Faculty of Science and Technology

MASTER'S THESIS

Study program/Specialization: Robotics and Signal Processing	Spring semester, 2021 Open
Writers: Stian Arnesen Vidar Haugen	[illegible] ………………………………………… (Writer's signature) [illegible] ………………………………………… (Writer's signature)

Faculty supervisor:
Kristian Thorsen
External supervisor(s):
Roald Kommedal

Thesis title:

A system for automatic cultivation of microorganisms in the biology laboratory.

Credits (ECTS): 2 x 30

Key words: SBR, Biology laboratory, Automation, GUI	Pages: 119 + enclosure: 6 pages appendix Stavanger, 15.06.2021 Date/year

University of Stavanger

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

A system for automatic cultivation of microorganisms in the biology laboratory

Master's Thesis in Robotics and Signal Processing
by

Stian Arnesen
Vidar Haugen

Supervisors:
Kristian Thorsen, UiS
Roald Kommedal, UiS

June 15, 2021

Abstract

Currently, the biology laboratory at the University of Stavanger has a manually controlled microorganism cultivation setup used for studying different types of biological wastewater treatment. The setup uses a sequencing batch reactor (SBR) where the sampling of wastewater and data logging must be done manually. With this in mind, this thesis aims to create an automated version of the SBR setup to make it easier for the lab users to perform experiments and to make it possible to get more reliable automatically measured and sampled, data for analysis.

Our automated setup consists of a computer and a developed control program, that communicate with connected devices such as sensors, pumps, valves, and magnetic mixers. All device communication is achieved by digital communication, where the devices support different data transmission schemes and communication protocols. Through this communication and with the use of the developed control program, the operation of the reactor and its internal environment is controlled. With the use of the control program's graphical user interface, the user can easily control and monitor the active reactor process.

The user of the control program can choose the control parameters of the process to meet the desired reactor environment. The user is also provided with three different control methods for the reaction stage; Oxygen uptake rate (OUR)-, reactor-, and sequential mode. These modes provide different options in the reactor's air supply control and for automatic estimation of bacterial growth. Throughout the process, all data and actions are logged into an excel spreadsheet for later analysis.

In addition, a control cabinet is made to support the interface hardware for the SBR setup. Every device connects to the computer through this control cabinet. If needed, the device's physical communication interface is converted/adapted to USB. The control cabinet also handles power distribution for both AC and DC components.

The verification tests show a working automated SBR setup, and the three control methods work as intended. Unfortunately, due to covid-19, some delays on the equipment prevent testing more than one reactor setup. Both the control program and the control cabinet is made to control up to four reactors simultaneously.

Acknowledgements

This thesis concludes our five-year study and the degree in Master of Technology in Robotics and Signal processing at the Department of Electrical Engineering and Computer Science at the University of Stavanger. The scope of this thesis is 30 credits, completed in the spring semester of 2021. The work with the thesis was very interesting, and we have gained more knowledge about demanding and interesting work tasks within our field of robotics and signal processing.

We want to thank our supervisors Kristian Thorsen and Roald Kommedal at the University of Stavanger for advice, guidance, and words of encouragement throughout the work with the thesis. We would also like to thank the engineers at Henrysalen, Ståle Freyer, Romuald Karol Bernacki and Didrik Efjestad Fjereide for their open-door policy and for providing a helping hand during our thesis. In addition, we would like to thank the biologist at the bio laboratory, and the technicians at the UiS workshop for their assistance.

Furthermore, we want to thank our families and friends who have supported us throughout the thesis period and contributed with inputs and motivational feedbacks, and our fellow classmates for a good and social work environment during the master's degree.

Table of Contents

Abstract i

Acknowledgements ii

1 Introduction 1
- 1.1 Task description . . . 1
- 1.2 Work method . . . 2

2 Theory 3
- 2.1 Activated sludge process . . . 3
- 2.2 Sequencing batch reactor . . . 4
- 2.3 Biomass (bacterial) growth . . . 5

3 Hardware 7
- 3.1 Fluid containers . . . 7
  - 3.1.1 Reactor . . . 7
  - 3.1.2 Flow cells . . . 9
- 3.2 Sensors . . . 10
  - 3.2.1 Hamilton sensors . . . 10
  - 3.2.2 Flood sensor . . . 11
- 3.3 Air pump . . . 11
- 3.4 Peristaltic pumps . . . 12
  - 3.4.1 Ismatec Reglo ICC pump . . . 12
  - 3.4.2 Heidolph Pumpdrive 5201 . . . 13
- 3.5 Weight scale and magnetic stirrer . . . 14
  - 3.5.1 IKA RET control-visc . . . 14
  - 3.5.2 IKA lab disc . . . 16
- 3.6 Solenoid valve . . . 16
  - 3.6.1 ASCO pinch solenoid valve, S106 . . . 17
- 3.7 Fluid transportation and air supply . . . 20
- 3.8 Control cabinet . . . 22
  - 3.8.1 I/O module & Ethernet switch . . . 25
  - 3.8.2 USB hub . . . 27
  - 3.8.3 Power supply . . . 28
  - 3.8.4 Circuit breaker . . . 29
- 3.9 Wiring . . . 30
  - 3.9.1 Wiring schematics . . . 31

4 Communication Interface **35**
4.1 Communication protocols 37
4.1.1 RS-232 Serial line 37
4.1.2 RS-485 Serial line 39
4.2 ASCII 40
4.3 MODBUS 40
4.3.1 MODBUS RTU 40
4.3.2 MODBUS TCP/IP 45
4.4 Hamilton sensor communication 47
4.5 Ismatec Reglo ICC communication 53
4.6 Heidolph communication 56
4.7 IKA RET control-visc communication 58
4.8 ADAM-6066 communication 60

5 Control program **63**
5.1 Graphical user interface 63
5.1.1 Main-tab 64
5.1.2 SBR tabs 66
5.1.3 Device communication tabs 69
5.2 Program operations 77
5.2.1 Setup 77
5.2.2 Initialize flow cells 79
5.2.3 Main sequence 80
5.2.4 Individual communication 90
5.2.5 Log 90

6 Verification **91**
6.1 Main tab 92
6.2 Control methods 93
6.2.1 OUR mode 94
6.2.2 Reactor mode 96
6.2.3 Sequential mode 97
6.3 User log 99

7 Limitations and further improvements **100**
7.1 Limitations 100
7.2 Further developments of the physical setup 100
7.3 Further developments of the user interface and control program 102

Bibliography

Appendix

A Specification Sheets
A.1 Hamilton sensors . . . . .
A.2 Magnetic mixers . . . . .
A.3 Peristaltic pumps . . . . .
A.4 ADAM-6066 & Ethernet switch . . . . .
A.5 Converters & Adapters . . . . .
A.6 USB hub . . . . .

A.7 Valves . . . .
A.8 Power supply & circuit breaker . . . .
A.9 Electrical sockets . . . .

B Operation Manuals
B.1 Hamilton sensors . . . .
B.2 Peristaltic Pumps . . . .
B.3 IKA RET control-visc . . . .
B.4 ADAM-6066 . . . .
B.5 USB to RS-485 converter . . . .

C Equipment list

D Calculations

E Communication appendix
E.1 CRC calculation . . . .

F Excel log

G Libraries

H User Manual

I Video of SBR process

Abbreviations

SBR - Sequencing Batch Reactor
OUR - Oxygen Uptake Rate
DO - Dissolved Oxygen
STR - Solids Retention Time
USB - Universal Serial Bus
TCP - Transmission Control Protocol
RTU - Remote Terminal Unit
PDU - Protocol Data Unit
ADU - Application Data Unit
CRC - Cyclical Redundancy Check
DTE - Data Terminal Equipment
DCE - Data Communication Equipment
HMI - Human-Machine Interface
GUI - Graphical User Interface
OD - Outer Diameter
ID - Inner Diameter
RPM - Revolutions Per Minute
RCD - Residual Current Breaker
RCBO - Residual Current Breaker with Overcurrent
W - Watt
V - Volt
A - Ampere
Ω - Ohm

1. Introduction

1.1 Task description

In the biology laboratory at the University of Stavanger, there is a manually controlled microorganism cultivation process used for studying different types of biological wastewater treatment. Our motivation is to develop a automated system of this kind and make the everyday easier for the lab users who do experiments on this type of wastewater treatment. The benefits of automating the process are that the lab user can collect more data during the process and, when running multiple experiments, the automated system will ensure that the environment and routines are the same for each experiment. This will result in better consistency and lower variance than when experimenting on a similar but manual process.

The lab users currently have to take samples manually from the reactor and place the wastewater inside another machine for analysis, resulting in two reactors where the environment could change in one of them. Therefore, the measurements are not as representative as one would wish, and by automating the process, analysis is happening continuously and gives a more representative result.

This study aims to automate and control a microorganism cultivation process in the biological laboratory at the University of Stavanger. The process is a sequencing batch reactor which is an activated sludge process for biological wastewater treatment [1]. Figure 1.1 illustrates the sequencing batch reactor setup in this study. The biological processes will be explained in detail in chapter 2.

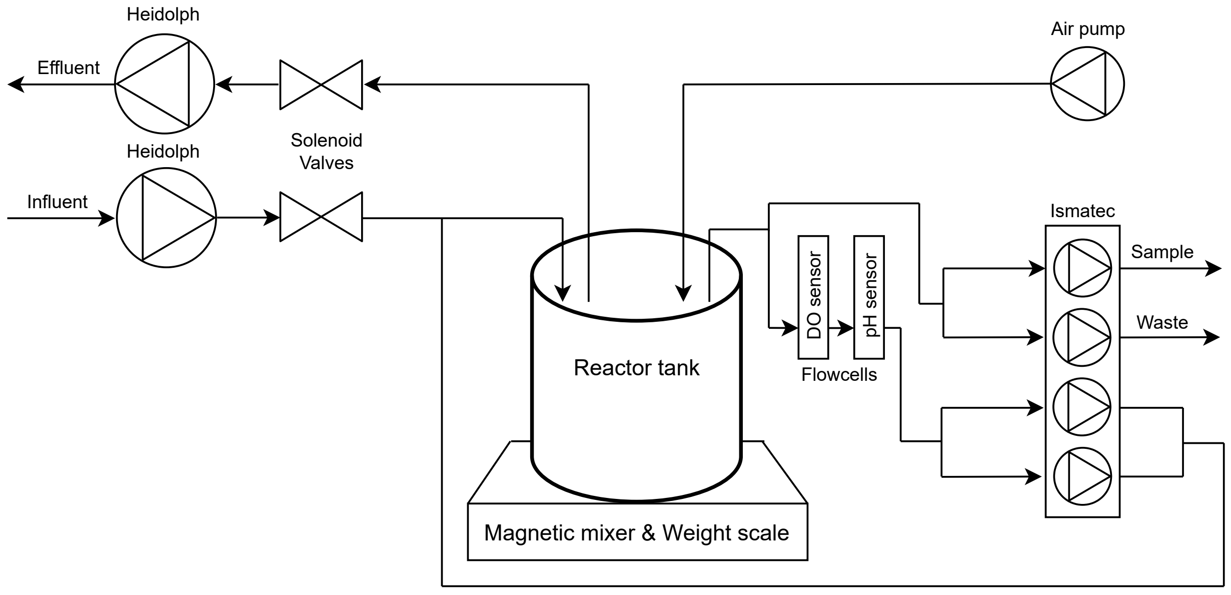


Figure 1.1: Illustration of the sequencing batch reactor setup.

By controlling the process through a computer using LabView, it is possible to measure, control, and do calculations like oxygen uptake rate (OUR) that benefit the user when analyzing the wastewater. The user controls the program and process through an informative graphical user interface. The software exports sensor data after every sequence into an excel spreadsheet for storage and later analysis. All electrical equipment connects to the control unit through two USB hubs and/or serial converters placed inside a control cabinet. The liquids and air are transported through silicon tubes which connect the fluid containers and flow cells together with the reactors. It is possible to control up to four of the setups in figure 1.1, simultaneously. Eight solenoid valves control which reactor setup should have inflow and outflow of liquids as two pumps (IN and OUT) handle all four reactor setups.

