YRKESHÖGSKOLAN NOVIA

Automatization of a System

Modification of an Armfield C6 Fluid friction apparatus

Maria Masot Posino

Degree Thesis for Bachelor of Engineering

Degree Programme in Energy Technology

Vaasa 2022

DEGREE THESIS

Author: Maria Masot Posino
Degree Programme and place of study: Electrical engineering and automation, Lleida
Specialisation: Energy Technology
Supervisors: Hans Lindén and Philip Hollins

Title: Automatization of a System: Modification of an Armfield C6 Fluid friction apparatus.

Date: 16.05.2022
Number of pages: 53
Appendices: 6

Abstract

In nature, all processes include some sort of energy-matter interaction. As a result, learning the fundamental principles of thermal-fluid sciences is an important aspect of engineering education. This thesis aimed to develop a user interface to increase the usability and precision of an Armfield C6 Fluid Friction Apparatus.

The system improvement consisted of upgrading the pump to a MAGNA3 Model D and installing three pressure sensors to substitute the previous manual measurement. The pump is controlled and monitored remotely through a CIM500 ethernet with Modbus TCP protocol. Assuming that a LabVIEW licence and the tools are already owned, the indicative cost of this project is 2082€ in VAT.

The interface communicating with the pump and controlling the acquisition of the measurements has been developed in the LabVIEW environment. The LabVIEW code is responsible for reading and processing the data received. Moreover, the code developed also implements peripheral functions to increase the convenience of the user interface.

As a result, the data from the upgraded Armfield C6 Fluid Friction Apparatus can be comfortably accessed, monitored, and saved from the user interface. The option to modify the pump settings is also given.

Language: English
Key Words: Fluid Friction Apparatus, Modbus TCP, LabVIEW, interface

Table of Contents

List of Figures

List of Tables

1 Introduction

In recent years, Finland's education has been established as one of the best worldwide, performing at world-class levels in most international tests like PISA (Oecd.org, 2018). Many factors, including a body of highly qualified teachers, have contributed to Finland's success, according to Hani Morgan (2014).

One of the aspects that Finnish education promotes is the right balance of technical and professional skills. To fulfil this purpose, the practical application of the knowledge acquired in the classroom is seen as a key part of academic life, and the students undertake multiple study credits in carrying out practical activities in the laboratories and workshops.

The academic centres tend to be very well equipped, with several educational tools that help the comprehensive learning purpose. However, obsolescence is inevitable over time, and, as the obsolescence process advances, the formative effectiveness of the educational tools decreases. The study conducted by Vergara et al. (2020) suggests that technology obsolescence negatively affects the student's interactivity and motivation.

Nowadays, protocols in academic laboratories tend to be heavily reliant on research staff and students to manually manipulate the tools, instruments and reagents (Holland and Davies, 2020). One way to reinvent the educational tools and increase the safety and efficiency of both the instruments and the people who use them is automation.

The Oxford Learner's Dictionary (2022) defines automation as "the use of machines and computers to do work that was previously done by people". Historically, automation has relieved humans from complex, dangerous, heavy, boring, or time-consuming tasks, helping both in production processes and supportive tasks (Frohm, 2008).

To serve its purpose, automation uses sensors to measure and collect physical data, which is treated by a computer. The data is collected and processed by a computer program designed to respond accordingly and produce the desired result. Interfaces are created to allow operators to monitor the automatized system and adjust it manually if necessary.

Novia University of Applied Sciences, based in Vaasa (Finland), is currently using an Armfield C6 Fluid friction apparatus in the Energy Technology Laboratory. The system is mainly used to conduct educational exercises in the framework of fluids mechanics.

1.1 Aims and objectives

This project was commissioned by Novia University of Applied Sciences, specifically the battery technology laboratory. This thesis aims to automatize a fluid system apparatus using different types of sensors and create an interface with LabVIEW.

The laboratory currently has an Armfield C6 Fluid friction apparatus. The system can be operated, but some precision problems have been detected. The ultimate goal is to enhance the existing system into a fully developed valuable educational tool.

To achieve the aim of this thesis, the following focuses are defined:

- Identify and install a range of new components to the system
- Develop a LabVIEW program capable of receiving information from the sensors and sending actions to the system
- Design an intuitive and user-friendly interface for students to control the system
- Perform validation testing to ensure that the system is more precise and user-friendly

1.2 Document structure

This thesis begins with a brief description of automation and a general overview of the context, purpose, and objectives of the project (Section 1). Section 2 aims to introduce the educational tool that is going to be automated and the theoretical concepts related to the system's usefulness.

The theoretical aspects related to the new implementations of the educational tool are explained in Section 3. This section includes an overview of the added hardware, an explanation of the Modbus protocol and a brief introduction to LabVIEW.

Section 4 itemizes and details the procedure followed to achieve the final result. Therefore, the hardware assembly, the LabVIEW program and the interface design are explained in this section. Finally, Section 5 and Section 6 evaluate the outcome of the project and future implementations.

2 Study background

In order to improve the existing Armfield C6 Fluid friction apparatus, it is important to have previous knowledge about the system, its use, performance, advantages and inaccuracies.

This section is a review of the current state of the system (prior to this thesis) and an evaluation of its pedagogical objectives. Section 2.1 is an overview of the fluid mechanical concepts that the students can put into practice with this apparatus. The detailed description and the operation process of the instrument are developed in Section 2.2.

2.1 Fluid Mechanic principles

Thermal-Fluid Sciences (or simply thermal sciences) are the physical sciences that study both the energy and the transfer, transport and conversion of energy (Çengel et al., 2012). Thermodynamics, heat transfer, and fluid mechanics are the three classical disciplines of thermal-fluid sciences (Turns, 2006).

In nature, all processes include some sort of energy-matter interaction. As a result, learning the fundamental principles of thermal-fluid sciences is an important aspect of engineering education.

The concepts that apply to the apparatus that is the object of study of this thesis belong to the discipline of fluid mechanics. In the book "Fundamentals of thermal-fluid sciences" (2012), considered one of the referents in the field, Çengel et al. define fluid mechanics as "the science that deals with the behaviour of fluids at rest (fluid statics) or in motion (fluid dynamics), and the interaction of fluids with solids or other fluids at the boundaries".

The science of fluid movement in which density and temperature changes are relevant is known as gas dynamics (Rathakrishnan, 2013). Aerodynamics, for example, is one of its fields.

Hydrodynamics is the study of the movement of fluids and their causes. This topic's analysis encompasses the fluid's mass itself, energy in various forms, the quantity of motion, and the kinetic moment. Hydraulics, a subclass of hydrodynamics that deals with liquid flow through pipes and open channels, is the subject of study in this section (ReadPhysics, 2021).

According to Douglas et al. (2005), fluids are substances whose molecules are subject to lower forces of attraction. As a result, a fluid is unable to maintain any unsupported shape; it flows under its own weight and adapts to the shape of the body that contains it. Depending on the adaptation to the containers (which depends on the intensity of particle attraction) fluids can be in two states: liquid and gas.

Fluid flow is classified as external or internal, depending on whether the fluid flows inside a pipe or channel, or over a surface (Chen and Lobo, 1995). The following explanations will be made considering and internal flow where the pipe is completely filled with an incompressible fluid (like water).

In a pipe, the fluid velocity is not constant at all the points. The velocity is zero at the walls due to the no-slip condition and increases until reaching a maximum at the pipe centre, as shown in Figure 1.

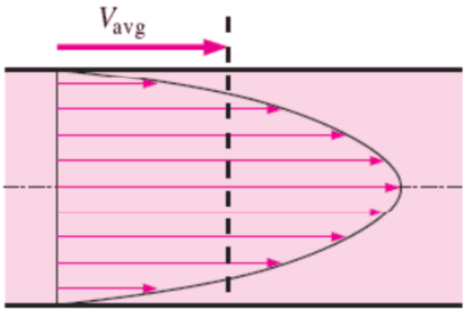


Figure 1: Velocity distribution of a fluid in a pipe (Source: https://www.thermal-engineering.org)

In fluid flow, it is convenient to work with an average velocity V_{avg}, which remains constant when the cross-sectional area of the pipe is constant.

There are two types of fluid flow: laminar and turbulent. As described in Fox and McDonald's (2020) book, a laminar flow is one on which the fluid is structured in superimposed parallel layers, each moving in relation to the two adjacent ones. The turbulent flow is characterized by turbulence, that is, the locally chaotic behaviour of fluid particles. The transition from laminar to turbulent flow does not occur suddenly; there is a region where the flow fluctuates between laminar and turbulent flows before becoming fully defined (Wygnanski and Champagne, 1973). This transition depends on various things such as geometry, surface roughness, flow velocity, surface temperature, and type of fluid.

After exhaustive experiments in the 1880s, Osborne Reynolds discovered the ratio of inertial forces to viscous forces in a flow (Jackson and Launder, 2007). This dimensionless ratio, known as the Reynolds number, is expressed for internal flow in a circular pipe as

$$Re = \frac{Internal\ forces}{Viscous\ forces} = \frac{\rho\ v\ D}{\mu} \tag{1}$$

Where ρ is the density of the fluid, v is the average velocity of the fluid, D is the pipe's internal diameter, and μ is the fluid's dynamic viscosity

Although the limit values of Re for each type of flow vary by author, the following can be used as reference values:

Laminar flow: Re<2300

Transitional flow: 2300 < Re < 400

Turbulent flow: Re > 4000

When a fluid flows through a conduit, the friction causes the fluid to lose hydraulic energy, which is then converted into internal energy (Çengel et al., 2012). As a result, the fluid experiments a pressure drop. Pressure losses in the linear sections of the pipes can be calculated using the Darcy-Weisbach formula

$$\Delta P = f \frac{\rho\, v^2}{2} \frac{L}{D} \tag{2}$$

Where ρ is the density of the fluid, v is the average velocity of the fluid, L is the length of the pipe, D is the internal diameter of the pipe, and f is a proportional constant caller Darcy friction factor.

In a laminar flow, the Darcy friction factor f (also known as the Darcy–Weisbach friction factor) depends on Re, while in a turbulent flow is a function of Re and the relative roughness ϵ (the relative roughness is the relation between the absolute roughness k and the internal diameter of the pipe D). The friction factor f for fully developed laminar flow in a circular pipe is

$$f = \frac{64}{Re} \tag{3}$$

While for turbulent flow the friction factor can be calculated from the Colebrook–White equation (or simply the Colebrook equation).

$$\frac{1}{\sqrt{f}} = -2 \log\left(\frac{\varepsilon}{3{,}7D} + \frac{2{,}51}{Re\sqrt{f}}\right) \tag{4}$$

Where ε is the surface roughness, D is the pipe’s internal diameter, and Re is the Reynolds number calculated in equation 1.

The calculation of the friction factor f can become quite complicated, especially for turbulent flow (equation 4). The friction factor can also be obtained from the Moody diagram.

A real fluid has a certain viscosity, which means that it has certain internal friction. As a result, for a fluid to flow, there must be a pressure difference between the pipe's ends (Acharya Samudra, 2020). The Hagen–Poiseuille equation (also known as Poiseuille´s law) explains the movement of a fluid as shown in equation 5.

$$Q = \frac{\pi \, R^4 \, \Delta P}{8 \, \mu \, l} \tag{5}$$

Where Q is the volume flow rate of the fluid, R is the radius of the pipe, ΔP is the pressure difference, μ is the dynamic viscosity of the fluid, and l is the length of the pipe.

To be able to use Poiseuille´s law (equation 5), the fluid flowing should be (a) incompressible, (b) in laminar fluid flow and (c) flowing in a cylindrical pipe. If the requirements aren´t met, the Poiseuille equation can be used as an approximation.

From equation 5 can be deduced the following:

- The volume flow rate of a fluid is directly proportional to the pressure gradient.

$$Q \propto \frac{\Delta P}{l} \tag{6}$$

- The volume flow rate of a fluid is inversely proportional to the viscosity of the fluid.

$$Q \propto \frac{1}{\mu} \tag{7}$$

- The volume flow rate of a fluid is directly proportional to the internal radius of the pipe. A change in the radius results in a proportional change raised to the power of 4.

$$Q \propto R^4 \tag{8}$$

Despite extensive research into the fluid theory, precise theoretical solutions are only obtained for a few basic cases such as fully developed laminar flow in a circular pipe. Çengel et al. (2012) state that a 10% (or more) inaccuracy in the results produced using the equations provided in this section is to be expected.

Because no two systems are the same, properly collected experimental data may be the most reliable. A manometer is required to measure the pressure of the fluid in a pipe. One simple and ubiquitous type is the U-tube manometer. Two different fluids are involved in its functioning. The fluid that is transferring the pressure (therefore, whose pressure is being measured) and the reference fluid (usually mercury or water).

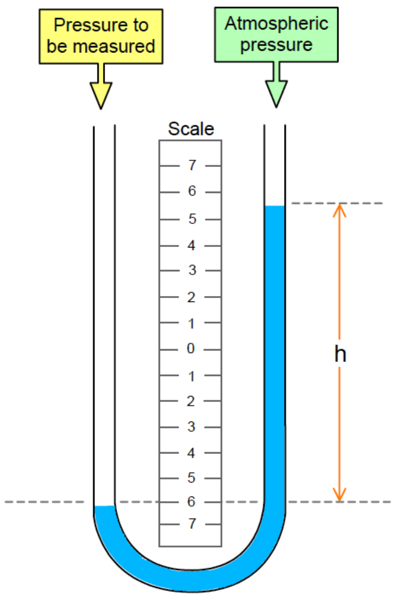


Figure 2: U-tube manometer (Source: https://instrumentationtools.com)

The working principle of the basic manometer is converting the pressure difference into height difference. Figure 2 illustrates the water levels in a U-tube with the left tube connected to a higher pressure point than the right tub (which is at atmospheric pressure). Once the reference liquid (represented in blue in Figure 2) has moved enough to balance the exerted pressure, the system reaches an equilibrium point.

The pressure can be calculated by measuring the height difference and applying the equation

$$\Delta P = |\rho_r - \rho_t| g \, \Delta h \tag{9}$$

Where ρ_r is the density of the reference fluid, ρ_t is the density of the transferring fluid, Δh is the height difference, and g is the Earth's gravity.

2.2 Armfield C6 Fluid friction apparatus

Piping systems are usually composed of pipes of varying diameters. These are connected to one other by several fittings or elbows to route the fluid, valves to control the flow rate, and pumps to pressurise the fluid. The Armfield C6 Fluid Friction device, which is the subject of this thesis study, is not an exception. Figure 3 is a photo of the actual apparatus.

Figure 3: Armfield C6 Fluid friction apparatus (Source: Author's own)

The Armfield C6 Fluid Friction Apparatus helps the purpose of studying the fluid friction head losses of an incompressible fluid. The fluid can be studied under different circumstances, as the system is formed by pipes of different sizes, shapes and conditions. Several valves, strategically positioned, allow the user to define the fluid's path.

To study the friction head losses in a straight pipe, the apparatus has four lines with different properties. The tubs are the following ones:

- 4 mm internal diameter smooth pipe
- 7 mm internal diameter smooth pipe
- 17,5 mm internal diameter roughened pipe
- 17,5 mm internal diameter smooth pipe

In Figure 4 the numbers represent the pipes from the list above in the same order.

a) b)

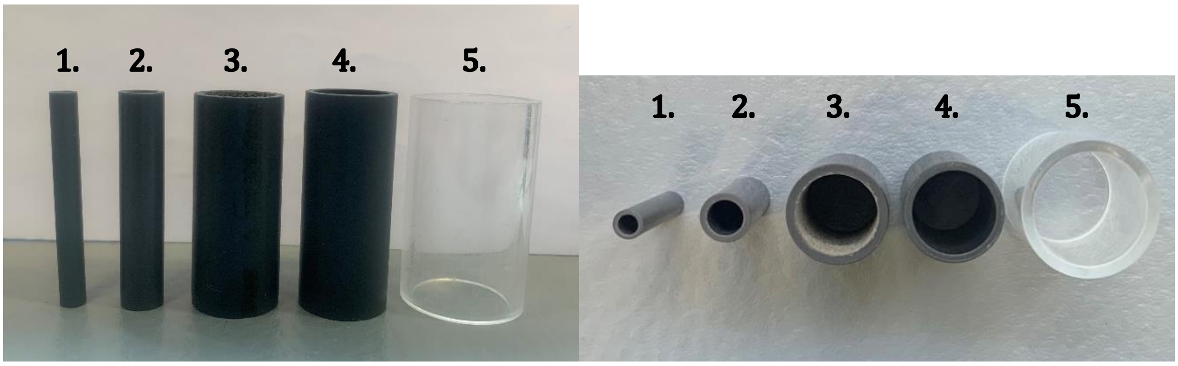


Figure 4: Types of pipes. a) Front view, b) Top view (Source: Author's own)

Within each line, there is a one-meter section that is actually measurable. Moreover, the system also contains other pipes, bends, valves and pipe flow metering devices to study the behaviour of the fluid. For instance, the acrylic pipe represented as number 5 in Figure 4 is a 24 mm pipe (internal diameter) that conducts the fluid through several of these valves.

All the components of the system have been identified in Figure 5.

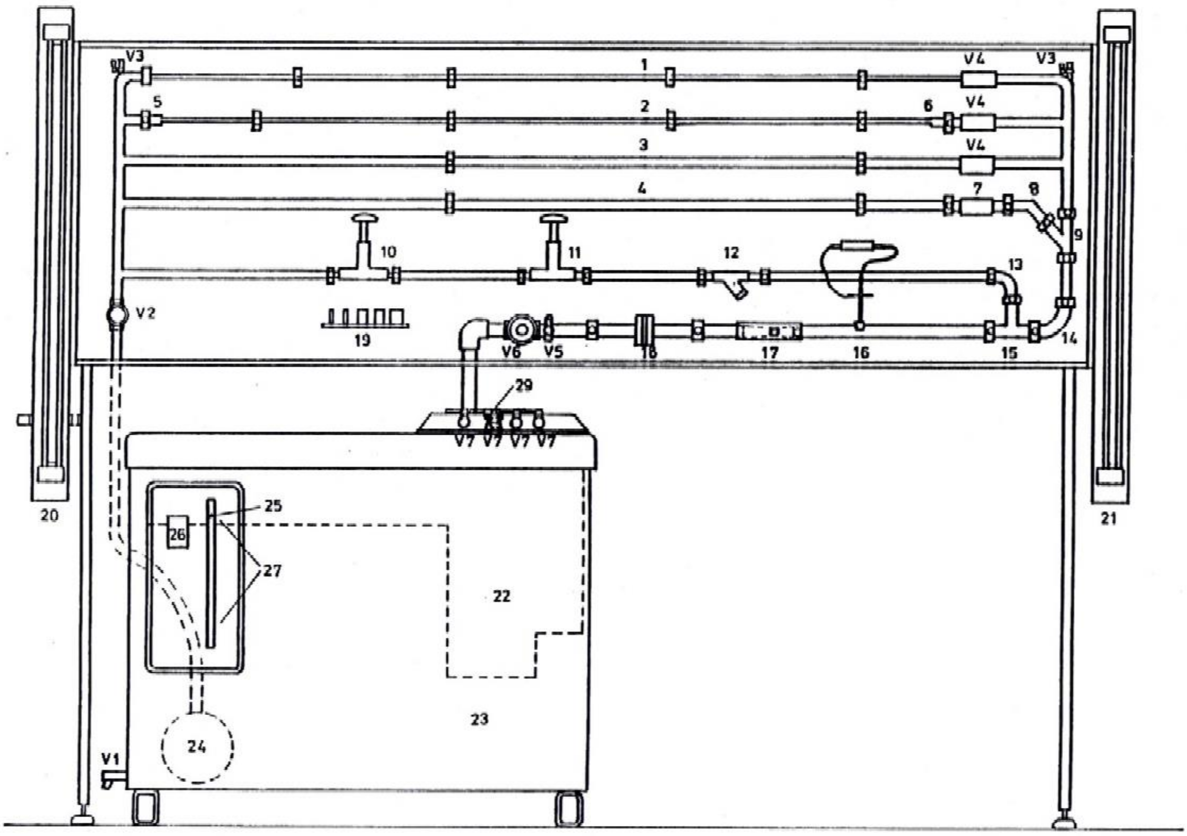


Figure 5: Armfield C6 Fluid Friction Apparatus Schematic (Source: https://armfield.co.uk/)

Table 1: Index to the schematic in Figure 3

1 - 6 mm smoothbore test pipe	16 - Pitotstatic tube	V1 - Sump tank drain valve
2 - 10 mm smoothbore test pipe	17 - Venturi meter	V2 - Inlet flow control valve
3 - Artificially roughened test pipe	18 - Orifice meter	V3 - Air bleed valves
4 - 17.5 mm smoothbore test pipe	19 - Test pipe samples	V4 - Isolating valves
5 - Sudden contraction	20 - 1 m mercury manometer	V5 - Outlet flow control valve (fine)
6 - Sudden enlargement	21 - 1 m pressurised water manometer	V6 - Outlet flow control valve (coarse)
7 - Ball valve	22 - Volumetric measuring tank	V7 - Manometer valves
8 - 45° elbow	23 - Sump tank	
9 - 45° "y" junction	24 - Service pump	
10 - Gate valve	25 - Sight tube	
11 - Globe valve	26 - Pump start /stop	
12 - In-line strainer	27 - Sight gauge securing screws	
13 - 90° elbow	28 - Measuring cylinder	
14 - 90° bend	29 - Dump valve	
15 - 90° "t" junction		

In order to be able to identify, quantify and study the fluid's behaviour, there are several measuring valves thought the pipes. These valves can be connected to the system manometer through hoses. The system has two different manometers, a U-tube type where mercury is the reference fluid and water is the transfer fluid, and one inverted U-tube with water (transfer fluid) and air (reference fluid). Both manometers have a scale that helps measure the height difference. The functioning of the U-tube manometers along with the formula to calculate the pressure differences (equation 9) have been detailed in late section 2.1.

