PV module Laminator

Ray Blokker

Hanze University of Applied Sciences
Mechanical Engineering

2021
May

Abstract

This report describes how a PV module laminator for high-end flexible PV modules can be made from an existing machine base. In this report, the lamination cycle is explained and lamination tests are performed from which the requirements are set up. This project started with an existing machine base upon which improvements and changes were made, the membrane has been mounted to the lid and the actuator of the lid has been improved to accommodate the added weight of the membrane. To control the temperature at which the lamination takes place a temperature controller is selected and a heating system is designed and manufactured. A pneumatic system is proposed, tested and manufactured that has a precise and repeatable lamination pressure control by the use of an Electronic Pressure Regulator. An electronic system is designed to accommodate the pneumatic system which includes a PLC for which software is written that controls the machine. These systems have been combined onto the existing machine base after which the functionality of the machine was tested. To aid the reliability of the lamination process an improvement to the mounting of the membrane was made after which the machine was successfully commissioned to be used in production.

Declaration by author

I hereby declare that I have written this thesis independently, without outside help, and that it is entirely the result of my own work.
I have only used the sources which have been faithfully and accurately quoted in the reference list, including books, magazines, handouts, internet sites and significant personal communication, and I have marked the pages taken either directly or as regards content from those sources.
The thesis was in this or similar style never shown to another board of examiners nor was it published.

I am aware that not complying with these requirements will result in the marking of the thesis to be declared invalid and that, if plagiarism is detected, disciplinary measures will be taken.

Name: Ray Blokker Signature:

Place/date/year: Groningen, 31 may 2021

A thesis submitted for the degree of Mechanical Engineering at the Hanze University of Applied Sciences.

Internship company: Mito Solar
Company supervisor: Jules van Haaren
Company electrical engineer: Danny Bokma

First assessor: Arjen de Ruijter
Second assessor: Hette Akkerman

Written by: Ray Blokker
Student number: 349281

Glossary

EPR Electro Pneumatic Pressure Regulator

EVA Ethylene Vinyl Acetate

HMI Human Machine interface

IR Infrared

NHL Noordelijke Hogeschool Leeuwarden

PID Proportional Integral Derivative

PLC Programable Logic Controller

PV Photovoltaic

RTD Resistance Temperature Detector

TPU Thermoplastic Polyurethane

Contents

1 Introduction **6**

2 Analysis **7**
- 2.1 Solar panel lamination process . . . 7
- 2.2 Testing with other machines . . . 7
- 2.3 Machine base . . . 10
- 2.4 Scope of work . . . 12
  - 2.4.1 Requirements . . . 12
  - 2.4.2 Planning . . . 13

3 Heating system **15**
- 3.1 Heating elements and sensors . . . 15
- 3.2 Control systems . . . 16
- 3.3 Power distribution . . . 16
- 3.4 Heating system fabrication work . . . 17

4 Mechanical system **19**
- 4.1 Lid actuator . . . 19
- 4.2 Membrane tensioner . . . 20
  - 4.2.1 Design options . . . 20
- 4.3 3D design . . . 20
- 4.4 Mounting of external parts . . . 21
- 4.5 Mechanical fabrication work . . . 21

5 Pneumatic system **25**
- 5.1 System overview . . . 25
- 5.2 Pressure regulation principles . . . 25
- 5.3 Testing of individual systems . . . 27
- 5.4 Pneumatic system design . . . 27
- 5.5 Pneumatic fabrication work . . . 28

6 Electronic system **30**
- 6.1 System overview . . . 30
- 6.2 Controller selection . . . 30
- 6.3 Electronic schematic . . . 31
- 6.4 Electronic fabrication work . . . 32
- 6.5 Software . . . 34

7 Testing and Optimisation **36**
- 7.1 Improvements . . . 37

8 Conclusion **40**
- 8.1 Recommendations . . . 40

References **41**

A Cost overview **42**

B PLC code **42**

C Log file **57**

1 Introduction

Mito Solar is a company focused on the small-scale production of custom PV modules for high-end applications. Customers range from student racing teams to yacht builders to space research laboratories. To optimise their production process, Mito wants to bring the laminating process to their in-house facilities.

To realize this, Mito has obtained a second-hand machine that came from an automated PV module production factory. This machine was removed from all of its electronics and automated in and outfeed mechanics and composed basically out of a frame, heat bed and lid. The project described in this report is making this machine base into a fully functional machine that can be used to manually make high-end PV modules. This report answers the following research question: How can a fully functional PV module laminator be made while trying to reuse as many parts as possible and keeping the cost down?

In the first chapter of this report, an analysis is performed on the production process of PV modules, tests are carried out on other machines and a list of requirements is made from these findings. This is followed by an analysis of the existing machine base to make a scope of work.

The following chapters are split up into the different subsystems of the machine. These are each analysed and a solution is proposed to get these into working order and complying with the requirements after which, the fabrication work is executed and described.

The final chapter of this report exhibits the testing and optimisation phase of this machine followed by the conclusion and recommendations.

2 Analysis

To fully understand the functionalities of a photovoltaic Photovoltaic (PV) module laminating machine, it is important to understand the lamination process. The following section displays the research conducted on this topic.

2.1 Solar panel lamination process

PV modules are made up of different layers as shown in figure 1. The PV cells itself, are composed of a thin layer of silicium that must be mechanically and environmentally protected. Therefore, the cells are encapsulated between two substrates, these substrates can be a polymer film or glass layer. The encapsulant is an adhesive bonding all the layers together, a commonly used material for this is Ethylene Vinyl Acetate (EVA) but other materials like Thermoplastic Polyurethane (TPU) or even UV-curable resins are gaining popularity [1].

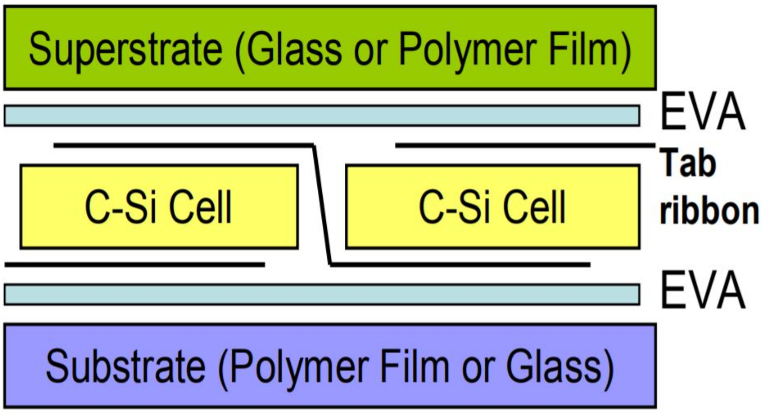

Figure 1: Typical lamination layup of PV modules

The most commonly used encapsulants like EVA and TPU are thermoplastic, these materials soften or melt when heated above a certain temperature and solidify when cooling down. This property is used in the production process of PV modules, a layup is made comprising out of the front substrate, encapsulant, cells and electrical connections, encapsulant and back substrate. This layup is then placed on a heated bed inside of a vacuum chamber that is split up by a membrane. Both chambers are evacuated (pulled to vacuum) to get the air out of the layup, after that the top chamber is purged back with air creating a pressure difference on the membrane that thereby presses all the layers together as is displayed in figure 2. The heat from the heat bed melts the encapsulant in the layup, combining all the materials into one laminate. After this, the pressure in both chambers is brought back to atmospheric pressure and the laminate is taken out to cool, solidifying the encapsulant. Figure 3 shows a graph of the pressure in both chambers during a lamination process.

2.2 Testing with other machines

To get a comprehensive feeling of the lamination process and to see where improvements could be made, a visit to the Noordelijke Hogeschool Leeuwarden (NHL) was made by Jules, Danny and Ray to use their PV module laminator machine. There, a small production run for a Mito Solar customer was performed, during which Ray analysed the machine. Besides that, a few test samples were made to use in later analysis.

This machine is from 1997 and the control is done by manually setting timers and restriction valves, adjusting the lamination pressure is not possible. Ray mainly focused on analysing the pneumatic system, as that system was the most unfamiliar. The mounting of the membrane in

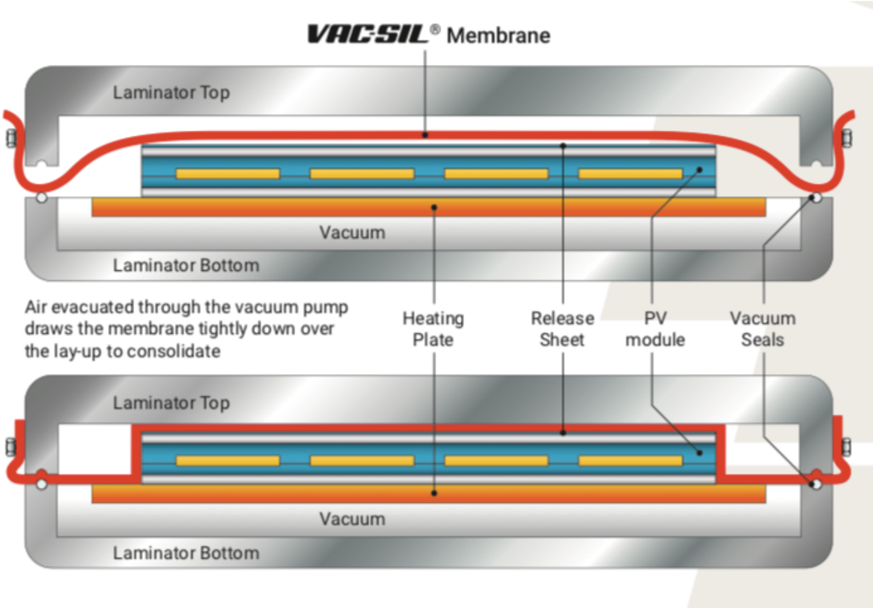

Figure 2: Cross section of a laminator in open state and during lamination

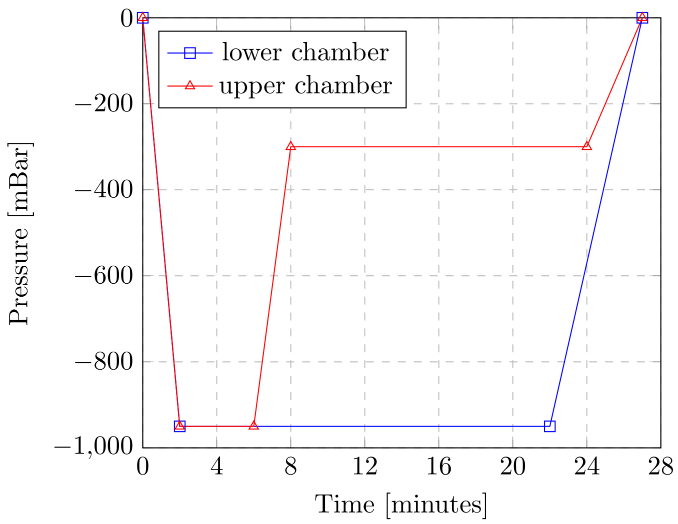

Figure 3: Typical lamination pressure cycle

this machine was part of the lid seal and placed inside of the lid, this would be very hard to retrofit to the existing machine that Mito Solar has and is therefore not further considered.

The pneumatic system, having no ramp time control or lamination pressure control, was simple. One of the interesting parts was that the lower chamber had a relatively large bleed valve in comparison to all the other valves (38mm vs. 10mm) this is most likely to improve the speed of the purging step in the lamination process (purging the lower chamber back to atmospheric pressure). The problem with working with vacuum is that: air expands, the same size hose or valve has considerably less mass flow at one bar below atmospheric pressure compared to one bar above atmospheric pressure. Due to the low-pressure difference between vacuum and atmospheric pressure, large-diameter tubing and valves are needed. This becomes more apparent when bleeding a large volume back to atmospheric pressure, the pressure difference then gradually reduces to zero thereby also reducing the average flow rate. Another solution to this would be to use compressed air to supply the bleed valve, this would increase the pressure difference and thereby lower the pressure variation during bleeding, thus improving the consistency of the ramp speed and lowering the size of the valves needed.

The biggest downside of this machine is that it cannot control the lamination pressure, this means that after evacuation the pressure in the upper chamber is purged to atmospheric pressure. Therefore the membrane is pressed onto the laminate with the pressure difference between the atmospheric pressure in the upper chamber and the vacuum in the lower chamber. The vacuum in the lower chamber is as high as the pump can reach, this is favourable to remove all the air from the laminate. However, pressing onto the laminate, with close to 1000mBar of pressure, can cause cracks in the PV cells, especially with flexible (non-glass) front sheets.

To be able to see if the PV cells have cracks, electroluminescence pictures are made. To make these pictures, the sample that is to be tested is placed in a dark room and connected to a power supply. The power supply is used to supply the PV module with a controlled current. In this state the PV cells emits Infrared (IR) light. If a picture is made with a camera that is sensitive to this light the functional areas show up as light coloured, the non-working areas or cracks show up as dark. Mito Solar has a test setup for this comprising of a DSLR camera without the IR filter, when a long exposure picture of a PV module is made in darkness it shows the IR light in purple colour.

The PV test samples that were laminated at the NHL while analysing there machine, were used to make electroluminescence pictures. Figure 4 displays one of these modules with cracks. This is caused by a locally higher force on the laminate because of the slight variation in thickness where the connections are made. To minimise this problem the pressure at which the membrane presses onto the laminate should be reduced. To do so the upper chamber must be kept at a controlled vacuum.

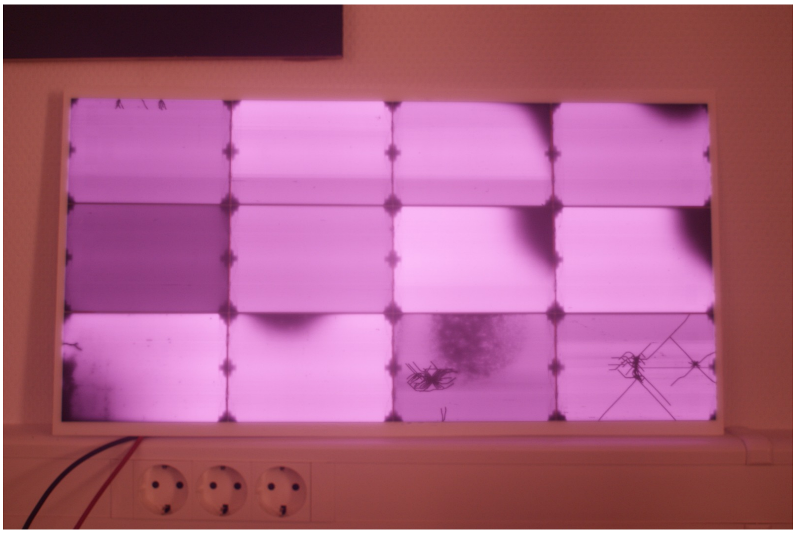

Figure 4: Electroluminescence picture of a module with cracks due to too high lamination pressure

After the test at the NHL, another production session was performed at the test facilities of Eurotron. Eurotron is a large-scale production automatization manufacturer making production machines for back-contact PV module manufacturers. These are state of the art machines that can make up to 90 modules an hour with only 4 operators. Their laminator was used to laminate a series of PV modules. The pneumatic system of this machine was significantly more advanced than the machine at the NHL, the pressure in the upper chamber during lamination was controlled by opening and closing a series of smaller valves either bleeding compressed air into the chamber or evacuating air out of the chamber by the vacuum pump. A notable aspect was the constant switching of these valves to keep the pressure constant during the lamination process.

The mounting of the membrane on this machine was done externally as opposed to the machine at the NHL. This is something that can be adapted to the existing machine that Mito Solar has and gave good insights into how the clamping and tensioning can be done. Figure 5 displays one side of the clamping and tensioning mechanism on this machine.

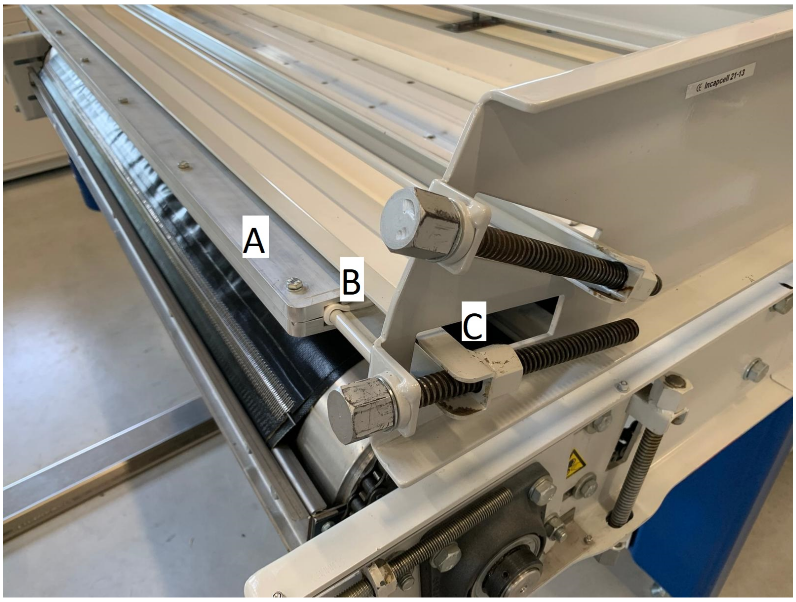

Figure 5: Membrane clamping and tensioning system of the Eurotron laminator, A membrane clamp, B membrane (white), C tensioner

2.3 Machine base

To get a good understanding of what work had to be done, it was first important to understand what was already there. The machine that Mito Solar has, is a machine that has been used as a development machine for a production facility. It was functional before but was in an incomplete state. The machine came with a partial pneumatic and electronic system, but this was incomplete and not working so it was removed. Most of the components were however still there. This machine used to be part of a production line, infeed and outfeed were done automatically and the membrane was part of the layup and not fixed to the machine. This is very inconvenient for the manual production that Mito Solar does, so this was one of the parts that Mito Solar wanted to be improved.