1.2 Work method

Throughout the project, a simplified V-model has been used as a work method. Figure 1.2 displays the work method graphically.

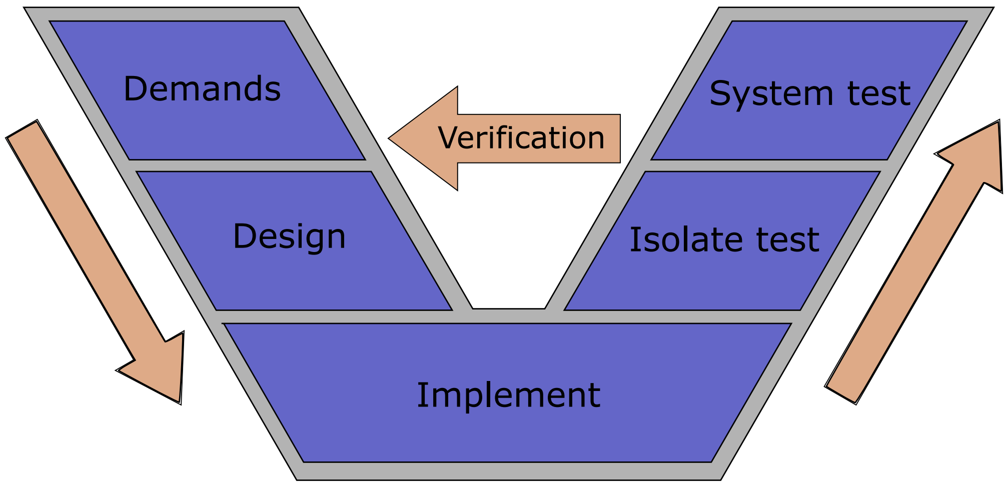


Figure 1.2: Work method V-model

The work method starts by defining clear requirements and demands, i.e., what is needed, to solve given issues. When the demands are defined, the design of the solution begins. Once the solutions have been designed, they can be implemented into the setup. First, the new implementation needs an isolated test to ensure that the solution meets the requirements set and solves it in a satisfactory manner. Further, the new implementation is tested with the whole system to ensure that the new solution does not disrupt other devices or other parts of the system. If any of the tests, isolate- or system test, does not satisfactorily solve the issues, the demands or the design is re-evaluated before the process repeats as shown in figure 1.2.

2. Theory

Wastewater is water that should not be reused or discharged back into natural waters due to contamination and should undergo treatment. In this study, biological treatment of wastewater is used, i.e., activated sludge process. The following section will discuss some of the basics about the activated sludge process and a variant called sequencing batch reactor, which we will focus on and automate.

2.1 Activated sludge process

The activated sludge process is a biological treatment process that treats wastewater and sewage water by the use of microorganisms, and air [2]. The activated sludge process consists of three main parts: A reaction/aeration tank, a settling tank, and a recycling system. The reaction tank is where the microorganisms responsible for the treatment are kept in suspension and aerated. The settling tank is where the solids are separated from the liquid. Some solids from the settling tank are recycled back into the reaction tank or wasted in the recycling system [1]. The figure 2.1 illustrates this.

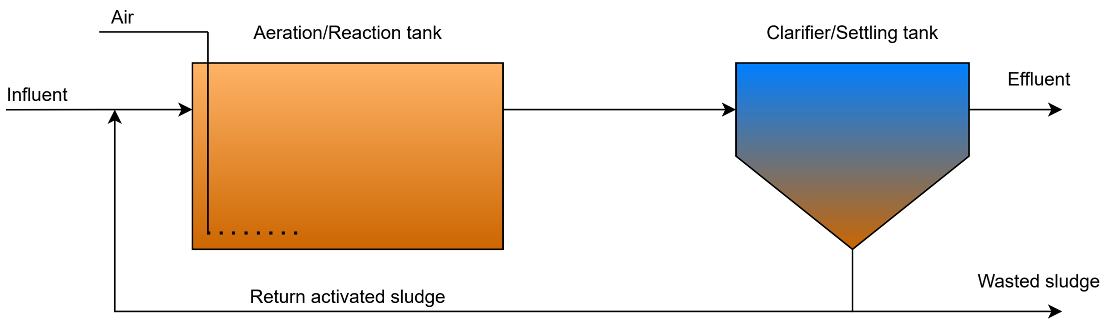


Figure 2.1: Diagram of a activated sludge process.

Influent is the raw, untreated wastewater that flows into the reaction tank together with excess activated sludge from previous cycles. Air bubbles through the mixture, and the reaction begins. The microorganisms break down organic material and consume carbon, nitrogen, and phosphor to grow. They form biological flocs (activated sludge), which are easier to separate from the liquids. The mixed liquids are discharged into the settling tank, and the sludge separates from the liquid by gravitation settling. The supernatant (treated influent, transparent layer of liquids) is extracted from the tank as effluent and can undergo further treatment before discharge [3]. The returned activated sludge is used to re-seed the new wastewater. Some of the sludge is wasted to keep the ratio of biomass to food (wastewater) supplied in balance. At steady state, the amount of sludge wasted is the net growth of sludge per time. The wasted sludge is stored and could also undergo further treatment prior to disposal [4].

Because the activated sludge process uses aeration, COD (chemical oxygen demand) is reduced by biological oxidation and this reaction demands oxygen [1]. COD is an indicative

measure of the amount of oxygen that reactions in a measured solution can consume. COD is helpful in terms of water quality by providing a metric to determine the effect an effluent will have on the receiving body [5].
There are many different variants of activated sludge processes, but the sequencing batch reactor is used in this study.

2.2 Sequencing batch reactor

The sequencing batch reactor (SBR) is a variant of the activated sludge process for wastewater treatment that performs every process step of an activated sludge process in a single mixed reactor. SBR is, in some cases, a convenient way of treating wastewater, especially in laboratory and small-scale setups, because of the simplicity of only using one reactor basin, and as it is quite space efficient it is also a popular technology when space is limited. The typical operation phases for the sequencing batch reactor is is shown in figure 2.2, and defined below [1]:

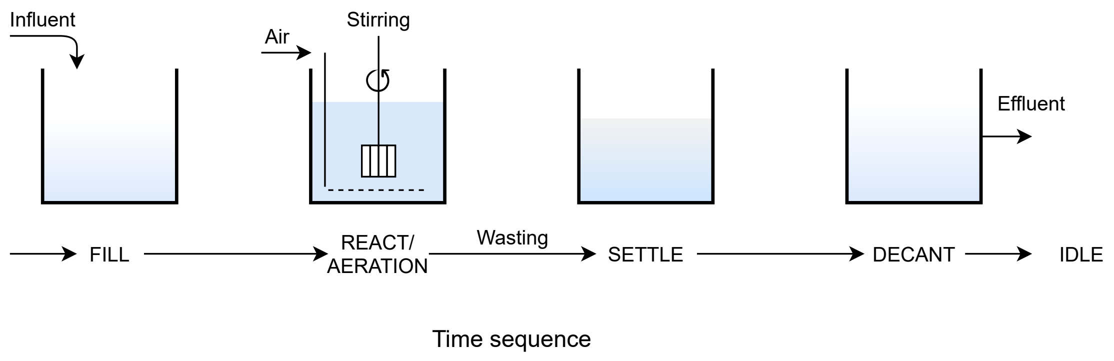


Figure 2.2: Diagram of a sequencing batch reactor process.

Fill: Influent wastewater is added to the reactor. The mixing and aeration can start in this step to promote reactions with influent wastewater.

React/Aeration: This is the main step, where the biomass consumes the substrate, the volume in this stage is constant. The aeration can be constant or cyclic. In this study, both constant and cyclic aeration is used. Cyclic aeration encourages nitrification and denitrification for nitrogen removal.

Wasting is introduced and keeps the reactor from overcrowding with microorganisms. Wasting occurs at the end of the react stage by extracting liquids and solids while stirring. This also affects the performance of the SBR [1]. This is the main mechanism for controlling the solids retention time (SRT), i.e., sludge age.

Settle: The aeration and mixing stop to allow the sludge/biomass to separate from the liquid. Here a transparent layer appears, which is the supernatant and is ready for decanting. The sludge at the bottom of the reactor is used to re-seed the next reactor cycle.

Decant:. The clarified supernatant is withdrawn from the reactor. There are many methods of decanting, e.g., pumps, valves, or adjustable floating weir. In this study, pumps are used.

Idle: This is the end of an SBR sequence. Idling is some time before a new sequence starts. This step is optional.

2.3 Biomass (bacterial) growth

For an organism to continue reproducing and functioning correctly, one must fulfill both growth conditions and nutritional requirements. The microorganisms need a carbon source, an energy source, and nutrients to synthesize new cellular material and cell growth.

The carbon source can be derived from either organic matter or carbon dioxide, depending on which type of organism. The organisms used in this study is heterotroph, which uses organic carbon for the formation of new biomass [1]

One could describe the growth patterns in the reaction phase of a batch reactor in four stages: **lag phase**, **exponential growth phase**, **stationary phase** and a **death phase**. At the start of the batch, there is only a small population of biomass that exists, and substrate and nutrients are present in excess [1]. The figure 2.3 illustrates these stages.

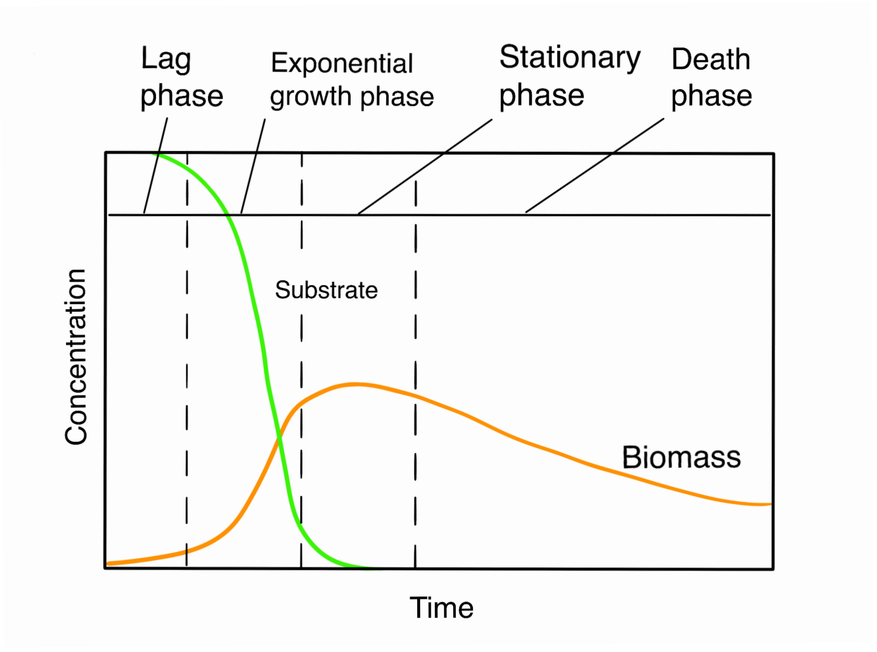


Figure 2.3: Batch process biomass growth stages with changes in substrate and biomass over time [1].

The **lag phase** occurs first. This phase represents the time needed for the microorganisms to adapt to their new environment before significant cell division and biomass production occur [1] In the **exponential growth phase** the microorganism cells are multiplying at their maximum rate, as there is no limitation due to substrate or nutrients [1]. As seen from figure 2.3 the growth curve increases exponentially, proportional to the biomass. Temperature is only the factor that affects the rate of the exponential growth in this step [1].
In the **stationary phase** the substrate concentration tends towards zero, and thus this leads to minor changes in the growth of biomass.