The apparatus works as a circular system. The water is pumped into the pipes, goes through the described path (where the friction head losses can be measured with the manometers), and returns to the tank, where it will be pumped again.

The pump also permits modifying the flow rate, which allows the user to experiment with different fluid speeds, different Reynolds numbers, and both laminar and turbulent flow. Overall, this equipment allows for a wide range of presentations, measurements, and workouts and provides for a broad range of laboratory practice at the university.

3 Practical framework

The Armfield Fluid Friction apparatus update requires new software and hardware implementations. This section intends to introduce the new components by briefly explaining the technical and theoretical aspects.

Section 3.1 is an overview of the added hardware, Section 3.2 explains the Modbus protocol concepts, and a brief introduction to LabVIEW can be found in Section 3.3.

3.1 Equipment

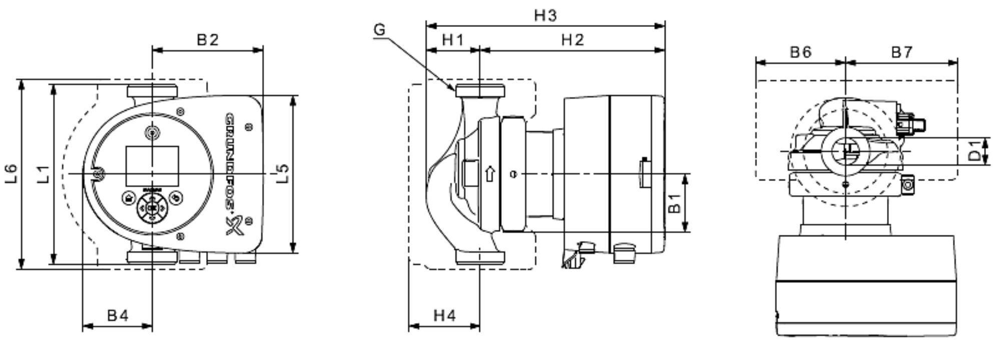
The Armfield C6 Fluid Friction Apparatus in the Energy Technology Laboratory has suffered some modifications since the original version. However, any of the modifications previous to this project have approached the system to a more automated state.

The new implementations to the system consist mainly of two components: a new pump and pressure sensors. This section aims to specify the new equipment required.

3.1.1 Pump (MAGNA3 Model D)

The new pump installed is a Grundfos MAGNA 3 model D circulator pump. This model has been designed to operate in systems that require an automated adjustment of pressure. However, the several operation and control modes available make the pump appropriate for a large number of applications. Some examples are heating systems, air-conditioning, and cooling systems (Grundfos, 2021b).

Specifically, the model of the pump is *MAGNA3 25-80 180* which means that the nominal diameter (DN) size of the inlet and outlet port is 25 mm, the maximum head is 80 dm, and the port-to-port length measures 180 mm. Figure 6 shows a schematic of the pump.



Pump type	Dimensions [mm]													[inch]
	L1	L5	L6	B1	B2	B4	B6	B7	H1	H2	H3	H4	D1	G
MAGNA3 25-80 (N)	180	158	190	58	111	69	90	113	54	185	239	71	25	1 1/2

Figure 6: Dimension MAGNA3 Model D single-head pump (Source: https://product-selection.grundfos.com)

This model is a highly automated pump that will allow precise control of the system. Besides its functionalities, the pump has been chosen for its small power consumption and little noise emissions (factor to consider as other students are doing class while the laboratory exercises are conducted).

More information can be found in the manuals "MAGNA 3. Installation and operating instructions" (Grundfos, 2021b) and "MAGNA 3. Model D. Circulator pumps" (Grundfos, 2021c).

3.1.2 CIM500 Ethernet module

The CIM 500 Ethernet module enables Grundfos products to establish communication to an Industrial Ethernet network (Grundfos, 2021a). The six industrial ethernet protocols supported are PROFINET, Modbus TCP, BACnet/IP, Ethernet/IP, GRM IP and Grundfos iSolutions Cloud (GiC).

Figure 7 shows the top part of the CIM500 Ethernet module.

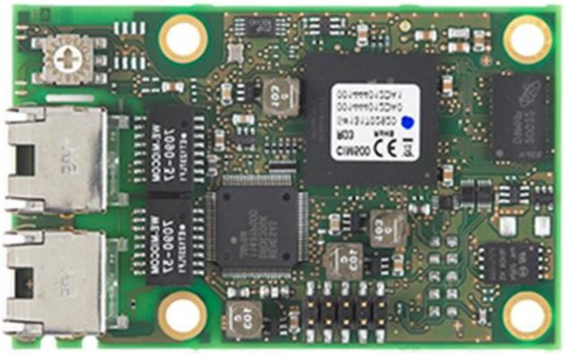


Figure 7: CIM500 Ethernet module (Source: https://product-selection.grundfos.com)

An ethernet module is essential to establish communication with the pump and control it. However, the pump selected (see Section 3.1.1) does not include the module as a build-in component. The assembly and configuration of the module are explained in Section 4.1.1.

The manual "CIM 500 Ethernet module. Installation and operating instructions" from Grundfos (2021a) contains more information.

3.1.3 Pressure sensor (ABPDRRV015PDAA5)

The pressure sensors selected to implement in the apparatus are the model ABPDRRV015PDAA5 from Honeywell. The model's name stands for ABP Series Amplified Basic Pressure Sensor, DIP package, RR pressure port (dual radial, barbed ports, same side), liquid media, no diagnostics, ±15 psi gage pressure range, analog output type, 10% to 90%

of Vsupply (analog), transfer function, no temperature output, no sleep mode, 5 Vdc supply voltage.

A schematic representation can be found in Figure 8. The pin connections are specified in Table 2.

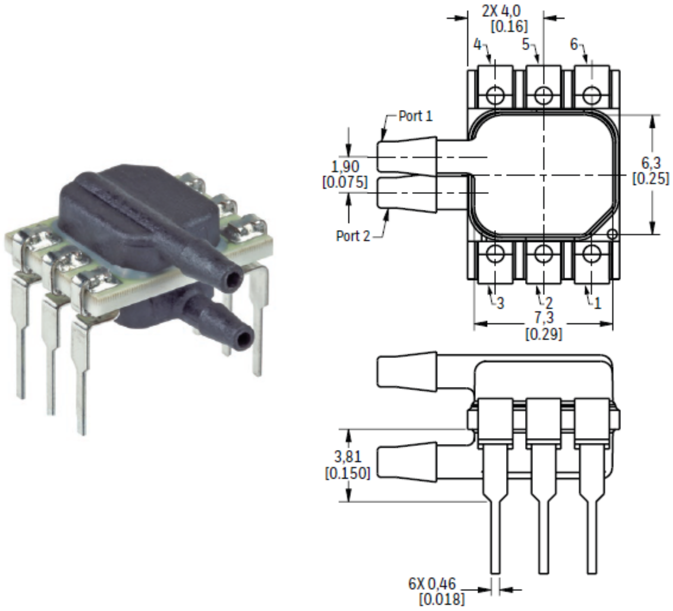


Figure 8: Pressure Sensor Schematic (Source: https://www.mouser.fi)

Table 2: Pressure sensor pinouts (Source: https://www.mouser.fi)

Output type	PIN 1	PIN 2	PIN 3	PIN 4	PIN 5	PIN 6
analog	GND	-	Vout	-	-	Vsupply

The pressure sensor, which is a piezoresistive silicon sensor that measures the differential pressure between the two ports, will replace the current manometers (see Section 2.2 for more information about the current system components). The installation has been described in Section 4.1.2. More information and technical specifications can be found in the datasheet "ABP SERIES. Basic Board Mount Pressure Sensors" (Honeywell, 2021).

3.1.4 NI USB-6008 device

The NI USB-6008 is a DAQ device from National Instruments whose main functionality is to create a bridge between LabVIEW and other components. The device provides eight single-ended analog inputs, two analog outputs, 12 DIO channels, and a 32-bit counter (National Instruments, 2015).

Figure 9: NI USB-6008 device (Source: https://www.ni.com)

The NI USB-6008 (Figure 9) will be used to connect the pressure sensors to the computer and process the data with LabVIEW. The implementation can be seen in Section 4.1.2. Technical information can be found in the user guide "USER GUIDE. NI USB-6008/6009 Bus-Powered Multifunction DAQ USB Device" by National Instruments (2015).

3.2 Modbus

Developed in 1979, Modbus is a serial communication protocol widely used in industrial automation. Modbus is an open protocol originally created by Modicon (nowadays Schneider Electric) to connect PLCs with other intelligent devices using the master-slave concept (Modbus Organization, 2022).

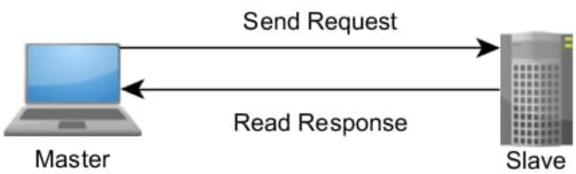


Figure 10: A Request - Response / Client - Server Model (Source: https://piembsystech.com)

Modus is a request-repose protocol, the communication always occurs in pairs, as illustrated in Figure 10. The master unit is responsible for initiating the interaction with a request. For every request sent, the slave device should give one response (NI, 2022; Modbus Organization, 2006).

Huitsing et al. (2008) state that the Modbus protocol has two principal variants, Modbus Serial and Modbus TCP. The Modbus serial protocol has two subcategories, ASCII and RTU, depending on the transmission mode between the master and the slave. The Modbus TCP protocol is based on an Ethernet connection (AUMA, 2020).

The communication of this project will be done via Modbus TCP. Modbus TCP protocol will be explained in greater detail in section 2.2.1. Section 2.2.2 is an overview of QModMaster, a simple program to communicate with slaves. The technical aspects related to the practical application of Modbus TCP protocol in the specific case of this thesis will be explained in section 2.3.3.

3.2.1 Modbus TCP

Modbus TCP is a variation of the Modbus protocol that includes both LAN-based Modbus networks (a master and its slaves) and IP-interconnected Modbus networks (multiples masters), according to AUMA (2020).

Instead of master/slave, Modbus TCP utilises the terminology client/server. In a TCP network, a client is linked to a switch or series of switches to which all the servers are also connected. Modbus TCP devices use a unique IP (Internet Protocol) address and a subnet mask, both represented by 8-bit numerical groups or octets (ProSoft Technology, 2014; Swales and Electric, 1999). The IP address is the location of a particular device on a network, while the subnet masks serve to simplify routing traffic within the network.

Figure 11 shows a possible Modbus TCP architecture. Note that there are 2 clients (or masters) in this example. Both Master A and Master B can poll with all the slaves.

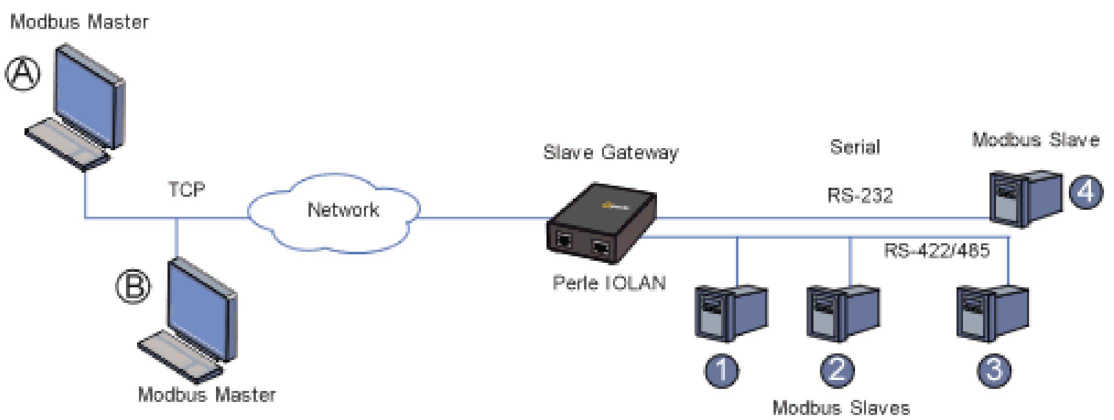


Figure 11: Modbus TCP architecture (Source: https://www.perlesystems.fr)

Another fundamental principle of Modbus TCP is that any device on a Modbus TCP/IP network can send a command, hence all the devices can act as a master. In most cases, however, only one device acts as a client.

The transactions are similar to the Modbus Serial protocol. The Modbus application protocol data unit (PDU) is included in the messages exchanged, but the Modbus TCP also includes the Modbus application protocol (MBAP) (Modbus Organization, 2006). The Modbus PDU has two fields, a one-byte function code and function parameters, while the

MBAP header has four fields: transaction identifier, protocol identifier, length, and unit identifier (Huitsing et al., 2008).

Figure 12 represents the Modbus TCP message building schematically.

MBAP Header				PDU	
Transaction ID	Protocol ID	Length	Unit ID	Function code	Data
(2 Bytes)	(2 Bytes)	(2 Bytes)	(1 Byte)	(1 Byte)	(Varies)

Figure 12: Construction of a Modbus TCP message (Adapted from: https://manishvishwakarma.wordpress.com)

Modbus Organization (2006) explains the functionality of the different fields as follows:

- Transaction ID: Message identifier used so that the master can identify the different responses (only useful when multiple requests are sent without waiting for the individual responses).
- Protocol ID: Always 0 for Modbus. It is a field reserved for possible future extensions.
- Length: Bite count for the rest of the message.
- Unit ID: Not used in Modbus TCP. Used for compatibility with serial bridging (the equivalent of slave address in Modbus RTU).
- Function code: Identifier of the function used. More information on the Modbus functions can be found in Section 2.3.3.
- Data: Information transmitted between devices. The length and content vary on the function used (maximum 252 bytes).

3.2.2 QModMaster

QModMaster is an open-source program used as a simple way to test Modbus communications (compatible with TCP and Serial, both RTU and ASCII, communication). The main functionality of the program is to turn the computer into a Master and use the serial port (Serial) or the Ethernet port (TCP) to communicate with a slave (zhanglongqi, 2017; Philbert, 2020). Note that an adaptor is required to connect the slaves to the serial/ethernet port.

The software must be configured in order to communicate with a slave via TCP. The *Modbus Mode* is selected as TCP; and the *Function Code* to execute, *Starting Address,* and *Number of Registers* to write/read are specified according to the user's needs. Note that the unit ID is not necessary for Modbus TCP.

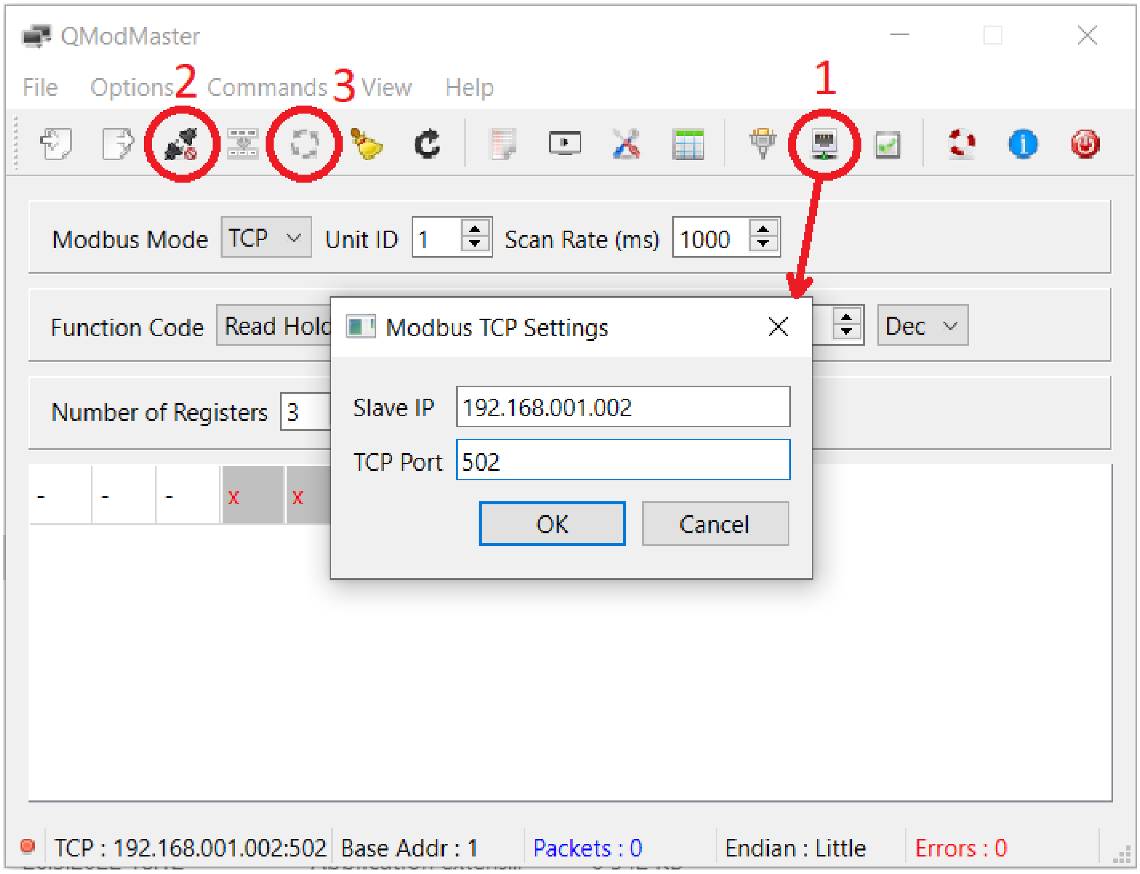


Figure 13: QModMaster steps to set up and use TCP connection (Source: Author's own)

The steps to set up and connect with the slave are indicated in Figure 13:

1. Click the Modbus TCP button to establish the slave IP. Note that the TCP Port is always 502.

2. Click the connect button to establish communication. An error appears if communication fails.

3. Click scan to start continuous scanning.

It will not be further specified in the documentation of this thesis, but QModMaster has been used through the project to check and verify the data obtained with other programs.

3.2.3 Modbus in this project

The slaves (or servers) store the information in registers. In Modbus devices, registers can have 4 different types: coil (Discrete output), discrete input (Status input), input register and holding register (Control Solutions, 2020).

To access the registers, the Modbus protocol defines 21 functions (Table 3). When communicating with a register, the function code needs to be specified together with the register's address (see Section 3.2.1 for more information).

Table 3: Modbus function codes (Adapted from: https://www.modbus.org)

Code	Function
0x01	Read coils
0x02	Read Discrete Inputs
0x03	Read Holding Registers
0x04	Read Input Registers
0x05	Write Single Coil
0x06	Write Single Register
0x07	Read Exception status
0x08	Diagnosis
0x0B	Get Comm Event Counter
0x0C	Get Comm Event Log
0x0F	Write multiple coils
0x10	Write multiple registers
0x11	Report Server ID
0x14	Read file record
0x15	Write file record
0x16	Mask write register
0x17	Read/Write multiple registers
0x18	Read FIFO Queue
0x2B	Encapsulated interface transport

The CIM 500 Ethernet module, used to establish Modbus TCP communication with the pump only supports functions 0x03, 0x04, 0x06, 0x08, and 0x10. Reading or writing coils are not supported. Instead, holding registers or input registers should be accessed (both contain the same data). The diagnosis subcodes are supported and the interpretation guide can be found in "Modbus for Grundfos pumps. Functional profile and user manual" (Grundfos, 2019).

3.3 LabVIEW

LabVIEW is a programming environment centred on test instrumentation. Developed in 1986 by National Instruments (commonly referred to only by NI), LabVIEW stands for Laboratory Virtual Instrument Engineering Workbench (electronics notes, 2022). LabVIEW is available from https://www.ni.com/fi-fi/shop/labview.htmlAn annual subscription license to the software ranges from 480€ to 3200€

Nowadays, LabVIEW has become a widely used tool for engineers to develop automated research, validation, and production test systems (NI, 2019). LabVIEW has a graphical approach, offering not only a visual programming environment but also interface development. The user can visualize all the aspects including hardware configuration, measurement data, and debugging (Farnell, 2022).

This thesis has used LabVIEW 2019 to develop both the program and the interface of the new automated Armfield C6 fluid friction apparatus.

4 Methodology

This section is centred on the discussion and reflection on the different factors that compound the final result, the user interface.

Section 4.1 is a summary of the steps followed to assemble the hardware as well as the configuration settings chosen for the components. An overview of the LabVIEW code developed, and the description of the functions implemented can be found in Section 4.2. Finally, Section 4.3 shows and itemizes the user interface designed.

4.1 Hardware installation and configuration

This project is centred on the development of a LabVIEW program to automatize the system. However, in order to communicate with the apparatus, some components need to be changed while others are introduced. The hardware components and their specifications have been explained in Section 3.1.