The machine came without documentation. However, there were some schematics and valves drawn onto the back of the machine, it also came with a console computer that operated as Human Machine interface (HMI) which was installed with Codesys and contained the software that was used to control the machine.

This software was analysed to understand the control system behind the laminating process. It quickly became apparent that this software was still work in progress and that a lot of things were unfinished or were commented with: "To Do". It did, however, give a better idea of the control of the valves: to regulate the pressure, vacuum- and bleed valves were switched on or off. This is similar to the system Eurotron uses on their machine. It did, appear that this had some complications which could also be seen back in the pictures of the original pneumatic system. Programs, codes and valves were added to get the system stable. The schematics on the drawings on the back of the machine (figure 6) are relatively simple, with a vacuum and a bleed valve on both chambers. There were however, more valves added, with the purpose of getting both a big and a small vacuum valve, and valves connected to compressed air to bleed faster. This is probably due to the inconsistency of the flow in the valves as these are dependent on the pressure drop.

These issues also became apparent in the software, there was a simple control system for the vacuum and bleed valves to control the pressure but then there were states added to step in if the process value differed too much from the set value. The hysteresis for these processes was set to 20mBar, and 100mBar for the quick responses, this gave the impression that this control style or the way it was implemented was not very precise or constant. The changes that Mito Solar wants to make to the recipe are in the region of 50 to 100mBar, a variation in the process of the same magnitude might cause problems and inconsistencies during production so a precision of at least

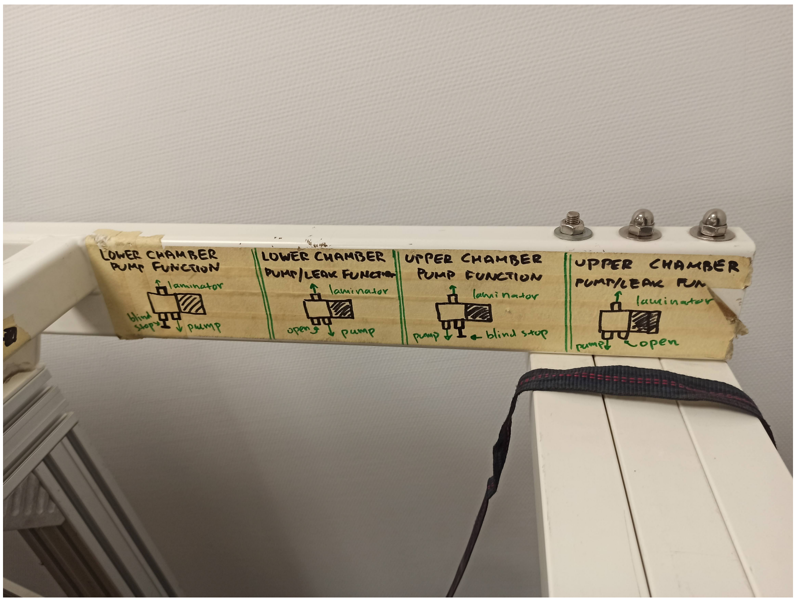

Figure 6: Drawings on the back of the machine explaining how the old pneumatic system worked

20mBar is needed for repeatable results.

Mechanically, the machine was in a decent and complete state. There where some small repairs and improvements to make but no major changes other than adding the membrane mounting and reinforcing the actuator mount. Figure 7 shows the state of the machine when work was started on it. On the back side of the machine the pneumatic actuator that opens and closes the lid is mounted, the mounting of this actuator has no reinforcements perpendicular to the direction that the forces are applied to, because of this the mounting bends a lot during actuation. Given the fact that the membrane will be mounted to the lid, the forces applied to this mounting will become bigger and this mounting will probably not suffice.

Figure 7: PV module laminator when Mito Solar obtained it, the red arrow points towards the actuator mounting

The wires of the heat bed were hanging underneath the machine, these were all labelled and there was a drawing of the numbering of the heat elements and the temperature sensors and their placing on the bed. There was however, no information on what type of sensors and heating elements were used or what their supply voltage was.

When Mito Solar bought the machine, there where large plates attached to the sides of the machine with all the electronics and pneumatics, this was all removed and boxed before this project was started. Other then pictures and notes written on tape there were no clear schematics of this setup. These components were analysed to see which could be reused.

2.4 Scope of work

The initial requirements where simple: make a fully functional PV module laminator while trying to reuse as many parts as possible and keeping the cost down. An additional requirement was to make sure that the membrane is fixed to the machine and that there is good and precise control over the lamination pressure. Based on this, the scope of work can be found in the following section.

After the tests that were done with the other machines, there was a good impression of what was needed to get a fully functional machine. Analysing the existing machine base had given a good idea of what there was and what needed to be added or modified. To make the work manageable, it was decided to split the project into different phases: heating system, mechanical system, pneumatic system and electronic system. After these subsystems are done, there is a testing and optimisation phase.

For the heating system, the hardest part was already done: the machine already had a heated bed, heating elements and temperature sensors and the wiring feedthrough into the vacuum system. It was expected that the heating capacity should be sufficient as it was previously designed for this application. If during testing, it was found that this was not the case, changes would be made. To get this functional, the first step would be to work out what heating elements and temperature sensors were installed, how these need to be connected and switched and how the temperature should be managed as there where no schematics or information available of how this was previously connected. Mito Solar does have the requirement that it needs to be easy to change the temperature setpoint. Furthermore, the heat bed needs to be able to reach at least 160°C to be able to use all the available encapsulants, and it needs to heat up within a reasonable time (<30min). If possible the machine needs to be able to work on one 3Phase 400V 16A outlet so the existing infrastructure in Mito Solar's workplace can be reused.

The mechanical system needs the addition of the membrane mounting, this will be an extra component that needs to be added to the lid. The way of mounting this to the lid and how the membrane will be secured and tensioned needs to be thought out, designed and made. The addition to the lid will lead to extra weight which will most likely put too much strain on the lid actuation mechanism, which will need to be improved. In addition to that, Mito has the requirement that all components need to be mounted to the machine frame to be able to move the machine easily. Therefore, a mounting for the compressor and vacuum pump will have to be made.

The pneumatic system phase started with desk research, there, indepth research was performed on this topic to get a proper understanding of all the possibilities and limitations. Following this, a more in-depth plan will be made which will likely include experiments to understand the behaviour of certain processes. If the design of the system is complete, all the missing components will be ordered, and the system will be built and tested.

The electronic system will also start with desk research. Decisions will have to be made on what kind of Programable Logic Controller (PLC) will be used and a complete schematic will be designed. This will then be made into an enclosure with a human-machine interface (HMI) and will be connected to the different components. Parallel to that the PLC software that controls the machine will be written, this will be made step by step to be able to test all the individual components and processes.

2.4.1 Requirements

Based on the analysis and the wishes of Mito Solar the following list of requirements is made, these requirements have been discussed with Mito Solar and are approved.

- Able to laminate flexible PV modules with both TPU and EVA encapsulant

- Easy to use and adjust settings from a single interface
- Lamination pressure control adjustable from 50mBar to 950mBar
- Lamination pressure stable with less than 20mBar deviation
- Pressure ramp control for evacuation, lamination and purging step
- Pressure control variation within 20mBar
- Datalogging to keep track of the lamination process
- Membrane mounted to the lid for ease of use
- Reuse parts if possible and keep the cost below €2500
- PLC software that can be easily modified for later changes or additions
- Self-contained machine with all the parts mounted to the frame so it can easily be moved
- Powered by one 3Phase 400V 16A outlet
- Can reach temperature of at least 160°C within 30 minutes

2.4.2 Planning

To keep track of the progress and to set milestones, a general planning (figure 8) is made. In this planning, the first month is reserved for analysis and research. In this phase all the preparations are done to be able to start the execution phase with all the necessary knowledge. In the execution phase, all the systems are designed, made, assembled and tested. The last step in the electronic system phase is writing the software. During this step the testing and optimisation phase will start as writing the software will be a iterative process based on how the machine performs. When the software is done the machine will be extensively tested and lamination recipe's will be made. Any improvements that are needed will also be made in this phase.

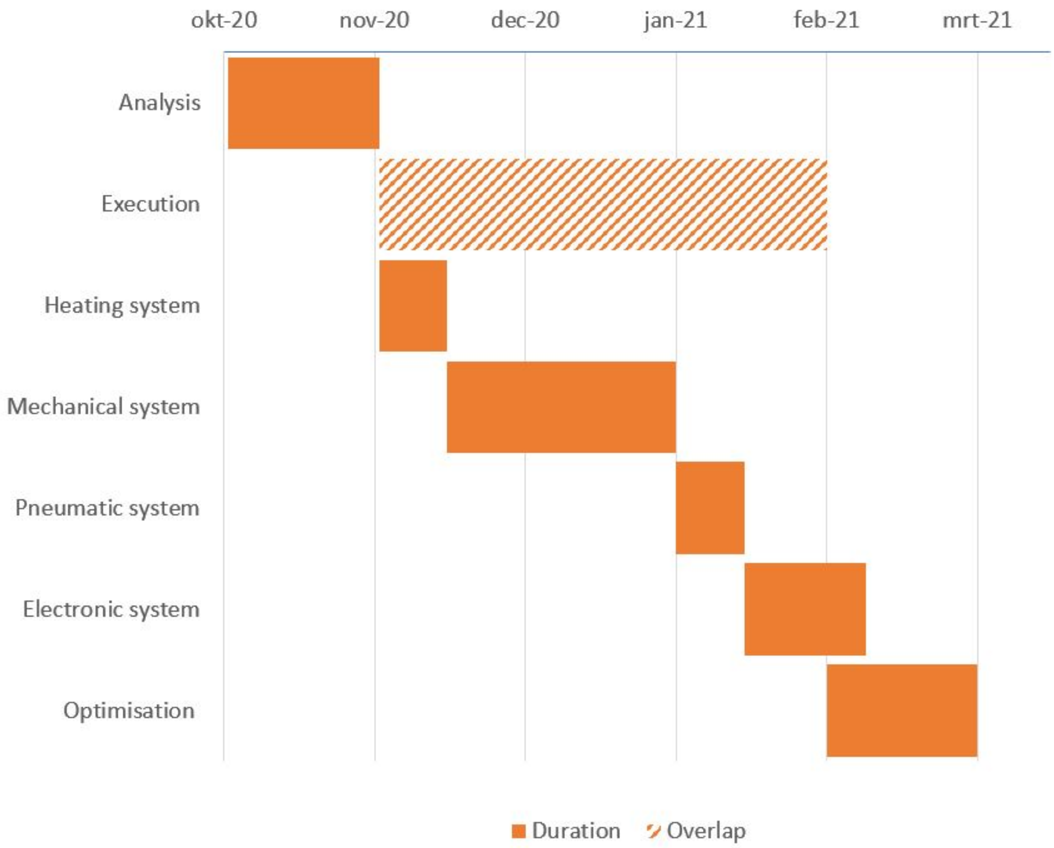

Figure 8: General project planning

3 Heating system

This chapter describes the process of designing and producing a functional heating system that complies with the requirements.

3.1 Heating elements and sensors

The machine was already equipped with heating elements and sensors, however it was not known which type these were and how they had to be connected. This was due to the sensors being mounted under the heat bed, where, they were inaccessible without disassembling the machine and possibly compromising the seals of the wire feed-throughs. Therefore, it was not possible to look for a mark or code to check which type of components were used. To find out which heating elements were used, the first step was to measure the resistance between all the connections and to reverse engineer the system based on those measurements.

The heat bed is split up into an inner and outer ring that are both divided into 4 sections resulting in 8 zones as shown in figure 9. This information came from a handmade drawing that was taped to the machine by the previous owners. Given that the solid-state relays that were previously mounted to the machine were rated to a maximum of 265V A,C it was unlikely that the heat elements were connected between phases (400V). It was assumed that they were connected between the phase and neutral wire (230V). The measured resistance for the inner heating elements was measured to be 43Ω and the outer heating elements were measured to be 77.4Ω, at a supply voltage of 230V, this gives a heating capacity of respectively 1230W and 683W per element. All the elements together have a combined heating capacity of 7.65kW.

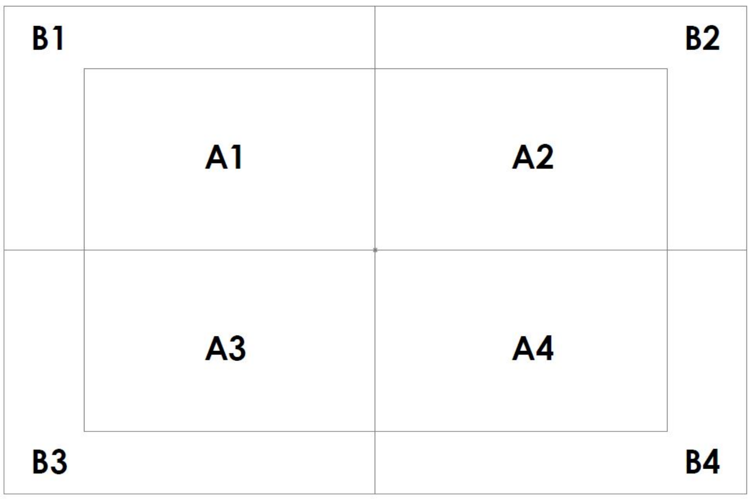

Figure 9: Zone division of the heat bed

Equally to the heating elements, the temperature sensors were also inaccessible without disassembling the machine. Therefore, another method was needed to check which type of sensors they were. To find out which sensors were used an analysis was performed as described in the following section.

The temperature sensors are a two-wire type. The wires are regular copper wires, so a thermocouple type sensor is ruled out. The resistance measured to be about 106Ω, therefore it was probably a Resistance Temperature Detector (RTD). Most RTD's are rated at a temperature of 0°C, a PT100 which is one of the most common used RTD's has a 100Ω resistance at 0°C, at 15°C this is about 105Ω. To check if this sensor was indeed a PT100, one of the heating elements was connected to a "schuko" plug and connected to a wall outlet, while the heating zone heated up, the resistance of the temperature sensor increased verifying that this sensor has a positive temperature coefficient. After a few minutes, the heating element was disconnected and the temperature was measured at 32°C. The resistance read about 112Ω verifying that this was indeed a PT100 type sensor.

3.2 Control systems

For a reliable and consistent lamination process, a consistent and precise heat bed temperature is needed. In addition to that there are further requirements set like the requirement that the temperature should be easily adjusted. In this section, the process of choosing the best system to achieve these requirements is described and the design of a system is proposed.

There are numerous ways to control the temperature of a device, therefore, design choices will have to be made. The most common heat temperature regulation principle is an on-off type controller that can be found in many house heating systems. This is an error based controller that turns on when the difference between the setpoint and the actual temperature (error) becomes negative under a certain hysteresis and turns off when the error is positive above the hysteresis. The hysteresis prevents the system from keeping turning on and off quickly after each other.

Another temperature regulation principle that is commonly used is the Proportional Integral Derivative (PID) type. This control loop mechanism employs a feedback system that does not only look at the error but also at the rate of change and the error duration. Therefore, it can prevent or reduce overshoot and reaches a more stable steady state.

Of both types there are off the shelf solutions available, another option is to include the temperature control inside of the PLC software that controls the rest of the machine, both control strategies could be implemented in this way but it will have to be programmed and set up accordingly. Table 1 presents the design choice which has been made using the Kesselring method that rates each aspect of these products to get to an objective result.

Aspect	On/Off controller	PID controller	On/Off in PLC	PID in PLC
Easy to adjust	3	3	5	5
Precise	1	5	1	5
Availability	3	4	2	2
Implementation	3	4	2	2
Programming	5	4	2	1
Cost	5	3	3	3
Total:	20	23	15	18

Table 1: Design choice of the temperature controller using the Kesselring method

The possible solutions have each been rated to several aspects which they need to conform to. These aspects are based on the requirements that are set in the first chapter and to aid practicality. Adjustability, preciseness and cost are requirements set by Mito, availability and implementation are not directly related to the requirements but are equally important to get a working machine. Programming is also not related to the requirements but comes from the engineering capacity available to this project.

Overall the stand-alone PID controller scores the highest, it scores lower in the "easy to adjust" aspect but there might be an option to connect this controller to the PLC via Modbus so that the setpoint can still be set from the same HMI. Given that there are 8 heat zones an 8 channel PID controller is preferred to aid simplicity. Based on availability and pricing the CJZ-8800 temperature controller is chosen, this is an 8 channel temperature controller with Modbus communication and auto-tuning. Therefore, it is possible to connect the temperature controller to the PLC to improve the adjustability and the tuning of the PID parameters can be set up automatically which will greatly improve implementation effort.

3.3 Power distribution

One of the requirements set in the previous chapter is that the whole system must be able to be powered by one 3Phase 400V 16A outlet. The division of the load over the phases is of great importance, given that the heat bed has a heating capacity of 7.65kW and that other components like the vacuum pump and the compressor also have a rather high power demand.

The compressor and vacuum pump were already supplied with the machine basis, they have a power rating of 600W on one phase and 1.5kW on 3 phases respectively. The inner zones of the heat, bed as shown in figure 9, have a power rating of 1.23kW each, the outer zones have a power rating of 683W each, the heating elements are each connected to one phase. Furthermore, there is an auxiliary power draw of the other components like the PLC, PID controller, human interface etc. which is expected to be less than 300W in total. There are three phases available with a maximum power rating of 3.6kW each. Table 2 shows the division that was made to ensure that the load would stay within the limits, the highest power draw is on L2 which is close to the limit of 3600W. However, having all these loads simultaneously is unlikely, as not all the heat elements will be on if the operating temperature is reached and the compressor will not be needed during warmup. If the full load is drawn, then there will still be some margin, as the circuit breaker fitted before the outlet is of the C type, this type of circuit breaker has a slow tripping mechanism capable of higher inrush currents and longer overloads compared to normal B type circuit breakers.