The last phase is the **death phase**. When the substrate is depleted, the population of cell growth stops. The change in population of biomass is the death of cells that occurs when no substrate is available.

Oxygen and substrate consumption

The biomass uses oxygen to grow as they consume the substrate [1]. Oxygen uptake rate (OUR) is the rate the biomass consume oxygen. By disabling the aeration and measure the DO (dissolved oxygen) using a DO sensor, it is possible to estimate the OUR. A DO measure describes how much oxygen is dissolved in the water.

In a batch reactor, the oxygen uptake rate is theoretically defined as:

$$\text{OUR} \equiv \frac{dO_2}{dt} \tag{2.1}$$

Equation 2.3 shows that OUR is the change in oxygen over time. A common way to estimate OUR in practice is to measure DO as it reduces while aeration is turned off and then use linear regression to calculate the rate of change in the DO.

3. Hardware

As mentioned in chapter 1, this project aims to fully automate a sequencing batch reactor, where a computer controls all operations. To achieve this, we will create a physical reactor setup with the necessary sensors and control program with a graphical user interface (GUI).

Firstly, we cover the physical setup, external equipment, e.g., magnetic mixer, pumps, sensors, fluid containers, and internal equipment like the control cabinet and its components. Figure 1.1 illustrates the necessary lab components chosen for this reactor setup. The following subsections details what properties and functions are required of the components.

3.1 Fluid containers

This section will now go through the fluid containers and glass risers used. These containers are the main reactor and the flow cells.

3.1.1 Reactor

As a container for the SBR, a DURAN GLS 80 stirred reactor with a capacity of 2000 ml is used [6]. The GLS 80 stirred reactor is suitable for a wide range of requirements for laboratory mixing processes. The reactor itself consists of a borosilicate glass 3.3, which is chemically and thermally highly resistant [7].

The top screw cap contains a magnetic stirring bar and four holes necessary for the supply of both fluids and air to the reactor. The magnetic stirrer consists of a stainless steel rod connected to the screw cap, with an anchor stirrer and spindle connected at the bottom. A standard commercial magnetic stirrer provides the drive for the stirrer. The stirrer's rotation speed is suitable for speeds up to 500 revolutions per minute (rpm) [6].

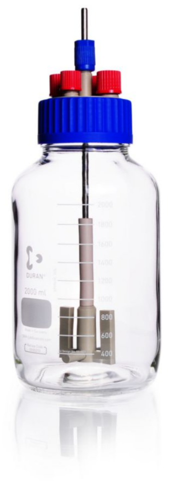


Figure 3.1: Image of a DURAN GLS 80 stirred reactor [6].

Glass risers

For fluid and air transportation to the reactor, glass risers are placed through the holes of the top cap. The used risers are bent to avoid the stirrer at the bottom of the reactor, as shown in figure 3.2.

One glass riser is placed in each of the top cap's four holes. Two of the four holes have an inner diameter (ID) of 5 mm, while the other two have an ID of 3 mm. Through the two different hole dimensions, glass risers with 3 mm and 5 mm OD are used.

The connection between the glass risers and the silicone tubes is made by pulling the silicone tubes over the risers. This connection creates a seal between the two parts.

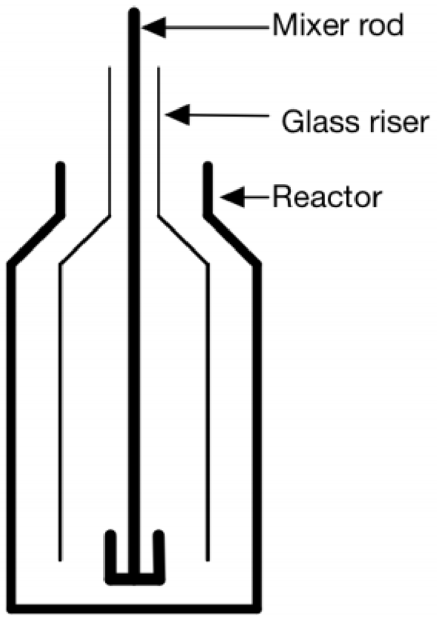


Figure 3.2: Illustration of the glass risers inside the reactor.

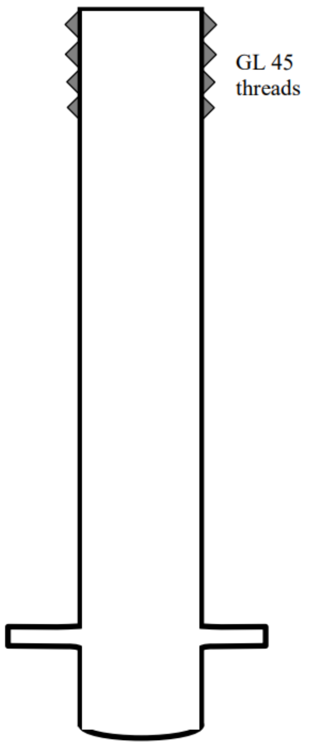
The 5 mm glass risers are dedicated to the filling and emptying pumps, which are responsible for the largest fluid transportation. While the filling riser is lowered to the bottom of the reactor, the emptying riser is lowered to the 400 ml mark inside the reactor so that the settled microorganisms will remain at the bottom of the reactor during emptying.

Having the emptying glass riser at the 400 ml mark will also prevent the emptying pump from pumping out all the fluids if a software error occurs. An eventual software error may result in the pump not stopping at its mark and continue emptying the reactor.

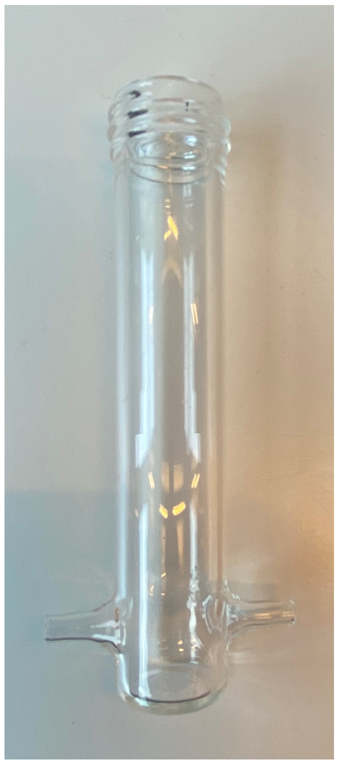
The 3 mm glass risers are dedicated to the air supply and the flow cell fluid loop. Both risers are lowered to the bottom of the reactor. An air stone diffuser is placed at the end of the air supply riser inside the reactor.

3.1.2 Flow cells

The selected reactor has limited options for sensor placement. Due to the stirrer, it is not possible to place the sensors deep into the reactor. Because of this limitation, the sensors are placed outside of the reactor in separate flow cells. The flow cells are connected to the reactor through silicon tubes in a pump-controlled fluid loop. Smaller magnetic mixers are used on each flow cell to keep the sludge from settling inside the flow cells. The used flow cell are shown as a illustration in figure 3.3a, and as an image in figure 3.3b.



(a) Illustration the flow cells used.



(b) Image of the flow cells used

Figure 3.3: Both an illustration and image of the flow cells that is used.

3.2 Sensors

Sensors are needed for both measurements and to check for potential leakage from any of the reactors. The measurement sensors are dissolved oxygen and pH (with temperature) sensors from Hamilton Company. To check for leakage two wires placed close to each other are used.

3.2.1 Hamilton sensors

In the reaction phase, DO level, pH level, and temperature of the wastewater inside each reactor are measured continuously. To measure these parameters, Hamilton's VisiFerm DO Arc 120 H2 and Hamilton's EasyFerm Bio PHI Arc 120 are used [8, 9]. The VisiFerm measures the dissolved oxygen, and the EasyFerm measures both the pH level and the temperature of the wastewater.

The sensors are so-called intelligent sensors. An intelligent sensor has built-in electrical circuits which allow the sensor to take measurements and also process and transmit the data using a processing unit (micro-transmitter) [10]. In contrast, a base sensor requires a separate transmitter. The intelligent sensors make it possible to send the data as digital signals over a serial bus to a connected computer. It is possible to have multiple sensing devices inside one smart sensor (e.g., temperature and pH). Because the Hamilton Arc sensors have a micro-transmitter integrated, they can store calibration data, diagnostic information, and sensor data. This simplifies calibration and maintenance and provides more reliable measurements [11].

Communication interface

Both sensors use the same communication protocol and support both an analog interface and a digital interface. Since the sensors have the option of a digital interface, it is preferred to use it because of the simplicity and precision of measured values. Hamilton sensors use a two-wire RS-485 MODBUS as its digital interface.

The sensors use a transmission speed of 19200 bit/s as default but can also be configured to other baud rates. The sensor uses the following configuration of the serial RS-485 interface: 8 data bits, 1 start bit, 2 stop bit, and no parity.

3.2.2 Flood sensor

Two trays for the reactor are used to prevent overflooding the laboratory with wastewater if a leakage in the system occurs—two reactors per tray. Inside each of the trays, a simple 2-wire flood sensor is mounted on the side. The two wires are placed close to each other, and if there is a leakage, the wires and the wastewater form a closed circuit, as figure 3.4 illustrates. The wires are connected to digital inputs of a I/O module. When the wires in either one of the trays connect, the control program stops the process. The process can not start as long as there is leakage in the system.

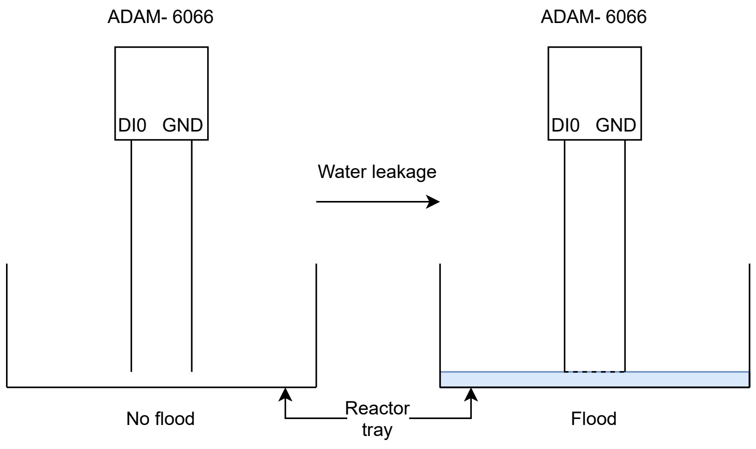


Figure 3.4: Illustration of one of the 2-wire flood sensor used in one of the reactor trays.

3.3 Air pump

As stated in section 2.1, the reactor will need a supply of air during the reaction phase. In this phase, the microorganisms use oxygen. An air pump is used to keep aerobic conditions in the reactor. It is also possible to have anaerobic conditions in the reactor by turning the air pump off. Each reactor has an air pump and is ON/OFF controlled by power relays, resulting in four air pumps if every reactor is used.

EHEIM air 200

The air pump used to aerate the process is the EHEIM air 200. This pump operates on 230 V AC and, as stated above, is controlled by power relays [12]. The pump has a control knob to adjust the air volume, but it is set to max in this setup. A silicon tube is attached to the pump outlet and the glass riser for air transportation to the reactor.

Figure 3.5: The air pump that is used, EHEIM air 200 [12].

3.4 Peristaltic pumps

The fluid pumps in use for this project are all peristaltic pumps. Peristaltic pumps are pumps that transfer fluid in a uniform, and controlled manner without any contact between the fluids and the pump itself [13]. The peristaltic pump is a positive displacement pump where the operation is based on pipe pressing [14]. The fluid content inside the tube is moved by a combination of both a suction- and a discharge principle. The fluid is first drawn into the pump by creating a vacuum on the suction side of the pump, and then the pump's rotor forces the product away along the discharge line of the pump. The rotor and its roller stems do both the suction and displacement operations. The roller stems create temporary seals in the tube or hose, then moves this seal in a clockwise or a counterclockwise direction, depending on the pumping direction.