This section is a small summary of the installation and configuration of the components. The aim is to give the reader the knowledge necessary to fix possible failures or duplicate the work. Section 4.1.1 details the set-up of the pump and assembly of the Ethernet module and alimentation cable. The information related to the installation of the sensors in the apparatus can be found in Section 4.1.2.

4.1.1 Assembly of the MAGNA3 pump and CIM500 Ethernet module

Before the installation of the pump in the system is necessary to assemble the power source and install the Ethernet module to allow communication.

The power cord has been adapted from a power cord already made to simplify the task. In this particular case, a computer cable has been used, replacing the 3-pin connector with the pump adapter. To do this, the cable has been cut just below the connector and the three wires have been connected to the adapter as shown in Figure 4 a), which has later been closed, and connected to the pump (Figure 4 b)).

a) b)

Figure 14: Pump power cord. a) Wire connections, b) Final result (Source: Author's own)

To install the ethernet module, the front part of the pump needs to be opened with a Torx screwdriver (T15H) as shown in Figure 15 a). The ethernet module needs to be carefully connected to the 10-pin connector and screwed to the pump (see Figure 15 b)).

a) b)

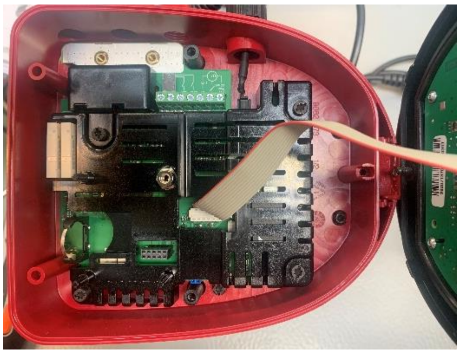


Figure 15: Magna 3 interior. a) Initial state, b) Ethernet module installed (Source: Author's own)

The last step is to connect the Ethernet cable that will establish the communication. To do so, one of the side holes of the pump needs to be opened. The ethernet cable is passed through and connected to the first port of the Ethernet module. The last step is to connect the cable shield to the ground through the earth clamp. The final result can be seen in Figure 16. More information about the installation can be obtained from the instruction manuals (see Appendix 8.1).

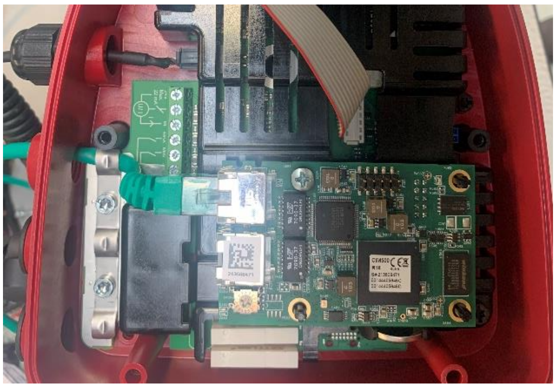


Figure 16: Final assembly of the Ethernet module in the pump (Source: Author's own)

Once the pump is completely assembled, it needs to be configured to communicate via Modbus TCP. In the first place, the rotary switch of the Ethernet module is positioned to 1 (Figure 17) to choose Modbus TCP as the Industrial Ethernet protocol. Information on the other possible configurations can be found in the manual "CIM 500 Ethernet module. Installation and operating instructions" (Grundfos, 2021a).

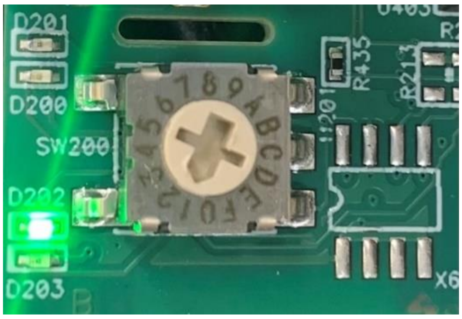


Figure 17: Selecting Modbus TCP as Industrial Ethernet Protocol CIM500 (Source: Author's own)

The CIM 500 ethernet module is by default set to a fixed IP address, which can be changed from the built-in webserver (this step is mandatory in the first connection). To establish a connection from a PC to CIM 500, the computer must have the same subnetwork as the Ethernet module (and the subnet mask 255.255.255.0) and be connected using the ethernet cable. To change the IP in the webserver, the following steps are required:

- Open an internet browser.
- Type the IP "192.168.1.100" in the URL field (the IP to the webserver does not change even if the Ethernet module IP is changed)
- Log in to the webserver (in this case, User: admin, Password: Novia-2022)
- Go to the left menu > Configuration > Real-Time Ethernet Protocol

The IP of the Ethernet module is 192.168.1.2. The full Ethernet Protocol configuration is shown in Figure 18. The other configuration parameters have been left in the default value.

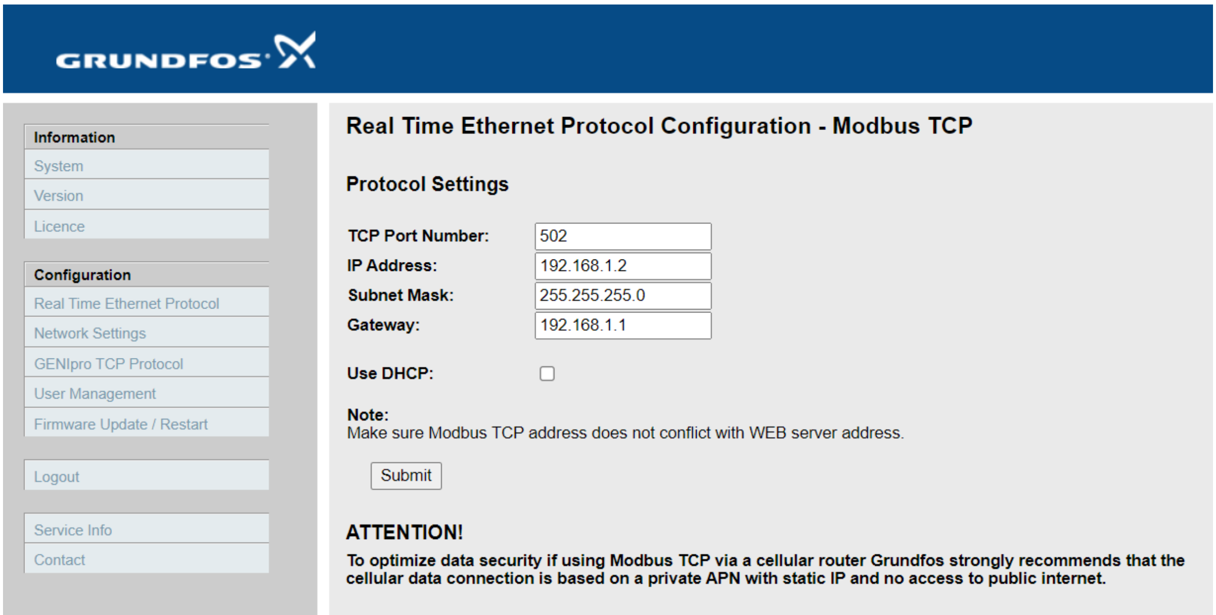


Figure 18: Real-Time Ethernet Protocol Configuration CIM500 (Source: Grundfos webserver)

When all this configuration has been done, the pump registers can be read with Modbus TCP, but not written. To enable the writing rights, the Access Mode (stored in register 00201 bit 8) should be changed from 0-Local to 1-Remote. This change must be done by writing the register 00101 bit 0 (for example with LabVIEW).

More information on the Modbus TCP communication configuration and implementation on this project can be found in Section 3.2.3 and Section 4.2.1.

4.1.2 Installation of the sensors

The sensor installation can be divided into two parts: the connection from the sensor to the computer and the connection from the sensor to the Armfield C6 Fluid Friction Apparatus.

To establish the connection with the computer, a protoboard and a NI USB-6008 device (see Section 3.1.4) are needed. The position and output of the sensor's pins are shown in Figure 8 and Table 2 in Section 3.1.3.

Figure 19 illustrates the implementation of the sensor in the protoboard.

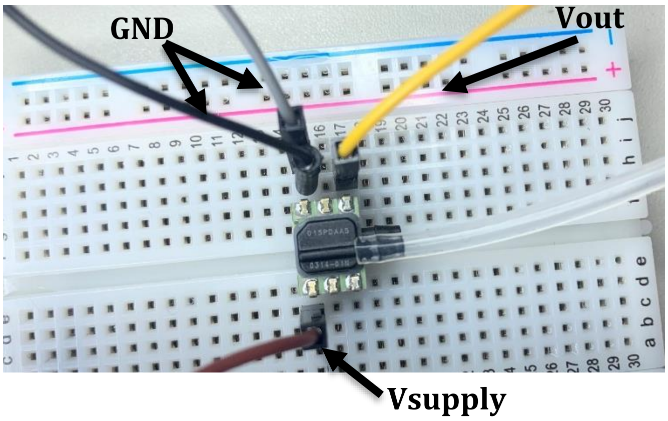


Figure 19: Pressure sensor protoboard connection (Source: Author's own)

The wires are connected to the USB-6008 device following the input specification in the user guide “USER GUIDE. NI USB-6008/6009 Bus-Powered Multifunction DAQ USB Device” by National Instruments (2015). The connections are described in Table 4.

Table 4: Summary of the pressure sensor to USB-6008 device connections (Source: Author's own)

Name	Sensor pin	NI USB-6008 port
Vout	3	2
Vsupply	6	31
Ground	1	3 & 32

Finally, the USB-6008 device is connected to the computer via cable USB, the option “LabVIEW-64bits” is selected in the pop-up window and the data is ready to be accessed. Note that a DAQ is necessary to read the data. In this case, ports 2 and 3 are the AI 0 (analog input 0) and the data received is a voltage. More information about the code used to read the pressure sensor can be found in Section 4.2.7.

The connection between the sensor and the apparatus will be done with the same kind of system uses by the current manometers. The elements required are a hose with an adapter to the system pipes (Figure 20 a) and a valvule with an adapter to the sensor (Figure 20 b).

a) b)

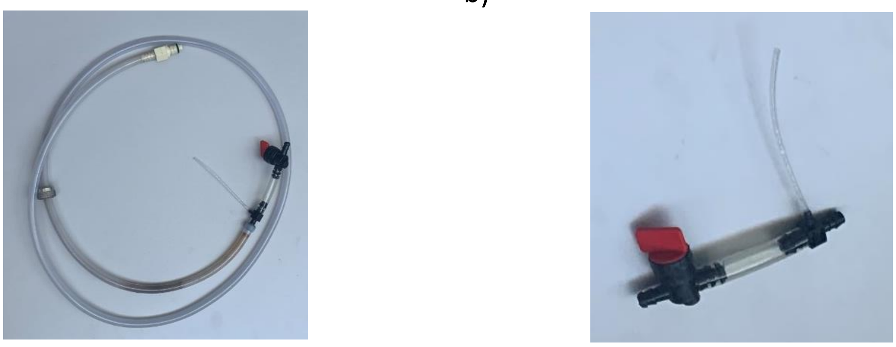


Figure 20: a) Hose with adaptors, b) Sensor adaptor valvule (Source: Author's own)

The valvule showed in Figure 20 b) is used to remove the air from the hose if necessary. The small tube is connected to the sensor port. The other adaptor that can be seen in Figure 20 a) can be connected to the system as shown in Figure 21.

Figure 21: Hose connection to the Armfield Fluid Friction Apparatus (Source: Author's own)

The hoses can be changed from one measurement point to another, so with two hoses (and adaptors) and one pressure sensor, the differential pressure between all the points can be measured. However, to avoid doing so many changes, three pressure sensors are going to be installed.

4.2 LabVIEW program development

The automatization of a system can be done in several ways. For this thesis, the desired result is to obtain the data from the pressure sensors and communicate with the pump. The implementation, which has been done with LabVIEW, is the discussion topic of this section.

The LabVIEW program developed has three big parts: communication with the pump, acquisition and treatment of the sensor data, and interface building. The program implementation has been itemized and respectively discussed in Sections 4.2.1 to 4.2.8.

The main program's VI can be seen in Appendix 8.5 and the whole project has been saved on the internal Novia MS OneDrive repository. The interface functionality and design have been explained in Section 4.3.

4.2.1 Modbus TCP communication

The communication with the pump is done via Modbus TCP as explained in Section 3.3 and Section 4.1.1. This function (Figure 22) initializes the master on the computer and prepares the communication.

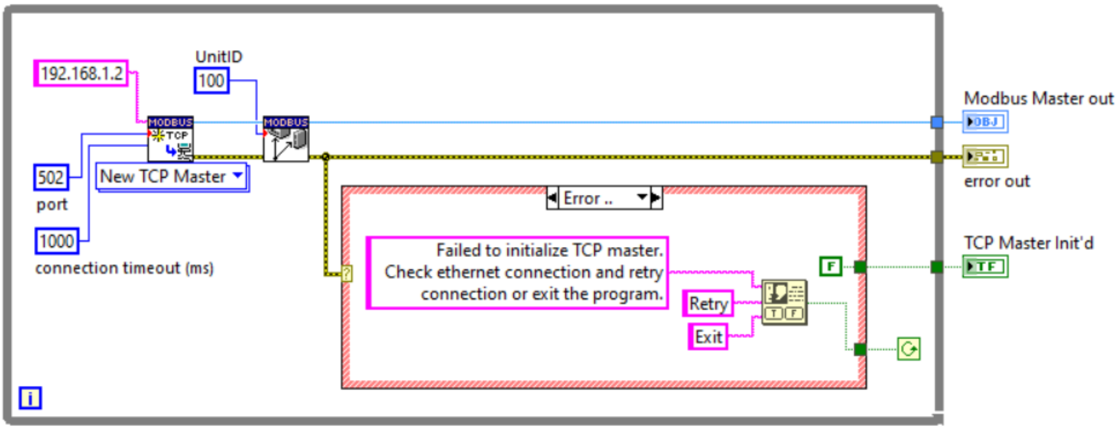


Figure 22: TCP function. TCPmasterInit (Source: Author's own)

The IP used is the pump´s IP (details on the pump configuration can be found in Appendix 8.2). On the other hand, the unit ID is usually not necessary for Modbus TCP communications, but it was included for future serial communication possibilities. A detailed explanation of the blocks and the configuration used can be found in the thesis written by Cànovas Trujillo and Francés Vigatà (2021). Note that these blocks form the "MBAP Header" explained in Section 3.2.

In the case where the master cannot be initialized, an error will pop up and the rest of the program will not be executed. The error handling will be detailed in Section 4.2.8.

4.2.2 Read pump stats

The pump stores the data about the stats in several registers, which can be accessed with the CIM 5000 Ethernet module. Some of the registers are read-only, while others can be modified and allow the control of the pump. The information, description, and address of all the registers can be found in the user guide "Modbus for Grundfos pumps. Functional profile and user manual" (Grundfos, 2019).

Before reading the registers with LabVIEW, the information needed had to be decided. An Excel sheet with all the registers was created to make the choice more comfortable and clearer (a more evolved version of this Excel sheet can be found in Appendix 8.3). Once the registers have been selected, the code can be implemented.

A close look at the code used can be seen in Figure 23. The LabVIEW block necessary to read is also explained in the thesis written by Cànovas Trujillo and Francés Vigatà (2021). Note that the address in the LabVIEW function is displaced one position from the one specified in the manual "Modbus for Grundfos pumps. Functional profile and user manual" (Grundfos, 2019). For more clarity, the address listed in the manual is introduced followed by a subtraction before inputting the value into the "Read holding register" function.

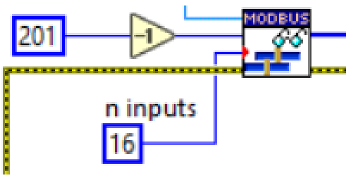


Figure 23: LabVIEW code. Read register via Modbus TCP (Source: Author's own)

Considering that starting individual register readings is slower than reading multiple registers at once, the usage of the “Read holding register” function has been avoided to the extent possible. For this reason, a balance between starting a new lecture and reading only the registers chosen was searched.

After the “Read holding register” function all the information received is concatenated in an array and finally, the non-desired registers that were read are removed.

This process described has been implemented in a subVI called “Read” (see the relationship between icons and names in Appendix 8.4). Figure 24 is an overview of the code.

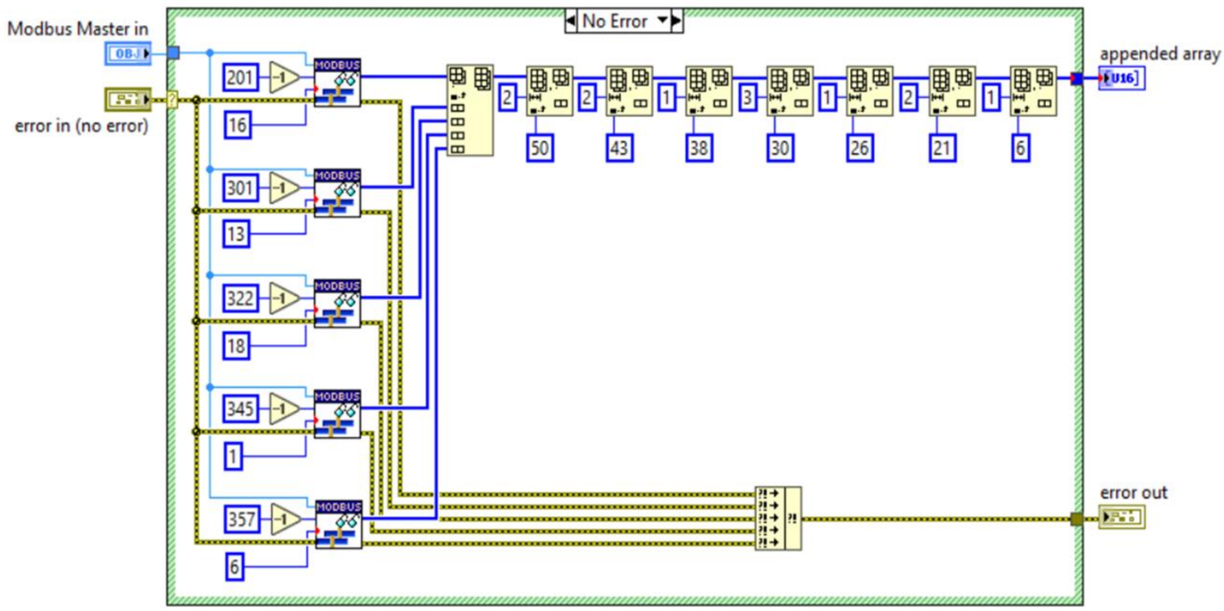


Figure 24: LabVIEW function. Read. (Source: Author's own)

The “Read” function implemented (Figure 24) has two inputs (Modbus Master in and the error) and returns the error code and an array with the data read. The error handling will be detailed in Section 4.2.8.

4.2.3 Process register data

The information stored in the registers is 16-bit binary unsigned integers. However, to show the values in a meaningful way in the interface, conversions need to be done. A cluster has been defined so the data types can be different, but the data can still be grouped in only one structure. The data type decided for each value along with the index in the cluster were registered in an Excel sheet and can be found in Appendix 8.3.

A subVI called "ProcessData" has been created to process and adapt the data stored in the registers. "ProcessData" is a large and complex function. To increase the clarity of this section, the code has been itemized in three parts. These are shown in Figure 25, Figure 26, and Figure 27, while the complete subVI can be found in Appendix 8.6.

"ProcessData" receives an array with all the data read so each element corresponds to the raw value of a register (in order). Figure 25 shows the treatment of the input array.

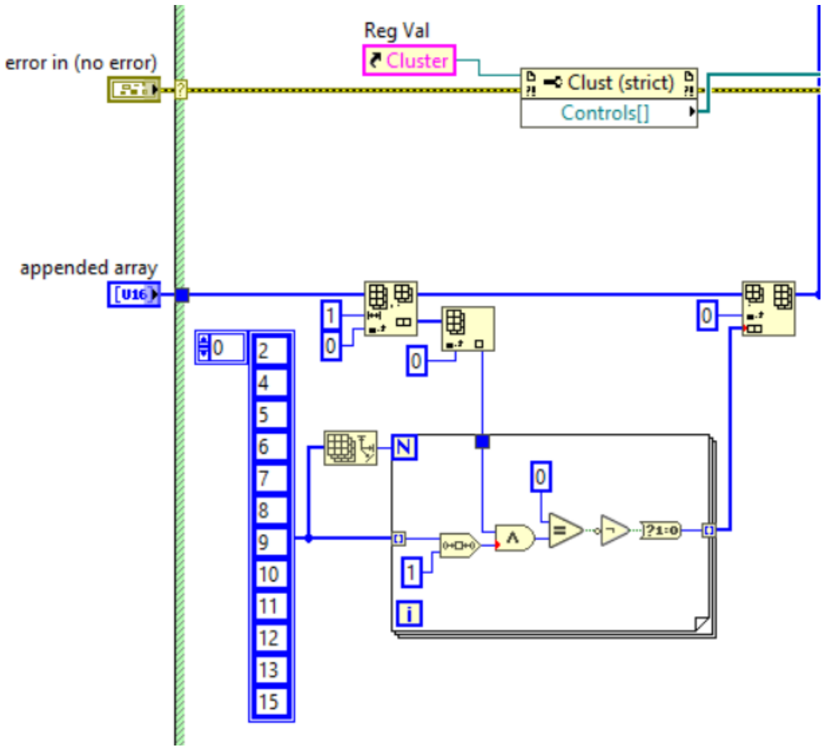


Figure 25: LabVIEW code. Bit separation and generation of a reference array of the cluster's controls (Source: Author's own)

The register 201, sorted in position 0, stores a different variable in each bit. The first step is to isolate the register 201, separate the desired bits, and append the values at the beginning of the array. The register values will be "copied" into the cluster "Reg Val" which is used as a temporary data storage in this subVI. To be able to transfer the data, an array with the references of all the controls in the "Reg Val" cluster is created.