	Vacuum	Compressor	Aux	A1+A3	A2+A4	B1+B2+B3+B4	Total
L1	500W		300W	2460W			3260W
L2	500W	600W			2460W		3560W
L3	500W					2730W	3230W

Table 2: Load division over the three phases

3.4 Heating system fabrication work

The heating system was the first part on which fabrication was executed. It was mandatory to keep sufficient space on the machine for wiring, hoses and inside of the enclosure for the rest of the electronics, as a lot of the details of the other systems were still unknown. The solid-state relay's, power supply and circuit breaker were reused from the parts that were leftover from the disassembly of the machine.

The first step was to make a schematic of the wiring and to lay out the components to create an overview of the placement and which other components were needed. Figure 10 displays the schematic, this is based on the datasheet of the chosen temperature controller with the adaptations to this specific application and the zone division over the phases. The layout of the control box is based on this schematic. This has first been drawn true-scale on a piece of paper after which it was transferred to the backplate that mounts all the electronics. Figure 11 displays the control box with the electronics mounted to the backplate and the wiring of the temperature controller, the leftover space has been reserved for the rest of the electronics that control the machine which will be engineered in a later stadium. The top of the control box is used as the human-machine interface, on this surface, the buttons, indicators, and the temperature controller are mounted.

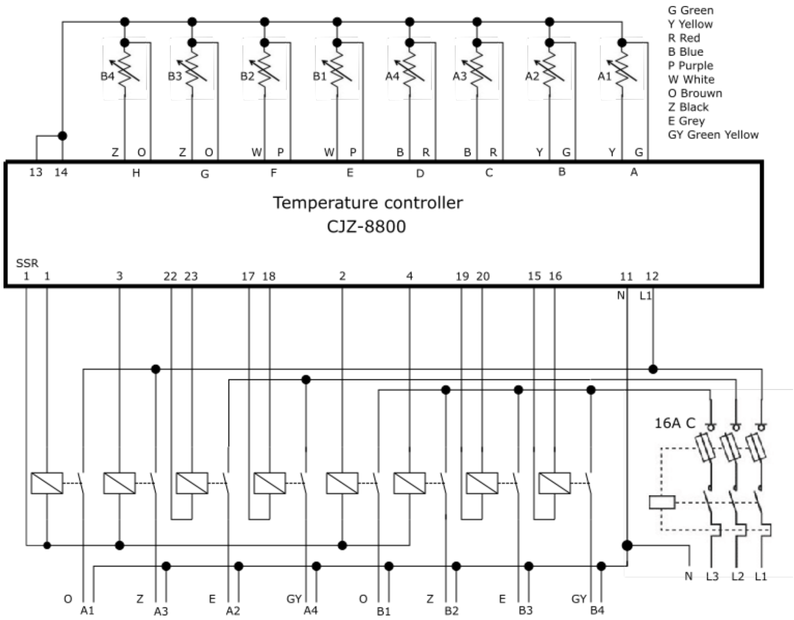

Figure 10: Schematic of the temperature controller

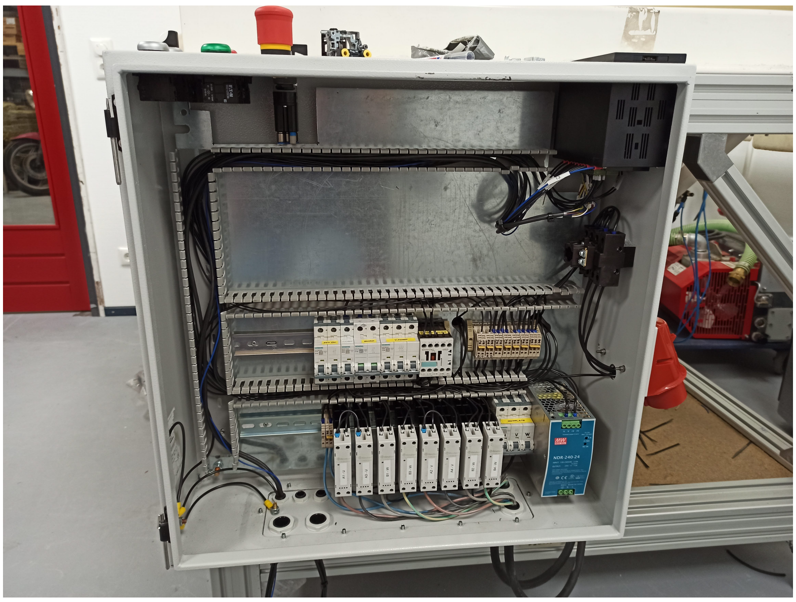

Figure 11: Control box with all the temperature controller electronics mounted and wired after the fabrication work on the heating system was done

4 Mechanical system

During the analysis phase, the existing machine base was analysed and it became apparent that the mechanical system needed improvements. Furthermore, the machine was not equipped with a provision to mount the membrane as this was likely part of the external system before. This chapter describes the work that has been done to the mechanical system starting with the lid actuator after which a membrane mounting and tensioning system is designed and fabricated.

4.1 Lid actuator

When analysing the machine it quickly became apparent that the lid actuator was a weak point of the machine. There was an extra mount fitted to the lid which was placed further away from the hinge point than the original one to get a higher actuation force as shown in figure 12. This also resulted in a torsion force on the actuator arm which made the system very prone to bending. Given that the lid would be equipped with a membrane mounting, the load on this actuator would be increased and the current system would not suffice.

Figure 12: Old actuator mounting point and later added, off-axis placed mounting point

The lid actuator comprises a pneumatic cylinder mounted to a horizontal beam that connects to the counterweights arms. The bottom mounting point was fitted to a triangulated bracket that only had stiffness in the direction perpendicular to the load as shown in 7. The pneumatic cylinder needed 6 bar pressure to be able to open the lid. As the weight of the membrane tensioner was unknown at the time there was no exact value for how much stronger the actuator needed to be. Therefore it was decided that an extended arm running from the old mounting point to the counterweights with multiple holes to fit the cylinder closer or further away from the hinge point would be the best solution. This would give a lot of flexibility and strength, especially combined with a bigger cylinder with a longer stroke. This also resolved the torsion force on the actuator arm which was now connected to the counterweights and spread the load over both sides. Figure 13 shows the new mounting arm design incorporating multiple actuation points and the Mito Logo; the bottom mount is also shown.

The bottom mount was stiffened by placing a frame profile section from the top to the bottom of the frame. With this setup the torsion force that was previously exited onto the frame was resolved as the load was now distributed between two points, greatly improving the stiffness. A new bracket was designed and fabricated to mount the actuator to the profile.

To compensate for the loss in opening angle by moving the actuation point further away from the hinge point, an actuator with a longer stroke was needed. The price difference between bore sizes of cylinders is minimal so one step bigger was chosen to lower the pressure that was

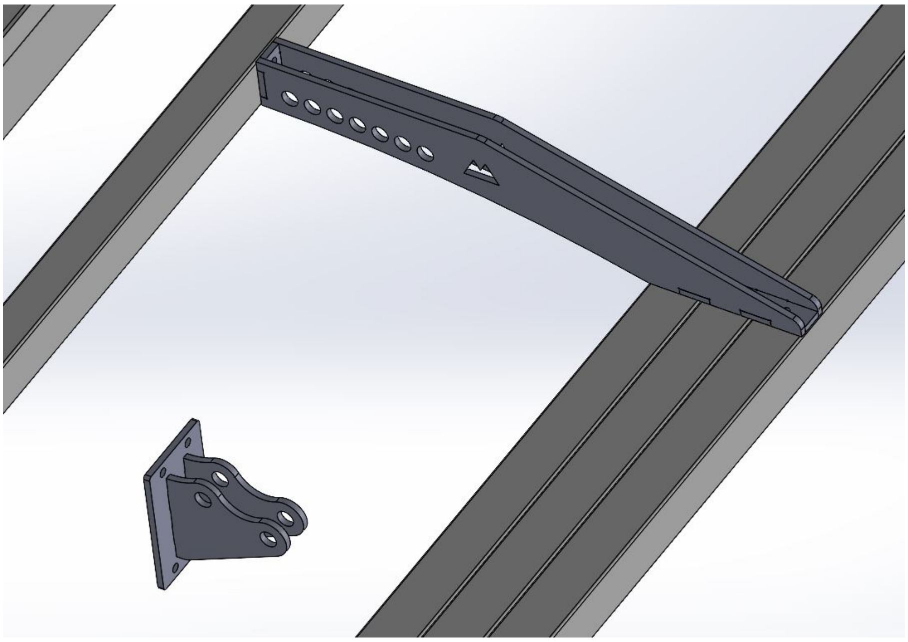

Figure 13: New top and bottom mounting arm design incorporating multiple actuation points to be able to adjust for the added weight of the membrane tensioner

needed to actuate the lid. Doing so made sure that the lid would still open even if the pressure in the compressor was low. This also made sure that the actuation force and speed remained constant as the pressure regulator could be set to a pressure below the tank pressure during a normal laminating cycle. In Equation 1 the new set pressure is calculated after switching from the original 50mm bore cylinder to the new 63mm bore cylinder.

$$6bar * \frac{50^2}{63^2} = 3.78bar \tag{1}$$

4.2 Membrane tensioner

One of the requirements that Mito set was that the machine should be equipped with a membrane mounting system. Without that the membrane has to be manually placed on top of the laminate every time the machine is used. During the analysis phase, the membrane mounting system on other machines was examined to get ideas of how this system can be added to the existing machine.

4.2.1 Design options

On the NHL machine, the membrane mounting is incorporated inside the lid. This is a very compact way of mounting but is also inherently incorporated in the design of the lid from an early stage on. Retrofitting this to the existing lid would be a very complex and expensive task, hence why this option was ruled out during the design phase.

The laminator at Eurotron has an external membrane mounting with an incorporated tensioning system as shown in figure 5. This design is very adaptable and can be incorporated onto the current lid design. The membrane itself is clamped around a steel profile which has a threaded nut on both sides that can be used for tensioning.

4.3 3D design

Some sketches and designs of variations of the Eurotron system were made to get an idea of how this can be mounted to the existing machine. Another consideration was to the direction in which the system was mounted to minimise the footprint of the machine and how the system could be mounted to the machine without compromising the airtightness of the lid or having to remove the glass.

The clamp was redesigned so that the tensioners could be mounted on top of the lid instead of on the sides of the machine as displayed in figure 14. This allowed a rather simple frame to go over the glass plate off the lid to mount the tensioners to. A trapezium threaded rod, welded to the strip that supports the clamp, is used to pull the strip up and thereby tensioning the membrane.

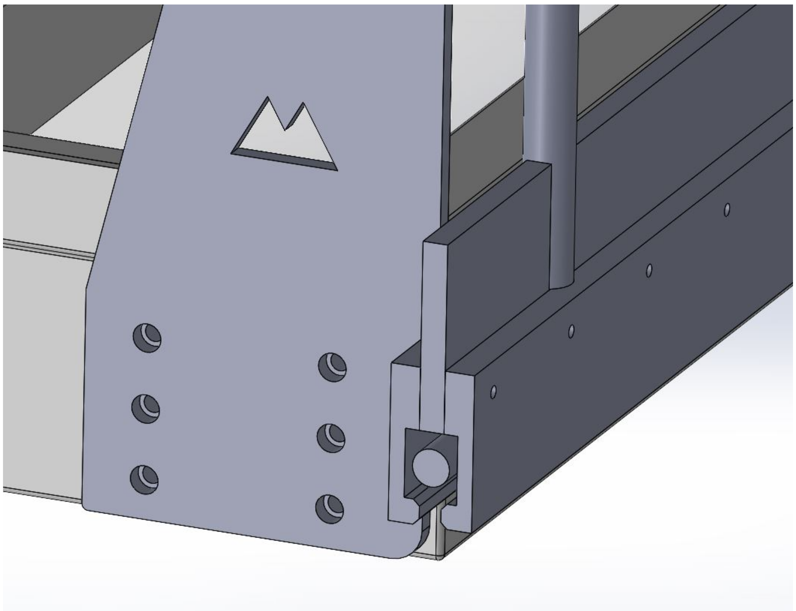

Figure 14: Membrane clamping system with the trapezium threaded rod ascending to the frame

To mount the frame to the existing lid a, thick plate (displayed in figure 15) with threaded holes is welded to the frame of the lid. this prevents the need of making holes and potentially creating leaks. To this plate, a flat riser is bolted that gives the frame upwards and sideways strength. On the backside of the frame a bracket, consisting out of two perpendicular plates welded together, is bolted directly through the frame and the hinge point as shown in figure 16.

4.4 Mounting of external parts

One of the requirements set by Mito was that the machine should be self-contained. This means that all the parts that are needed for the machine should be mounted to the machine. Therefore, the compressor, vacuum pump, pneumatics and electronics all need to be mounted to the machine.

As the frame of the machine is made out of aluminium profile-system sections, it is easy to adapt or add parts to the machine. The machine itself has a large space under the heat bed, it was however decided that this space was best kept empty so it could be used for storage of supplies and (semi)finished products. Therefore, the most logical place to mount those components was behind the machine where there was an open space for the counterweight of the lid. The control box itself was mounted directly to the side of the machine to make this easily accessible as this also functioned as the HMI. To mount the vacuum pump and the compressor 4 profile sections were added in an outward direction on which rubber vibration dampeners were mounted to reduce vibrations transferring into the machine. Figure 17 shows the compressor and vacuum pump mounted to the machine.

4.5 Mechanical fabrication work

Because Mito shares their workspace with two other companies, there was access to a workshop with a lot of machinery. Therefore, a lot of the fabrication work could be done in-house. The in-house capabilities were kept into account during the design phase so only one order of laser-cut

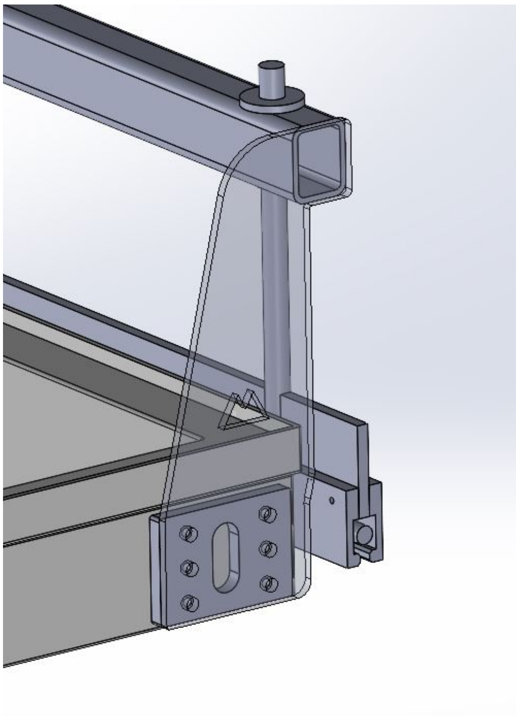

Figure 15: Front side mounting of the tensioner frame with the threaded plate that is to be welded to the lid frame

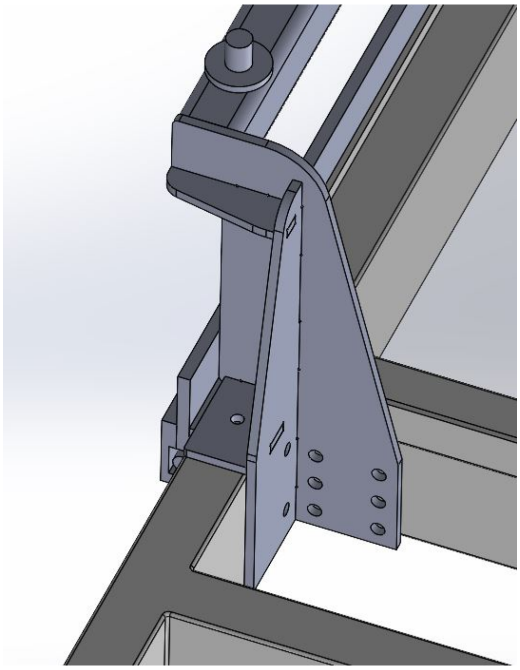

Figure 16: Backside mounting of the tensioner frame consisting out of multiple plates welded together to create stiffness in all directions

Figure 17: Vacuum pump and the compressor mounted to the backside of the machine

plate material had to be outsourced. The rest was based on standard components or was made or adopted in-house.

The laser-cut plate materials were designed with fingers that fitted into slots so that the parts would all be aligned in the right way during assembly. Figure 18 shows the plates assembled and prepared for welding, figure 19 shows the welded back mount. After the mounts were done, the mounting plate with pre-tapped holes was welded to the front of the machine while keeping a wet

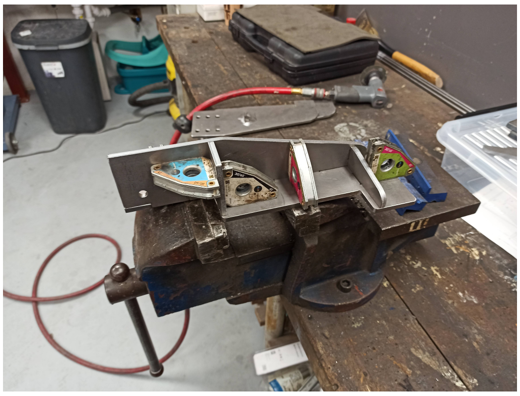

Figure 18: Laser-cut plates of the backside mount assembled and prepared for welding

Figure 19: Backside mount after welding

cloth to the steel to prevent heat transferring to the sealant of the glass plate. On the backside of the machine, the mounting holes were drilled after which the mounts could be bolted to the machine.

With the mounts in place the box sections that connected between both mounts could be cut to the right size after which they were welded to the mounts. By doing this with everything mounted to the machine, it was assured that the alignment was correct and that everything would fit. Figure 20 shows the preparation for welding the mounts to the box section while mounted to the machine.