Figure 3.6 shows how the peristaltic pump operates in two steps. First, the rotor creates a temporary seal on the suction side of the pump with one of its roller stems. When the rotor rotates in a clockwise direction, where it creates a vacuum and draws fluids into the pump. When the next roller stem reaches the suction side, it creates another temporary seal shown in the middle figure below. As this seal moves along in a clockwise direction, it pushes the fluids in front of the seal and onto the discharge line. Combining these suction and discharge principles results in a powerful self-priming positive displacement action [14].

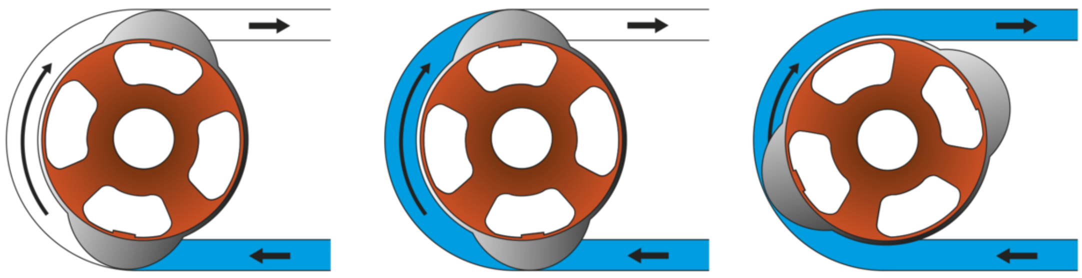


Figure 3.6: Functionality of a peristaltic pump with two rollers [14].

3.4.1 Ismatec Reglo ICC pump

Ismatec's Reglo ICC peristaltic pumps are used as one of two pump types. This pump type is the smallest of the two and used to control the flow cell loop, wasting, and sampling in each reactor [15]. The Ismatec Reglo pump allows for four separate channels for four different operations. Each channel has eight rollers for better accuracy in the peristaltic pump, delivering a flow rate between 0.0002-35 mL/min. The Ismatec Reglo pump use MS/CA Click-n-Go cassettes, which offer quick changeovers of different tube diameters.

Figure 3.7: Ismatec Reglo ICC peristaltic pump [16].

Communication

The Ismatec Reglo ICC serial command protocol supports both USB and RS-232 to communicate with the pump. The pump uses a baud rate of 9600, 8 data bits, 1 stop bit, no parity, and no flow control serial communication.

The Reglo Digital model gives two different options to communicate with multiple pumps from a controlling device (PC), either with an RS-232 communication bus where the pumps are "daisy-chained" or by using a point-to-point link with each pump through USB. By using USB communication, it allows controlling each pump independently. As it is desired to communicate with each pump individually during operations, thus the USB communication is used to communicate with the Reglo ICC pumps. The USB connection makes a virtual COM port from the controlling device and communicates with the same commands as if the pump were connected with RS-232.

3.4.2 Heidolph Pumpdrive 5201

Heidolph's Pumpdrive 5201 is used for both filling and emptying the reactors. The Heidolph is also a peristaltic pump with the opportunity of multiple individual cassettes for one single pump but without individual control. Meaning all individual cassettes will have the same direction of rotation with the same speed. Because of this, only one channel will be used.

The Heidolph Pumpdrive 5201 offers flow rates from 0.38 to 813 ml/min for its single-channel pumps, which are a lot more than Ismatec's Reglo pump can offer. In both the fill and decant stages of the SBR, larger volumes are to be transferred; therefore, the Pumpdrive 5201 is a better option.

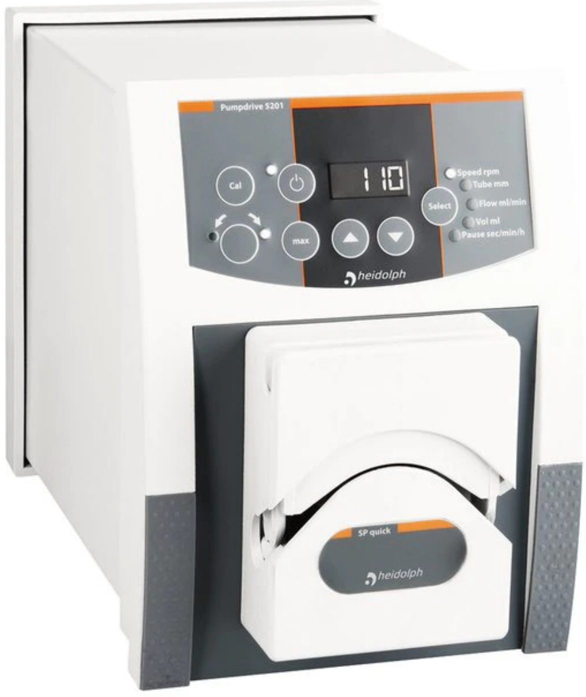


Figure 3.8: The Heidolph Pumpdrive 5201 [17].

Communication interface

The Heidolph Pumpdrive 5201 pump supports both digital and analog interfaces, with RS-232 as the digital communication protocol. Since it is desired to use digital communication, the RS-232 communication protocol is used. The Heidolph pump uses the following characteristics for the RS-232 interface; 9600 baud, no parity, 8 data bit, 1 stop bit, and no flow control.

3.5 Weight scale and magnetic stirrer

Two types of magnetic stirrers are used to keep the microorganisms from settling in the reactors and flow cells. The magnetic mixers rotate the stirrers placed inside the reactors and flow cells by a magnetic field. It is important when mixing that the speed is not so high that the stirrer destroys any microorganism flocs made in the reaction.

The magnetic stirrers used for the reactors also includes a weight scale function. This weight scale function is used to measure the weight/volume of the liquid when in the process stages: filling, wasting, and decanting. Controlling these stages with a measured weight will result in a more accurate inflow and outflow than if the pumps are controlled by the pumped volume, as the lab users did before.

3.5.1 IKA RET control-visc

For stirring, heating, and weighting functions, IKA's RET control-visc is used [18]. This allows for all these operations in one single device for each reactor setup. The RET control-visc magnetic stirrer has a speed range of 50-1700 rpm with an accuracy of 10 rpm. The magnetic stirring motor can supply a motor rating output of 9 W. It also comes with a torque trend measurement that can indicate changes in viscosity for the reactor fluid. The torque trend only shows relative changes for the viscosity and is not designed to measure the fluids' absolute viscosity [18].

By default, the heating function uses the temperature of the hot plate to regulate the temperature inside the reactor. If more precision is desired, the RET control-visc comes with an external temperature probe that can be connected and placed directly into the reactor medium. It supports the use of both PT 100 and PT 1000 temperature sensors. If an external temperature sensor is connected, this value will be the controlling value instead of the hot plate temperature. Even though the temperature setting range is between 0 - 340 °C, the heating temperature range is between; (room temperature + device self-heating) - 340 °C [18]. This is because the RET can only supply heating and not cooling. The room temperature needs to change to achieve lower temperatures. Heating is not implemented in this project but can be added on in the future.

The internal weight scale gives the user the possibility to perform simple weighing tasks, such as filling and emptying. The RET control-visc has a weighting range between 10-5000 g with an accuracy of $\pm(0.3\% + 2)$ g. In addition to these functions, the RET control-visc may also measure pH if desired. This can be done by connecting an external pH probe.

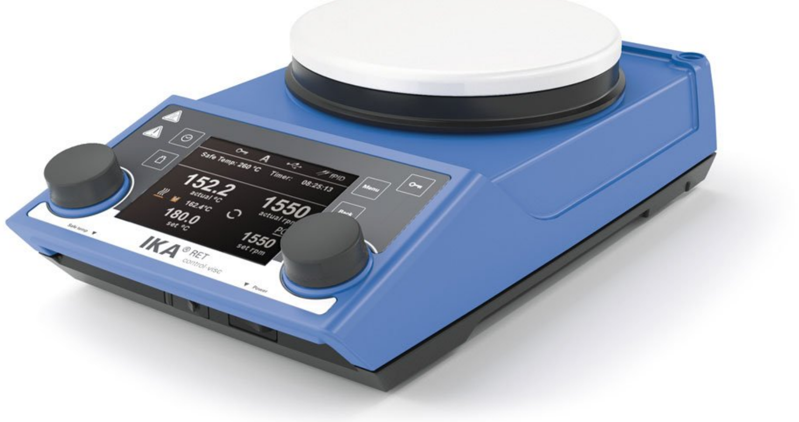


Figure 3.9: IKA RET control-visc [18].

Communication

The RET control-visc supports communication over Bluetooth, RS-232, and USB. By using the USB interface, the controlling computer sets up a virtual COM port to communicate through. The configuration, command syntax, and commands of the virtual COM port are the same as if the RS-232 interface is used.

The RET control-visc uses a transmission speed of 9600 bit/s, 1 start bit, 7 character bits, 1 parity bit (even), and 1 stop bit. The transmission type is full-duplex, and data flow control is hardware handshake RTS/CTS [18].

3.5.2 IKA lab disc

The IKA lab disc is a compact, more straightforward magnetic mixer than the IKA RET control-visc. The lab disc have a speed range between 15 - 1500 rpm, where the disc can reverse the direction of rotation automatically every 30 seconds to ensure a better mixing [19]. The operation voltage is 100 - 240 V AC. An image of the mixer is shown in figure 3.10.

Due to the mixers' late arrival in the system, they are only manually controlled. The user can turn off the mixers when not using the reactor system; otherwise, they spin all the time.

Figure 3.10: IKA lab disc [19].

3.6 Solenoid valve

If using four reactor setups, eight IN and OUT pumps (Heidolph pumps from section 3.4.2) are needed. Eight solenoid valves are used to decrease the number of pumps needed. By using eight solenoid valves, the number of pumps is decreased to two pumps, making the setup more economical and space-saving. Figure 3.11 illustrates how the setup is with the eight solenoid valves and the two pumps for every reactor. The tubes are connected onto a manifold on both IN and OUT side, connecting every reactor to the pumps.

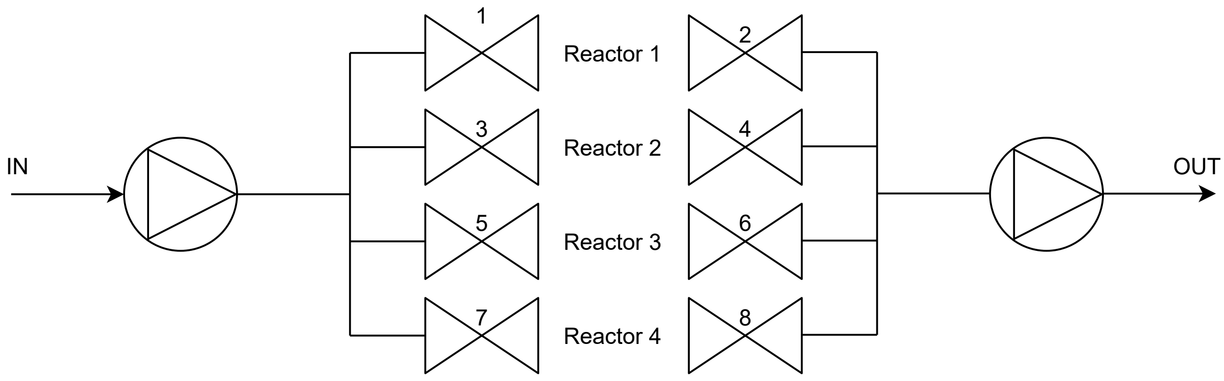


Figure 3.11: This simplified illustration shows how the IN and OUT pumps are connected through the different valves and into the reactors.