Figure 26 shows the process to change the data type of the values.

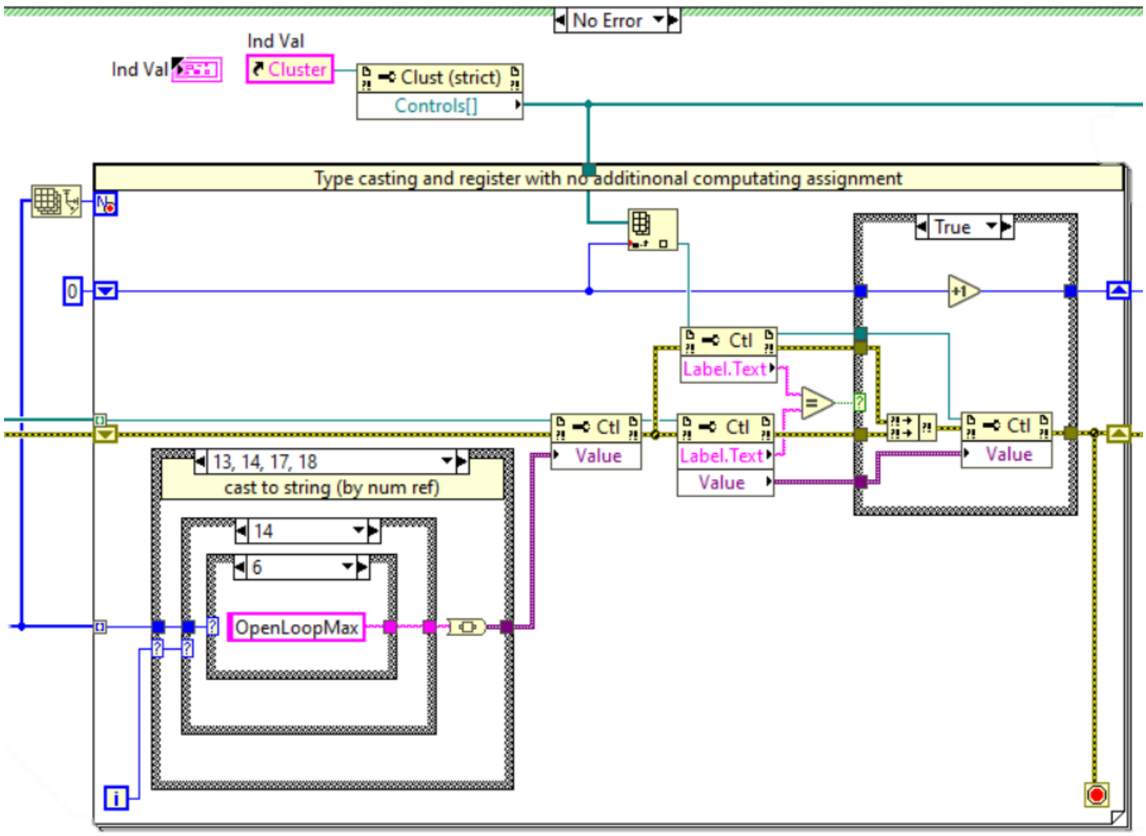


Figure 26: LabVIEW code. Type casting and register with no additional computation assignment (Source: Author's own)

One by one, all the array values are converted into the corresponding data type and inserted into the cluster. The data type being cast is known by the index of the value. Since all the data types have been previously decided, the program only needs to check the value´s index.

To do the type casting, a conditional structure, where each case corresponds to a data type casting, is used. The casts to Boolean and integer types are direct. The cast to double takes into account the precision of the least significant bit of the data. The cast to a string requires some additional logic since all the string values corresponding to register values are unique

The cluster “Ind Val” is the final output of this subVI. This cluster will contain the data that is going to be shown in the main panel’s indicators. Some of these values correspond directly to values (after typecasting) of only one register while others are the result of merging data from two or more registers.

The indicators of “Ind Val” that correspond directly to one register’s value are assigned in the loop shown in Figure 26. To identify these values, the indicators’ labels from “Ind Val” are compared to the ones from “Reg Val”. If they match, the value from the current control of “Reg Val” is transferred to the selected indicator from “Ind Val”. This is only possible because the indicators of “Ind Val” are in the same order as the controls of “Reg Val” and the additional indicators of “Ind Val” whose value corresponds to a merged value are indexed as the last indicators of the cluster.

Figure 27 details the last part of the function, where the values of the indicators of “Ind Val” are assigned as a result of merging more than one value of the controls of “Reg Val”.

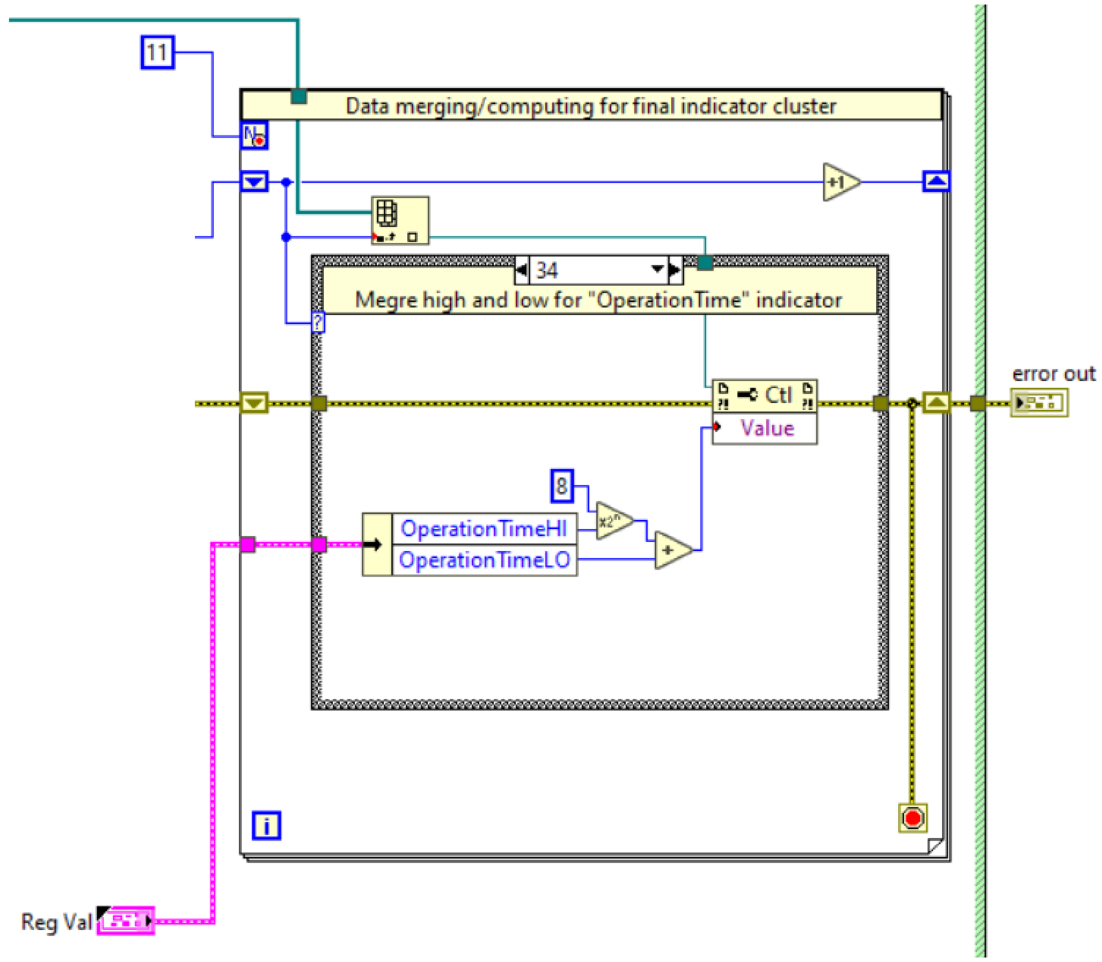


Figure 27: LabVIEW code. Data merging/computing for final indicator cluster (Source: Author's own)

There are eleven cases and each of them is treated individually with an if-case. Most of the cases require the mix of two registers into one value, like the one shown in Figure 27.

4.2.4 Update indicators

The function “UpdatePanel” is responsible for updating all the indicator values in the interface. This function has three inputs, the error code, the cluster “Ind Val” resulting from function “DataProcess” and an array of all the references. The array is a static variable that contains all the references of the indicators to be updated. The order of the references in the array is strictly the same as the values used for their update in the cluster. This way, all indicators that have no direct link to a value in the cluster are added at the end of the references array.

There are three possibilities depending on the relationship between the indicator and the cluster. The simplest and most common action is the direct assignation of the value stored in the cluster to the value of an indicator in the interface. The second option is the assignation of a value stored to another property of an indicator of the interface (e.g. the caption). The last possibility is that the indicator's value is related indirectly a one or several values of the cluster.

The function iterates for each indication in the reference array. The first part of the function (Figure 28) determines in which of the beforementioned cases the indicator belongs. If the indicator needs anything else than a direct assignation of its value (options 2 and 3), the process is done in each of the cases of the conditional structure.

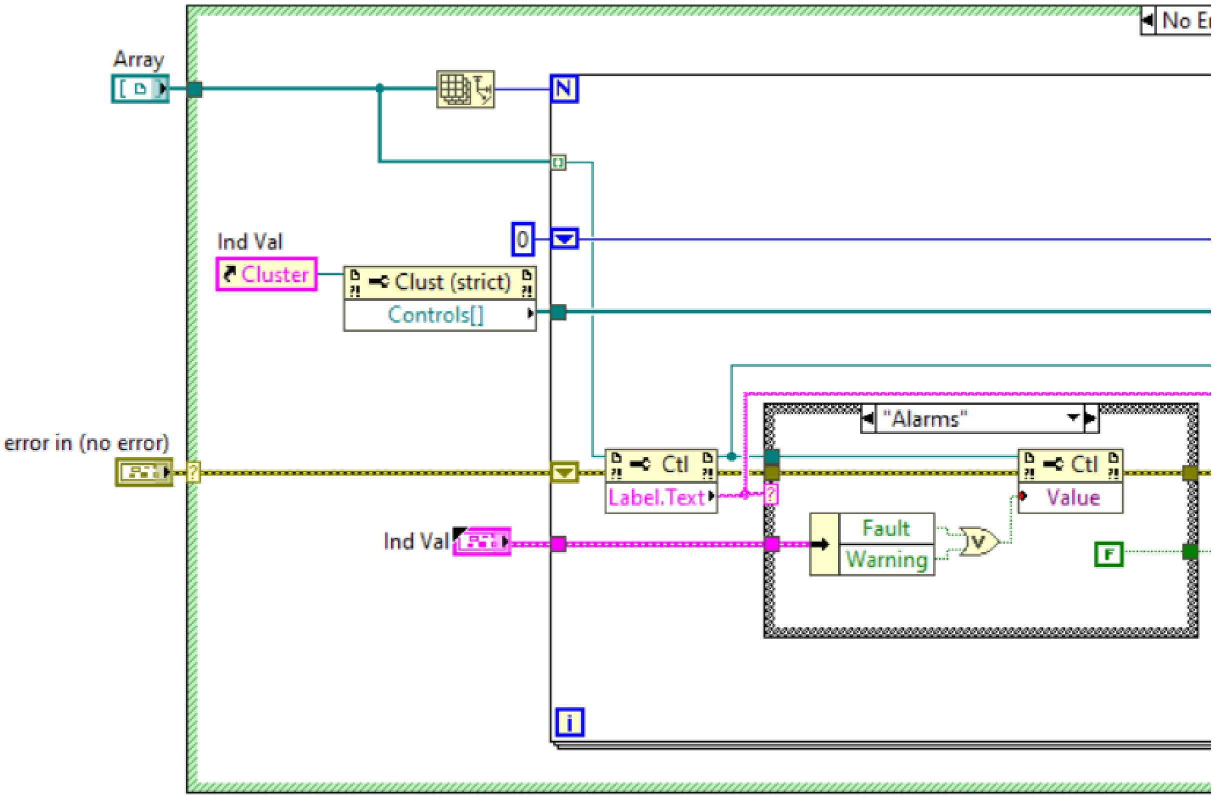


Figure 28: LabVIEW code. Indicator analysis (Source: Author's own)

The direct assignation of the value is done with the code shown in Figure 29. However, not all the indicators are assigned a direct value of the cluster. The indicators that depend indirectly on the cluster value (option 3) are not part of this assignation. To differentiate them, their if-cases have a “false” output, while the other has a “true” output.

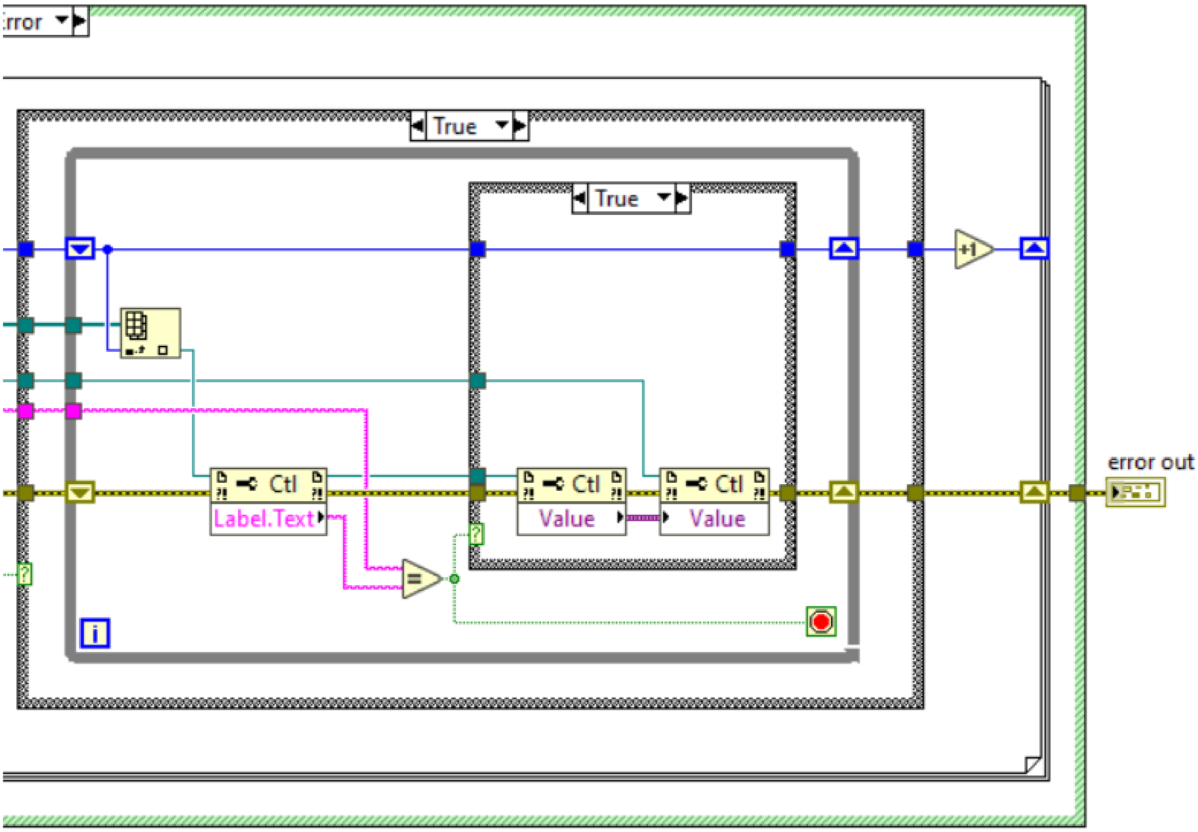


Figure 29: LabVIEW code. Indicators' value assignation (Source: Author's own)

The direct assignation of the values is done by comparing the labels of both the reference array and the cluster and transferring the value to one another if a match occurs. Otherwise, the cluster reference is skipped and the next one is compared to the same indicator label. This process continues until a match occurs.

4.2.5 Write pump registers

The program developed also supports the control of the pump. The Grundfos (2019) user guide "Modbus for Grundfos pumps. Functional profile and user manual" contains the information about the registers that can be written. Only 6 of the possible options will be editable for the user. These are the On/Off request (register 101.1), the enabled of the flow limit (register 101.5), the control mode (register 102), the operation mode (register 103), the setpoint (register 104), and the max flow limit value (register 106).

All the changes to the beforementioned registers are controlled from the interface. For this reason, an event handler has been created. The event handler only activates when the action previously specified is done. This approach helps optimize the usage the computer resources.

Figure 30 shows the "On/Off request" section of the event handler. The request is controlled by a boolean button in the interface, so the event will activate when the button value has changed.

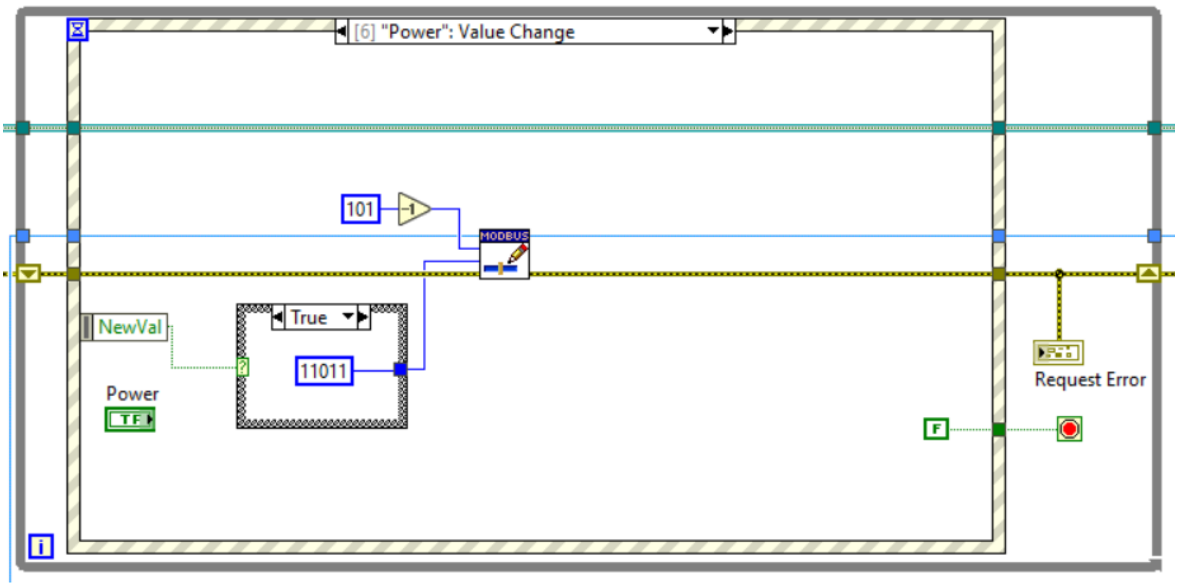


Figure 30: LabVIEW code. On/Off request (Source: Author's own)

The action done by the event described in Figure 30 is quite simple. If the new value of the button is 1, the "open" request is written. Otherwise (value 0), the off request is the one sent. The on/off request is only one bit of the register. However, all the other values are known (most of them can´t change in the context of this thesis), so the whole register is written, as the implementation is easier.