The clamping system comprised two milled aluminium stips bolted to a steel profile. These aluminium strips were machined in-house on the CNC portal mill. This mill is not intended for milling aluminium and was not equipped with a cooling system. During the first tests it was quickly found that cooling was necessary, so a bath which is shown in figure 21 was made to be able to keep the part under a layer of coolant at all times. After drilling and tapping the holes, the parts were assembled and the membrane was clamped up as shown in figure 22, after which the whole assembly was fitted to the machine as shown in figure 23.

Figure 20: Mounts bolted to the machine in preparation for welding the box section

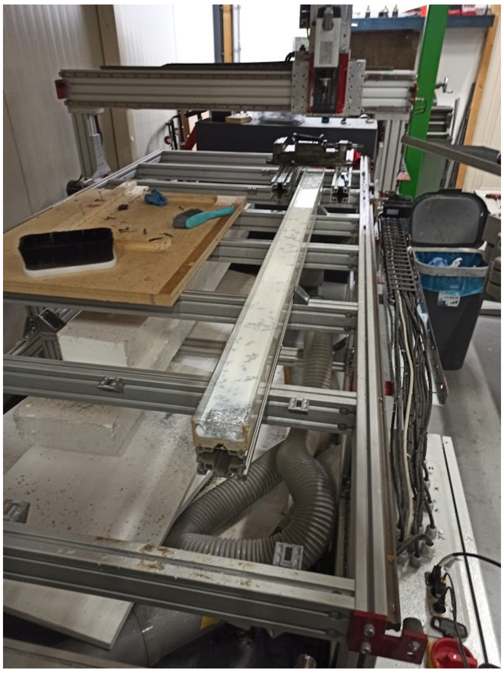

Figure 21: Cooling bath on the CNC portal mill used to mill the aluminium clamping strips

Figure 22: Close up of the membrane clamped up in the clamping system

Figure 23: Clamping system with the membrane mounted to the machine (including the later added front and back clamp)

5 Pneumatic system

This chapter describes the design work, design choices and fabrication work that has been done to the pneumatic system. This chapter starts with a detailed description of the process that the system runs through during a lamination cycle, after which the different options to achieve this are described. From these proposals, the best option is chosen, and a system design is proposed and fabricated.

5.1 System overview

Based on the findings of the analysis of the lamination process which is described in section 2.1 the lamination process is split up in different steps. To run through a normal lamination cycle, the following steps are taken:

- Evacuation ramp
- Evacuation
- Upper chamber ramp
- Lamination
- Bleed

During the "evacuation ramp" both the upper and the lower chamber are evacuated of all air to create a vacuum. The ramp rate of the pressure drop can be adjusted if a longer evacuation time is needed than the vacuum pump normally achieves. This step ends when a set vacuum threshold is reached.

During "evacuation" both chambers are kept at the maximum amount of vacuum that the pump can achieve, during this time the air that is trapped between the laminate can escape preventing air bubbles in the finished product. This step ends when the desired evacuation time has passed.

During "upper chamber ramp" the pressure in the upper chamber is increased with a set ramp rate to the desired pressure difference between the upper and lower chamber. This step ends when the desired pressure difference is reached.

During the "lamination step" the pressure difference between both chambers is kept at a constant set pressure difference. This pressure difference defines the force upon which the membrane pushes onto the laminate. The encapsulant gets into a liquid state during this step which combines all the layers into one product. This step ends when the desired lamination time has passed.

When the "lamination step" is finished the "bleed step" starts. During this step the pressure in both chambers is gradually increased back to atmospheric pressure after which the lid can be opened and the laminate can be taken out of the machine to cool down.

A pressure graph of a typical lamination cycle is shown in figure 3.

The pressure in the bottom chamber has two states, atmospheric and full pump vacuum, however, in the transition between these states, the speed of this transition needs to be adjustable. This rate is not expected to be very critical, so simple adjustment methods should suffice.

The top chamber has, besides the same states as the bottom chamber, a state where an exact pressure difference between the top and the bottom chamber should be sustained. The ramp from evacuation to this state should also be controllable.

5.2 Pressure regulation principles

During the analysis phase, the pneumatic systems of other machines have been analysed, both machines are based on valves that are either open or closed. The machine at the NHL does not have pressure control and ramp rates are set by flow regulators. The machine at Eurotron regulates the pressure and ramp rate by opening and closing a series of smaller valves at a high switching rate. This is similar to what is expected to have been fitted to the Mito machine previous to Mito's acquiring based on the findings during the analysis phase.

The system as used on the NHL machine does not suffice to the requirements set in the first chapter, therefore this system will not be further considered.

As precise pressure control was one of the requirements, an additional option for pressure regulation was sought in place of the open-closed method. During this research the option of using an Electro Pneumatic Pressure Regulator (EPR) was discovered. An EPR can control air/vacuum pressure steplessly in proportion to an electric signal [2].

The SMC ITV series have a reverence, supply, outlet, and exhaust connection. Internally a pressure is generated equal to the setpoint relative to the reverence connection, this pressure is applied to the pilot chamber above a diaphragm as shown in figure 24. If the pressure on the outlet connection is higher than the pilot chamber, the diaphragm opens the valve towards the exhaust which is connected to the vacuum. Respectively if the pressure is lower, the supply valve is opened which is connected to compressed air. In this way, this valve can be used in both directions and is thus ideal for this application.

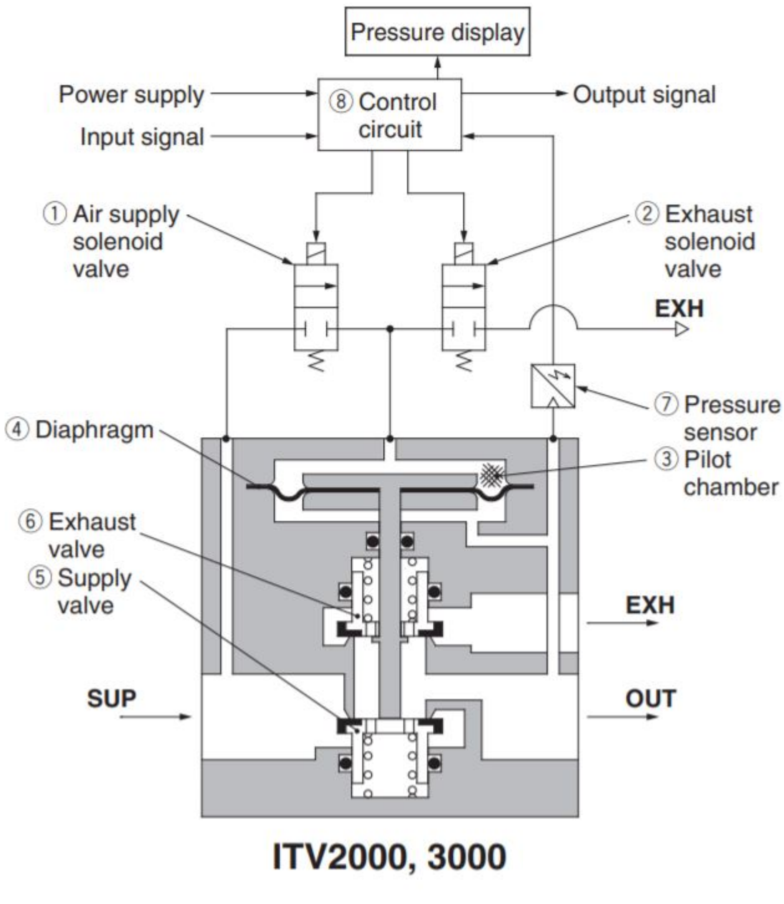

Figure 24: Working principle of the EPR (source [2])

In the laminator application, after the evacuation phase has passed, the pressure in the upper chamber of the laminator will be increased with compressed air by the regulator until the pressure in the upper chamber is equal to the set pressure difference. If the pressure rises during the lamination process due to thermal expansion or leaking around the membrane, the valve automatically opens towards the vacuum connection to bring the chamber back to the set pressure difference. If the pressure in the lower chamber changes, the top chamber will automatically follow as the regulator is referenced to the bottom chamber. The amount that valve opens depends on the pressure error, therefore the regulation is more stable than the open-closed system as used in the Eurotron machine which is always fully open or closed, resulting in big steps and relatively large hysteresis.

To make a good choice as to which system would be the best to implement on this machine table 3 is set up based on the Kesselring method.

Aspect	Electronic pressure regulator	Open - closed
Precise	5	2
Availability	2	5
Implementation	4	3
Programming	5	2
Cost	2	4
Total:	18	16

Table 3: Design choice of the pressure regulation system using the Kesselring method

Based on the outcome of table 3 the electronic pressure regulator would be better, however, this is still based on objectiveness. A similar application where this type of regulator was used could not be found and the manual did not specify much about using the regulator with both vacuum and compressed air at the same time, therefore it was unsure if this regulator would indeed be the right solution. After discussing this with Mito the decision was made to try the Electronic pressure regulator, as a cheap second-hand one was available on E-bay. This meant that we could try this method without risking a lot of costs.

5.3 Testing of individual systems

To get a good understanding of what does and doesn't work and what the limitations are of the existing components, parts of the control system were build up and manually tested. During these tests it quickly became apparent that for the vacuum connections it was needed to use large diameter hoses and valves. In the leftover parts from the old system there are two 16mm valves left which seemed to flow sufficiently for the evacuation step.

When bleeding back to atmospheric pressure this became even more apparent, with a 16mm hose completely disconnected from any valve or obstruction it took several minutes before the lid could be opened. This is due to the large surface area of the lid, a very low vacuum in the chambers is already sufficient to keep the lid closed. At the same time, this low vacuum will only create a small flow of air coming in from outside of the camber because of the low-pressure difference. To improve this, the inlet of the chambers was connected to compressed air to create a larger pressure difference. This greatly improved the bleeding speed and gave great and constant control of the bleed time when combined with a flow regulator.

Testing with the EPR was performed with a power supply and a potentiometer controlling the desired set value. This worked adequately, however one of its downsides was its lack of speed. Evacuation of the chamber to full vacuum through the EPR takes about 5 minutes, this is due to the low flow and because the minimum setpoint cannot be set negative. Therefore the bottom chamber first needs to evacuate after which the EPR will follow. Regulating to a desired set speed does go well if the pressure side is supplied with 3 to 4 bars of compressed air. The speed can be adjusted by setting a higher than-desired setpoint up until this setpoint is almost reached, thereafter the EPR will be set to the desired value. It was however found, that the regulator would still react to changes in the outlet pressure even when turned off as the last set pressure is still inside of the pilot chamber, therefore an extra valve should be added to the outlet to prevent unwanted behaviour during the evacuation step.

Doing these experiments gave a adequate understanding about how the machine responds to certain inputs, what sizes of valves, couplings and hoses were needed and which control methods could be used. The testing with the EPR gave sufficient trust that in the achievability that the system will be further developed and implemented.

5.4 Pneumatic system design

Based on the results and conclusions that were made during testing the pneumatic design was made. Figure 25 shows the pneumatic diagram, added to this diagram is a non-return valve between the upper and lower chamber. This valve is added to prevent the membrane from being sucked up too far by the vacuum and possibly rupturing. In a normal lamination cycle, the pressure in the upper chamber will always be higher or equal to the lower chamber, however, in case of a programming error or a leak of a faulty valve, the non-return valve should prevent this from happening.

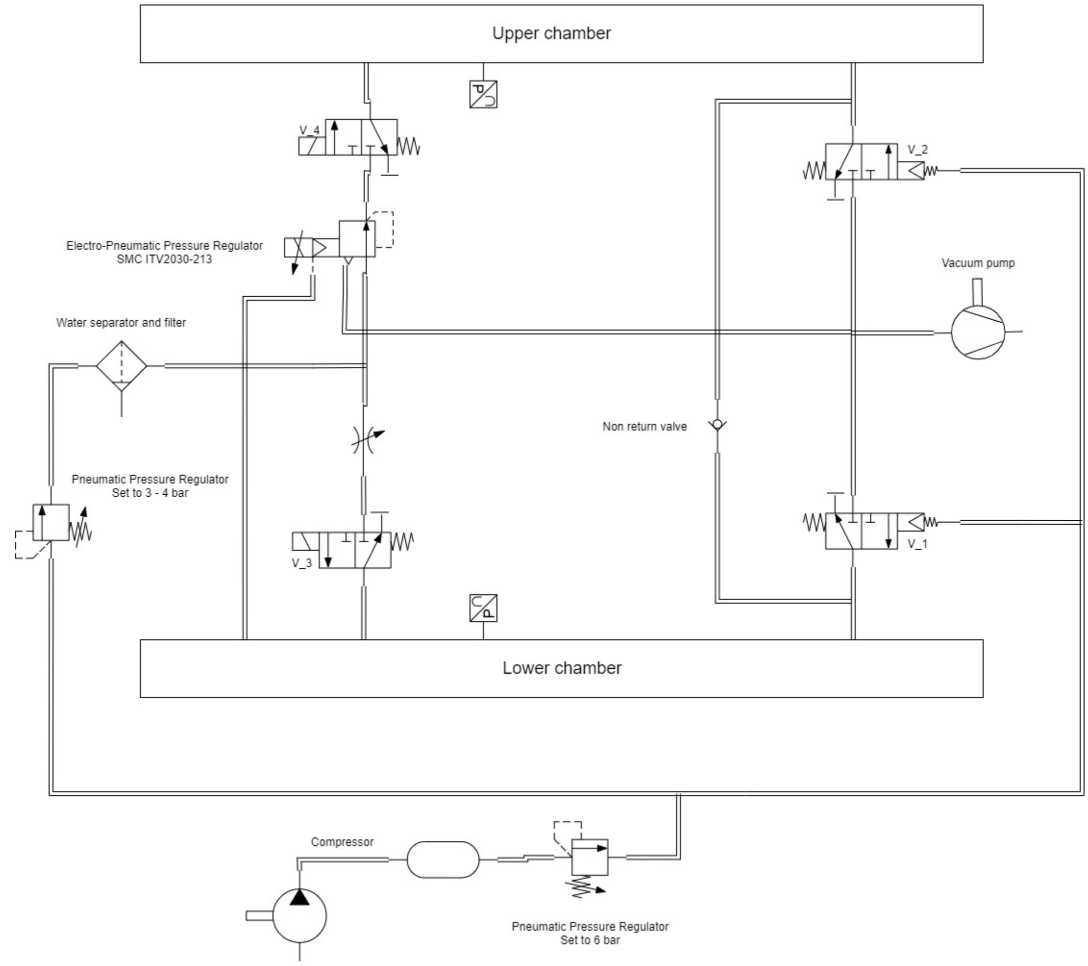

Figure 25: Pneumatic schematic of the laminator

5.5 Pneumatic fabrication work

A lot of the parts used in the pneumatic system were reused from the original system, the only additions needed were the EPR, a manual pressure regulator with water separator and filter, a non-return valve, flow regulator and various couplings and hoses.

The pneumatic system was mounted onto the back of the machine as there was an unused and rather protected area available close to the vacuum pump and the connections of the upper and lower chamber. An aluminium plate was used to mount all the components, this plate was mounted directly onto the frame of the machine and added additional rigidity to the lid actuator mounting.

Figure 26 shows the mounting plate during initial layout and routing, figure 27 shows the finished pneumatic system mounted to the machine and connected to the chambers, vacuum pump and compressor.

