor moves into the main procedure in the picking module. Both controllers overwrite the process name to a string for the HMI screen. The process of getting product type is inserted into the if condition in IRB660 because there is a possibility of having two different product types on the pallets. That is why, the controller has to make the pallet decision at first, then it gets the product type. It is not necessary in IRB460 controller since there is only one product type comes out from the conveyor.

After getting the product type the program cursor goes into the picking procedure. Picking procedures are different in IRB460 and IRB660. IRB660 goes to the pallet and picks the products regardless the type. It activates all the vacuum generators at once and picks the whole layer. In IRB460, the program cursor goes into a test instruction to get the information about how many picks should be done. The roller conveyor in IRB460 station feeds two products at once. These products move till the end of the conveyor. When they reach the end, conveyor belts which are placed 90 degrees rotated between the rollers rise up and drives the products to the corners of the conveyor. This is a referencing process for the robot to provide the same picking position all the time.

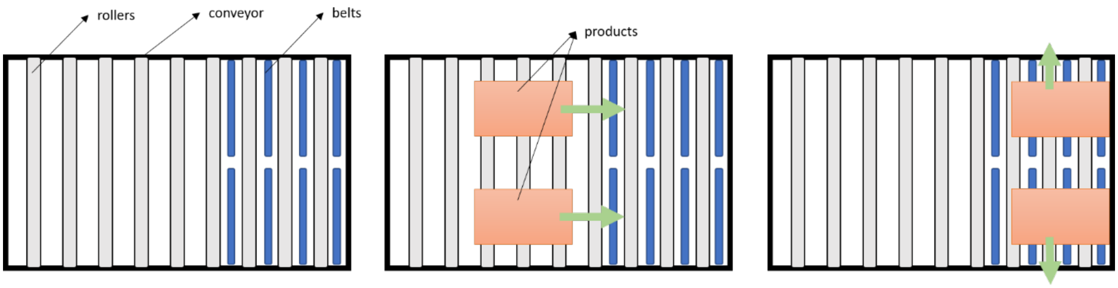

Figure 16. Product positioning on conveyor

According to the product dimensions, robot makes the picking with an offset. This offset information comes from the cases in test instruction. Another important information for the robot is acquired in 'pGetProductType' procedure. When the product number in a layer is acquired, the controller knows how much picking steps shall be. Furthermore, getting the gripper configuration from the same procedure provides information about which groups shall act together or separately. When the referencing is done, the PLC sends a signal to the controller that the picking can be done. The controller also checks the pallet statuses to ensure that the placing can be done. If the pallets do not exist or pallets are full, the controller waits until a new pallet is inserted to the slots. After the two products on conveyor are picked, the controller sends an output to PLC that the conveyor area is clean. Then the system provides the next two products. After all the products are picked, IRB460 goes an intermediary position between the conveyor and the pallets.

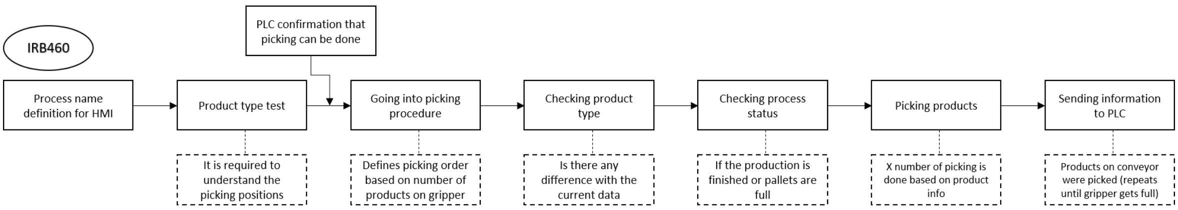

Figure 17. IRB460 picking process

In IRB660, the robot picks the products from the pallets since it is a depalletizing process. Before the picking process, robot checks the pallet statuses. If both pallets are in progress, robot stops the program flow. There is only one product type can be fed during a production. If the pallet statuses are okay, the controller checks the pallet statuses one more time to understand which pallet it should go. If there is one pallet in progress, it has the priority. If there is only one pallet present, robot goes directly to that pallet. If both pallets are present and ready to be picked, the first pallet has the priority because of the logic sequence. The controller checks one more condition before starting the depalletizing. If the depalletizing is about to start, and the product thickness is 3 mm, it sets the stopper slider boolean to FALSE as a last warning before the picking process to prevent collision. After operator approval, it starts the picking process.

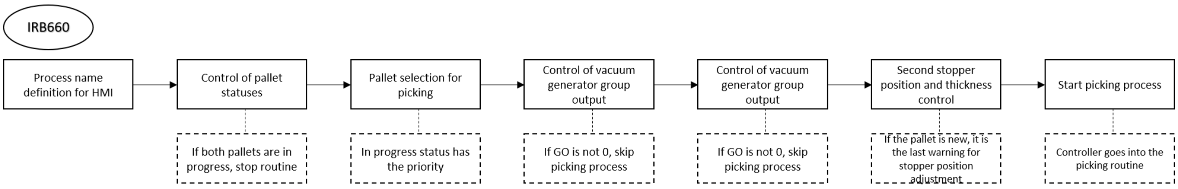

Figure 18. IRB660 getting ready for the picking process

First, robot goes to a position which is higher than the pallet height to avoid any possible collision. If the pallet is a new pallet, the controller initiates the height search procedure. The height search procedure is run under two conditions. The first case is the new pallet case. Since the pallets are loaded without counting number of layers, the controller has to know the pallet height to make a safe picking process. The second case occurs after 12 picking cycles. Because of the different tolerances of the products, the height value cannot be calculated with 100% precision. Due to unexpected situations like curvature of the product surface, small differences in product height, deviations occur in the pallet height. The second case is created to make safe and controlled picking processes.

Pallet height search procedure starts with writing the procedure name to the HMI screen. The procedure checks a condition to reduce the time that shall be spent during a height search. If the pallet is a new pallet, the height search starts at a higher position to avoid a potential collision. If the pallet is in progress, the height search starts at a level which is estimated according to the previous height search result. The reason of using different height levels is to mitigate the time spent during the search. Since the height search is a delicate process, the robot approaches to the pallet with a low speed level. If the pallet is partially depalletized, starting the search at a high level shall result in excessive time consumption. The search process is done with the help of one of the optical sensors on the gripper. The robot initializes the height search according to the condition above and starts getting close to the top layer. When the optical sensor signal changes from 0 to 1, robot stops the search and stores its current position. This position’s Z axis coordinate is used in a calculation to estimate the pallet height. The calculation subtracts sensor offset to the gripper surface from the stored position to calculate the real pallet height and resets the picked layer number. The picked layer number is used to count 12 layers (or cycles) mentioned above. The procedure has a search distance limit. If the limit is exceeded, an error is generated by the controller. In case of an error, robot records the pallet height as the minimum height limit and the controller checks the condition one more time. Since the minimum height limit is reached, the controller shall go to the retract position and changes the pallet status as finished.

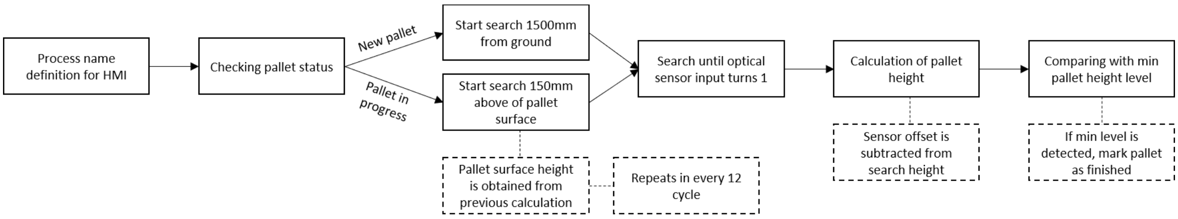

Figure 19. Pallet height search process

The pallet status was checked before the height search procedure. The controller does the height search if the pallet is new or the picked layer number reach to its maximum. If the controller can do the height search, pallet status is changed to pallet in progress. According to the obtained height, the robot approaches a position which is 40 mm higher than the pallet top layer. Right after, the robot goes 5 mm under the top layer surface to ensure the full contact between suction cups and boards. The controller activates all vacuum generators. If the picked products have 32 mm thickness, the robot rises with an offset value of 1700 mm from the pallet ground level and then approaches to the conveyor. The recorded pallet height is reduced as the picked product thickness. The controller checks the pallet height. If it reaches the minimum height level, pallet status is marked as finished. If the product thickness is 3 mm, the controller follows another set of instructions. These instructions are needed for the layer separation. If the product thickness is 3 mm, the air blowing mechanism is activated after robot goes a higher position than the pallet level. The mechanism is controlled by a procedure by setting and resetting the related cylinder and solenoid valve outputs. The pneumatic cylinder is used to make the sliding movement of the bar which contains the air nozzles on it. The cylinder action is controlled by a 5/2 solenoid valve. The air blown by the nozzles is controlled by a 3/2 solenoid valve. The controller checks the capacitive sensor group output to see if there is an unexpected thickness value in any sensor. If everything is normal, the controller goes to the approach position of the conveyor. If the group output does not give the expected value, the controller tries to separate the layers again. After the third trial, the robot raises 400 mm on Z axis and stops if the group output does not match with the expected value. The controller gives an error message to the operator that the separation is failed. The reason that the robot goes to 400 mm offset on Z axis is to ease the manual separation for the operator. After the operator makes the separation, robot has to be initiated from the flex pendant. The program flow is the same with the thicker product procedure except for the additional layer separation process.

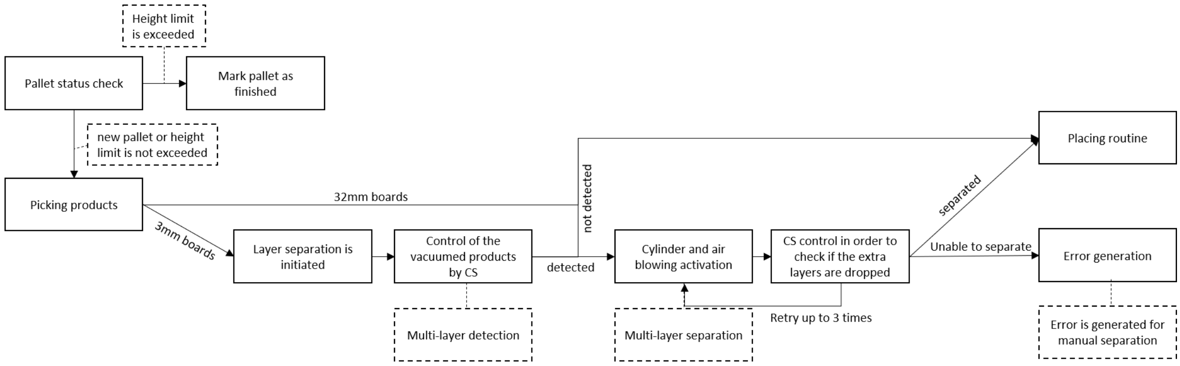

Figure 20. IRB660 picking & layer separation

Both controllers start placing the picked products by writing the process name to the HMI screen. In IRB460, the controller firstly decides where to place the products. If it is the first layer, the controller adjusts its placing speed according to it. The speed adjustment is necessary because of the layer pad on the pallet. If the robot approaches to the placing position with a high speed, products may slip after releasing the products. In order to avoid this situation, the robot approaches to the pallet with a lower speed then releases the products. The pallet height is calculated for the following processes. The calculation is done by multiplying the product thickness with the palletized layer numbers. If the pallet is a new pallet, its status is changed to pallet in progress after the placement. Also, palletized layer number is increased by one after each cycle. 40 layers is the highest limit of a pallet. If the layer number reaches 40, the pallet status is marked as finished. The robot goes to retract position before it goes to pick the next products. It waits the input from the PLC for the next picking process.

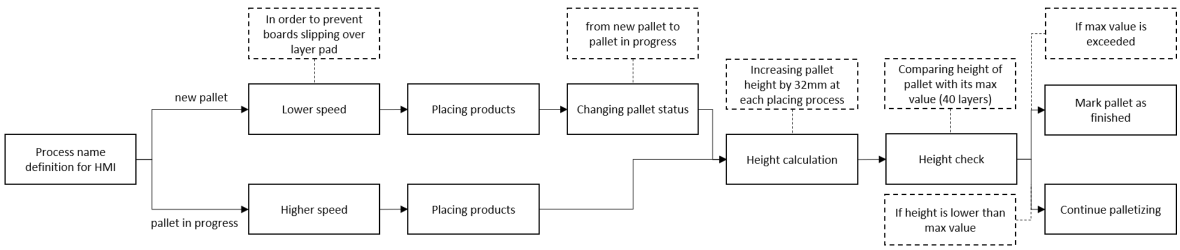

Figure 21. IRB460 layer placing

In IRB660, the logic is different because of the product dimensions. Products have to be placed on the conveyor according to their dimensions. The orientation of the products which shall be served to the conveyor is important. The controller decides the product placement orientation and height according to the product type and thickness. The thickness is an important factor in order to avoid releasing products from higher level than the conveyor. A test instruction is used to make the decision. Cases inside the test instruction include the logic for the product orientation and placement height. After the controller choses the necessary data, the product placement starts. The number of the products on the gripper was acknowledged in 'pGetProductType' procedure. The controller knows how many placements shall be done according to this information. How many products shall be placed on the conveyor at once was decided by the gluing machine manufacturer. It is related to the length of the gluing roller. The products on the gripper can be placed by one by one, groups of two, or all the products can be placed at once. Each type contains its own placing coordinates. The number of the placements are specified in the procedures. The controller waits for the placement until the approval comes from the conveyor. When free to load signal is raised, the controller sends a signal to conveyor that the placing area is occupied by the robot arm. This information is given to avoid any conveyor actuation during the placement. If any actuation occurs, the products might be driven by conveyor while the vacuum is still active, and the suction elements might get damaged. After the information is given to the conveyor, the robot gets closer to the conveyor for the placement of the products. When the robot gets closer to the conveyor, the capacitive sensors might be raised because of the conveyor rollers. The controller stops reading the capacitive sensor inputs to avoid interrupting the robot movement during the placement of the products. The controller logic is made to understand the vacuum generator sequence. After each product placement, the controller counts down the picked product number and this number is associated with the generator sequence. If a 3 mm board is being placed, the controller offsets the placement coordinate 15 mm lower on Z axis to avoid the case of dropping the products from a higher level. After the all placements are done, the robot goes an intermediary position for the next cycle. The intermediary position is needed to avoid collision with the conveyor and the gluing machine. If the robot makes the picks from the right pallet, it follows a different way to approach the pallet because of the gluing machine. Since the machine height is higher than the conveyor level, it goes over the gluing machine to avoid any collision.

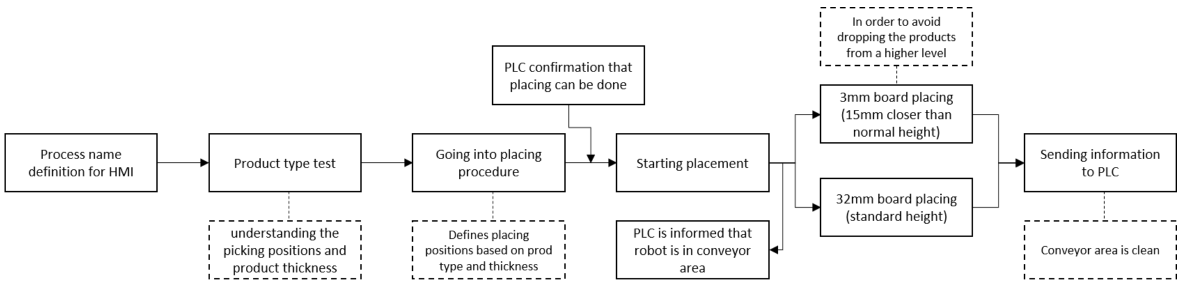

Figure 22. IRB660 board placing

The controller checks a condition to understand if the depalletizing process is desired to be finished before the pallet gets empty. There is a section in the HMI to adjust the number of the pickings before the process is terminated. If a shift is done or the number of orders is achieved, the operator can terminate the depalletizing before the pallet is emptied by the robot. According to the number entered by the operator, the controller makes the pickings, then it terminates the process.

There are three trap routines in the main module. The trap routines are used to respond interrupts. Two of these interrupts are common in both controllers. One of them is only used in IRB660 controller. The common interrupts occur when a product falls from the grippers or when an undefined object is detected in the work area. The first interrupt stops the controller action when a product falls from the grippers. It is connected with the OS inputs. If an input signal of an OS goes low while a vacuum generator output signal of the related section is high during the picking process or the transfer between the pallet and the conveyor, the trap which is assigned to respond the interrupt calls an error routine. The error routine executes a message to the flex pendant screen about the fallen product and the steps that the operator shall follow.

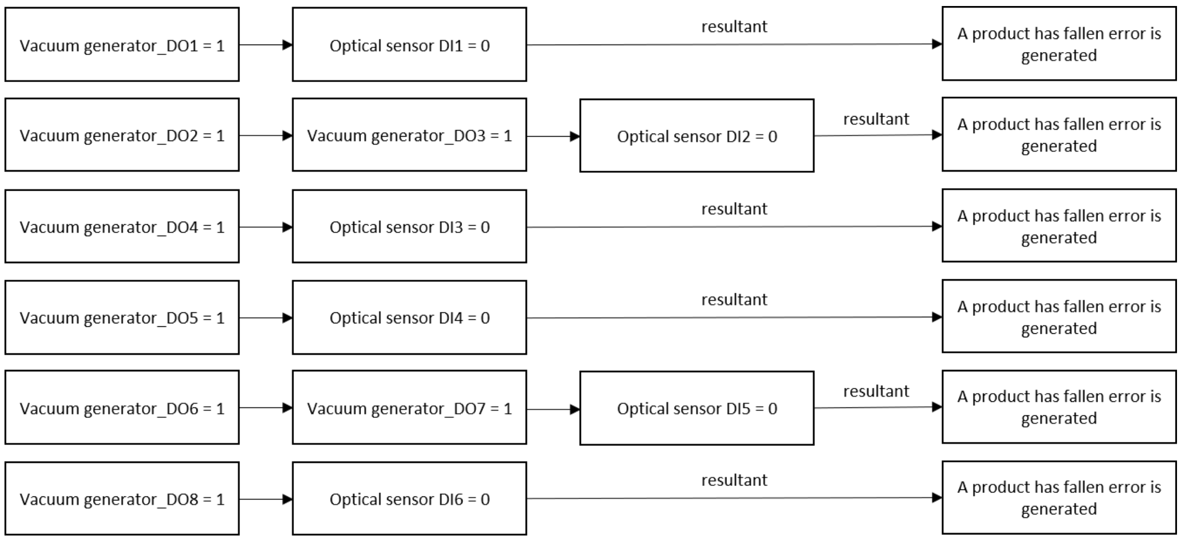

Figure 23. Fallen product trap conditions

The second common trap routine is in relation with a cross connection. A cross connection checks the given logic relations between the digital I/O signals or group I/O signals. The cross connection made for the second trap routine contains an AND logic between the input signal comes from the area scanner and the system output which shows the status of the controller. If the controller is in auto mode, the system output goes high. If both the scanner and the system outputs are high, the cross connection rises an output as a result of the AND logic (Figure13-1). This output orders an interrupt to stop the controller action. The robot does not start acting until the signal goes low. Whenever the signal goes low, the controller activates an alert using the buzzer mounted to the fences. It also blinks the green light of the light stack. It is meant to inform the operator that the robot is about to move. After 6 seconds, the controller automatically starts the robot action. The purpose of the trap routine is to reduce the activation time of the cell. When a pallet is placed or removed to/from the cell, the operator does not have to go to flex pendant and restart the process. During the test of the cells, it was noticed that, the wood dust may also activate the area scanner and it would cause the undetermined system stops. To avoid from the situation, a timer is placed inside of the trap routine. If the activation of the area scanner is lower than 0.8 seconds, the controller makes a 3 seconds halt but, it does not activate the light stack or the alert. The halt is maintained in case of an error of the area scanner.

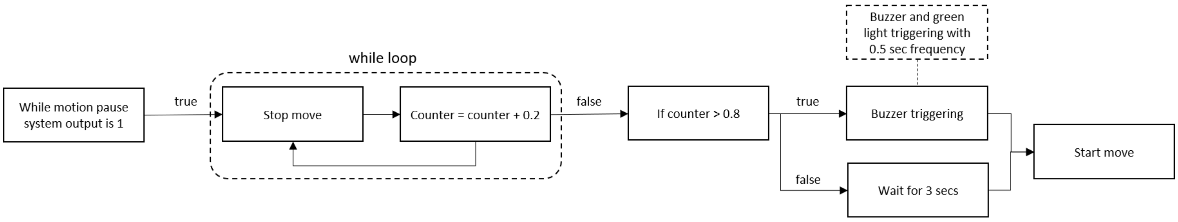

Figure 24. Motion pause trap flow chart

The final trap routine which is only used in IRB660 controller controls the existence of the multi-layers on the gripper. A multi-layer product group may not be detected during the picking process because of the springiness of the mounting elements. If the CS sensor in the related group detects a multi-layer later on when the related vacuum generator output signal is active, the trap calls an error routine to inform the operator. The error message depicts the steps that the operator shall follow. The trap is active only the thickness of the product is 3 mm.

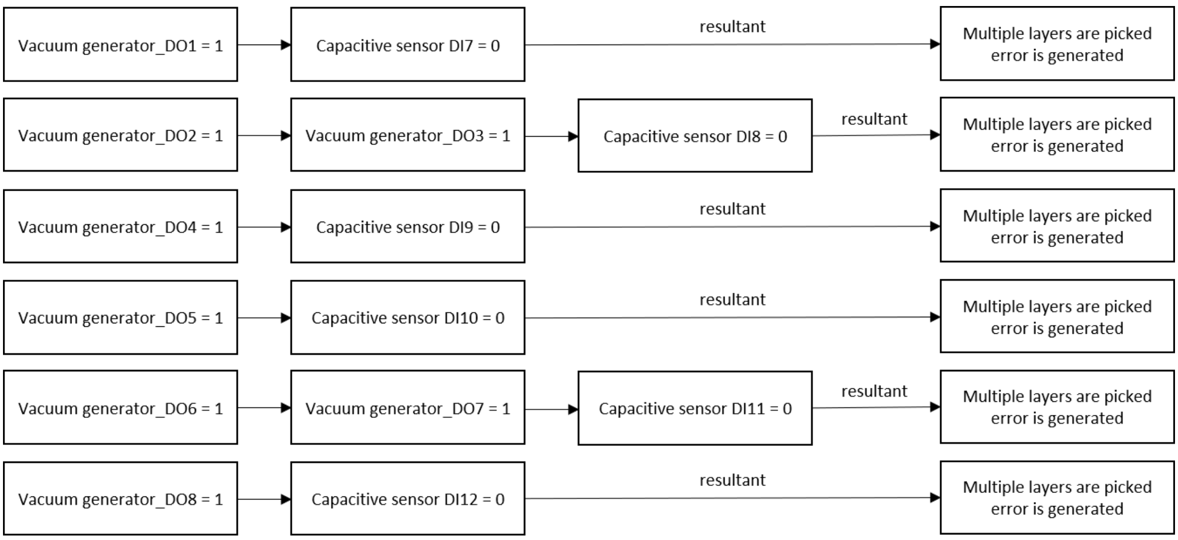

Figure 25. Multiple layer trap conditions

4.3.2 Semi-static task for control of continuous station signals

The signals task is a continuous task. Program cursor of the task keeps moving even the other tasks are stopped. If any modification is needed in the task, it has to be stopped manually. It contains various important procedures which have to be controlled continuously. The program cursor goes into each procedure sequentially. There is a main procedure at the beginning of the task and the other procedures are called from this procedure continuously with the help of a WHILE TRUE DO loop. WHILE TRUE DO statement means loop forever. It is initiated when the controller is started.