The other events follow a similar structure. The control mode and the operation mode are edited with a ring indicator, which allows the user to choose from a list of different options. The ring indicator is configured from the front panel and associates each option with a number. In these cases, the event handling is activated when the ring value changes and checks which option number has been chosen. Depending on the option, one value or another is written in the register. Figure 31 shows the writing code of the operation mode (Figure 31 a) and the control mode (Figure 31 b). Note that LabVIEW automatically converts a decimal number to binary when writing a register.

a)

b)

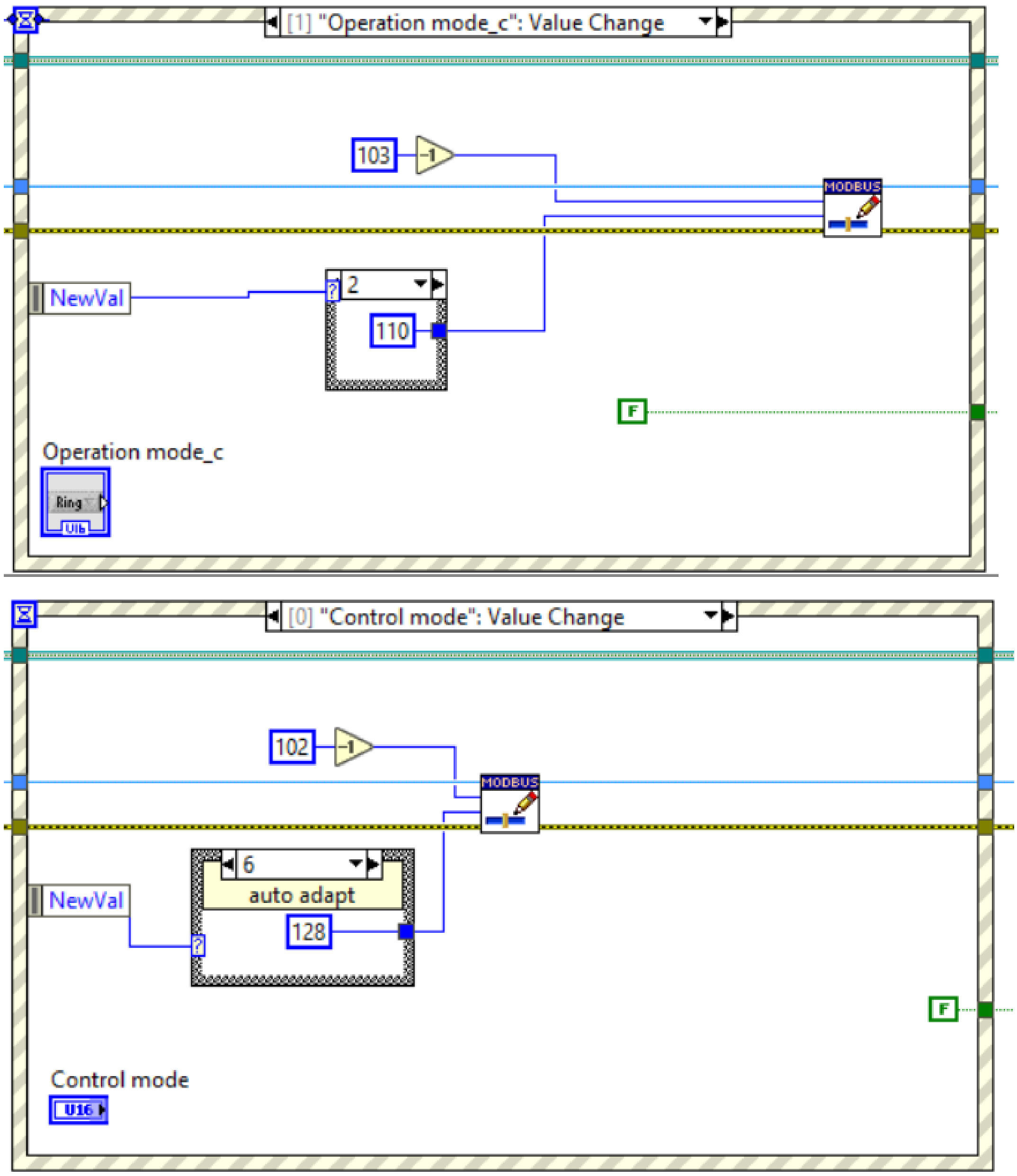


Figure 31: LabVIEW code. a) Operation Mode request, b) Control Mode request (Source: Author's own)

The definition of a new setpoint also follows the structure explained in the previous cases. However, the way the setpoint is represented in the register can be confusing. Aiming to make this functionality more logical for the students, the value requested in the interface is expressed differently from the one that must be written in the register. The value is currently represented as a percentage. Figure 32 shows the code implementation that allows modifying the pump setpoint.

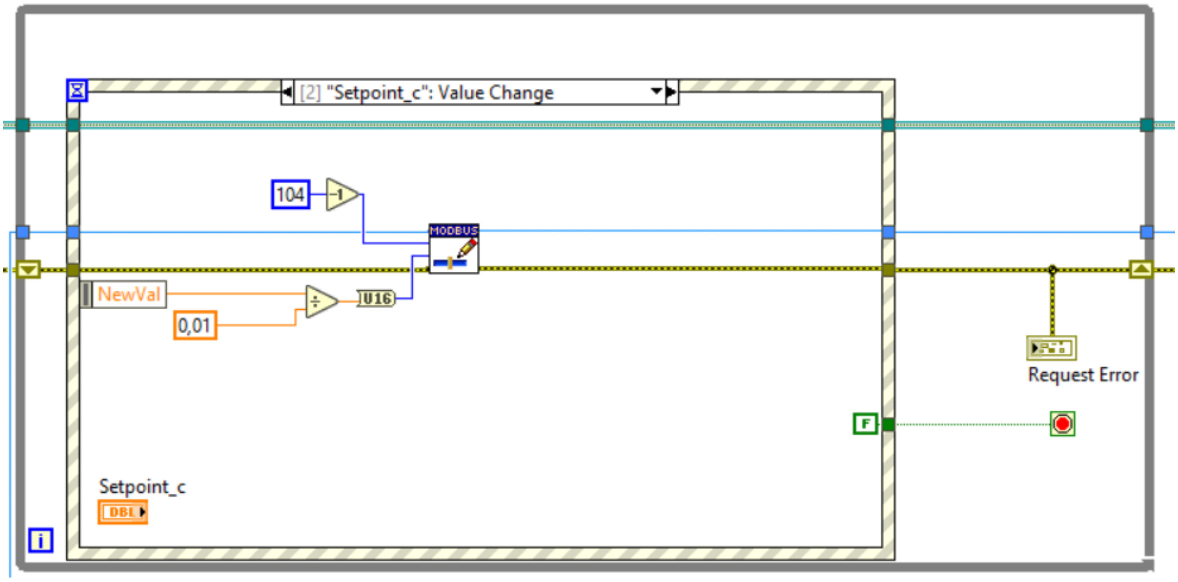


Figure 32: LabVIEW code. Setpoint request (Source: Author's own)

The last values modifiable by the user interface are related to the flow limit. “EnableMaxFlowLimit” is represented by another bit of register 101, the same as the “On/Off” request. The implementation has been the same as the described previously for the “On/Off” request, as shown in Figure 32. However, writing the whole register has inconveniences, as all the other bits will also be modified. In this case, when enabling the flow limit, the pump will automatically open.

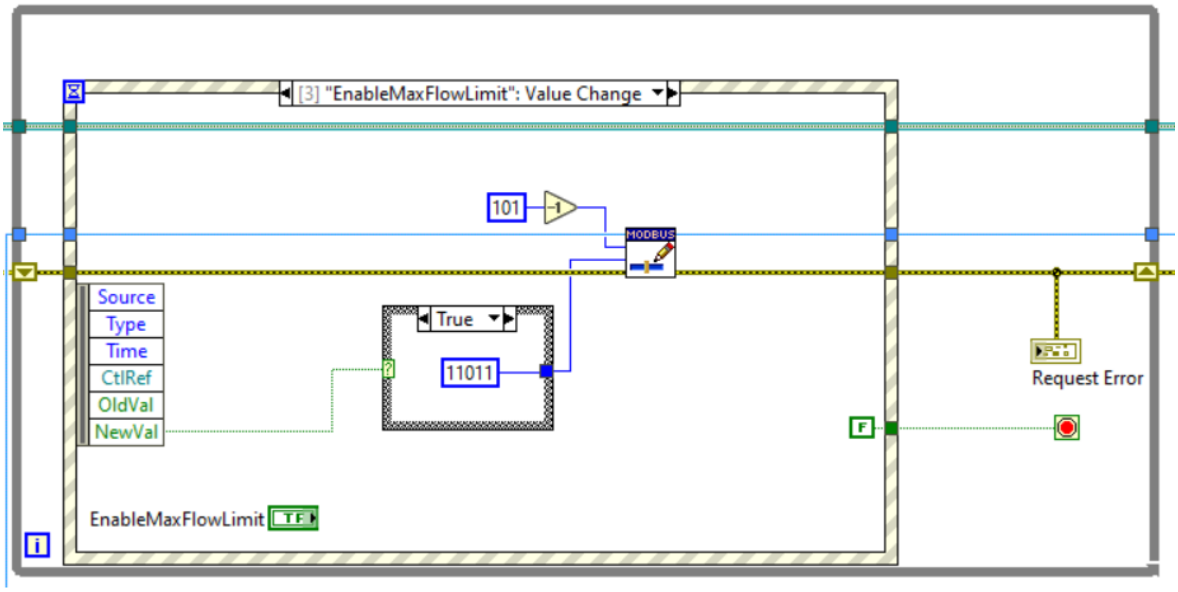


Figure 33: LabVIEW code. Enable max flow request (Source: Author's own)

If the limit to the flow has been enabled, the user has the option to adjust the value conveniently. “SetMaxFlowLimit” is stored in register 106 with a precision of 0,01 m^3/h. The implementation, shown in Figure 34, is similar to the setpoint, as the precision is the same.

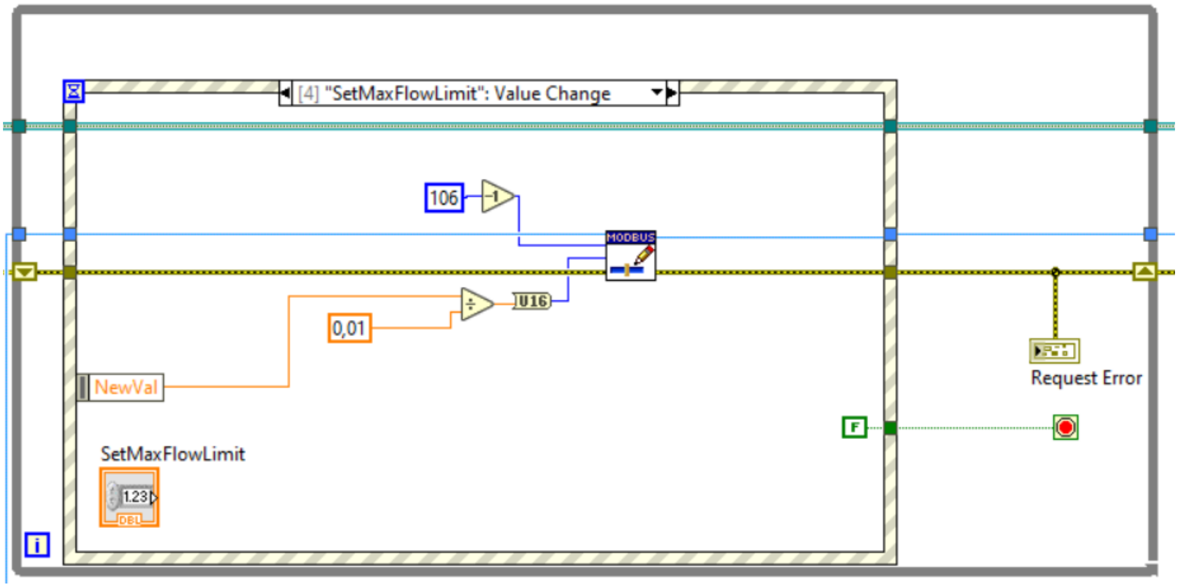


Figure 34: LabVIEW code. Set max flow limit request (Source: Author's own)

The events shown through this section are activated with an interactive component in the interface. The position of each of them depends on the importance and the category that the resulting action has. More information about the interface can be found in Section 4.3.

4.2.6 Acquisition of sensor data

The improved Armfield C6 Fluid Friction Apparatus has three pressure sensors (Section 3.1.3). An NI USB-6008 device (Section 3.1.4) allows the communication between the LabVIEW code and the sensors. The code used to obtain and present the data is shown in Figure 35.

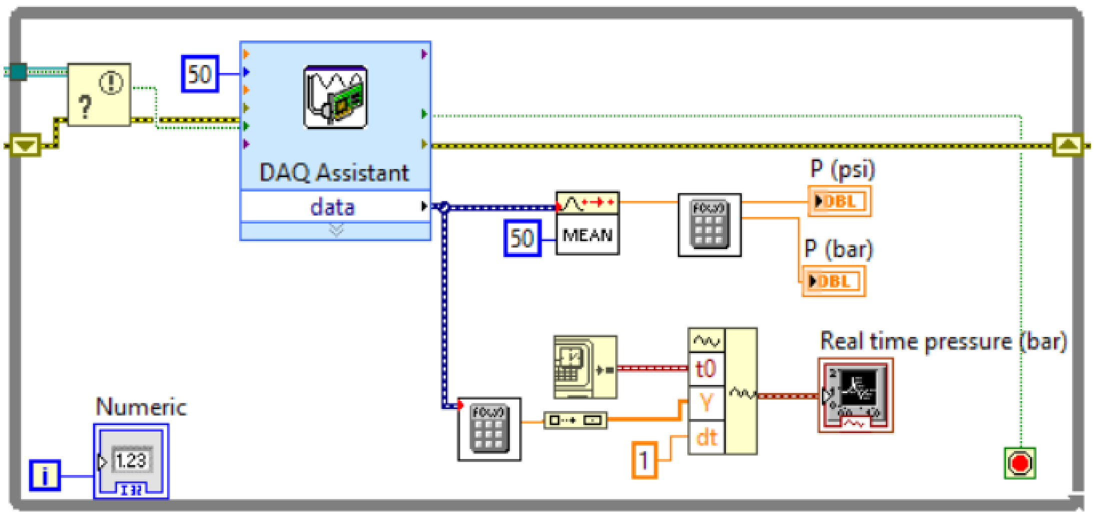


Figure 35: LabVIEW code. Sensor data acquisition (Source: Author's own)

The “DAQ Assistant” VI is responsible for acquiring the data. The mentioned VI is configured individually for each signal. In this case, the settings specified are the Analog Input (a0, a1 or a2 depending on the sensor that is being read), voltage as the magnitude being read, 1 Hz as the sampling rate and “Continuous measurement”.

The sensor measurement is not usually stable, some oscillations are expected. To prevent the value from changing continuously in the interface indicator, a mean of the last 50 values has been implemented.

Figure 36 shows the function “CalculateSensorPressure” (represented with the calculator icon in the code in Figure 35). This function converts the voltage value obtained from the sensor to a pressure value. The formula used is given by the sensor manufacturer in the datasheet. The resulting value is expressed in psi, so a manual conversion to bar and kPa has also been implemented.

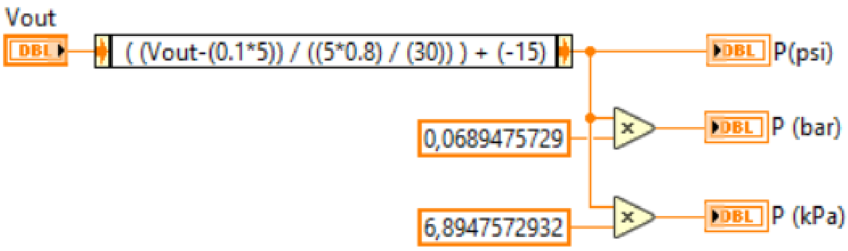


Figure 36: LabVIEW function. CalculateSensorPressure (Source: Author's own)

The pressure resulting from the “CalculateSensorPressure” function (Figure 36) is displayed in the interface both in a real-time chart and a display.

This same code (Figure 35) has to be implemented as many times as sensors to be read. Go to Appendix 8.5 to see the complete implementation with three sensors. Note that every DAQ block has to be individually programmed. Up to 4 analog inputs can be entered by the USB-6008 device. However, the “DAQ” configuration implies that the device has to be always connected to the same pot of the same computer. Else, the DAQ has to be reconfigured.

4.2.7 Data saving in .csv file

An extra functionality that has been implemented is the option to save data in a .csv file. This functionality has been divided into two: create a new document and save data points. Both options are represented with a button in the interface.

At the start of the program, the “Save data point” button will be disabled, leaving only the option to create a new document. Once the document is created, the user can choose either to add data points to the existing document or create a new one.

Both buttons have been implemented as events in the event case, similarly to the interface elements that allowed control of the pump. Figure 37 shows the event triggered by the “New file” document

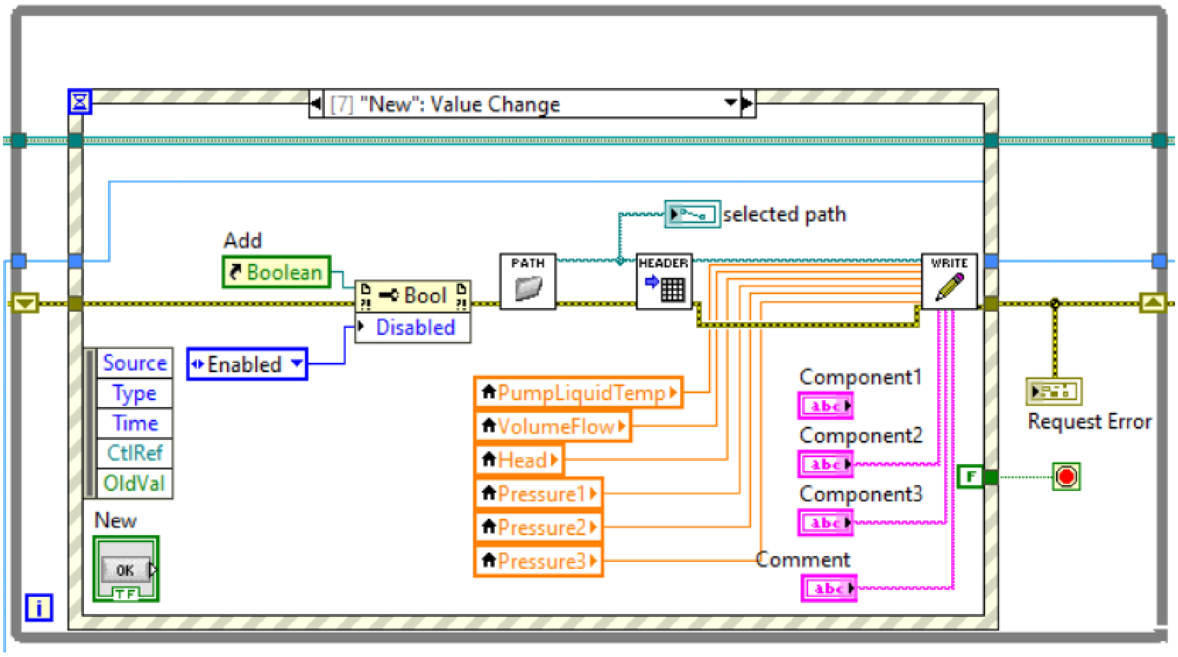


Figure 37: LabVIEW code. Create a new file (Source: Author's own)

By pressing the “New file” button, four actions are done: choose a file path, write the heading, write the first line of data and enable the “save setpoint” functionality. The first three actions have been implemented in a SubVI.

Figure 38 shows the code necessary to choose the file path. In a nutshell, the function establishes a default name for the file (Fluid_fric_yyyymmdd_HHMMSS.csv) and the user can choose where to save it using the screen that pops up. However, the user also has other options, such as cancelling (the file path will be empty), choosing a new name for the file or overwriting an existing file. All these circumstances are also controlled by the function.

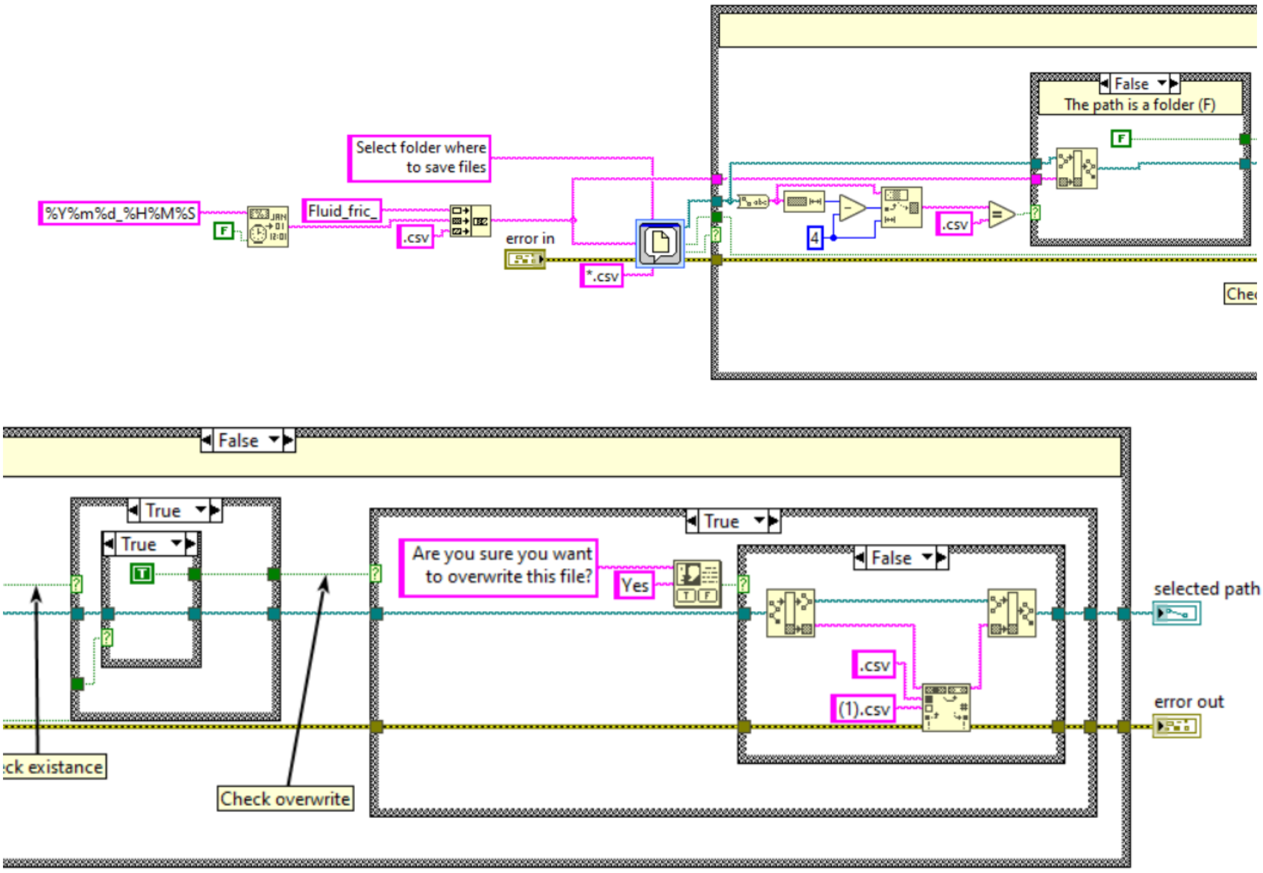


Figure 38: LabVIEW function. SelectPath (Source: Author's own)

Once the file's path is chosen, the function "FileHeader" (see Figure 39), writes the header of the file. The first row of the file contains the data and the hour, the second row indicates the name of the variable that the column will contain while the third one shows the units. In this case, the columns are the following: Temperature (K), Flow (m^3/h), Pressure (KPa), Component, and Comments. This is constant information and will only be written once when the document is created.