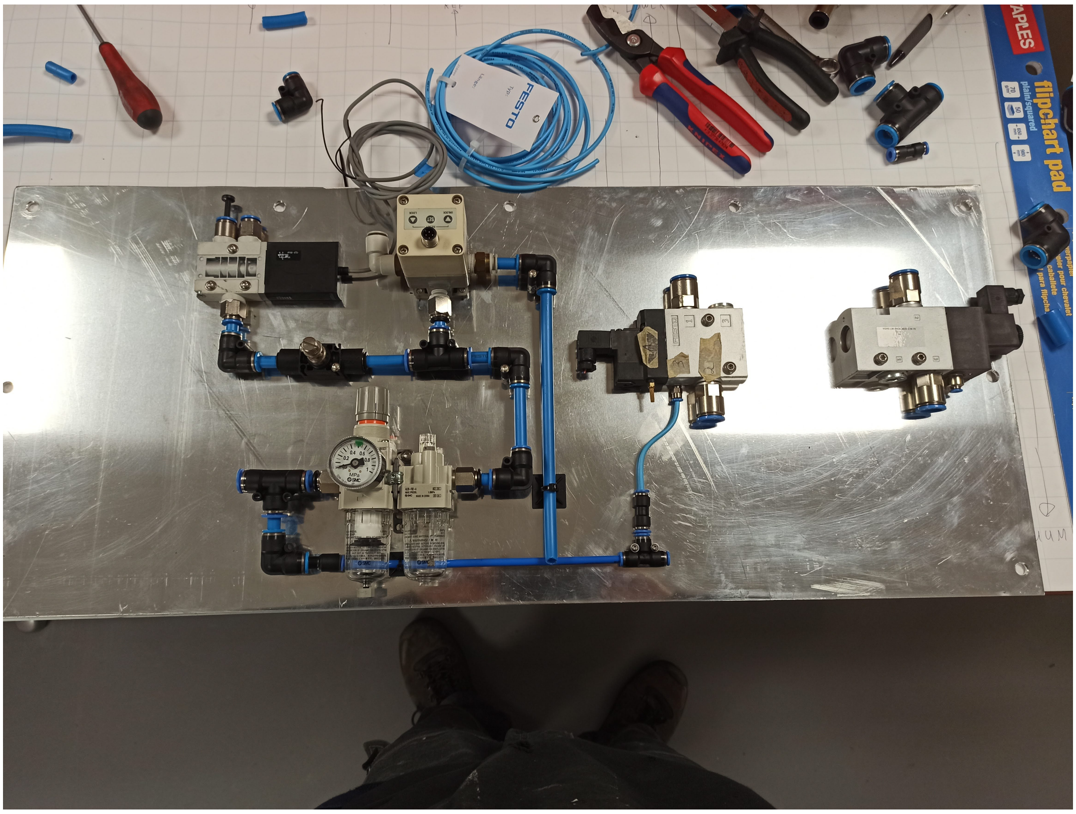

Figure 26: Mounting plate during initial layout and routing of the pneumatic system

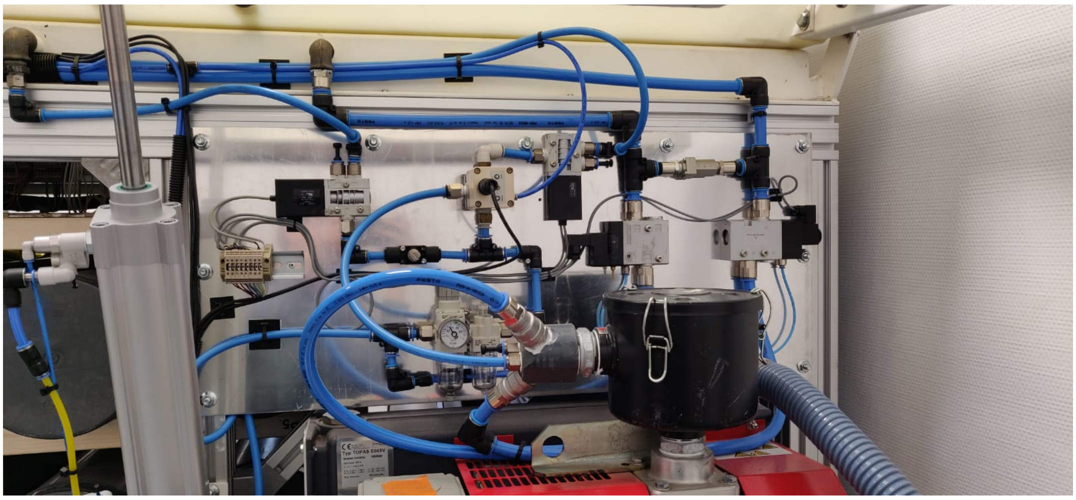

Figure 27: Complete pneumatic system mounted to the machine

6 Electronic system

This chapter describes the functionality, design, and fabrication of the electronic system. This starts with a system overview from which the controller is selected, based on that the schematic is made. The fabrication work is performed, after which the software for the controller is written.

6.1 System overview

The electronic system is the interface between the user, the pneumatic system and the heat bed controller. The user needs to be able to make a lamination recipe, which the electronic system runs, it sets all the outputs and gives the user feedback on the process.

The heating system is a standalone system that works independently from the system described in this chapter. It is however the intention to connect the communication of the temperature controller to the electronic system controller. Doing so would allow setting all the recipe settings including the temperature through the same interface.

The control cabinet is equipped with indicators and buttons, these will be used as the interface between the user and the machine. An additional interface to make the recipe and to keep track of the progress will be needed.

The controller will need to control the pneumatic valves, the EPR, the vacuum pump and read the pressure in the upper and lower chamber as well as setting indicators and reeding buttons.

6.2 Controller selection

To be able to select a suitable controller it is first necessary to know what inputs and outputs are needed. Therefore, the following list has been made.

- Inputs:
  - o Upper chamber pressure (0-10V analog) (came with the machine)
  - o Lower chamber pressure (0-10V analog) (came with the machine)
  - o Mode switch (I/O digital)
  - o Start button (I/O digital)
  - o Stop button (I/O digital)
- Outputs
  - o Lower chamber vacuum valve (24Vdc coil)
  - o Upper chamber vacuum valve (24Vdc coil)
  - o Lower chamber bleed valve (24Vdc coil)
  - o EPR on/off valve (24Vdc coil)
  - o EPR setpoint (0-10V analog)
  - o Run indicator (24V)
  - o Error indicator (24V)
  - o Standby indicator (24V)
- Communication
  - o Temperature controller (Modbus RS485)
  - o Human Machine interface (t.b.d.)

The mode switch is a three-way switch that can either be in the middle position where the machine is in off-mode and only the temperature controller is activated, standby-mode where the controller gets a high signal on the input which prepares the machine to start a lamination cycle, or in pump-mode where the contactor of the vacuum pump is directly engaged and all the valves are kept closed. The pump-mode is used if a vacuum source is externally needed for other applications than making PV modules like laminating composite parts.

These requirements were discussed during a meeting with Danny Bokma, the electronics engineer at Mito. During this meeting, further requirements were set and the requirements that were made during the analysis phase were added, which resulted in the following list.

- 3 digital inputs
- 2 analog inputs (0-10V)
- 8 digital outputs (24V inductive load tolerant)
- 1 analog output
- RS485 communication
- SD card interface for data logging
- Software that can be easily modified for later changes or additions
- Extra in and outputs for later changes or additions (undefined)
- Software language that is easy to understand and learn
- USB and/or Ethernet connection to use a computer for monitoring and as an HMI
- Industrial self-contained unit, so no enclosures or extra hardware will have to be made
- 24V supply voltage and I/O, so one power supply can be used (reused from the old setup)

These requirements were used as a basis for the controller selection, Danny Bokma had good experiences with using an Arduino based PLC from Industrial Shields which he advised to use. This PLC has the advantage that it can be programmed in the C programming language, this is a very common programming language for which a lot of tutorials and examples are available. Therefore, otherwise difficult processes like writing to an SD card are simplified as standard libraries containing most of the difficult functions and leaving only the implementation to the user.

The Industrial Shields: M-DUINO PLC Arduino Ethernet 19R+ fits within all these requirements and has plenty of extra capabilities for later changes or additions. After looking into other solutions, the M-DUINO also appeared to be the most cost-effective solution, therefore the decision was made to choose that controller.

6.3 Electronic schematic

Based on the manual [3] of the M-DUINO the following input/output list is made to divide all the components over the available connections.

- Inputs:
    - o I0.0 Mode switch
    - o I0.1 Start button
    - o I0.2 Stop button
    - o I0.4 Lower chamber pressure
    - o I0.5 Upper chamber pressure
- Outputs
    - o R0.1 Run indicator
    - o R0.2 Error indicator
    - o R0.3 Standby indicator
    - o R0.4 Lower chamber vacuum valve
    - o R0.5 Upper chamber vacuum valve

o R0.6 Lower chamber bleed valve

o R0.7 EPR on/off valve

o EPR setpoint

Based on this list a schematic is made which is displayed in figure 28.

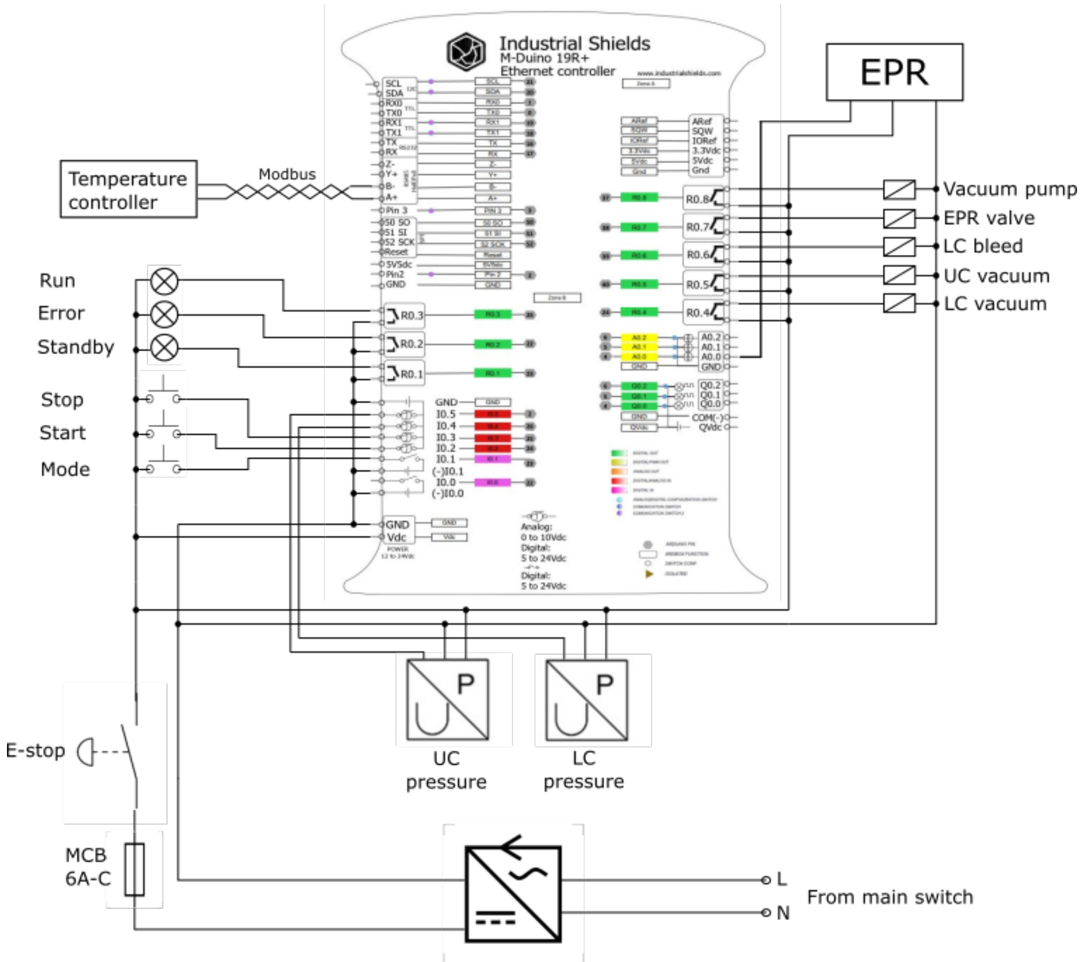

Figure 28: Schematic of the electrical system without the heating system

6.4 Electronic fabrication work

After the schematic was completed, all the necessary components where gathered from either the old setup or were ordered. After this, the electronic system could be fabricated. This started with the mounting off all the components into the electronics cabinet. First, all the components were placed in such a way that there was enough space for cable trays between the components, and terminal blocks were placed to connect all the incoming wires. Terminal blocks were also placed to distribute power and ground lines to all the components. The electronics cabinet is also shared with the heating system and has pneumatic lines for opening and closing the lid which is directly controlled by a pneumatic switch on the control panel on top of the cabinet. Figure 29 shows the finished electronics cabinet mounted to the machine.

The electronics cabinet exists out of the following components:

A) Solid-state relay for heating elements

B) Circuit breaker for heating elements

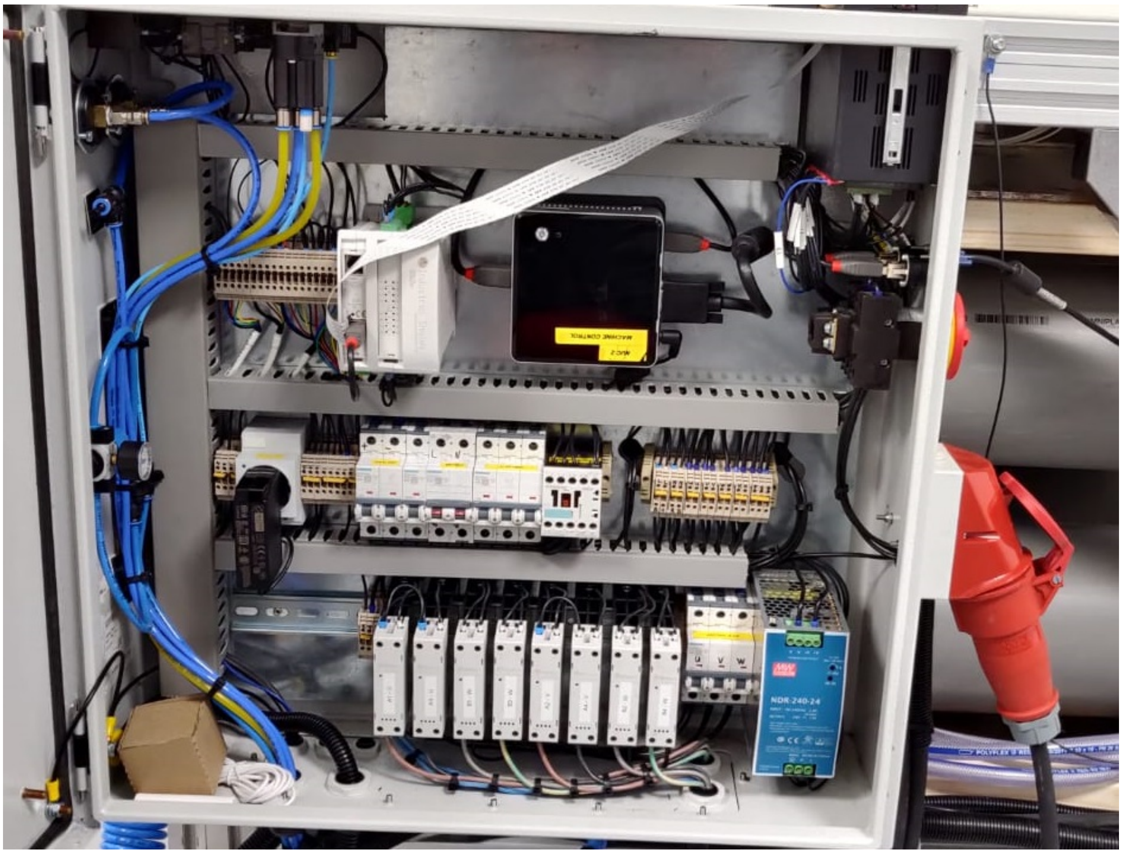

Figure 29: Electronics cabinet mounted to the machine including the pneumatic lines for the lid actuator

C) Temperature controller

D) Mains input and main machine switch

E) 24V dc power supply

F) High voltage power distribution

G) Circuit breakers for: Aux components, compressor, vacuum pump and 24V

H) Contactor for the vacuum pump

I) Terminal block for sensor and valve wiring coming from the pneumatic system

J) M DUINO PLC

K) Intel NUC panel PC for human-machine interface (connected to an external monitor)

L) Power supply for the Intel NUC

M) SD card extension cable, added to be able to place and remove the SD card outside

N) Pneumatic switch for opening and closing the lid of the machine

O) Indicators, buttons and switch mounted in the control panel

P) Low voltage power distribution

The control panel is mounted in the top of the electronics cabinet, the empty cabinet was already mounted to the machine with the switches and indicators mounted before this project started. The temperature controller was also placed in the control panel, and a panel PC (Intel NUC) which was leftover from the old system was placed in the electronics cabinet to run the Human Machine Interface. This PC was connected to a monitor, mouse and keyboard, mounted to the side of the machine. During the testing phase, two vacuum manometers were added on the side of the cabinet to be able to take a glance over the pressures in the top and bottom chamber. Figure 30 shows the complete Human Machine Interface mounted to the machine.

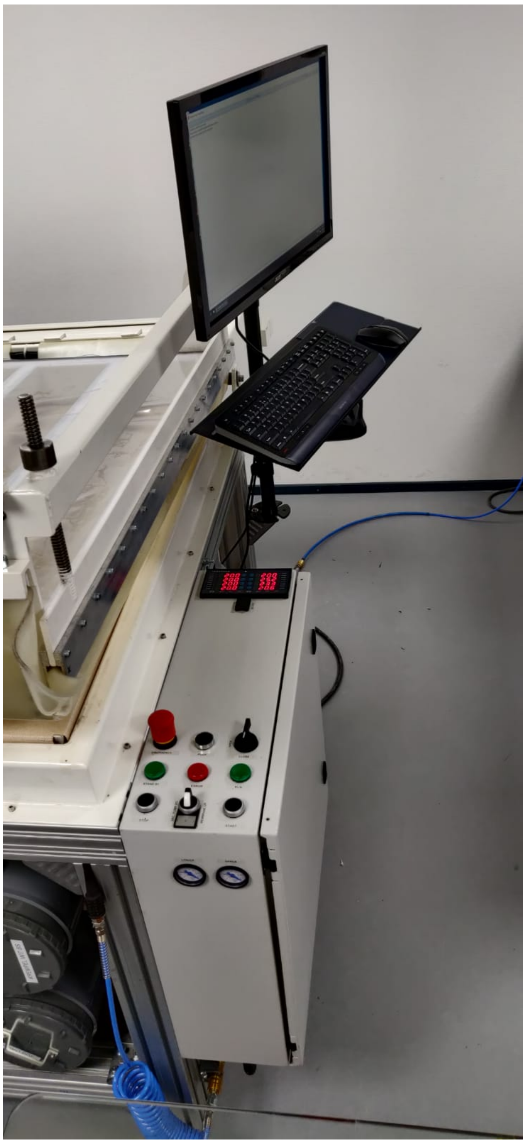

Figure 30: Human Machine interface consisting of the control panel, monitor, keyboard and mouse

6.5 Software

To be able to control the machine the PLC had to be programmed with software, which consisted of a learning process by trial and error as there was no previous experience in writing software. For this reason, the software is not described in depth in this report.

To understand which steps have to be taken by the PLC, the lamination process is first broken down into a flow chart as displayed in figure 31.This flow chart has been discussed with Jules and Danny to confirm the process and the technical correctness. From the flow chart, the states are written in a case structure in the software. Each case sets the outputs depending on the machine's state, and checks wherever a statement is met which starts the next state. Besides the case structure, other functions having tasks like keeping track of time, reading inputs and writing outputs, writing data to the SD card and the printing data to the HMI and communicating with the temperature controller had to be written. The complete software is shown in appendix B.

