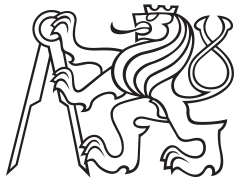


Bachelor Thesis



**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Measurement**

Multichannel load station controlled by LabVIEW

Ondřej Pavlín

Supervisor: Ing. Jan Sobotka, Ph.D.

Field of study: Cybernetics and Robotics

May 2024

I. Personal and study details

Student's name: **Pavlín Ondřej** Personal ID number: **507664**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Measurement**
Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Multichannel load control by LabVIEW

Bachelor's thesis title in Czech:

řízení vícekanálové zátěže v LabVIEW

Guidelines:

The goal is to design and build a complete measurement rack station capable of configuring resistive and inductive loads. Required Loads are in the range of 0 A to 2 A with 100 mA steps on all 18 channels of the resistive load. The user interface will be provided by a computer communicating with the other electronics via Ethernet. The control software will be programmed in LabVIEW. The work includes the design and implementation of the rack housing. Design and implement suitable overload protection. The system will be equipped with a safety light and a STOP button.

Bibliography / sources:

- [1] Guru, Bhag S. Hiziro lu, Hüseyin R.. (2001). Electric Machinery and Transformers (3rd Edition). Oxford University Press. Retrieved from <https://app.knovel.com/hotlink/toc/id:kpEMTE0002/electric-machinery-transformers/electric-machinery-transformers>
[2] Rodríguez-Quiñonez, Julio César Real-Moreno, Oscar. (2022). Graphical Programming Using LabVIEW™ - Fundamentals and Advanced Techniques. Institution of Engineering and Technology (The IET). Retrieved from <https://app.knovel.com/hotlink/toc/id:kpGPULVIEO/graphical-programming/graphical-programming>
[3] National Instruments LabVIEW Wiki. <https://labviewwiki.org>

Name and workplace of bachelor's thesis supervisor:

Ing. Jan Sobotka, Ph.D. Department of Measurement FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **16.01.2024** Deadline for bachelor thesis submission: _____

Assignment valid until:

by the end of summer semester 2024/2025

Ing. Jan Sobotka, Ph.D.
Supervisor's signature

Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Acknowledgements

I want to thank Siemens for allowing me to work on this project, where I gained hands-on experience with professional laboratory equipment and was able to purchase many components and devices. I also owe a huge thank you to my supervisor, Jan Sobotka, for guiding me through this project. Finally, I also want to thank my parents for their financial support and my classmates for the opportunity to share our experiences and overcome academic challenges together.

Chtěl bych poděkovat společnosti Siemens za to, že mi umožnila pracovat na tomto projektu, kde jsem získal praktické zkušenosti s profesionálním laboratorním vybavením, a na který jsem mohl zakoupit mnoho potřebného materiálu. Velké díky patří také mému vedoucímu Janu Sobotkovi, který mi byl celý projekt nápomocný. V neposlední řadě chci také poděkovat svým rodičům za finanční podporu a svým spolužákům za možnost sdílet naše zkušenosti a společně překonávat studijní problémy.

Declaration

I declare that I have worked on the presented thesis independently and that I have listed all used resources in accordance with the Methodological Guideline on the Observance of Ethical Principles in the Preparation of university theses. Prague,

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 23. May 2024

Podpis

Abstract

This thesis presents the design and development of a standalone multichannel load station. It is capable of configuring load tests that help to verify the robustness and reliability of industrial PLC modules under development.

The thesis starts by outlining the electrical and mechanical hardware design, including the selection of devices.

Next, the software design of the control program is discussed using the LabVIEW engineering environment. It unifies all devices and includes features such as defining test queues, measuring and displaying real-time data and automatically generating an Excel report of performed tests.

Finally, a series of experiments was performed to verify and evaluate that the station and all its parts are functioning correctly, as designed.

Keywords: LabVIEW, resistive and inductive load, rack station, industrial automation, PLC, hardware, software, Siemens

Supervisor: Ing. Jan Sobotka, Ph.D.

Abstrakt

Tato práce pojednává o návrhu a vývoji ucelené vícekanálové stanice. Umožňuje konfigurovat zátěžové testy, které pomáhají ověřovat robustnost a spolehlivost vyvíjených průmyslových modulů PLC.

Teze začíná nastíněním návrhu elektrického a mechanického hardwaru včetně výběru vhodných přístrojů.

Dále se zabývá softwarovým návrhem řídicího programu s využitím programovatelného prostředí LabVIEW. Ten všechny přístroje sjednocuje a nabízí funkce jako je vytváření testovacích front, měření a zobrazování dat v reálném čase a automatické generování reportu o průběhu všech testů ve formě Excel dokumentu.

Nakonec byla provedena řada experimentů k ověření a vyhodnocení, že stanice a všechny její části fungují správně, jak bylo navrženo.

Klíčová slova: LabVIEW, rezistivní a induktivní zátěž, racková stanice, průmyslová automatizace, PLC, hardware, software, Siemens

Contents

1 Introduction	1	3.5 View Test Queue VI	44
2 Hardware design	3	3.5.1 Structure of the Test Queue	45
2.1 Requirements and instrument selection	3	3.6 Data View VI	46
2.1.1 Theoretical foundations	4	3.7 Help VI	49
2.1.2 Power Supplies	6	3.8 Integration of Devices	50
2.1.3 Electronic Loads	7	3.8.1 Electronic load control	51
2.1.4 Other devices	9	3.8.2 Power supply control	52
2.2 Design of Inductive Load unit	9	3.8.3 NI USB module control	53
2.2.1 Inductance value and its calibration	9	3.8.4 Excel report generation	54
2.2.2 Basic design	11	3.9 Conclusion	56
2.2.3 Inductor heating experiments	13	4 Experiments	57
2.2.4 Temperature regulation	15	4.1 Noise temperature measurement	57
2.2.5 Measuring Temperature	20	4.1.1 With integrated signal averaging	57
2.2.6 Fan control	24	4.1.2 Disabled signal averaging	58
2.2.7 PCB for temperature regulation	26	4.1.3 Temperature Regulation	60
2.2.8 Cooling tunnel with integrated power resistors	29	4.2 Testing entire station and running tests	64
2.3 Design of power supply and communication	29	5 Conclusion	67
2.3.1 Power and safety	30	6 Glossary and Abbreviations	69
2.3.2 Communication	32	Bibliography	71
2.4 Design of rack housing	33	A List of all used components	73
3 Software design	37	B Photos of the finished load station	75
3.1 Introduction	37		
3.2 Program concept	38		
3.3 Main structure	39		
3.4 Test Setup VI	42		

Figures

1.1 3D concept visualisation of the load station design	1	2.15 The cheaper Arduino UNO (left), source: [2] and the more robust Arduino PRO Portenta (right), source: [3]	18
2.1 PLC system with DQ modules with a typical rating of 0.5 A or 2 A	3	2.16 National Instruments DAQ module called USB-6001 (left) with its pinout showing arrangement of analog and digital pins (right) [A.2.6]	19
2.2 IO Link Master with 12 A rating	4	2.17 Diagram of simple temperature measurement using PT1000 resistance temperature detector and NI DAQ module.	21
2.3 Direction of the current loop through an inductive load represented by a RL circuit. On the left with the module's digital output switched ON, on the right side switched OFF.	5	2.18 The analog and linear temperature sensor TMP37 [A.0.1] with 20 mV/°C scale factor in TO-92 package [4]	22
2.4 Charging of inductor and the time constant	5	2.19 Low-noise, low-dropout voltage regulator [A.0.2] powering the analog temperature sensors	22
2.5 Laboratory climatic chamber WEISS-WK3 180/70, source: [1]	6	2.20 Visualisation of sampling the sensor output voltage at 400 Hz, with averaging to remove a 50 Hz noise.	23
2.6 Programmable Power Supply TDK Lambda GEN40-38 [A.2.2]	6	2.21 Schematic of temperature measurement design of inductive load using linear sensor TMP37 and voltage regulator TPS791	24
2.7 Electronic Load Master H&H PMLA [A.2.3]	7	2.22 Schematic of a sub-circuit controlling the 24 V fan by NI module digital output pin	25
2.8 Limitations of PMLA active load channels 1-16	8	2.23 Schematic of a sub-circuit stabilising the 5 V output of NI module to supply the TMP37 sensors	26
2.9 Design of inductive load with eight channels	11	2.24 Two-layer PCB design with 10 MOSFET sub-circuits and LDO circuit stabilising TMP37 power supply	27
2.10 Photo of the 1.15H inductor used for inductive load tests in the Siemens laboratory	12	2.25 Photo of an inductor with attached temperature sensor and fan connected to a simple PCB	28
2.11 Graph showing AC square signal used in real inductive load tests, and the corresponding RMS signal	14		
2.12 Inductor temperature under 1.41A DC load with the fan turned on at 0:15	14		
2.13 Cheaper bimetal disc with fixed switching temperature (left) and an expensive thermostat with adjustable switching characteristics (right)	16		
2.14 Circuit diagram of one inductive load channel including a bimetal temperature control	17		

2.26 Aluminium tunnel for effective cooling of the power resistor corresponding to inductive load channels	29	3.9 Block diagram of <code>Init_Devices.vi</code> establishing connection with all devices	51
2.27 Power distribution cabinet in a Siemens laboratory	30	3.10 Communication architecture change due to malfunctioning MOXA server	52
2.28 Three-phase station powering diagram with safety devices	31	3.11 Block diagram of <code>ReadAI_NI</code> reading voltages of the temperature sensors TMP37	54
2.29 Powering and communication scheme of the station	32	3.12 The first summary page of the automatically generated report . . .	54
2.30 Physical placement of devices in the rack cabinet	34	3.13 The second page corresponding to Test 1 with the measured data . . .	55
2.31 Finished panel of inductive load (9) mounted on the front side of station	35	3.14 Block diagram of the <code>CreateExcelReport.vi</code> with the code divided to four parts	56
3.1 NI LabVIEW logo	37	3.15 Example of few icons used in LabVIEW implementation to improve code readability	56
3.2 Basic concept of the LabVIEW program	38	4.1 Measured output signal from TMP37 sensor with its Standard Deviation	58
3.3 Screenshot of the <code>Main_Control</code> user interface. Only a part of the sub-panel is shown.	39	4.2 Measured output signal from TMP37 sensor without averaging approach	59
3.4 The <code>Main_Control</code> block diagram, the actual LabVIEW code.	40	4.3 Comparison of deviation for each measured sample for both approaches	60
3.5 The <code>Test_Setup</code> interface with control elements, loaded in the <code>Main_Control</code> sub-panel	42	4.4 Temperature regulation of inductors L_2 and L_7	61
3.6 Diagram of <code>Test Setup</code> VI code implemented as a State-Space Machine	43	4.5 Temperature regulation with loading of all inductors simultaneously	62
3.7 Front panel of <code>Data View</code> section, with switchable graph, Test information indicator and buttons for Excel report generation and admin login features	47	4.6 Long experiment of temperature regulation with loading all inductors	63
3.8 Admin settings window with device limits, state indicator and Status history, that only authorized operator can access	49	4.7 Photograph of the load station running the final experiment conducted for this thesis	66

B.1 The completed load station with a red light on, indicating it is turned on 75

B.2 Station power wiring with main switch and circuit breakers 76

B.3 Shelf behind the “Power & Communication” panel containing LAN server, PCB and NI module . 76

B.4 Assembled inductive load unit . 77

Chapter 1

Introduction

In the constantly evolving field of industrial automation, the reliability and durability of programmable logic controllers (PLCs) under harsh environmental conditions have become more important than ever. These devices are a fundamental part of controlling complex industrial processes and must endure a wide range of stress factors, such as extreme temperatures, humidity, and loads. Therefore, it is essential to develop automated tools like the "Multichannel Load Station", which aims to help evaluate the robustness and reliability of Siemens PLC equipment under harsh environmental conditions. This thesis outlines the concept, design, and implementation of this laboratory test station.

The multi-channel load station is built for demanding automatised resistive and inductive load tests. It has an easy-to-use architecture that enables testers to configure test sequences, monitor the tested device with real-time measurements and generate test reports for analysis. The unit's ability to simulate a broad range of operating scenarios with configurable loads ensures a comprehensive assessment of the PLC device's robustness. This load station will specifically test I/O and Digital Output (DQ) modules, which control critical industrial systems and require maximum reliability.

The central element of the Multichannel Load Station is a laptop that controls and monitors the entire testing process via the LabVIEW programming environment. This includes dynamic configuration of power supplies, precise control of electronic loads and temperature regulation of inductive load.



Figure 1.1: 3D concept visualisation of the load station design

It is essential to note that although the Multichannel Load Station is capable of simulating a variety of electrical load conditions, it does not replicate environmental stress factors such as temperature and humidity. These conditions are carefully created and managed in a separate climate chamber, ensuring that PLC devices are exposed to the worst limit conditions.

My thesis is structured to provide an overview of the multichannel load station, from the basic principles of its hardware and mechanical design to the extensive software implementation. The project aims to contribute to the field of industrial automation with a user-friendly yet comprehensive unit that significantly advances the reliability assessment of PLC equipment under harsh industrial conditions.

Chapter 2

Hardware design

This chapter discusses the electrical and mechanical design and development of the load station. I will begin by outlining the system requirements and highlighting the significance of conducting load tests on industrial PLC equipment. Next, we'll delve into the selection of instruments and the design of station powering and communication, including overcurrent and overload protection and emergency safety features. The chapter also covers the complexities of designing an inductive load unit, which presents many challenges. The mechanical and 3D designs will be slightly outlined at the end of the chapter. All the further discussed components and devices are listed in the appendix A, including links to their official websites.

2.1 Requirements and instrument selection

This section outlines the basic requirements for the system and the instruments chosen to meet them. In the context of Programmable Logic Controllers (PLCs), which are the heart of many industrial automation systems, two common types of modules that we'll encounter are DQ (Digital Output) modules and IO-Link I/O modules. These components are essential in interfacing the PLC with the real world, allowing it to receive inputs from and send outputs to various devices and sensors. The station requirements are primarily defined by the electrical characteristics of the PLC DQ and I/O modules that Siemens develops and sells, along with relevant technical standards and laboratory conditions.



Figure 2.1: PLC system with DQ modules with a typical rating of 0.5 A or 2 A



Figure 2.2: IO Link Master with 12 A rating

Configuring loads: Simulating maximum resistive and inductive loads is crucial for verifying the durability of PLC modules. These devices must handle specific loads, and their performance under worst-case conditions needs thorough validation before marketing. PLC modules, such as those referenced [5] [6], typically support a maximum single channel output of 0.5 A or 2 A at 28.8 V. Additionally, IO-Link Master modules [7] can handle up to 12 A input current, as other I/O devices can be connected to the Master module in series. This can seriously overheat and destroy specific components in the Master device due to the large current flow through the device and thus endanger the entire hypothetical controlled industrial system. The resistive load channels of the designed station should be, therefore, capable of automatically configuring at least these load values or higher.

Movable rack cabinet: The complete station should be built in a movable rack cabinet, commonly used in industrial and laboratory environments. The station would need to be moved across the laboratory and used with a specific climatic chamber. Such rack cabinets are durable, space-efficient, and their standardised dimensions are considered by most instrument manufacturers.

Control via laptop All instruments used should be unified and controlled by a laptop with a clear and user-friendly interface. Using a laptop also offers remote access to monitor the performing tests, further pushing the automation limitations and benefits.

■ 2.1.1 Theoretical foundations

When a resistive load is applied to a PLC module, energy dissipation occurs within the module, leading to increased temperatures on certain components, which can lead to reduced lifetime or destruction of components in the device. Every real component has its resistance, that converts the electric energy into heat that is dissipated into the environment. That energy dissipated over time can be expressed as

$$P = UI = I^2R \rightarrow W_R = Pt = I^2Rt \quad (2.1)$$

where P stands for resistor's power, W_R for energy, t for time and U , I , R for voltage, current and resistance respectively.

Furthermore, when switching an inductive load, high energy dissipation occurs on the protection diodes inside the module. As current flows through the inductor, energy accumulates on it in the form of an electromagnetic field. When the circuit is opened by the device, the coil resists changes in the electric current and dissipates the stored energy back into the circuit in an impact manner, creating a large voltage pulse and at the same time the current flowing through the coil fades gradually. The two diagrams 2.3 show

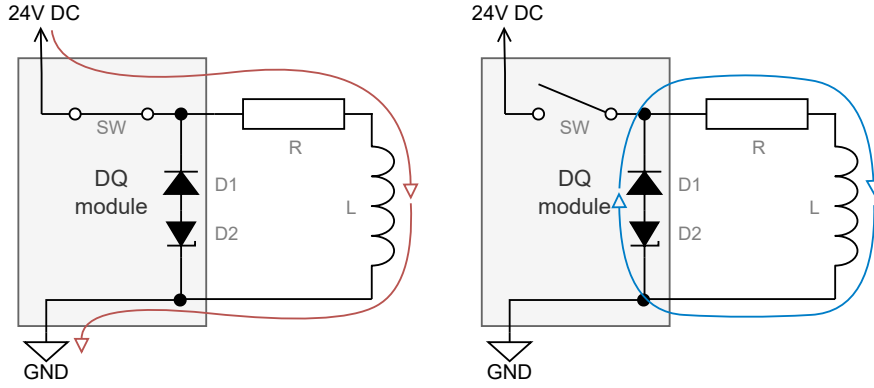


Figure 2.3: Direction of the current loop through an inductive load represented by a RL circuit. On the left with the module's digital output switched ON, on the right side switched OFF.

the current loops when the inductive load (represented by RL circuit) is switched on and off by a DQ PLC module. After switching off, the coil's accumulated energy is dissipated primarily on the protective diodes D1 and D2, which must withstand this load. Part of the energy is also dissipated on the coil series resistance (R), which also heats up the coil.

The energy accumulated on inductor can be expressed as [8, pp.45]

$$W_L = \int_0^{t_x} P dt = \int_0^{t_x} u i dt = L \int_0^{i_x} i di = \frac{1}{2} L i_x^2 \quad (2.2)$$

where L is the coil's inductance and i_x the passing current. The charging of a fully discharged coil can be described by an exponential function, in which the time constant τ plays an important role because it indicates how fast and dynamic the RL system is, and how long it takes to reach the steady state. The time constant determines the current flowing through the inductor by the following equation, as stated in [8, pp.115]. I will return to these dependencies later.

$$i_L(t) = I_0(1 - e^{-\frac{t}{\tau}})$$

$$i_L(\tau) = 0.63I_0 = 63\% \text{ of steady state current} \quad (2.3)$$

$$i_L(3\tau) = i_L(T_{0.95}) = 0.95I_0$$

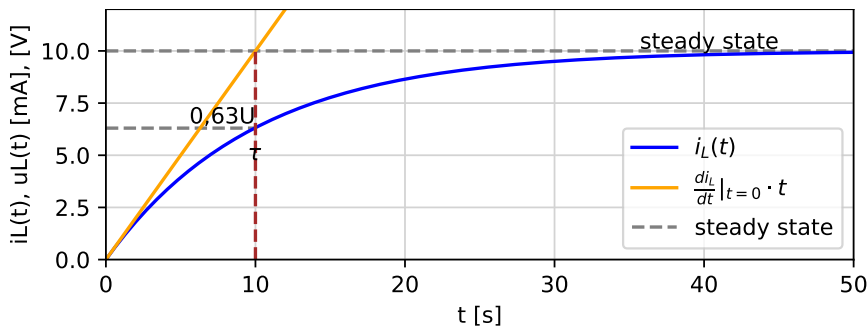


Figure 2.4: Charging of inductor and the time constant

The temperature inside of the PLC modules and on its specific crucial components has to be measured during the performed tests. The PLC is placed in a climatic chamber that ensures critical temperature and humidity, according to the operator's requirements. There are five different climatic chambers in our Siemens laboratory. One such chamber is shown in Figure 2.5, that is capable of setting temperatures from -70°C to $+180^{\circ}\text{C}$ and humidity control.

A lot of small temperature sensors are then placed and fixed inside the PLC plastic housing to monitor the temperatures of specific parts or components. Usually, a combination of thermocouples and RTDs (PT100) sensors is used. The operator then decides whether the maximum permissible temperature of the measured component inside of the module has been exceeded. If so, additional derating tests must be performed and the device's design should be changed or discussed with PLC developers. Testing PLC devices under the maximum allowed resistive and inductive loads is a rigorous approach to validate that the equipment can withstand operational demands and maintain reliable performance over its expected service life.

The detailed design of the inductive load unit will be further explained in section 2.2.



Figure 2.5: Laboratory climatic chamber WEISS-WK3 180/70, source: [1]

2.1.2 Power Supplies

To power the PLC equipment under test will be used “TDK Lambda Genesys 1520W” [A.2.2] power supply with output characteristic of 38 A/40 V/1520 W. The output range and accuracy from the instrument's datasheet are summarised in Table 2.1. The accuracy is expressed as a percentage of the rated output value.



Figure 2.6: Programmable Power Supply TDK Lambda GEN40-38 [A.2.2]

	Voltage	Current	Power
Range	0 V – 40 V	0 A – 38 A	0 W - 1520 W
Accuracy	$\pm 0.05\%$	$\pm 0.3\%$	-

Table 2.1: Range and accuracy of the power supply

The power supply is equipped with high-quality LabVIEW drivers, allowing straightforward implementation into the control program. It communicates with the control computer through either RS232 or RS485 serial standards.

If there is a need for station expansion in the future, these power supplies can be connected in series using a single RS485 cable, which will still require only one communication cable to the laptop.

In cases where all loads are fully used, the equipment will require a current greater than 38A. To address this concern, two units have been procured. The station's front panel will have the output power brought out to the front of the station, thus enabling the tester to connect it to specific PLC equipment. Given that the laptop is not equipped with a RS232 connector, a server that can convert the communication is required and discussed in section 2.1.4.

2.1.3 Electronic Loads

To meet the testing requirements of most PLC devices, I chose accurate and powerful electronic (active) loads H&H PMLA that can be purchased with arbitrarily configured channel units. It supports various modes such as constant current, voltage, resistance, and power, allowing precise control over load conditions for thorough evaluation of the device under test.

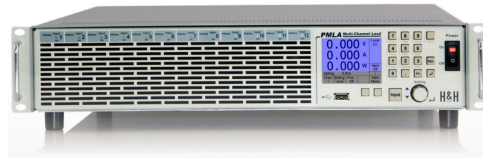


Figure 2.7: Electronic Load Master H&H PMLA [A.2.3]

Like the TDK-Lambda power supply, this device also offers LabVIEW drivers, allowing remote control over the load channels. From the total of 18 required channels, 16 were chosen less powerful (150 W / 30 A / 40 V) and the remaining 2 stronger (600 W / 120 A / 40 V). The technical datasheet of the instrument provides the ranges and accuracies, which are summarized in Table 2.2.

	Voltage	Current	Resistance	Power
Range (ch 1-16)	1 V - 40 V	0 A - 30 A	0.067 Ω - 133 Ω	0 W - 150 W
Range (ch 17-18)	1 V - 40 V	0 A - 120 A	0.017 Ω - 33 Ω	0 W - 600 W
Accuracy	$\pm 0.1\%$	$\pm 0.25\%$	$\pm 1.4\%$	$\pm 0.7\%$

Table 2.2: Ranges and accuracies of the setting value on the electronic load

Such a specific configuration of channels is possible thanks to separately purchased and installed units in the electronic load. This combination satisfies the testing requirements of most of the Siemens PLC equipment, including the common 2 A DQ modules [6] and also the stronger 12 A IO-Link modules [7]. It is important to note that the 150 W limit (and 600 W respectively) cannot be exceeded, and the current limits must be adjusted with respect to the output voltage of the tested PLC device. The correct and ideal power supply for PLC modules should be 24 V, based on the Siemens datasheets. However, Siemens allows this voltage to be exceeded by up to 20%. This leads to the necessity of testing PLC devices at the maximum voltage of 28.8 V, which still does not violate the manufacturer's datasheet values.

The physical dimensions of these 18 channels require the use of 2 separate PMLA devices which, similar to the power supplies, will be connected to each

other in series. Consequently, only one LAN cable will be used to connect the laptop to both electronic loads. The electronic load unit communicating directly with the laptop is referred to as the “Master” unit and it also has a small display with control buttons, and the second one as the “Slave” unit that lacks these additional features.