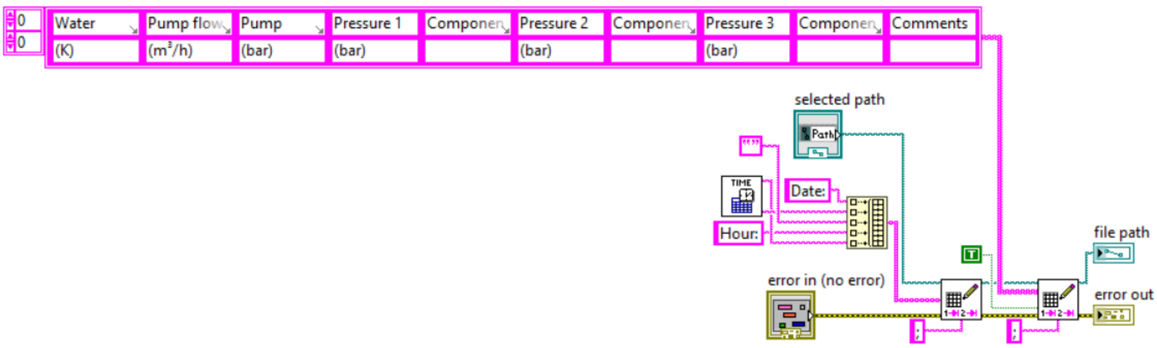


Figure 39: LabVIEW function. FileHeader (Source: Author's own)

Finally, the first row of data is written. The subVI receives the data, arranges it, and prints it to the first empty row. The data is provided by the interface component and must be displayed in the same order specified in the heading. The function "WriteFile" content is shown in Figure 40.

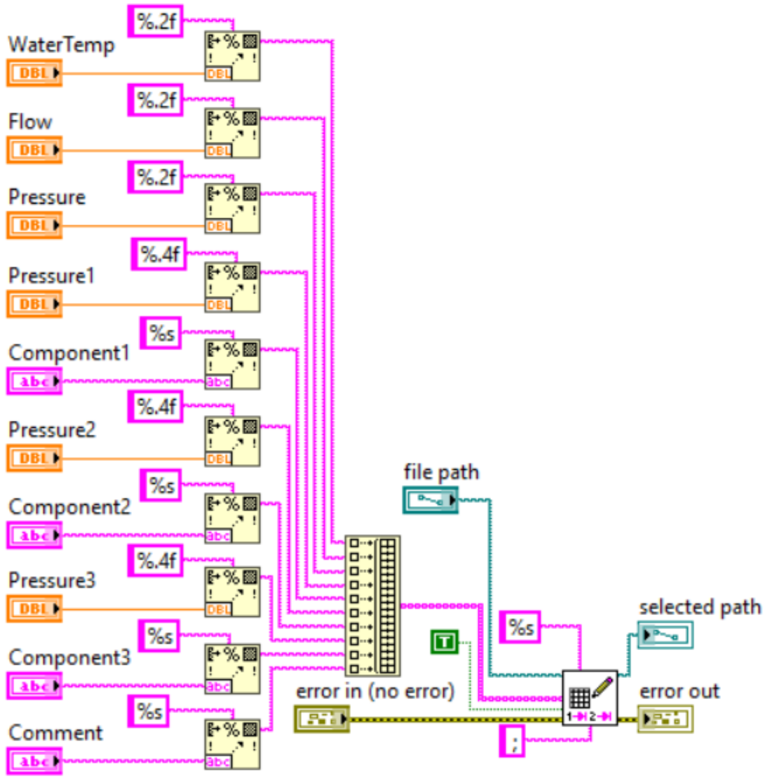


Figure 40: LabVIEW code. WriteFile (Source: Author's own)

The event triggered when the "Save datapoint" is pressed can be seen in Figure 41.

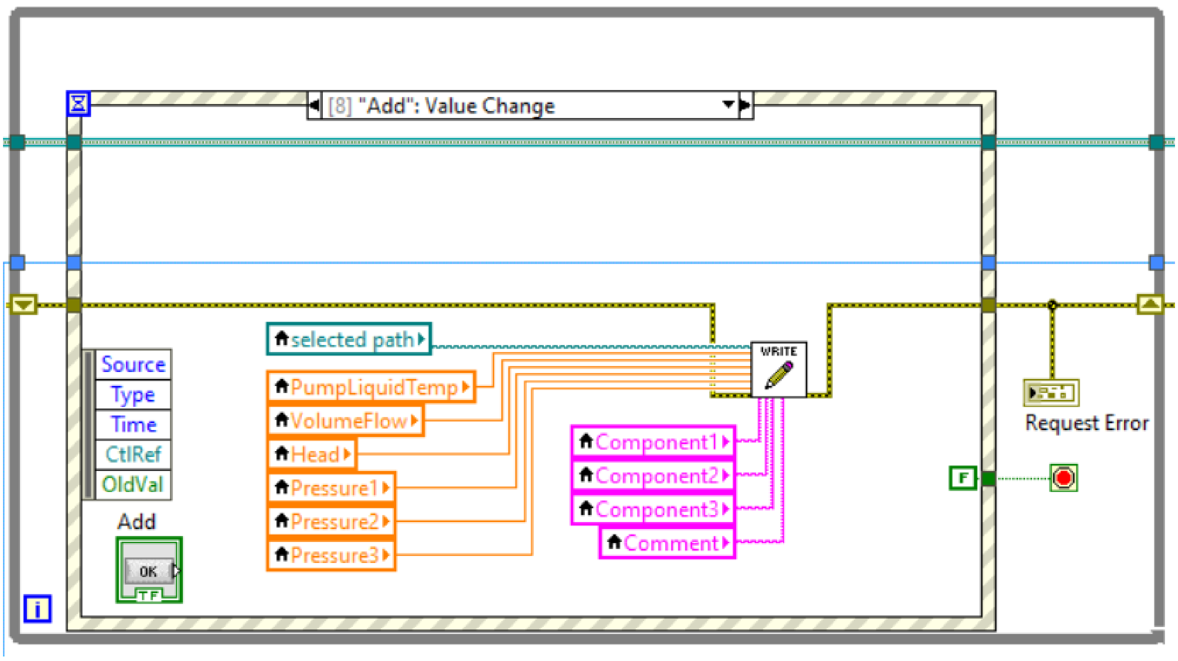


Figure 41: LabVIEW code. Save data point in a file (Source: Author's own)

This event is simple. Using the “WriteFile” function created (Figure 40), the data shown on the screen at the time is printed in the first empty row of the document. Note that, as the indicators containing the information are the same, local variables have been created.

4.2.8 Error Handling

Error handling determines how the program will respond to and recover from errors. Error handling functionality is essential in any software application (Techopedia, 2017).

The LabVIEW code developed contains some error handling strategies to prevent the malfunctioning of the program in unexpected situations. Most of the processes are subject to an “error” check before execution. If there has been an error, the process is skipped.

One clear example is the relationship established between “TCPmasterInit” (the first function of the program) and the rest of the code. If an error has occurred, a message asking the user to check the TCP connection pops up (Figure 42).

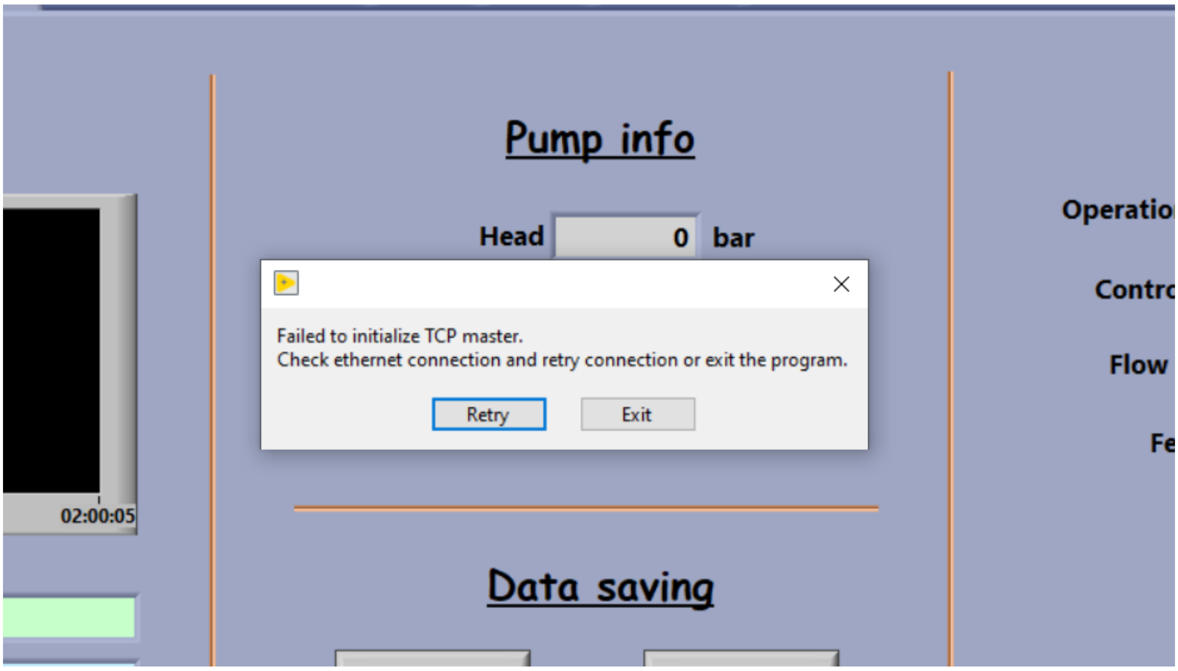


Figure 42: LabVIEW interface. Failed to initialize TCP master error (Source: Author's own)

The error message will continue appearing until the problem is not solved. Meanwhile, the other functionalities won’t execute, as they are inside an if-structure.

4.3 LabVIEW interface design

The user interface ideation, design and implementation are the culmination of this thesis and the purpose of the code development. Cambridge Dictionary (2022) defines an interface as “a connection between a person and a computer”. Applied to this project, the interface will allow the students to communicate with the pump and obtain data from the system.

The aim of the design is the creation of a complete, user-friendly and intuitive interface. To fulfil this objective several opinions were taken into consideration, especially the one from the teacher conducting the exercises in the system.

The final design is formed by 4 different pages (Home, Stats, Alarms, and Advanced configuration) and a permanent heading. Starting from the heading, each of the interface sections is described in the following subsections (from Section 4.3.1 to Section 4.3.5).

4.3.1 Header

The heading is a permanent part of the interface, this means that can be seen regardless of the page the user is at the moment. The information considered essential has been duplicated and can be found both in its “logical” place and in the heading. The heading also contains relevant buttons (e.g. the “Exit” and the ones to access other pages) and general information such as the name of the program and Novia’s logotype.

Figure 43 is a picture of the interface heading. The components information can be found in Table 5. The numbers in the picture correspond to the table number.

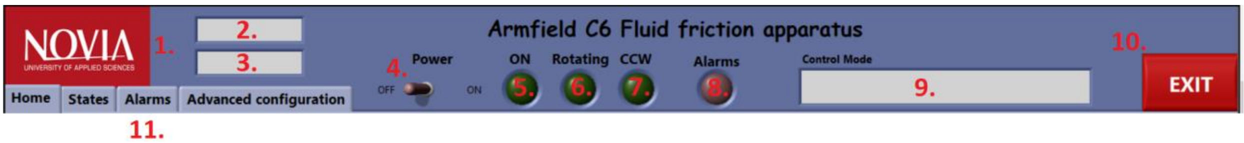


Figure 43: LabVIEW interface. Heading (Source: Author's own)

Table 5: LabVIEW Interface. Heading components information (Source: Author's own)

Number	Name	Source
1	Novia Logotype	Photo
2	Time	LabVIEW function
3	Date	LabVIEW function
4	Pump On/Off	101.1
5	Pump ON	201.9
6	Pump rotating	201.6
7	CCW direction	201.7
8	Pump alarms	201.10
9	Pump current control mode	201.11
10	Exit program	LabVIEW event
11	Access to the different pages	LabVIEW built-in function

The Exit button from the header is used to quit the program and stops all the processes at the same time. To do so, the several stops are connected with notifier blocks. The full implementation can be seen in the main VI (Appendix 8.5)

4.3.2 Home

When starting the program, the default page is the "Home". This page has been designed considering the laboratory exercise. Therefore, all the data and functionalities needed to do the practical exercise have been implemented on the "Home" page.

The "Home" page, shown in Figure , has been divided into three sections: Sensor info, Pump info and Control. The legend can be found in Table 6, where the name and source of the data are specified.

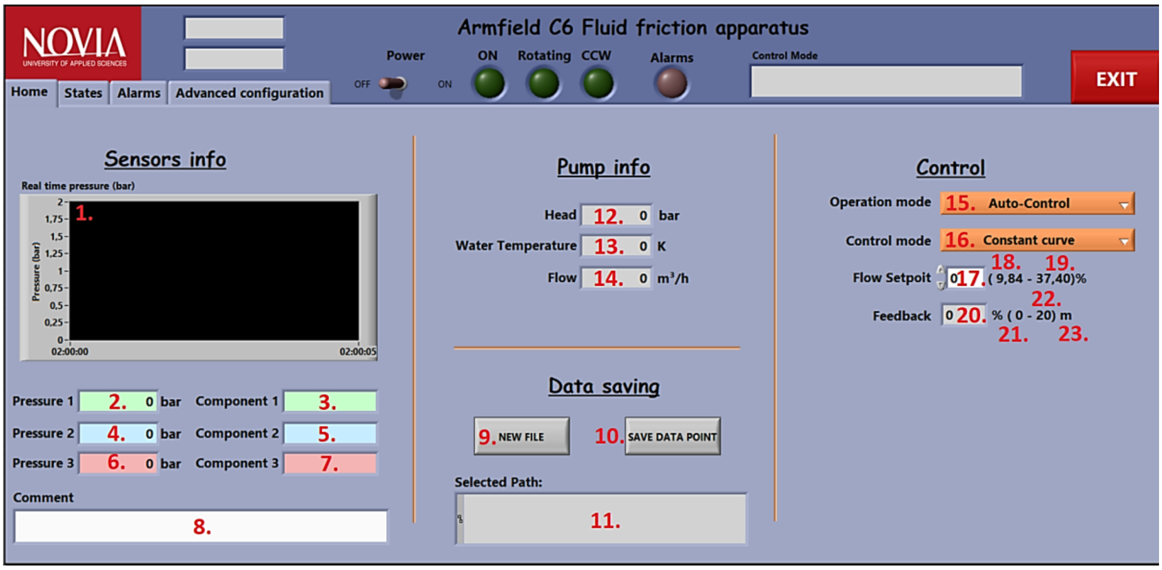


Figure 44: LabVIEW interface. Home (Source: Author's own)

Table 6: LabVIEW interface. Home components information (Source: Author's own)

Number	Name	Source
1	Pressure in real-time	Main VI. Read sensor process
2	Pressure Sensor 1	LabVIEW function
3	Component 1	Input string
4	Pressure Sensor 2	LabVIEW function
5	Component 2	Input string
6	Pressure Sensor 3	LabVIEW function
7	Component 3	Input string
8	Comment	Input string
9	New file	LabVIEW event
10	Save data point	LabVIEW event
11	Selected file path	LabVIEW function
12	Head	301
13	Water temperature	322

14	Flow	302
15	Operation mode	103
16	Control mode	102
17	Setpoint	104
18	Setpoint range max	216
19	Setpoint range min	215
20	Feedback	202
21	Feedback min	210
22	Feedback max	211
23	Feedback unit	209

In a standard usage of the system, the students do not need to change the page. The interface has been designed so all the essentials can be accessed either from the heading or the “Home” page.

4.3.3 States

The “States” page (Figure 45) contains most of the data that can be obtained from the pump. Depending on the type of information, 4 categories have been formed: control info, motors stats, configuration data and historic data.

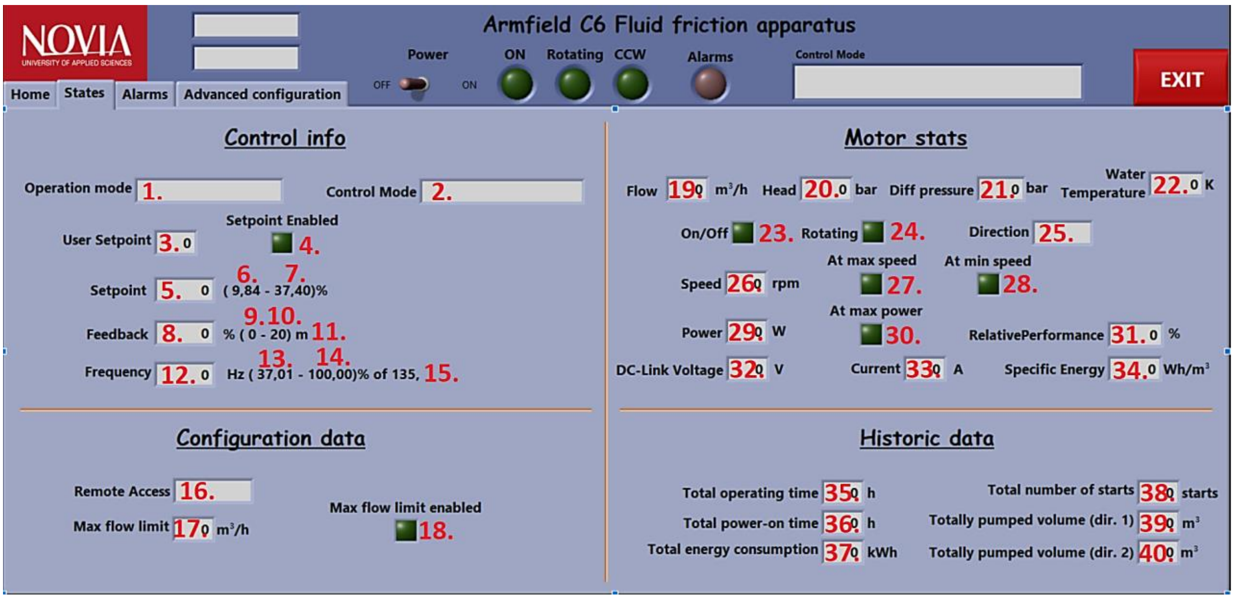


Figure 45: LabVIEW interface. States (Source: Author's own)

The name and address of the registers accessed to obtain the data are specified in Table 7 (heading not included). The number shows where the data is shown in the interface.

Table 7: LabVIEW interface. States components information

Number	Name	Source
1	Operation mode	204
2	Control mode	203
3	User Setpoint	338
4	Setpoint Enabled	201.4

5	Current Setpoint	308
6	Setpoint min	215
7	Setpoint max	216
8	Feedback	202
9	Feedback min	210
10	Feedback max	211
11	Feedback units	209
12	Frequency	305
13	Frequency min	213
14	Frequency max	214
15	Nominal frequency	212
16	Remote access	Combination 201.8+201.12
17	Max flow limit	345
18	Max flow limit enabled	201.2
19	Flow	302
20	Head	301
21	Differential pressure	339
22	Temperature	322
23	On/Off	201.9
24	Rotation	201.6
25	Direction	201.7
26	Speed	304
27	At maximum speed	201.13
28	At minimum speed	201.15
29	Power	312+313
30	At maximum power	201.5
31	Relative Performance	303
32	DC link Voltage	310
33	Motor Current	309
34	Specific energy	326
35	Total operation time	327+328
36	Total powered time	329+330
37	Total energy consumption	332+333
38	Number of starts	334+335
39	Volume direction 1	357+358
40	Volume direction 2	361+362

When the pump is being controlled by Ethernet communication, no changes are allowed to be done from the pump. This section, together with the advanced configuration was created to solve that limitation.

4.3.4 Alarms

The “Alarms” page (Figure 46) displays the alarm and warning codes of the different parts of the improved system. The section has been divided into two parts: the pump alarms and warnings and the Program alarms and warnings. The address and name of the registers containing the information and the functions of the LabVIEW code are specified in Table 8.