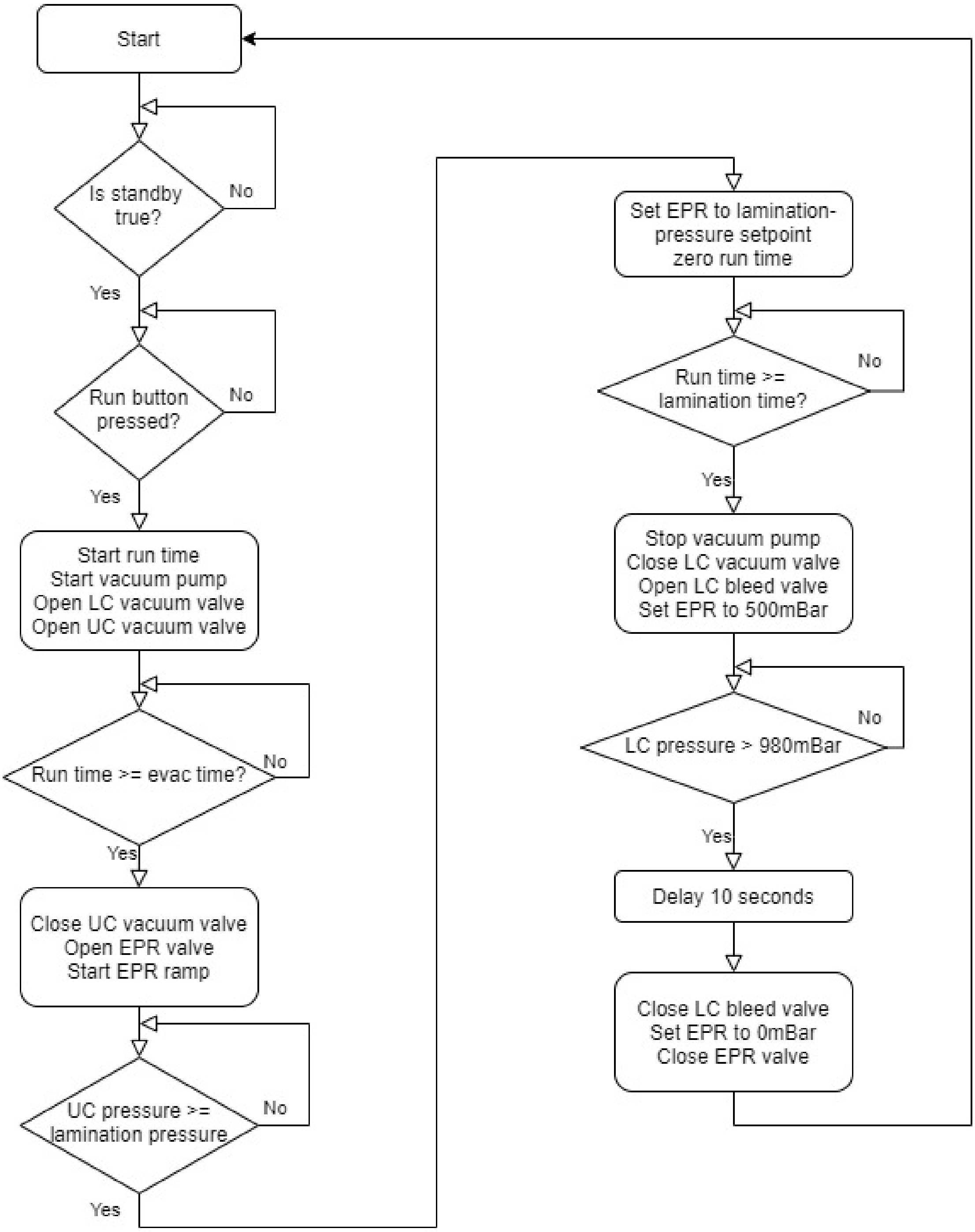

Figure 31: Simplified flow chart of the lamination cycle process that is runned by the PLC

7 Testing and Optimisation

The testing and optimisation phase started when the software for the PLC was written during the electronics phase. During the writing of the software, individual sections of the machine started working. On each of these sections, tests were performed and improvements were made to the software until all functions worked according to the requirements.

After implementation of all the in and outputs and the addition of a case structure based on the flow chart that was presented in the previous chapter, it became possible to run the machine through a lamination process. This process was however far from stable. During these preliminary tests it also became apparent that a means of logging was needed to be able to check how well the machine performed. The internal SD card of the PLC was utilized to create a comma-separated file that contained all the settings, the state of the machine and all the measurements that were recorded every second. Appendix C displays the first page of such log file. These log files can be opened in Excel to create a graph, figure 32 shows the graph created from a log file during a lamination process.

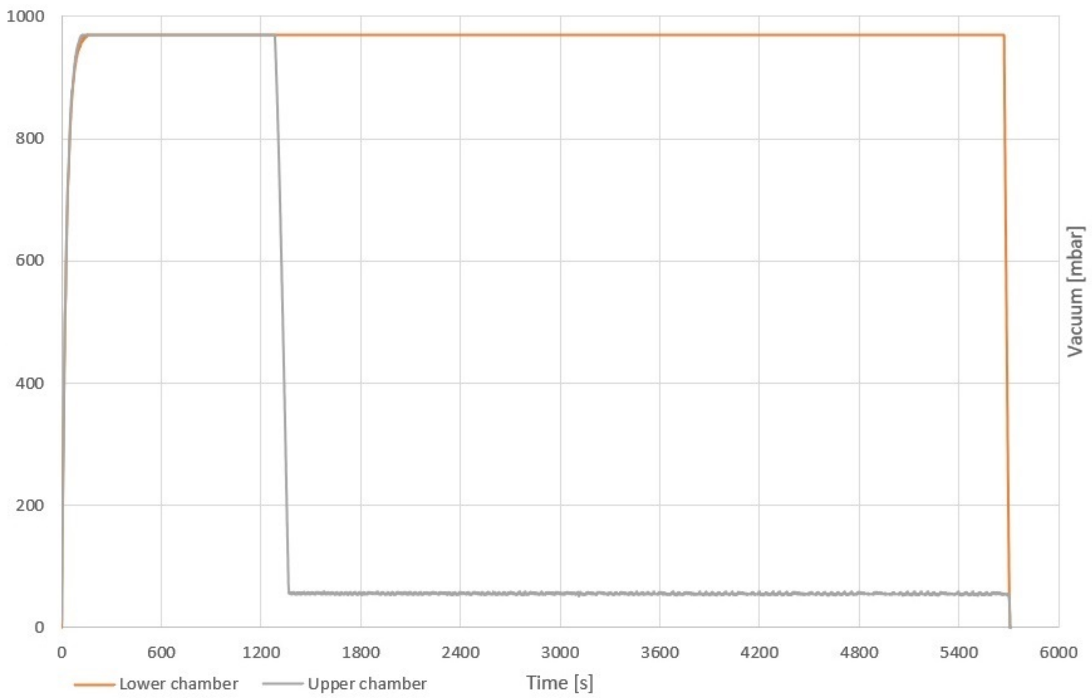

Figure 32: Graph created from a log file, the recipe settings were: evacuation ramp 0s, evacuation time: 1200s, lamination ramp time, 60s, lamination time 5400s, lamination pressure, 920mBar

Using the log files, it was easy to create a visual representation of the lamination process which helped with optimising the process, this way ramp rates could be checked, and overshoot or deviations could easily be found and optimised upon. To add a unique identifier for each log file the Real-Time Clock of the PLC was used. This clock named the log file to the exact date and time at the moment the run was started, the file name was also printed on the human-machine interface so it could be printed onto a label that can be placed on the laminated product. In this way, it would always be possible to look up which settings were used or how the lamination process commenced of that individual product.

When the lamination process became more stable, additional features were added like evacuation ramp control. This function can be used if a slow evacuation ramp is preferred to aid evacuation. The speed of this ramp is normally limited by the capacity of the vacuum pump and takes about 120 seconds if no evacuation ramp rate is specified. If an evacuation ramp is specified the PLC closes the vacuum valves if the pressure drops faster than the specified rate, the valves are reopened when the ramp rate has caught up. This open-closed method does have a large hysteresis but is expected to be precise enough for the evacuation ramp.

During testing, it was found that the back sheet of the laminate was occasionally wrinkled. No noticeable differences could be found in the log files when this happened, and it was unknown what caused this. After a while, it became apparent that this issue was more common if the lid had been closed right before inserting the new laminate or if a lamination process was started right after another one.

This problem turned out to be caused by the hot membrane touching the backside of the laminate during the evacuation step, this caused the back side encapsulant to melt before pressure was applied to it. To prevent this the membrane would have to be kept in an upwards position during the evacuation step. This seemed simple at first as a small vacuum in the upper chamber would be enough to keep the membrane up. However, this solution worked very inconsistently.

This was due to the volume in both chambers changing relative to the position of the membrane, and the small pressure difference between both sides of the membrane that was needed to move it up. A pressure difference of 5mBar was sufficient to hold the membrane up, however, a pressure difference of 30mBar could pull the membrane in so far that it would close off the vacuum and bleed ports of the chamber making it almost impossible to get the membrane back down. This low-pressure difference was also not enough to generate much difference in the flow out of the vacuum valves during the evacuation ramp. Therefore the membrane would always be pulled to the chamber where the volume was the lowest.

To gain control over the position of the membrane during the evacuation phase, The membrane needed to be lifted up before the lamination process was started, in this way it was known that the volume in the upper chamber was always lower than in the bottom chamber. During the evacuation phase, the upper chamber valve could shortly be closed if the pressure difference would get too high. To keep the membrane in the up position, the PLC constantly measures the vacuum in the upper chamber if it is in standby mode. If the vacuum in the upper chamber has gotten too close to atmospheric, then the vacuum pump will be turned on and the upper chamber vacuum valve will be opened until the target vacuum is reached.

Another issue that arose with keeping the membrane up is that the sides of the membrane get sucked into the lid when the lid is open as the membrane is only mounted at the left and right sides, to resolve this the front and back sides of the membrane will also have to be fixated.

7.1 Improvements

The only significant improvement that had to be made to the machine was the addition of the membrane mount. To see if a tensioning system would also be needed the membrane was first fixed to the frame with clamps. This already was a big improvement so only a simple clamping mechanism was needed. Most of the tensioning could still be done with the membrane tensioner that was already fitted to the machine.

On the backside of the machine, the membrane could simply be clamped to the actuator arm of the frame. For this, a system with short sections of C profile with a bolt through it was used to generate the clamping force. A flat piece of steel strip was used to distribute the load so that the membrane would be evenly tightened up to the frame. Figure 33 shows the top view of newly made clamping system on the backside of the machine, figure 34 shows the bottom side of the clamp.

On the front of the machine, there was not anything to clamp the membrane to, the only suitable mounting points were the front mounts of the membrane tensioner that were already in place. Therefore, a clamping strip was designed that could be mounted to the frame with a threaded rod. This allowed for additional tensioning of the membrane. The clamp itself is made of a steel U profile, a flat strip on the inside that slides on 4 bolts clamps onto the membrane, bolts from the backside push onto the strip to clamp everything together. Figure 35 shows how the membrane is clamped in the strip. Figure 36 shows the front clamping system mounted to the machine, 2 pieces of angle iron provide the mounts for the threaded rod.

After this improvement was made, the machine performed very well. The membrane could now easily be pulled upwards and barely leaked when the lid was open. The vacuum pump would only turn on for a few seconds every 5 minutes to sustain the vacuum in the upper chamber in

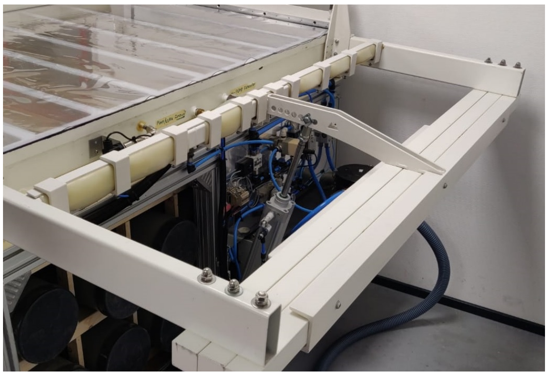

Figure 33: Additional membrane clamp that was added to the backside of the machine

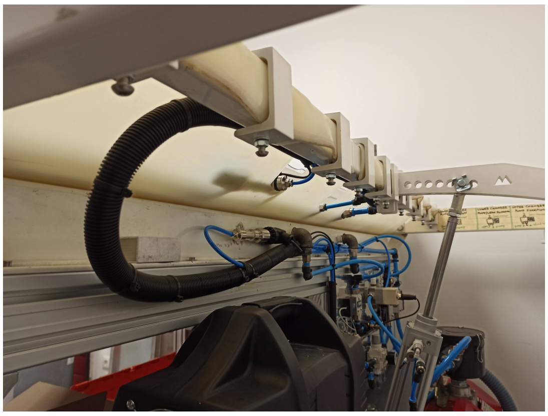

Figure 34: Bottom view of the additional membrane clamp that was added to the backside of the machine

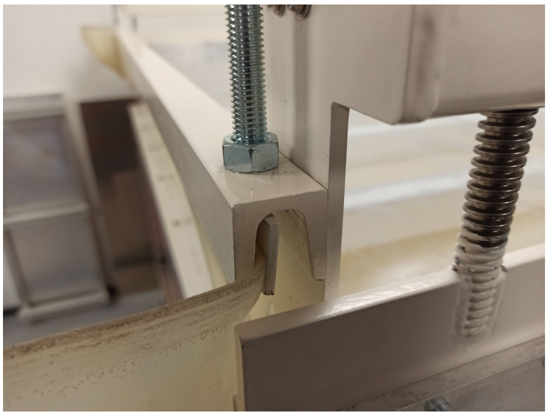

Figure 35: Close-up of the front membrane clamp

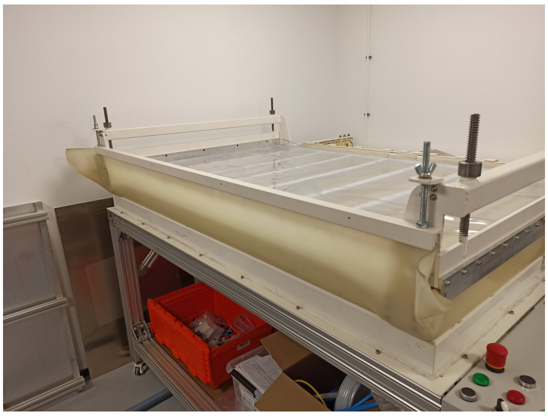

Figure 36: Front clamping system mounted to the machine

standby mode. Figure 23 shows the membrane in the up position with the additional clamping system in place.

The addition of extra parts on the lid, did however add extra weight, therefore additional weight needed to be added to the counterweights to restore the balance. The remaining section of C profile that was left over from the back clamps was mounted to the counterweights and also provided extra support to the weight plate that was already added when the left and right membrane tensioner was mounted.

With all the improvements done, the machine was in good working order. Jules and Danny from Mito Solar started using the machine to do numerous tests and to make recipes. The ease of adjustability helped a lot with optimising the lamination process to get even better results.

The complete machine in its finished state is displayed in figure 37.

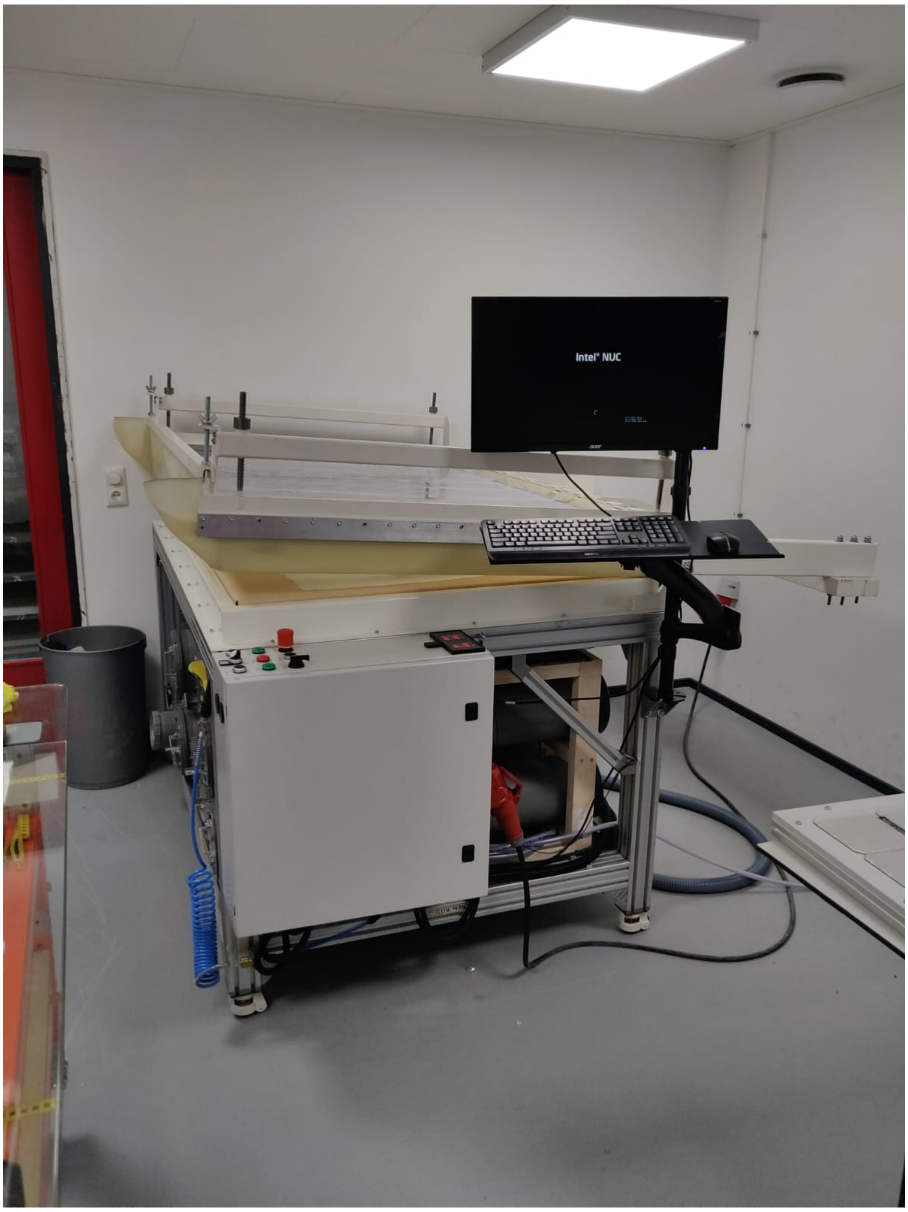

Figure 37: The complete machine in its finished state

8 Conclusion

In this report the following question is answered: How can a fully functional PV module laminator be made while trying to reuse as many parts as possible and keeping the cost down? This question has successfully been answered, in addition to that this report describes how a fully functional PV module laminator can be made from an existing machine base, it also shows how existing parts can be reused as much as possible and how the cost can be kept to a minimum.