The User Manual [A.2.3] also commands the use of external fuses for each and every channel because the active load circuit does not have fuse protection. I assume that the channel will be loaded with 28.8 V, which is the maximum limit across almost all PLC modules. The current $I_{max_{16}}$ that will flow through the load when reaching the maximum allowed power limit for channels 1-16 is

$$I_{max_{16}} = P_{max}/U = 150 \text{ W}/28.8 \text{ V} = 5.2 \text{ A}$$

and similarly for the remaining stronger channels 17-18

$$I_{max_{18}} = P_{max}/U = 600 \text{ W}/28.8 \text{ V} = 20.8 \text{ A}.$$

These limitations are schematically shown in the figure 2.8, showing the maximum device output limits, and the maximum allowed current value for this specific project. I would use a combination of software and hardware over-

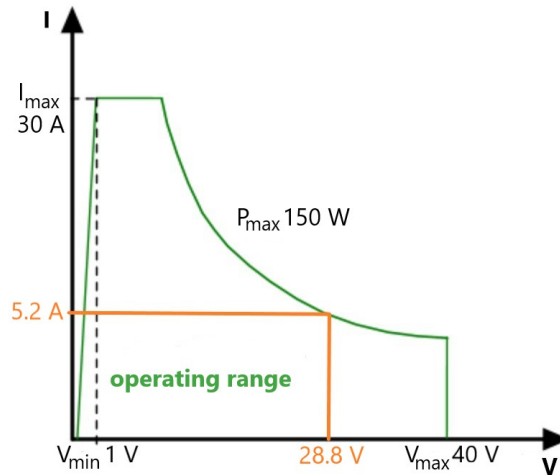


Figure 2.8: Limitations of PMLA active load channels 1-16

current protection. The LabVIEW program will have the software protection implemented that prevents the user from selecting a current higher than 5 A. In addition, a physical cartridge fuse will be used for comprehensive protection. The value of this external fuse should be chosen so that it allows normal operation of the system without unnecessary interruptions, but will interrupt the circuit if the current exceeds a safe level. It’s common practice to select a fuse with a rating slightly above the maximum system current to provide a safety margin. For example 15% above the limit. This accounts for possible inaccuracies in the current measurement or variations in the manufacturing of fuses and components. Thus, I have chosen the cartridge fuses with a current rating of

$$I_{fuse} = I_{max} \cdot 1.15 = 6 \text{ A}$$

for channels 1-16. For the stronger channels 17-18, I have selected fuses with a current rating of 20 A because the one with a 23.9 A rating was not available.

Thus the software channel protection would be implemented as 17 A, keeping the 15% safety margin.

■ 2.1.4 Other devices

- For controlling the entire station via LabVIEW program is used laptop HP ZBook Fury [A.2.1]. It has a single Ethernet RJ-45 connector and two USB 3.1 ports that can be used for communication with instruments. The laptop will sit on a pull-out drawer, and therefore, it is desirable to have as few communication cables connected to it as possible.
- A D-Link LAN server [A.2.4] will be used to connect the Laptop to multiple devices simultaneously via Ethernet, extending the laptop's capabilities.
- Additionally, a MOXA N-Port server [A.2.5] will be used to convert the LAN communication to RS232, because the chosen power supplies do not have a LAN port.

All components used in implementation are summarised in the appendix A.

■ 2.2 Design of Inductive Load unit

This section builds on the explained theoretical foundations 2.1.1 and describes the design process of a custom inductive load unit. It shows the final version and also discusses the discarded previous versions of temperature measurement, fan regulation circuitry and PCB design.

■ 2.2.1 Inductance value and its calibration

In this section will be explained why the inductive load does not need to be dynamically controlled and why a fixed inductance value should be used. The explanation is grounded in the definition of the time constant, earlier explained in 2.1.1, and the technical guidelines set forth by the IEC standard for electronic device testing.

Based on IEC standard 60947-5-1 [9] can be determined the value of inductive load for testing the worst possible inductive load conditions. It does not mention the exact inductance of the load but the time constant τ of the RL circuit representing the real coil in inductive load. The standard's annex C states the following quoted paragraphs:

“The special durability tests described in this annex are conducted at the discretion of the manufacturer. If the manufacturer declares a mechanical and/or electrical durability, the value shall correspond to the special tests described respectively in C.2 and/or C.3.”

“Electrical durability tests are carried out by operating the device under the conditions defined in Table C.1” [9]

To calibrate the inductive load utilized for testing our PLC equipment, it’s recommended in table C.1 to measure the time required to reach 95% of the steady-state current ($T_{0.95}$). This measurement should meet the following:

$$T_{0.95} = 6P = 6UI \quad (2.4)$$

Here, P refers to the power, while U and I represent the voltage and current of the steady state, respectively.

“The value $6P$ results from an empirical relationship which is found to represent most DC magnetic loads to an upper limit of $P = 50$ W, i.e. $6 \times P = 300$ ms. Loads having power consumption greater than 50 W are assumed to consist of smaller loads in parallel. Therefore, 300 ms is to be an upper value, irrespective of the power.” [9]

For an RL circuit that represents the inductive load, the time constant τ satisfies the following equations.

$$\tau = \frac{L}{R} \quad (2.5)$$

From the inductor’s settling function can be also written the relationship between the time constant τ and $T_{0.95}$.

$$\begin{aligned} i_L(t) &= I_0(1 - e^{-\frac{t}{\tau}}) \\ 0.95I_0 &= I_0(1 - e^{-\frac{T_{0.95}}{\tau}}) \\ 0.05 &= e^{-\frac{T_{0.95}}{\tau}} \\ \ln(0.05) &= -T_{0.95}/\tau \\ \tau &= \frac{1}{3}T_{0.95} \end{aligned}$$

The calibration process of the inductive load with respect to IEC [9] is carried out according to the following steps:

1. First, adjust the current flowing through the coil to 0.5 A by adding the appropriate resistor in series.
2. Next, capture the settling curve using an oscilloscope.
3. Finally, read the values of time constant τ and $T_{0.95}$ from the captured curve.
4. If $T_{0.95}$ reaches $6P$ in milliseconds with a maximum deviation of $\pm 10\%$, and τ equals $\frac{1}{3}6P$ in milliseconds with a maximum deviation of $\pm 20\%$, the load is considered properly calibrated.

I can then determine the load's inductance by using the formula

$$\tau = \frac{L}{R} \rightarrow L = \tau R$$

where τ denotes the RL circuit's time constant and R represents the overall resistance. This method is similarly applicable to 2 A, resulting in an inductance value very similar to the method with 0.5 A.

For a perfectly calibrated inductive load, the inductance L would reach the value of 1.15 H because of the following dependencies.

$$T_{0.95} = 6P = 6 \cdot 24 \cdot 2 = 288 \text{ ms}$$

$$\tau = T_{0.63} = \frac{1}{3}T_{0.95} = 96 \text{ ms}$$

$$R = \frac{U}{I} = \frac{24}{2} = 12\Omega$$

$$\tau = \frac{L}{R} \rightarrow L = \tau R = 0.096 \cdot 12 = 1.15 \text{ H}$$

The same formulae are applicable for the 0.5 A variant and the inductance comes out equal to 1.15 H again. The time constant τ of this RL circuit is different, but since the series resistance R is also different, it gives us the same inductance. The entire calibration is recorded in the attached Excel document `Coils_Calibration.xlsx`, including the calibration definition and its graphical representation.

2.2.2 Basic design

The inductive load unit consists of eight individual channels, each of which includes a coil with the inductance of 1.15 H, along with two adjustable power resistors. One resistor is calibrated to such resistance that there would flow 0.5 A of steady-state current through the load, while the other is set for a current of 2 A.

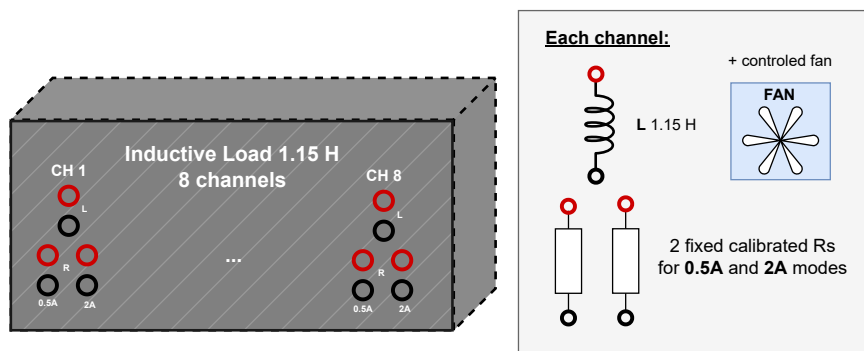


Figure 2.9: Design of inductive load with eight channels

The inductance value does not need to be changed dynamically in real time or between tests. All PLC devices with the ability to switch inductive loads

are tested using this inductive load, calibrated according to the standards outlined in [9]. The operator of the multichannel load station can use each inductive load channel in either 0.5 A or 2 A mode, or attach their own different resistor if a different current is required. This design is shown in the schematic image 2.9.

The inductor from Siemens laboratory consists of a thin copper wire wound on an iron transformer core as seen in Figure 2.10. The core is composed of two pieces, between which there is an air gap, the size of which significantly affects the inductance of the coil. By precisely adjusting this gap, the inductance can be calibrated to the desired value.



Figure 2.10: Photo of the 1.15H inductor used for inductive load tests in the Siemens laboratory

However, because copper has a high temperature coefficient α , the inductor's resistance also quite dramatically changes with the change in temperature. According to the article [10], the change in resistance ΔR with the change in temperature Δt can be expressed by the formula

$$\Delta R = \alpha R_0 \Delta t \quad (2.6)$$

where α is the copper material constant called temperature coefficient with a value of $3.92 \cdot 10^{-3} \text{K}^{-1}$, R_0 is the initial coil resistance, and Δt is the change in temperature.

It was experimentally measured by a digital multimeter that the coil winding resistance at 22°C is approximately 7.6Ω . This value really corresponds only to the coil resistance and not the whole impedance Z , because digital multimeters measure using direct current (DC). The reactive component of the impedance would only show up in an AC measurement. Thereafter, I can estimate how much the resistance of this coil changes when the temperature changes by 50°C for example.

$$\Delta R = 3.92 \cdot 10^{-3} \cdot 7.6 \cdot 50 = 1.5 \Omega$$

A temperature change of 50°C should therefore change the coil's resistance by approximately 1.5Ω , or 20% of the initial resistance value.

As the resistance drifts, the current through the inductive load fluctuates, which is undesirable when performing the tests, because an operator wants to rely on the station when performing the load tests. The inductance and current values should be precisely defined and stabilised values. Therefore, the coil's temperature must be somehow controlled so that the resistance drift is minimal.

■ 2.2.3 Inductor heating experiments

I conducted several experiments testing the coil's heating behaviour when passing a large current. One of them will be discussed below. The experiment is also recorded in the attached Excel document `Temp_Measurements.xlsx`.

PLC modules are usually tested by switching the inductive load on and off with 50% duty cycle and frequency 0.5 or 1 Hz. During the experiments, I was using a simple laboratory DC power supply. In order to simulate the same power dissipation, and thus simulating the same heating of the coil when using a DC supply, I needed to calculate the RMS (or effective) current value. The RMS value of a current is effectively the equivalent DC value that would produce the same amount of heat in a resistor as the AC current over one complete cycle. As the name RMS (Root Mean Square) suggests, the I_{RMS} definition 2.7 includes the root, mean and square operations

$$I_{RMS} = \sqrt{\frac{1}{T} \int_0^T (i(t))^2 dt} \quad (2.7)$$

where T represents the AC signal period and $i(t)$ the current value.

The rectangular signal representing switching inductive loads with a frequency of 0.5 Hz and 50% duty cycle is shown in Figure 2.11. The resulting signal does not have this rectangular shape because of the inductor's dynamic behaviour, but this is the signal that the PLC module attempts to create by switching the digital output.

Its corresponding RMS signal is shown in red, with its value computed as

$$I_{RMS} = \sqrt{\frac{1}{2} \int_0^2 (i(t))^2 dt} = \sqrt{\frac{1}{2} \left(\int_0^1 2^2 + \int_0^1 0^2 dt \right)} = \sqrt{\frac{1}{2} (4 + 0)} = \sqrt{2} \approx 1.41 \text{ A}$$

Both of these signals cause the same power and thermal dissipation on the inductor series resistance.

During the experiment, a two-channel power supply was used, and the first channel supplied the inductor with a constant current of 1.41A. Next, a 24 V axial fan was placed in front of the coil for cooling, which a simple switch could manually turn on and off. The second power supply channel fed the fan, which was set to 24 V and sank 150 mA. Using two thermocouples, I measured the temperature at the front and back of the coil winding. I also measured the temperature using a digital FLIR camera with emissivity set to 0.95 to see the temperature gradient on the copper winding surface.

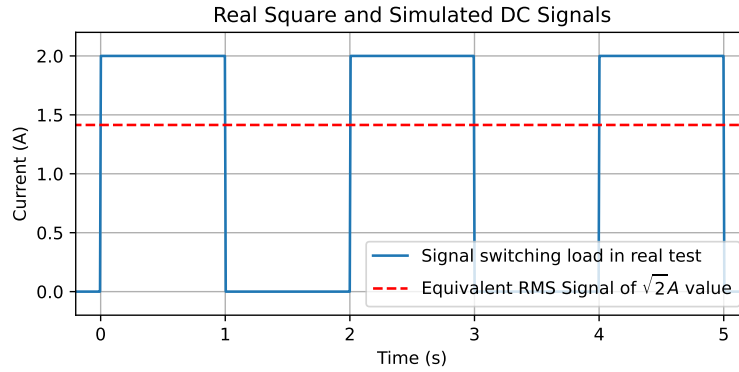


Figure 2.11: Graph showing AC square signal used in real inductive load tests, and the corresponding RMS signal

For one hour, the coil was heated by the passing current, and every five or ten minutes I measured its temperature and read the voltage across the coil from the constant current source. The 24 V axial fan was initially turned off and after 15 minutes was turned on, simulating a simple temperature regulation. The measured data are shown in the graph 2.12.

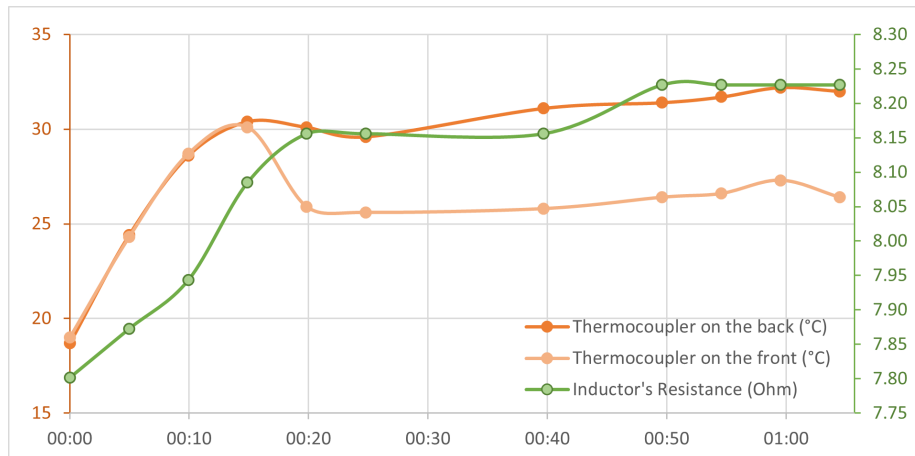


Figure 2.12: Inductor temperature under 1.41A DC load with the fan turned on at 0:15

The experiment results showed some properties of this particular inductor, which I will then consider when designing the temperature control. The test showed that the coil is inhomogeneously heated when cooled by the axial fan, with the difference of approximately 5°C. After the fan turned on and the temperature stabilised, the coil's resistance drifted by less than 0.1 Ω, with the temperature fluctuating by approximately 2°C. Additionally, with a total temperature change of 14°C, the resistance of the coil winding changed by 0.43 Ohm. The resistance of the coil copper winding should change by 0.40 Ω according to the theoretical estimate 2.6 when plugging in my specific values. That corresponds well with the measured results.

The conducted experiments also showed that the temperature operating point should be set above 30°C. It was observed that the actuator (fan) intervention would be more effective if the setpoint temperature is higher. That is because

the fresh air blown by the fan would be at room temperature and would more effectively cool down a hotter object, than an object that is only slightly warmer than the blown air. The temperature of the operating point was chosen at 45°C, being a compromise between a higher temperature for better actuator efficiency and a lower temperature for safety.

■ 2.2.4 Temperature regulation

After conducting a few experiments, it was evident that the coil winding temperature must be regulated to maintain constant resistance during tests. When switching the inductive load, the current of 0.5 A or 2 A is supposed to pass through the coil, but the fluctuation of the winding resistance would change the passing current magnitude.

Key observations and recommendations include:

- **Temperature indication:** It is also highly recommended to include a coil temperature indicator in the design. This can be very useful for the station operator to have information about the temperature on all coils to ensure that the temperature control is working correctly and that the temperature has stabilised. The temperature control system will not feature an additional heating element, requiring the operator to wait for the coil to naturally warm up to the desired temperature. Once this temperature is reached, the feedback control should maintain it.
- **Different temperature setpoint for each mode:** Conducted experiments indicated that a 0.5 A current cannot heat the coil to 45°C. It causes 16 times less power dissipation than the 2 A current, and it was observed that the coil winding temperature stabilised at approximately 24°C when it was 20°C in the climatized laboratory. I originally thought that this would require the implementation of an additional heating element, such as a power resistor, to heat up the coil. But then I realised I just needed to calibrate the resistor in series with the inductor cleverly. As you can see in the figure 2.9, both modes have their own resistor, calibrated to pass the specified current through the inductor when the temperature stabilises.

2 A mode:

- Inductor heated and regulated to 45°C.
- Inductor resistance around 8.3 Ω at that temperature setpoint.
- Required external series resistance of $\sim 6.1 \Omega$
- Total channel resistance of 14.4 Ω at the 45°C temperature to ensure 2 A at the PLC maximum voltage of 28.8 V.

0.5 A mode:

- Inductor temperature naturally self-stabilised at only 24°C.
- Inductor resistance around 7.5 Ω at that temperature.
- Required external series resistance of $\sim 50.1 \Omega$
- Total channel resistance of 57.6 Ω to ensure 0.5 A at the PLC maximum voltage of 28.8 V.

The following pages will cover the development of the temperature regulation system, including the reasons for discarding the earlier versions that were considered.

1. **Bimetal thermostat.** The first idea that came to my mind for designing a temperature regulating system involved using a bimetal thermostat. This switch makes use of a bimetallic strip that bends at different temperatures to complete or break an electrical circuit, thereby controlling the power supply based on temperature changes. There are different versions of thermostats available, ranging from cheaper ones with fixed temperature characteristics to more expensive ones that allow for adjustable switching temperature and hysteresis.



Figure 2.13: Cheaper bimetal disc with fixed switching temperature (left) and an expensive thermostat with adjustable switching characteristics (right)

For each channel of the inductive load, one separate bimetal switch would be needed. This switch would be thermally connected to the coil winding through a thermal paste for better sensitivity. This design has the advantage of being simple and straightforward, as it requires only a single component for both temperature measurement and fan switching. The bimetal disc is also small (with a diameter of 2 or 3 cm) and inexpensive (8 USD or 190 Kč), making it convenient when eight of them are needed for the inductive load unit. The following scheme 2.14 illustrates this design with an additional safety bimetal switch that would disconnect the inductor when a critical temperature is reached. The SITOP device is a 24 V DC stabilised power supply required for supplying the axial fans.

However, this method does not satisfy all requirements because it lacks an external temperature indicator. This means that the user cannot be sure that the regulation is working properly. Additionally, the temperature switching point and hysteresis cannot be adjusted during the design, making it more difficult to implement. It is not clear in advance at what temperature the control should stabilize the coil temperature for optimal performance, and this operating point will need to be experimentally determined and possibly changed during the design process. Calculating or simulating the heating and cooling processes is not easy because the

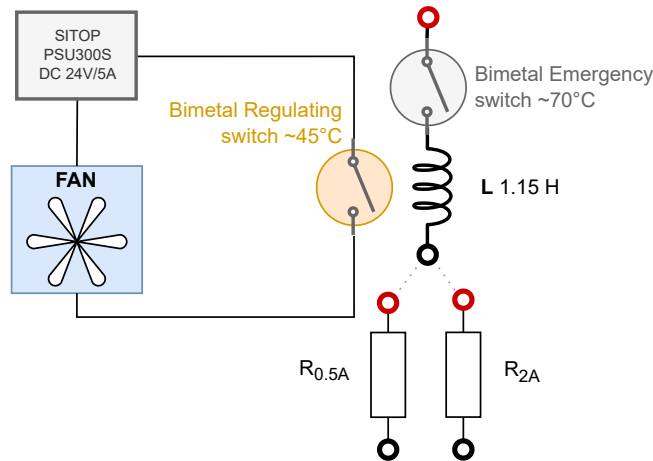


Figure 2.14: Circuit diagram of one inductive load channel including a bimetal temperature control

dynamic thermal behavior of the entire system would be strongly affected by the resulting physical size, shape, and material of the enclosure, as well as the placement and distance of the coils to each other.

The more expensive version of the thermostat (shown in the right picture 2.13) can significantly compensate for the shortcomings of the cheaper version. It is more robust and has an adjustable both temperature switching level and hysteresis. However, it still lacks an indicator of the coil temperature. It is also rather inconvenient that eight of these thermostats would be needed because each coil will heat up differently during the testing. In fact, a current of 2 A, or 0.5 A, or no current may pass through the coil because the user may not use all eight load channels. In conclusion, this bimetal-switch design would have some advantages, but in the end, is not suitable as it does not satisfy the important requirements.

2. **PLC.** A Programmable Logic Controller (PLC) was integrated as an alternative method in the temperature control design. This system would be capable of measuring temperature, independently controlling fan speed, and sending temperature data to the control laptop. I considered using the Siemens ET200SP family of PLCs [11], which is robust, reliable, and has many advanced features. The development program called TIA Portal also allows for the implementation of more complex algorithms. Any input and output modules for reading or writing analog or digital values can be connected to the main CPU device that controls the entire PLC system. For this project, an eight-channel module with analog inputs (AI) would be required to read temperature sensor data, and an eight-channel module with digital outputs (DQ) would be needed for switching fans. No additional circuit, such as a transistor or relay, would be needed to switch the fans because the digital outputs of the PLC can supply a sufficiently large current. Only 150 mA is needed for the selected 24 V axial fans, which the PLC digital module with 0.5 A or 2 A can handle. This design version would be more robust and reliable because PLCs are specifically built for industrial environments.

Additionally, it would be more transparent as it could send the measured coil temperatures to a laptop, where the information could be displayed in a graph for the user's convenience.

Although this design meets all the requirements, it is quite complex and, most importantly, expensive for such a simple temperature control. The cost of purchasing all the necessary parts would exceed 33,000 Kč (1450 USD). The problem of a high price could be solved by choosing a different, simpler PLC family, such as the Siemens LOGO! However, this simpler product offers only a very small number of I/O modules for connecting to the main CPU. Thus, it is not possible to build a LOGO! system that contains only the required modules with 8x AI and 8x DQ channels. Instead, it would have to be composed of multiple, more general expansion modules. One LOGO! module is equipped with a maximum of two analog inputs, so a lot of modules would be needed as a result, again increasing the cost and complexity of the system. Coupled with the high cost and complexity of the resulting system, communication with the LabVIEW program on the laptop would not be very straightforward either, making it less suitable for this project after all.

3. **Arduino.** The next design idea included the open-source single-board microcontroller called Arduino, which comes equipped with digital and analog input/output pins that can be connected to various circuits and sensors. It would require additional temperature sensors for each inductor, as well as a relay or transistor switch, programming it to activate the fan when the temperature crosses a certain threshold. The maximum current that the DQ pin can supply is usually 20 mA, which is not enough for one of my 24 V 150 mA fans.