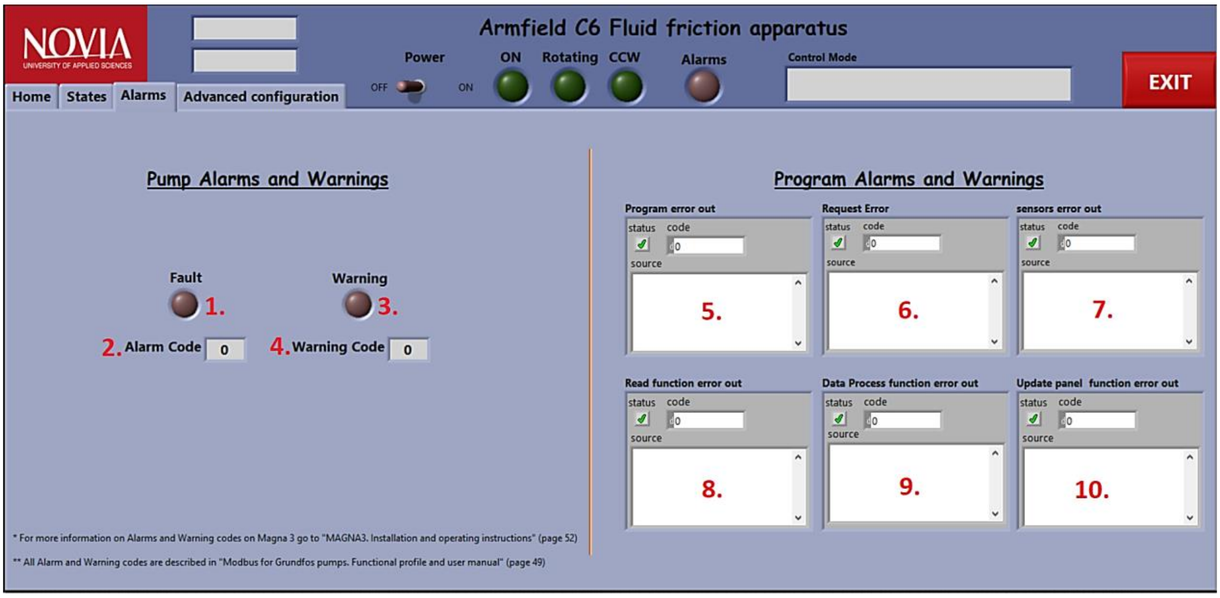


Figure 46: LabVIEW interface. Alarms (Source: Author's own)

Table 8: LabVIEW interface. Alarms components information

Number	Name	Source
1	Fault	201.10
2	Fault code	205
3	Warning	201.11
4	Warning code	206
5	Program error out	Main program error
6	Request error out	Event case error out
7	Sensors error out	Read sensor process error
8	Read function error out	LabVIEW function
9	Data process error out	LabVIEW function
10	Update panel error out	LabVIEW function

Similarly to the "States" page, "Alarms" has not been designed for daily usage. The main objective is the facilitation of searching for errors and malfunctions if there are some. If the pump has either one alarm or a warning, the heading indicator lights up. Specific information about the error must be looked at from the "Alarms" page, and the meaning of the pump error codes can be found in the document "Modbus for Grundfos pumps. Functional profile and user manual" (Grundfos, 2019).

4.3.5 Advanced configuration

Currently, the advance configuration page only contains the option to enable or disable the flow limit and the option to choose a value. This page has been implemented thinking about future implementations of the system. Time will tell which parameters should also be included on this page and which ones should be more accessible.

5 Results and discussion

This section is an inside look and evaluation of the thesis' results. The project was built around the main objective of automatizing a fluid system apparatus and the building of a LabVIEW user interface. The final results have been explained and shown throughout Section 4.

The thesis outcome is the combination of a hardware improvement and a software design. Assuming that a LabVIEW licence and the tools (screwdrivers, cables…) are already owned, the indicative cost of this project is 2082€ in VAT. The components were selected based on performance availability and cost. Detailed information can be found in Section 3.1 and Appendix 8.1.

If the project was to be duplicated, the estimated time for configuring and assembling would be 2 hours. The LabVIEW program and interface should be able to adapt without further issues than configuring the USB port in the DAQ blocks. In the situation where the code needs to be modified, the time cannot be estimated, as the nature of the changes plays too big of a role.

Since the acquisition of the current Armfield C6 Fluid Friction Apparatus, Armfield has continued releasing models. One of the newest models, called "C6-MKII-10 – Fluid Friction Measurements" works with an interface (Armfield, 2021). The advantages of an officially approved interface include access to official diagrams and icons and the possibility of assembling the sensors during the fabrication process, increasing the quality of the installation and precision.

Although the newest version includes a user interface, the hardware and functionality do not have big differences. Besides avoiding changing the apparatus, the clear benefit of the code and interface developed in this project is that they have been made specifically for the laboratory exercise the teacher wants to carry. In order to have a complete overview, the work done will be reflected from several perspectives.

Initial aims and objectives

The first objective was to provide the system with a new range of components. The components chosen and the installation and assembly process required have been explained in Sections 3.1 and 4.1 respectively. The initial intention was to include a wider variety of components to monitor other variables, but in the end, the data provided by the pump together with the pressure sensors was enough to fulfil the needs.

The second objective, intended to develop a LabVIEW program capable to communicate with the system, has been the focus of this project. This objective was met and also surpassed, as the LabVIEW program (explained in Section 4.2) ended up evolving further than expected. The final code not only allows to read and send values but also includes extra applications such as saving the data into a file, functions to avoid undesired situations, graphs, and other improved visualization of the information.

A big part of the additional functionalities of the code is tightly related to the third objective, the design of a user-friendly interface. The interface discussed in section 4.3 has been carefully ideated, designed, and implemented. The interface has also undergone several tests from different people to have a wider range of opinions and improvements. The suggestions and recommendations of the teacher who is going to use the equipment have been strongly considered.

The last objective was intended to verify whether the basic purpose of the automation, improving the system, was accomplished. Unfortunately, due to circumstances out of the scope of this thesis, the whole system could not be implemented. Thus, the laboratory practice could not be recreated and the increment in the precision of the measurement could not be confirmed. However, thanks to the interface, it is a fact that the system is more user-friendly.

Limitations

When developing the program, some limitations have been found. The first thing to consider is the limited knowledge of the programming language. Some of the implementations could not be done in the ideal way that they were thought.

The biggest limitations come from the components' specifications. The precision of the measurement material plays a crucial role in the data obtained. The lack of accuracy in the control of the pump flow has some severe consequences, as the measurements of the pipe with the smallest diameter cannot be done (the flow required is too small).

Related to the communication with the components, the program has another significant limitation. The LabVIEW VI responsible for reading the data from the sensors is specifically programmed to one port of one computer. This means that the program will need to be always executed from the same computer.

None of the limitations described has a critical impact on the objectives of this thesis, but they will limit the future usage and applications of the program developed.

Consistencies and inconsistencies

The idea of how to structure and solve the project has changed and evolved considerably through the process. This has led to some inconsistencies in the documentation. One clear example is Section 2.1, which talks about the theoretical principles of fluid mechanics. The beforementioned section was meant to introduce the Armified C6 Fluid Friction Apparatus and present the concepts necessary to recreate the laboratory exercise in the testing section. Unfortunately, the testing could not be carried out due to reasons external to this thesis.

The program was developed consistently and designed as a SMoRES code allowing and facilitating future upgrades. Aiming to create an automatized and versatile code, all the recurrences and similarities in the data treated have been exploited. The result is a more compact and optimized program. However, the method used has some inconsistencies, as the code is formed by several complex loops and iterations that index the values in order. This means that if a value in between was going to be added, all the indexes of the following variables would need to be changed. However, appending another variable at the end is easy.

Improvement suggestions

The program and the interface have been developed keeping in mind a specific laboratory exercise. One natural step would be to generalize the functionality by adding different sensors and restructuring the home page of the interface.

Right now, the program is completely functional. However, a necessary improvement would be the further development of the error handling. Error handling is a complex yet crucial part of a program. Errors can crush a program or make it behave unpredictably. The current error handling implemented is very basic, leaving some uncontrolled circumstances.

The other peripheric functionalities, such as the data saving in a .csv file, are also quite basic. Adding more functionalities and improving the existing ones will also be needed at some point in the future.

6 Conclusion

Laboratory exercises play a great role in Finnish education. The correct functioning and ease of use of the equipment used are essential for the correct accomplishment of the learning objectives. This thesis has developed a user interface to increase the usability and precision of an Armfield C6 Fluid Friction Apparatus.

The system improvement consisted of upgrading the pump to a MAGNA3 Model D and installing three pressure sensors (ABPDRRV015PDAA5) to substitute the previous manual measurement. The pump is controlled and monitored remotely through a CIM500 ethernet module with Modbus TCP protocol.

When considering the new purchase of an obsolete education tool, a cost-effective improvement to the current apparatus might be the best option. In this thesis's context, a "homemade" upgrade was especially profitable, as the basic functioning and purpose of the system have not changed. The hardware components required to automate the system have an approximate price of 2082€ (itemized information in Appendix 8.1)

The interface communicating with the pump and controlling the acquisition of the measurements has been developed in the LabVIEW environment. The LabVIEW code is responsible for reading and processing the data received. The program was designed as a SMoRES code, giving importance to the possibility of a future update. Moreover, the code developed also implements peripheral functions (such as exporting the measurements in a .csv file) to increase the convenience of the user interface. All the processes have been programmed in parallel, therefore, several actions can be taken place at the same time. Not stopping the measurements improves the accuracy and the reliability of the code.

Thanks to the work done in this thesis, all the data regarding the laboratory exercise can be comfortably accessed, monitored, and saved from the user interface. The option to modify the pump settings is also given.

With the user interface being implemented and functional, the main objective of this thesis was accomplished. However, due to external technical adversities, the system could not be tested as a whole. Therefore, the improvement of the Fluid system's performance could not be proved.

In conclusion, the LabVIEW program is robust for standard usage. However, future improvements in the error handling of the code might be needed. The development was centred on a specific laboratory exercise, but this program enables the future incorporation of other exercises for the Fluid system apparatus.

7 References

Armfield. (2021). *C6-MKII-10 - Fluid Friction Measurements - Armfield*. [online] Available at: https://armfield.co.uk/product/10225/

AUMA (2020). *Modbus TCP*. [online] AUMA - Solutions for a world in motion. Available at: https://www.auma.com/es/productos/sistemas-de-comunicacion/industrial-ethernet/modbus-tcp/

Cambridge Dictionary (2022). interface. [online] @CambridgeWords. Available at: https://dictionary.cambridge.org/fr/dictionnaire/anglais/interface

Cànovas Trujillo, A. and Francés Vigatà, A. (2021). *Solar Photovoltaic Energy Storage System*. [online] Available at: https://www.theseus.fi/bitstream/handle/10024/503435/Solar%20Photovoltaic%20Energy%20Storage%20System.pdf?sequence=2&isAllowed=y

Çengel, Y. A., Turner, R. H., Cimbala, J. M. & Kanoglu, M. 2012. Fundamentals of thermal-fluid sciences. 4th ed. in SI units. Singapore: McGraw-Hill.

Chen, J.X. and Lobo, N. da V. (1995). Toward Interactive-Rate Simulation of Fluids with Moving Obstacles Using Navier-Stokes Equations. Graphical Models and Image Processing, [online] 57(2), pp.107–116. Available at: https://www.sciencedirect.com/science/article/abs/pii/S107731698571012X

Control Solutions (2020). Modbus Tutorial from Control Solutions. [online] Csimn.com. Available at: https://www.csimn.com/CSI_pages/Modbus101.html

Douglas, J. F., Gasiorek, J. M., Swaffield, J. A., & Jack, L. B. (2005). Fluid mechanics. Pearson education. Available at: https://books.google.fi/books?hl=fr&lr=&id=LUMec3jraA4C&oi=fnd&pg=PR19&dq=fluid+definition+fluid+mechanics&ots=WRaWVzdTAC&sig=Aw9LYtBoFLYGOqN7ts7T5CVXCD4&redir_esc=y#v=onepage&q=fluid%20definition&f=false

electronics notes (2022). What is LabVIEW: Graphical System Design» Electronics Notes. [online] Electronics-notes.com. Available at: https://www.electronics-notes.com/articles/test-methods/labview/what-is-labview.php

Farnell (2022). *LabVIEW from NI*. [online] Farnell. Available at: https://be.farnell.com/ni-labview

Fox, R. W., McDonald, A. T., & Mitchell, J. W. (2020). Fox and McDonald's introduction to fluid mechanics. John Wiley & Sons. Available at: https://books.google.fi/books?hl=ca&lr=&id=v4T3DwAAQBAJ&oi=fnd&pg=PA2&dq=Fluid+flow+is+classified+as+external+or+internal,+depending+on+whether+the+fluid+is+forced+to+flow+over+a+surface+or+in+a+conduit.+Internal+and+external+flows+exhibit+very+different+characteristics.&ots=2a9salymIg&sig=bCd_Q56nuAoX7JZFEur7THRQHCw&redir_esc=y#v=onepage&q&f=true

Frohm, J. (2008). *THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY Levels of Automation in Production Systems*.

Grundfos (2019). *Modbus for Grundfos pumps. Functional profile and user manual*. [online] Available at: https://api.grundfos.com/literature/Grundfosliterature-6012947.pdf.

Grundfos (2021a). *CIM 500 Ethernet module. Installation and operating instructions.*

Grundfos (2021b). *MAGNA 3. Installation and operating instructions.*

Grundfos (2021c). *MAGNA 3. Model D. Circulator pumps.*

Hani Morgan (2014) Review of Research: The Education System in Finland: A Success Story Other Countries Can Emulate, Childhood Education, 90:6, 453-57, DOI: 10.1080/00094056.2014.983013

Holland, I. and Davies, J.A. (2020). Automation in the Life Science Research Laboratory. *Frontiers in Bioengineering and Biotechnology*, [online] 8. Available at: https://www.frontiersin.org/articles/10.3389/fbioe.2020.571777/full

Honeywell (2021). *ABP SERIES. Basic Board Mount Pressure Sensors.* [online] Available at: https://www.mouser.fi/datasheet/2/187/honeywell_sensing_basic_board_mount_pressure_abp_s-1662208.pdf

Huitsing, P., Chandia, R., Papa, M. and Shenoi, S. (2008). Attack taxonomies for the Modbus protocols. *International Journal of Critical Infrastructure Protection*, [online] 1, pp.37–44. Available at: https://www.sciencedirect.com/science/article/pii/S187454820800005X?casa_token=Ql_UzdHkfJoAAAAA:6nuXOSx-qKjbLhkS9NW1ycyzPqVKFqukpa4U3LbEE__CuQx87MRUbb-GjK9MKbQUtddCzXMw1_Y#b1

Jackson, D., & Launder, B. (2007). Osborne Reynolds and the publication of his papers on turbulent flow. Annu. Rev. Fluid Mech., 39, 19-35. Available at: https://www.annualreviews.org/doi/abs/10.1146/annurev.fluid.39.050905.110241

Modbus Organization (2006). *MODBUS Messaging on TCP/IP Implementation Guide V1.0b*. [online] Available at: https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.

Modbus Organization (2012). Modbus application protocol specification V1.1b3. [online] Modbus.org. Available at: https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Modbus Organization (2022). *Modbus Specifications and Implementation Guides*. [online] Modbus.org. Available at: https://modbus.org/specs.php

National Instruments (2015). *USER GUIDE. NI USB-6008/6009 Bus-Powered Multifunction DAQ USB Device.* [online] Available at: https://www.ni.com/pdf/manuals/375295c.pdf

NI (2019). What is LabVIEW? [online] Ni.com. Available at: https://www.ni.com/fi-fi/shop/labview.html#

NI (2022). *Application Development with Modbus*. [online] Ni.com. Available at: https://www.ni.com/fi-fi/innovations/white-papers/14/application-development-with-modbus.html

Oecd.org. (2018). *Publications - PISA*. [online] Available at: https://www.oecd.org/pisa/publications/pisa-2018-results.htm

Oxford Learner's Dictionaries (2022). *automation noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com*. [online] Available at: https://www.oxfordlearnersdictionaries.com/definition/english/automation

Philbert (2020). *SourceForge*. [online] SourceForge. Available at: https://sourceforge.net/projects/qmodmaster/

ProSoft Technology (2014). *Understanding Modbus Serial and TCP/IP*. *YouTube*. Available at: https://www.youtube.com/watch?v=k993tAFRLSE.

Rathakrishnan, E. (2013). Applied Gas Dynamics. [online] Google Books. Available at: https://books.google.fi/books?hl=fr&lr=&id=W1-HDwAAQBAJ&oi=fnd&pg=PP15&dq=gas+dynamics&ots=cO_bdRZBEd&sig=5gBBj-0NRD_9udF8wBp1deH7VPU&redir_esc=y#v=onepage&q=gas%20dynamics&f=false

ReadPhysics. (2021). What is hydrodynamics? Laws - Applications - Exercise - Pitot meter..... [online] Available at: https://readphysics.com/what-is-hydrodynamics/#:~:text=What%20does%20hydrodynamics%20study%3F,only%20liquids%2C%20but%20also%20gases

Swales, A. and Electric, S. (1999). *OPEN MODBUS/TCP SPECIFICATION*. [online] Available at: http://www.dankohn.info/projects/Fieldpoint_module/Open_ModbusTCP_Standard.pdf.

Techopedia (2017). *Error Handling*. [online] Techopedia.com. Available at: https://www.techopedia.com/definition/16626/error-handling [Accessed 11 May 2022].

Turns, S. R. 2006. Thermal-fluid sciences: An integrated approach. Cambridge: New York: Cambridge University Press.

Vergara, D., Extremera, J., Rubio, M.P. and Dávila, L.P. (2020). The Technological Obsolescence of Virtual Reality Learning Environments. *Applied Sciences*, [online] 10(3), p.915. Available at: https://www.mdpi.com/2076-3417/10/3/915

Wygnanski, I., & Champagne, F. (1973). On transition in a pipe. Part 1. The origin of puffs and slugs and the flow in a turbulent slug. Journal of Fluid Mechanics, 59(2), 281-335. doi:10.1017/S0022112073001576. Available at: https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/abs/on-transition-in-a-pipe-part-1-the-origin-of-puffs-and-slugs-and-the-flow-in-a-turbulent-slug/EDF674FFD2DEDE5D6DC3DF9E67C7415D

zhanglongqi (2017). *GitHub - zhanglongqi/qModMaster: The maintainer's repo. https://github.com/ed-chemnitz/qmodbus/*. [online] GitHub. Available at: https://github.com/zhanglongqi/qModMaster

8 Appendices

8.1 Components purchase information

Name	Manufacturer	Model	Price	General link	Datasheet	Picture
Pump	Grundfos	MAGNA 3 model D 25-80 180	870.49 €	https://product-selection.grundfos.com/products/magna/magna3/magna3-25-80-97924246?tab=variant-specifications&pumpsystemid=1559246162	https://product-selection.grundfos.com/products/magna/magna3-d?tab=documentation	
Ethernet module	Grundfos	Cim 500 PROFINET Modbus TCP BACnet IP EtherNet/IP	815€	https://product-selection.grundfos.com/products/cimciu/cimciu-profinet-modbus-tcp-bacnet-ip-grm-ip/cim-500-profinet-modbus-tcp-bacnet-ip-ethernetip-98301408?tab=variant-specifications	https://manualzz.com/doc/22936936/cim-500-ethernet-module	
Pressure sensor	Honeywell	ABPDRRV015PDAA5	21.89€	https://www.mouser.fi/ProductDetail/Honeywell/ABPDRRV015PDAA5?qs=OTrKUuiFdkba7Xsn8eTsNA%3D%3D	https://www.mouser.fi/datasheet/2/187/honeywell_sensing_basic_board_mount_pressure_abp_s-1662208.pdf	
Multifunction I/O Device	National instruments	NI USB-6008	330€	https://www.ni.com/fi-fi/support/model.usb-6008.html	https://www.ni.com/pdf/manuals/375295c.pdf	

8.2 Important configuration parameters

<u>Computer configuration</u>

IP: 192.168.1.105

Subnet mask: 255.255.255.0

Default gateway: 192.168.1.1

<u>Pump configuration</u>

IP: 192.168.1.2

Subnet mask: 255.255.255.0

Gateway: 192.168.1.1

<u>Cim 500 Ethernet Module configuration</u>

Selector position: 1 (Modbus TCP)

Ethernet port used: 1

<u>Grundfos Webserver</u>

Url: 192.168.1.100

User: admin

Password: Novia-2022

<u>Modbus parameters</u>

Port: 502

Pump Unit ID: 100

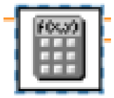
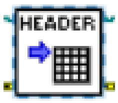
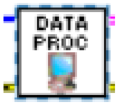
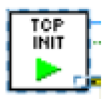
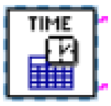
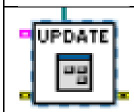
<u>Sensor configuration</u>