The makings in this report resulted in a PV module laminator that is capable of laminating flexible PV modules with all the encapsulants that are widely available and required by Mito. The lamination recipe can easily be adjusted through the human-machine interface and the process is logged onto an SD card. These logs proved that the lamination pressure is stable within 4mBar of variation, the maximum static offset is less then 15mBar and the lamination pressure can range from 20mBar to 980mBar. The heat bed gets up to 160°C within 30 minutes, however, it does take about 15 minutes to fully stabilize.

The membrane is mounted to the lid of the machine and all the external parts are mounted to the machine itself, thus the machine is self-contained. Therefore the machine can be easily moved without having to connect or remove components and it can be used anywhere where there is a 3 phase 400V 16A outlet available. The Arduino based PLC has extra in- and outputs available for later changes or additions, the PLC software is written in an opensource program so it can always be changed without the need for a license.

The total cost of all the parts that had to be bought to get the machine to the completed state were €2,094.96, this is within the budget of €2,500.00 that was set by Mito Solar. Most expenses were spent on the pneumatic system, this also includes the lid actuator and evacuation hose to get the vapour from the vacuum pump out of the workspace. Appendix A contains an overview of the expenses.

The finished machine has already successfully been used to produce about 10kWp of PV modules without major issues. And has been used almost continuously in the last two months.

8.1 Recommendations

A possible improvement to the machine would be to add a graphical interface to the HMI. The current HMI is text command-based; therefore, the full recipe must be written every time one setting needs to be changed. If this can be filled-in in individual text boxes on a graphical interface, this process would get more convenient. Additionally, the ability to save and reopen these settings would make it a lot easier to switch between encapsulants.

Furthermore, the addition of graphical feedback on the state of the machine during the lamination process could improve the reliability as issues could be spotted more quickly and can be acted upon. The experience in writing software is too limited to add these improvements, therefore, if someone with more knowledge on software were to add these features and improve the PLC software that was written, the machine could be brought to a more high-end level and the user would have a more convenient machine to work with and have less chance of making mistakes.

References

[1] John Pern. Module encapsulation materials, processing and testing, December 2008.

[2] SMC CORPORATION. Electro-pneumatic regulator, electronic vacuum regulator.

[3] Industrial Shields. M-duino plc arduino ethernet 19r ios relay analog digital plus, August 2019.

Appendices

A Cost overview

Component	Cost
Pneumatics	€708.03
Electronics	€205.95
Consumables and others	€197.50
Bare steel and aluminium	€304.19
Laser cutting plates	€99.67
PLC M-DUINO PLC 19R+	€282.45
EPR SMC ITV2030-213L	€102.45
Temperature controller CJZ-8800	€194.72
Total:	€2,094.96

B PLC code

```

/*
 * Program to run laminator through all the steps
 *
 * written by Ray Blokker for Mito Solar
 * rayblokker@gmail.com
 *
 */

#include <EEPROMex.h>
#include <ModbusRTUMaster.h>
#include <RS485.h>
#include <SPI.h>
#include <SD.h>
#include <RTC.h>


//Setings
bool debug = true;                                    //print readable data
int evac_ramp_time = 0;                               //evacuation ramp time
unsigned long evac_time = 300000;                     //evacuation time (...
    from the moment of opening the valves) in milisecond (*1000)
unsigned long lamination_time = 600000;               //lamination time (...
    from the moment the perssure set point has been reached) in miliseccond ...
    (*1000)
int lamination_rise_time = 60;                        //time that the ramp ...
    takes to get from evacuation to lamination pressure in secconds
int lamination_pressure = 500;                        //lamination pressure ...
    difference between upper and lower chamber in mBar
int temp_sp = 150;                                    //temperature setpoint...
     in C
int evac_done_pressure = 920;                         //pressure at wich LC ...
    is assumed to be fully evacuated
int evac_press_diff = -10;                            //pressure difference ...
    between upper and lower chamber during evac (negative = membrame up) in ...
    mBar
int evac_press_hyst = 4;                              //pressure difference ...
    hystereses during evac to prefent valves from oscilating in mBar
const int UC_stop_pressure = 150;                     //upper chamber ...
    pressure where the EPPR stops following the lower chamber in mBar
const int LC_stop_pressure = 2;                       //lower chamber ...
    pressure where the proces end and the post bleed time starts in mBar
```

```
const unsigned long post_bleed_time = 4000;          //time that the bottem...
     chamber bleed valve keeps bleeding after getting close to atmosphere in ...
    miliseccond (*1000)

//Variables
int state = 0;
int lamination_ramp;
int evac_ramp;
int evac_ramp_set = 0;
unsigned long run_time;
unsigned long start_time = 0;
unsigned long Lamination_Start_time;
unsigned long last_ramp_time = 0;
unsigned long post_start_time;
unsigned long last_millis = 0;
unsigned long bleed_start = 0;
unsigned long evac_start = 180000;
bool newData = false;
bool evac_start_flag = false;

//outputs
bool Pump;
bool Run_Indicator;
bool Error_Indicator;
bool Stndby_Indicator;
bool LC_vacuum_valve;
bool UC_vacuum_valve;
bool LC_bleed_valve;
bool EPPR;
int EPR_SP = 0;

//inputs
bool startPin;
bool modeSwitch;
int UC_pressure;
int LC_pressure;
int Pressure_diff;

//sd card
const int chipSelect = 53;
bool Log = false;
char filename[] = "0000/00000000.CSV";
char foldername[] = "0000";

//serial input vars
const byte numChars = 32;
char receivedChars[numChars];
char tempChars[numChars];                //temporary array for use when ...
    parsing

//temp controller vars
ModbusRTUMaster master(RS485);
uint32_t lastSentTime = 0UL;
int temp_read;
int channel = 1;
int channel_set = 1;
int temp = 0;
int avg_temp;
uint16_t setpoint = 150;
int change = 1;

void setup() {
  //read EEPROM values
  evac_time = EEPROM.readLong(0);                             //evacuation...
     time (from the moment of opening the valves) in milisecond (*1000)
  lamination_time = EEPROM.readLong(4);                       //lamination...
     time (from the moment the perssure set point has been reached) in ...
    miliseccond (*1000)
  lamination_rise_time = EEPROM.readInt(8);                   //time that ...
    the ramp takes to get from evacuation to lamination pressure in secconds
```

```
  lamination_pressure = EEPROM.readInt(10);                         //lamination...
     pressure difference between upper and lower chamber in mBar
  temp_sp = EEPROM.readInt(12);                                     //...
    temperature setpoint in C
  evac_ramp_time = EEPROM.readInt(14);                              //evacuation...
     ramp time

  RS485.begin(9600, HALFDUPLEX, SERIAL_8N1);                        //begin ...
    RS485
  master.begin(9600);                                               //begin ...
    Modbus
  Serial.begin(9600);                                               //begin ...
    serial

  if (!SD.begin(chipSelect)) {                                      //check if ...
    SD card is pressent
    Serial.println("Card failed, or not present");                  //and tell ...
    it is
  }
  else{
    Serial.println("card initialized.");                            //tell card ...
    is initialized
    Log = true;                                                     //set lof ...
    flag
  }

  print_start();                                                    //print ...
    start serial data
}

void loop() {
  recvWithStartEndMarkers();                                    //check if new ...
    data is entered

  //read inputs
  UC_pressure = map(analogRead(I0_4), 0, 1023, 970, 0);     //Read upper ...
    chamber pressure
  LC_pressure = map(analogRead(I0_5), 0, 1023, 970, 0);     //Read upper ...
    chamber pressure
  Pressure_diff = LC_pressure - UC_pressure;                //calculate ...
    pressure difference


  if(state == 0){                                           //when in standby ...
    state
    if(bleed_start + 3000 < millis()) LC_bleed_valve = LOW;     //close ...
    lower chamber bleed valve after 5 sec

    if(digitalRead(I0_1) == HIGH){                        //if mode switch is in...
     standby
      digitalWrite(Stndby_Indicator, HIGH);
      if(UC_pressure < 2){                                //if UC chamber vacuum...
     is low (membrane down)
        Pump = HIGH;                                      //turn on pump
        UC_vacuum_valve = HIGH;                           //close upper chamber ...
    vacuum valve
      }
      if(LC_pressure > 2){                                //if there is vacuum ...
    in lower chamber
        LC_bleed_valve = HIGH;                            //open lower chamber ...
    vacuum valve
        bleed_start = millis();                           //start bleed time
      }
      if(UC_pressure > 10){                               //if UC chamber vacuum...
     is back up (membrane up)
        Pump = LOW;                                       //turn off pump
        UC_vacuum_valve = LOW;                            //close upper chamber ...
    vacuum valve
      }
      if(digitalRead(I0_1) == LOW){
```

```
        Pump = LOW;                                     //turn off pump
        UC_vacuum_valve = LOW;                          //close upper chamber ...
  vacuum valve
      }

      if(digitalRead(I0_0) == HIGH){                   //if start button is ...
  pushed
        getFilename(filename);                          //get file name
        Serial.println(filename);                       //print file name
        WriteSettings();                                //write settings to SD...
   card
        start_time = millis();                          //record start time
        state = 1;                                      //if start button is ...
  HIGH then set state to 1
        UC_vacuum_valve = HIGH;                         //open upper chamber ...
  vacuum valve
        LC_vacuum_valve = HIGH;                         //open upper chamber ...
  vacuum valve
        Run_Indicator = HIGH;                           //eluminate run ...
  indicator
        Pump = HIGH;                                    //turn on pump
        LC_bleed_valve = LOW;                           //close lower chamber ...
  bleed valve
      }
    }

    else{                                               //if not in standby
      Pump = LOW;                                       //make sure pump is ...
  off
      UC_vacuum_valve = LOW;                            //close upper chamber ...
  vacuum valve
      digitalWrite(Stndby_Indicator, LOW);              //turn off standby ...
  indicator
    }
  }

  if(digitalRead(I0_2) == HIGH){                        //if stop button is ...
    pressed
    EPR_SP = 400;                                       //set Electro-...
    Pneumatic Pressure Regulator setpoin to 0
    state = 4;                                          //go to state 4
  }

  //keep track of run time
  run_time = millis() - start_time;

  //start of program --------------------------------
  switch (state) {
  case 0:                                               //standby
    Run_Indicator = LOW;                                //de-eluminate run ...
    indicator
    Error_Indicator = LOW;                              //de-eluminate run ...
    indicator
    LC_vacuum_valve = LOW;                              //close lower chamber ...
    vacuum valve
    EPPR = LOW;                                         //turn off the Electro...
    -Pneumatic Pressure Regulator
    EPR_SP = 0;                                         //set Electro-...
    Pneumatic Pressure Regulator setpoint to 0
    run_time = 0;                                       //zero run time
    evac_ramp_set = 0;                                  //zero laminaton ramp ...
    set value
    evac_start_flag = false;                            //reset evacuation ...
    start flag
    break;
  case 1:                                               //program start
                                                        //try to keep the ...
    vacuum in the upper chamber higher (membrane up)
    if(Pressure_diff < (evac_press_diff - evac_press_hyst)){
```

```
      UC_vacuum_valve = LOW;                          //open upper chamber ...
    vacuum valve
    }
    if(Pressure_diff > (evac_press_diff + evac_press_hyst)){
      UC_vacuum_valve = HIGH;                         //open upper chamber ...
    vacuum valve
    }

    if(evac_ramp_time > 0){                           //if a evacuation ramp...
     rate is set
      evac_ramp = evac_done_pressure / (evac_ramp_time*2);  //calculate △ P ...
    for evacuation ramp
      if(run_time > (last_ramp_time + 499)){         //after 1 second has ...
    elapsed
        last_ramp_time = run_time;                    //save new time
        evac_ramp_set += evac_ramp;                   //add ramp step to ...
    setpoint
        if(LC_pressure > evac_ramp_set){              //if process is above ...
    setpoint
          LC_vacuum_valve = LOW;                      //close lower chamber ...
    vacuum valve
        }
        else{                                         //if process is under ...
    setpoint
          LC_vacuum_valve = HIGH;                     //open lower chamber ...
    vacuum valve
        }
      }
    }
    else{                                             //if ramp controll is ...
    not activated
      if(Pressure_diff < 1 ){                         //if pressure diff is ...
    smaler then 1 (membrane atliest not down)
        LC_vacuum_valve = HIGH;                       //open lower chamber ...
    vacuum valve
      }
    }
    if(Pressure_diff > evac_press_hyst){              //if pressure ...
    difference is very much positive (membrane pushed down)
      LC_vacuum_valve = LOW;                          //close lower chamber ...
    vacuum valve
    }

    if(LC_pressure > evac_done_pressure && evac_start_flag == false){//if ...
    lower chamber is at decent vacuum: start evacuation time
      evac_start = run_time;
      evac_start_flag = true;                         //set flag that ...
    evacuation has started
    }

    if(run_time ≥ evac_time + evac_start){           //if evac time is ...
    elapsed
      last_ramp_time = run_time;                      //save run time
      state = 2;                                      //go to state two
      EPR_SP = 400;                                   //put a ofset on the ...
    ERP_SP to speed it up a bit
    }
    break;
  case 2:
    UC_vacuum_valve = LOW;                            //close upper chamber ...
    vacuum valve
    LC_vacuum_valve = HIGH;                           //open lower chamber ...
    vacuum valve (to make sure its open)
    EPPR = HIGH;                                      //turn on the Electro-...
    Pneumatic Pressure Regulator

    lamination_ramp = lamination_pressure / lamination_rise_time;    //...
    calculate △ P for lamination ramp