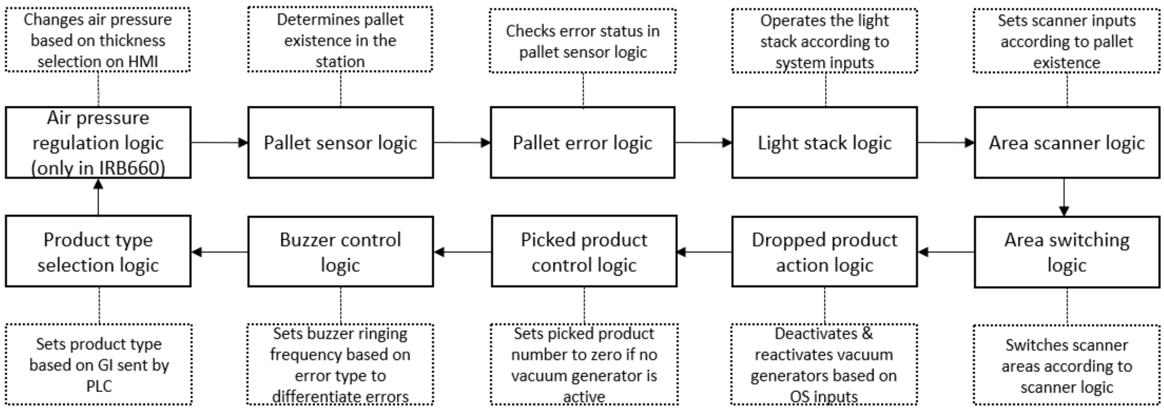

Figure 26. Process flow of signal task

After the thickness selection is made from the HMI, a persistent numeric data is overwritten. According to the selected thickness, a procedure called 'pSelectThickness' makes the decision of the pressure which is later used by the vacuum generators. The purpose of the procedure is regulating the air pressure to maintain a controlled picking process. Two 3/2 solenoid valves are used to separate the air pressure goes into the vacuum generators. The main air supply which is 7 bars separated into two lines with the help of a T fitting. One of the output channels of the T fitting is directly connected to the one of the mentioned solenoid valves. The second output channel of the T fitting is connected to a regulator which is set to 3.5 bars. Output of the regulator is connected to the second solenoid valve. If the 3 mm thickness is selected by the operator, the input signal of the second solenoid valve is set to 1 and the input signal of the first solenoid valve is set to 0. This configuration enables the air flow with 3.5 bars through vacuum generators. The gripper picks the thinner products with a lower pressure. If 32mm thickness is selected on HMI, the input signals are reversed. This configuration bypasses the air regulator and directly transmits 7 bars air flow to the vacuum generator. Since the thicker products are heavier than the thinner ones, a higher air pressure is needed to lift up the products. The air regulation is used only in IRB660.

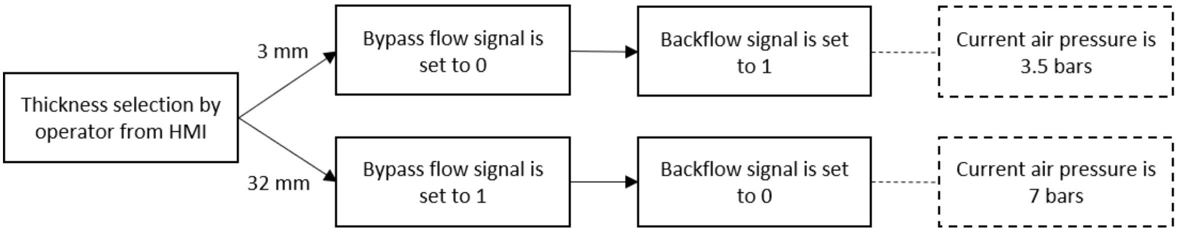

Figure 27. Air pressure regulation conditions

The second procedure is used to create a logic for the pallet sensors. Two sensors at each pallet slot controls the existence of the pallets. The pallet statuses are updated based on result of a set of conditions which were created by using sensor inputs. One of the sensors gives the signal when the pallet exists. Another sensor is inverted to give 0 value when the pallet exists. The reason of using the sensors in this way to reduce possible sensor errors. If there is any electric loss or any material except the pallet covers the sensor area, both sensors shall give the same value and the error shall be identified. Whenever the first sensor gives the value of 1 and the second sensor gives the value of 0, the pallet sensor status is updated as pallet exists. If the values are in the opposite way, the pallet sensor status is updated as no pallet. Any other case leads the controller to the error situation. When the error situation is triggered, the controller does not directly go into the error status. There might occur temporary errors during the pallet placement. The sensor might change its value frequently until the pallet is correctly placed to the slot. Furthermore, the sensor might start blinking because of the wood dust or similar objects. A timer is set to avoid the possible temporary error situations. Otherwise the process might be interrupted frequently. The controller activates the timer if there is a mismatch in the signal logic. The controller checks the clock status (which is raised to 1 when there is a mismatch in the pallet sensor logic), compares the timer with the maximum time limit, and checks the current pallet status. The pallet status is checked to ensure not to repeat the error messages. If all the conditions are maintained, the controller calls the pallet error procedures. If the pallet status is no pallet exists, the pallet sensor logic indicates that there is a pallet in the slot and, there is no pallet error currently, the pallet status is changed to pallet is present. If the pallet sensor logic is in error state but there is no pallet in the station currently, another condition is used to reset the pallet status. The reset logic is maintained if the operator resets the pallet error from the HMI under the given conditions above. The pallet error procedure contains the buzzer activation and error message indication. Firstly, the buzzer is activated to get attraction of the operator. Meanwhile, the error message related to the pallet error is displayed on the flex pendant screen. The error message contains the information of the slot where the error occurred, explains the possible reasons causing the error and the possible approaches to solve the problem and, if the error is not cleared up, it directs the operator to the user manual to get more detailed information.

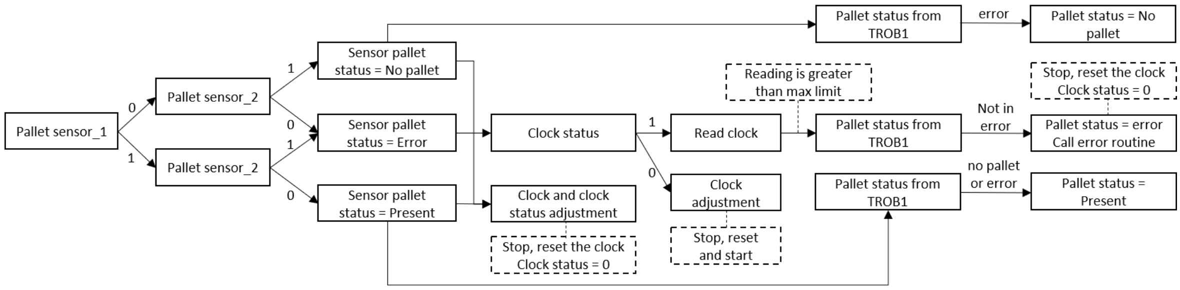

Figure 28. Pallet sensor logic

The third procedure was created to increase the control over the vacuum groups. A set of conditions are written in order to avoid excessive amount of air usage. The outputs which are sent to the vacuum generators are controlled by the OS inputs during the product transfer. The vacuum generators are activated or deactivated by the OS inputs according to the product type sequence.

Table 6. Vacuum generator control according to the OS input and product type

	Prod-1	Prod-2-4-5-7-8	Prod-3-6-9
OS-1	Vac-1	Vac-1 & Vac-2	Vac-1
OS-2	Vac-2 & Vac-3	Vac-3	Vac-2 & Vac-3
OS-3	Vac-4	Vac-4	Vac-4
OS-4	Vac-5	Vac-5	Vac-5
OS-5	Vac-6 & Vac-7	Vac-6	Vac-6 & Vac-7
OS-6	Vac-8	Vac-7 & Vac-8	Vac-8

If there is no product attached to any section of the gripper along the product transfer from pallet to conveyor or vice versa, related vacuum generators are deactivated by the program logic with the help of the OS readings. Vacuum groups and optical sensors relations can be observed from Table 6. If a product is dropped during the transfer, controller halts the manipulator action and gives an alert to the operator. When operator goes into the station to replace the product on the gripper, second part of the logic takes place. After the operator replaces the product on the related section, the vacuum generator is activated by the OS when the input of OS turns to 1. This action takes place with a delay. The delay is required to give enough time to the operator due to avoid mispositioning of the product.

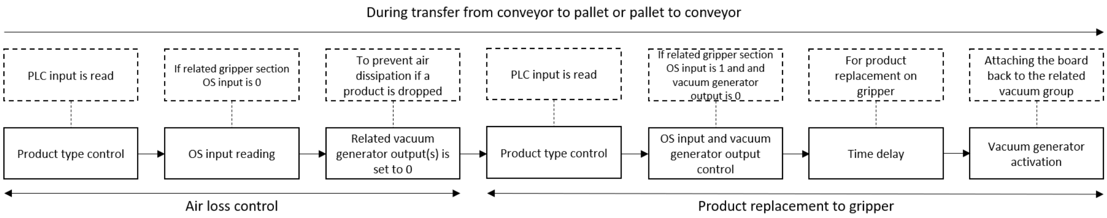

Figure 29. Air loss control and product replacement to gripper flow chart

The controller counts the number of picked or released product(s) during a cycle. The data stored for this purpose controls the vacuum generator sequence during picking or releasing according to the process. The number can be lost in case of an error or a malfunction during a cycle. Such a scenario may cause a conflict for the following cycle. The controller may activate or deactivate wrong suction group during picking or releasing. In order to avoid such circumstances, group output of vacuum generators is checked continuously by another logic. If the group output of vacuum generators is zero, then the picked panel number is set to zero as well.

The buzzers in the stations are controlled by two different procedures. The reason of using two procedures is to make difference between the alerts. Two different alert types are used according to the purpose of the alert. First alert type is activated when someone gets into the robot area. The second alert type is activated if any error occurs during a cycle. The difference between the alerts is the activation time of buzzer. While the buzzer is activated for 0.2 and 0.8 second intervals in error cases, it is activated for 0.5 second intervals in motion halt cases. The operator can understand the difference between the alerts with the help of different sequences of the buzzer actuation and act according to it.

Next procedure checks the pallet statuses to overwrite a persistent boolean data for the activation of warning lights. If one of the pallet statuses is raised to the error level, then it overwrites the boolean data to FALSE and this data is read by the warning light control procedure. This procedure is used to transform a numeric data into a boolean data. With the help of this, it is easier to understand the programming logic later on.

Warning lights are used to increase operator awareness. Each light indicates a different status. Details about the warning lights shall be given in safety section. The buzzer is also located on the light stack but since the procedure controls the buzzer activation was mentioned before, only light control logic shall be mentioned in this subsection. To control the lights on the stack, a procedure was created. The green light on the stack is activated when there is no error in the station. The condition is satisfied when there is no one in the station, which means the area scanner does not detect any anomaly in the station, and the controller is in task executing state. The procedure takes the task executing information from a digital output. The digital output was registered as a system output. Registering an input or an output allows giving instructions or receiving information to/from the controller. The system output continuously reads information from the controller about the task execution. Task execution status means there is no error in the program flow or in the controller. Station area control is done by the area scanner. According to the information taken from the scanner trap routine, area clear message is received. In the end, if both condition inputs are satisfied, controller activates the green light. Yellow light is controlled by the following condition in the same procedure. The condition is satisfied by two signals. Both signals are resultants of cross connection signals. First signal is related to motion pause of the manipulator. It checks the scanner signals and mode of the controller. If the scanner signals are high, that means it detects something unexpected in the station. Controller mode is checked because only in automatic mode the scanner is allowed to stop the manipulator. If both conditions are maintained, then the motion pause signal is set to 1. Second signal is related to a stop motor command. It is set to 1 when the controller is in manual mode, the scanner detects something unexpected in the station and controller is not in task executing status. Previously, it was possible to run the controller even the scanner detects something unexpected in the station when the controller status is switched to automatic from manual mode. This cross connection was created to prevent this situation. If one of the cross connections is set to 1, the yellow light is activated. The condition controls the red light is satisfied by a digital output and two persistent boolean data. The digital output is set to 1 if any procedure in ERROR Module is run. The following two boolean data is set to TRUE if any error occurs in pallet statuses. Independent from each other, any of the statements in the condition is satisfied, then the condition sets a digital output which controls the red light to 1 and the red light is activated.

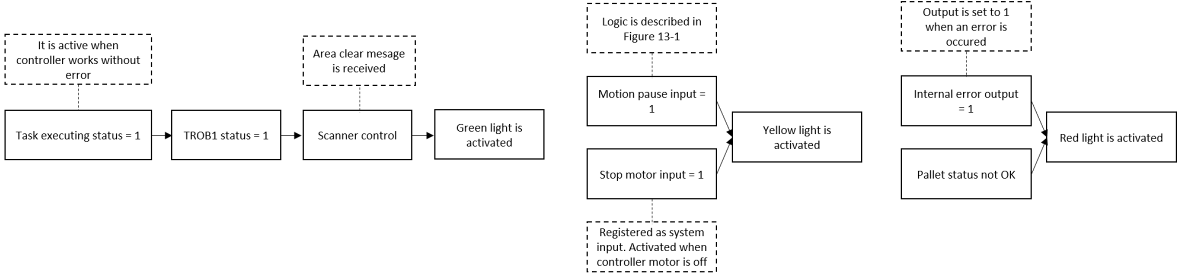

Figure 30. Light stack activation

Information about pallet existence is declared to the area scanner from controller in order to avoid fault state in area scanner logic. The information is obtained by the pallet sensor logic procedure. According to information received from pallet sensor logic, a procedure overwrites a variable numeric data. The numeric data inputs are as follows: only right pallet exists, only left pallet exists, both pallets exist and, no pallet exists in the station. The area scanner has three inputs. Six different conditions can be obtained according to input signal values. These conditions are:

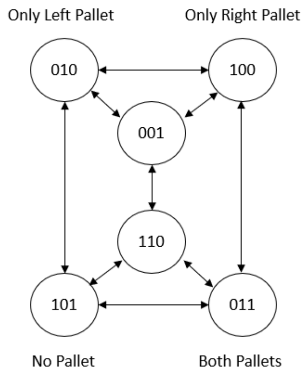

Figure 31. Area scanner zone logic

There are four possibilities of pallet statuses in the stations as it is seen from the Figure 31. The area scanner logic must be created according to the possible pallet existence scenarios. The scanners have to be programmed according to understand if the current configuration of the pallets is okay or not. This is very important to understand if someone or something else is standing instead of a pallet in the pallet slots. The scanners have to be programmed correctly to avoid the possible hazards.

The pallet sensor logic procedure gives out a numeric value for each pallet existence. For example, if only left pallet exists, then related pallet status numeric data is set to 1. If the left pallet does not exist, then the related pallet status numeric data is set to 0. According to the pallet status numeric data, the scanner control procedure creates the possible scenarios. Firstly, the pallet statuses are checked. If the controller is just turned on, then it must estimate the previous pallet status since there is no information about the previous pallet configuration in the station (The previous pallet configuration information data is set to 0 if the controller is reset. This is a necessary measurement since there might be changes in pallet slots while the controller is turned off.) The previous pallet configuration estimation does not have to be 100% correct. It is needed to initialize the previous status numeric data for the area switching process since 0 value for previous status is not included in area switching conditions.

The initialization contains possible previous status of the station. For example, if both pallet slot sensors give 0 signal to the controller, the previous state could be only left pallet exists or only right pallet exists since there is no possibility to operate with two different products at the same time. Or, if only one slot sensor gives 1 signal to the controller, the previous state could be no pallet exists or both pallets exist in the station. According to the estimations, controller initialize the previous pallet status configuration. Besides, it sets the current pallet status configuration based on the pallet slot sensors' inputs. The estimation part is not necessary if the pallet change is done when the controller is already ON.

The area switching procedure was created to switch inputs signals of the scanners according to pallet existence in the stations. The input combinations are as in the Figure 31. The controller gets the configuration from the previous procedure and sets the input signals according to it. Sometimes, scanner cannot jump directly from one state to another. For example, if the state is only left pallet exists and the controller would like to change the scanner zone to both pallets exist, then it has to follow the route (previous pallet status initialization is made for this purpose). Firstly, it shall go to the no pallet exists or only right pallet exists state, then it can go to the both pallets exist state. 0.5 seconds wait time was added between passing states to avoid fault status in scanners. In the end, the data from area to controller and controller to area is equalised until the next status change.

5. HUMAN MACHINE INTERFACE DESIGN

5.1 Description

The human-machine interface was designed to monitor the process variables, possess control over the gripper and sensors and give a better understanding of the system to the operator during the palletizing and depalletizing processes. Besides the indicators in the HMI, it also includes signal information which come from the robot controller to increase awareness and find out solutions for the errors expeditiously.

Both HMIs were designed in Robot Studio's Screen Maker. Screen Maker has directly access to the robot controller which enables using and controlling the system parameters and signal information. What is more, it includes such features that increases awareness of the operators by using led, bar graphs, custom images, numeric editors, action triggers etc.

There are differences in the HMIs of the IRB460 and IRB660 since some processes are different and require different treatments. The HMI of IRB660 includes more options compared to the HMI of IRB460. The reason is IRB660 requires more delicate and detailed operations such as multi-layer control, mechanical barrier control, thickness selection, the product information etc. Besides the differences, there are common traits such as pallet and height indicators, gripper control panel, process monitoring etc. In general, all HMIs consist of three different screens.

The operators were trained before the commissioning of the line to learn how to use the HMI and cope with the pre-defined errors which might be encountered if any inconvenience is occurred during the process. Furthermore, two HMI manuals were written in order to consolidate the training. Possible error scenarios and the solutions of these errors are explained in the manuals.

There are two ways to reach the HMI screens. The robot controllers automatically open the HMI when they are turned on or restarted. If the screen closed for some reason, it is possible to reopen it from the main menu of the flex pendant.

There are two versions of HMIs uploaded to the controller. The first one is written in English and the second one is written in Estonian.

The HMI part shall not be divided into two parts as IRB460 and IRB660 since there are quite many similarities in the interface structures. The differences shall be explained during the related part.

5.2 Main Screen

The main screen of IRB660's HMI consists of eight main sections whereas the main screen of IRB460 consists of five main sections. Each section in both HMI's is designated to monitor a different part or a process of the station. They show process status and enable the control of particular components in the station. The whole five sections in the IRB460's HMI also take place in the IRB660's HMI.

First section is connected to a string defined in the main module and it monitors the particular process that the robot performs at that moment. Whenever program cursor jumps into a new process, the string is overwritten by the name of the new process with help of a line at the beginning of that particular process. After that, the text bar on the HMI reads the overwritten string and updates itself. The strings defined in IRB460 and IRB660 are;

- Height search process (Only in IRB660)
- Picking process
- Placing process
- Retracting process

Second section is designated to assign number of the cycles before the production ends. The function is used to inform the robot beforehand in case of end of a shift or a production material change. The operator enters the numeric editor and writes how many cycles is needed. After compilation of the number of written cycles, robot goes to its main position and stands still until a new pallet is provided to the station. A persistent numeric data is defined in the main module for the action. The controller counts down the overwritten value and when it reaches to zero, it halts the robot. The second section is only necessary for the IRB660. There is no need for such function in the IRB460 since the robot stops functioning whenever a pallet is filled/removed, or the conveyor stops feeding the ready products.

Third section is created to monitor the ongoing product type properties. A text bar shows the product code, the product thickness, the product dimensions (length x width) and the product amount on one layer of the pallet, respectively. The product types are defined in the main module of the controller. A persistent product data is overwritten whenever the new product information is delivered by the press. The product data is read by the text editor in the third section and it is monitored continuously. Third section is in use in both IRB460 and IRB660 interfaces.

Fourth section monitors the active gripper vacuum components. There are eight groups on the screen. Whenever a vacuum group is activated by the robot or an operator, the HMI changes the background colour of the related group to yellow to indicate that the vacuum group is in use at that moment. Each vacuum group indicator is connected to the related signal that activates the vacuum generators. Sub-section "picked" is controlled by three different signal groups, which are the vacuum generator signals, the optical sensor signals and the capacitive sensor signals, come from the gripper. Whenever a group of action is done, which means a vacuum generator is activated, the related optical sensor detects the product and the related capacitive sensor confirms that there is only one product is picked, the counter next to "picked" label raises itself by one. This section is common in both IRB460 and IRB660. The fourth section leads the user to another page when the user taps to the related graphical area. The new page enables a group of actions for the user which are related to the gripping process. This page shall be elaborated in the following sub-sections.

Section five enables the thickness selection for the production. The processes that the IRB660 follows vary according to the panel thickness. While the process for thick panels is straight forward, the process for the thin panels has more detailed steps. Furthermore, it is important to know the panel thickness to calculate the pallet height correctly after each picking process. To provide the correct information to the robot, the fifth section is created. There are two buttons for 3 mm and 32 mm panel thickness values and whenever a button is clicked, it turns its colour to green. The section five is only exists in IRB660's HMI. There is no need of use the thickness selection in IRB460 since all the products have the same thickness value. After a new pallet is placed to the station, it is impossible to run the robot until a thickness value is selected.

Section six is dedicated demonstrating pallet status in both IRB460 and IRB660 robots. There are two labels in the sixth section and the labels represent the pallets in the station. The labels are set to present different colours and texts according to changes in the pallet status. There are five status in total which are:

- No pallet in the station (grey colour)
- New pallet is presented (blue colour)
- The pallet in use (yellow colour)

- The pallet is full (green colour)
- Error (red colour)

A persistent numeric data is used in programming logic to provide information about the pallet status. The logic behind works as a counter. It uses information comes from the pallet sensors and the robot controllers. Pallet sensors work as a couple in each slot. If one sensor acts differently than the other one, then the error is raised by the controller.