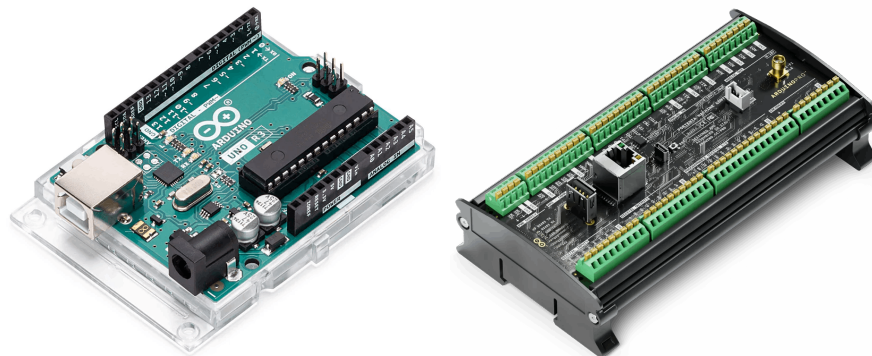


Figure 2.15: The cheaper Arduino UNO (left), source: [2] and the more robust Arduino PRO Portenta (right), source: [3]

There are many different Arduino models available, ranging from the classic Arduino UNO [2], which is popular among hobbyists, to more industrial-oriented ones like the Arduino PRO Portenta Machine Control [3]. The great advantage of using this small board is that it already contains all the necessary analog and digital channels, which means that it will take up very little space, as no additional modules are required. Additionally, the Arduino is much more affordable than a PLC system, with the Arduino UNO costing around 700 Kč (30 USD) and the Arduino PRO Portenta 7500 Kč (330 USD).

The portable version of Arduino is even equipped with an RJ45 connector, which would allow it to communicate directly with a laptop via Ethernet, making it more suitable for this particular project. The Arduino UNO, on the other hand, doesn't have this feature and would have to be connected either via USB, or an additional Ethernet shield [12] would have to be purchased to extend the motherboard with this connector.

Overall, this design would be well-suited for the temperature regulation project, as it is not as complicated as a PLC system while having all the necessary features. It would measure the coil temperatures with an analog or digital sensor and autonomously control the fans based on the readings, while also sending the data to the LabVIEW control program. The only disadvantage is that the cheaper Arduino UNO is not very reliable, and since this multichannel load station is supposed to last for many years, it would be necessary to choose a more robust Arduino Portena. The Portena has a better design of connectors that press the wire firmly, similar to industrial PLCs. Secondly, the communication between Arduino and the control LabVIEW program would not be as straightforward as it is with the further discussed and finally selected version. I would have to deal with formatting data into TCP/IP packets and designing the low-level communication.

4. **NI USB module.** The final design that I have chosen is very similar to the Arduino version, but with the added advantage of simple and straightforward communication. The implementation uses a low-cost Data Acquisition (DAQ) device called NI USB-6001 [A.2.6] from National Instruments, which also develops the LabVIEW programming environment. This makes the module very smoothly compatible with the control program.

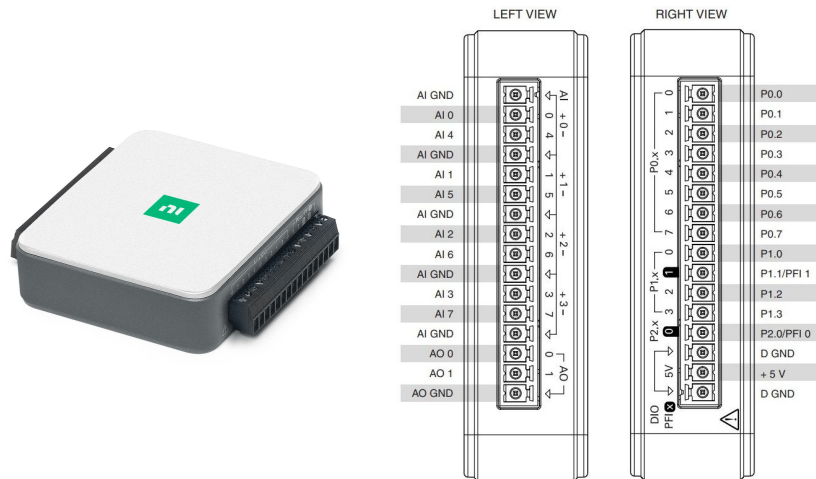


Figure 2.16: National Instruments DAQ module called USB-6001 (left) with its pinout showing arrangement of analog and digital pins (right) [A.2.6]

The NI module has eight analog inputs, two analog outputs and thirteen digital pins configurable as inputs or outputs, as shown in Figure 2.16. It also has a +5 V constant voltage pin and a single USB port for connecting

to the control laptop. The AI pins have a typical accuracy of ± 6 mV and 0.7 mV RMS system noise. The maximum sample rate is 20 kS/s with a maximum input range of ± 10 V, and 14-bit ADC (Analog to Digital Converter). The digital pins set as output have output voltages between 2.1V and 3.6V, which depends on the current draw and represents a logic 1.

National Instruments provides well-supported LabVIEW drivers, which makes it easy to read data with AI, write with DQ and implement the whole temperature regulation. However, external temperature sensors and an auxiliary circuit for the fan switching would be needed. Their design and implementation are further discussed.

2.2.5 Measuring Temperature

There are a lot of ways to measure the temperature of the coil winding. I have investigated a few of them, evaluated their advantages and disadvantages and chose the suitable one for this project.

1. When considering temperature measurement methods, the first one that came to mind was utilising a **thermocouple**. These sensors measure temperature by using the junction of two different metals to produce a voltage that is temperature-dependent. They can measure a wide range of temperatures and respond quickly, but they are measuring only the relative temperature between the cold and hot junction. An additional temperature sensor measuring the absolute temperature of the thermocouple cold junction would be required. The NI DAQ module is not equipped with a feature to measure cold junction temperature and may not be suitable for very low-voltage measurements due to the ± 6 mV typical accuracy. Depending on the specific thermocouple type, the sensor may produce a voltage of around 1 mV when there is a temperature difference of 30°C between the hot and cold junctions. Thus, an operational amplifier would be strongly required. These limitations, combined with the inability of the NI module to measure the cold junction temperature, make this method not ideal.
2. The next idea was to use a **Resistance Temperature Detector (RTD)**, such as PT100 or PT1000. They measure the absolute temperature by correlating the resistance of a metal, typically platinum, with temperature. PT100s have a resistance of $100\ \Omega$ at 0°C and are more accurate and stable than thermocouples. Although they have a narrower temperature range and slower response times, these characteristics are not problematic for my design. PT1000s work similarly but have a higher resistance of $1000\ \Omega$, allowing longer cable runs without signal degradation. A simple measuring circuit that uses a voltage divider composed of a fixed resistor and the PT1000 sensor is shown in figure 2.17.

The problem here still remains that the NI module is not designed for measuring very small and fine voltages, and the voltage drop across the sensor is very small, as shown below. A common PT1000 sensor [13] has a maximum allowed current of 0.4 mA, as listed in the datasheet.

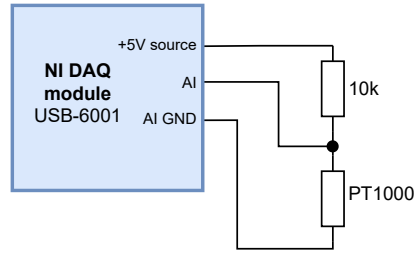


Figure 2.17: Diagram of simple temperature measurement using PT1000 resistance temperature detector and NI DAQ module.

The resistance of the sensor ranges from 1000 ohm to 1380 ohm over the interval from 0°C to 100°C. From the Ohm's law, I can calculate the change in voltage drop ΔR across the sensor when the maximum current is passed.

$$U = \Delta RI = (1380 - 1000)0.4 \cdot 10^{-3} = 0.15 \text{ V}$$

With the 100°C change in temperature, the measured voltage across the PT1000 sensor would change only by 0.15 V. That would result in a large measurement inaccuracies in addition to the ± 6 mV accuracy of the analog inputs, which can be improved by using a circuit with an operational amplifier.

The next problem is that the +5V voltage source has only a $\pm 3\%$ accuracy, resulting in voltage fluctuations between 4.85 and 5.15 V. This would very unpleasantly spoil the whole measurement because it relies on measuring the voltage drop across the PT1000 sensor, which requires a constant supply voltage to the voltage divider.

This uncertainty can be solved by using a low-dropout voltage regulator (LDO) to stabilise the supplied voltage.

Although the mentioned issues could be resolved by using additional amplifiers and stabilisators, I decided to use a different temperature sensor for my final implementation, which has slightly better characteristics from the beginning.

3. Finally, I have chosen an analog temperature sensor called **TMP37** that provides a linear voltage output directly proportional to the measured temperature. It requires a low input voltage of 2.7 V to 5.5 V and has a 20 mV/°C linear scale factor. With the 1°C rise in temperature, the sensor's output voltage changes by 20 mV. The sensor does not require any external calibration to provide a typical accuracy of $\pm 1^\circ\text{C}$ at room temperature. This temperature range is a lot narrower than that of thermocouples, but a wider range is not required for this project.

Although this sensor does not require a perfectly stable voltage source, as with the RTDs, I will connect a simple voltage regulator from Texas Instrument to the sensor input. Fluctuations in the input voltage of up to $\pm 3\%$, as indicated in the NI module datasheet [A.2.6], could make the temperature measurement slightly inaccurate. Thus, adding a single low-noise, low-dropout voltage regulator TPS791 [A.0.2] to power all eight temperature sensors will improve the performance of all the sensors.

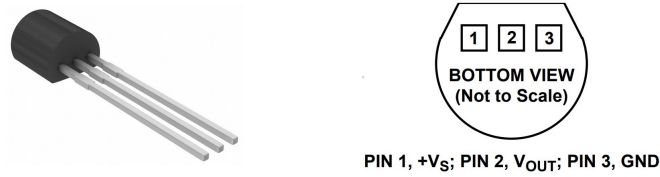


Figure 2.18: The analog and linear temperature sensor TMP37 [A.0.1] with 20 mV/°C scale factor in TO-92 package [4]

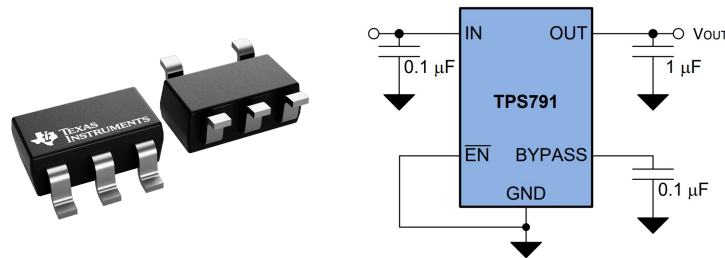


Figure 2.19: Low-noise, low-dropout voltage regulator [A.0.2] powering the analog temperature sensors

The LDO output will have a fixed voltage of 4.7V and will be able to supply a maximum current of 100mA, which is more than sufficient since one TMP37 temperature sensor only requires 50 μ A. The LDO stabiliser also excels with its ultra-low noise of only 15 μ V RMS. Connecting additional ceramic bypass capacitors between the component pins and the common ground further improves the transient response, noise rejection, and ripple rejection, as shown in the component datasheet [14, p. 16]. Connecting the enable pin to the ground would turn on the regulator as soon as it receives the supply voltage.

I was concerned that noise could interfere with the sensor output signal, causing additional uncertainties. The station will be in a laboratory with other industrial devices that switch high voltages or emit electromagnetic fields, such as electric motors, generators, and power supplies, which commonly emit 50 Hz noise and its higher harmonics. Low-voltage temperature sensors like TMP37 are highly sensitive to external noise. Even small levels of interference can significantly affect the accuracy of temperature readings. At first, I considered using a hardware Low-Pass filter. The temperature signal is, or should, change very slowly since coil heating is a very slow process, based on the experiments conducted. With a cut-off frequency chosen at approximately 10Hz, the 50Hz noise would be effectively eliminated. Even though this implementation would most likely work, it involves additional circuitry and eight separate filters – one for each individual sensor.

Instead, I chose a software approach for greater flexibility. When measuring the temperature, I will take 16 samples at a sampling frequency of 400 Hz and compute the arithmetic mean. In other words, I would measure the sensor signal while averaging over two entire periods of 50 Hz noise, eliminating its influence. This approach is schematically shown in Figure 2.20. This would not only reduce the error caused by 50 Hz

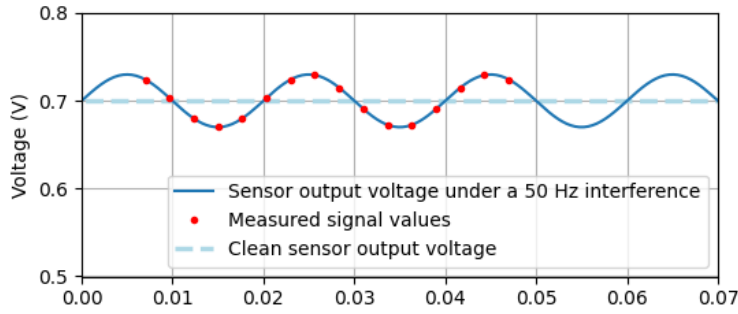


Figure 2.20: Visualisation of sampling the sensor output voltage at 400 Hz, with averaging to remove a 50 Hz noise.

noise but noise at higher frequencies in general. Additional protection would be using shielded cables, minimising the noise interference from the very beginning.

In order to calculate the total uncertainty of temperature measurement, all partial sources of uncertainty must be mentioned. I am not taking into account the noise interference because I do not know its magnitude before the implementation and filter efficiency. Any conversion of voltage to temperature uncertainty can be calculated as

$$T_{error} = \frac{V_{error}}{20 \cdot 10^{-3}}$$

because of the sensor's linear characteristic between voltage and temperature. There are several sources of partial uncertainty to consider:

- The LDO regulator has output noise of $\pm 15 \mu\text{V}$ when using the bypass capacitor $0.1 \mu\text{F}$ as shown in scheme 2.19. This results in only $(\pm 75 \cdot 10^{-5})^\circ\text{C}$ error.
- The NI module analog inputs have a full-scale uncertainty of $\pm 6 \text{ mV}$, resulting in $\pm 0.3^\circ\text{C}$ based on the module specifications.
- The temperature sensor itself has a typical uncertainty of $\pm 1^\circ\text{C}$ stated in the datasheet.

The total temperature error can be calculated as the following, assuming independent error values and Gaussian distribution.

$$T_{error} = \sqrt{T_{e1}^2 + T_{e2}^2 + T_{e3}^2} = \sqrt{(75 \cdot 10^{-5})^2 + 0.3^2 + 1^2} \approx 1.04^\circ\text{C} \quad (2.8)$$

This level of uncertainty is acceptable given the nature of the project. Moreover, based on the conducted experiments, I can expect a larger measurement uncertainty due to uneven heating of the coil and the quality of the thermal connection between the sensor and the coil winding.

To summarise, the final design for measuring the temperature of the coils inside the inductive load unit will be as follows: A linear TMP37 sensor will be connected thermally to each of the eight coil windings (see photos 2.25). It will be powered by a low-noise stabilised LDO, improving the sensor characteristics. The LDO will be powered from

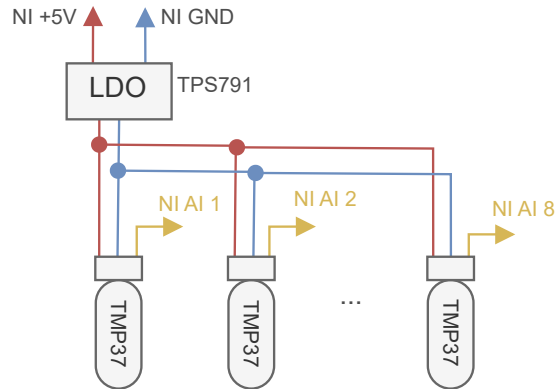


Figure 2.21: Schematic of temperature measurement design of inductive load using linear sensor TMP37 and voltage regulator TPS791

the +5V pin on the NI DAQ module, which generally has poorer noise and dropout characteristics. The output voltage from each temperature sensor will be periodically read by the analog inputs on the NI module and further processed in the LabVIEW control program.

2.2.6 Fan control

Since the digital output pins of the NI USB-6001 module can only supply 4 mA current, an additional circuit is needed to assist in switching fans rated for 150 mA and 24 V. This circuit will be later integrated into a bigger printed circuit board (PCB). I have decided to use a MOSFET transistor instead of a relay, because of several factors.

- **No Physical Wear.** Relays contain mechanical contacts that physically open and close, leading to wear over time. MOSFETs are solid-state devices with no moving parts, so they do not suffer from physical wear, which generally allows them to have a longer lifespan when used in switching applications.
- **Speed.** MOSFETs can switch on and off much faster than mechanical relays. In the future implementation, I would try to control the fan by a PWM signal, and the relay would not be able to switch fast enough.
- **Low Power Drive Requirements.** A MOSFET can be directly driven by the low-current output from the digital pin because they require very little current to switch. The MOSFET gate pin is isolated so there is virtually no current flow. In contrast, relays often require more current than the digital pin can provide, necessitating an additional driver circuit.

I have chosen a P-Channel SMD MOSFET [A.0.3] with the maximum Drain-to-Source voltage of 30 V and 0.6 Ω resistance in ON state. It can supply a maximum of 760 mA continuous drain current, which is suitable for switching the 24 V 150 mA fan. I have added some more components to the circuit, as shown in figure 2.22, and I will further explain their meaning.

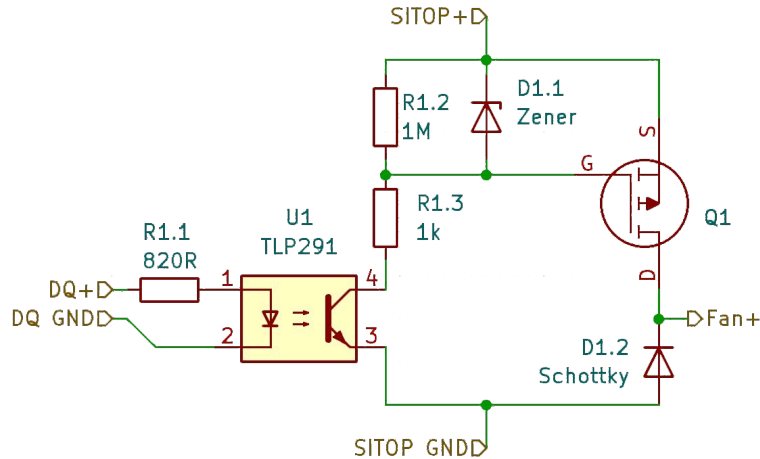


Figure 2.22: Schematic of a sub-circuit controlling the 24 V fan by NI module digital output pin

As previously mentioned, the SITOP is a device from Siemens that provides 24 V DC voltage to the axial fans. The output pins of this device are labelled as SITOP+ and SITOP GND at the top and bottom of the circuit respectively. In the circuit, the zener diode D1.1 serves as an over-voltage protection of the transistor gate. I have chosen a diode [A.0.5] with 15V zener voltage, which stabilises the voltage across it to approximately 15V in the reverse breakdown region. The MOSFET has a maximum Gate-to-Source voltage of ± 20 V, and the diode will clamp the SITOP 24 V voltage to a safe 15V magnitude when the gate is grounded.

The R1.2 is a pull-up resistor needed to discharge the transistor gate, which acts as a capacitor. When the gate is connected to ground, there is 15 V between the gate and source, turning on the MOSFET. When the gate disconnects from the ground and connects to the 24 V source, the transistor should turn off, opening the circuit. But without this pull-up resistor, there will still be low voltage on the gate because of its capacity, causing it not to turn off.

The resistors R1.1 and R1.3 only limit the current flow to a lower magnitude. The U1 represent an optocoupler called TLP291. Its function is to galvanically separate the NI DAQ module from the switching circuit so that the possible noise from it is not affecting the low voltage temperature measurements. All ground pins of the NI module are connected to its chassis. Thus, the ground noise from the DQ GND pin would directly introduce the noise to the AI GND pin.

When the NI module digital output pin switches to the logical 1, the current flowing through the optocoupler LED closes the circuit between pins 4 and 3. Thanks to that, the MOSFET gate is connected to ground, introducing approximately a 15V difference between the gate and source. This enables a current flow from the source to drain pin, turning on the axial fan connected between the Fan+ and SITOP GND pins. There is virtually no voltage drop on the MOSFET because it only has 0.6Ω in the ON state, passing the 24 V to

the Fan+ pin.

The Schottky diode D1.2 serves as a flyback protective diode. At the heart of the axial fan is a DC electric motor consisting of wound copper coils. Because of it, it can be modelled as a simple inductive load from the circuit's perspective. When switching off the fan, the collapsing magnetic field generates a high-voltage spike in the opposite polarity, potentially damaging sensitive components in the circuit, including the MOSFET. The fast-acting Schottky [A.0.6] diode will clamp the voltage spike to a safe level by providing a current path in this opposite direction.

The electronic components like resistors and diodes have been marked with a 1. prefix. This is because the final circuit design will include at least eight of the discussed switching circuits. This is due to the fact that there needs to be a separate control for each fan of the eight inductive load channels. Additionally, there will be supporting fans mounted on the front and back sides of the load housing, enhancing the airflow through the whole unit.

2.2.7 PCB for temperature regulation

I have designed a larger PCB connecting the explained sub-circuits 2.22 (controlling 8 fans directly at the coils, and 2 fans for additional support) and LDO regulator for powering the temperature sensors. Additionally, I have added two LEDs to the circuit for a better indication if the circuit works properly. The first LED is placed between the SITOP supply pins, indicating if the PCB is connected properly. The second LED is connected between the stabilised LDO output pins, showing if the temperature sensors are powered and, therefore, the temperature at their output can be read. A schematic of the circuit that stabilises the power supply to all of the TMP37 sensors is shown in Figure 2.23 and is inspired by the scheme from LDO regulator's datasheet 2.19.