3.6.1 ASCO pinch solenoid valve, S106

The solenoid valve selected for this project is the ASCO pinched solenoid valve S106. When using a pinching valve, the valve and fluid will not be in contact since the valve pinches the tubing to close. This will protect the fluid from contaminations that might occur. The solenoid valve is a 2/2 normally closed valve, meaning it is a two-way valve with two pipe connections. When closed, it has a pinching strength of 1.1 kg, which will be enough for its purpose. The solenoid valve is suitable for soft silicone tubing, with a maximum tubing hardness of 50 shore A, supporting tubing sizes of 4.8 mm ID and 7.9 mm OD [21, 22].

Figure 3.12: Asco solenoid valve, S106 [20].

The valve needs a supply voltage of 24 V and is operated by three wires; positive voltage supply, return ground, and safety ground. The solenoid valves are controlled through power relays. A power supply of 13 W is required to open and keep the valve open [22].

Figure 3.13 illustrates the layout of the valve assembly box, with the assembled solenoid valves inside.

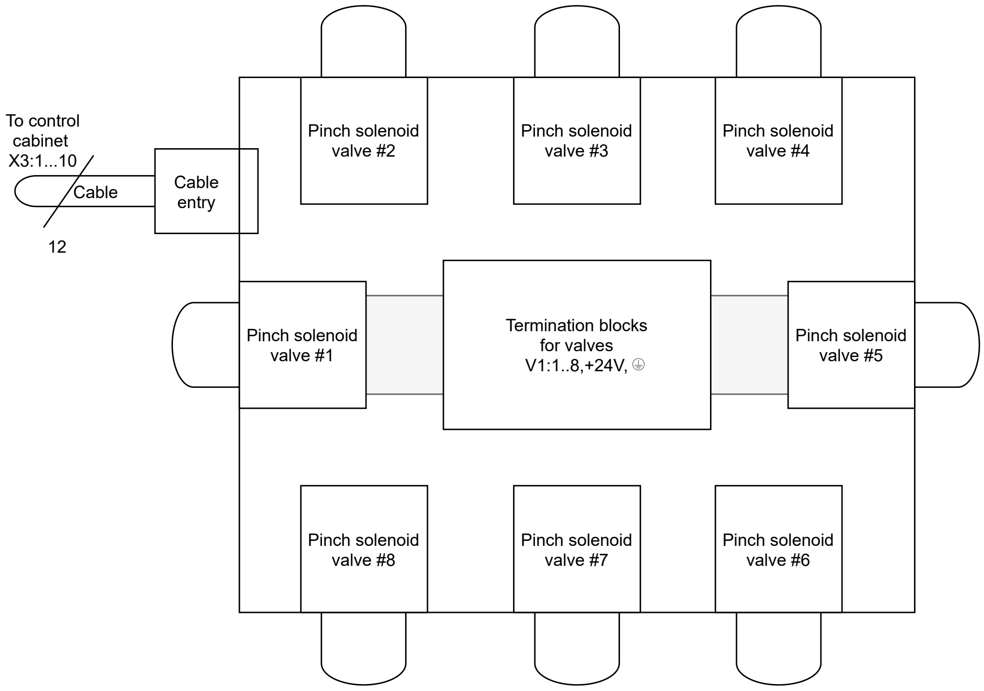


Figure 3.13: Solenoid valve assembly box layout.

Solenoid valve assembly

Every valve has a three-pin connector where three wires are attached. As mentioned, these three wires are the positive voltage supply, return ground, and safety ground.

Since these valves are placed near the reactors and pumps, there is a risk of fluid leakage. Therefore, the valves are placed into an enclosed box to ensure that no wastewater is spilled onto the electric wires and short the circuit. The valve and its corresponding housing are connected through drilled 24 mm holes. In addition, an extra hole is drilled for the wires that connect the valves to the control cabinet. Figure 3.14 show how the valves are assembled inside this enclosed box.

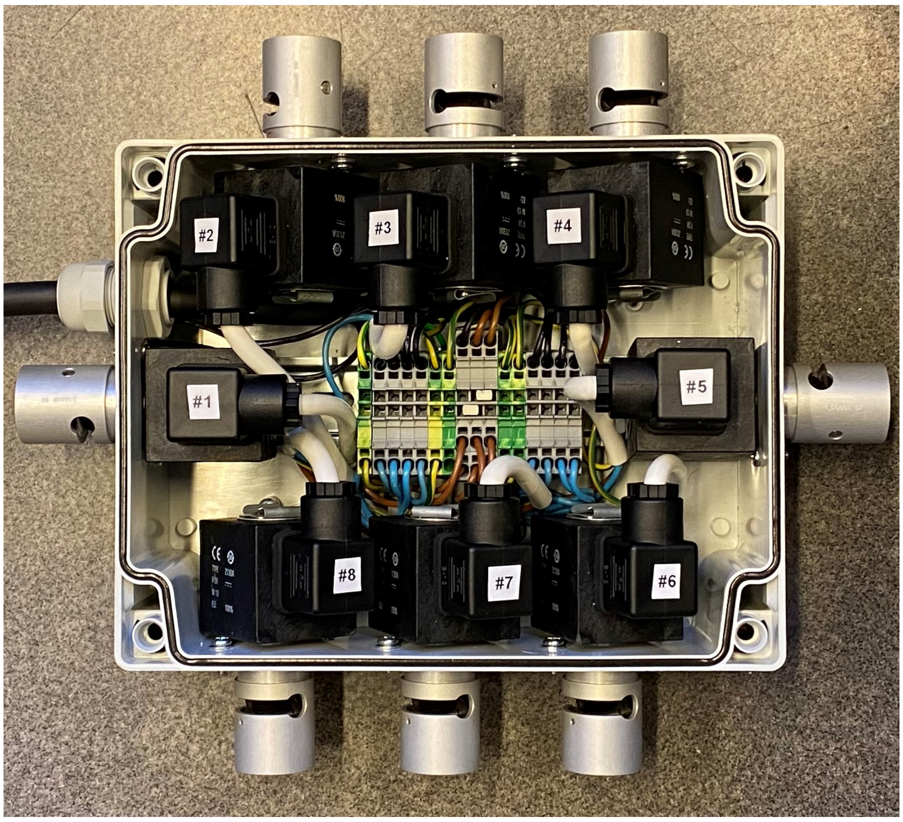


Figure 3.14: Solenoid valve assembly.

This is an external box, and since the valves are relay-controlled, a 12-wired cable is used. The assembly box contains terminal blocks, making it easy to connect everything. All eight valves use the same positive voltage supply and safety ground, while the return ground is individual for each valve. The return ground goes through the relays in the control cabinet, where each valve is controlled. Therefore, individual wires are required for the return ground.

There are two spare wires, so some changes or additions could be made in the future. A wiring diagram of the solenoid valve assembly box is illustrated in figure 3.15. The wire size is planned for a maximum of two valves that can be ON simultaneously. The terminal block inside the control cabin dedicated to the assembly box's positive power supply contains a 1.5 A glass fuse. If a software fault causes more than two valves to be open simultaneously, it will result in an overcurrent on the positive power supply wire. This could lead to the wire overheating and potentially cause a fire when they draw too much current than designed for.

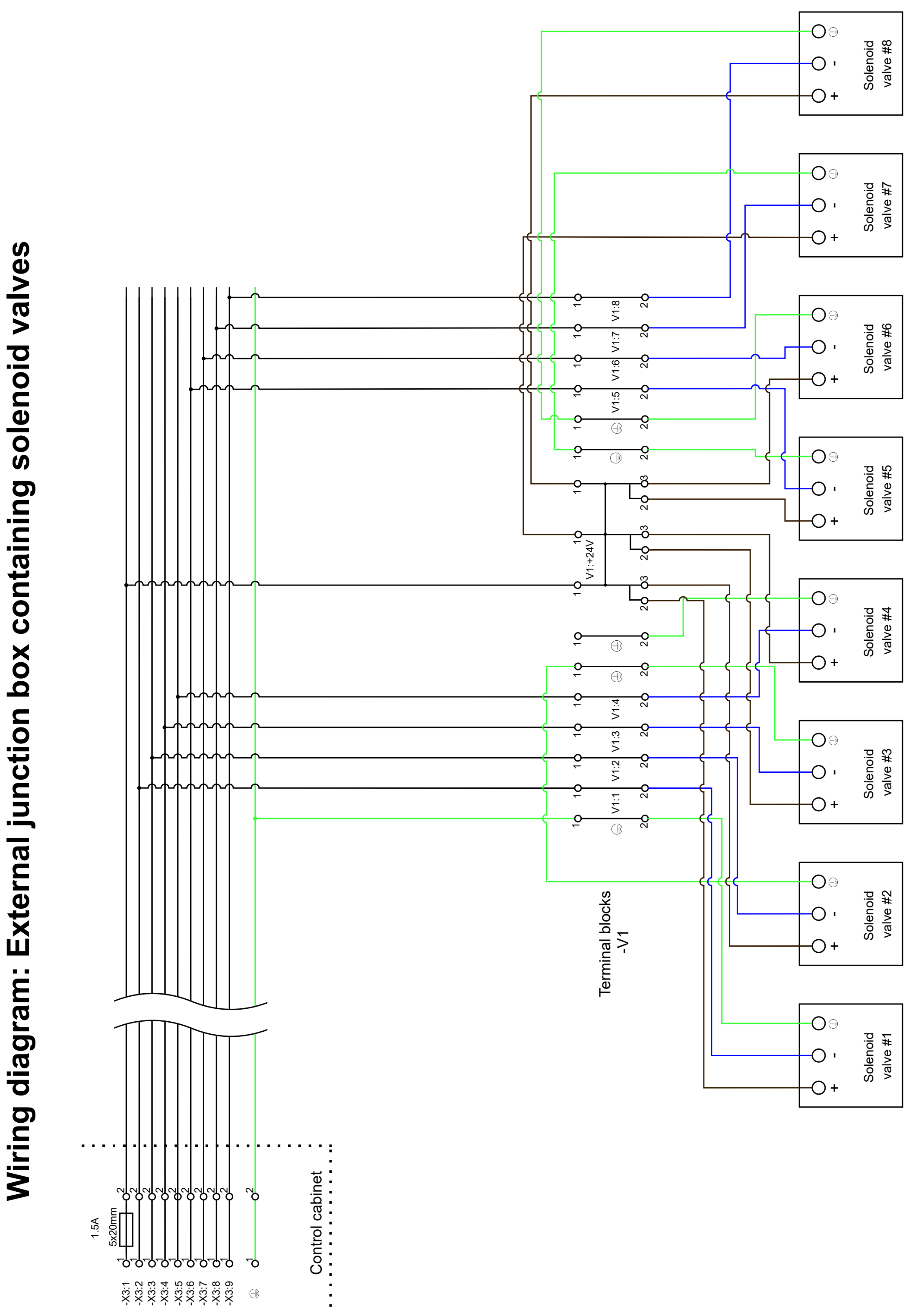


Figure 3.15: Wiring diagram of junction box

3.7 Fluid transportation and air supply

The SBR system is dependent on fluid transportation and air supply, and figure 3.16 shows these transportation lines and their connected pumps. The figure clearly shows how the Ismatec and Heidolph pumps are connected to the reactor through different tubings and the size of all connected tubings. The SBR1 setup seen in the figure is identical to SBR2, SBR3, and SBR4. As earlier stated, there are two Heidolph pumps dedicated to the filling and decanting of all reactors. The figure shows how the tubing after the filling pump is split into four before each tubing continues to its reactor. It will be the opposite for the decant lines, where the four tubings connected to each reactor's decant line are merged into one tubing before going through the Heidolph decant pump.