Section seven consists of two bar graphs representing the pallets in the station. Bar graphs work differently in IRB460 and IRB660. In IRB460, bar graphs raise themselves by one after a layer of products is placed to the related pallet. It has a maximum value of 35 layers. Whenever a new pallet is placed to the station, controller resets the related bar. In IRB660, the bar graphs count the depalletized layer number. The reset logic is the same with the IRB460. The fourth section leads the user to another page when the user taps to the related graphical area. The new page enables a group of action for the user related to the pallet operations. This page shall be elaborated in the following sub-sections.

The eighth and the last section on the main screen is designated to expand awareness of the users for the mechanical barriers at the beginning of the process. The mechanical barriers are used to limit the thin panel movements during separation process. Since dimensions of the panels are different for each product type, the mechanical barrier distance has to be adjusted depending on the product width. If the operator forgets the distance adjustment of the mechanical barriers, the barriers may hit the stack and damage itself or the products. Section eight is used to avoid such consequences. The click button sets itself to “NO” when a new pallet is placed to the slot and the robot shall not work until a new selection is made by the operator. The section eight only exists in IRB660’s HMI since there is no mechanical barrier in IRB460.

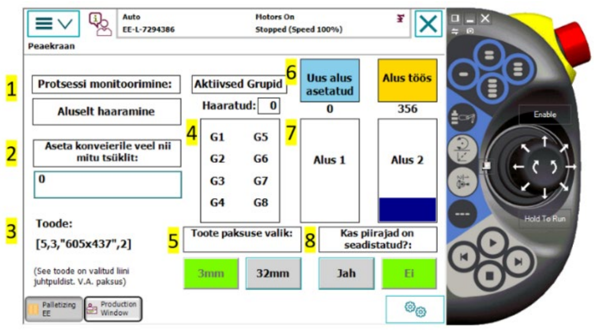

Figure 32. The main screen of the IRB660's HMI

5.3 Gripper action control screen

The gripper action control screen consists of ten buttons and twelve indicators. When an operator taps the graphical area that takes place at the middle of the main screen, the gripper action control screen appears. The screen is in use to control and monitor the gripping process in case any error occurs during the transfer of the products between the pallets and the conveyors. It exists in both IRB460 and IRB660 HMI's.

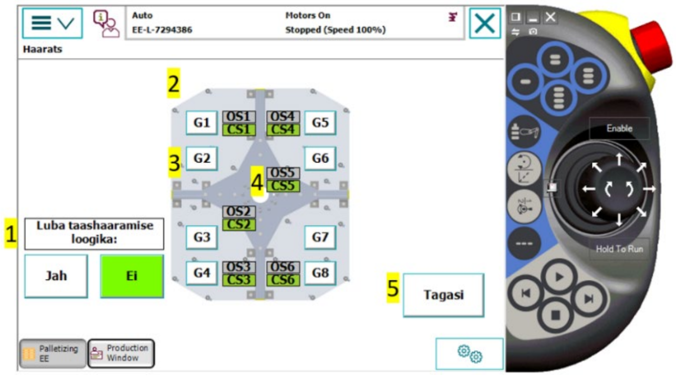

Figure 33. Gripper action control screen

Section one consists of two functional buttons that activates or deactivates the re-gripping logic of the controller. The re-gripping logic is created to use in case of a picking process fails. Whenever a product drops during a picking process, it is hard to pick the fallen product when the related gripping section is active. The vacuum generator has to be deactivated and then activated again to obtain a firm grip by the suction cups. Otherwise, because of the air leakage during the replacing the fallen product to the gripper, it is almost impossible to obtain a firm grip. To provide the opportunity to the operator, a logic is used in the controller that, when the yes button is clicked, it provides the deactivation and reactivation automatically with a 1.5 seconds delay to the operator while the operator places the fallen product to the gripper.

Section two consists of eight buttons at the middle (Group 1 to Group 8) to control the signals that activate the vacuum generators. A gripper image is placed behind the buttons to ease the understanding of the button and sensor locations. It has two main functions. The first function is the indication of the active groups. Whenever a vacuum generator is activated by the controller, the button turns its colour to yellow to indicate the activation. The second function is the manual activation of the vacuum generator. If any of the illustrated buttons is clicked by the operator, it activates or deactivates the related generator depending on the signal status. The indicators labelled with OS and CS represent the optical and the capacitive sensors, respectively. Whenever one of the gripping groups picks a product, the mentioned sensors above checks the picking process in case of a product fall or a multi-picking. If everything is okay, indicators turn their colour to green while they are initially grey.

Fourth and the last section consists of a single button which is created to go back to the main screen. When the button is clicked, it closes the current screen and shows the main screen immediately.

5.4 Pallet control screen

Pallet control screen was created to control actions and indicate the changes in loading/unloading area. There are two pallet control screens which are functionally identical for the pallets in the stations. When the bar graphs, numbered as 7 in section 7.2, on the main screen are clicked, it directs the user to the related pallet control screen. The pallet control screens are used in both IRB460 and IRB660 HMIs. There is one additional button in IRB660's HMI numbered as 3, which shall be elaborated in this section.

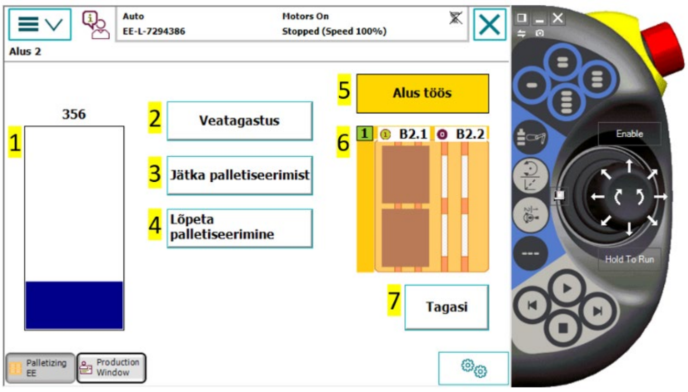

Figure 34. Pallet control screen

First section has the same visual function as it is described in section 7.2. In addition to its visual function, a numeric editor is attached to the bar graph in IRB460's HMI. When the bar graph is clicked, a numeric editor appears. The numeric editor enables inserting palletized layer numbers. The function was added to be used in case of using an uncompleted pallet from a previous production. There is no need to use such feature in IRB660 since it is able to search the height of the pallets.

Second section was designated to reset errors in pallet status. An error is raised in case of an anomaly in pallet sensor signals. Abnormal sensor triggering or an inconsistency in sensor inputs can raise an error. Operator clicks the reset error button to reset the error. When the button is clicked, a pop-up screen appears. Controller asks to the user about proceeding with the error reset. If the user clicks the 'YES' button, controller sets the pallet status to 'New Pallet' in IRB660 while it sets the pallet status to 'Pallet in Progress' and asks the layer number in IRB460. The reason of the difference serves the same purpose in theory. When a new pallet is provided to IRB660, the controller sets the pallet status to 'New Pallet' and it searches the height. After the height detection, the pallet status is changed to 'Pallet in Progress'. Since it is easy to count layers in IRB460, the controller skips the 'New Pallet' status and asks the layer number directly. If the operator selects 'NO' button, pallet status is set to 'PalletFinished' in both stations. There is also a 'CANCEL' button to go back to the pallet control screen.

In some cases, even a pallet is not totally empty, production can be terminated due to shift hours. When the pallet status is set to 'PalletFinished', the operator has to take the pallet out and put it in the slot again to reset the sensors. Since it takes quite much time and effort, an additional button is added to set the pallet status to 'New Pallet' in IRB660. An operation can be continued easily with the help of this button in third section. The same button was not added to IRB460's HMI because, the production on IRB460 station is calculated and the operators do not leave the shift without completing the order.

Fourth section is designated to terminate the palletizing process in related pallet slot. When the button is clicked, the controller sets the pallet status to 'PalletFinished'. When the pallet status is set to 'PalletFinished', the controller stops functioning in the related slot and if there is a new pallet in the other slot, it starts palletizing/depalletizing the other pallet. This section is common in both IRB460 and IRB660 HMIs.

Fifth section consists of a text editor with multiple status. It indicates the pallet status with different texts and colours as it is described in section 7.2 (number 6). It was added to the pallet control screen as a reminder of the status. It is common in both IRB460 and IRB660.

6. LAYER SEPARATION TESTS

6.1 Description

Multiple layer picking became an issue right after program testing of the depalletizing station had started. Manipulator (IRB660) picks more than one layer in one or several vacuum groups for the products which have 3 mm thickness value. Porosity of HDF boards causes this issue. After vacuum gripping initiated, more than one product may be picked because of the porosity. After depalletizing, there comes the gluing process, and multi-layer picking situation is undesired because of structural inconvenience of gluing mechanism. Same issue is not observed with the boards which have 32 mm thickness because board weight is sufficient to separate layers. Picture below depicts the error:

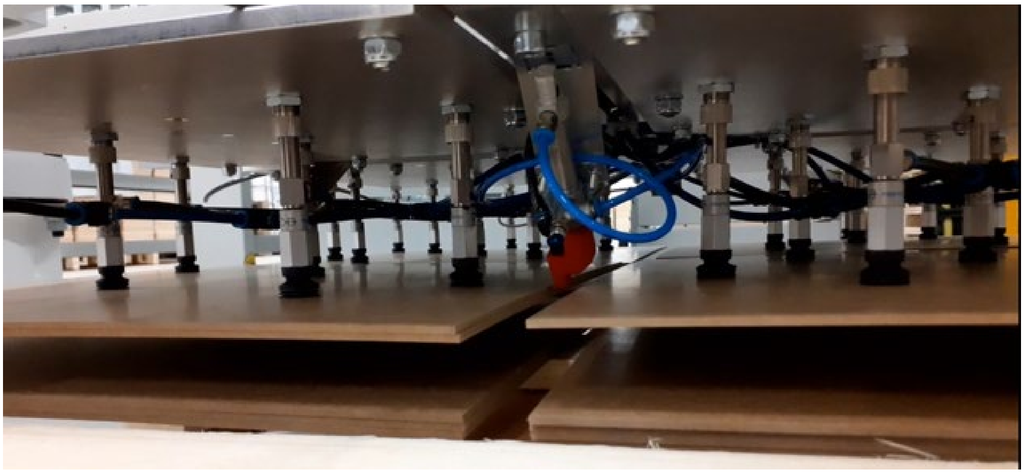

Figure 35. An example of double layer picking

Occurrence of the error was examined for two days to see does it really affect the overall performance of the process. IRB660 was employed for only depalletizing purpose. Scenario was picking the boards from pallet and stopping the robot after 300 mm offset from pallet surface. Then, operator was collecting the boards which are attached to gripper. This cycle was run for several pallets. Full process (placing the boards to conveyor) was not run to use the time efficiently. Result was extraordinarily confusing because of inconsistency of product quality. While some batches were in good condition and additional separation is not needed for almost 90% of a stack, some batches were quite inconsistent in that manner. After observation of depalletization of different pallets, it was decided that an additional separation tool and/or process is needed.

Firstly, the equipment already installed on robot gripper was used to solve the issue. Different methods were tried to separate layers but most of them was insufficient to separate the layers. Some methods were promising, but overall performance of those methods was unsatisfying. Afterwards, it was decided to use additional tools to solve out the problem. Capacitive sensors are used to detect multi-layer picking. Detailed description is given in gripper features section. The methods were tried to solve the issue is listed below:

- Increasing / decreasing vacuum surface area
- Regulation of infeed air
- Regulation of offset speed and height
- Blowing air through the boards (vertical to board surface)
- High frequency vacuum activation / deactivation
- Sliding boards along the pallet surface
- Blowing air from inner edge of the boards (parallel to board surface)

6.2 Vacuum surface area

Initially, it was thought that changing the vacuum area may solve the problem. First suction cups attached to gripper had 20 mm diameter. Suction surface area calculation considering the cup shape is a circle, and there are 36 suction cups on gripper:

$$\pi. 20^2. 36 \cong 45240\ mm^2\ (total\ area\ of\ suction\ surface)$$

This area is divided between the boards according to product number in one layer. It was desired testing a narrower suction surface to see if it helps solving the problem. It was thought that current vacuum surface area might be redundant for the current product configuration. In order to reduce the surface area, suction cups were switched with 10 mm diameter cups. New surface area calculation would be:

$$\pi. 10^2. 36 \cong 11310\ mm^2\ (total\ area\ of\ suction\ surface)$$

The vacuum area surface was shrunk four times in order to separate the boards. The separation was more stable with this configuration but shrinking the vacuum surface area arose a new issue. The gripper started failing with picking process of 32 mm thick boards. Besides, the method didn’t solve the multi-layer picking issue completely.

Second approach was reducing the number of pneumatic components on the gripper to reduce the vacuum surface area. This test was carried out with 20 mm diameter suction cups since 10 mm diameter suction cups are not able to pick 32 mm thick boards. There are eight vacuum groups on the gripper. While group 1, 4, 5 and 8 contains six suction cups, group 2, 3, 6 and 7 contains three suction cups. Two elements were removed from vacuum group 1, 4, 5 and 8, and one element was removed from group 2, 3, 6 and 7. This test failed in both multi-layer picking and 32 mm boards picking. It was observed that, remainder vacuum elements were not able to create enough suction force in order to lift 32 mm boards. Besides, there was just a slight change in multi-layer separation performance which didn't make a jump in the overall performance. Result of this tests can be summarised as:

- Usage of 20 mm diameter suction cups must be maintained
- Number of vacuum components on gripper must not be reduced

6.3 Regulation of infeed air

Infeed air comes to the station from a compressor. The air is delivered to manipulator end effector via an air hose through manipulator's second and third axis. There is an air regulator (Festo MS4-LR-1/4-D6-AS) mounted on the gripper. It has a range of 0 to 7 bars. Initially, all boards with 3 mm and 32 mm thickness values were picked with 6 bars air pressure. While this value was okay for 32 mm boards, it was apparently high for 3 mm boards.

Firstly, a common air pressure for both thickness values was investigated. Tests ran in the range of 3 bars to 6 bars. While lower air pressure enhanced the layer separation performance considerably, it affected the 32 mm boards picking process in a negative way. 32 mm boards started to fall during picking process.

The intermediary result arose the idea of using different air pressure values for different thickness values. In order to test the idea, two pieces of 3/2 solenoid valves (SMC VP742R-5YOD1-04FA and SMC VP742R-5YZD1-03FA-F) were installed to gripper. The air comes from ground was split into two hoses with the help of a T type air fitting. While one hose goes directly to the regulator, second hose bypasses the regulator and goes directly to the one of the solenoid valves. The regulated air (4 bars) goes to the second solenoid valve after the air regulator. In this way, it is possible to select the air pressure by setting the signals of the solenoid valves. Second solenoid valve was installed in order to prevent back flow comes from the bypass valve. In case of using a single solenoid valve, ground air shall be flowing back to the regulator and it shall weaken the picking pressure.

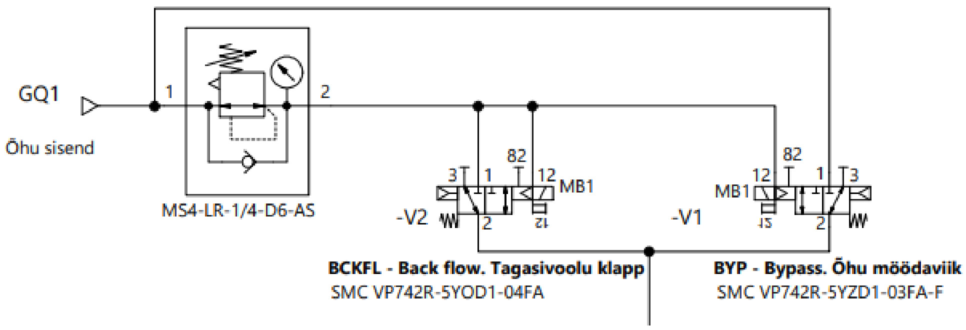

Figure 36. Pneumatic scheme of the air regulation

As a result, the air regulation enhanced the performance of layer separation and also maintained the performance of 32 mm boards picking process. But this test also didn't remove the problem totally. It was observed that an additional solution was required to solve the issue.

6.4 Offset speed and height adjustment

Offset speed and height adjustment is made in order to enhance the picking performance. It was observed that, depart after picking process affects the overall picking performance. If the depart speed is too high, and the gripper picked more than one layer in this particular picking process, weight of the extra layers detach the first layer from the gripper. It was observed that, if the depart speed is reduced, separation of extra layers is easier. With the reduced air pressure, boards stuck each other started releasing themselves after a second. The method aimed to solve two problems. First, it reduces the chance of dropping all products right after picking process. With a slower speed, vacuum cups are able to grip the boards firmer. Second, it helps separating extra layers from the first layer vacuumed by the gripper. Height adjustment is made to prevent boards sliding on each other after the layer separation. If the offset height after picking is too much, then boards fall from a higher level. Because of the smooth surface of HDF boards with the acceleration they gain during the free fall, fallen boards slide and lose their standard position on the pallet. It also increases the chance of multi-layer picking for the following cycle since mispositioned boards allow extra surface to gripper for the layer under the top surface of the pallet.

6.5 Sliding the boards along the pallet surface

A mechanical resistance was created in order to separate the boards. If additional layers were detected after the picking process, the gripper was sent to 10 mm under the picking height and the boards were pressed to downwards (step 1 in Figure 20). The next step was driving the gripper towards the conveyor direction for 10 mm to 20 mm without deactivating the vacuum generators (step 2 in Figure 20). Final step was the departure of the gripper from pallet surface (Step 3 in Figure 20). The idea was generating resistance by using the friction between board surfaces, and the extra layers underneath the top boards may lose their bond with the top layer.

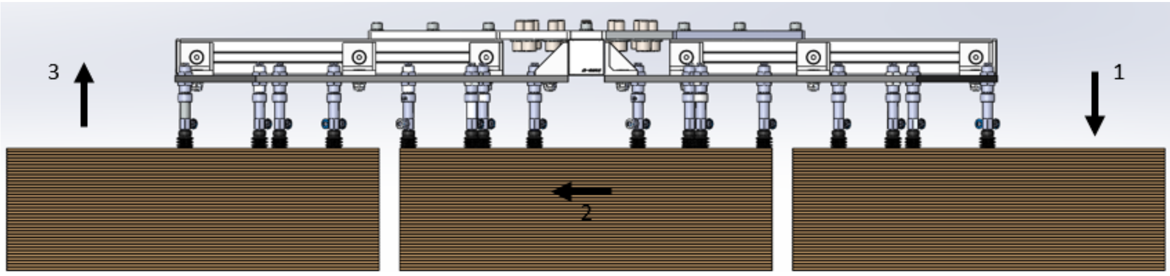

Figure 37. Board sliding illustration

When the gripper slid the boards towards the conveyor, two problems were observed. The first problem was deformity of the suction cups. The suction cups were folded because of its bellow structure and rubber material. The deformity caused connection lost between the boards and suction cups. Second problem was that extra layers were travelling with the top layer. The method didn't help to separate the layers at all. After few trials, it was seen that the method was not promising, and the tests came to a halt.

6.6 High frequency vacuum activation / deactivation

Triggering the vacuum generator in a high frequency creates vibration effect on the boards. Extra boards stuck to the top board have a weaker bond compared to the top one aimed to be picked by the gripper. It is because of the lower vacuuming effect due to the distance and additional weight of the boards. A high frequency vibration may separate the boards using the advantage of the weak connection of the lower boards. The picking program was adjusted to test the new method. After the picking process, if capacitive sensors detect additional boards on the gripper, the controller starts triggering the vacuum generators with small periods.

When the period was lower than 0.2 seconds, the vibration effect was barely observed. When the period was higher than 0.5 seconds, the gripper failed in holding the boards. The best vibration effect was obtained with the period of 0.3 to 0.4 seconds. When the desired vibration was obtained, the method was tested in order to separate the extra boards.

The result was unsteady. Even though it worked well during few tests, overall performance was unsatisfactory. First reason was the long time that spent for the separation because of the inconsistent board porosity. Second reason was dropping all boards which were stuck to each other. Even though the method gave better results compared to few other tests, the decision was negative because of the inefficient time usage and inconsistency in holding the top board.

6.7 Blowing air through the boards (vertical to board surface)

The porosity of the boards brought the idea of a reverse process to see is there any improvement on the current issue. Blowing air through the boards before picking process may help with layer separation in that manner. Bypass valve (SMC VP742R-5YOD1-04FA) was used to blow air directly to board surface. In this case, vacuum generators were bypassed, and the pipe comes from the bypass valve was directly connected to the suction cups. With the current configuration, there is only one air source to the vacuum generators which is the regulated air (4 bars). Since the test was done on the 3 mm boards, the 32 mm board picking were neglected by using the bypass valve for air blowing purpose.

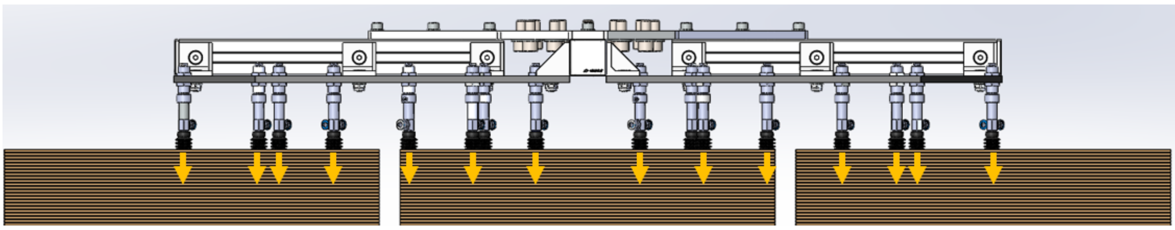

Figure 38. Blowing air vertically through boards

After a test with one full pallet, the result was not satisfying. First issue came along was the noise. When the gripper blows air through the boards, a high level of noise was appearing even though the surrounding noise. Second issue was insufficient and inconsistent performance of the test. Some boards were still attached to gripper after fourth trial of blowing air in a row. Considering that the method was implemented brought more issues to the process, it marked as an ineffective test.