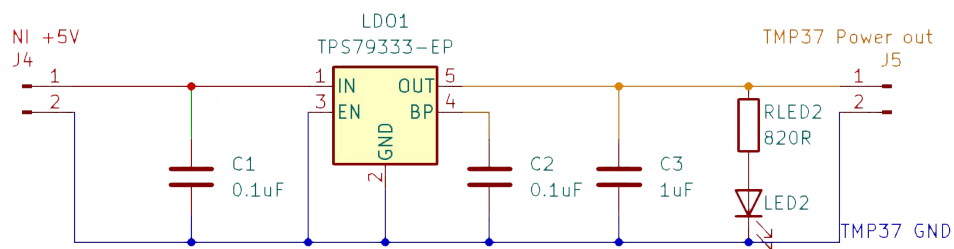
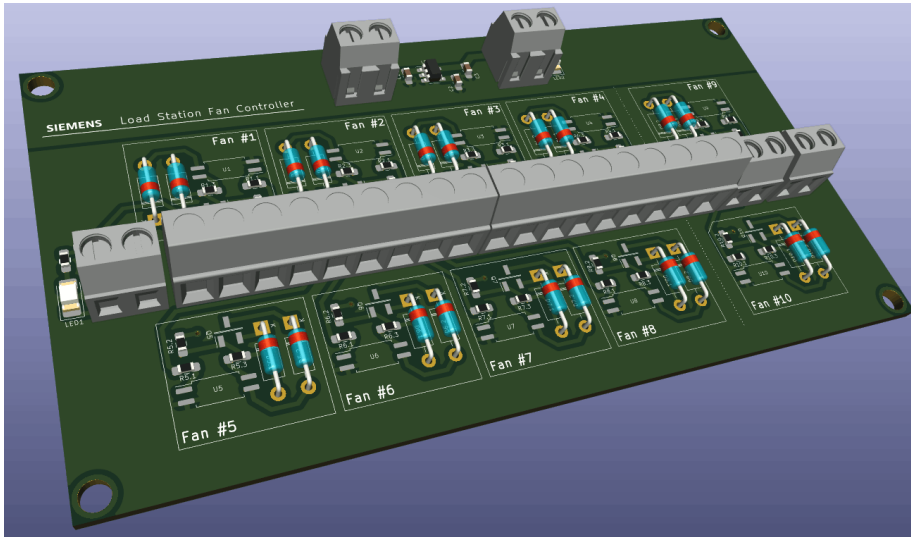
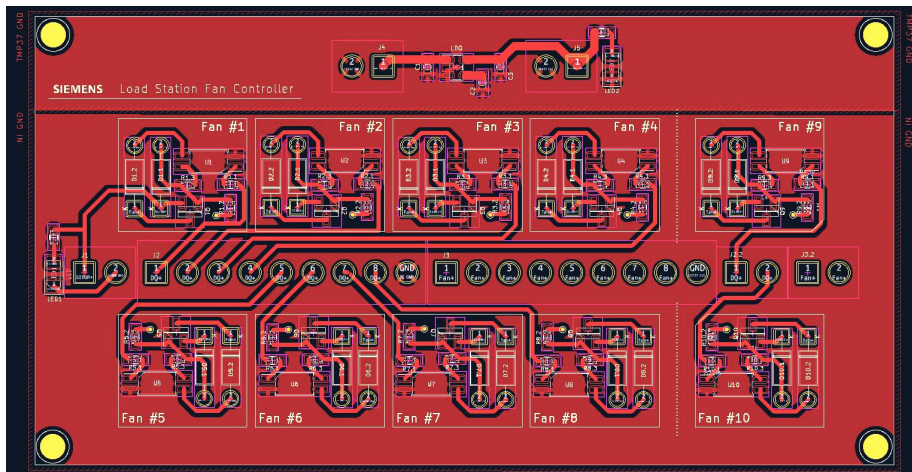


Figure 2.23: Schematic of a sub-circuit stabilising the 5 V output of NI module to supply the TMP37 sensors

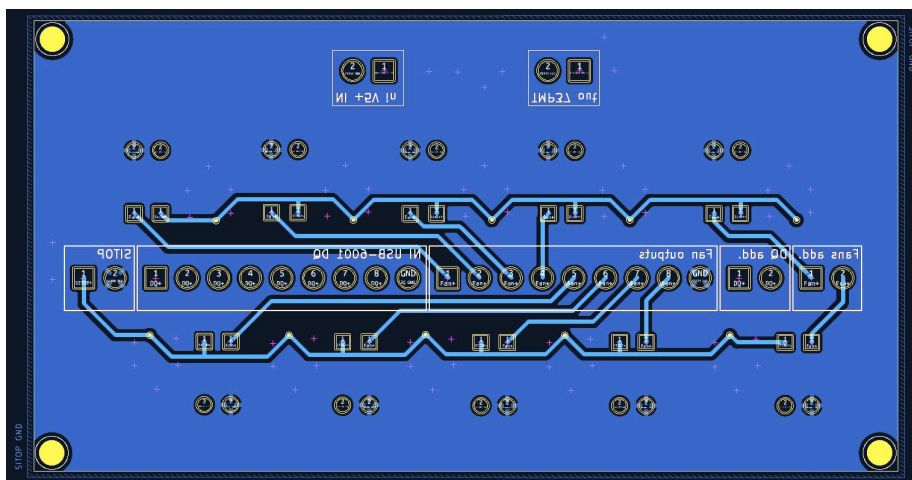
The final PCB, shown in Figure 2.24, consists of two copper layers - top and bottom. It has four holes in the corners for easy attachment to the station shelf where the NI USB-6001 module will also be placed. The front layer contains 10 optically separated MOSFET circuits arranged around several PCB connectors known as terminal blocks [A.0.9].



(a) : 3D visualisation of the PCB with the top side shown



(b) : Top layer PCB



(c) : Bottom layer PCB

Figure 2.24: Two-layer PCB design with 10 MOSFET sub-circuits and LDO circuit stabilising TMP37 power supply

It allows a wire to be firmly connected to the PCB by tightening a screw on the top side of the connector. Similar connectors are often used in industrial applications for their mechanical reliability.

Starting from the left side, the terminal blocks connect the PCB to

- SITOP 24 V power supply
- Digital outputs of the NI module
- Outputs to controlled fans

The upper section of the printed circuit board is dedicated to the LDO stabilisation sub-circuit. It also includes two terminal blocks – one on the left for connecting the NI module 5 V supply and one on the right for connecting the temperature sensors. The entire PCB, including the schematic, was designed using KiCad 7.0, which is open-source software for creating professional PCBs. Additionally, I have uploaded the PCB Gerber files, which contain the fabrication data, to the JLCPCB company website [15]. It offers custom, high-quality PCB manufacturing. After delivery, all that was left to do was to fit the board with all the components, summarised in table A.1. The finished PCB compactly unified the sub-circuits, with printed labels, indicator LEDs and attachment holes in the four corners for firm attachment, resulting in a more professional look.

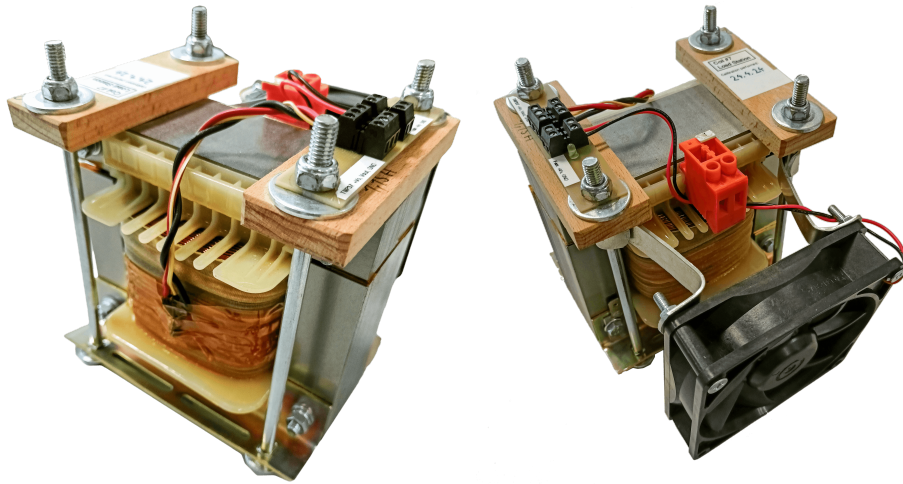


Figure 2.25: Photo of an inductor with attached temperature sensor and fan connected to a simple PCB

It is necessary to note that I have manufactured additional simple PCBs, that serve only as a connector. When the coil needs to be removed from the load unit for calibration, I can just disconnect the wires from this PCB with the connectors and the fan and temperature sensor will still remain attached and wired to the coil. The Figure 2.25 shows a photo of the inductor with the attached temperature sensor using thermally conductive adhesive. The fan is firmly connected to the coil structure and oriented to blow the air directly at the coil winding. Those components are then connected to the mentioned PCB.

2.2.8 Cooling tunnel with integrated power resistors.

Additionally, I have designed and assembled a narrow tunnel for the resistors corresponding to the 0.5 A and 2 A current through the inductive load. These resistors would dissipate a lot of energy in the form of heat. Although their resistance does not change with temperature as significantly as the coil winding, they need to be cooled down to maintain a safe temperature. An additional pair of fans is used for this purpose, ensuring a constant airflow through the resistors. The chosen resistors are power-adjustable resistors that can be easily calibrated using their sliding pin with a screw that touches the conductive part of the resistor.

For the 2 A current mode, a single 10 Ω , 50 W resistor would be used [A.0.11]. Based on the following equation can be calculated the approximate resistance of the adjustable resistor labeled R_R .

$$I = \frac{U}{R_{coil} + R_R} \rightarrow R_R = \frac{U}{I} - R_{coil} = \frac{28.8}{2} - 8 = 6.4 \Omega$$

Since the coil resistance is typically 8 Ω , it would be adjusted to 6.4 Ω in order to provide 2 A current when the 28.8 V is connected.

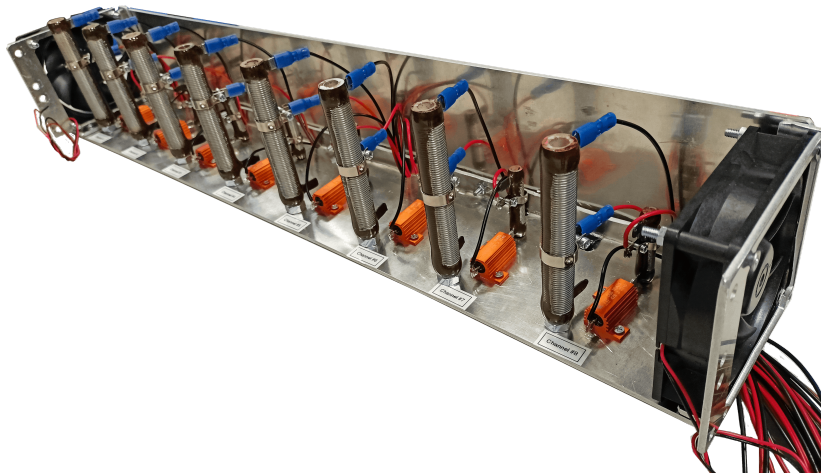


Figure 2.26: Aluminium tunnel for effective cooling of the power resistor corresponding to inductive load channels

Further, for the 0.5 A mode, a fixed resistor (47 Ω , 25 W [A.0.13]) in series with a second adjustable resistor (10 Ω , 12 W [A.0.12]) would be used. This would allow a more precise adjustment to a value around 49.6 Ω to have the 57.6 Ω resistance in total. Thus, the cooling tunnel would consist of 24 lined-up power resistors, optically divided into the 8 channels for clarity when performing calibrations as shown in Figure 2.26.

2.3 Design of power supply and communication

When designing the power supply for the entire station, including all the instruments, I had to take into account the safety standards, the input

characteristics of the instruments, and the implementation of protections in the laboratory where the station would normally be located. There are multiple power distribution cabinets (shown in Figure 2.27) each with three single-phase outlets rated for 16 A (labelled as 1,2,3), one three-phase outlet rated for 16 A (labelled as 4), and one three-phase outlet rated for 32 A (labelled as 5). There are also circuit breakers rated for the same current values.

2.3.1 Power and safety

The biggest burden on the power infrastructure will undoubtedly be programmable power supplies, so I paid attention to them first. The manufacturer states in the manual [16] that the device can draw up to 11 A at full power when connected to a 230 V single-phase mains supply. After further investigation, it was clear that the entire station had to be fed from a three-phase network, with the two power supplies separately on one phase and the rest of the less demanding devices on the remaining third phase. Thus, the station will be connected to the three-phase outlet labelled with the number 4.

The power supply for the entire station will first pass through the main switch, which will turn off the power supply for the entire station. The main switch will be located at the back of the station and will not act as an emergency stop button. Right after the switch, a red signaling light will be connected. The light will indicate that the dangerous high voltage is connected to the station, and it should be disconnected before working on the station circuitry. The operator can be sure that there is no risk of electric shock if the main switch and the light indicator are off. This is a safety standard that is mandatory for industrial stations in general. The next step to improve safety is to install a three-phase main circuit breaker (MCB) with a current rating of 16 A. It would protect the station and its electronics from damage caused by overcurrent. It interrupts the circuit and prevents the risk of fire.

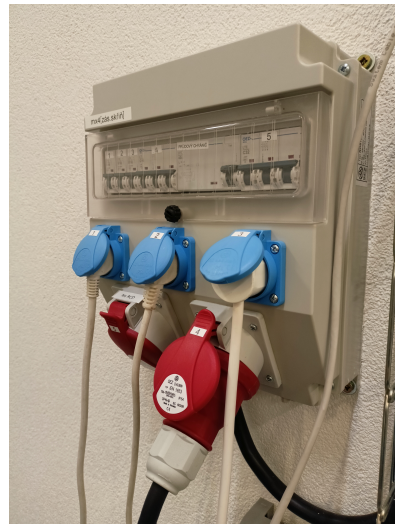


Figure 2.27: Power distribution cabinet in a Siemens laboratory

Additionally, a residual current device (RCD) will be installed to monitor the current flow. It quickly breaks the powering circuit to prevent serious harm from an ongoing electric shock when it detects a current flow imbalance between the live and neutral wires. This will ensure that any electrical faults that may occur will be handled efficiently, thereby minimising the risk of electrical hazards. An RCD (Residual Current Device) and an MCB (Miniature Circuit Breaker) can be found integrated into the same device, called a RCBO (Residual Current Circuit Breaker with Overcurrent protection). The

RCBO is capable of detecting both overload current and supply imbalance. This device is usually cheaper and takes up less physical space, making it a convenient option for electrical protection.

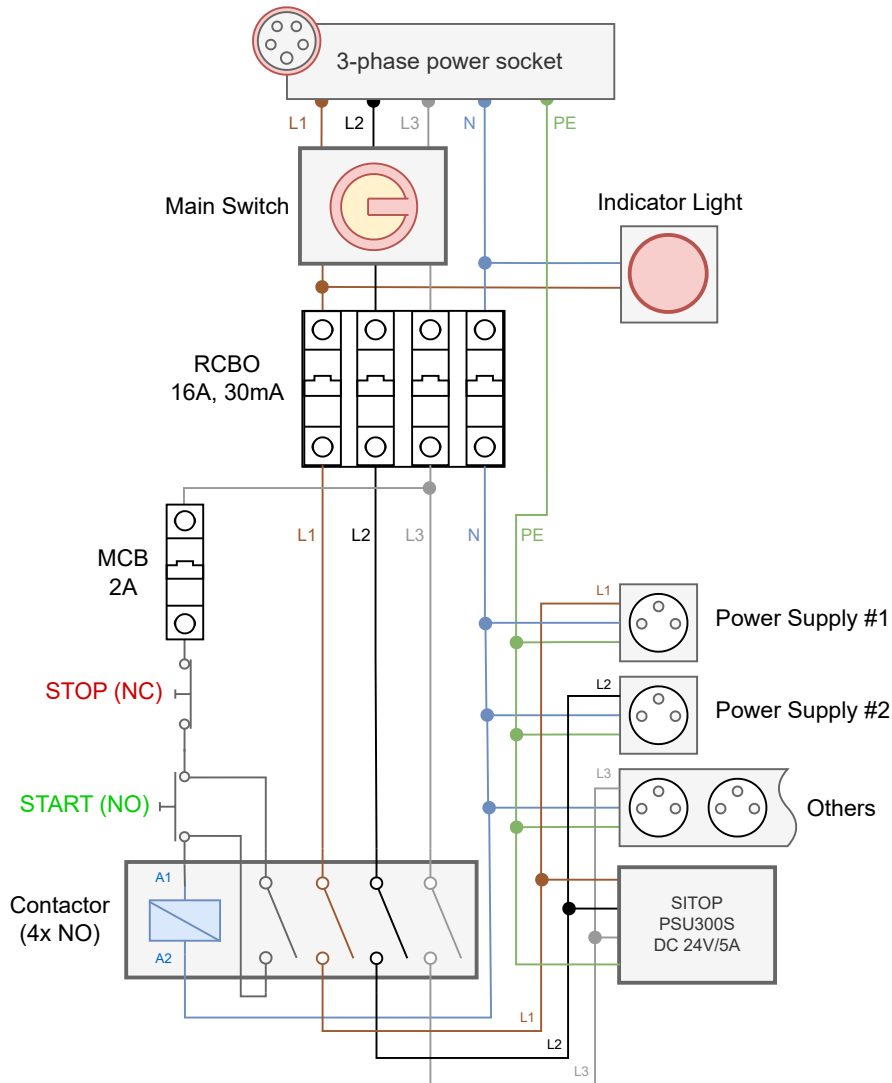


Figure 2.28: Three-phase station powering diagram with safety devices

According to safety standards, an unplugged station must always stay turned off when plugged back into the three-phase network. To meet this requirement, a Contactor should be used in conjunction with an additional start button. The Contactor consists of four normally opened (NO) contacts and uses an electromagnet to close the powering circuit mechanically. If the current flow through the electromagnet is interrupted, all four contacts are opened, turning off the entire station and its devices. This not only satisfies the safety standard but also allows for the integration of an emergency switch mushroom button that can be placed in the control circuit loop. The start and stop buttons will be located at the front of the station for easy access. At the end of each phase wire, a 16A socket will be installed. The first and second phases will only power the two power supplies individually due to their maximum 11A consumption. The third phase will have no trouble powering

the remaining devices, including laptops, electronic loads, LAN servers, and other devices.

The entire discussed powering design is shown in the diagram 2.28. Note that the protective conductor (PE) is not interrupted by any switching element, which is also one of the safety standards. The station should always be grounded. There is also an additional three-phase device called SITOP, which acts as a stabilised 24 V DC power supply for the axial fans integrated into the inductive load unit, discussed in 2.2.6.

2.3.2 Communication

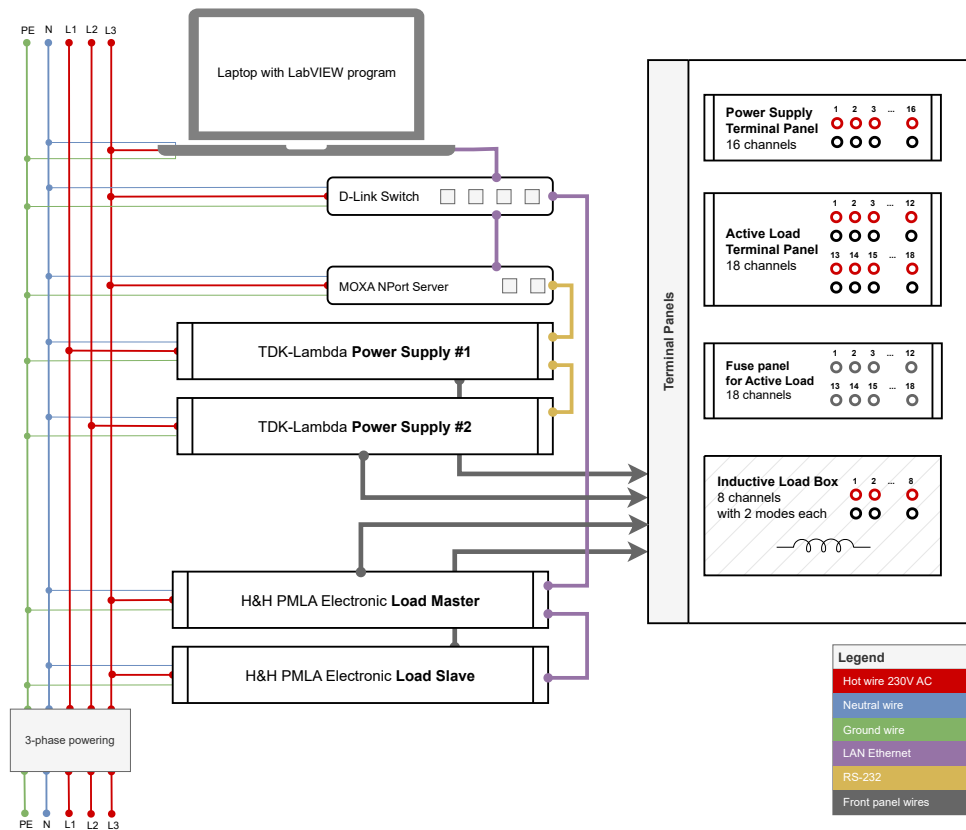


Figure 2.29: Powering and communication scheme of the station

The laptop, which controls all instruments and devices through the LabVIEW program, only has a single RJ-45 LAN connector. An additional LAN server is required to communicate with multiple devices. The server understands the Data link layer of the standard communication OSI model, which allows it to forward a message to the intended subnet without needlessly spreading it to other subnets. The LAN communication is marked in violet on the scheme shown in Figure 2.29.

The TDK Lambda programmable power supplies do not support LAN connection, but only serial RS232 or RS485. Therefore, another server device is required to convert LAN to RS232. The MOXA NPort 5250A can perform

this conversion, eliminating the need for additional cables to connect to the laptop. Only one LAN cable is needed, which is ideal since the laptop will be placed on a pull-out drawer, and additional cables would be inconvenient.

Note that each TDK Lambda power supplies are connected separately to the first two phases and the rest of the devices to the third phase to spread the power consumption as evenly as possible. The instruments' outputs are connected to easily accessible aluminium panels located at the front of the station, also shown in the visualisation 1.1 and scheme 2.30.

2.4 Design of rack housing

The entire multichannel load station should be mobile to be easily moved across the laboratory whenever required and used with a selected climatic chamber, which is also an important part of testing the PLC equipment's durability. The station devices should be placed in an industrial rack cabinet that has a standard width of 19" because the mentioned devices come in this size. The rack is also vertically divided into sections called units, often referred to as "U", which has a standardised height of 1.75" (44.45 mm), and all industrial devices are designed to fit within the standard rack units.

The diagram 2.30 shows the actual placement of devices in the rack cabinet with the components numbered as follows:

1. TDK Lambda Genesys power supplies connected serially via RS485 cable and then to MOXA server via single RS238 cable.
2. H&H PMLA electronic loads Master and Slave connected to each other with LAN cable, and then to D-Link server via single LAN cable.
3. Fixed rack shelf holding the devices securely in place.
4. Pull-out drawer with the laptop controlling the entire station via the LabVIEW program.
5. Box with emergency stop button and start button on the front side and communication devices (servers) hidden on the inside.
6. Terminal panel with the outputs from both TDK Lambda power supplies brought to Banana connectors for easy access. All eight power supply channels are internally connected to a single output from the device.
7. Terminal panel with external cartridge fuses in fuse holders for each electronic load channel. The fuses can be easily replaced when broken or worn.
8. Terminal panel with all electronic load channels brought to Banana connectors. These channels are actually separated and not connected to a single device output.
9. Heavy box with eight separate inductive load channels and a cooling system. The box is situated at the very bottom due to its weight of approximately 40kg.

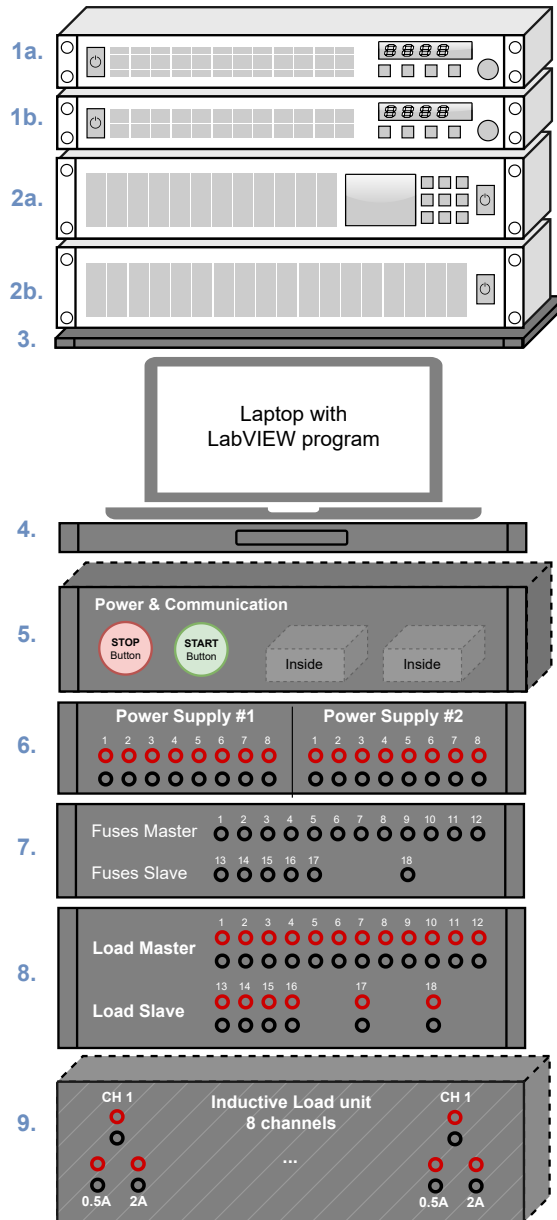


Figure 2.30: Physical placement of devices in the rack cabinet

The mechanical design of the terminal panels was really time-consuming but also gave the final result a more professional look. I designed seven panels made of 4mm thick aluminium that form the front side of the station. The instrument outputs and channels are brought to the front and placed within the panels via Banana connectors in a clear and practical way.

The panels unify specific channels or features and are named accordingly. The number in brackets refers to Figure 2.30.