```

```
    if(run_time > (last_ramp_time + 999)){              //after 1 second has ...
    elapsed
      last_ramp_time = run_time;                         //save new time
      EPR_SP = EPR_SP + lamination_ramp;                 //add lamination ramp ...
    to setpoint
    }
    if(Pressure_diff + 20 ≥ lamination_pressure){     //if setpoint is bigger...
     or equal to lamination_pressure
      EPR_SP = lamination_pressure;                      //write exact value to...
     setpoint
      Lamination_Start_time = run_time;                  //save run time
      state = 3;                                         //go to state 3
    }
    break;
  case 3:
    if(abs(lamination_pressure - Pressure_diff) > 6) EPR_SP = ...
    lamination_pressure + 6 * (lamination_pressure - Pressure_diff);       //...
    if ofset, add it to the setpoint mitiplied by 6
    if(abs(lamination_pressure - Pressure_diff) ≤ 2) EPR_SP = ...
    lamination_pressure;                                                   //...
    write exact value to setpoint
    if(run_time ≥ (Lamination_Start_time + lamination_time)){              ...
                                                               //wait for ...
    lamination time to pass
      state = 4;                                         //go to state 4
    }
    break;
  case 4:
    LC_vacuum_valve = LOW;                               //close lower chamber ...
    vacuum valve
    LC_bleed_valve = HIGH;                               //open lower chamber ...
    bleed valve
    EPPR = HIGH;                                         //turn on the Electro-...
    Pneumatic Pressure Regulator
    EPR_SP = -(Pressure_diff - UC_stop_pressure);      //to get the membrane ...
    up
    if(UC_pressure ≤ UC_stop_pressure){               //if upper chamber has ...
    reached stop pressure
      EPPR = LOW;                                        //turn off the Electro...
    -Pneumatic Pressure Regulator
      EPR_SP = 0;                                        //set Electro-...
    Pneumatic Pressure Regulator setpoin to 0
    }

    if(UC_pressure < 2){                                 //if upper chamber ...
    vacuum get to low (membrane down)
      UC_vacuum_valve = HIGH;                            //open upper chamber ...
    vacuum valve
    }
    if(UC_pressure > 8){                                 //if vacuum is back up...
     again (membrane up)
      UC_vacuum_valve = LOW;                             //close upper chamber ...
    vacuum valve
    }

    if(LC_pressure ≤ LC_stop_pressure){               //if lower chamber ...
    pressure is lower than or equal to LC_stop_pressure
      post_start_time = run_time;                        //save run time
      state = 5;                                         //go to state 5
    }
    break;
  case 5:
    if(run_time ≥ (post_start_time + post_bleed_time)){    //if post bleed ...
    time has passed
      LC_bleed_valve = LOW;                              //close lower chamber ...
    bleed valve
      Pump = LOW;                                        //turn off pump
      state = 0;                                         //go back to standby ...
    mode
```

```
      print_end();                                    //print program end serial stuff
    }
    break;
  default:
    // if nothing else matches, do the default
    // Error state
    state = 0;
    break;
  }

  //end of program ---------------------------------

  write_outputs();                                    //write outputs
  if(millis() >= last_millis + 500){                  //if enough time has passed
    if(debug == true && state != 0){                  //and program and debug is true
      print_serial();                                 //serial print info
      write_SD();                                     //write log data to SD card
    }

    read_temp();                                      //request temperature reading from PID controller
    last_millis = millis();
  }
  temp_response();                                    //check if PID controller has answered
  delay(100);                                         //slow the loop a bit

}//loop end


//=====================================

void write_outputs(){
  digitalWrite(R0_1, Run_Indicator);                  //set run indicator
  digitalWrite(R0_2, Error_Indicator);                //set run indicator
  digitalWrite(R0_3, Stndby_Indicator);               //set run indicator
  digitalWrite(R0_4, LC_vacuum_valve);                //set lower chamber vacuum valve
  digitalWrite(R0_5, UC_vacuum_valve);                //set upper chamber vacuum valve
  digitalWrite(R0_6, LC_bleed_valve);                 //set lower chamber bleed valve
  analogWrite(A0_0, constrain(map(EPR_SP, 0, 2020, 0, 52), 0, 125));//set Electro-Pneumatic Pressure Regulator setpoint (5V = 5 bar, 255/10 = 25.5/V, 1V = 1000mBar, 25=0.98V=980mBar)
  digitalWrite(R0_7, EPPR);                           //turn on or off the Electro-Pneumatic Pressure Regulator
  digitalWrite(R0_8, Pump);
}

//=============

void getFilename(char *filename) {


 RTC.read(); int year = RTC.getYear(); int month = RTC.getMonth(); int day = RTC.getMonthDay(); int hour = RTC.getHour(); int minute = RTC.getMinute();
  filename[0] = '2';
  filename[1] = '0';
```

```
  filename[2] = '2';
  filename[3] = year%10 + '0';
  filename[4] = '/';
  filename[5] = day/10 + '0';
  filename[6] = day%10 + '0';
  filename[7] = month/10 + '0';
  filename[8] = month%10 + '0';
  filename[9] = hour/10 + '0';
  filename[10] = hour%10 + '0';
  filename[11] = minute/10 + '0';
  filename[12] = minute%10 + '0';
  filename[13] = '.';
  filename[14] = 'C';
  filename[15] = 'S';
  filename[16] = 'V';

  foldername[0] = '2';
  foldername[1] = '0';
  foldername[2] = (year/10)%10 + '0';
  foldername[3] = year%10 + '0';

    //check if folder exists
  if (!SD.exists(foldername)) {
    Serial.print("Creating new folder: ");
    Serial.println(foldername);
    SD.mkdir(foldername);
  }


  return;
}

//=============

void WriteSettings(){
  //write files to SD card
  if(Log == true){
    // make a string for assembling the data to log:
    String dataString = "";

    dataString +=("Evacuation ramp time ");
    dataString += String(evac_ramp_time);
    dataString +=(" seconds");
    dataString += ",";
    dataString += ("Settings are: ");
    dataString +=("Evacuation time ");
    dataString += String(evac_time/1000);
    dataString += (" seconds");
    dataString += ",";
    dataString += ("Lamination ramp time ");
    dataString += String(lamination_rise_time);
    dataString += (" seconds");
    dataString += ",";
    dataString += ("Lamination time ");
    dataString += String(lamination_time/1000);
    dataString += (" seconds");
    dataString += ",";
    dataString += ("Lamination pressure ");
    dataString += String(lamination_pressure);
    dataString += (" mBar");
    dataString += ",";
    dataString += ("Temperature setpoint ");
    dataString += String(temp_sp);
    dataString += (" C");
    dataString += ",";


    // open the file. note that only one file can be open at a time,
    // so you have to close this one before opening another.
    File dataFile = SD.open(filename, FILE_WRITE);
```

```
    // if the file is available, write to it:
    if (dataFile) {
      dataFile.println(dataString);
      dataFile.close();
    }

    //print second row with file format
    dataString = "";      //empty string

    dataString += "state";
    dataString += ",";
    dataString += "UC_pressure";
    dataString += ",";
    dataString += "LC_pressure";
    dataString += ",";
    dataString += "Pressure_diff";
    dataString += ",";
    dataString += "run_time/1000";
    dataString += ",";
    dataString += "UC_vacuum_valve";
    dataString += ",";
    dataString += "LC_vacuum_valve";
    dataString += ",";
    dataString += "EPR_SP";
    dataString += ",";
    dataString += "avg_temp";
    dataString += ",";

    // open the file. note that only one file can be open at a time,
    // so you have to close this one before opening another.
    dataFile = SD.open(filename, FILE_WRITE);

    // if the file is available, write to it:
    if (dataFile) {
      dataFile.println(dataString);
      dataFile.close();
    }

  }
}

//============

//prints serial data at start
void print_start(){
  Serial.println("Mito laminator recipe maker!");
  Serial.println("Enter data in this style < evacuation ramp time, ...
    evacuation time, lamination ramp time, lamination time, lamination ...
    pressure, temperature setpoint>  ");
  Serial.println("Times are in secconds, pressure in mBar, temperature in ...
    Celsius");
  Serial.print("For example < ");
  Serial.print(evac_ramp_time);
  Serial.print(", ");
  Serial.print(evac_time/1000);
  Serial.print(", ");
  Serial.print(lamination_rise_time);
  Serial.print(", ");
  Serial.print(lamination_time/1000);
  Serial.print(", ");
  Serial.print(lamination_pressure);
  Serial.print(", ");
  Serial.print(temp_sp);
  Serial.println("> ");
  Serial.println();
  Serial.println("Current settings are:");
  Serial.print("Evacuation ramp time ");
  Serial.print(evac_ramp_time);
  Serial.println(" seconds");
```

```
  Serial.print("Evacuation time ");
  Serial.print(evac_time/1000);
  Serial.println(" seconds");
  Serial.print("Lamination ramp time ");
  Serial.print(lamination_rise_time);
  Serial.println(" seconds");
  Serial.print("Lamination time ");
  Serial.print(lamination_time/1000);
  Serial.println(" seconds");
  Serial.print("Lamination pressure ");
  Serial.print(lamination_pressure);
  Serial.println(" mBar");
  Serial.print("Temperature setpoint ");
  Serial.print(temp_sp);
  Serial.println(" C");
  Serial.println("");
}

//============

//prints serial data at end of program
void print_end(){
  Serial.println("");
  Serial.println("Program is done!");
  Serial.println("");
  Serial.println("To change parameters:");
    Serial.println("Enter data in this style < evacuation ramp time, ...
    evacuation time, lamination ramp time, lamination time, lamination ...
    pressure, temperature setpoint>  ");
  Serial.println("Times are in secconds, pressure in mBar, temperature in ...
    Celsius");
  Serial.print("For example < ");
  Serial.print(evac_ramp_time);
  Serial.print(", ");
  Serial.print(evac_time/1000);
  Serial.print(", ");
  Serial.print(lamination_rise_time);
  Serial.print(", ");
  Serial.print(lamination_time/1000);
  Serial.print(", ");
  Serial.print(lamination_pressure);
  Serial.print(", ");
  Serial.print(temp_sp);
  Serial.println("> ");
  Serial.println();
  Serial.println("Current settings are:");
  Serial.print("Evacuation ramp time ");
  Serial.print(evac_ramp_time);
  Serial.println(" seconds");
  Serial.print("Evacuation time ");
  Serial.print(evac_time/1000);
  Serial.println(" seconds");
  Serial.print("Lamination ramp time ");
  Serial.print(lamination_rise_time);
  Serial.println(" seconds");
  Serial.print("Lamination time ");
  Serial.print(lamination_time/1000);
  Serial.println(" seconds");
  Serial.print("Lamination pressure ");
  Serial.print(lamination_pressure);
  Serial.println(" mBar");
  Serial.print("Temperature setpoint ");
  Serial.print(temp_sp);
  Serial.println(" C");
  Serial.println("");
}

//============

//prints serial data during program
```

```
void print_serial (){
  Serial.print("state = ");
  Serial.print(state);
  Serial.print("\t UC_pressure = ");
  Serial.print(UC_pressure);
  Serial.print("\t LC_pressure = ");
  Serial.print(LC_pressure);
  Serial.print("\t Pressure_diff = ");
  Serial.print(Pressure_diff);
  Serial.print("\t run_time = ");
  Serial.print(run_time/1000);
  Serial.print("\t UC_vacuum_valve = ");
  Serial.print(UC_vacuum_valve);
  Serial.print("\t LC_vacuum_valve = ");
  Serial.print(LC_vacuum_valve);
  Serial.print("\t LC_bleed_valve = ");
  Serial.print(LC_bleed_valve);
  Serial.print("\t EPR_SP = ");
  Serial.print(EPR_SP);
  Serial.print("\t avg_temperature = ");
  Serial.println(avg_temp);
  Serial.println("");
}

//==============

void write_SD(){
  //write files to SD card
  if(Log == true){
    // make a string for assembling the data to log:
    String dataString = "";
    dataString += String(state);
    dataString += ",";
    dataString += String(UC_pressure);
    dataString += ",";
    dataString += String(LC_pressure);
    dataString += ",";
    dataString += String(Pressure_diff);
    dataString += ",";
    dataString += String(run_time/1000);
    dataString += ",";
    dataString += String(UC_vacuum_valve);
    dataString += ",";
    dataString += String(LC_vacuum_valve);
    dataString += ",";
    dataString += String(EPR_SP);
    dataString += ",";
    dataString += String(avg_temp);
    dataString += ",";

    // open the file. note that only one file can be open at a time,
    // so you have to close this one before opening another.
    File dataFile = SD.open(filename, FILE_WRITE);

    // if the file is available, write to it:
    if (dataFile) {
      dataFile.println(dataString);
      dataFile.close();
    }
  }
}

//==============

void read_temp(){
 switch (channel) {
  case 0:                                                    //only happens after writing register (to dump response)
    master.readHoldingRegisters(1, 0x1001, 0x01);
    break;
```

```
  case 1:
    master.readHoldingRegisters(1, 0x1001, 0x01);
    break;
  case 2:
    master.readHoldingRegisters(1, 0x1002, 0x01);
    break;
  case 3:
    master.readHoldingRegisters(1, 0x1003, 0x01);
    break;
  case 4:
    master.readHoldingRegisters(1, 0x1004, 0x01);
    break;
  case 5:
    master.readHoldingRegisters(2, 0x1001, 0x01);
    break;
  case 6:
    master.readHoldingRegisters(2, 0x1002, 0x01);
    break;
  case 7:
    master.readHoldingRegisters(2, 0x1003, 0x01);
    break;
  case 8:
    master.readHoldingRegisters(2, 0x1004, 0x01);
    break;
  }
}

//============

void temp_response(){
    if (master.isWaitingResponse()) {        // Check available responses
      ModbusResponse response = master.available();
      if (response) {
        if (response.hasError()) {
        // There is an error. You can get the error code with response.getErrorCode()
        } else {
        // Response ready: print the read holding registers
        for (int i = 0; i < 1; ++i) {
          temp_read = response.getRegister(i);

          if(channel > 0) temp = temp + temp_read;
          channel = channel +1;        //add up temperatures

          if(channel > 8){        //and take averiges after all 8
            channel = 1;
            avg_temp = temp/80;
            temp = 0;
            if(change > 0){        //if setpoint has changed
              delay(1000);
              switch (channel_set) {
              case 1:
                master.writeSingleRegister(1, 0x000A, temp_sp*10);
                break;
              case 2:
                master.writeSingleRegister(1, 0x0012, temp_sp*10);
                break;
              case 3:
                master.writeSingleRegister(1, 0x001A, temp_sp*10);
                break;
              case 4:
                master.writeSingleRegister(1, 0x0022, temp_sp*10);
                break;
              case 5:
                master.writeSingleRegister(2, 0x000A, temp_sp*10);
                break;
```

```
                case 6:
                  master.writeSingleRegister(2, 0x0012, temp_sp *10);
                  break;
                case 7:
                  master.writeSingleRegister(2, 0x001A, temp_sp *10);
                  break;
                case 8:
                  master.writeSingleRegister(2, 0x0022, temp_sp *10);
                  break;
                }
                channel_set = channel_set +1;
                channel = 0;

                if(channel_set > 8){
                  channel_set = 1;
                  change = change +1;
                  if(change > 2){
                  change = 0;
                  }
                }
              }
            }
          }
        }
      }
  }
}

//=============

void recvWithStartEndMarkers() {
    static boolean recvInProgress = false;
    static byte ndx = 0;
    char startMarker = '<';
    char endMarker = '>';
    char rc;

    while (Serial.available() > 0 && newData == false) {
        rc = Serial.read();

        if (recvInProgress == true) {
            if (rc != endMarker) {
                receivedChars[ndx] = rc;
                ndx++;
                if (ndx >= numChars) {
                    ndx = numChars - 1;
                }
            }
            else {
                receivedChars[ndx] = '\0'; // terminate the string
                recvInProgress = false;
                ndx = 0;
                newData = true;
            }
        }

        else if (rc == startMarker) {
            recvInProgress = true;
        }
    }

    if (newData == true) {
    strcpy(tempChars, receivedChars);
        // this temporary copy is necessary to protect the original data
        // because strtok() used in parseData() replaces the commas with \0
    parseData();
    showParsedData();
    newData = false;
  }
}
```

```
//============

void parseData() {      // split the data into its parts

    char * strtokIndx; // this is used by strtok() as an index

    strtokIndx = strtok(tempChars,",");      // get the first part - the ... string
    evac_ramp_time = atoi(strtokIndx);     // convert this part to an ... integer

    strtokIndx = strtok(NULL, ","); // this continues where the previous ... call left off
    evac_time = 1000 * atol(strtokIndx);     // convert this part to an long

    strtokIndx = strtok(NULL, ","); // this continues where the previous ... call left off
    lamination_rise_time = atoi(strtokIndx);     // convert this part to an ... integer

    strtokIndx = strtok(NULL, ",");
    lamination_time = 1000 * atol(strtokIndx);     // convert this part to ... an long

    strtokIndx = strtok(NULL, ","); // this continues where the previous ... call left off
    lamination_pressure = atoi(strtokIndx);     // convert this part to an ... integer

    strtokIndx = strtok(NULL, ","); // this continues where the previous ... call left off
    temp_sp = atoi(strtokIndx);     // convert this part to an integer

    lamination_pressure = constrain(lamination_pressure, 0, 950);   //keep ... values whitin limits

    //write EEPROM values
    EEPROM.writeLong(0,evac_time);
    EEPROM.writeLong(4,lamination_time);
    EEPROM.writeInt(8,lamination_rise_time);
    EEPROM.writeInt(10,lamination_pressure);
    EEPROM.writeInt(12,temp_sp);
    EEPROM.writeInt(14,evac_ramp_time);

    change = 1;                                          //set flag for data ... change
}

//============

void showParsedData() {
    Serial.print("Evacuation ramp time ");
    Serial.print(evac_ramp_time);
    Serial.println(" seconds");
    Serial.print("Evacuation time ");
    Serial.print(evac_time/1000);
    Serial.println(" seconds");
    Serial.print("Lamination ramp time ");
    Serial.print(lamination_rise_time);
    Serial.println(" seconds");
    Serial.print("Lamination time ");
    Serial.print(lamination_time/1000);
    Serial.println(" seconds");
    Serial.print("Lamination pressure ");
    Serial.print(lamination_pressure);
    Serial.println(" mBar");
    Serial.print("Set temperature ");
    Serial.print(temp_sp);
    Serial.println(" C");
```

```
        Serial.println("");
}
```

C Log file

Evacuation ramp time 0 seconds;Settings are: Evacuation time 1200 seconds;;;;;;;
Lamination ramp time 60 seconds;Lamination time 5400 seconds;;;;;;;
Lamination pressure 920 mBar;Temperature setpoint 85 C;;;;;;;
state;UC pressure;LC pressure;Pressure diff;run time/1000;UC vac valve;LC vac valve;EPR SP;avg temp
1;9;0;-9;0;1;1;0;85
1;30;19;-11;1;1;1;0;85
1;57;46;-11;1;1;1;0;85
1;82;72;-10;2;1;1;0;85
1;107;96;-11;3;1;1;0;85
1;131;119;-12;3;1;1;0;85
1;156;144;-12;4;1;1;0;85
1;178;165;-13;5;1;1;0;85
1;199;184;-15;5;0;1;0;85
1;209;198;-11;6;0;1;0;85
1;219;209;-10;7;0;1;0;85
1;230;221;-9;7;0;1;0;85
1;240;233;-7;8;0;1;0;85
1;251;245;-6;8;0;1;0;85
1;268;260;-8;9;1;1;0;85
1;290;282;-8;10;1;1;0;85
1;309;300;-9;11;1;1;0;85
1;326;317;-9;11;1;1;0;85
1;343;334;-9;12;1;1;0;85
1;359;351;-8;13;1;1;0;85
1;375;367;-8;13;1;1;0;85
1;390;382;-8;14;1;1;0;85
1;406;396;-10;15;1;1;0;85
1;421;411;-10;15;1;1;0;85
1;436;425;-11;16;1;1;0;85
1;471;458;-13;18;1;1;0;85
1;484;472;-12;18;1;1;0;85
1;497;484;-13;19;1;1;0;85
1;509;495;-14;19;1;1;0;85
1;519;507;-12;20;0;1;0;85
1;525;514;-11;21;0;1;0;85
1;531;522;-9;22;0;1;0;85
1;538;529;-9;22;0;1;0;85
1;544;536;-8;23;0;1;0;85
1;550;543;-7;24;0;1;0;85
1;561;552;-9;24;1;1;0;85
1;572;565;-7;25;1;1;0;85
1;600;590;-10;26;1;1;0;84
1;610;601;-9;27;1;1;0;84
1;619;609;-10;28;1;1;0;84
1;628;619;-9;28;1;1;0;84
1;638;628;-10;29;1;1;0;84
1;646;637;-9;30;1;1;0;84
1;655;645;-10;30;1;1;0;84
1;663;654;-9;31;1;1;0;84
1;672;661;-11;32;1;1;0;84
1;680;668;-12;32;1;1;0;84
1;688;677;-11;33;1;1;0;84