6.8 Blowing air through the boards (parallel to board surface)

Second trial of blowing air was done parallel to board surface. First air blowing was done vertical to board surface because the standard equipment on the gripper was installed in vertical direction. Before ordering or designing any new equipment to the gripper, test was run with an air gun. Firstly, a multi-layer picking was performed in order to run the test. Afterwards, air was blown with the help of the gun between the boards. The result was very promising. After a couple more tests, initial results can be summarised as:

- Separation performance was much higher compared to other tests
- Time efficient method since one stroke of air mostly separates the layers
- Risk of fallen board gliding towards outer surface of the pallet due to pressurised air

The initial results were encouraging to move one step forward. A mechanism was designed in order to blow air between the top layer and the rest. The mechanism was attached to two hinges and actuated by a cylinder to move as a swing. This action is required because of changes in the vacuumed product height. An additional solenoid valve was installed in order to control blowing action. The air pressure goes through the valve is controlled by a pressure regulating fitting.

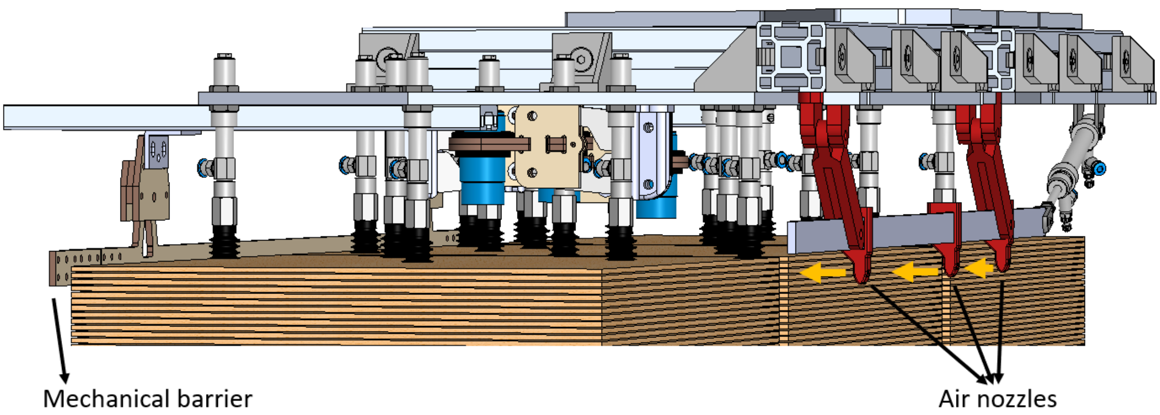

Figure 39. Section view of the gripper for horizontal air blowing process illustration

After installation of the mechanism to the gripper, second part of the test was begun. The mechanism was placed centre of the gripper and air is blown towards both outer sides of the gripper. Air regulation of the blowing air was done to minimize the board glide.

Intermediary results showed that the method was efficient when air regulation of the vacuum generation was used. So, picking process of the 3 mm boards were done with 4 bars of air pressure. With lesser vacuum generation, which is sufficient for the 3 mm boards proved by the previous tests, it was easier to separate extra layers. On the other hand, it brought the issue of gliding boards towards outer sides of the pallet. A mechanical stopper was designed and installed to prevent the undesired movement of the separated boards. It is adjusted manually by the operator depending on the product type.

Overall, the combination of solutions worked well together. The separation is done by the attached swinging mechanism and order of the products on the pallet after separation is controlled by the mechanical barrier. Even though it does not remove the multilayer picking completely, it is sufficient by the customer. The customer agrees that the inconsistent porosity of the boards cannot be changed by the board provider. Thus, current solution was accepted by the customer.

6.9 Conclusion

Seven methods were tried to perform separation of the extra boards stuck to the top board. Different sources were used to identify the root cause. Internet search, brain storming with robotic team, and customer consultation were done in order to generate a solution. In the end, tests were done in the field gave the best result.

Some methods were identified as ineffective, although the results contributed in a way to find out the final solution. Adjustment of vacuum area showed that the best gripping was provided by the 20 mm diameter suction cups for both board thickness values. Regulation of the air pressure showed that diversity in air pressure increases the overall picking performance. Offset speed adjustment strengthened the knowledge about overall holding performance of the suction cups. Offset height adjustment was effective to understand the fact about board gliding issue. Blowing air from top surface of the boards brought the idea of blowing air towards inner edge of the boards. Even though not all the results directly contributed the solution, they led us to figure out the final solution.

Diversification in the porosity of the boards was the main problem. It was observed that while some batches came from provider can be separated without extra effort, some batches were extremely resistant for the separation. It was also discussed with the provider by the customer, but it was asked from ABB to solve the problem in order not to stop the production.

Tests, including extra layer detection and separation of the layers, and generation of the solution had taken an extra month until the final solution was implemented. The result was satisfactory for the customer. Even though it added an extra time to the estimated cycle time, the overall performance of the station is able to provide the raw material to the gluing machine without stopping it. If the solution fails after three trials, it was agreed that the operator shall interfere and make the separation manually.

7. SAFETY

Safety equipment, as implied by its name, is a fundamental part of any robotic cell to provide sufficient protection for both workers and system equipment. Mainly, the equipment helps avoiding serious injuries and casualties may arise by nature of risk associated in the work. General reasons why the safety equipment is vital in the stations can be summarised as:

- Avoid human & economic losses
- Be free from legal risk/penalties from bodies like member states (for European market)
- Figure out health & safety requirements for applications
- Globalize measurements for the products & applications
- Obey the law & approvals
- Protect the brand image

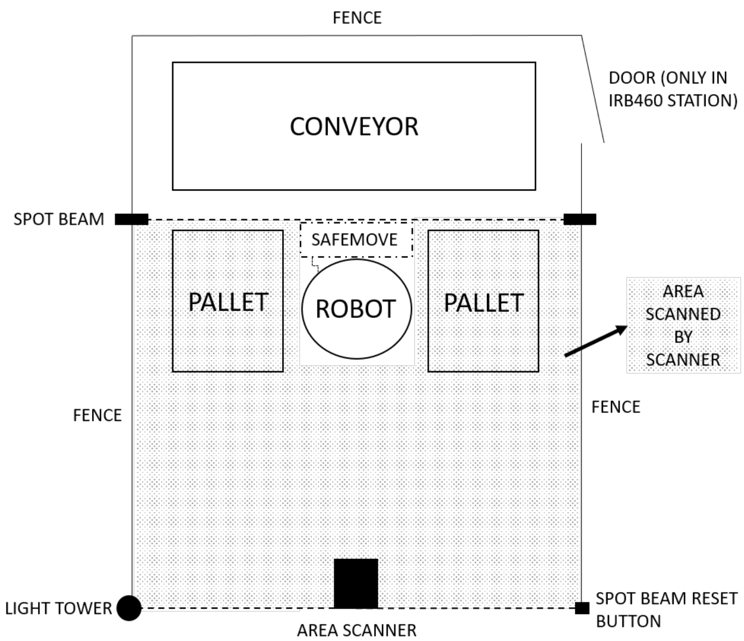

Figure 40. Safety equipment in the stations

The equipment depicted in the figure are stated sequentially below:

- Safety fences
- Spot beam
- Spot beam reset button
- Area scanners
- Light tower
- SafeMove
- Sense7 door switch

Safety solutions were discussed with customer. Fence height and safety distances were calculated according to ISO 13855 and 13857 standards. In addition to these, area scanners and light beams were placed to increase safety options. Finally, SafeMove option of ABB IRC5 controller were used to increase stability and safety.

7.1 Area scanning

Area scanner is a safety equipment that provides safeguard for static applications such as robotic cells. Datalogic SLS-SA5-08SLS model area scanner is used in each cell. It offers monitoring and dynamically changing the shape and the position of safety and warning areas up to six configurations from a single point, and dust protection. The range for the safety area is 5.5 m over 275 degrees. The programming of the device is done by its own software via ethernet cable. The device comes with M12 4-pole connector cable (which allows the programming), M12 8-pole connector cable (which includes power supply and inputs and outputs) and mounting brackets.

The reason to use the area scanner instead of light curtain at entrance of the robot cells is the customer desire to pause and restart the lines automatically after operator interference (for instance, a pallet changing during the process) and continuous scanning of the cells during the process in order to obtain a safer working environment.

It allows area mapping up to six configurations by providing three input cables to the manipulator controller (available mappings can be observed from Figure-18). When an object is detected in the scanned area, output signals from the scanner to controller to trigger ’StopMove’ command. The restart of the scanner can be done in two different ways: automatic or manual restart. The automatic start by triggering the ’StartMove’ command is used in the robot cells.

Four area mapping configurations are used in each robot cell. First configuration includes no pallet existence in the station. The scanner scans the pallet slots in addition to the area between the entrance of the cell and the manipulator. Second configuration includes only left pallet existence in the station. It means the scanner scans the right pallet slot beside the common area. Third configuration includes only right pallet existence in the station. The scanned area is opposite of the second configuration. Fourth and final configuration includes both pallet existence in the station. The scanner only scans the common area.

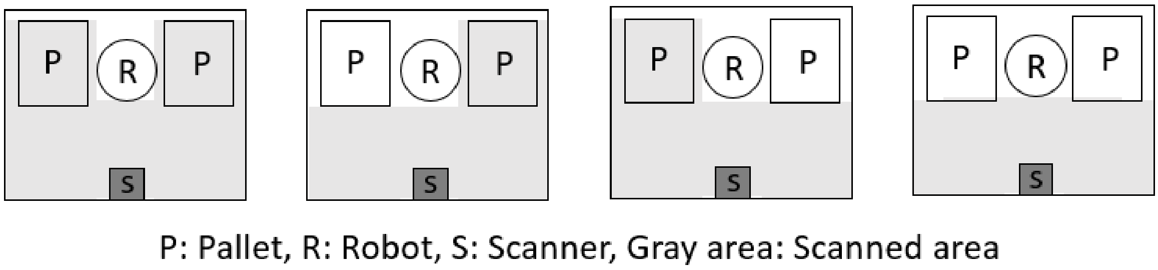

Figure 41. Different configurations of the area scanner

The area scanners safety relay outputs are directly connected to the local IO instead of safety inputs. It is initial target is to pause and restart the controller automatically if an interference occurs. Since the logic of the automatic pause & restart could not be managed to do in safety system, the logic is done in the robot tasks. It causes slightly slower response. The customer had informed about the situation and, it was agreed to use it that way not to stop the production. Additional changes shall be done in the future.

7.2 Safety Light Beam

Safety light beams are generally used for human detection as a guard of an entrance to a hazardous area. It is composed of a transmitter and a receiver and it must be connected to a safety controller or a safety PLC. ABB Spot 10 is used with mounting brackets and mirrors. The light beam has 10 m range and required distance in the station is 4.55 m.

Need of the light beam arises from the hazardous area between the pallets and conveyor since the area scanner is not able to scan this area. After an interference to the station, if operator goes behind the pallet slots, the area scanner shall start the manipulator action and operator shall be in the high-risk area. In order to prevent the hazardous situation, light beam is placed right behind the pallet slots. If someone wants to pass this area, light beam has to be disturbed first.

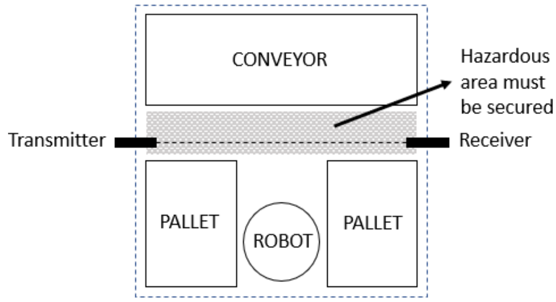

Figure 42. Light beam placement

After interference of the light beam, the controller goes into motor-off state. In order to restore the robot state, light beam must be activated manually. Light beam activation is done by pressing the reset button outside of the station. A 'Smile 11 EAR' reset button is placed right side of the entrance for this purpose. After resetting the light beam, the operator must press the motor-on button on the controller and start process from flex pendant.

In order not to disturb the light beam cables and maintain a stable position for the beams, light beams are attached to the fences vertically. To maintain the light contact between the beams, two mirrors are mounted with 45 degrees on top of the transmitter and the receiver as it is shown on Figure 12.

The autonomous start is not used in this area because light beams are not working as area scanners. Once the safety chain is broken, it must be reset by the operator in order to confirm that the area is clear for the process.

7.3 Fences

Fence is a safeguard equipment that prevents humans entering into a hazardous work area by enclosing the risk area. It is a low-cost preventive measure to enhance the safety. It consists of aluminium profiles, mesh, net locks or polycarbonate panels. Height and distance of fence structure is chosen according to ISO 13857 [23].

To define height of the fencing, maximum height that robot may reach during the process must be calculated. In both stations, the maximum height can be calculated according to the maximum pallet height. Manipulator raises its maximum height when it places last layer of a pallet. The fence height is selected as 2000 mm considering that there can be maximum of 40 layers on a pallet, product thickness can go up to 32 mm, euro pallet height is 144 mm, gripper height is 230 mm, and layer placing approach offset is 50 mm:

$$(32 \times 40) + 144 + 230 + 50 \cong 1700\ mm$$

Minimum distance from robot working area to station entrance depends on few factors. Based on safety requirement EN ISO 13855, the typical calculation for the minimum safety distance is given by the equation below [24]:

$$S = (K \times T) + C \qquad 7.2$$

Where, S = Minimum safety distance (mm), K = Approach speed parameter (mm/s), T = Total response time (s), C = Total additional distance (mm).

K value is 2000 mm/s if the calculated S value is ≤ 500mm. K value is 1600 mm/s if the calculated S value is > 500 mm. T value is the total response time of the manipulator (t1) and safety scanner (t2). C value is calculated as 8x(d-14) mm if the detection capability (d) is ≤ 40 mm. C value is 850 mm for devices with detection capability > 40 mm (Detection capability of the safety scanner in the stations is 70 mm). Minimum safety distance is calculated considering K = 1600 mm/s, t1 = 0.86 s for IRB660 and 0.45 s for IRB460, t2 = 0.062 s, and C = 850 mm.

$$[1600x(0.86 + 0.062)] + 850 \cong 2325\ mm\ for\ IRB660$$

$$[1600x(0.45 + 0.062)] + 850 \cong 1700\ mm\ for\ IRB460$$

7.4 SafeMove

SafeMove let's manipulators and operators work together more closely, efficiently, and safely. It is a safety controller placed in the IRC5 controller. It provides safe zones, speed limits, safe standstill positions etc. It reduces the need for conventional safety measures such as fences, light curtains, safety relays etc. SafeMove provides more compact cells by restraining manipulator action precisely what is required for an explicit application.

In addition to the area scanners, robot stations are equipped with the SafeMove Basic option. It allows to define a work area for the robot. If robot leaves the defined area for any reason, controller shall stop, and an error will appear on Flex Pendant screen. To restore the controller, it must be switched to manual mode and jogged to its safe area manually.

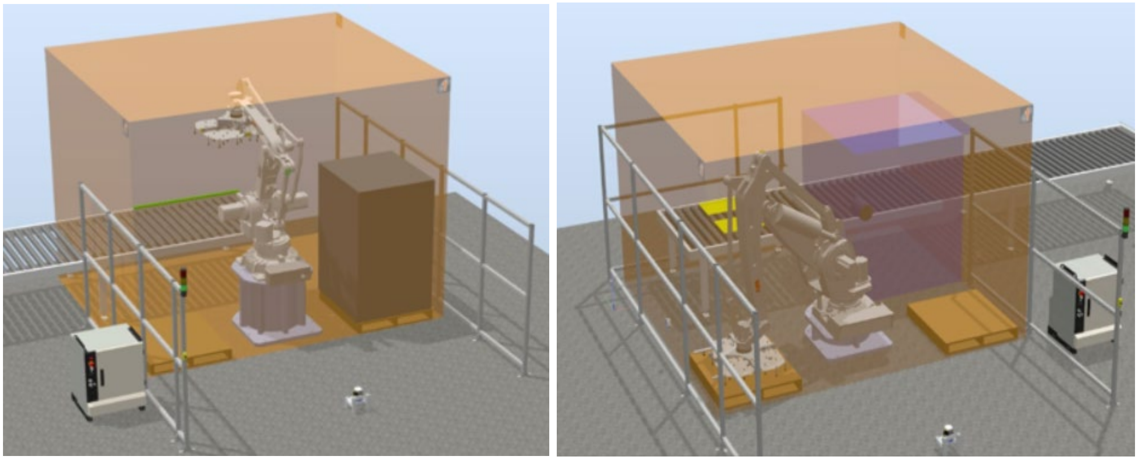

Figure 43. SafeMove configuration of IRB460 and IRB660 respectively

Both areas start with the furthest pallet edges, and end with the furthest conveyor edges. The enclosure dimension is 3200 mm on X axis, 4200 mm on Y axis and 3300 mm on Z axis for IRB460. The enclosure dimension is 4450 mm on X axis, 4450 mm on Y axis and 3300 mm on Z axis for IRB660.

A combination of methods such as fences, spot beams, SafeMove, and area scanners are used to achieve a higher and an effective safety level. All ABB robots and controllers fulfil EN ISO 10218-1:2006, safety requirements for robots, and comply with category 3 as described in EN 954-1. In addition, and for additional safety, SafeMove fulfil performance level and category 3 of the new EN ISO 13849-1.

7.5 Signal tower lights

Signal tower lights are generally used in industrial cells to provide visual and audible indication of the instrument's status to operators. It helps operators to notice system errors as the error occurs. The stack on the tower can go up to five differently coloured segments according to the process requirements.

ABB KS70 model signal tower is used in both stations. Signal tower in stations consists of three light elements and a buzzer. First light element is a green LED, second element is a yellow LED, and third element is a red LED. Mainly, red element is used to state the danger situation for persons, emergency situation of process, and faulty state of instrument. Yellow element is used to state warning/caution situation for persons, abnormal condition of process, and abnormal condition of instruments. Green element is used to state safe situation for persons, and normal state for process condition and instruments. Buzzer is used to increase awareness of the current status. It is activated when manipulator is about to start working, an unexpected situation occurs in one of the stations like a process anomaly, and to notify the operator when a process is done.

7.6 Sense7 safety switch

ABB Sense7 non-contact coded safety switch is used to maintain position interlock for the moving guards. It comes with two parts, one part is placed to stationary part of the guard, and second part is placed to moving part of the guard. It has 2NC and 1NO contacts and a sensing distance to the actuator of 14 mm.

A fence door is required by customer to reach behind the conveyor area in palletizing station. In order to provide sufficient safety measure, Sense7 switch is used in IRB460 palletizing station. It is connected to a safety relay. Whenever the door is open, the controller halts the process immediately with the help of the switch. After restoring the switch position, a manual motor-on and start is required since the area behind the conveyor is not scanned by the area scanner due to the blockage of robot, conveyor, and pallets.

8. SUMMARY

Harviker OÜ asked for a quotation for two automated robotic cells for a pick and place application of HDF wood boards. While the first cell depalletizes HDF boards and places them onto a conveyor at the beginning of the line, the second cell picks the processed wood boards from the conveyor at the end of the line and places them to the pallets at each side of the robot. Each robot can palletize/depalletize one pallet at a time. After a layer of HDF boards is depalletized, boards go into gluing, honeycomb structure assembly, and glue drying process. When the drying process is done, finished products are transferred to the final conveyor for palletization. Only one product type can be produced in a cycle.

There are nine different product types in length and width wise. Furthermore, there are two different thickness values. Considering each product type would have both thin and thick variations, there are 18 product types in total.

The tasks and the system layout plan for the cells were given by the customer. The tasks include a custom-designed gripper for pick and place application, pallet height search functionality to understand the initial picking position, product thickness detection to determine the offset value for updating picking position after each picking process, automatic pause and restart of the cells in case of an interference to the cells, pallet detection by sensors to understand existence of the pallets, detection of products on the gripper to check number of the picked products, HMI for controllers to possess control over the process, and safety measures to provide standard safety.

The contributed work can be summarised as; pedestal design of the manipulators, assembly and modifications of the grippers (including optical & capacitive sensor mounting and range adjustment of the sensors, pneumatic equipment mounting and air connections), trajectory programming and coding of the robots, custom error log structure design, HMI design for both controllers, tests to prevent multi-layer picked boards (and separation of layers, in case multiple are picked), signal tower connection and programming, pneumatic diagrams of the grippers, and process test & control.

Robot pedestals were needed to provide enough reachability for the manipulators. The pedestals' main purpose is to elevate the manipulators to the necessary height for maintaining pick & place applications without any reach restrictions. SolidWorks software was used for designing. Manipulator manuals were used to determine robot base connections and dimensions. The height of the pedestals was determined according to conveyors' heights and system design given by the customer.

Gripper modifications were made during process tests. Suction cups with different diameter values were installed and tested during layer separation tests to optimize the picking and the layer separation performance. Capacitive sensors were installed, adjusted, and tested to detect extra layers attached to gripper after the picking process. Optical sensors were installed and tested to detect the boards' absence/existence after the picking process. Furthermore, one of the optical sensors is used to search the height of the pallets. Additional solenoid valves were installed and tested to provide air to the air blowing mechanism, and to switch the air pressure for picking optimization.

IRB460 trajectory programming was made to teach the pick & place targets to the manipulator. IRB660 programming structure and trajectory programming were made to run the system efficiently and decently. Programming structure includes basic concepts like understanding product type, gripper picking configurations based on the product type information, height search of the pallets, extra layer separation, air pressure adjustment based on product thickness, picking and placing processes, etc.

Custom error log structures were created for operators to respond to the errors in a proper way. Understanding the reasons for the errors is important for reacting to the failures promptly. Error logs include possible error scenarios and help the operator to understand what to do by giving instructions. Each custom log has its own unique code to differentiate itself from others. Details about error logs are given in the user manual.

The human-machine interface was designed to monitor the process variables, and to possess control over the sensors and pneumatic equipment. It was designed in RobotStudio's HMI add-in. The operator can see details about products, pallets, and processes on HMI. The user can manipulate some pneumatic equipment on grippers and some system variables to mitigate the errors and to make quick changes during a process.

Some challenges had been encountered during the test stage of the cells. The most critical challenge was not picking several layers of 3 mm thick boards during the depalletizing process. Multi-layer board picking had been experienced because of the porous structure of the HDF boards. To solve the issue, several different methods were tested. In the end, adjusting air pressure according to product thickness before the picking process, and blowing air through the picked multi-layer boards solved the issue. While air pressure adjustment helped decreasing the number of additionally picked boards by applying less vacuum force, blowing air through the picked multi-layer boards helped separating and dropping the extra boards back to the pallet. Additional mechanisms were designed and attached to the gripper for air blowing, and for maintaining product alignment and keeping position of the separated layers on pallets after the layer separation. Multi-layer picking became a problem for thickness detection of the products. The thickness detection was initially intended to be done at the beginning of the depalletizing process. Since the controller does not know the thickness value, it could not adjust the picking air pressure according to product thickness. The situation was explained to the customer, and it was decided to insert the thickness value manually by operators.