There are three different sizes for the fluid transportation lines, 2.79 mm, 3.17 mm, and 6 mm diameter size. The 6 mm tubes are used for the main fluid transport lines, such as filling, decanting, and fluid circulation. Using a larger tube size will give the system a higher flow rate, which is beneficial for our system. Higher flow rates will decrease the duration of both the filling and decant stages and circulate larger fluid volumes in the reaction stage.

The 2.79 mm and 3.17 mm tubes are used due to limitations in the Ismatec Reglo ICC pumps. The largest tube size that the pumps support is 3.17 mm tubes. This limitation is no problem for the sample and waste stages, as these will not transport large fluid volumes and have no time constraints. Therefore, 2.79 mm tubes are used for these two operations. However, the fluid loop will need a higher flow rate than one channel of the Ismatec can provide. Practical tests showed higher sludge build-up inside the flow cell and difficulties reaching the upper levels of DO inside the flow cell. Changing to two channels, the duration it takes to reach the upper DO level inside the flow cells was acceptable. This is why two of the channels are dedicated to fluid circulation in the flow cell loop. Here, two Y-splits are used, one before and one after the pump. The 6 mm tube is split into two 3.17 mm tubes before entering the Ismatec channels. The two 3.17 mm tubes are then merged together into the previous 6 mm tube.

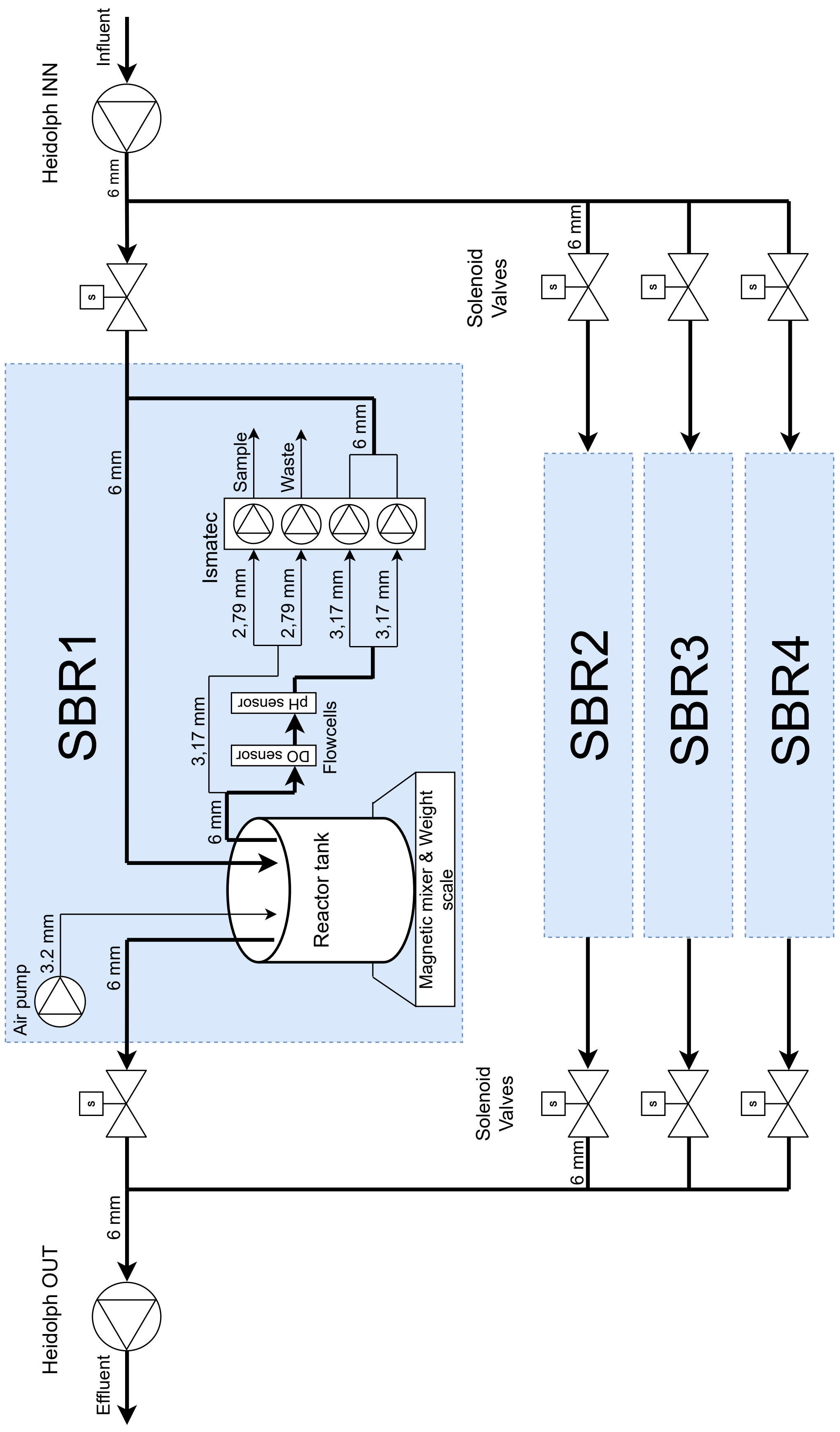


Figure 3.16: Fluid transportation for our system.

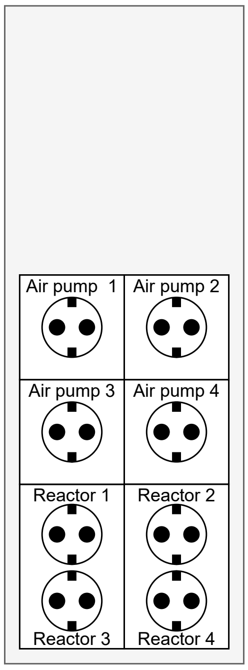
3.8 Control cabinet

The control cabinet contains equipment and components to connect everything to the computer. Figure 3.18 illustrates the electrical control panel layout in the control cabinet. A complete list of items bought in to create the control cabinet is listed in appendix C.

We are using terminal blocks for the external equipment that connects through cables, making the cabinet modular. In the future, if any equipment needs to be added or replaced, their wires can easily be inserted or removed from the terminal boxes.

There are four electrical sockets mounted on the side of the cabinet; see figure 3.17a. They operate on 230 V and supply the reactor equipment with power.
There is one socket dedicated for each reactor setup. The idea is to insert a power strip into each socket, where each power strip can handle the power consumption and has enough sockets for a complete reactor setup. By connecting every piece of equipment from one reactor to the same power source, everything turns off if any electrical error occurs. Thus, removing the possibility of some devices still operating while other devices have lost their power. The sockets dedicated to the reactor setups are shown in figure 3.17b.

The attached wires at the back of the sockets use ferrules that minimize the risk of any electrocution if someone should touch the back of the sockets by accident. The air pumps are connected separately to other sockets, which is shown in figure 3.17c. These sockets are controlled by power relays, later described in section 3.8.1.



(a) Illustration of the side of the cabinet, where the sockets are installed.

(b) ELKO S1091 PT [23].

(c) ELKO S1014 PT [24].

Figure 3.17: Illustration of the sockets mounted on the side of the cabinet and the types of electrical sockets used.

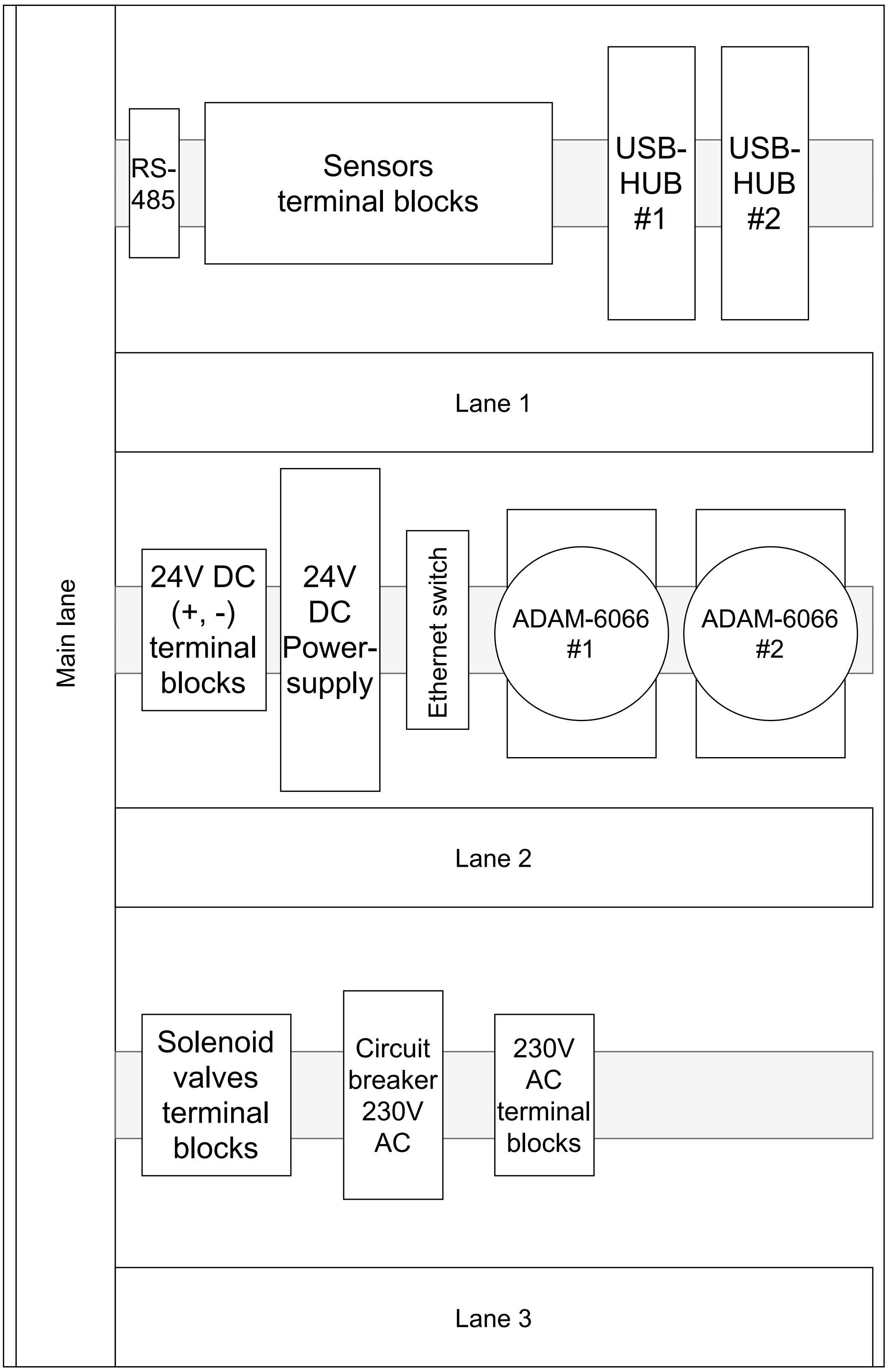
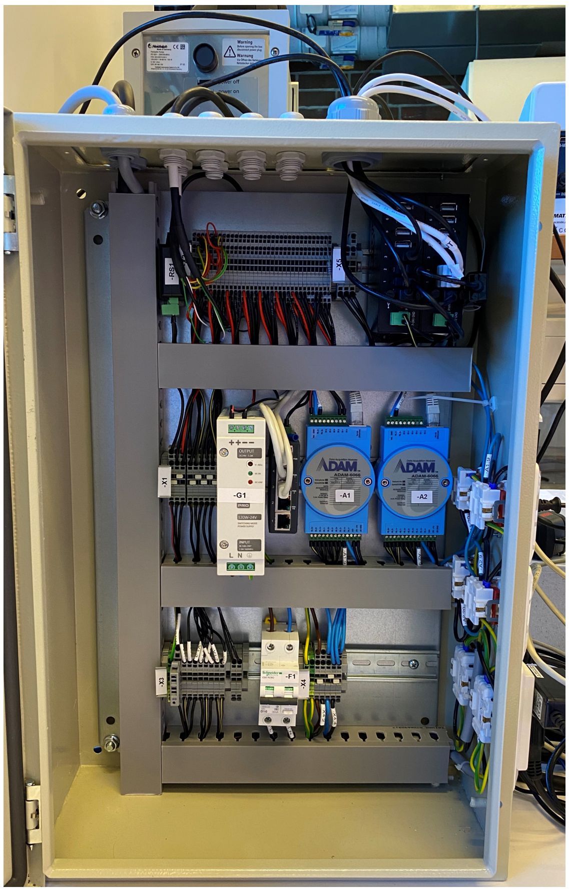


Figure 3.18: Electrical control panel layout.