- Powering & Communication (4)
- Power Supplies output (5)
- Active Load cartridge Fuses (7)
- Active Load channels (6)

- Inductive load unit front side (channels and fan holes) (9)
- Inductive load unit back side (fan holes only) (not shown in the figure)
- Main Switch and Power socket (not shown in the figure)

I had to precisely measure the dimensions and all the different sizes of the rack cabinet, including the hole spacings. Additionally, I had to choose all of the hardware components beforehand to consider their dimensions. I was drawing the design in a program called AutoCAD (version) from Autodesk, which is widely used in mechanical engineering and architecture, and I have access to it within the CTU license. After the design was completed, I ordered the panels from the "CNC Metal" company, which proved to be very professional in terms of communication and the quality of the final order. They finished the surface of the panels with black matte anodised into which they lasered the text.



Figure 2.31: Finished panel of inductive load (9) mounted on the front side of station

Additionally, I created a 3D model of the entire station, including the aluminium panels and main devices, using Autodesk 3ds Max. This program allowed for high-quality visualizations, which were used to present the concept to Siemens and explain it to the CNC metalworking company. The visualization is shown in Figure 1.1. I was already familiar with the Autodesk programs, which made the process quicker.

The final assembly and wiring of all devices, components, and panels took me approximately 14 days straight. I put great emphasis on neat cable management for better clarity and safety of the entire station. More photos of the final assembled station are available in the Appendix B.

Chapter 3

Software design

This chapter will discuss the development of the control and data acquisition software for the multichannel load station. LabVIEW, a system-design platform and development environment, is central to this discussion. LabVIEW played a central role in this project, including hardware integration, user interface design, data handling, and generating final reports.

3.1 Introduction

LabVIEW, a system-design platform and development environment created by National Instruments, is specifically tailored for engineering and automation applications, control systems, and hardware system development. Unique to LabVIEW is its graphical programming approach, which utilises a visual language known as G [17] to create programs in a block diagram format. This makes it particularly accessible for engineers and scientists who may not have a traditional background in text-based programming. It is important to note that G-code is also used in different contexts to refer to an engineering language for controlling CNC machine tools. In our context, G language refers to LabVIEW graphical code.



Figure 3.1: NI LabVIEW logo

Despite its strengths, LabVIEW has limitations. The graphical nature of the language can make it cumbersome for developing large-scale or complex algorithms, where a graphical user interface or hardware control is not needed. Debugging graphical code might also be less straightforward compared to traditional text-based languages.

For engineers already familiar with other programming languages, understanding these basic principles of LabVIEW can open new opportunities for integrating hardware and software in their projects. Almost all programmable devices, such as power supplies, have their official LabVIEW drivers. Thus, their integration into the control system is very straightforward, making

LabVIEW highly suitable for automated testing and hardware control in general.

In LabVIEW, programs are referred to as Virtual Instruments (VIs) [18]. Each VI has three main components: the Front Panel, the Block Diagram, and the Connector Pane. The Front Panel (FP) serves as the user interface, where controls and indicators such as knobs, buttons, graphs, and other input/output elements are placed. These elements allow users to interact with the VI. The Block Diagram contains the graphical source code that defines the functionality of the VI. All the logic and operations are represented by graphical symbols and wires that imply data flow, making the programming intuitive and closely aligned with the engineer's natural way of thinking about their processes.

VIs can be modular and hierarchical, meaning one VI can call another as a subVI. This allows complex applications to be broken down into manageable pieces and promotes code reuse. LabVIEW's dataflow programming model also executes code in an order determined by the flow of data between nodes, which can lead to highly efficient parallel execution.

3.2 Program concept

The following image 3.2 displays the fundamental concept of the LabVIEW program, where optional features are marked in grey. The program requires an operator to define a queue (sequence) of tests at the beginning. Each test is composed of parameters that need to be set on all devices such as power supplies and electronic loads. The operator also needs to specify the duration of the test and its ID, which represents the order of the test in the queue. Moreover, the operator has the option to save the defined tests as an XML file onto their flash disk, which can later be easily loaded back into the LabVIEW program the next time. This feature can save a considerable amount of time since each test has numerous parameters.

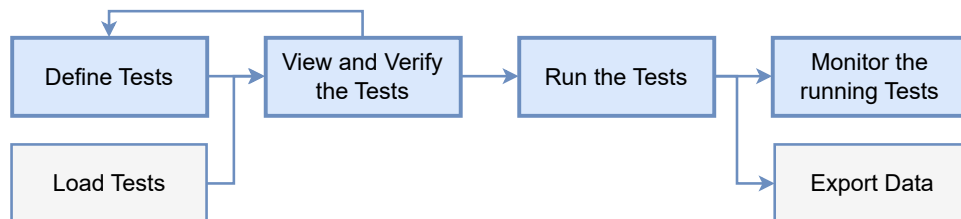


Figure 3.2: Basic concept of the LabVIEW program

After the operator defined or loaded all tests he wants to perform, he can double-check the entire queue before deciding to run it. If he made a mistake, he can go back and rewrite a specific test. Thereafter the queue of tests can be launched just by pressing a Run button, and the tests are automatically run in the defined order. The devices also measure and store real-time data and the operator can see them in a graph. The acquired data can also be exported as a comprehensive Excel report during or after the tests.

3.3 Main structure

The central component of the entire LabVIEW program is the `Main_Control` VI. It acts as a parent VI of all the other subVIs, and I will refer to it as control program. Its front panel is designed with only a few controls, as described depicted in figure 3.3 and described below.

- **Menu buttons:** The operator can switch between the four main subVIs and access their interface with specific controls and functionalities. This concept divides the huge user interface into four smaller sub-pages. The subVIs are called `TestSetup`, `ViewTestQueue`, `DataView` and `Help`.
- **Spacious sub-panel:** This is where the user interface of the selected subVI is loaded. It takes up most of the space.
- **Status message with LED indicator:** The station provides information about its current state and whether everything is operating correctly. If there is a problem, the error message is displayed, and the LED turns red.
- **Stop button:** Pressing this button terminates the entire station program. The program is designed to safely terminate all subVIs sequentially, from the lowest to the highest level, to ensure that all devices are powered off and communication is safely closed. It is important to avoid abrupt termination of higher level VIs, as that would leave the power supplies and other devices on, with communication lost. For this functionality is used a `global stop` variable so that all subVIs can register it.

It is essential to note that this interface is always shown to the user. The four subVIs never overlap this parent VI, because their controls are loaded into the sub-panel. That also ensures that the Stop button is always accessible.

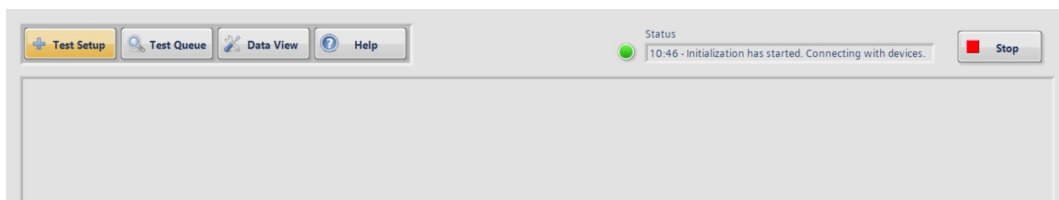


Figure 3.3: Screenshot of the `Main_Control` user interface. Only a part of the sub-panel is shown.

The following figure 3.4 displays the actual LabVIEW code of the `Main_Control` VI written in the graphical G language. Throughout the implementation of this project, I dedicated significant effort towards ensuring that the entire code of every VI could fit on a single screen. To accomplish this, I divided the code into four main subVIs, which further use other supporting virtual instruments. This approach enhances code readability by breaking it down into logical pieces, making it easier to debug and reuse.

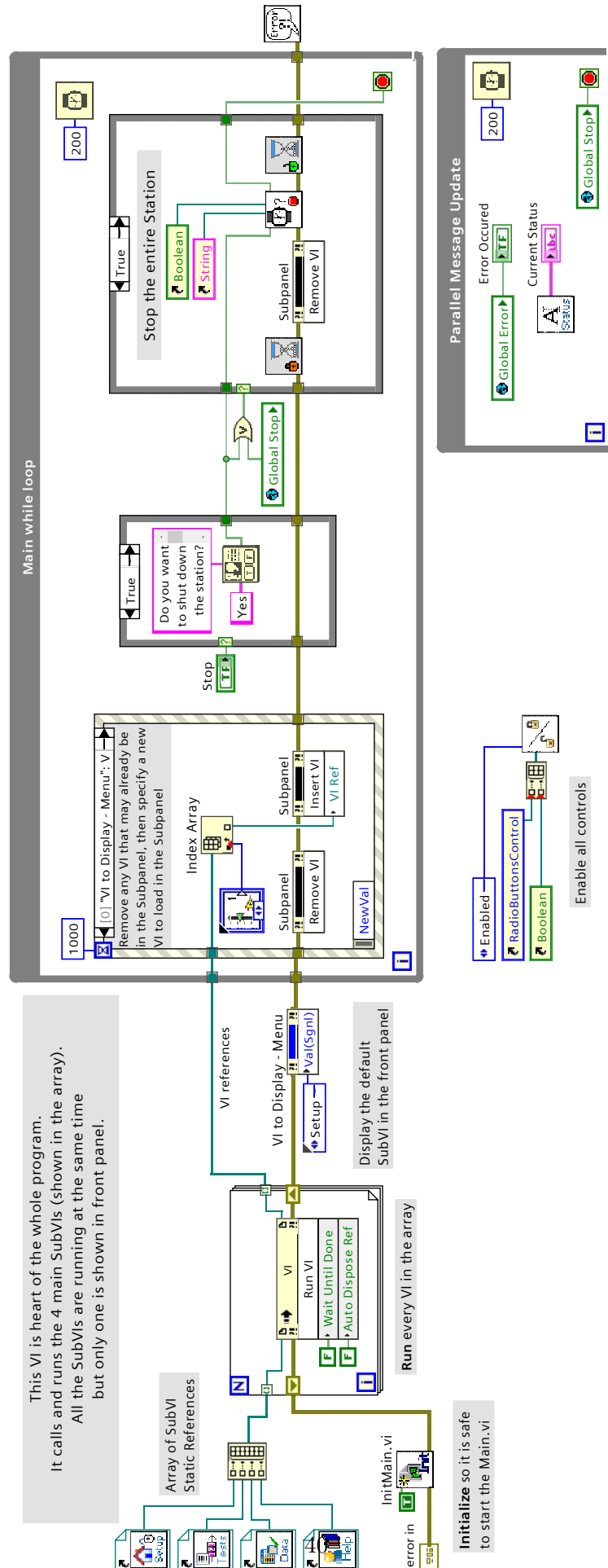


Figure 3.4: The Main_Control block diagram, the actual LabVIEW code.

I will now briefly discuss the structure and data flow of `Main_Control` – the hearth of the entire program.

- The code starts on the left side, where static references to the four main subVIs are stored in an array. The array signal then leads to a for loop that would run the subVIs, but it must wait for the completion of all signals leading into it from the left. In this case, it must wait for the execution of the `Init` VI and its error signal. In LabVIEW, connecting all important elements with an error signal to propagate any information is standard. In addition, this also clearly defines the order in which the code is executed.
- The `Init` VI below prepares all data structures to their default values, sets a default message to the Status window and ensures that all SubVIs are ready to launch.
- Moving on to the right, the for loop launches the four subVIs from the array and then the `Test Setup` subVI is displayed in the front panel by default. This is where the operator can define and run all tests.
- The signals then lead to the Main while loop. The program remains in this loop until the end of the entire program. It handles the event of switching between the control pages using the menu buttons by the `Event structure` on the left. When a user does so, the current subVI is removed from the GUI panel, and the selected VI is inserted. There is also a 1 second timeout for non-blocking behaviour (without the timeout, a deadlock situation would occur).
- The rest of this while loop is responsible for monitoring whether the Stop button has been pressed. If so, a message will be displayed to the operator to confirm if he wants to terminate the station, thus stopping all running tests. The station also terminates if a global variable called `Global Stop` is switched on. This happens if a serious error occurs and the station needs to shut down to ensure safety. Afterwards, the VI will be removed from the GUI panel and the `Gradual Termination` will be executed, which ensures that the communication with devices and low-level subVIs has finished. This strategy is well-known from multithread programming concepts. The while loop also has a waiting function that ensures that each iteration executes after 200 ms or more, to reduce the burden on the laptop's CPU.
- The smaller while loop at the bottom updates the current status info. It is better that this process runs in parallel to the main loop to avoid blocking the main loop due to the dataflow nature, which can occur when the `Gradual Termination` is being executed. In such a scenario, the main loop would be unable to update the status indicator.

To summarise, this parent VI primarily segments the code into four smaller pieces that run simultaneously in parallel. Secondly, it has the ability to stop the entire station safely. The `Main_Control` sub-panel with a subVI loaded in is shown in the later figure 3.5. I will now explain the concept of the four mentioned subVIs (`TestSetup`, `ViewTestQueue`, `DataView` and `Help`).

3.4 Test Setup VI

When the control program is launched, the **Test Setup** subVI is loaded and shown in the main front panel by default. Here, the operator can define a queue of tests and run them. The controls are separated into three sections:

1. **Power Supply.** Each of the supplies can be turned on and off, and the voltage and current limits can be set for each supply independently.
2. **Active Electronic Load.** A mode and value can be set for each of the eighteen channels. The modes are constant current (I), constant voltage (U), constant resistance (R), constant power (P) and channel disabled (Off). For example, if mode I with the value 0.5 is selected, the load channel would set the specified constant current draw of 0.5 A regardless of the voltage across it. The channels 17-18 are also optically separated from the others because they are stronger and have higher current and power limits, as noted in the 2.1.3 section.
3. **Test Parameters.** Each test in the queue needs its time duration and ID to define how long the test will run and the order in the queue.

All of the control elements are also managed programmatically to prevent the operator from setting an incorrect value that exceeds the limits. The program checks the validity of the entered value whenever a control is clicked. Since the allowable values differ for each mode, the input limits must be managed programmatically while considering the selected mode. This functionality is implemented in **Limit_TDK** and **Limit_PMLA** VIs ensuring a fault-tolerant system. The device limits are specified in the Admin Settings, and only an authorised person can modify them.

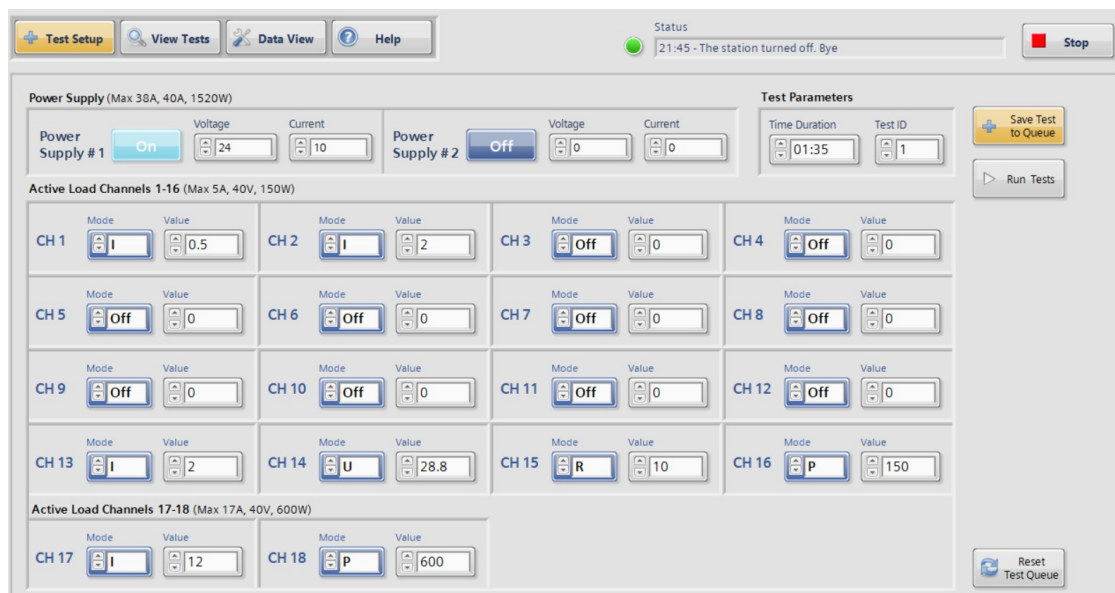


Figure 3.5: The Test_Setup interface with control elements, loaded in the Main_Control sub-panel

The code is implemented as a Finite State-Space Machine. It is a programming concept that simplifies the implementation in a way that the program can be only in one state at a time, and transitions to the next state based on evaluated conditions. I have learned about Finite Space Machines in my LSP lectures [19] which claim that

The finite automaton-based model reduces reality to the minimum facts needed to solve the problem.

The implemented machine consists of five states: `Init`, `Idle`, `Setup Test`, `Monitor Test` and `Error`, marked in blue and transition conditions in orange in Figure 3.6.

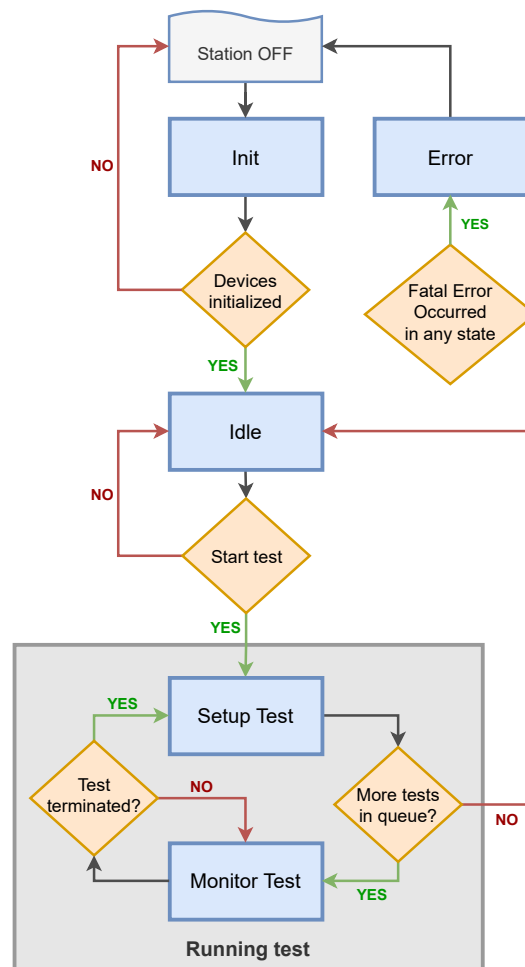


Figure 3.6: Diagram of Test Setup VI code implemented as a State-Space Machine

- The machine moves from the Off state to the `Init` state as soon as it starts. The Station Off is not actually an implemented state but just a logical representation that the control program is turned off.
- During the `Init` state, the program attempts to establish a connection with all physical devices, and switches them to a remote control state. It also ensures that all outputs are safely turned off. If the initialization process is unsuccessful and some devices do not respond correctly, an

error message is displayed to the user, describing the specific problem, and the station terminates. It is pointless to proceed if everything is not working correctly. Even if one device is not functioning, the tests cannot be launched properly.

- After successful initialization the machine moves to the **Idle** state. Here the program settles until the tests are ready to be launched. The state takes care of checking test data validity, adding tests to the test queue and launching the test execution itself. All of these features are implemented by a single **Event** structure. It waits until an event occurs, then executes the appropriate case to handle that event.
- When the user presses the **Run Tests** button, the machine transitions to **Setup Test** state. Here, It pops a test from the queue and prepares specific internal information about the current time, test duration and ID for future computations. Then it sets the values specified in the popped test on all devices. So it turns on the power supply outputs, sets the specified voltage and current limit, and respectively with the other devices. If there is no test left in the queue, the machine transitions back to the **Idle** state, allowing the operator to define and run new tests.
- If there still was an test in the queue and was set up, the machine transitions to **Monitor Test** state. The program checks if the test time has passed or not based on it's defined duration. Additionally, the devices perform real-time measurements, acquire the data and show them on a graph for the operator, however this is not implemented here. The entire data acquisition and graphing process is implemented in **Data View VI**, that can take measurements only if this state is active. When the test duration has passed, the program will reset the outputs of all devices to safe zero values. It will also signalises that the test has ended and the measured data should be saved.
- Additionally, if a serious error occurs in any state, the machine transitions to the **Error** state. Here, the program disables the control buttons so that no further actions can be made and the gradual termination process is launched, safely closing communication with devices and terminating all the subVIs.

Note that there are a lot of complex processes and communication with devices happening in each state, but since the transition conditions are relatively simple, the code is more robust. Debugging State-Space Machine is also very straightforward because we can simply view the current state and monitor the values of conditions. This structure can also be easily expanded in the future if additional states are required.

3.5 View Test Queue VI

This is the second subVI from the main menu. User can scroll through the list of all defined tests and double-check how many tests are defined and all the specified parameters. Additionally, he can save the entire queue of

tests to an XML file, which can be stored on a flash disk by simply clicking the **Save Tests** button. The next time the tests need to be performed, user can load the tests back into the LabVIEW program by pressing **Load Tests**. The program will ask the user if he wants to overwrite the current tests by the loaded ones, ensuring a more user-friendly interface. This feature can be especially useful when repeating the same tests on the PLC equipment.

The saving and loading of the XML file itself is implemented in **XML_Tests VI**, using the LabVIEW File I/O palette that offers specific functions for writing and reading from files. When choosing the format to save the test parameters, I decided between XML, Binary and INI files, which LabVIEW can easily work with. A binary file takes up less memory, but its format is not human-readable. The INI format is then more suitable for storing configuration files than directly storing LabVIEW variables. The XML file is written in a format that humans can easily read, and the entire queue of tests can be saved using a single I/O block called **Flatten to XML**. This means that if a user needs to check the saved tests, they can open the XML file on their computer and read the defined values of each test. While modifying the XML file is possible, it is not recommended as the same file format needs to be maintained.

■ 3.5.1 Structure of the Test Queue

Although the queue structure itself is also implemented in LabVIEW, an array offers way more functionalities for working with data, thus appearing more efficient for this use case. Queues in LabVIEW are ideal for managing data flow between processes within a program, especially in producer/consumer scenarios. The queue of tests is, therefore, implemented as a global array. Still, it appears to the user as a queue because tests can be added to the end and are automatically taken from the front in the running process. However, this structure also allows rewriting a specific test from the middle of the queue and other supporting features.

Further, a single test from the queue is implemented as an instance of a class called **TestClass.lvclass**. In LabVIEW, employing classes improves data organisation by encapsulating related functionalities and variables into a single structure. This method ensures that each test, with its specific parameters, becomes a distinct instance of the class, thereby streamlining data management. Moreover, this structured approach fosters code re-usability and simplifies modifications, ultimately leading to more efficient and manageable software development. The data of the test are also secured and only the methods from the same class can read and modify them. For this purpose I have implemented **GetTestParams.vi** and **SetTestParams.vi** as methods of the **TestClass**. Each class instance represents an individual test from the queue with parameters structured into clusters.

A cluster in LabVIEW is a data structure that groups multiple elements, which can be of different data types, into a single entity. This differs from an array, which consists of elements all of the same data type. Clusters are

beneficial for organising related but varied data together.

Each class instance representing a test contains three clusters:

- Power Supply cluster divided into two sub-clusters for each supply, because there are two individual power supply devices used. Further, each sub-cluster contains data of the selected mode (On/Off), voltage and current values. A boolean data type represents the mode, while voltage and current values are both represented as a double.
- Cluster representing electronic load divided into 18 sub-clusters for each channel. Each channel sub-cluster stores data about the selected mode (U, I, R, P, Off, meaning constant voltage, current, resistance, power and channel turned off) represented as an enumerated integer data type. Further, the channel contains the numeric value for the selected mode, which is represented as a double data type.
- Cluster named `Other Parameters` directly containing Time Duration and Test ID values represented as a double and a long integer, respectively.

Structuring the test data to clusters enables the grouping of logically related data, making it easier to pass multiple data points through wire connections in the graphical code. Additionally, if the load station is expanded by new devices, their parameters can be easily added into the cluster structure.

3.6 Data View VI

The `Data View` is quite a complex Virtual Instrument that encompasses all load station measurements and data acquisition processes. It features a large graph of the measured data, as shown in 3.7. The operator can switch between the devices and its channels, and the corresponding graph will be displayed. On the right-hand side, general information about the currently performing test is provided, including ID of the running test, how many tests are in the queue, total time duration of the queue and duration of this single test. Below that information indicator are two buttons for easy Excel report generation and also admin settings that only an authorised user with knowledge of the password can access.