Signal towers were installed to increase awareness of the operators about system status. Green, yellow and red LEDs on the signal tower indicate different statuses. Green LED indicates a running system without errors. Yellow LED indicates an abnormal situation, or a system pause. Red LED indicates an error in the system. Programming of signal towers was made by using system outputs and custom variables of the controllers.

Pneumatic diagrams were designed by using Festo's FluidDraw software. The diagrams help to identify the pneumatic equipment in the system. If an error or malfunction occurs, a quick replacement or mend can be performed by tracing the diagrams.

The system works as it was intended. The customer is satisfied with the solution. Robot cells are fast enough to feed/empty the gluing & heating systems. Layer separation was successfully applied for the depalletizing process. Area scanners can detect operator interference to the robot cells during a cycle. Automatic pause and restart of the system save time for the production by eliminating manual restart of the cells.

The development will continue with the safety system. The area scanners safety relay outputs are connected to the local IO instead of safety inputs to provide automatic pause and restart for the system. The robots are stopped by the robot tasks instead of the safety system when an interference occurs. It causes slightly slower response. The customer was informed regarding the situation, and further development shall be provided to improve the safety measures. If new products are introduced to the system, the product range shall be extended with gripper and program modifications.

9. SUMMARY IN ESTONIAN

Harviker OÜ küsis pakkumist kahele automatiseeritud robotüksusele HDF puitplaatide ladustamiseks. Kui esimeses robotüksuses haaratakse HDF plaadid aluselt ning asetatakse need konveierile tooteliini alguses, siis teises robotüksuses haaratakse töödeldud puitplaadid konveierilt liini lõpust ja asetatakse need alustele mõlemal pool robotit. Iga robot suudab töötada ühe alusega korraga. Kui üks kiht HDF plaate on aluselt eemaldatud, siis plaadid liimitakse, koostatakse meekärjestruktuur ja liim kuivatatakse. Kui liim on kuivanud, liiguvad valmistooted lõppkonveierile alusele ladustamiseks. Ühes tsüklis on võimalik toota vaid ühte tootetüüpi.

Pikkuse ja laiuse järgi on erinevaid tootetüüpe üheksa. Lisaks on tooteid kahes erinevas paksuses. Arvestades, et igat tootetüüpi valmistatakse kahes paksuses, on tootetüüpe kokku 18.

Lähteülesanded ja asetusplaani robotsüsteemide jaoks andis klient. Lähteülesanded sisaldavad spetsiaalset haaratsid plaatide ladustamiseks, aluse kõrguse leidmise funktsiooni, toote paksuse tuvastamist korjamispositsioonide uuendamiseks peale igat korjamisprotsessi, robotüksuste automaatset peatumist ja taaskäivitumist tõrke korral, sensoorikat aluste olemasolu tuvastamiseks, haaratsi poolt korjatud toodete arvu tuvastamist, kasutajaliidest süsteemi juhtimiseks ja turvameetmeid nõuetekohase turvalisuse tagamiseks.

Autori panus selle süsteemi arendusse võib kokku võtta järgnevalt: pjedestaalide planeerimine robotmanipulaatoritele, haaratsite koostamine ja modifitseerimine (k.a. optilise sensori ja mahtuvusliku sensori kinnitamine, häälestamine ning pneumokomponentide kinnitamine ja ühendused suruõhuga), robotite programmeerimine ning tööliigutuste testimine, spetsiaalse vearegistri struktuuri kavandamine, kasutajaliidese kujundamine mõlema kontrolleri jaoks, testimine mitme plaadi korraga haaramise välistamiseks (nende eraldamine, juhul kui mitu on korraga haaratud), valgusfooride ühendamine ja programmeerimine, haaratsite pneumoskeemide loomine, ning protsessi testimine ja seadistamine.

Pjedestaalid olid vajalikud, et tagada robotite piisav haardeulatus. Pjedestaalide peamine eesmärk on tõsta robotmanipulaatorid piisavale kõrgusele, et ladustamisprotsess saaks toimuda ilma takistusteta. Pjedestaalide loomiseks kasutati SolidWorks tarkvara. Robotite kinnituste ja dimensioonide määramiseks kasutati robotite tootemanuaale. Pjedestaalide kõrguse määrasid ära konveierite kõrgus ning kliendi spetsifikatsioon.

Muudatused haaratsile tehti süsteemi testimise käigus. Plaatide eraldamise testimisel paigaldati ja testiti erineva läbimõõduga vaakumiminappasid, et optimeerida haaramise ja eraldamise töökindlus. Et tuvastada, kas haaratsi küljes on peale haaramisprotsessi plaate rohkem kui üks, paigaldati haaratsi külge mahtuvuslikud sensorid. Et tuvastada plaatide olemasolu või puudumist haaratsi küljes peale haaramisprotsessi, paigaldati ning testiti optilised andurid. Lisaks kasutatakse ühte optilist andurit aluse kõrguse leidmiseks. Et tagada suruõhk puhumismehhanismi jaoks ja õhurõhu reguleerimiseks haaramise optimeerimiseks, paigaldati ja testiti lisa pneumojaotid.

IRB460 tööliigutuste programmeerimine viidi läbi, et õpetada robotmanipulaatorile haaramise ja asetamise positsioonid. IRB660 koodistruktuuri ja trajektooride programmeerimisel peeti silmas süsteemi efektiivsust ja korralikku tööd. Koodistruktuur sisaldab põhilisi funktsioone nagu tootetüübi mõistmine, tootetüübi järgi haaratsi seadistamine, aluste kõrguse määramine, üleliigse kihi eraldamine, õhurõhu reguleerimine toote paksuse põhjal, ladustamisprotsessid jne.

Et operaatoritel oleks võimalik vigade korral korrektselt käituda, loodi spetsiaalsed vearegistrid. Vigadele kiirel reageerimisel on oluline mõista vigade põhjuseid. Vearegistrid sisaldavad võimalikke veastsenaariumeid ning juhendeid operaatorile, kuidas käituda. Igal vearegistril on eristumiseks unikaalne kood. Vearegistrid on detailselt lahti kirjeldatud kasutajajuhendis.

Kasutajaliides loodi, et jälgida protsessimuutujaid ja juhtida andureid ning pneumaatikat. Kasutajaliides loodi RobotStudio tarkvaras. Kasutajaliides võimaldab operaatoril näha detaile toodete, aluste ning protsesside kohta. Kasutaja saab juhtida teatud pneumaatikakomponente haaratsi küljes ning süsteemimuutujaid, et vältida vigu ning teha kiireid muudatusi süsteemi töös.

Robotüksuste testimisfaasis tuli ette erinevaid katsumusi. Kõige kriitilisem väljakutse oli olukorra vältimine, kus robot korjas aluselt plaadi haaramise käigus mitu 3 mm paksust plaati. Mitme plaadi korraga korjamise põhjustas HDF plaatide poorne struktuur. Et probleemi lahendada, katsetati erinevaid meetodeid. Lõpuks osutus lahenduseks õhurõhu reguleerimine olenevalt toote paksusest ning suruõhu puhumine läbi korjatud plaatide. Kui õhurõhu reguleerimine aitas vähendada korjatud plaatide arvu, vähendades vaakumi tugevust, siis suruõhu puhumine läbi korjatud plaatide aitas neid haaratsist eraldada ning tagasi alusele poetada. Haaratsile loodi lisamehhanismid õhu puhumiseks ning haaratud ja alusel olevate toodete positsioonide säilitamiseks. Mitme plaadi korjamine häiris ka toote paksuse tuvastamist. Esmalt oli paksuse tuvastamine planeeritud aluselt korjamise protsessi alguses. Kuna kontroller toote paksust ei tea, ei olnud see võimeline reguleerima õhurõhku toote paksusest lähtuvalt. See olukord seletati kliendile ning koos otsustati, et paksuse väärtus sisestatakse käsitsi kasutajaliidese kaudu.

Et hoida operaatoreid süsteemi olekust teadlikuna, paigaldati valgusfoorid, kus rohelised, kollased ja punased leedlambid erinevaid olekuid indikeerivad. Roheline tuli osutab sellele, et süsteem töötab ilma vigadeta. Kollane tuli osutab kõrvalekaldele normaalsest tööst või pausile. Punane tuli osutab veale. Valgusfooride programmeerimisel kasutati süsteemiväljundeid ning spetsiaalseid muutujaid kontrollerites.

Pneumoskeemide loomisel kasutati Festo tarkvara FluidDraw. Skeemid aitavad määrata süsteemis kasutatud pneumaatilisi komponente. Vea või rikke eemaldamiseks on komponendid võimalik skeemide abil kiiresti üles leida.

Süsteem toimib, nagu oli kavatsetud. Klient on lahendusega rahul. Robotüksused töötavad piisava kiirusega, et täita ja tühjendada liimimis- ja kuumutusseadmeid. Edukalt lisati haaramisprotsessile kihtide eralduse funktsioon. Alasensorid suudavad tuvastada süsteemi töö häirimise operaatori poolt. Automaatne seiskumine ja taaskäivitumine säästavad tootmises aega manuaalse käivitamise arvelt.

Järgneb veel süsteemi turvalisuse arendamine. Alasensorid ei ole hetkel ühendatud turvasüsteemiga, et tagada süsteemi automaatne seiskumine ja taaskäivitumine. Häire korral seiskab robotid programm, mitte turvaahela katkemine. Olukorrast informeeriti klienti ning turvameetmete arendamine jätkub. Et lisada uusi tootetüüpe, on võimalik süsteemi poolt käideldavat tootevalikut suurendada muudatustega programmile ja haaratsile.

10. REFERENCES

1. https://www.schmalz.com/en/vacuum-technology-for-automation/vacuum-gripping-systems/
2. https://onrobot.com/en/products/vg10-electric-vacuum-gripper
3. https://www.destaco.com/food-processing-packaging.html
4. T. Takahashi, K. Nagato, M. Suzuki, S. Aoyagi Flexible Vacuum Gripper with Autonomous Switchable Valves 2013 IEEE International Conference on Robotics and Automation (ICRA) Karlsruhe, Germany, May 6-10, 2013
5. Huanle Chen, Jietao Chen, Jian Luo, Hao Zhang, Yisheng Guan, Haifei Zhu, Development of A Novel Vacuum-Suction Gripper Capable of Changing Gripping Position International Conference on Robotics and Biomimetics December 12-15, 2018, Kuala Lumpur, Malaysia
6. https://grabitinc.com/products/
7. Jean-Philippe Roberge, Wilson Ruotolo, Vincent Duchaine, Mark Cutkosky Improving Industrial Grippers with Adhesion-Controlled Friction, IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 3, NO. 2, APRIL 2018
8. Mohammad Dadkhah, Zhanyue Zhao, Nicholas Wettels, Matthew Spenko, A Self-Aligning Gripper Using an Electrostatic/Gecko-Like Adhesive, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Daejeon Convention Center October 9-14, 2016, Daejeon, Korea
9. Xin Li, Ning Li, Guoliang Tao, Hao Liu, Toshiharu Kagawa, Experimental Comparison of Bernoulli Gripper and Vortex Gripper, INTERNATIONAL JOURNAL OF PRECISION ENGINEERING AND MANUFACTURING Vol. 16, No. 10, pp. 2081-2090 SEPTEMBER 2015
10. Subir Das, Tuhin Subhra Sarkar, Badal Chakraborty and Himadri Sekhar Dutta, Study on Wooden Block Thickness Measurement using an Opto-Imaging Sensor, Proc. of Int. Conf. on Computing, Communication & Manufacturing 2014
11. https://www.sick.com/ag/en/industries/wood/sawmill-industry/stripping-veneer/measurement-of-veneer-thickness/c/p518981
12. https://cdn.sick.com/media/docs/2/82/182/Special_information_Precision_meets_Quality_Our_solution_for_measurement_sensors_en_IM0064182.PDF
13. https://www.schmalz.com/en/applications/application-reports/details/vacuum-suction-spider-ssp-facilitates-handling-of-veneer/
14. Trang Hieu Luu, Thanh Hung Tran, 3D Vision for Mobile robot Manipulator on Detecting and Tracking Target, 2015 15th International Conference on Control, Automation and Systems (ICCAS 2015) Oct. 13-16, 2015 in BEXCO, Busan, Korea

15. Milos Vasic, Aude Billard, Safety Issues in Human-Robot Interactions, 2013 IEEE International Conference on Robotics and Automation (ICRA) Karlsruhe, Germany
16. Iñaki Maurtua, Aitor Ibarguren, Johan Kildal, Loreto Susperregi, Basilio Sierra, Human-robot collaboration in industrial applications: Safety, interaction and trust, July 2017 International Journal of Advanced Robotic Systems
17. Emanuele Magrini, Federica Ferraguti, Andrea Jacopo Ronga, Fabio Pini, Alessandro De Luca, Francesco Leali, Human-robot coexistence and interaction in open industrial cells, Robotics and Computer Integrated Manufacturing 2020
18. Om Prakash Sahu, Bibhuti Bhusan Biswal, Saptarshi Mukherjee, Panchanand Jha, Multiple Sensor Integrated Robotic End-effectors for Assembly, 2nd International Conference on Innovations in Automation and Mechatronics Engineering, ICIAME 2014
19. https://www.sick.com/us/en/ud18-2-ultrasonic-sensor-for-double-layer-and-splice-detection/w/press-ud18-2/
20. Dirk Holz, Angeliki Topalidou-Kyniazopoulou, Francesco Rovida, Mikkel Rath Pedersen, Volker Kruger, Sven Behnke, A Skill-Based System for Object Perception and Manipulation for Automating Kitting Tasks
21. Krishnanand N. Kaipa, Carlos W. Morato, Satyandra K. Gupta, Design of Hybrid Cells to Facilitate Safe and Efficient Human–Robot Collaboration During Assembly Operations
22. Technical reference manual, System parameters, RobotWare 6.11, ABB
23. ISO 13857-2019 Safety of machinery – Safety distances to prevent hazard zones being reached by upper and lower limbs
24. ISO 13855-2010 Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body