(a) Inside the finished control cabinet.

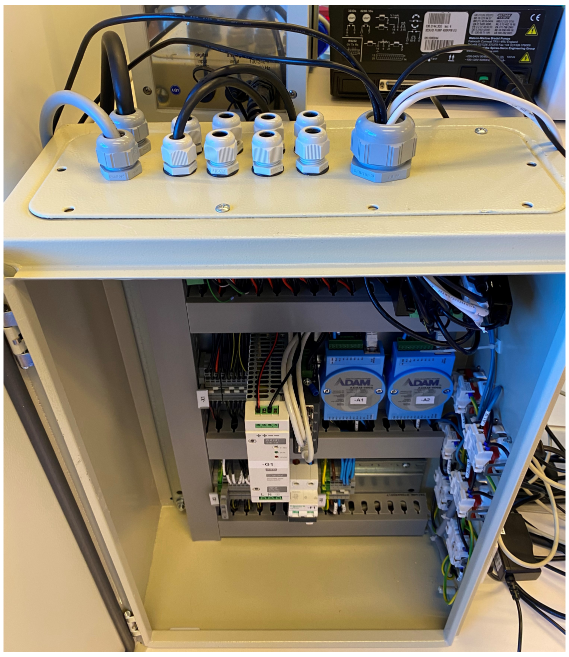

(b) Top side of the control cabinet with cable glands.

Figure 3.19: Finished control cabinet.

Figure 3.19a and figure 3.19b show the finished control cabinet with the cables going through the cable glands at the top of the cabinet.

Any modifications to the cabinet should be done by persons who know electrical systems so that no harm occurs. Safety is important, and the cabinet is made to be safe for any electrocution under normal operation conditions.

3.8.1 I/O module & Ethernet switch

To be able to control the valves, air pumps and monitor for leakage, two I/O modules are used. The air pumps and valves can easily be controlled by an ON/OFF method with available relay outputs. The relay outputs support the operation of both 230 V AC to the air pumps and 24 V DC to the valves. For the flood sensors, two digital inputs are used.

ADAM-6066 module

The ADAM-6066 module consists of six digital inputs and six relay outputs. The module communicates with the computer through Ethernet and uses the MODBUS TCP messaging protocol, as later described in section 4.3.2. The contact rating of the module's relay outputs is 250 V AC @ 5 A, and 30 V DC @ 3 A [26], which are well above our operation area. The module requires a 10-30V DC power input to run.

Figure 3.20: Image of ADAM-6066 power relay [25].

Each reactor setup requires three relay outputs (one for the air pump, two for the corresponding valves), which is a total of 12 relay outputs for the four reactor setups. Thus two ADAM-6066 modules are used. The first ADAM-6066 module cover the first and second reactor setups, and the second ADAM-6066 module covers the third and fourth reactor setup. The flood sensors are connected to the digital inputs on the first ADAM-6066 module.

The relays operate the valves and air pumps by ON/OFF control. Because of the contact rating of the module, they are suitable for both the 230 V AC operated air pumps and 24 V DC operated valves.

EKI-2525LI Ethernet switch

To connect the two ADAM-6066 modules with the connected computer, the industrial EKI-2525LI Ethernet switch is used. The switch connects the two ADAM-6066 modules to the same network, using two Ethernet cables. An additional Ethernet and a USB-A to Ethernet adapter are used to connect the switch to the computer.

The switch requires a power input of 12-48 V DC and will use the 24 V DC same power supply as the rest of the components. The EKI-2525LI Ethernet switch comes with a DIN-Rail connection, making the switch easy to mount within the control cabinet.

Figure 3.21: Industrial Ethernet switch, EKI-2525LI [27].

Wiring of the I/O module

Figures 3.22 and 3.23 shows how the solenoid valves, the air pumps, and the flood sensors are connected to the two ADAM-6066 modules. As previously stated, only the first module is connected to the flood sensors.

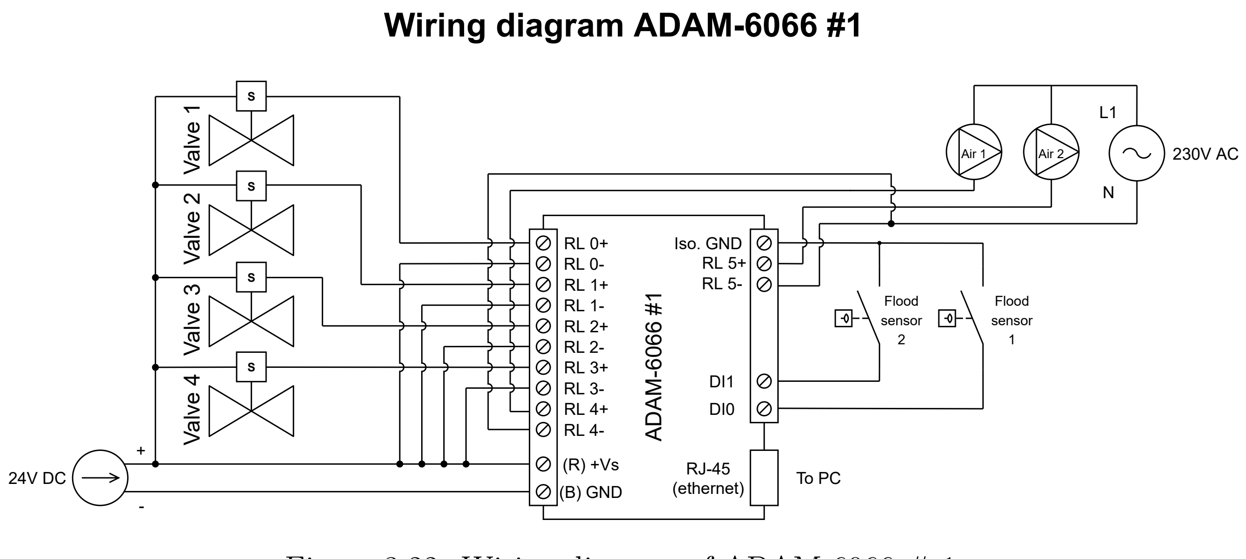


Figure 3.22: Wiring diagram of ADAM-6066 # 1.

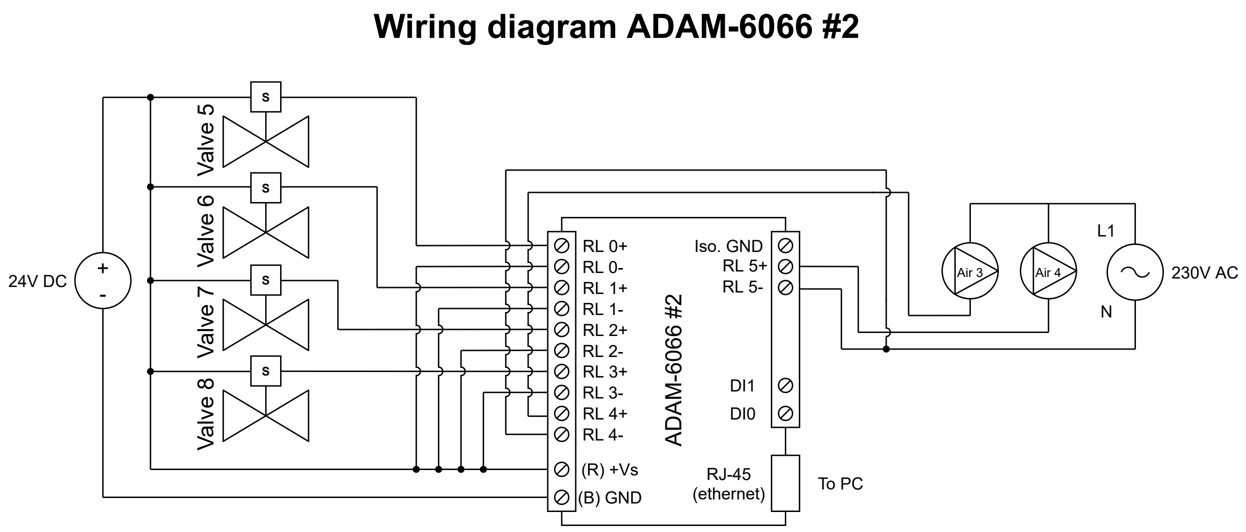


Figure 3.23: Wiring diagram of ADAM-6066 # 2.

3.8.2 USB hub

Most of the components mentioned in this chapter communicate with the controlling computer through USB-A ports. Those components that do not communicate through USB directly use adapters/converters that allow for a USB communication interface from the controlling device. With USB hubs, the number of cables between the control cabinet and the computer is minimized.

Figure 3.24: Exsys Industrial USB hub [28]

Two Exsys Industrial USB-hubs with 7 USB-A 2.0 sockets each, shown in section 3.8.2, are used to meet the required amount of USB-A ports [28]. These USB hubs come with a DIN-Rail kit, which allows mounting these inside the control cabinet. Two of these hubs will give a total of 14 USB-A ports. Each of the USB hubs comes with an external power supply option of 7-24 V DC and can supply a data transfer rate of up to 480 Mbit/s. The table 3.1 gives an overview of the required USB ports used by the devices.

Amount of USB ports	Device	Connection
4	Ismatec Reglo ICC Pumps	Connection directly from the device.
4	Ika Ret control-visc	Connection directly from the device.
2	Heidolph pumps	Connected through the USB to RS-232 converter.
1	Hamilton Sensors	Connected through the USB to RS-485 converter.
1	Ethernet Switch	Connected through the USB to Ethernet adapter

Table 3.1: Overview of the number of USB ports needed

3.8.3 Power supply

By now, we have discussed different equipment that requires a 24 V DC power supply to operate. Finding the total current drawn from the equipment makes it possible to find a suitable power supply. See appendix D for the calculations. When choosing the power supply, the ripple voltage is something to take into account. The power supply used is sufficient enough and complies with the ripple voltage limitations of the equipment.

The needed power supply must be able to output voltage of 24 V DC and output current of 3.07 A (73.6 W). The power supply chosen is a 230 V AC to 24 V DC with an output current of 5 A (120 W) and a ripple voltage of 120 mV.

RS PRO DIN rail power supply

The RS PRO DIN rail power supply is a switched-mode power supply from RS PRO's range of high-performance power supplies. It is DIN-rail mounted and compact form; it is easy to mount it inside the control cabinet. The power supply accepts single-phase AC input for conversion into a single DC output. As mentioned above, the output voltage is 24 V DC with an output current of 5 A; thus, the model number is RS stock #136-8319. The input voltage is universal, e.g., voltage range from 90 to 264 V AC. Figure 3.25 displays the chosen power supply.

Figure 3.25: RS PRO 24 V 5 A power supply [29].

3.8.4 Circuit breaker

The control cabinet is equipped with a circuit breaker on the main intake with a rated operational voltage (Ue) of 230 V AC. The outlets are rated 16 A (In), and anyone in the lab (non-expert) could use them. Thus a residual-current device (RCD) with an earth-leakage sensitivity of 30 mA is required to maintain the safety of any users operating the system. A mandate in the Norwegian rules of low voltage electrical installations (NEK 400-4-41 411.3.3) states this. A leakage voltage over 50 V AC or 120 V DC could be deadly if anyone gets shocked. The RCD prevents if there is any earth leakage in the electrical system.