The code block diagram contains four parallel while loops explained below.

1. **Data Acquisition:** This loop is contained in a specialized subVI called `Measure Devices`. is responsible for periodically acquiring data from all devices and storing them in memory. The data is then graphed and displayed on the `Data View` front panel. The data acquisition period is a variable and can be modified by the user when launching the tests. Depending on the duration of the test, the period can be adjusted to prevent the operator from being overwhelmed with a lot of data. For instance, a test that lasts 20 hours would require a longer period compared to a test that lasts only an hour.

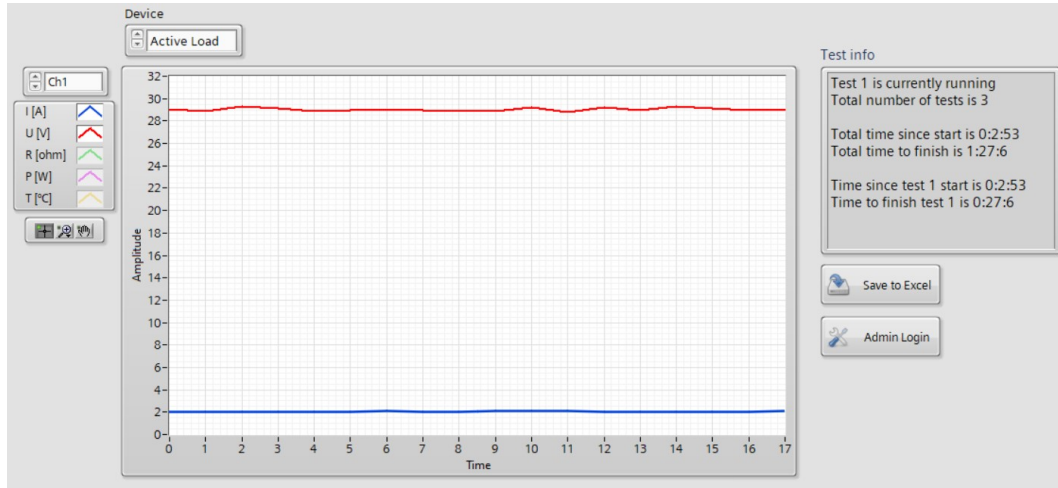


Figure 3.7: Front panel of Data View section, with switchable graph, Test information indicator and buttons for Excel report generation and admin login features

Each device's data has the format of a cluster comprising several arrays. Each array represents an instrument channel and contains the measured physical quantities over time. Therefore, the data is a 2D array where one axis contains the measured quantities, and the other axis contains the data over time. This structure is convenient for later data plotting to graph and switching between signals so that only those selected by the user are displayed. However, there is no timestamp corresponding to real-time, only the measured quantities in chronological order. Additionally, the program acquires time stamps at the same time as the instrument data.

This approach is advantageous because the timestamp is stored only once and not in each instrument's data. It is also better than calculating the time based on the start time and the selected period as it is more accurate. It should not happen, but it may take longer than the desired period to measure and collect the data, either because of sluggish communication with the instruments or a slow laptop.

The data sending for graphing and displaying to the user is implemented using the LabVIEW queue system from **Synchronization** palette. Each device has its queue created during the initialisation. Whenever new data is measured, it is stored in the queue, and the graphing loop can read it immediately. When new data is being measured after the time period passes, the old data is still accessible for graphing. Only when the new data are fully prepared and added to the queue, the old data are deleted, bringing the new data to the front of the queue. If I implemented this system using simple variables, old data would disappear from the graph when new data is being measured and updated, and the graph would remain empty for a moment until the new data is successfully saved. This queue system ensures that the producer and consumer loops can operate independently at varying execution rates without data loss or synchronization issues.

- 2. Graph updating:** This loop runs faster than the measuring devices, as it directly interfaces with the user. It contains an event handler structure for handling actions related to the graph. By clicking the “Device” control, the user can switch between the instruments acquiring data. Simultaneously, not only the graph changes, but also the control box with the written channels, because each instrument has a different number of channels. Thus, each instrument has its corresponding graph to which the measured data is uploaded, the graphs are stacked at each other in the same place on the front panel, but only the one selected graph is shown at a time.

I have implemented this differently, more easily, earlier in the development. At first I have only had a single graph and I loaded the data of the selected instrument into it. This resulted in a really weird behaviour. The graph was displaying signals of the power supply, and when the user switched to “Active load”, the graph showed half of the signal from the power supply, and half of the graph from the load. I was only able to switch the source of the signal, not the data of previously selected instrument. When erasing the graph before loading data of newly selected instrument, only a section of the measured data will be shown, and not from the beginning of the test.

I needed the system to behave in such way, that the measured data are continuously loading to all of the graphs, even to the hidden ones. When switching between devices or channels, the whole measured signal can be displayed, belonging to the correct device.

- 3. Admin Login:** This loop contains an Event Handler that monitors whether the operator wants to log in as Admin. When the “Admin Login” button is pressed on the user screen, a pop-up window appears where the username and password must be entered. This feature is implemented using `Prompt User.vi` from LabVIEW’s “Dialog & User Interface” palette. If the operator enters the correct login credentials, the VI marks the Admin Settings as opened and temporarily disables the Login button to prevent multiple concurrent openings of the settings. The reference login data is stored in a global variable that only a developer can modify.

After a successful login, the event handler marks the Admin settings as opened and executes a specialised subVI called `Admin Settings`. This opens the settings window for the operator as a pop-up window, which only authorized operators can access, and is shown in figure 3.8. Most space is devoted to device lower and upper limits that cannot be exceeded when defining a test. On the right side is shown the current state of the state-space machine, which is explained in 3.4 The Admin can modify and save the settings or click “cancel” to discard any modifications made.

All of these settings are saved to the `AdminSettings.ini` configuration file. This is necessary because LabVIEW variables are wiped out when the program terminates, resulting in data loss. To save the settings permanently, I have implemented `INI_Settings.vi`, which can save and load the settings from the configuration file. This feature uses the “Config File VIs” sub-palette from the “File I/O” palette. The configuration

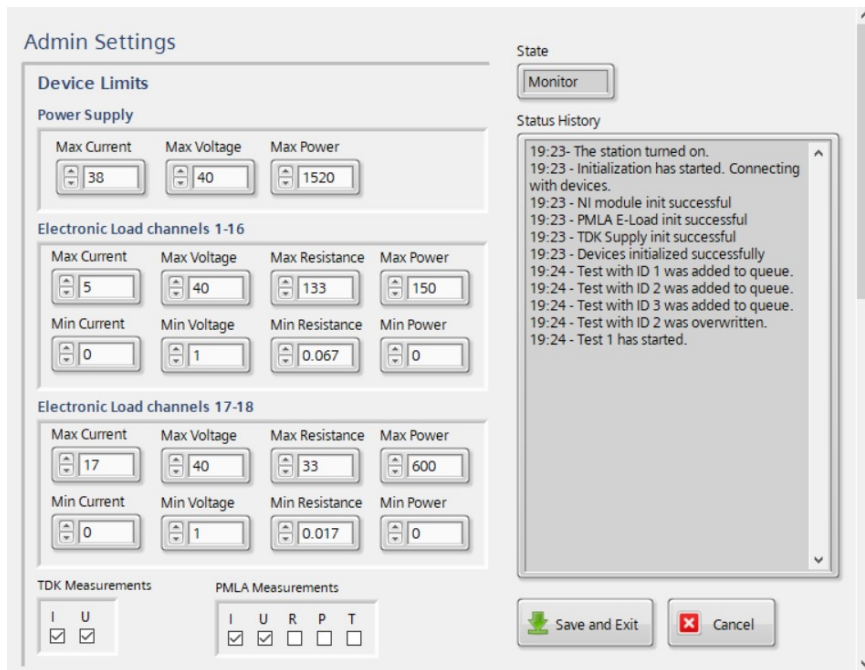


Figure 3.8: Admin settings window with device limits, state indicator and Status history, that only authorized operator can access

file is read and loaded during program initialisation and saved when the admin makes any modification.

3.7 Help VI

The help section contains only three buttons that allow you to access the following resources:

- The Quick Start Guide: This PDF guide provides essential information on how to operate the station safely.
- The User Manuals: A summary file with links to manufacturer user manuals for all instruments and devices used in the station implementation. They can be helpful when debugging or investigating an issue with the station or specific devices.
- Contact Details: A PDF file including contact information to get in touch with the Load Station developer, the person responsible for the laboratory equipment, and the climate chamber specialist. The station user can contact them with any issues or provide feedback to further improve the station interface and features.

All of this important information is written as PDF files so that they can also be placed on the Desktop of the laptop or uploaded to the Siemens employee manual website. Therefore, the station user can access and study these documents without running the control program or having to physically approach the station in the laboratory. Although these documents are not yet

fully completed, the block diagram code of the Help window is implemented. It contains a while loop with a single event structure to open the corresponding documents when the user presses a button. The case structure provides the gradual termination strategy explained in 3.3 section. When registering that the station is shutting down, it indicates itself as terminated and the while loop ends.

For opening the PDF files from LabVIEW, I have implemented an additional virtual instrument called `OpenPDF.vi`. It executes a system command `cmd /c "DocumentName"` via the Command Prompt terminal. The "DocumentName" is VI's input argument, which means that this VI can be used to open all the mentioned supporting documentation, leading to cleaner and reusable code. The PDF files are placed in the LabVIEW root directory in the `Documentation` folder.

3.8 Integration of Devices

One of the reasons why the mentioned devices were chosen is that they have high-quality LabVIEW drivers, which allows for straightforward implementation into our system. These devices are controlled from either `TestSetup.vi` ensuring a successful initialisation or closing of communication, or `DataView.vi` performing real-time measurements and graphing out the data.

Each device has implemented the following VIs and some additional:

- `Init` VI establishes connection with the device, resets its outputs for safety and reports if there are any issues.
- `Close` VI turns off the device outputs and terminates the communication safely, for when the station control program is gradually terminating.
- `DataAcq` VI acquires data from device inputs and adds them to the created structure of measured data for that specific device. As mentioned in 3.6 section, the data structure is a cluster of multiple 2D arrays, where each array corresponds to one device channel.
- `CheckConnection` performs a self-test to verify communication between the device and the laptop. If the device does not respond or a communication error occurs, it will attempt to reinitialize the device. If the connection fails again, the program will report the error, and the light indicator on the main screen will turn red to inform the operator (see Figure 3.3).
- Additional VIs for reading values from device inputs or writing values to its output channels.

These VIs use the device specific drivers for communication and actions, but I have also implemented VIs unifying all the devices. For example `Init_Devices.vi`, shown in Figure 3.9, tries to establish a connection with all the devices at the beginning of the program. It further uses the specific

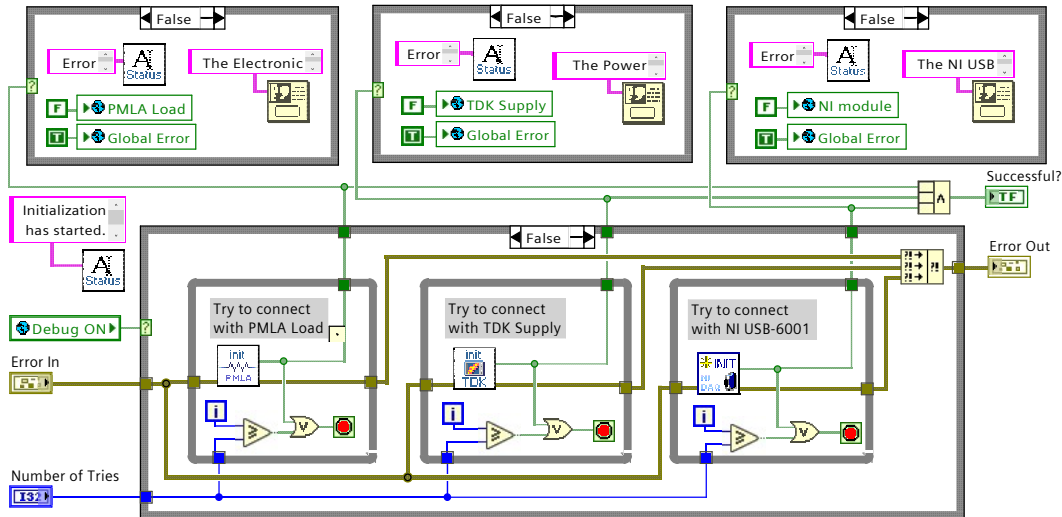


Figure 3.9: Block diagram of `Init_Devices.vi` establishing connection with all devices

initialisation VIs for each device and evaluates if the devices communicate correctly. However, if an error occurs during the initialisation, a warning message is displayed to the user, and the station is safely shut down. If the devices are not powered on or have disconnected communication cables, this problem must be resolved before running the automated tests.

A debugging feature is also implemented throughout the entire program. No user can access it, but the developer can turn on the global variable called `Debug ON` and the program would only simulate communication with the instruments, including initialisation and measurements. This is useful for developing and the program at home, outside the laboratory with the actual station and instruments. Without this option, the program could not be run because initialisation would fail, and the program would terminate with an error message.

3.8.1 Electronic load control

I downloaded the LabVIEW drivers for the H&H PMLA load from the National Instruments website [20]. However, I faced issues while communicating with the device, even though its IP address was visible and responding properly to the `ping` command in the command prompt terminal. I contacted the manufacturer, who advised me that I had to configure the device from an additional application called NI MAX (Measurement & Automation Explorer). There, I created a VISA resource name and linked the device's IP address to it, hopefully making the communication work.

Afterwards, I read the example projects that are often included in the drivers to get inspiration for my own implementation. For quite a long time, I struggled with the fact that the instrument has 18 channels, and it was necessary to implement for-loops acquiring the measured data. However, the acquired data was in the wrong format, than which is needed for subsequent

plotting in the graph. Consequently, I have resolved this issue in the `Data_Acq VI` that adds the new measured data to the existing structure in a way that suits the graphing needs.

3.8.2 Power supply control

I have approached the TDK-Lambda Genesys supply similarly, downloading the 2017 drivers from the NI website [21]. Here, I also faced a communication issue, that was caused by a more serious problem. After further investigation, I identified the problem with the MOXA N-Port 5250A server [A.2.5], which converts LAN to RS232. Using the `ping` command, I found faulty communication between the laptop and the server. Reinstalling server drivers and modifying the device configuration did not resolve the issue. Ultimately, I resolved the problem by using a simpler USB-RS232 cable adapter.

However, a new issue arose: the lack of USB ports on the laptop. The station operator needs to connect a USB flash drive to upload or save measured data. The HP laptop [A.2.1] has only two USB ports, one occupied by the NI DAQ module and the other by the TDK-Lambda supply. To address this, I obtained an i-tec USB Hub [A.2.7] with three USB ports. This setup left one USB port on the laptop and two out of three slots on the hub available, allowing for up to two additional flash drives or peripherals.

This change in communication architecture is schematically shown in Figure 3.11, with the original setup on the left.

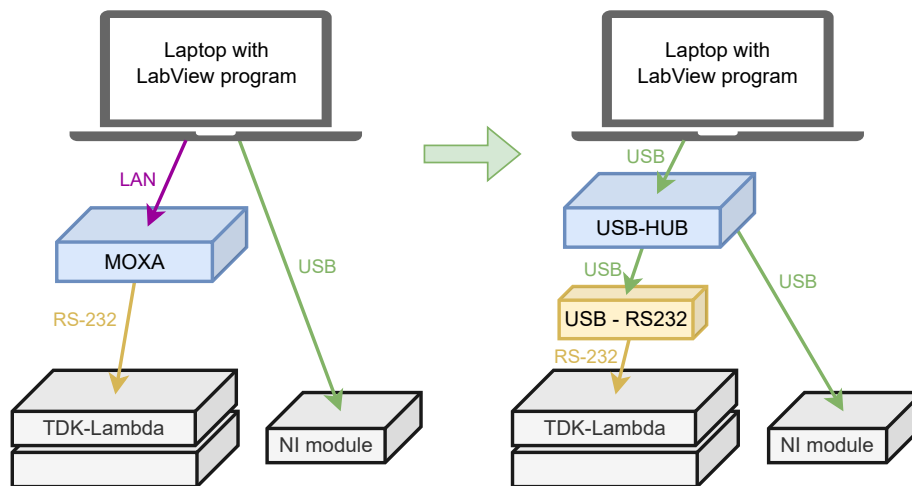


Figure 3.10: Communication architecture change due to malfunctioning MOXA server

After resolving the communication issue, I studied the example projects and then implemented my own VIs in the same style as the PMLA load, using the same data structures for acquiring data. Working with this device and its data was easier since each supply has a single output channel – two channels in total.

■ 3.8.3 NI USB module control

The module NI USB-6001, used for temperature regulation in the inductive load unit, is made by National Instruments. Therefore, communication with LabVIEW worked smoothly from the very beginning. The only issue was that this device needs an additional USB port so I obtained a USB HUB [A.2.7] for maintaining a single USB cable connected to the laptop.

For this device, I have also implemented the data acquisition VI, ensuring the same data format structure as the previous devices. The difference here is that the digital output channels need to be adjusted (regulated) based on the measured data. This temperature regulation also needs to be independent of the measuring period that the user selected, so that it works in the same way every time. Thus, the regulation must be an asynchronous process and is implemented as a separate while loop in `Measure_Devices.vi` that is included in `DataView.vi`.

The temperature regulation is divided into three parts:

- **Regulation of each coil:** Each coil of the inductive load unit has its own dedicated fan and temperature sensor TMP37. When the measured temperature exceeds the setpoint temperature of 42°C, the fan is turned on by the corresponding digital output pin with the help of the PCB with MOSFETs. This simple regulation is often referred to as Bang-bang [22].
- **Support fans on the back:** If this temperature of 42°C is exceeded on at least one TMP37 sensor, the fans at the rear of the unit are also switched on. This improves airflow through the load unit when cooling is required.
- **Fans in resistor tunnel:** As noted earlier in 2.2.8, the power resistors are placed in a narrow tunnel with a fan on each side. These fans are turned on as soon as the tests starts – station transitions to MonitorTest state.

This regulation is executed once every 30 seconds. Therefore, implementing a hysteresis does not seem necessary because of this longer time period.

Another difference here is that during the initialisation, so-called “Tasks” are defined. Each task corresponds to a specific measurement or action and stores its settings (sample rate, number of AIs or DQs). During the measurement, the program just loads this task, runs it and then stops it after it is finished. This approach makes the initialisation longer but simplifies the data acquisition and regulation code. The following LabVIEW code 3.11 shows a virtual instrument called `ReadAI_NI` that runs and stops a task called `NI AI task` that was specially created during initialisation for this measuring purpose – using the analog inputs with the specified sampling frequency of 400 Hz and the number of taken samples set to 16, as explained in section about temperature measurement 2.2.5.

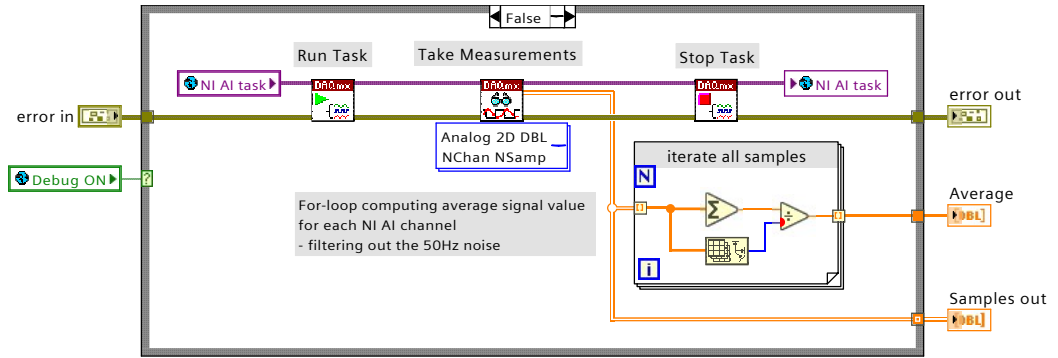


Figure 3.11: Block diagram of ReadAI_NI reading voltages of the temperature sensors TMP37

3.8.4 Excel report generation

Automatic report generation for performed tests is necessary. This allows the operator to maintain a convenient record of test details and outcomes. The report has an Excel .xlsx file format and contains several sheets. The first sheet (see Figure 3.12) summarises the important information such as:

- Date and Time
- Number of performed tests and the total time duration
- Main info about each test, including duration and number of enabled device outputs

SIEMENS

Load Station Test Report	
Author	-
Date	28 April, 2024
Time	13:38
Document Version	v1.0

Summary	
Number of Tests	3
Total duration (h)	07:45

Test 1					
Duration (h)	01:30	Power Supplies ON	2	Electronic Load channels ON	1
Test 2					
Duration (h)	04:00	Power Supplies ON	1	Electronic Load channels ON	4
Test 3					
Duration (h)	02:15	Power Supplies ON	2	Electronic Load channels ON	16

Figure 3.12: The first summary page of the automatically generated report

Each following page of the report corresponds to one performed test and contains more information and the measured data during the test. Here, each column corresponds to a device channel and each row contains measured data over time. A small cutout of such a page is shown in Figure 3.13.

Note that the number of enabled channels from the summary page 3.12 corresponds with the measured data in the page 3.13. The report was also

generated only three minutes after the test started, which can be said from the “End” value.

Test 1	
Duration	01:30:00
Start	13:35:31 28/04/2024
End	13:38:31 28/04/2024

Measured Data									
Time from start (min)	Power Supply				Electronic Load		Inductive Load Temperatures		
	Channel 1		Channel 2		Channel 1		Channel 1	Channel 2	
	Current (A)	Voltage (V)	Current (A)	Voltage (V)	Current (A)	Voltage (V)	Temp (°C)	Temp (°C)	
13:35:31	2	28.8	2	28.8	2.09	29.20	40	45	
13:35:41	2	28.8	2	28.8	2.02	28.85	40	45	
13:35:51	2	28.8	2	28.8	2.06	29.15	40	45	
13:36:01	2	28.8	2	28.8	2.02	28.98	40	45	
13:36:11	2	28.8	2	28.8	2.10	29.10	40	45	
13:36:21	2	28.8	2	28.8	2.09	29.18	40	45	
13:36:31	2	28.8	2	28.8	2.06	29.09	40	45	

Figure 3.13: The second page corresponding to Test 1 with the measured data

I will now explain the basics of how the code behind this report generation works. I am using the Report Generation Toolkit for Microsoft Office that can be downloaded from the National Instruments website [23]. This toolkit provides a palette of VIs that are very easy to use for creating documents in formats like Word, Excel, and HTML. However, it does have some limitations and can only perform a small fraction of editing and actions available in the real Excel application.

The block diagram code is structured into four sections, where each of them is handled by specialised VI, as shown in the block diagram 3.14.

1. **Create a new Excel from template.** A new Excel file is created based on a clean Excel template located in the `ConfigFiles` folder in the project root directory. The code also generates the name of this file based on the current date and time. Afterwards, the second Excel sheet (in format shown in Figure 3.13) needs to be copied as many times as there are performed tests. The Report Generation Toolkit does not have this ability so I needed to implement a simple supporting Python script that copies the Excel sheet multiple times and renames them accordingly. The script called `copy_worksheet.py` is launched using `System Exec.vi` from `Connectivity` palette.
2. **Generate the 1st summary sheet.** This VI fills in the date and time information, then iterates through the tests from the queue and computes the corresponding summary information. It also formats the text, including fonts and background colours, since the toolkit does not preserve this formatting.
3. **Generate the following test sheets.** Here, the code iterates through the tests again and fills in the measured data from the global variable `Measured Data` where they are stored by the `DataView.vi`.
4. **Save and close the Excel file.** Finally, the changes are saved, and the Excel file is brought to the front so the user can see it and save it to his flash drive, for example. But before that, another Python script is

run. Due to the fact that there are eighteen channels of the electronic load, only a few of them can be actually used, and the report would contain a lot of empty columns. To resolve this issue, I wrote a Python script `delete_cols.py` that deletes these empty columns based on the information stored in the `TestClass` instance from the queue of tests. Again, the toolkit does not have this editing feature.