APPENDIX A

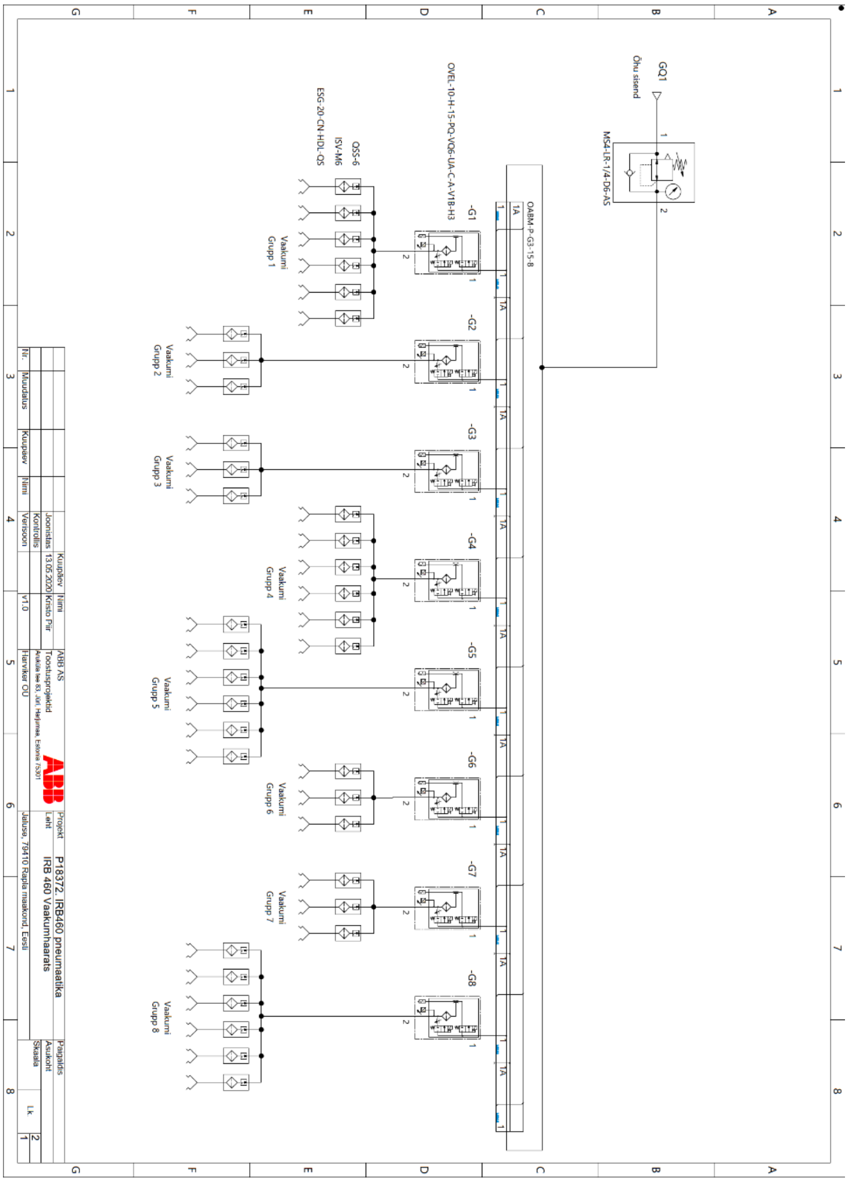

Figure 44. Pneumatic scheme of IRB460 gripper

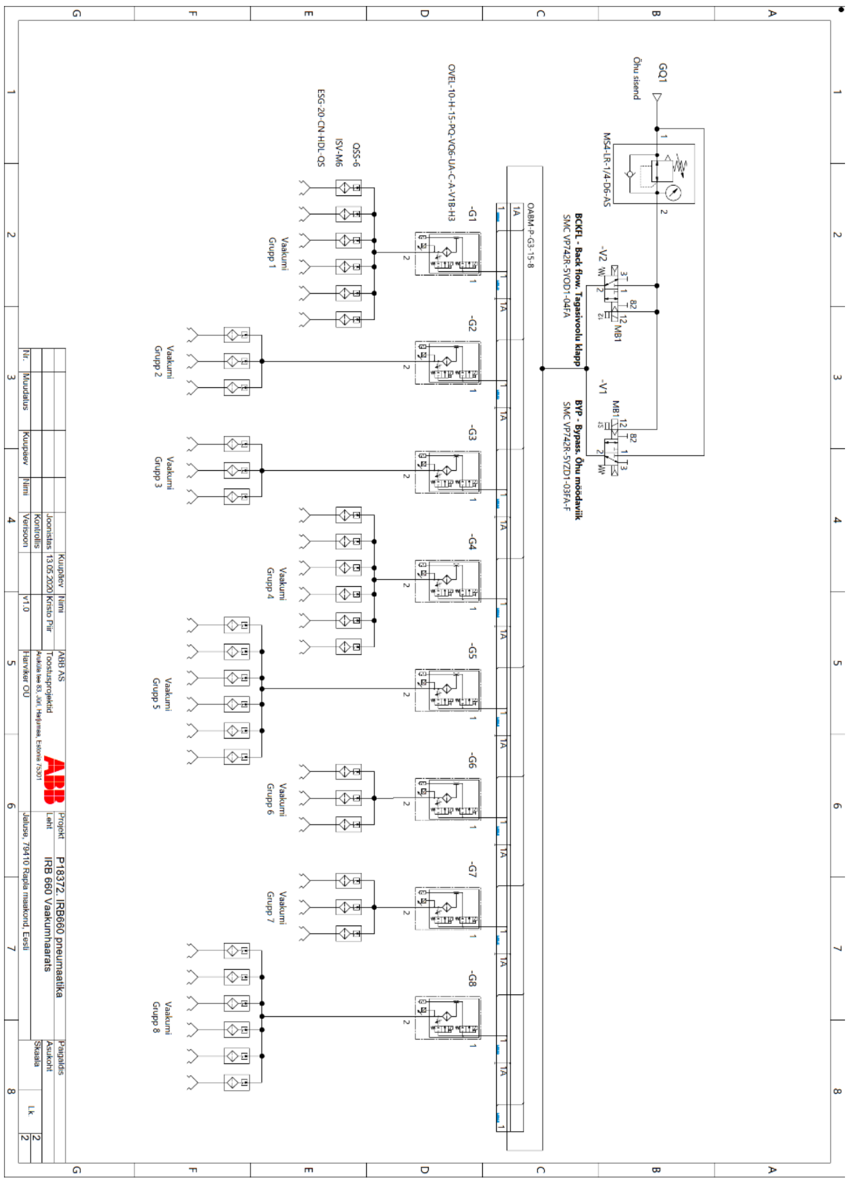

Figure 45. Pneumatic scheme of IRB660 1/2

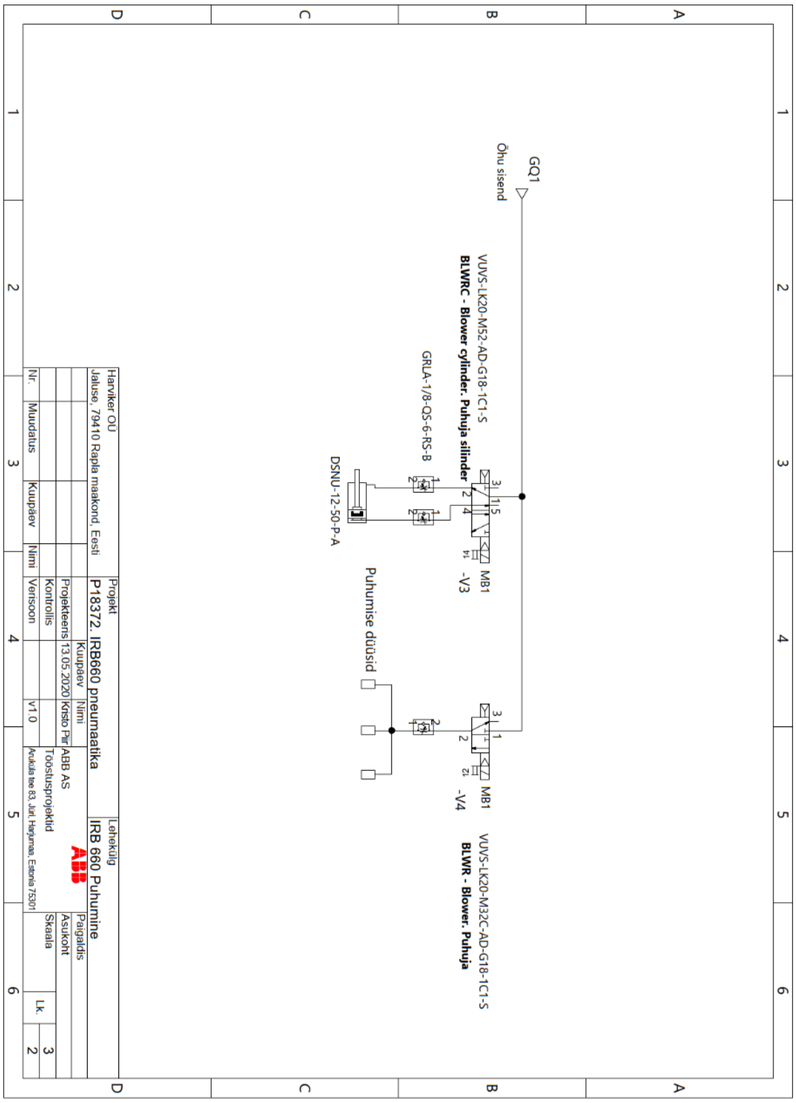

Figure 46. Pneumatic scheme of IRB660 2/2

APPENDIX B

```
MODULE MainModule
  RECORD gripperData
    num nPickedPanels;
  ENDRECORD

  RECORD productData
    num nProdType;
    num nProdThickness;
    string sProdDimensions;
    num nGripSections;
  ENDRECORD

  PERS num nPallet1Status;
  PERS num nPallet2Status;
  PERS num nPalSensStat1;
  PERS num nPalSensStat2;
  PERS bool bResetPal1:=FALSE;
  PERS bool bResetPal2:=FALSE;
  !Statuses:
  !0 - Palletizing area is empty. New pallet can be placed to the depalletizing area.
  !1 - New pallet in pos. Need to measure edges and height.
  !2 - pallet has been measured and destacking is in progress.
  !3 - Pallet is empty and need to be removed from depalletizing area.
  !-1 - Error. Can't find pallet at height of the 1800mm.
  !-2 - Error. Can't find pallet BELOW the height of 1800mm.
  !-3 - Error. Pallet 1 presence sensor fault!
  !Sensors can get dirty/dusty over time and they can give faulty readings.
  !Here is, how we try to discover that happening.
  VAR intnum inPal1Sens;
  VAR intnum inPal2Sens;
  PERS num nPalletizedLayersPal1;
  PERS num nPalletizedLayersPal2;
  PERS num n3mmOr32mm;
  VAR num nPrev3mmOr32mm:=-1;
  VAR clock cPallet1Clock;
  VAR clock cPallet2Clock;
  VAR num nPallet1ClockValue;
  VAR num nPallet2ClockValue;
  VAR num nPallet1ClockStatus;
  VAR num nPallet2ClockStatus;
  !Both sensors B1.1 and B1.2 OR B2.1 and B2.2 must have correct state in that time (seconds):
  CONST num nPalletSensorTimeLimit:=25;
  !Palletizing constants
  CONST num nNoPallet:=0;
  CONST num nPalletPresent:=1;
  CONST num nPalletizingInProgress:=2;
  CONST num nPalletizingFinished:=3;
  CONST num nERRPalletSensError:=-3;
  CONST num nPalletHeightHigh:=45;
  PERS num nScannerZonesSwDelay:=0.5;
  CONST num nScannerOnlyLeft:=010;
  CONST num nScannerOnlyRight:=100;
  CONST num nScannerBothPallets:=011;
  CONST num nScannerNoPallets:=101;
  VAR num nScannerFromArea;
  VAR num nScannerToArea;
  PERS productData ProdData;
  PERS num nPickedPanels;
  PERS num nAreaOccupied;
  PERS num nVacGuard:=1;
  PERS num nDoERRBeeping;
  PERS num nEnFallenPartsRegrip;
  PERS num nRegrpiDelay:=1.5;
  PERS bool bPal1StatusOK;
  PERS bool bPal2StatusOK;
  !Error messages
  VAR num errid_PalletSensor:=4803;
  VAR errstr ERR_PalletSensor:="ERR_PalletSensor!";
  VAR errstr ERR_msg1_PalletSensor:="Left pallet sensors (B1.1 and/or B1.2) fault!";
  VAR errstr ERR_msg2_PalletSensor:="1) Pallet may be not in position correctly!";
  VAR errstr ERR_msg3_PalletSensor:="2) Clean the sensor in case it covered by dust.";
  VAR errstr ERR_msg4_PalletSensor:="3) Check user manual!";
  VAR num errid_Pallet2Sensor:=4804;
  VAR errstr ERR_Pallet2Sensor:="ERR_PalletSensor!";
  VAR errstr ERR_msg1_Pallet2Sensor:="Right pallet sensors (B2.1 and/or B2.2) fault!";
  VAR errstr ERR_msg2_Pallet2Sensor:="1) Pallet may be not in position correctly!";
  VAR errstr ERR_msg3_Pallet2Sensor:="2) Clean the sensor in case it covered by dust.";
  VAR errstr ERR_msg4_Pallet2Sensor:="3) Check user manual!";
```

```
PROC main()
  !Initialize interrupts.
  !Powering up the laser sentinel scanner.
  !We need to use DO and relay combination
  !and activate the relay only, when IRC5 has booted up.
  SetDO Local_DO8_LaserSentinelPower,1;
  WHILE TRUE DO
    WaitTime 0.2;
    pSelectThickness;
    pPalletsSensorLogic;
    pIsAnyErrorActive;
    pWarningLights;
    pScannerControl;
    pPickedPanelsToZero;
    pVacControlDroppedProds;
    pERRBeeping;
    pGroupInpToSignalDI;
  ENDWHILE
ENDPROC

PROC pGroupInpToSignalDI()
  !It was not possible to do it in cross connection
  IF ginput(gripperGroupCS) = 63 THEN
    SetDO UserSys8_DO_CS_ReadingOK,1;
  ELSE
    SetDO UserSys8_DO_CS_ReadingOK,0;
  ENDIF
ENDPROC

PROC pIsAnyErrorActive()
  IF nPallet1Status=nERRPalletSensError THEN
      bPal1StatusOK:=false;
    ELSE
      bPal1StatusOK:=true;
  ENDIF
  IF nPallet2Status=nERRPalletSensError THEN
      bPal2StatusOK:=false;
    ELSE
      bPal2StatusOK:=true;
  ENDIF
ENDPROC

PROC pVacControlDroppedProds()
  IF nVacGuard=1 THEN
    IF Presse_GI00_ProductType=1 THEN
      IF Gripper_DI01_OS1=0 SetDO Gripper_DO1_Vac1,0;
      IF Gripper_DI02_OS2=0 SetDO Gripper_DO2_Vac2,0;
      IF Gripper_DI02_OS2=0 SetDO Gripper_DO3_Vac3,0;
      IF Gripper_DI03_OS3=0 SetDO Gripper_DO4_Vac4,0;
      IF Gripper_DI04_OS4=0 SetDO Gripper_DO5_Vac5,0;
      IF Gripper_DI05_OS5=0 SetDO Gripper_DO6_Vac6,0;
      IF Gripper_DI05_OS5=0 SetDO Gripper_DO7_Vac7,0;
      IF Gripper_DI06_OS6=0 SetDO Gripper_DO8_Vac8,0;
    ENDIF
    IF Presse_GI00_ProductType=2 OR Presse_GI00_ProductType=4 OR Presse_GI00_ProductType=5 OR Presse_GI00_ProductType=7 OR
Presse_GI00_ProductType=8 THEN
      IF Gripper_DI01_OS1=0 SetDO Gripper_DO1_Vac1,0;
      IF Gripper_DI01_OS1=0 SetDO Gripper_DO2_Vac2,0;
      IF Gripper_DI02_OS2=0 SetDO Gripper_DO3_Vac3,0;
      IF Gripper_DI03_OS3=0 SetDO Gripper_DO4_Vac4,0;
      IF Gripper_DI04_OS4=0 SetDO Gripper_DO5_Vac5,0;
      IF Gripper_DI05_OS5=0 SetDO Gripper_DO6_Vac6,0;
      IF Gripper_DI06_OS6=0 SetDO Gripper_DO7_Vac7,0;
      IF Gripper_DI06_OS6=0 SetDO Gripper_DO8_Vac8,0;
    ENDIF
    IF Presse_GI00_ProductType=3 OR Presse_GI00_ProductType=6 OR Presse_GI00_ProductType=9 THEN
      IF Gripper_DI01_OS1=0 SetDO Gripper_DO1_Vac1,0;
      IF Gripper_DI02_OS2=0 SetDO Gripper_DO2_Vac2,0;
      IF Gripper_DI02_OS2=0 SetDO Gripper_DO3_Vac3,0;
      IF Gripper_DI03_OS3=0 SetDO Gripper_DO4_Vac4,0;
      IF Gripper_DI04_OS4=0 SetDO Gripper_DO5_Vac5,0;
      IF Gripper_DI05_OS5=0 SetDO Gripper_DO6_Vac6,0;
      IF Gripper_DI05_OS5=0 SetDO Gripper_DO7_Vac7,0;
      IF Gripper_DI06_OS6=0 SetDO Gripper_DO8_Vac8,0;
    ENDIF
  ENDIF
  IF nEnFallenPartsRegrip = 1 THEN
    IF Presse_GI00_ProductType=1 THEN
      IF Gripper_DI01_OS1=1 AND Gripper_DO1_Vac1=0 THEN
        waittime nRegrpiDelay;
        SetDO Gripper_DO1_Vac1,1;
      ENDIF

      IF Gripper_DI02_OS2=1 AND Gripper_DO2_Vac2=0 AND Gripper_DO3_Vac3=0 THEN
        waittime nRegrpiDelay;
        SetDO Gripper_DO2_Vac2, 1;
```

```
            SetDO Gripper_DO3_Vac3, 1;
          ENDIF
          IF Gripper_DI03_OS3=1 AND Gripper_DO4_Vac4=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO4_Vac4,1;
          ENDIF
          IF Gripper_DI04_OS4=1 AND Gripper_DO5_Vac5=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO5_Vac5,1;
          ENDIF
          IF Gripper_DI05_OS5=1 AND Gripper_DO6_Vac6=0 AND Gripper_DO7_Vac7=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO6_Vac6, 1;
            SetDO Gripper_DO7_Vac7, 1;
          ENDIF
          IF Gripper_DI06_OS6=1 AND Gripper_DO8_Vac8=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO8_Vac8,1;
          ENDIF
        ENDIF
        IF Presse_GI00_ProductType=2 OR Presse_GI00_ProductType=4 OR Presse_GI00_ProductType=5 OR Presse_GI00_ProductType=7 OR
Presse_GI00_ProductType=8 THEN
          IF Gripper_DI01_OS1=1 AND Gripper_DO1_Vac1=0 AND Gripper_DO2_Vac2=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO1_Vac1, 1;
            SetDO Gripper_DO2_Vac2, 1;
          ENDIF
          IF Gripper_DI02_OS2=1 AND Gripper_DO3_Vac3=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO3_Vac3,1;
          ENDIF
          IF Gripper_DI03_OS3=1 AND Gripper_DO4_Vac4=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO4_Vac4,1;
          ENDIF
          IF Gripper_DI04_OS4=1 AND Gripper_DO5_Vac5=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO5_Vac5,1;
          ENDIF
          IF Gripper_DI05_OS5=1 AND Gripper_DO6_Vac6=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO6_Vac6,1;
          ENDIF
          IF Gripper_DI06_OS6=1 AND Gripper_DO7_Vac7=0 AND Gripper_DO8_Vac8=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO7_Vac7, 1;
            SetDO Gripper_DO8_Vac8, 1;
          ENDIF
        ENDIF
        IF Presse_GI00_ProductType=3 OR Presse_GI00_ProductType=6 OR Presse_GI00_ProductType=9 THEN
          IF Gripper_DI01_OS1=1 AND Gripper_DO1_Vac1=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO1_Vac1,1;
          ENDIF
          IF Gripper_DI02_OS2=1 AND Gripper_DO2_Vac2=0 AND Gripper_DO2_Vac2=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO2_Vac2, 1;
            SetDO Gripper_DO3_Vac3, 1;
          ENDIF
          IF Gripper_DI03_OS3=1 AND Gripper_DO4_Vac4=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO4_Vac4,1;
          ENDIF
          IF Gripper_DI04_OS4=1 AND Gripper_DO5_Vac5=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO5_Vac5,1;
          ENDIF
          IF Gripper_DI05_OS5=1 AND Gripper_DO6_Vac6=0 AND Gripper_DO7_Vac7=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO6_Vac6, 1;
            SetDO Gripper_DO7_Vac7, 1;
          ENDIF
          IF Gripper_DI06_OS6=1 AND Gripper_DO8_Vac8=0 THEN
            waittime nRegrpiDelay;
            SetDO Gripper_DO8_Vac8,1;
          ENDIF
        ENDIF
      ENDIF
      !Stop
    ENDPROC

    PROC pPickedPanelsToZero()
      IF gripperGroupVac=0 THEN
        nPickedPanels:=0;
      ENDIF
    ENDPROC
```

```
PROC pWarningLights()
    IF UserSys4_tROB1Status=1 AND nAreaOccupied=0 THEN
        SetDO Local_DO1_GREEN,1;
    ELSEIF UserSys4_tROB1Status=0 AND nAreaOccupied=0 THEN
        SetDO Local_DO1_GREEN,0;
    ENDIF
    IF UserSys2_MotionPause=1 OR UserSys5_StopMotor=1 THEN
        SetDO Local_DO2_YELLOW,1;
    ELSE
        SetDO Local_DO2_YELLOW,0;
    ENDIF
    IF UserSys6_InternalError=1 OR bPal1StatusOK = FALSE OR bPal2StatusOK = FALSE THEN
        SetDO Local_DO3_RED,1;
    ELSE
        SetDO Local_DO3_RED,0;
    ENDIF
ENDPROC

PROC pERRBeeping()
    IF nDoERRBeeping = 1 THEN
        SetDO Local_DO4_BUZZER,0;
        SetDO Local_DO4_BUZZER,1;
        WaitTime 0.2;
        SetDO Local_DO4_BUZZER,0;
        WaitTime 0.8;
        SetDO Local_DO4_BUZZER,1;
        WaitTime 0.2;
        SetDO Local_DO4_BUZZER,0;
        WaitTime 0.8;
        SetDO Local_DO4_BUZZER,1;
        WaitTime 0.2;
        SetDO Local_DO4_BUZZER,0;
        WaitTime 0.8;
        SetDO Local_DO4_BUZZER,1;
        WaitTime 0.2;
        SetDO Local_DO4_BUZZER,0;
        WaitTime 0.8;
        SetDO Local_DO4_BUZZER,1;
        WaitTime 0.2;
        SetDO Local_DO4_BUZZER,0;
        WaitTime 0.8;
        nDoERRBeeping:=0;
    ENDIF
    !Stop;
ENDPROC

PROC pSelectThickness()
    !3mm panel
    IF n3mmOr32mm=3 THEN
        SetDO Gripper_DO10_BYP,0;
        SetDO Gripper_DO11_BCKFL,1;
        nPrev3mmOr32mm:=n3mmOr32mm;
        !32mm panel
    ELSEIF n3mmOr32mm=32 THEN
        SetDO Gripper_DO11_BCKFL,0;
        SetDO Gripper_DO10_BYP,1;
        nPrev3mmOr32mm:=n3mmOr32mm;
    ENDIF
ENDPROC

PROC pPalletsSensorLogic()
    !Read input
    IF Local_DI01_Pal1InPos=1 AND Local_DI05_Pal12InPos=0 THEN
        nPalSensStat1:=nPalletPresent;
    ELSEIF Local_DI01_Pal1InPos=0 AND Local_DI05_Pal12InPos=1 THEN
        nPalSensStat1:=nNoPallet;
    ELSE
        nPalSensStat1:=nERRPalletSensError;
        bResetPal1:=FALSE;
    ENDIF
    IF Local_DI02_Pal2InPos=1 AND Local_DI06_Pal22InPos=0 THEN
        nPalSensStat2:=nPalletPresent;
    ELSEIF Local_DI02_Pal2InPos=0 AND Local_DI06_Pal22InPos=1 THEN
        nPalSensStat2:=nNoPallet;
    ELSE
        nPalSensStat2:=nERRPalletSensError;
        bResetPal2:=FALSE;
    ENDIF
!Start --------- PALLET 1 LOGIC -------- start
    IF nPalSensStat1=nERRPalletSensError AND nPallet1ClockStatus=0 THEN
        Clkstop cPallet1Clock;
        ClkReset cPallet1Clock;
        Clkstart cPallet1Clock;
        nPallet1ClockStatus := 1;
    ELSEIF nPalSensStat1 <> nERRPalletSensError THEN
```

```
      Clkstop cPallet1Clock;
      ClkReset cPallet1Clock;
      nPallet1ClockStatus := 0;
   ENDIF
  IF nPallet1ClockStatus=1 AND ClkRead(cPallet1Clock) > nPalletSensorTimeLimit AND nPallet1Status<>nERRPalletSensError THEN
      nPallet1Status := nERRPalletSensError;
      Clkstop cPallet1Clock;
      ClkReset cPallet1Clock;
      nPallet1ClockStatus:=0;
      pPallet1SensorERR;
   ENDIF
   IF nPallet1Status=nNoPallet AND nPalSensStat1 = nPalletPresent AND nPallet1Status<>nERRPalletSensError THEN
         nPallet1Status:=nPalletPresent;
   ELSEIF nPalSensStat1 = nNoPallet AND nPallet1Status<>nERRPalletSensError THEN
         nPallet1Status := nNoPallet;
   ENDIF
   IF bResetPal1 and nPalSensStat1 = nPalletPresent and nPallet1Status = nERRPalletSensError THEN
      nPallet1Status:=nPalletPresent;
      bResetPal1:=FALSE;
   ELSEIF bResetPal1 and nPalSensStat1 = nNoPallet and nPallet1Status = nERRPalletSensError THEN
      nPallet1Status:=nNoPallet;
      bResetPal1:=FALSE;
  ELSEIF bResetPal1 THEN
     bResetPal1:=FALSE;
   ENDIF
  !END --------- PALLET 1 LOGIC -------- END
  !Start --------- PALLET 2 LOGIC -------- start
   IF nPalSensStat2=nERRPalletSensError AND nPallet2ClockStatus=0 THEN
      Clkstop cPallet2Clock;
      ClkReset cPallet2Clock;
      Clkstart cPallet2Clock;
      nPallet2ClockStatus := 1;
   ELSEIF nPalSensStat2 <> nERRPalletSensError THEN
      Clkstop cPallet2Clock;
      ClkReset cPallet2Clock;
      nPallet2ClockStatus := 0;
   ENDIF
  IF nPallet2ClockStatus=1 AND ClkRead(cPallet2Clock) > nPalletSensorTimeLimit AND nPallet2Status<>nERRPalletSensError THEN
      nPallet2Status := nERRPalletSensError;
      Clkstop cPallet2Clock;
      ClkReset cPallet2Clock;
      nPallet2ClockStatus:=0;
      pPallet2SensorERR;
   ENDIF
   IF nPallet2Status=nNoPallet AND nPalSensStat2 = nPalletPresent AND nPallet2Status<>nERRPalletSensError THEN
         nPallet2Status:=nPalletPresent;
   ELSEIF nPalSensStat2 = nNoPallet AND nPallet2Status<>nERRPalletSensError THEN
         nPallet2Status := nNoPallet;
   ENDIF
   IF bResetPal2 and nPalSensStat2 = nPalletPresent and nPallet2Status = nERRPalletSensError THEN
      nPallet2Status:=nPalletizingFinished;
      bResetPal2:=FALSE;
   ELSEIF bResetPal2 and nPalSensStat2 = nNoPallet and nPallet2Status = nERRPalletSensError THEN
      nPallet2Status:=nNoPallet;
      bResetPal2:=FALSE;
   ELSEIF bResetPal2 THEN
     bResetPal2:=FALSE;
   ENDIF
   !END --------- PALLET 2 LOGIC -------- END
ENDPROC

PROC pScannerControl()
   !stop;
   IF nPalSensStat1=0 AND nPalSensStat2=0 THEN     !No pallet
      IF nScannerFromArea=0 nScannerFromArea:=nScannerOnlyRight;
      nScannerToArea:=nScannerNoPallets;
   ELSEIF nPalSensStat1=1 AND nPalSensStat2=0 THEN !LeftPallet
      IF nScannerFromArea=0 nScannerFromArea:=nScannerNoPallets;
      nScannerToArea:=nScannerOnlyLeft;
   ELSEIF nPalSensStat1=0 AND nPalSensStat2=1 THEN !RightPallet
      IF nScannerFromArea=0 nScannerFromArea:=nScannerBothPallets;
      nScannerToArea:=nScannerOnlyRight;
   ELSEIF nPalSensStat1=1 AND nPalSensStat2=1 THEN !BothPallets
      IF nScannerFromArea=0 nScannerFromArea:=nScannerOnlyLeft;
      nScannerToArea:=nScannerBothPallets;
   ENDIF
   IF nScannerFromArea<>nScannerToArea THEN
      pSwitchAreas nScannerFromArea,nScannerToArea;
      nScannerFromArea:=nScannerToArea;
   ENDIF
ENDPROC
```

```
    PROC pSwitchAreas(num lnFrom,num lnTo)
        !Possible scenarios. N - No pallets. L - Left, R - Right, B - Both
        !N>L, N>R, R>B, R>N, L>B, L>N, B>L, B>R.
        !101
        IF lnFrom=nScannerNoPallets THEN
            IF lnTo=nScannerOnlyLeft THEN
                SETDO Local_DO5_ScannerAreaSwitch1,0;
                SETDO Local_DO6_ScannerAreaSwitch2,1;
                SetDO Local_DO7_ScannerAreaSwitch3,0;
            ELSEIF lnTo=nScannerOnlyRight THEN
                SETDO Local_DO5_ScannerAreaSwitch1,0;
                SETDO Local_DO6_ScannerAreaSwitch2,1;
                SetDO Local_DO7_ScannerAreaSwitch3,0;
                WaitTime nScannerZonesSwDelay;
                SETDO Local_DO5_ScannerAreaSwitch1,1;
                SETDO Local_DO6_ScannerAreaSwitch2,0;
                SetDO Local_DO7_ScannerAreaSwitch3,0;
            ENDIF
        ENDIF
        !stop;
        !100
        IF lnFrom=nScannerOnlyRight THEN
            IF lnTo=nScannerBothPallets THEN
                SETDO Local_DO5_ScannerAreaSwitch1,0;
                SETDO Local_DO6_ScannerAreaSwitch2,1;
                SetDO Local_DO7_ScannerAreaSwitch3,1;
            ELSEIF lnTo=nScannerNoPallets THEN
                SETDO Local_DO5_ScannerAreaSwitch1,0;
                SETDO Local_DO6_ScannerAreaSwitch2,1;
                SetDO Local_DO7_ScannerAreaSwitch3,0;
                WaitTime nScannerZonesSwDelay;
                SETDO Local_DO5_ScannerAreaSwitch1,1;
                SETDO Local_DO6_ScannerAreaSwitch2,0;
                SetDO Local_DO7_ScannerAreaSwitch3,1;
            ENDIF
        ENDIF
        !010
        IF lnFrom=nScannerOnlyLeft THEN
            IF lnTo=nScannerBothPallets THEN
                SETDO Local_DO5_ScannerAreaSwitch1,1;
                SETDO Local_DO6_ScannerAreaSwitch2,0;
                SetDO Local_DO7_ScannerAreaSwitch3,0;
                WaitTime nScannerZonesSwDelay;
                SETDO Local_DO5_ScannerAreaSwitch1,0;
                SETDO Local_DO6_ScannerAreaSwitch2,1;
                SetDO Local_DO7_ScannerAreaSwitch3,1;
            ELSEIF lnTo=nScannerNoPallets THEN
                SETDO Local_DO5_ScannerAreaSwitch1,1;
                SETDO Local_DO6_ScannerAreaSwitch2,0;
                SetDO Local_DO7_ScannerAreaSwitch3,1;
            ENDIF
        ENDIF
        !011
        IF lnFrom=nScannerBothPallets THEN
            IF lnTo=nScannerOnlyLeft THEN
                SETDO Local_DO5_ScannerAreaSwitch1,1;
                SETDO Local_DO6_ScannerAreaSwitch2,0;
                SetDO Local_DO7_ScannerAreaSwitch3,0;
                WaitTime nScannerZonesSwDelay;
                SETDO Local_DO5_ScannerAreaSwitch1,0;
                SETDO Local_DO6_ScannerAreaSwitch2,1;
                SetDO Local_DO7_ScannerAreaSwitch3,0;
            ELSEIF lnTo=nScannerOnlyRight THEN
                SETDO Local_DO5_ScannerAreaSwitch1,1;
                SETDO Local_DO6_ScannerAreaSwitch2,0;
                SetDO Local_DO7_ScannerAreaSwitch3,0;
            ENDIF
        ENDIF
    ENDPROC

    PROC pPallet1SensorERR()
        nDoERRBeeping:=1;
        ErrLog errid_PalletSensor,ERR_PalletSensor,ERR_msg1_PalletSensor,ERR_msg2_PalletSensor,ERR_msg3_PalletSensor,ERR_msg4_PalletSensor;
    ENDPROC

    PROC pPallet2SensorERR()
        nDoERRBeeping:=1;
        ErrLog