Computer port: COM2

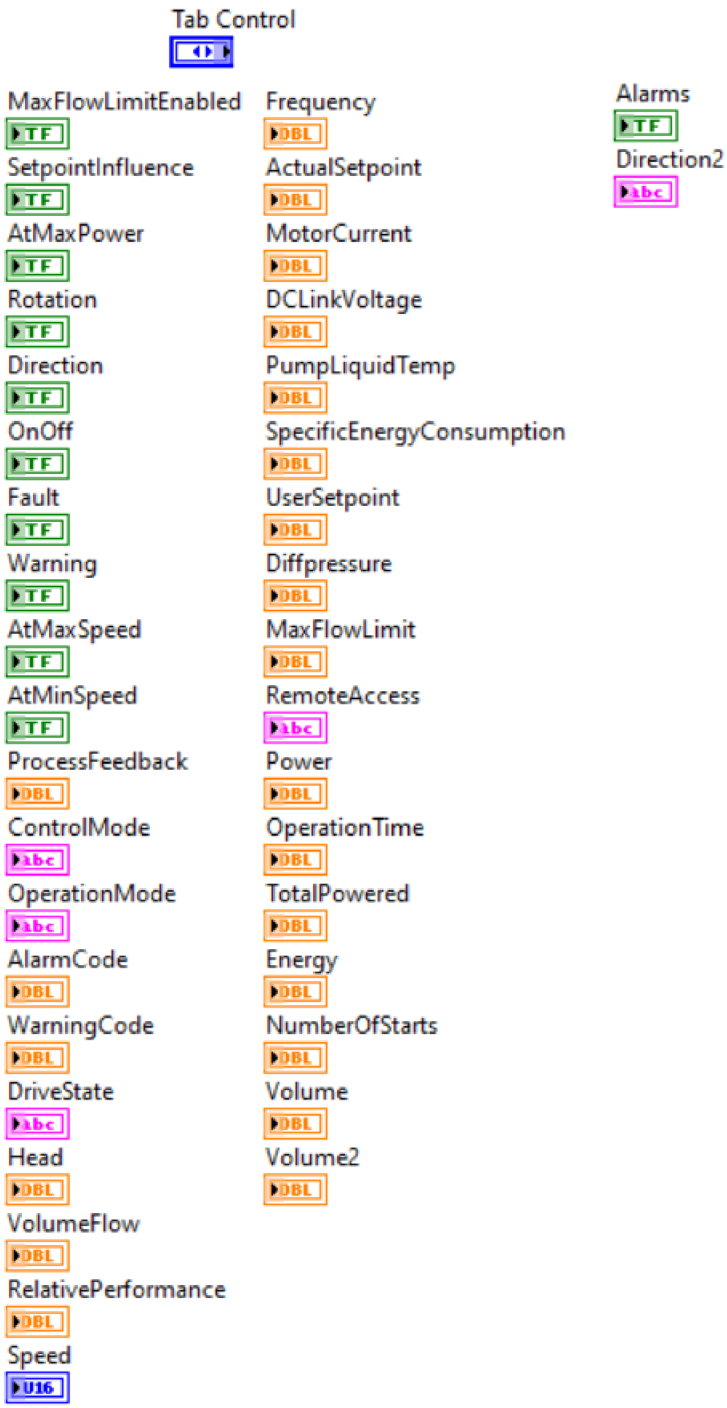
8.3 Read registers information

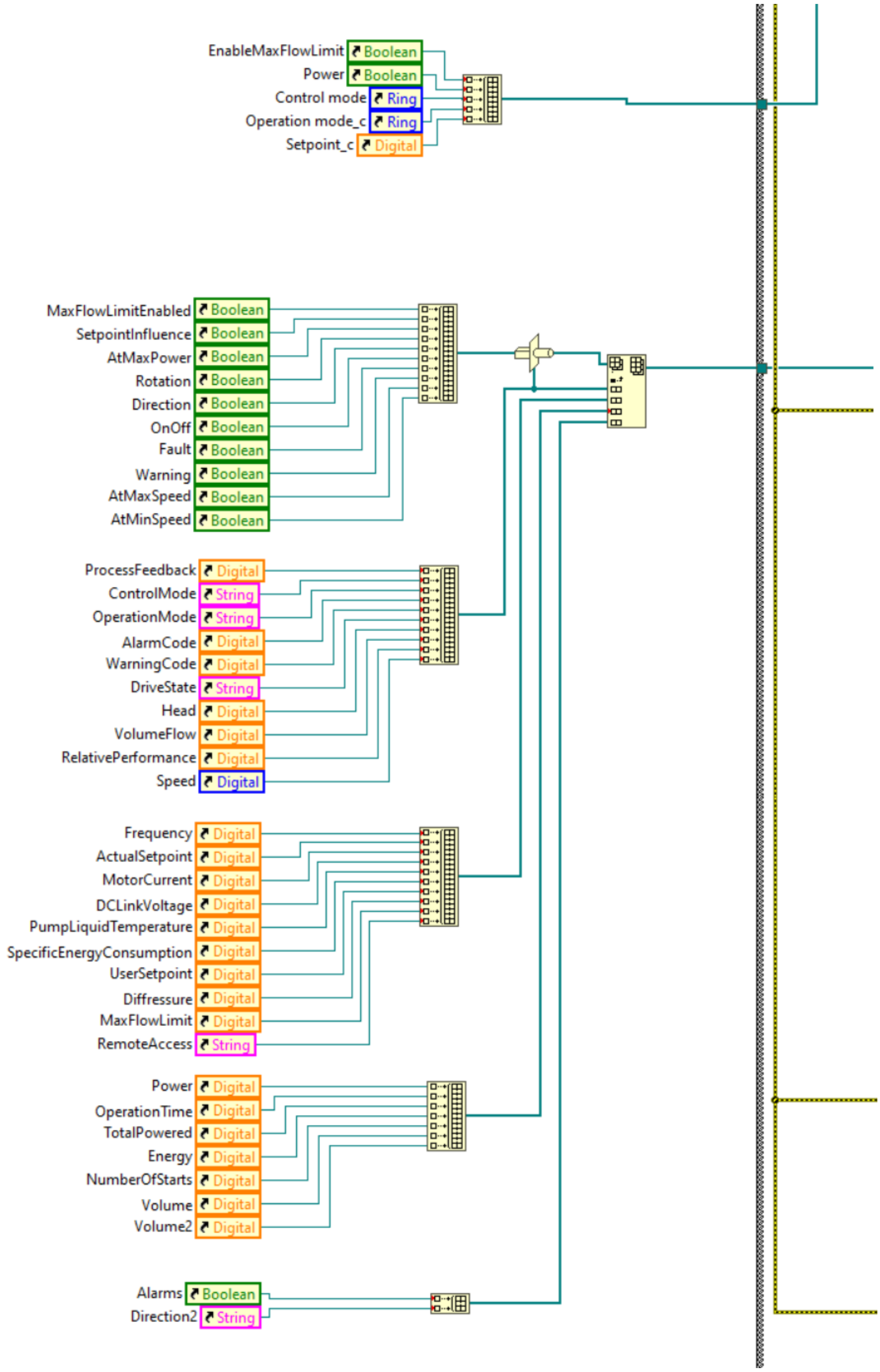
Address	Bit	Name	Array index	Data type	Type specif.	Double mult	var index	Ind index
201	2	MaxFlowLimitEnabled	0	bool			0	0
201	4	SetpointInfluence	0	bool			1	1
201	5	AtMaxPower	0	bool			2	2
201	6	Rotation	0	bool			3	3
201	7	Direction	0	bool/string			4	4
201	9	OnOff	0	bool			6	5
201	10	Fault	0	bool			7	6
201	11	Warning	0	bool			8	7
201	13	AtMaxSpeed	0	bool			10	8
201	15	AtMinSpeed	0	bool			11	9
202	-	ProcessFeedback	1	number	double	0.01	12	10
203	-	ControlMode	2	string			13	11
204	-	OperationMode	3	string			14	12
205	-	AlarmCode	4	number	uint8		15	13
206	-	WarningCode	5	number	uint8		16	14
208	-	DriveState	7	string			17	15
301	-	Head	16	number	double	0.001	26	16
302	-	VolumeFlow	17	Number	double	0.1	27	17
303	-	RelativePerformance	18	number	double	0.01	28	18
304	-	Speed	19	number	uint16		29	19
305	-	Frequency	20	number	double	0.1	30	20
308	-	ActualSetpoint	23	number	double	0.01	31	21
309	-	MotorCurrent	24	number	double	0.1	32	22
310	-	DCLinkVoltage	25	number	double	0.1	33	23
322	-	PumpLiquidTemp	29	number	double	0.01	36	24
326	-	SpecificEnergyConsumption	33	number	uint16		37	25
338	-	UserSetpoint	45	number	double	0.01	46	26
339	-	Diffpressure	46	number	double	0.001	47	27
345	-	MaxFlowLimit	47	number	double	0.1	48	28
		Remote Access		string				29
		FeedbackCaption		caption				30
		FrequencyCaption		caption				31
		SetpointCaption		caption				32
		Power		number				33
		OperationTime		number				34
		TotalPowered		number				35
		Energy		number				36
		NumberOfStarts		number				37
		Volume		number				38
		Volume2		number				39

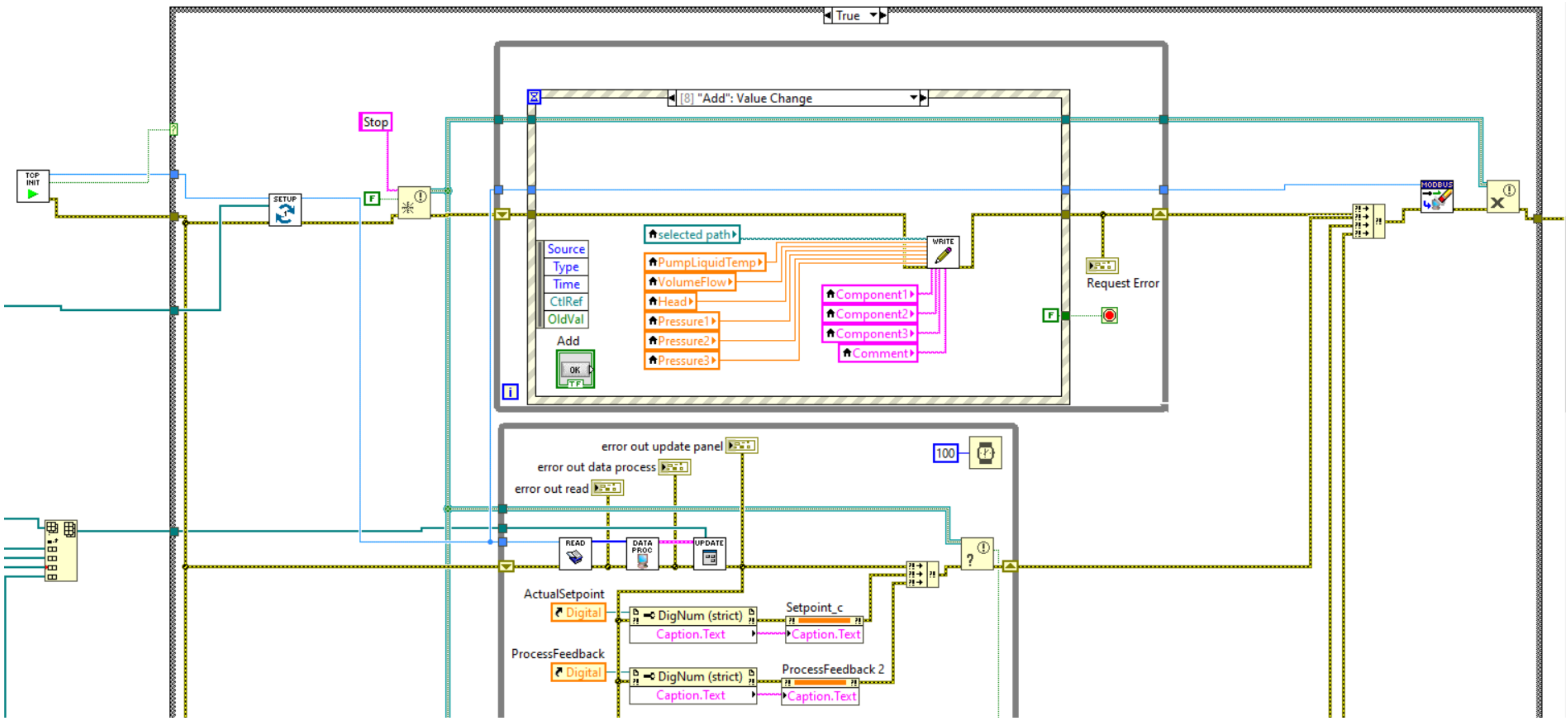
8.4 LabVIEW user-defined functions overview

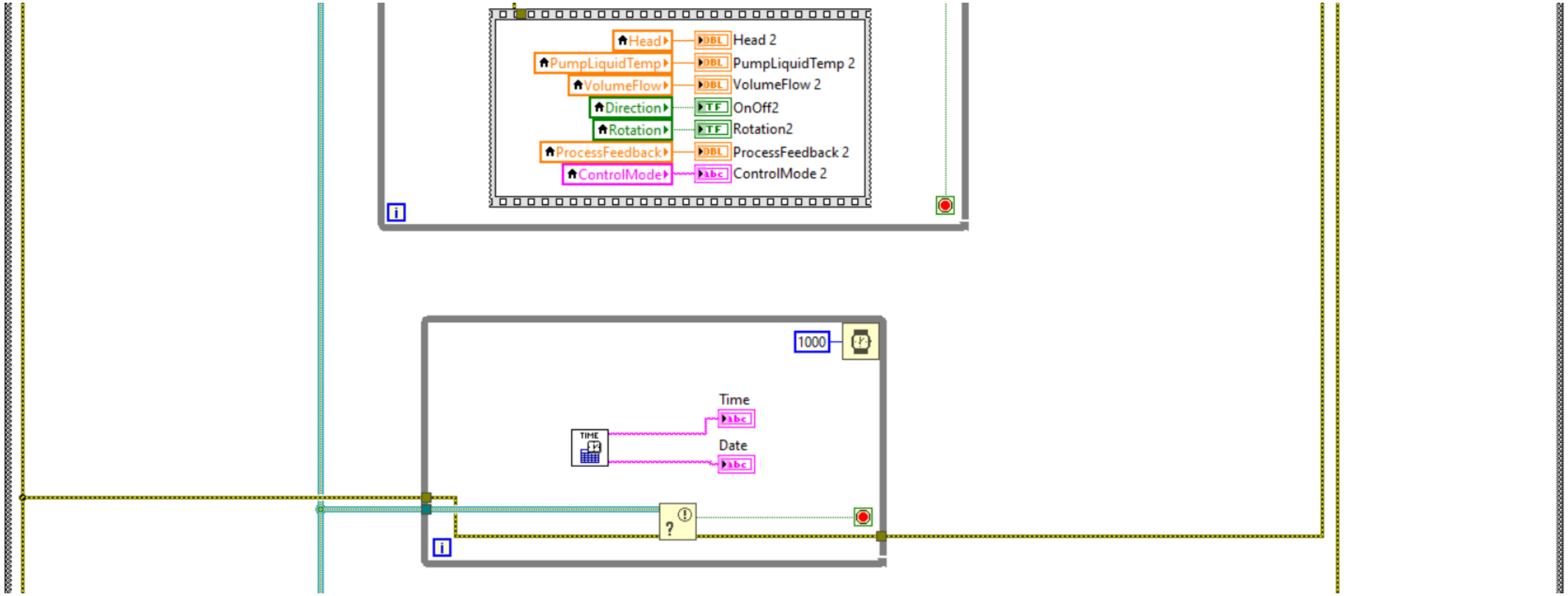
Icon	Name
	Calculate Sensor Pressure
	File Header
	Process Data
	Read
	Select Path
	Start Update
	TCP Master Init
	Time
	Update Panel
	Write File

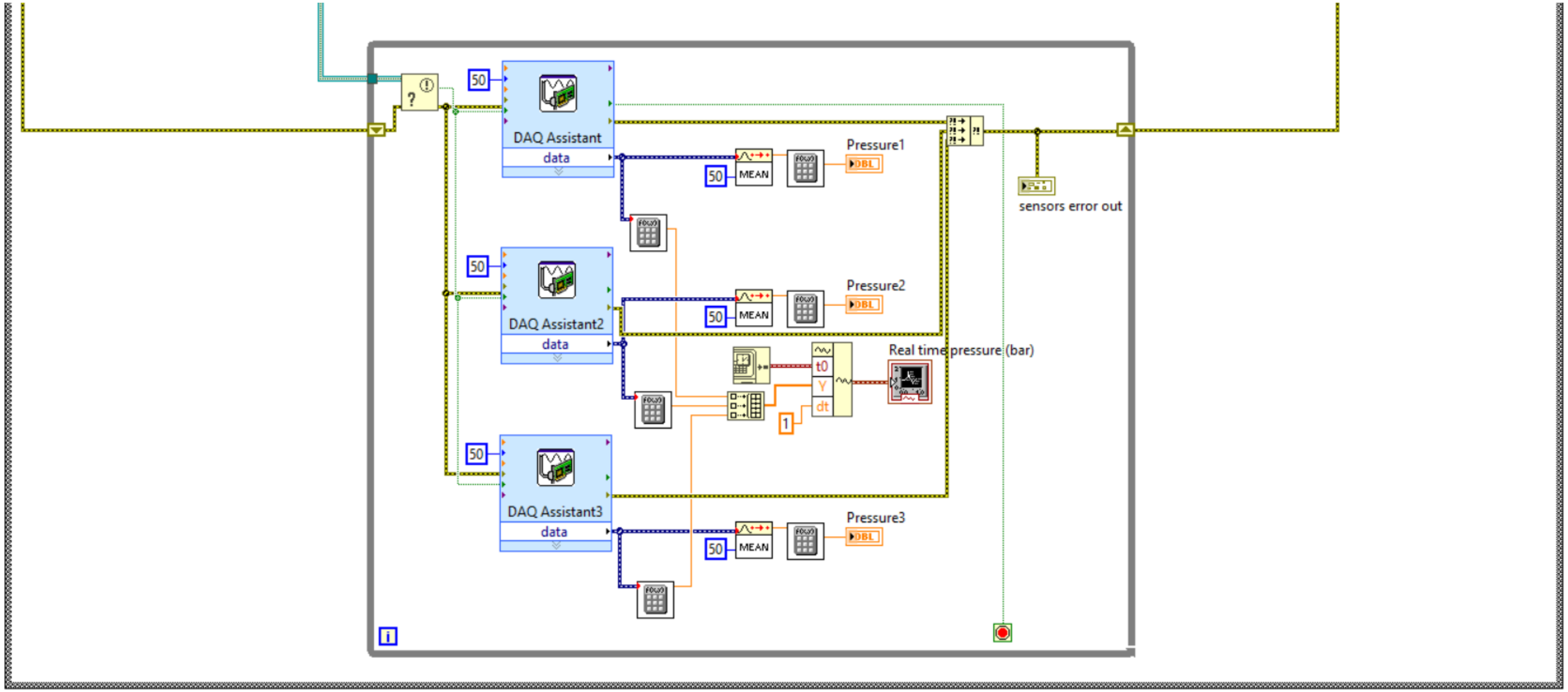
8.5 LabVIEW main VI




EnableMaxFlowLimit Boolean
Power Boolean
Control mode Ring
Operation mode_c Ring
Setpoint_c Digital
MaxFlowLimitEnabled Boolean
SetpointInfluence Boolean
AtMaxPower Boolean
Rotation Boolean
Direction Boolean
OnOff Boolean
Fault Boolean
Warning Boolean
AtMaxSpeed Boolean
AtMinSpeed Boolean
ProcessFeedback Digital
ControlMode String
OperationMode String
AlarmCode Digital
WarningCode Digital
DriveState String
Head Digital
VolumeFlow Digital
RelativePerformance Digital
Speed Digital
Frequency Digital
ActualSetpoint Digital
MotorCurrent Digital
DCLinkVoltage Digital
PumpLiquidTemperature Digital
SpecificEnergyConsumption Digital
UserSetpoint Digital
Diffressure Digital
MaxFlowLimit Digital
RemoteAccess String
Power Digital
OperationTime Digital
TotalPowered Digital
Energy Digital
NumberOfStarts Digital
Volume Digital
Volume2 Digital
Alarms Boolean
Direction2 String


True
[8] "Add": Value Change
Stop
TCP INIT
SETUP
selected path
PumpLiquidTemp
VolumeFlow
Head
Pressure1
Pressure2
Pressure3
Component1
Component2
Component3
Comment
WRITE
Source
Type
Time
CtlRef
OldVal
Add
OK
Request Error
MODBUS
error out update panel
error out data process
error out read
READ
DATA PROC
UPDATE
100
ActualSetpoint
Digital
DigNum (strict)
Caption.Text
Setpoint_c
ProcessFeedback
ProcessFeedback 2


Head
PumpLiquidTemp
VolumeFlow
Direction
Rotation
ProcessFeedback
ControlMode
Head 2
PumpLiquidTemp 2
VolumeFlow 2
OnOff2
Rotation2
ProcessFeedback 2
ControlMode 2
1000
Time
Date


DAQ Assistant
data
DAQ Assistant2
data
DAQ Assistant3
data
50
MEAN
Pressure1
Pressure2
Pressure3
DBL
Real time pressure (bar)
t0
Y
dt
1
sensors error out

8.6 LabVIEW "ProcessData" function