Both Python scripts use the `openpyxl` library [24] to read and modify the Excel `.xlsx` files. The scripts are placed in folder `ConfigFiles` in the LabVIEW project directory together with the Excel template.

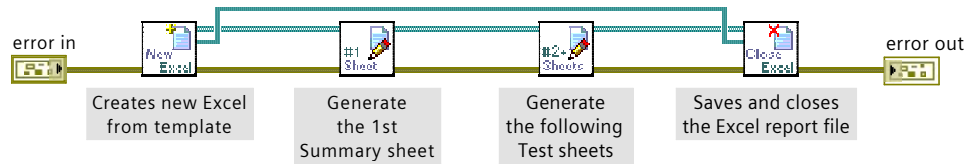


Figure 3.14: Block diagram of the `CreateExcelReport.vi` with the code divided to four parts

3.9 Conclusion

Throughout the software implementation process, I created simple yet clear icons for the implemented SubVIs. These icons help to improve the clarity of the code and give an idea of what a given block of code does. They are stored in the root directory in the `Pictures` folder. I have provided a few examples of these icons in Figure 3.15. The first icon on the left represents



Figure 3.15: Example of few icons used in LabVIEW implementation to improve code readability

the initialization VI of the TDK-Lambda power supply. The second one stands for the `DataView.vi`, which stores and displays real-time measured data. The last icon represents VI for creating a comprehensive Excel report.

Additionally, I want to connect the station to the Keysight 34980A multifunction measure unit [25] in the future. The Siemens laboratory is equipped with several of these units. It is used to measure the temperature of specific parts and components of the tested PLC device inside the climatic chamber. This measuring unit is also equipped with LabVIEW drivers, and the acquired data will be added to the Excel report document. Therefore, the operator would have all the important measured data embedded in a single document.

Chapter 4

Experiments

After completing the construction of the entire station and implementing the control program, I conducted a series of experiments to verify the station's functionality. The following experiments were performed to verify the temperature regulation of the inductive load unit. All of them are recorded and available in the attached folder `Experiments`.

4.1 Noise temperature measurement

As discussed earlier, the temperature measurement from the TMP37 sensor has an uncertainty of $\pm 1.04^\circ\text{C}$, as specified by equation 2.8. However, this estimate does not consider potential noise interference in the sensor's signal. Such noise is typically emitted by the switching power supplies of various devices commonly found in the Siemens laboratory. Consequently, this interference can impact the accuracy of the measurements, as low-voltage temperature sensors like the TMP37 are highly sensitive to external noise. To determine this additional uncertainty, I conducted experiments to measure the output signal of all temperature sensors, while they were connected to the inductors, in the same way as they would be during the station's operation.

4.1.1 With integrated signal averaging

In the first experiment, I measured the TMP37 output signal with the averaging mechanism discussed in 2.20, taking sixteen samples of the signal with a 400 Hz sampling frequency. The signal values were also already converted to degrees Celsius based on the sensor's linear characteristic of 20 mV/ $^\circ\text{C}$.

I measured more than 2000 samples of each sensor signal, shown in Figure 4.1, calculated its mean μ and then the Standard Deviation SD as

$$SD = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \quad (4.1)$$

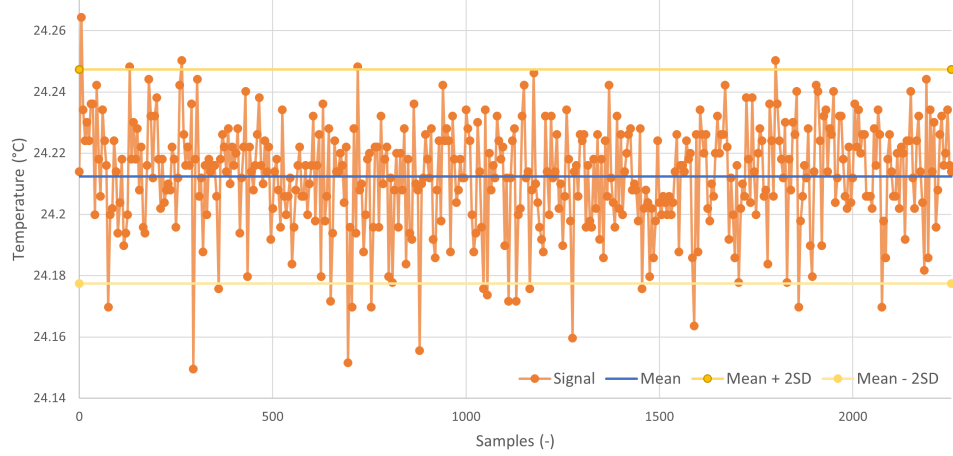


Figure 4.1: Measured output signal from TMP37 sensor with its Standard Deviation

where x_i represents a signal sample, μ denotes the mean value and N is the number of samples. Assuming that the interfered noise follows a Gaussian distribution, 95% of the measured samples should fall into the range of $\langle \mu - 2SD; \mu + 2SD \rangle$ on the y-axis. Therefore, the measured temperature T with its uncertainty can be expressed as

$$T = \mu \pm 2SD \quad (4.2)$$

After calculating the values for this particular signal, I have obtained values

$$T = (24.21 \pm 0.03)^\circ\text{C}$$

Then, following the same process, I computed the average standard deviation for the all other sensors, which resulted in 0.018°C . The average value of doubled standard deviation is

$$2SD = 0.035^\circ\text{C}$$

This value is very small compared to the previously calculated uncertainty of $\pm 1.04^\circ\text{C}$. That means that interference on the sensor wires does not significantly degrade the accuracy of the temperature measurement, and there is no need to implement additional systems such as Low Pass or adding shielded wires. The complete temperature uncertainty is still 1.04°C based on the following calculation.

$$T_{error} = \sqrt{T_{e1}^2 + T_{e2}^2} = \sqrt{1.04^2 + 0.035^2} \approx 1.04^\circ\text{C} \quad (4.3)$$

4.1.2 Disabled signal averaging

In the second experiment, I disabled the averaging feature to examine the impact of noise on the measured signal. The goal was to compare the results of the averaging approach with the raw signal. Over 2000 samples were collected from the TMP37 sensor and converted to degrees Celsius using its

20 mV/°C characteristic. The data are shown in the following graph 4.2. Only by looking at the y-axis, it is evident that the measured signal fluctuates more. Approximately by 0.6°C around the mean value, whereas in the previous experiment, the fluctuation was significantly lower – only by 0.1°C.

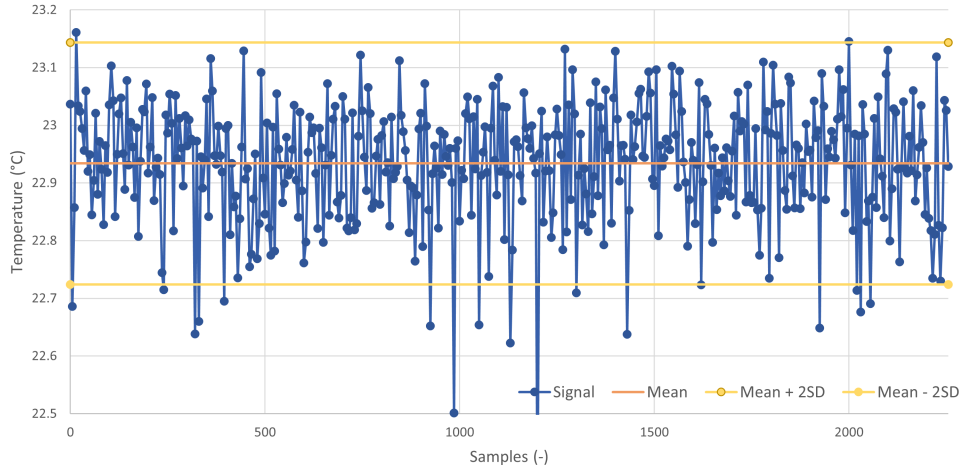


Figure 4.2: Measured output signal from TMP37 sensor without averaging approach

I computed the Standard Deviation (SD), and by multiplying the SD by two, the measurement uncertainty was determined:

$$T = \mu \pm 2SD$$

$$T = (22.93 \pm 0.21)^\circ\text{C}$$

This uncertainty is significantly smaller than the inherent uncertainty of the TMP37 sensor. Consequently, the overall measurement uncertainty remains largely unaffected:

$$T_{error} = \sqrt{T_{e1}^2 + T_{e2}^2} = \sqrt{1.04^2 + 0.21^2} \approx 1.06^\circ\text{C} \quad (4.4)$$

While the averaging mechanism does not substantially reduce the overall measurement uncertainty due to the inherent sensor uncertainty, it significantly enhances measurement reliability. It reduces the likelihood of individual samples deviating significantly from the true value, as illustrated by sample number 1200 in the following graph 4.3.

The graph compares the two approaches—averaging versus non-averaging—by plotting the absolute deviation from the mean value of each sample on the y-axis.

In summary, the averaging approach improves the quality of temperature measurements by reducing the influence of noise by a factor of seven. However, it does not significantly affect the overall measurement uncertainty due to the TMP37 sensor's inherent uncertainty.

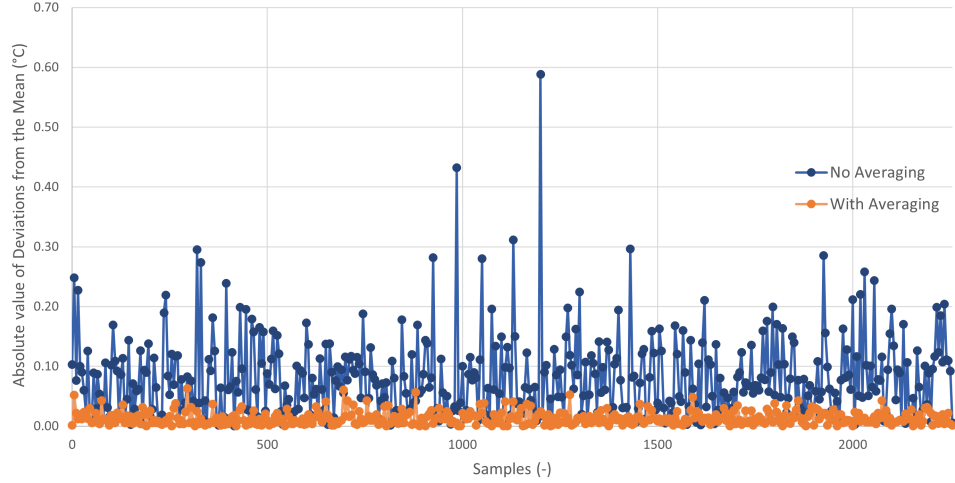


Figure 4.3: Comparison of deviation for each measured sample for both approaches

4.1.3 Temperature Regulation

I conducted multiple experiments heating the coils in the inductive load unit to verify the functionality of temperature regulation and to eventually re-calibrate the load channels. Ensuring a consistent 2A current flow (or 0.5A) throughout the test is essential. To test the PLC device under worst-case conditions, the current must not drop below 2A at any point. It is only acceptable to have a current flow slightly above 2 A to test the PLC edge conditions.

1. The first experiment included heating of two inductors L_2 and L_7 . The entire experiment is recorded and attached as `Temp_Regulation_01.xlsx`. The document was automatically generated by the LabVIEW program as described in 3.8.4 section. Inductor L_2 was supplied by 2 A and L_7 by 0.5 A DC current. This is not a standard use-case because the PLC equipment under test switches the digital outputs on and off with a 50% duty cycle, resulting in smaller power dissipation on the inductor. I conducted this experiment to see if the system is capable of cooling the coils even with this DC current signal with double the power dissipation. After 15 minutes, the temperature reached the defined temperature operating point, and the current flow was stabilised, as shown in graph 4.4. The visible zig-zag pattern is due to a simple temperature control where the fan is either fully on or off. However, for the 2 A version, the current started from value 2.12 A and dropped below the limit to 1.95 A. Based on Ohm's law, the load channel has 13.7Ω at 20°C and 14.7Ω at 45°C . Using the equation that describes copper material 2.6, the expected change in resistance ΔR is

$$\Delta R = \alpha \Delta T R_0 = 3.92 \cdot 10^{-3} \cdot 25 \cdot 7.6 = 0.76 \Omega \quad (4.5)$$

but the actually measured ΔR is 1Ω . This is probably because the coil's core is significantly hotter than the measured surface of the coil, up to

approximately 8°C . That can be calculated after rearranging equation 4.5 so that now is unknown the change in temperature ΔT instead of the change in resistance ΔR . The value 0.24 is the difference between measured and expected resistance.

$$\Delta T = \frac{\Delta R}{\alpha R_0} = \frac{0.24}{3.92 \cdot 10^{-37.6}} \approx 8^{\circ}\text{C}$$

Based on this fact, all load channels for 2 A must be re-calibrated 0.3Ω lower so that the channel really has 14.4Ω at the stabilised temperature.

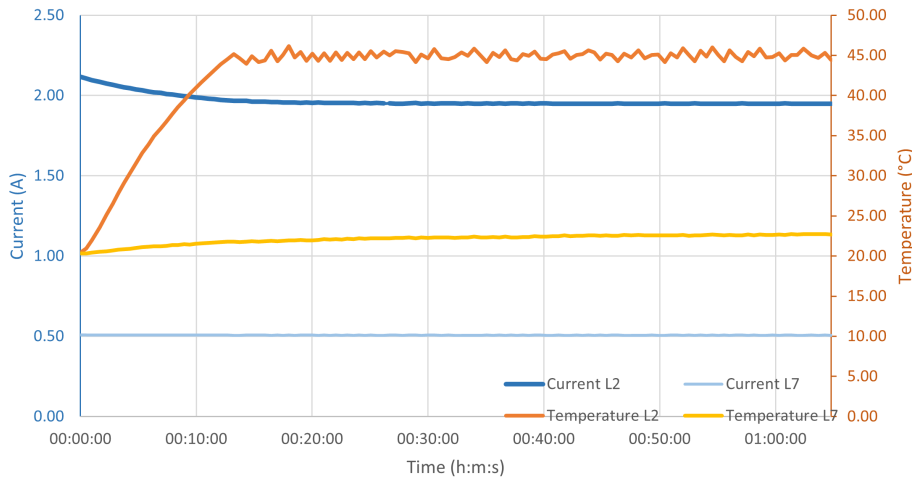


Figure 4.4: Temperature regulation of inductors L_2 and L_7

2. In the second experiment, two inductors were loaded with a switching current of 2 A at a frequency of 0.5 Hz and a duty cycle of 50%. To conduct the experiment, I set up a rack with PLC equipment (SIMATIC S7-1500) and created a project in the TIA Portal program with simple code for periodically switching the digital outputs.

As expected, the coils heated up at a slower rate under this load, about twice as slowly. The signal has an I_{RMS} equal to $\sqrt{2}$ A and half the power dissipation. This particular load with a 50% duty cycle is the common use case for testing the PLC modules. Therefore, I lowered the temperature operating point from 45°C to 42°C so that the coils can comfortably warm up to the desired temperature in about 37 minutes instead of the original 60 minutes. It is important to tune the temperature regulation specifically for this load case.

3. The third experiment included loading all inductors to verify the cooling abilities of the whole system. Inductors L_1 to L_4 were loaded by switching signal 2 A, 0.5 Hz, 50% duty cycle and I_{RMS} of $\sqrt{2}$ A. While the other half of inductors L_5 to L_8 were loaded by signal 0.5 A, 0.5 Hz, 50% duty cycle and I_{RMS} of $\sqrt{2}/4$ A. This coil configuration is also designed to investigate the potential problem of temperature influence between the coils L_1 – L_4 , that are physically close together. The measured temperatures are shown in Figure 4.5

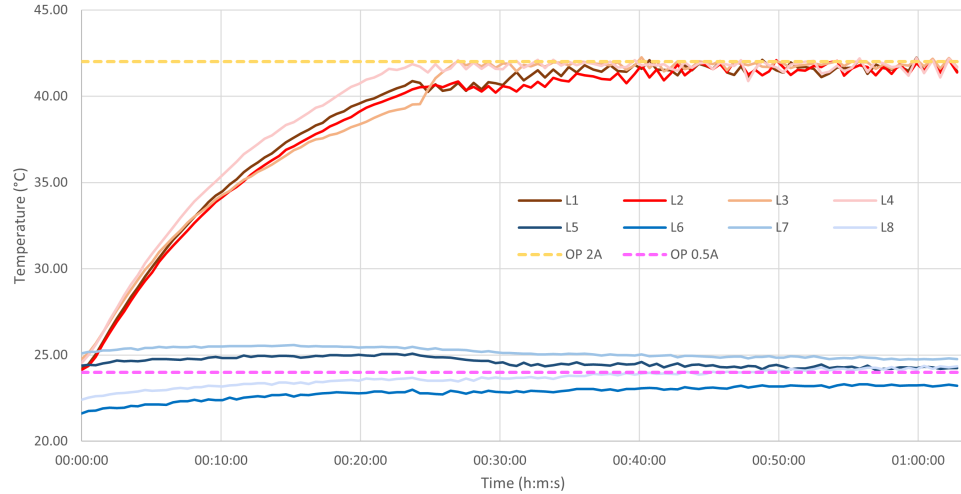


Figure 4.5: Temperature regulation with loading of all inductors simultaneously

After approximately 25 minutes, the temperature of all coils stabilised at the defined setpoint of 42°C and 24°C. With the temperature stabilised around the setpoint, the current was fluctuating by only 0.01 A with a measuring accuracy of ± 0.014 A. The coils L_5 – L_8 , loaded with 0.5 A, had different temperatures at the beginning because of some previously conducted experiments. However, their temperatures eventually converged to the desired setpoint.

These results are very promising. However, this test only lasted an hour, and the temperature of coils L_1 – L_4 was very similar at the beginning of the experiment. I was concerned that if an inductor was already warmed up at the beginning of the experiment, it would need to be cooled down sooner. This cooling would likely affect the temperature of the surrounding coils, resulting in them not being able to warm up to the 42°C operating point. Additionally, I was unsure if I would achieve such good results in a longer test, where the air inside the unit could heat up, and the temperature regulation could work worse. Therefore, I conducted the fourth, more extended experiment.

4. During the fourth test, half of the inductors L_5 – L_8 were loaded by the switching current and the other half L_1 – L_4 by DC current with the same I_{RMS} value. I also arranged the coils so that each odd coil is loaded with 2 A current and the even coil with 0.5 A current (or corresponding DC value). This arrangement serves the purpose of testing the interaction between the coils and detecting any potential issues. The coils were set up the other way around during the previous experiment.

Before the fourth experiment was conducted, I preheated the inductor L_1 to 32 degrees Celsius. Therefore, it reached the temperature setpoint much sooner, only after 19 minutes and its corresponding fan started to cool it down, to maintain this temperature (see graph 4.6). This cooling also affected one of the adjacent coils, namely L_5 , which could not reach the operating point fast enough. This is due to the fact that the coils are not physically separated from each other, as seen for example in

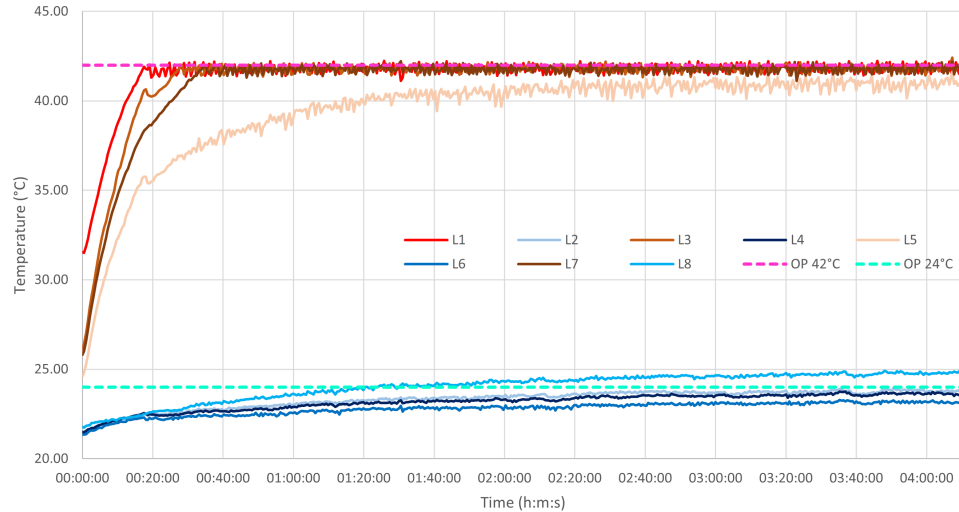


Figure 4.6: Long experiment of temperature regulation with loading all inductors

Figure B.4. Fortunately, if the coil is significantly colder than desired, its resistance will be lower, and the current passing through will be higher, according to Ohm's law. Alternatively, if the inductor was hotter than desired and the current lower than needed, the entire test of the PLC equipment would be useless since the PLC would not be exposed to the critical conditions. It is essential to ensure that the current does not drop below 2 A (or 0.5 A) during the test. This longer experiment was also carried out to verify that prolonged use of the load unit would not cause the air inside to heat up significantly, leading to the degradation of the regulation's properties. However, this was not a problem as the ventilation system successfully replaces the heated air with fresh air, based on the measured data.

In summary, the temperature regulation is working correctly as designed. The temperature typically stabilises after 30 minutes, and the current flow fluctuates less than 0.02 A, which is reasonable. In the future, I plan to equip the unit with sheet metal barriers to reduce the thermal interaction between the inductors so that the discussed problem that occurred between inductors L_1 and L_5 is eliminated. Eventually, I plan to add a heat-up mode, that would prepare the temperature of all coils so that all inductors are then immediately ready to perform tests. The station operator would load all inductors before the actual test, and the station would stabilise the temperature and notify the operator that it is ready to perform tests.

4.2 Testing entire station and running tests

This last experiment involves testing the entire station, from start to finish. I will use this experiment to summarise the functionality of the entire station and explain how the PLC devices are tested.

1. **Preparing the PLC:** Initially, I prepared the PLC device for testing. For simplicity, the device was not placed in a climatic chamber, which is typical in real tests. From the first DQ module (DQ 16x24VDC/0.5A), four outputs were connected to the electronic load channels (1-4). From another DQ module (DQ 8x24VDC/2A HF), two outputs were connected to electronic load channels (13-14), and another two outputs were connected to inductive load channels (1-2), which were set to 2A mode by mechanically connecting the corresponding resistor with the inductor (see Figure 2.31). Then, I connected the first power supply with the PLC CPU and its DQ modules to provide their power during tests. This setup is shown in the following Figure 4.7. In the TIA portal, I wrote code that turned on the outputs connected to the electronic load and periodically switched the outputs connected to the inductive load at a frequency of 0.5 Hz with a 50% duty cycle. The TIA code is available in the attached `Experiments` folder.
2. **Powering up the station:** I connected the three-phase power supply (16 A) from the laboratory power distribution cabinet (see Figure 2.27) to the rear of the station. Then I turned on the main switch. This illuminated the red signalling light and the front of the station, indicating the station was powered on. Then, I pressed the green Start button, enabling power to all devices. Subsequently, I turned on all necessary devices (electronic loads, power supplies, laptop). On the laptop, I launched the control program `Load Station Control.exe`. The program's interface appeared, and shortly after, a message in the `Status` window confirmed the successful initialization of all devices.
3. **Running the tests:** In the `Test Setup` menu, I defined three tests to match the connected PLC setup. The defined tests are summarised in Table 4.1.

	Duration (h:m)	Power Supplies ON	Electronic Load channels ON	Loaded inductive channels
Test 1	0:20	1	3×0.5 A + 2×2 A	0
Test 2	1:00	1	1×0.5 A	2×2 A
Test 3	0:50	1	2×0.5 A	0

Table 4.1: Three defined tests for the experiment illustrating the operation of the station

It is important to note that switching the inductive loads is defined only in the PLC program. The station only regulates its temperature automatically. I then moved to the `View Tests` menu, where I reviewed the defined tests and saved them in XML format by pressing the corresponding button. To repeat these tests in the future, this XML file

can be easily loaded into the program. Finally, I started the tests in the program, and the PLC device was activated simultaneously.