errid_Pallet2Sensor,ERR_Pallet2Sensor,ERR_msg1_Pallet2Sensor,ERR_msg2_Pallet2Sensor,ERR_msg3_Pallet2Sensor,ERR_msg4_Pallet2Sensor;

    ENDPROC
ENDMODULE
```

```
MODULE CalibData
    PERS wobjdata wobjPallet1:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[-398,1568,-145],[1,0,0,0]]];
    PERS wobjdata wobjPallet2:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[-408.6,-1547.84,-145],[1,0,0,0]]];
    PERS wobjdata wobjCnv:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[0,0,845],[1,0,0,0]]];
    PERS tooldata tVacGripper:=[TRUE,[[0,0,163],[1,0,0,0]],[27.3,[0,0,50],[1,0,0,0],0,0,0]];
    CONST speeddata vSearch:=[100,30,100,100];
    CONST speeddata vPick:=[20,30,100,100];
    CONST speeddata vNoLoad:=[2200,90,100,100];
    VAR speeddata vCarry;
    CONST speeddata vCarryThin:=[1800,50,100,100];
    CONST speeddata vCarryThick:=[400,30,100,100];
    CONST speeddata vSlow:=[200,15,100,100];
    CONST speeddata vRetract:=[400,30,100,100];
    VAR speeddata vRise;
    CONST speeddata vRiseThin:=[4,30,70,100];
    CONST speeddata vRiseThick:=[3,30,70,100];
ENDMODULE

MODULE ERR_Module
    !Common vars:
    VAR errstr UnusedStr:=" ";
    VAR errstr ERR_msg2_ACTION:="ACTION:";
    !Error definitions:
    !#########################
    VAR num errid_ProdDropd:=4800;
    VAR errstr ERR_ProdDropd:="ERR_ProductDropped!";
    VAR errstr ERR_msg1_ProdDropd:="Product has fallen from the gripper. Please check!";
    VAR errstr ERR_msg2_ProdDropd:="1) Adjust the stacks on pallet";
    VAR errstr ERR_msg3_ProdDropd:="2) Check suction cups and tubing";
    VAR errstr ERR_msg4_ProdDropd:="3) Place the product back to gripper";
    !#########################
    VAR num errid_MultipleLayers:=4801;
    VAR errstr ERR_MultipleLayers:="ERR_MultipleLayers!";
    VAR errstr ERR_msg1_MultipleLayers:="Robot can't separate additional layers!";
    VAR errstr ERR_msg2_MultipleLayers:="1) Check panel thickness";
    VAR errstr ERR_msg3_MultipleLayers:="2) Remove additional layers from the gripper";
    VAR errstr ERR_msg4_MultipleLayers:="3) Exit the station. Press Motors ON and Play";
    !#########################
    VAR num errid_MultipleLayers2:=4802;
    VAR errstr ERR_MultipleLayers2:="ERR_MultipleLayers2!";
    VAR errstr ERR_msg1_MultipleLayers2:="Additional layers were discovered during movement!";
    VAR errstr ERR_msg2_MultipleLayers2:="1) Check panel thickness";
    VAR errstr ERR_msg3_MultipleLayers2:="2) Remove additional layers from the gripper";
    VAR errstr ERR_msg4_MultipleLayers2:="3) 'Blue' Capacity sensor's sensitivity needs adjustment";
    !#########################
    PERS num nVacGuard:=1;

    PROC pFallenProdERR()
        VAR robtarget rTempTarget;
        ErrLog errid_ProdDropd,ERR_ProdDropd,ERR_msg1_ProdDropd,ERR_msg2_ProdDropd,ERR_msg3_ProdDropd,ERR_msg4_ProdDropd;
        SetDO UserSys6_InternalError,1;
        nVacGuard:=1;
        nDoERRBeeping:=1;
        IF bInPicking THEN
            stop;
            rTempTarget:=CRobT(\Tool:=tVacGripper\WObj:=wobj0);
            rTempTarget.trans.z:=rTempTarget.trans.z + 300;
            MoveL rTempTarget, vRetract, fine, tVacGripper\WObj:=wobj0;
            StopMove;
            stop;
            StartMove;
        ELSE
            StopMove;
            Stop;
            Startmove;
        ENDIF
        SetDO UserSys6_InternalError,0;
    ENDPROC

    PROC pMultipleLayersERR()
        ErrLog
errid_MultipleLayers,ERR_MultipleLayers,ERR_msg1_MultipleLayers,ERR_msg2_MultipleLayers,ERR_msg3_MultipleLayers,ERR_msg4_MultipleLay
ers;
        SetDO UserSys6_InternalError,1;
        nDoERRBeeping:=1;
        StopMove;
        Stop;
        Startmove;
        SetDO UserSys6_InternalError,0;
    ENDPROC


    PROC pMultipleLayersMovingERR()
        ErrLog
errid_MultipleLayers2,ERR_MultipleLayers2,ERR_msg1_MultipleLayers2,ERR_msg2_MultipleLayers2,ERR_msg3_MultipleLayers2,ERR_msg4_Multip
leLayers2;
```

```
        SetDO UserSys6_InternalError,1;
        nDoERRBeeping:=1;
        StopMove;
        Stop;
        Startmove;
        SetDO UserSys6_InternalError,0;
    ENDPROC
ENDMODULE

MODULE MainModule
    !***********************************************************
    !Järg: Search
    ! Module:  MainModule
    !
    ! Description:
    !   <Insert description here>
    !
    ! Author: eekrpii
    !
    ! Version: 1.0
    !
    !***********************************************************

    RECORD gripperConfData
        num g1;
        num g2;
        num g3;
        num g4;
        num g5;
        num g6;
        num g7;
        num g8;
    ENDRECORD

    !nProdType 1 - 9
    !nProdType 3mm, 32mm.
    !sProdTye "mm x mm"
    !nGripSections 2 – 6

    RECORD productData
        num nProdType;
        num nProdThickness;
        string sProdDimensions;
        num nGripSections;
    ENDRECORD

    !        _______
    !       | 1   5 |
    !       | 2   6 |
    !       | 3   7 |
    !       |_4___8_|

    !Gripper is built to have 8 separate vacuum generators/
    !/8 separate groups. (Look picture above)
    !These groups allow us to combine variety of different
    !vacuum sections.
    !The numbers are representing the order of the groups.
    !
    !On the following lines are defined different
    !configurations of the vacuum sections - reading order
    !is from left to right. The numbers in configuration
    !below, are showing, into what section the given group
    !belongs to. Also, how many sections are used
    !in representing configuration.
    !
    !For example. As we can see. "gripperCnf1" has configured
    !to have only two sections - First on the left hand side
    !and second on the right hand side. You can see
    !different configurations in few lines below.
    !***---GRIPPER VARIABLES---***
    !***Constants***
    CONST gripperConfData gripperCnf1:=[1,1,1,3,2,2,2,4];
    CONST gripperConfData gripperCnf2:=[2,2,2,2,1,1,1,1];
    CONST gripperConfData gripperCnf3:=[1,1,1,1,1,1,1,1];
    CONST num nTime_GripRelease:=0;
    VAR string sProdSize;
    !Commands for the gripper
    CONST num nGrip:=1;
    CONST num nRelease:=0;
    CONST num nAllGroups:=99;
    PERS num nPickedPanels;
    !Currently active gripper configuration
    PERS gripperConfData grActiveConf:=[2,2,2,2,1,1,1,1];
    !!!__GRIPPER CONFIGURATIONS VISUALIZATION__!!!
    !   Gripper 2 sections. sGripperConf1:
    !        _______ Product Types: 12;14;15;17;18
```

```
!      | 1   2 |
!      | 1   2 |
!      | 1   2 |
!      |_1___2_|
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!   Gripper 4 sections. gripperCnf2:
!        _______ Product Types: 2;4;5;7;8
!      | 1   3 |
!      | 1   3 |
!      | 2   4 |
!      |_2___4_|
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!   Gripper 6 sections. gripperCnf3:
!        _______ Product Types: 1;
!      | 1   2 |
!      | 4   3 |
!      | 4   3 |
!      |_5___6_|
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!   Gripper 2 sections. gripperCnf4:
!        _______ Product Types: 3;6
!      | 1   1 |
!      | 1   1 |
!      | 2   2 |
!      |_2___2_|
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!   Gripper 3 sections. gripperCnf5:
!        _______ Product Types: 9
!      | 1   1 |
!      | 2   2 |
!      | 2   2 |
!      |_3___3_|
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!END---GRIPPER VARIABLES---END
!***---PRODUCT VARIABLES---***
PERS num nPrevProductType:=5;
!************************************************************
PERS productData ProdData:=[5,31,"605x437",2];
!************************************************************
!***Variables***
!************************************************************
!***---PALLETS VARIABLES---***
PERS num nLayersOnPal1:=0;
PERS num nLayersOnPal2:=239;
PERS num nPallet1Status:=-3;
PERS num nPallet2Status:=-3;
PERS bool bProductionFinishedPB:=FALSE;
CONST num nProductThickness:=31;
PERS num n3mmOr32mm:=32;
CONST num n3mm:=3;
CONST num n32mm:=32;
PERS num nPrev3mmOr32mm:=32;
!Statuses:
!0 - Palletizing area is empty. New pallet can be placed to the depalletizing area.
!1 - New pallet in pos. Need to measure edges and height.
!2 - pallet has been measured and destacking is in progress.
!3 - Palletizing finished. Can be removed from palletizing/depalletizing area.
!-3 - Error. Pallet presence sensor fault!
!************************************************************
!Contstants
CONST num Yes:=1;
CONST num No:=0;
!Palletizing constants
CONST num nNoPallet:=0;
CONST num nPalletPresent:=1;
CONST num nPalletizingInProgress:=2;
CONST num nPalletizingFinished:=3;
CONST num nERRPalletSensError:=-3;
CONST num nGripperSensorZoffs:=34;
VAR intnum inMotionsStop;
VAR intnum inFallingProduct{6};
VAR intnum inMultiLayer{6};
VAR num nMultipleLayerGuard;
PERS num nDoERRBeeping;
VAR num errorid:=9903;
VAR errstr arg:="P1";
PERS num nPickThisMuchMoreLyrsAndStop;
VAR bool bPickThisMuchMoreLyrsAndStop;
PERS num nAreaOccupied;
PERS string ProcessName;
PERS string ProcessNameEE;
```

```
    PROC pInitialize()
        AccSet 40,40;
        pRetract;
        !Stop;
    ENDPROC

    PROC main()
        StartMove;
        SetDO UserSys6_InternalError,0;
        nAreaOccupied:=0;
        !!
        IDelete inMotionsStop;
        CONNECT inMotionsStop WITH tMotionPause;
        ISignalDI UserSys2_MotionPause,1,inMotionsStop;
        !!
        FOR i FROM 1 TO 6 DO
            IDelete inFallingProduct{i};
            CONNECT inFallingProduct{i} WITH tFallingProducts;
        ENDFOR
        ISignalDI Gripper_DI01_OS1,0,inFallingProduct{1};
        ISignalDI Gripper_DI02_OS2,0,inFallingProduct{2};
        ISignalDI Gripper_DI03_OS3,0,inFallingProduct{3};
        ISignalDI Gripper_DI04_OS4,0,inFallingProduct{4};
        ISignalDI Gripper_DI05_OS5,0,inFallingProduct{5};
        ISignalDI Gripper_DI06_OS6,0,inFallingProduct{6};
        !!
        FOR i FROM 1 TO 6 DO
            IDelete inMultiLayer{i};
            CONNECT inMultiLayer{i} WITH tHoveringMultiLayerControl;
        ENDFOR
        ISignalDI Gripper_DI07_CS1,0,inMultiLayer{1};
        ISignalDI Gripper_DI08_CS2,0,inMultiLayer{2};
        ISignalDI Gripper_DI09_CS3,0,inMultiLayer{3};
        ISignalDI Gripper_DI10_CS4,0,inMultiLayer{4};
        ISignalDI Gripper_DI11_CS5,0,inMultiLayer{5};
        ISignalDI Gripper_DI12_CS6,0,inMultiLayer{6};
        !stop;
        pInitialize;
        !stop;
        WHILE TRUE DO
            !Stop;
            IF nPickedPanels=0 AND (((nPallet1Status=nPalletPresent OR nPallet1Status=nPalletizingInProgress) AND
nPallet2Status<>nPalletizingInProgress) OR ((nPallet2Status=nPalletPresent OR nPallet2Status=nPalletizingInProgress) AND
nPallet1Status<>nPalletizingInProgress)) THEN
                pGetProductType;
                pPickingMain;
            ENDIF
            IF nPickedPanels>0 THEN
                pPlacingMain;
            ENDIF
            IF nPickThisMuchMoreLyrsAndStop > 0 THEN
                        bPickThisMuchMoreLyrsAndStop := TRUE;
                        nPickThisMuchMoreLyrsAndStop :=nPickThisMuchMoreLyrsAndStop - 1;
            ENDIF
            IF nPickThisMuchMoreLyrsAndStop <= 0 AND bPickThisMuchMoreLyrsAndStop THEN
                MoveJ rMidPos,vNoLoad,z200,tVacGripper\WObj:=wobj0;
                        bPickThisMuchMoreLyrsAndStop := FALSE;
                        Stop;
            ENDIF
            waittime 0.1;
            !stop;
        ENDWHILE
        !stop;
    ENDPROC

    PROC pSearchPalletHeight(robtarget searchPos,inout num palletHeight,inout wobjdata currWobj, inout num nPalHeight, num lnSearchHeightLimit, num
nPalStat)
        VAR robtarget rStoredPos;
        ProcessName:="Height search";
        ProcessNameEE:="Kõrguse otsimine";
        IF nPalStat = nPalletPresent THEN
            MoveL Offs(searchPos,0,0,1500),vNoLoad,z50,tVacGripper\WObj:=currWobj;
        ELSE
            MoveL Offs(searchPos,0,0,nPalHeight + 150),vNoLoad,z50,tVacGripper\WObj:=currWobj;
        ENDIF
        SearchL\SStop,Gripper_DI01_OS1,rStoredPos,offs(searchPos,0,0,lnSearchHeightLimit),vSearch,tVacGripper\WObj:=currWobj;
        palletHeight:=rStoredPos.trans.z-nGripperSensorZoffs;
        !Stop;
        ERROR
        nPalHeight:=nPalletHeightLimit;
        TryNext;
    ENDPROC
```

```
PROC pGripper(gripperConfData act,num groupNr,num gripRelease)
  IF act.g1=groupNr OR groupNr=nAllGroups THEN
    SetDO Gripper_DO1_Vac1,gripRelease;
  ENDIF
  IF act.g2=groupNr OR groupNr=nAllGroups THEN
    SetDO Gripper_DO2_Vac2,gripRelease;
  ENDIF
  IF act.g3=groupNr OR groupNr=nAllGroups THEN
    SetDO Gripper_DO3_Vac3,gripRelease;
  ENDIF
  IF act.g4=groupNr OR groupNr=nAllGroups THEN
    SetDO Gripper_DO4_Vac4,gripRelease;
  ENDIF
  IF act.g5=groupNr OR groupNr=nAllGroups THEN
    SetDO Gripper_DO5_Vac5,gripRelease;
  ENDIF
  IF act.g6=groupNr OR groupNr=nAllGroups THEN
    SetDO Gripper_DO6_Vac6,gripRelease;
  ENDIF
  IF act.g7=groupNr OR groupNr=nAllGroups THEN
    SetDO Gripper_DO7_Vac7,gripRelease;
  ENDIF
  IF act.g8=groupNr OR groupNr=nAllGroups THEN
    SetDO Gripper_DO8_Vac8,gripRelease;
  ENDIF
  !If releasing vacuum, gripper will be empty.
  IF gripRelease=nGrip THEN
    nPickedPanels:=ProdData.nGripSections;
    AccSet nAccNCarry,nRampCarry;
  ENDIF
  WaitTime nTime_GripRelease;
  !stop
ENDPROC

PROC pGetProductType()
  ProcessName:="Getting product type";
  ProcessNameEE:="Tootetüübi kontroll";
  ProdData.nProdType:=GInput(Presse_GI00_ProductType);
  IF ProdData.nProdType<>nPrevProductType OR n3mmOr32mm<>nPrev3mmOr32mm THEN
    TEST ProdData.nProdType
    CASE 1:
      ProdData.nProdThickness:=n3mm;
      !we are giving default value for the thickness variable.
      !However, if the product type is e.g. 11, then the thickness will be overwritten
      IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
      ProdData.sProdDimensions:="450x337";
      ProdData.nGripSections:=4;
      grActiveConf:=gripperCnf1;
      rPickingPositionPal1:=rPallet1_PickPos;
      rPickingPositionPal2:=rPallet2_PickPos;
      !!This data is given in placing module according to the conveyor config and case type
    CASE 2:
      ProdData.nProdThickness:=n3mm;
      IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
      ProdData.sProdDimensions:="605x337";
      ProdData.nGripSections:=2;
      grActiveConf:=gripperCnf2;
      rPickingPositionPal1:=rPallet1_PickPos;
      rPickingPositionPal2:=rPallet2_PickPos;
    CASE 3:
      ProdData.nProdThickness:=n3mm;
      IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
      ProdData.sProdDimensions:="905x537";
      ProdData.nGripSections:=2;
      grActiveConf:=gripperCnf2;
      rPickingPositionPal1:=rPallet1_Pick900;
      rPickingPositionPal2:=rPallet2_Pick900;
    CASE 4:
      ProdData.nProdThickness:=n3mm;
      IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
      ProdData.sProdDimensions:="450x437";
      ProdData.nGripSections:=2;
      grActiveConf:=gripperCnf2;
      rPickingPositionPal1:=rPallet1_PickPos;
      rPickingPositionPal2:=rPallet2_PickPos;
    CASE 5:
      ProdData.nProdThickness:=n3mm;
      IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
      ProdData.sProdDimensions:="605x437";
      ProdData.nGripSections:=2;
      grActiveConf:=gripperCnf2;
      rPickingPositionPal1:=rPallet1_PickPos;
      rPickingPositionPal2:=rPallet2_PickPos;
    CASE 6:
      ProdData.nProdThickness:=n3mm;
      IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
```

```
            ProdData.sProdDimensions:="905x437";
            ProdData.nGripSections:=1;
            grActiveConf:=gripperCnf3;
            rPickingPositionPal1:=rPallet1_Pick900;
            rPickingPositionPal2:=rPallet2_Pick900;
        CASE 7:
            ProdData.nProdThickness:=n3mm;
            IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
            ProdData.sProdDimensions:="450x537";
            ProdData.nGripSections:=2;
            grActiveConf:=gripperCnf2;
            rPickingPositionPal1:=rPallet1_PickPos;
            rPickingPositionPal2:=rPallet2_PickPos;
        CASE 8:
            ProdData.nProdThickness:=n3mm;
            IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
            ProdData.sProdDimensions:="605x537";
            ProdData.nGripSections:=2;
            grActiveConf:=gripperCnf2;
            rPickingPositionPal1:=rPallet1_PickPos;
            rPickingPositionPal2:=rPallet2_PickPos;
        CASE 9:
            ProdData.nProdThickness:=n3mm;
            IF n3mmOr32mm=n32mm ProdData.nProdThickness:=nProductThickness;
            ProdData.sProdDimensions:="905x337";
            ProdData.nGripSections:=2;
            grActiveConf:=gripperCnf2;
            rPickingPositionPal1:=rPallet1_Pick900;
            rPickingPositionPal2:=rPallet2_Pick900;
        DEFAULT:
        ENDTEST
        !In case the product type has been changed in the middle of the palletizing progress, we are marking the pallet as finished.
        !Operator must come to change the pallet.
        nPrevProductType:=ProdData.nProdType;
        nPrev3mmOr32mm:=n3mmOr32mm;
        bIsPanelSlidingStopperSetUp:=FALSE;
        IF nPallet1Status=nPalletizingInProgress nPallet1Status:=nPalletizingFinished;
        IF nPallet2Status=nPalletizingInProgress nPallet2Status:=nPalletizingFinished;
        IF gripperGroupVac=0 THEN
            nPickedPanels:=0;
        ELSE
            !RAISE error Product type has changed, but gripper still holds products
            Stop;
        ENDIF
    ENDIF
    pRobSpeed;
ENDPROC

PROC pRetract()
    VAR jointtarget jTempTarget;
    VAR robtarget rTempTarget;
    !Stop;
    ProcessName:="Retracting";
    ProcessNameEE:="Taganemine";
    rTempTarget:=CRobT(\Tool:=tVacGripper\WObj:=wobj0);
    !AND jTempTarget.robax.rax_1 > jRetractJoint.robax.rax_1 + 10 OR jTempTarget.robax.rax_1 < jRetractJoint.robax.rax_1 - 10
    jTempTarget:=CJointT();
    IF (rTempTarget.trans.z < rRetractPos.trans.z)THEN
        rTempTarget.trans.z:=rRetractPos.trans.z;
        MoveL rTempTarget,vRetract,z50,tVacGripper\WObj:=wobj0;
    ENDIF
    jTempTarget:=CJointT();
    jTempTarget.robax.rax_2:=jRetractJoint.robax.rax_2;
    jTempTarget.robax.rax_3:=jRetractJoint.robax.rax_3;
    MoveAbsJ jTempTarget\NoEOffs,vRetract,z50,tVacGripper;
    jTempTarget.robax.rax_1:=jRetractJoint.robax.rax_1;
    MoveAbsJ jTempTarget\NoEOffs, vRetract, z0, tVacGripper;
    MoveL rRetractPos, vSlow, fine, tVacGripper\WObj:=wobj0;
    !stop;
ENDPROC

PROC pRobSpeed()
    IF n3mmOr32mm=n3mm THEN
        vCarry:=vCarryThin;
        vRise:=vRiseThin;
    ELSEIF n3mmOr32mm=n32mm THEN