4. **Monitoring the tests:** I switched to the **Data View** menu, where I could see the measured data of all devices in the graph in real-time. To the right of the graph, I could check how long the test was running and how much time was left. After 20 minutes, the Status in the upper right corner indicated that the first test had ended and the second test had begun. During the second test, I also manually measured current through the inductive load in five-minute intervals, using the current probe [26]. I switched the graph on the laptop to see the inductors' temperature and see when it stabilises. After the one hour elapsed, the status window again indicated that the second test had ended and the third test had begun. The inductors were not loaded during the third test, and I could see their gradual cooling in the graph. After 30 minutes, I decided it was enough to illustrate the station's operation and, therefore, stopped the testing.
5. **Termination:** Before shutting down the program, I generated an Excel report with a single button press in the **Data View** menu. The generation took about four seconds, and the report was saved to a folder on the desktop and left open for me. Finally, I shut down the program by pressing the **Stop** button. The status window reported that the station was terminating, all devices turned off properly, and the entire program closed.

The experiment summarised and verified the operation of the entire station without any issues. The saved tests and generated test report are attached to the thesis as `TestQueue.xml` and `TestReport.xlsx`.

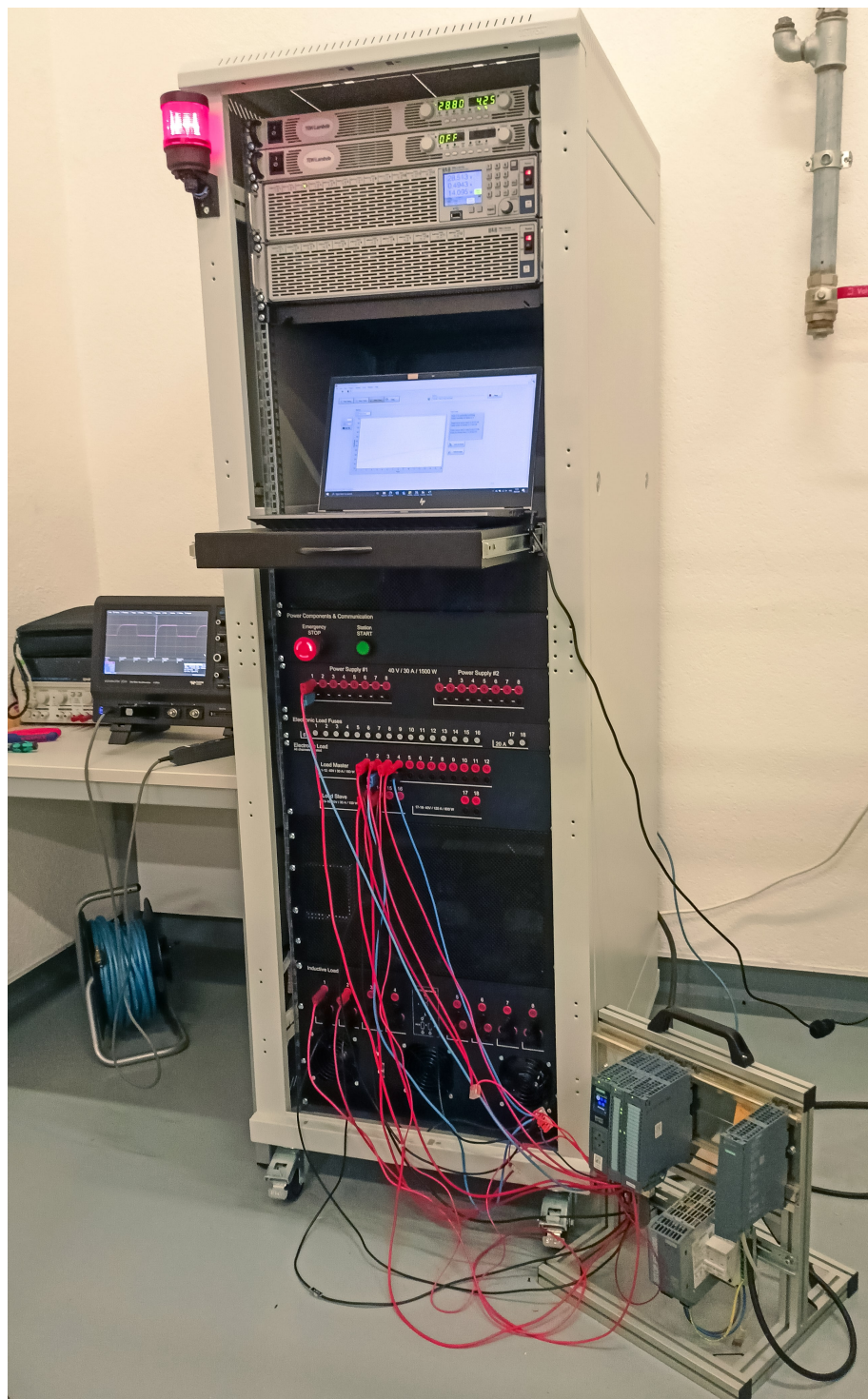


Figure 4.7: Photograph of the load station running the final experiment conducted for this thesis



Chapter 5

Conclusion

The thesis successfully achieved its objective of creating a fully functional stand-alone multichannel load station, aimed at testing various PLC equipment under harsh conditions. The resulting project integrates multiple instruments into a stand-alone system, that can be used with any climatic chamber in the laboratory. Key features include the ability to load, save or define a set of tests that will be automatically performed by the station. It also monitors real-time data from the devices and generates a comprehensive test report with all the measured data neatly arranged. Moreover, the temperature inside the inductive load unit is automatically regulated to ensure the resistance does not fluctuate over time, and the tested PLC device is loaded with the correct current. Not only the project demonstrates a significant advancement in load station design but also lays a solid foundation for future enhancements and applications. Overall, this thesis contributes to the field of industrial automation, showcasing innovation and practical engineering solutions

In the future I plan to expand the station with additional features. Firstly, I intend to install additional metal dividers into the inductive load unit to eliminate the thermal interactions between separate coils. I would also investigate if it would be beneficial to implement an automated warm-up mode for the inductive load, that would stabilise the inductors' temperatures and notify the operator that it is ready for actual test. Secondly, I am planning to integrate the Keysight device [25], which measures temperatures inside the climatic chamber to have all the important data embedded into a single Excel report document. This will make it much easier to track and analyse the data. Moreover, I am going to enable remote access to the laptop so that the station can be controlled and monitored from home. Lastly, I will be writing technical documentation of the entire station and help guides for the help menu in the LabVIEW program. These documents will also be uploaded to the Siemens website, unifying all employee manuals. In addition, I plan to request feedback from station users to further improve the LabVIEW interface and functionalities, or fix potential bugs.

Throughout the project, I learned how to design custom powering systems and gained valuable insights into the importance of their safety protections.

5. Conclusion

I dedicated a lot of time to selecting suitable hardware components, reading their datasheets and designing the mechanical components and housing of the station. This was also my first LabVIEW project and I learned the programming language as I went along. The gained knowledge will be useful when working on future projects in the industrial automation field.

Chapter 6

Glossary and Abbreviations

Term	Definition
PLC	Programmable Logic Controller popular in industrial automation.
DQ	Digital Output of a controller or module.
AI	Analog Input of a controller or module.
LabVIEW	Laboratory Virtual Instrument Engineering Workbench. A programmable environment by National Instruments.
VI	Virtual Instrument. Software module from LabVIEW, comprising block diagram code and front panel interface.
RTD	Resistance Temperature Detector, usually made of platinum.
PT100, PT1000	The most popular type of RTD with the resistance of 100 Ω and 1000 Ω respectively.
LAN	A Local Area Network that connects computers and devices in a limited area.
Ethernet	A family of networking technologies used in LANs, with the ability to handle high-speed data transfer.
RJ-45	A connector, typically used for Ethernet networking, featuring twisted pair cable with eight pins.
RS232, RS485	Recommended Standard for serial communication transmission of data popular in industrial automation.
USB	Universal Serial Bus. A popular communication standard used for powering and communication.
ping	A diagnostic tool command used in networking to test the reachability of a device on a network.
IEC	International Electrotechnical Commission standards for all electrical technologies.
DC, AC	Direct Current, Alternating Current.

Table 6.1: Glossary of Terms



Bibliography

- [1] “Example of a climatic chamber from laboratory WEISS-WK3,” <https://weiss-na.com/wp-content/uploads/Climate-Test-Chambers-WK3-WKS3-1.pdf>, accessed: 11-04-2024.
- [2] “Arduino UNO official website,” <https://store.arduino.cc/products/arduino-uno-rev3?queryID=undefined>, accessed: 15-04-2024.
- [3] “Arduino PRO Portenta Machine Control official website,” <https://store.arduino.cc/collections/pro-family/products/arduino-portenta-machine-control>, accessed: 15-04-2024.
- [4] “TO92 package of the TMP37 analog sensor,” <https://blog.mbedded.ninja/pcb-design/component-packages/to-92-component-package/>, accessed: 15-04-2024.
- [5] “Siemens DQ module 0.5A,” <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7132-6BH00-2AA0>, accessed: 16-03-2024.
- [6] “Siemens DQ module 2A,” <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7132-6BD20-0BA0>, accessed: 16-03-2024.
- [7] “Siemens IO Link Master 12A,” <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7148-6JE00-0BB0>, accessed: 24-03-2024.
- [8] J. Hospodka, “EPO FEL CVUT study material,” <http://hippo.feld.cvut.cz/vyuka/soubory/ElektronickeObvody.pdf>, accessed: 16-03-2024.
- [9] “IEC 60947-5-1:2016 standard,” <https://webstore.iec.ch/publication/24940>, accessed: 16-03-2024.
- [10] “Wikipedia temperature coefficient,” https://cs.wikipedia.org/wiki/Teplo%C3%AD_sou%C4%8Dinitel_elektrick%C3%A9ho_odporu, accessed: 12-04-2024.
- [11] “Siemens ET200SP PLC family,” <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10144488?tree=CatalogTree>, accessed: 24-03-2024.

- [12] “Arduino UNO Ethernet Shield official website,” https://store.arduino.cc/collections/shields-carriers/products/arduino-ethernet-shield-2?_pos=5&_fid=18df3921d&_ss=c, accessed: 15-04-2024.
- [13] “TE Connectivity PT1000 RTD sensor,” https://cz.farnell.com/sensor-solutions-te-connectivity/nb-ptco-006/rtd-sensor-thin-film-platinum/dp/2748862?pf_custSiteRedirect=true, accessed: 15-04-2024.
- [14] “Texas Instruments low-dropout voltage regulator TPS791,” <https://www.ti.com/product/TPS791?keyMatch=TPS79147DBVT&tisearch=search-everything&usecase=OPN>, accessed: 15-04-2024.
- [15] “JLCPCB service website offering custom PCB manufacturing,” <https://jlcpcb.com/>, accessed: 20-04-2024.
- [16] “TDK Genesys 40-38 User Manual,” https://product.tdk.com/system/files/dam/doc/product/power/switching-power/prg-power/instruction_manual/gen1u-750-1500w_user_manual.pdf, accessed: 13-02-2024.
- [17] “LabVIEW G programming language,” <https://labviewwiki.org/wiki/G>, accessed: 03-04-2024.
- [18] “LabVIEW Virtual Instruments (VIs),” https://labviewwiki.org/wiki/Virtual_Instrument, accessed: 05-04-2024.
- [19] R. Šusta, “LSP FEL CVUT finite space machine,” https://moodle.fel.cvut.cz/pluginfile.php/394855/mod_resource/content/3/LSP2023pr09_FSM_V11.pdf, accessed: 06-04-2024.
- [20] “PMLA load LabVIEW drivers 2011,” https://sine.ni.com/apps/utf8/niid_web_display.download_page?p_id_guid=45C21EC995EF0D3FE05400144FF9A492, accessed: 27-04-2024.
- [21] “TDK-Lambda power supply LabVIEW drivers 2017,” https://sine.ni.com/apps/utf8/niid_web_display.download_page?p_id_guid=F9944941B4CF18E6E0340003BA230ECF, accessed: 27-04-2024.
- [22] “Bang-bang Control,” https://en.wikipedia.org/wiki/Bang%E2%80%993bang_control, accessed: 15-05-2024.
- [23] “NI Report Generation Toolkit,” <https://www.ni.com/en/support/downloads/software-products/download.labview-report-generation-toolkit.html#411222>, accessed: 28-04-2024.
- [24] “Python library openpyxl for working with Excel sheets,” <https://pypi.org/project/openpyxl/>, accessed: 29-04-2024.
- [25] “Keysight 34980A multifunction measure unit,” <https://www.keysight.com/us/en/products/modular/data-acquisition-daq/digital-acquisition-benchtop-system/34980a-multifunction-switch-measure-mainframe-and-modules.html>, accessed: 03-05-2024.
- [26] “Teledyne Lecroy current probe CP030A,” <https://www.teledynelecroy.com/probes/current-probes/cp030a>, accessed: 10-05-2024.

Appendix A

List of all used components

Component name	Manufacturer	Part Number	Description	Official website URL
Temperature sensor	Analog Devices	TMP37	low-voltage linear temperature sensor	Farnell
LDO regulator	Texas Instruments	TPS79147-DBVT	Low-noise, low-dropout voltage regulator	Texas Instruments
MOSFET	INFINEON	IRLML5103TRPBF	P-channel power MOSFET	Farnell
Optocoupler	Toshiba	TLP291	single-channel 50mA optocoupler	Farnell website
Zener diode	Semtech	BZX55/C15	15V Zener diode for over-voltage protection	GME
Schottky diode	Yangjie	BAT43	30V Schottky diode for flyback protection	GME
Fan	Sanyo Denki	9A0924S402	DC axial fan 24V 150mA	Farnell
Fan finger guard	Multicomp PRO	MCSC92-W2B	Black, Steel Fan finger guard, wire form	Farnell
Two-pole Terminal block	Degson	DG381-3.81-02P	Two-pole terminal block for easy PCB connection	GME
Green LED	Lucky Light	S150UGC-1B	Green SMD LED in 1206 package	GME
Adjustable Power Resistor	Ohmite	D50K10RE	10 Ω , 50 W Slide Adjustable Power Resistor	DigiKey
Adjustable Power Resistor	Ohmite	D12K10RE	10 Ω , 12 W Slide Adjustable Power Resistor	DigiKey
Fixed Power Resistor	TE Connectivity	HSA2547RJ	47 Ω , 25 W Wirewound Chassis Mount Resistor	DigiKey

Table A.1: Low-level components used in the inductive load unit

Component name	Manufacturer	Part Number	Description	Official website URL
Socket connector	Phoenix Contact	804024	Pin & Sleeve socket connector for DIN rail mount	Farnel
Cartridge Fuse	Littlefuse	0312006.HXP	6A fast-acting cartridge fuse	Farnell
Cartridge Fuse	Littlefuse	0314020.HXP	20A fast-acting cartridge fuse	Farnell
Cartridge Fuse holder	Littlefuse	03453LS7H	Cartridge fuse holder with quick-connect and easy panel mount	Farnell
RJ-45 adapter	Amphenol LTW	RCP-5SPFFV-TCU7001	RJ-45 in-line adapter for easy panel mount	Farnell
Contactora	Siemens	5TT5030-0	Contactora with 4 NO contacts	Siemens
Circuit Breaker	Siemens	5SY4102-7	Miniature 1-pole Circuit Breaker	Siemens
RCBO	Siemens	5SU1346-7FP16	4-pole Residual Current Breaker with Overcurrent (RCBO)	Siemens
Main Switch	Siemens	3LD2003-0TK53	Station's Main switch, 3-pole	Siemens
START button	Siemens	3SU1106-0AB40-1BA0	Illuminated green pushbutton for easy panel mount	Siemens
STOP button	Siemens	3SU1158-1HB20-1PT0	Emergency mushroom stop pushbutton for easy panel mount, rotate to unlatch	Siemens
Three-phase Socket	BALS	282001	5-pin 16A/400V power socket	K&V Electro
Three-phase Outlet	BALS	132001	5-pin 16A/400V power outlet	K&V Electro

Table A.2: Other Low-level components used in station design

Device name	Manufacturer	Description	Official website URL
Laptop HP ZBook Fury	HP	Laptop controlling the entire station via LabVIEW program	HP
Lambda Genesys 40-38	TDK	Programmable power supply with LabVIEW drivers	TDK User Manual
PMLA C	Hoecherl-Hackl	Programmable electronic load with LabVIEW drivers	H&H
DGS-1100-05V2	D-Link	Smart managed Gigabit LAN switch	D-Link
Nport 5250A	MOXA	2-port device server converting LAN to RS-232	MOXA
USB-6001	National Instruments	Data acquisition module perfectly compatible with LabVIEW	National Instruments
USB HUB	i-tec	Metal HUB with 3x USB ports	i-tec
SITOP PSU300S	Siemens	Stabilised 24V / 5A industrial power supply	Siemens

Table A.3: Advanced high-level instruments and devices used in station

Appendix B

Photos of the finished load station



Figure B.1: The completed load station with a red light on, indicating it is turned on

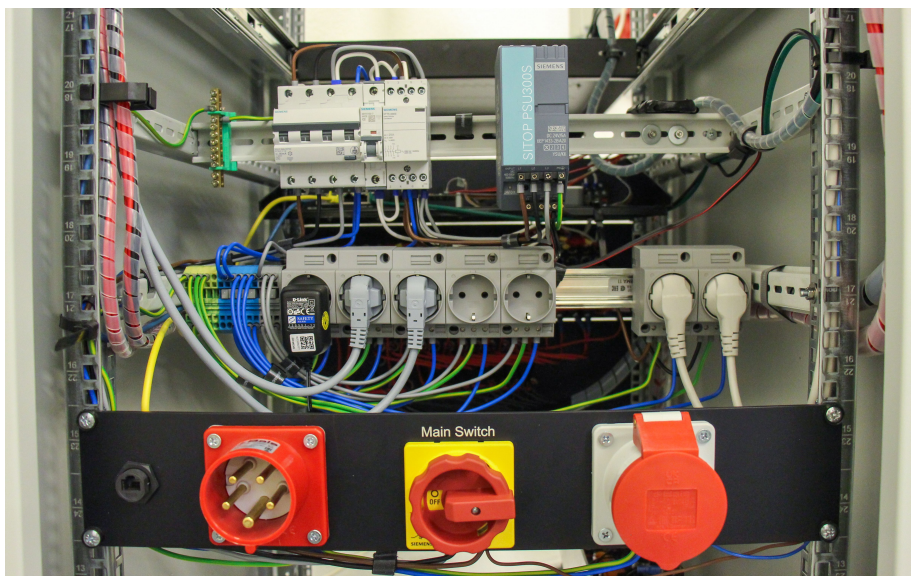


Figure B.2: Station power wiring with main switch and circuit breakers

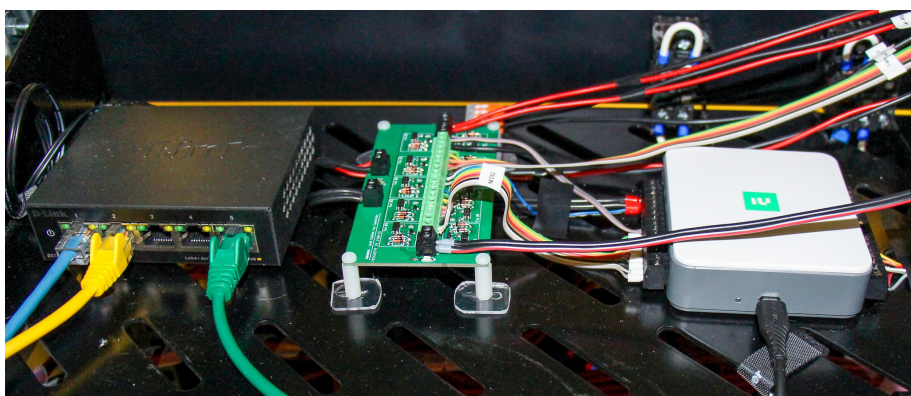
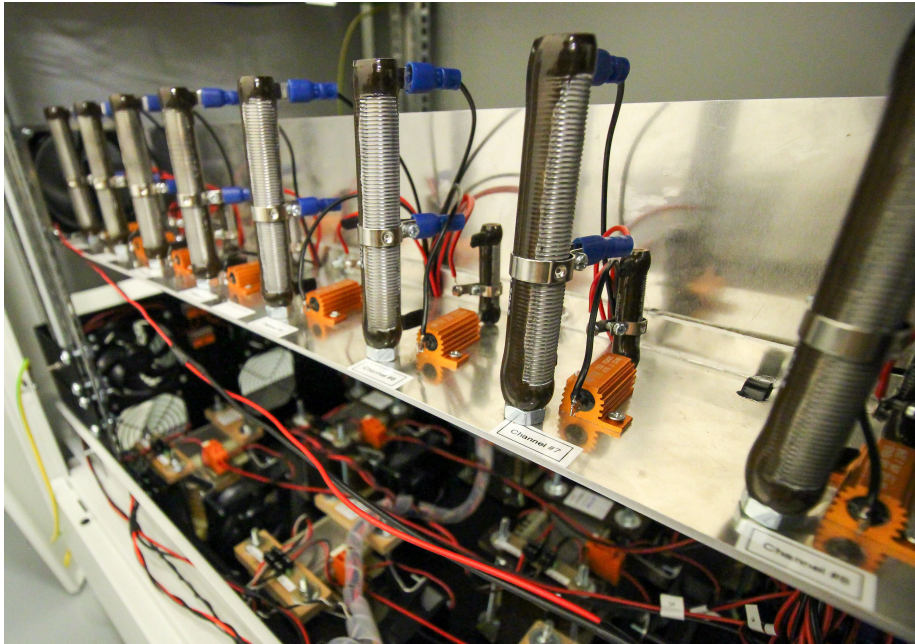
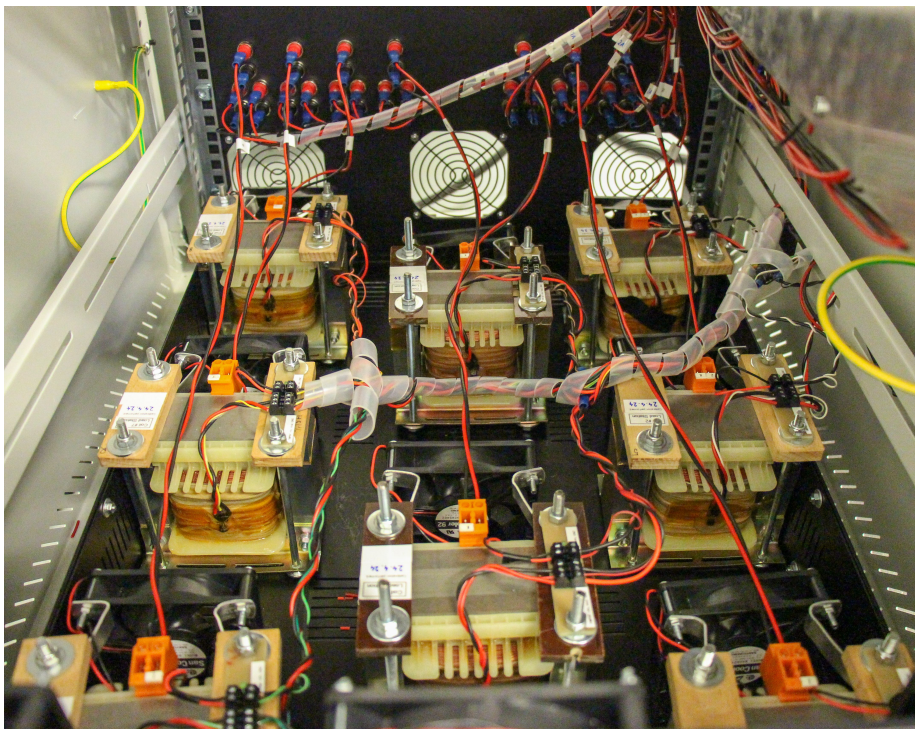


Figure B.3: Shelf behind the “Power & Communication” panel containing LAN server, PCB and NI module



(a) : Cooling tunnel with adjustable power resistors



(b) : Eight inductors with attached temperature sensors and fans

Figure B.4: Assembled inductive load unit