        vCarry:=vCarryThick;
        vRise:=vRiseThick;
    ENDIF
ENDPROC

TRAP tMotionPause
    !This trap is called only, if the system is in auto mode.
    !The "AND AUTO MODE" is done in cross connections
                                var num seconds;
    IDelete inMotionsStop;
```

```
    CONNECT inMotionsStop WITH tMotionPause;
    ISignalDI UserSys2_MotionPause,1,inMotionsStop;
    WHILE UserSys2_MotionPause=1 DO
      nAreaOccupied:=1;
      StopMove\Quick;
                                    waittime 0.2;
                                    seconds:=seconds+0.2;
    ENDWHILE
                        IF seconds > 0.8 THEN
      SetDO Local_DO4_BUZZER,0;
      SetDO Local_DO4_BUZZER,1;
      SetDO Local_DO1_GREEN,1;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,0;
      SetDO Local_DO1_GREEN,0;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,1;
      SetDO Local_DO1_GREEN,1;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,0;
      SetDO Local_DO1_GREEN,0;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,1;
      SetDO Local_DO1_GREEN,1;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,0;
      SetDO Local_DO1_GREEN,0;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,1;
      SetDO Local_DO1_GREEN,1;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,0;
      SetDO Local_DO1_GREEN,0;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,1;
      SetDO Local_DO1_GREEN,1;
      WaitTime 0.5;
      SetDO Local_DO4_BUZZER,0;
      SetDO Local_DO1_GREEN,0;
      WaitTime 1;
      SetDO Local_DO1_GREEN,1;
    ELSE
      waittime 3;
    ENDIF
    StartMove;
    nAreaOccupied:=0;
  ENDTRAP

  TRAP tFallingProducts
    !Stop;
    IF Gripper_DO1_Vac1=1 AND Gripper_DI01_OS1=0 pFallenProdERR;
    IF Gripper_DO2_Vac2=1 AND Gripper_DO3_Vac3=1 AND Gripper_DI02_OS2=0 pFallenProdERR;
    IF Gripper_DO4_Vac4=1 AND Gripper_DI03_OS3=0 pFallenProdERR;
    IF Gripper_DO5_Vac5=1 AND Gripper_DI04_OS4=0 pFallenProdERR;
    IF Gripper_DO6_Vac6=1 AND Gripper_DO7_Vac7=1 AND Gripper_DI05_OS5=0 pFallenProdERR;
    IF Gripper_DO8_Vac8=1 AND Gripper_DI06_OS6=0 pFallenProdERR;
    !Stop;
  ENDTRAP

  TRAP tHoveringMultiLayerControl
    IF nMultipleLayerGuard=1 AND n3mmOr32mm=n3mm THEN
      IF Gripper_DO1_Vac1=1 AND Gripper_DI07_CS1=0 pMultipleLayersMovingERR;
      IF Gripper_DO2_Vac2=1 AND Gripper_DO3_Vac3=1 AND Gripper_DI08_CS2=0 pMultipleLayersMovingERR;
      IF Gripper_DO4_Vac4=1 AND Gripper_DI09_CS3=0 pMultipleLayersMovingERR;
      IF Gripper_DO5_Vac5=1 AND Gripper_DI10_CS4=0 pMultipleLayersMovingERR;
      IF Gripper_DO6_Vac6=1 AND Gripper_DO7_Vac7=1 AND Gripper_DI11_CS5=0 pMultipleLayersMovingERR;
      IF Gripper_DO8_Vac8=1 AND Gripper_DI12_CS6=0 pMultipleLayersMovingERR;
    ENDIF
  ENDTRAP
ENDMODULE

MODULE PickingModule
  CONST num nPalletHeightLimit:=9;
  PERS num nPalletHeightPal1:=4.87183;
  PERS num nPalletHeightPal2:=-3.55887;
  CONST num nPal1SearchHeightLimit:=50;
  CONST num nPal2SearchHeightLimit:=40;
  VAR num nPanelsSeparationTries;
  PERS num nPickedLayers:=4;
  CONST num nPickedLayersLimit:=12;
  PERS num nEnFallenPartsRegrip;
  PERS num nRegrpiDelay;
  CONST num nPal1FirstRiseLevel:=12;
  CONST num nPal2FirstRiseLevel:=9;
  CONST num nPallet1HeightCoef:=-7;
  CONST num nPallet2HeightCoef:=-5;
```

```
   VAR bool bInPicking;
   PERS bool bIsPanelSlidingStopperSetUp;

   PROC pBlowAirBetweenPanels(\switch BlowLonger)
      IF ProdData.nProdThickness=3 THEN
         !Blowing with the nozzle and cylinder
         WaitRob\InPos;
         !Blowing air and cylinder outwards movement
         IF Present(BlowLonger) THEN
            SetDO Gripper_DO12_BLWRC,1;
            PulseDO\PLength:=2,Gripper_DO09_BLWR;
            WaitDI Gripper_DI13_CylFB,1\MaxTime:=5;
            SetDO Gripper_DO12_BLWRC,0;
            WaitDI Gripper_DI13_CylFB,0;
            !SetDO Gripper_DO09_BLWR,0;
         ELSE
            SetDO Gripper_DO12_BLWRC,1;
            SetDO Gripper_DO09_BLWR,1;
            WaitDI Gripper_DI13_CylFB,1\MaxTime:=2;
            !Stop blowing and pull the cylinder back in.
            SetDO Gripper_DO09_BLWR,0;
            WaitTime 0.7;
            SetDO Gripper_DO12_BLWRC,0;
            WaitDI Gripper_DI13_CylFB,0;
            !Stop;
            WaitTime 1;
         ENDIF
      ENDIF
   ERROR
      TRYNEXT;
   ENDPROC

   PROC pPickingMain()
      ProcessName:="Picking process";
      ProcessNameEE:="Aluselt haaramine";
      IF nPallet1Status=nPalletizingInProgress AND nPallet2Status=nPalletizingInProgress RETURN ;
      !Pallet 1 picking
      IF (nPallet1Status=nPalletPresent OR nPallet1Status=nPalletizingInProgress) AND nPallet2Status<>nPalletizingInProgress AND nPickedPanels=0
THEN
         IF nPallet1Status=nPalletPresent AND n3mmOr32mm=n3mm bIsPanelSlidingStopperSetUp:=FALSE;
         !Stop;
         pPicking rPickingPositionPal1,offs(rPickingPositionPal1,885,-
300,1780),nPallet1Status,nPalletHeightPal1,wobjPallet1,nPal1FirstRiseLevel,nPal1SearchHeightLimit,nPallet1HeightCoef,1;
      ENDIF
      !Pallet 2 picking
      IF (nPallet2Status=nPalletPresent OR nPallet2Status=nPalletizingInProgress) AND nPallet1Status<>nPalletizingInProgress AND nPickedPanels=0
THEN
         IF nPallet2Status=nPalletPresent AND n3mmOr32mm=n3mm bIsPanelSlidingStopperSetUp:=FALSE;
         pPicking
rPickingPositionPal2,offs(rPickingPositionPal2,1250,640,1925),nPallet2Status,nPalletHeightPal2,wobjPallet2,nPal2FirstRiseLevel,nPal2SearchHeightLimit
,nPallet2HeightCoef,2;
         !Stop;
      ENDIF
      !Layers counting is for searching re-search
      nPickedLayers:=nPickedLayers+1;
   ENDPROC

   PROC pPicking(inout robtarget rPalPickPos,robtarget rPalAppPos,inout num nPalStat,inout num nPalHeight,inout wobjdata wobjPal,num
lnFirstRiseLevel,num lnSearchHeightLimit,num nPalletHeightCoef,num nPalNo)
      VAR robtarget rStoredPosDummy;
      MoveL rPalAppPos,vNoLoad,z200,tVacGripper\WObj:=wobjPal;
      MoveJ offs(rPalPickPos,0,0,1700),vNoLoad,z50,tVacGripper\WObj:=wobjPal;
      IF nPalStat=nPalletPresent OR nPickedLayers>=nPickedLayersLimit THEN
         pSearchPalletHeight rPalPickPos,nPalHeight,wobjPal,nPalHeight,lnSearchHeightLimit,nPalStat;
         nPickedLayers:=0;
         IF nPalHeight-(ProdData.nProdThickness/2)<=nPalletHeightLimit THEN
            nPalStat:=nPalletizingFinished;
            pRetract;
            RETURN ;
         ENDIF
         IF NOT bIsPanelSlidingStopperSetUp stop;
      ELSEIF nPalStat=nPalletizingInProgress THEN
         MoveL offs(rPalPickPos,0,0,nPalHeight+40),vNoLoad,z5,tVacGripper\WObj:=wobjPal;
         IF NOT bIsPanelSlidingStopperSetUp stop;
      ELSE
         pRetract;
         RETURN ;
      ENDIF
      nPalStat:=nPalletizingInProgress;
      PICK:
      bInPicking:=TRUE;
      IF nPanelsSeparationTries=0 THEN
         MoveL offs(rPalPickPos,0,0,nPalHeight+nPalletHeightCoef),vPick,fine,tVacGripper\WObj:=wobjPal;
         StartMove;
         pGripper grActiveConf,nAllGroups,nGrip;
         IF n3mmOr32mm=n3mm THEN
```

```
                MoveL offs(rPalPickPos,0,0,nPalHeight+lnFirstRiseLevel),vRise,z5,tVacGripper\WObj:=wobjPal;
                pBlowAirBetweenPanels;
SearchL\SStop,UserSys9_DI_CS_ReadingOK\HighLevel,rStoredPosDummy,offs(rPalPickPos,0,0,nPalHeight+50),vRise,tVacGripper\WObj:=wobjPal;
                WaitRob\InPos;
            ELSE
                MoveL offs(rPalPickPos,0,0,nPalHeight+30),vRise,fine,tVacGripper\WObj:=wobjPal;
            ENDIF
            !Stop;
        ELSEIF nPanelsSeparationTries=1 THEN
            MoveL offs(rPalPickPos,0,0,nPalHeight+14),vSlow,fine,tVacGripper\WObj:=wobjPal;
            pBlowAirBetweenPanels;
            MoveL offs(rPalPickPos,0,0,nPalHeight+50),vCarry,fine,tVacGripper\WObj:=wobjPal;
        ELSEIF nPanelsSeparationTries=2 THEN
            WaitTime 1.5;
            MoveL offs(rPalPickPos,0,0,nPalHeight+14),vSlow,fine,tVacGripper\WObj:=wobjPal;
            pBlowAirBetweenPanels\BlowLonger;
            MoveL offs(rPalPickPos,0,0,nPalHeight+50),vCarry,fine,tVacGripper\WObj:=wobjPal;
        ELSEIF nPanelsSeparationTries=3 THEN
            MoveL offs(rPalPickPos,0,0,nPalHeight+400),vCarry,fine,tVacGripper\WObj:=wobjPal;
            WaitTime 1.5;
        ELSE
            pMultipleLayersERR;
            nPanelsSeparationTries:=0;
            !Stop;
            !GOTO PICK;
        ENDIF
        nEnFallenPartsRegrip:=1;
        bInPicking:=FALSE;
        IF n3mmOr32mm=n3mm THEN
            IF gripperGroupCS=63 THEN
                nPanelsSeparationTries:=0;
            ELSE
                nPanelsSeparationTries:=nPanelsSeparationTries+1;
                GOTO PICK;
            ENDIF
        ENDIF
        !nPalHeight:=nPalHeight-ProdData.nProdThickness;
        MoveL offs(rPalPickPos,0,0,1700),vCarry,z200,tVacGripper\WObj:=wobjPal;
        nMultipleLayerGuard:=1;
        MoveL rPalAppPos,vCarry,z200,tVacGripper\WObj:=wobjPal;
        nPalHeight:=nPalHeight-ProdData.nProdThickness;
        IF nPalHeight-(ProdData.nProdThickness/2)<=nPalletHeightLimit THEN
            nPalStat:=nPalletizingFinished;
        ENDIF
        !Stop;
        IF nPalNo=2 THEN
            MoveJ offs(rPickingPositionPal2,1500,1500,1925),vCarry,z150,tVacGripper\WObj:=wobjPallet2;
        ENDIF
    ERROR
        IF ERRNO=1072 OR errno=1073 THEN
            !Errorhandling for searchL in picking
            TRYNEXT;
        ENDIF
    ENDPROC
ENDMODULE

MODULE PlacingModule
    CONST num nAccNoLoad:=50;
    CONST num nRampNoLoad:=70;
    CONST num nAccNCarry:=5;
    CONST num nRampCarry:=5;

    PROC pPlacingMain()
        ProcessName:="Placing process";
        ProcessNameEE:="Asetamine";
        TEST ProdData.nProdType
        CASE 1:
            IF n3mmOr32mm=n3mm pPlaceType1;
            IF n3mmOr32mm=n32mm pPlaceType11;
        CASE 2:
            IF n3mmOr32mm=n3mm pPlaceType2;
            IF n3mmOr32mm=n32mm pPlaceType12;
        CASE 3:
            IF n3mmOr32mm=n3mm pPlaceType3;
            IF n3mmOr32mm=n32mm pPlaceType13;
        CASE 4:
            IF n3mmOr32mm=n3mm pPlaceType4;
            IF n3mmOr32mm=n32mm pPlaceType14;
        CASE 5:
            IF n3mmOr32mm=n3mm pPlaceType5;
            IF n3mmOr32mm=n32mm pPlaceType15;
        CASE 6:
            IF n3mmOr32mm=n3mm pPlaceType6;
            IF n3mmOr32mm=n32mm pPlaceType16;
        CASE 7:
            IF n3mmOr32mm=n3mm pPlaceType7;
```

```
      IF n3mmOr32mm=n32mm pPlaceType17;
    CASE 8:
      IF n3mmOr32mm=n3mm pPlaceType8;
      IF n3mmOr32mm=n32mm pPlaceType18;
    CASE 9:
      IF n3mmOr32mm=n3mm pPlaceType9;
      IF n3mmOr32mm=n32mm pPlaceType19;
    DEFAULT:
    ENDTEST
    AccSet nAccNoLoad,nRampNoLoad;
    !stop;
      rCurrentPlacePos:= CRobT(\Tool:=tVacGripper\WObj:=wobjCnv);
      rMidPos.rot.q1:=rCurrentPlacePos.rot.q1;
      rMidPos.rot.q2:=rCurrentPlacePos.rot.q2;
      rMidPos.rot.q3:=rCurrentPlacePos.rot.q3;
      rMidPos.rot.q4:=rCurrentPlacePos.rot.q4;
      MoveJ rMidPos, vNoLoad, z200, tVacGripper\WObj:=wobj0;
    !stop;
  ENDPROC

  PROC pPlaceType1()
    !Product type 1
    !450x337
    !6pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>1 RETURN ;
    !1-2
    pPlacing rPlace11,nPickedPanels;
    pPlacing rPlace12,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacing rPlace13,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !5-6
    pPlacing rPlace14,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
  ENDPROC

  PROC pPlaceType2()
    !Product type 2
    !605x337
    !4pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>2 RETURN ;
    !1-2
    pPlacing offs(rPlace21,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacing rPlace21,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
  ENDPROC


  PROC pPlaceType3()
    !Product type 3
    !905x537
    !2pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>3 RETURN ;
    !1:
    pPlacing rPlace31,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !2
    pPlacing rPlace31,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
  ENDPROC

  PROC pPlaceType4()
    !Product type 4
    !450x437
    !4pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>4 RETURN ;
    !1-2
    pPlacing rPlace41,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacing rPlace41,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
  ENDPROC
```

```
PROC pPlaceType5()
    !Product type 5
    !605x437
    !4pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>5 RETURN ;
    !1-2
    pPlacing offs(rPlace51,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    waitRob\inpos;
    !waittime 8;
    pPlacing rPlace51,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType6()
    !Product type 6
    !905x437
    !2pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>6 RETURN ;
    !1
    pPlacing rPlace61,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType7()
    !Product type 7
    !450x537
    !4pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>7 RETURN ;
    !1-2
    pPlacing rPlace71,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacing rPlace71,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType8()
    !Product type 8
    !605x537
    !4pcs to pick for one layer
    !Stop;
    !To be sure..
    IF ProdData.nProdType<>8 RETURN ;
    !1-2
    pPlacing rPlace81,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacing rPlace81,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType9()
    !Product type 9
    !905x337
    !3pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>9 RETURN ;
    !1
    pPlacing rPlace91,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !2
    pPlacing rPlace91,nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType11()
    !Product type 11
    !450x337
    !6pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>1 RETURN ;
    !1-2
    pPlacingThick offs(rPlace11,0,0,0),nPickedPanels;
    pPlacingThick offs(rPlace12,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
```

```
    !3-4
    pPlacingThick offs(rPlace13,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !5-6
    pPlacingThick offs(rPlace14,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType12()
    !Product type 12
    !605x337
    !4pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>2 RETURN ;
    !1-2
    pPlacingThick offs(rPlace21,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacingThick offs(rPlace21,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType13()
    !Product type 13
    !905x537
    !2pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>3 RETURN ;
    !1:
    pPlacingThick offs(rPlace31,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !2
    pPlacingThick offs(rPlace31,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType14()
    !Product type 14
    !450x437
    !4pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>4 RETURN ;

    !1-2
    pPlacingThick offs(rPlace41,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacingThick offs(rPlace41,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType15()
    !Product type 15
    !605x437
    !4pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>5 RETURN ;

    !1-2
    pPlacingThick offs(rPlace51,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacingThick offs(rPlace51,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType16()
    !Product type 16
    !905x437
    !2pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>6 RETURN ;
    !1
    pPlacingThick offs(rPlace61,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !2
    !pPlacingThick offs(rPlace61,0,0,30),nPickedPanels;
    !pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC
```

```
PROC pPlaceType17()
    !Product type 17
    !450x537
    !4pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>7 RETURN ;
    !1-2
    pPlacingThick offs(rPlace71,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacingThick offs(rPlace71,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType18()
    !Product type 18
    !605x537
    !4pcs to pick for one layer
    Stop;
    !To be sure..
    IF ProdData.nProdType<>8 RETURN ;
    !1-2
    pPlacingThick offs(rPlace81,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !3-4
    pPlacingThick offs(rPlace81,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlaceType19()
    !Product type 19
    !905x337
    !2pcs to pick for one layer
    !To be sure..
    IF ProdData.nProdType<>9 RETURN ;
    !1
    pPlacingThick offs(rPlace91,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !2
    pPlacingThick offs(rPlace91,0,0,0),nPickedPanels;
    pCnvAreaClear Yes\productPlaced;
    !Stop;
ENDPROC

PROC pPlacing(robtarget rPlacePosx,INOUT num nPanelsPlaced)
    VAR num nCurrentGripperSection;
    nCurrentGripperSection:=nPickedPanels;
    IF nPickedPanels=ProdData.nGripSections THEN
        !stop;
        MoveJ Offs(rPlacePosx,0,0,500),vCarry,z40,tVacGripper\WObj:=wobjCnv;
        nMultipleLayerGuard:=0;
    ENDIF
    if Presse_DI01_FreeForLoad = 0 MoveL Offs(rPlacePosx,0,0,60),vCarry,fine,tVacGripper\WObj:=wobjCnv;
    WaitDI Presse_DI01_FreeForLoad,1;
    !stop;
    pCnvAreaClear No;
    MoveL rPlacePosx,vCarry,fine,tVacGripper\WObj:=wobjCnv;
    nEnFallenPartsRegrip:=0;
    nVacGuard:=0;
    pGripper grActiveConf,nCurrentGripperSection,nRelease;
    MoveL Offs(rPlacePosx,0,0,60),vCarry,z10,tVacGripper\WObj:=wobjCnv;
    !Stop;
    nPickedPanels:=nPickedPanels-1;
    !Stop;
ENDPROC

PROC pPlacingThick(robtarget rPlacePosx,INOUT num nPanelsPlaced)
    VAR num nCurrentGripperSection;
    nCurrentGripperSection:=nPickedPanels;
    IF nPickedPanels=ProdData.nGripSections THEN
        !stop;
        MoveL Offs(rPlacePosx,0,0,500), vCarry, z200, tVacGripper\WObj:=wobjCnv;
        nMultipleLayerGuard:=0;
        !StartMove;
    ENDIF
    MoveL Offs(rPlacePosx,0,0,50),vCarry,z20,tVacGripper\WObj:=wobjCnv;
    WaitDI Presse_DI01_FreeForLoad,1;
    MoveL Offs(rPlacePosx,0,0,-15),vCarry,fine,tVacGripper\WObj:=wobjCnv;
    pCnvAreaClear No;
    !MoveL rPlacePosx,vSlow,fine,tVacGripper\WObj:=wobjCnv;
    nEnFallenPartsRegrip:=0;
    nVacGuard:=0;
    pGripper grActiveConf,nCurrentGripperSection,nRelease;
```

```
        MoveL Offs(rPlacePosx,0,0,30),vCarry,z10,tVacGripper\WObj:=wobjCnv;
        !Stop;
        nPickedPanels:=nPickedPanels-1;
        !Stop;
    ENDPROC

    PROC pCnvAreaClear(num YesNo\switch productPlaced)
        SetDO Presse_DO02_ConveyorAreaClear,YesNo;
        IF Present(productPlaced) THEN
            PulseDO\PLength:=0.35,Presse_DO01_PanelsPlaced;
        ENDIF
    ENDPROC
ENDMODULE
```