Figure 3.26: Schneider iC60 RCBO [30].

We find the suiting circuit breaker by calculating the maximum total current of every piece of equipment that uses 230 V AC to operate. (See calculations in the appendix D). This results in a 16 A residual current breaker with overcurrent (RCBO), e.g., circuit breaker combined with an RCD. Because the system does not have any considerable in-rush current, a curve code B is sufficient enough for this setup. A curve code B circuit breaker trips between 3-5 times rated current in a short circuit situation [31]. The RCD is a type A with an earth-leakage sensitivity of 30 mA, which reacts on both AC and pulsating DC. The RCD is for safety precautions in case of a fault, so no one gets shocked.

The chosen residual current breaker is the *16 A Schneider iC60 RCBO* as showed in figure 3.26 [30].

3.9 Wiring

Every component in the cabinet is connected to a power source and/or another component for communication/control. The conductors in the wires are stranded; thus, they have to have ferrules crimped on them in connections for safety purposes, which also gives a more stable connection than not using ferrules [32].

The wire sizes depend on the current flowing through the wires. Thus by looking at the calculations mentioned earlier, appendix D, the appropriate wiring sizes are found by using an wire sizing calculator app, Prysmian Group Norway Cable app [33]. Table 3.2 lists the wiring sizes used at the different applications.

Application	**Wire size**
230 V Intake (External cable)	2.5 mm^2
230 V Circuit (To power supply and outlets)	2.5 mm^2
24V Power circuit (From power supply to terminal blocks)	1.5 mm^2
Hamilton sensors	0.75 mm^2
24 V DC Power circuit (from terminal blocks to components) and 2-Wire flood sensors	1 mm^2
Solenoid valves (External cable to solenoid valve box)	1.5 mm^2

Table 3.2: Wire sizes at the different applications.

The wiring sizes in the cabinet do variate because different components do consume different amounts of power.

230 V AC and 24 V DC are separated as much as possible to avoid distortion noise in the DC circuit due to the 50 Hz frequency in the AC circuit.

By using different colors and markers on the wires, it is easy to identify where it belongs. There are various colors on the 230 V AC and 24 V DC circuits described in table 3.3.

	L or +	N or -	Safety Earth	Signal
230 V AC	Brown	Blue	Green/Yellow	
24 V DC	Black/Red and solid red	Black/Yellow and solid black	Green/Yellow	Solid black

Table 3.3: Colors of wires used in the control cabinet and external solenoid valves box.

Every wire has a marking according to its terminal number on the connected component to keep a systematic overview over the connected wires.

3.9.1 Wiring schematics

The wiring is documented on four schematic wiring diagrams, two for power distribution of 230 V AC and 24 V DC, one for sensor communication, and one for the wiring of the solenoid valve assembly box. Every component has a label that aids when wiring everything together.

230 V Power distribution

This wiring diagram shows how the 230 V AC power distribution among the components that need 230 V AC to operate. As mentioned, four of the outlets mounted on the cabinet are for air pump control. Thus this diagram shows the wiring from the outlets to the power relay input and output to the neutral wire. See figure 3.27 for the wiring diagram. Since we use Schuko plugs on the inlet cable, we can not be completely sure that the L and N are not switched in the wiring diagram. Schuko plugs can be rotated and plugged in both ways.

24 V Power distribution

This wiring diagram shows how the 24 V DC power distribution and control of the solenoid valves through the power relays. The solenoid valves are connected to the terminal blocks X3:1..9 in the control cabinet. See figure 3.12 for the wiring of the separate solenoid valve box. See figure 3.28 for the wiring diagram.

Sensor wiring

The sensor wiring diagram, see figure 3.29, shows how the Hamilton sensors and the flood sensors are wired. A bus is created by jumping one set of wires from the first Hamilton sensor to the next and further on. Follow the color order shown in figure 3.29 of the Hamilton sensor when adding additional sensors to the system.

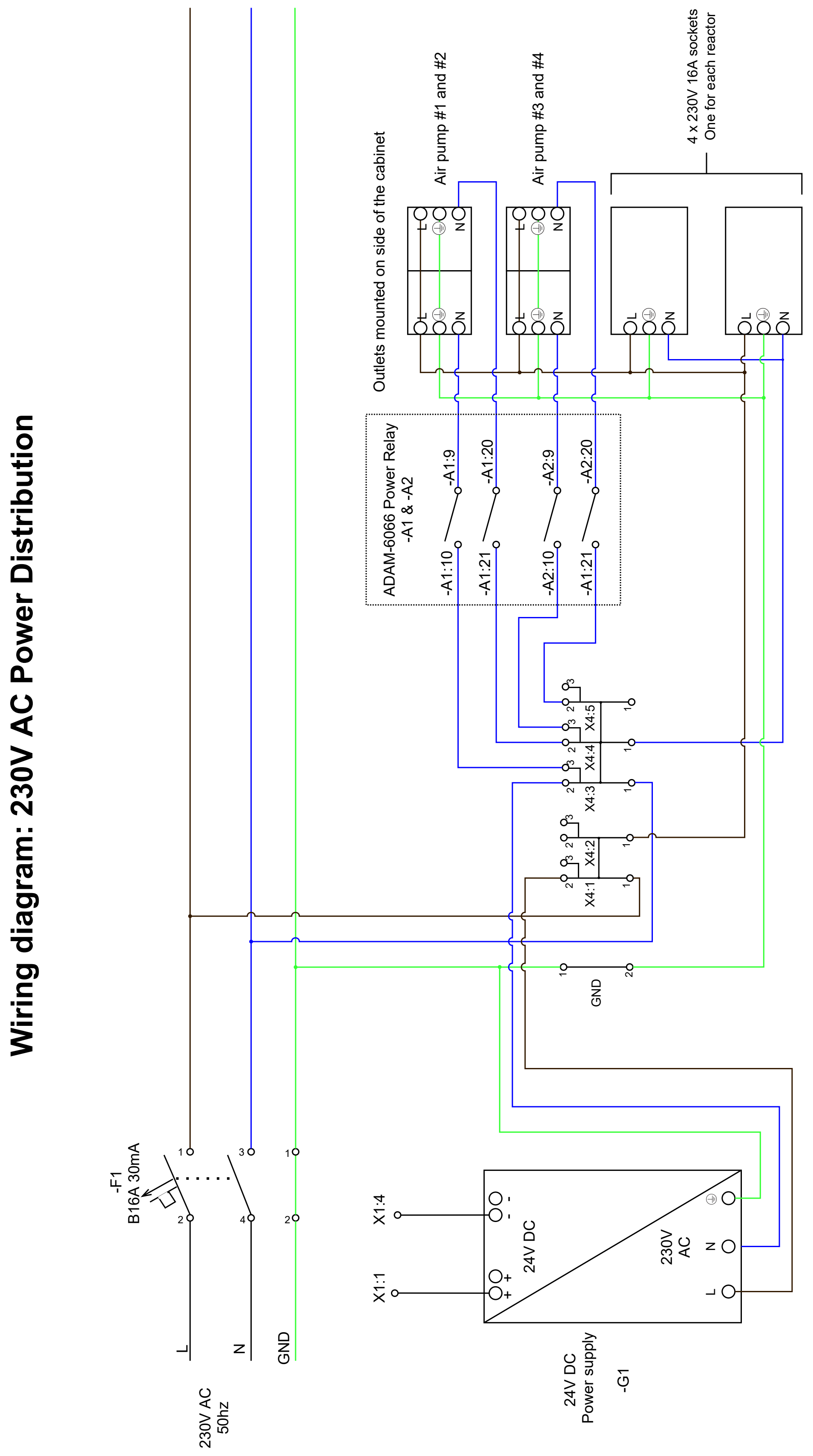


Figure 3.27: Wiring diagram of 230V AC power distribution and 230V AC power relay control.

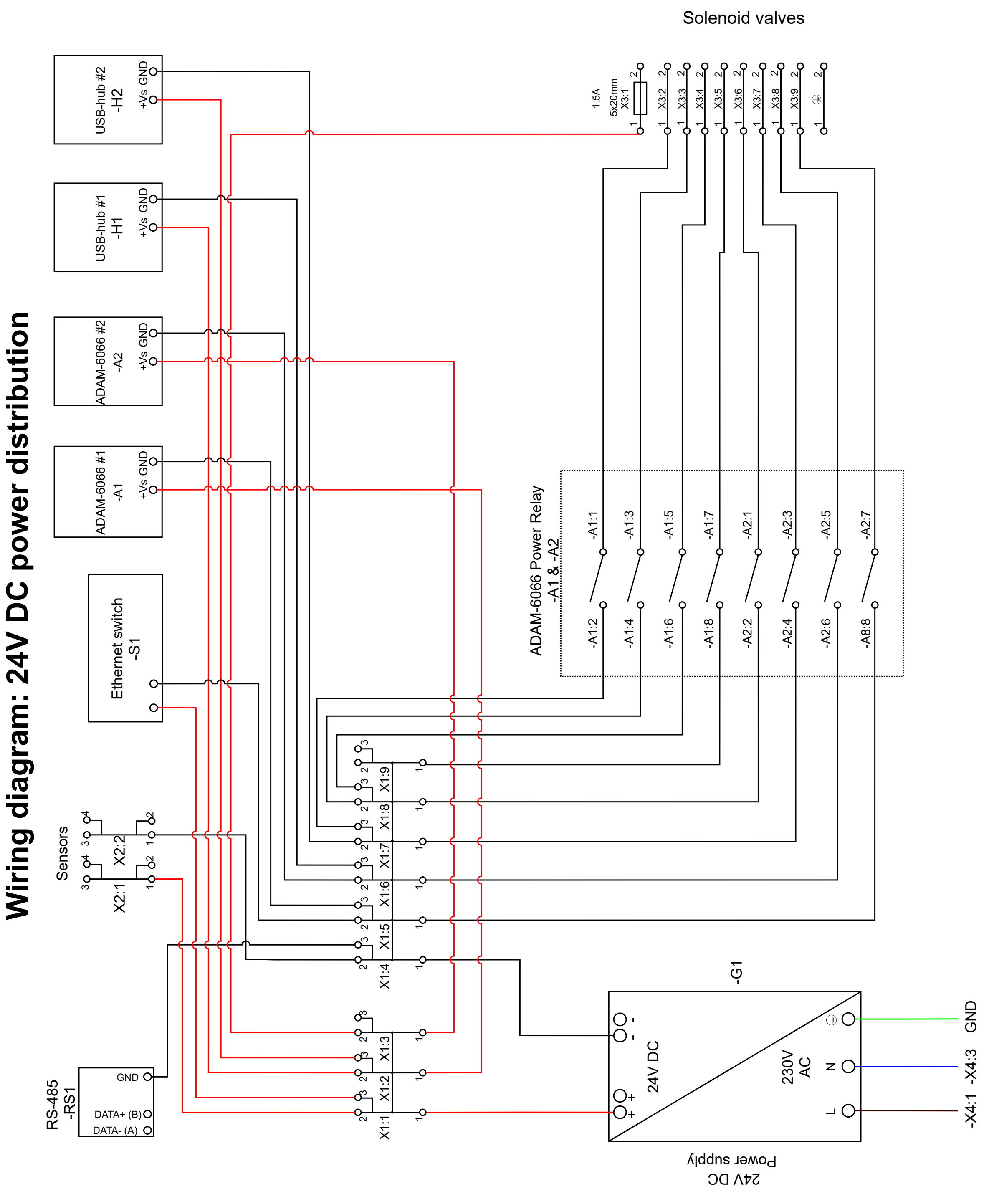


Figure 3.28: Wiring diagram of 24V DC power distribution and 24V DC power relay control.

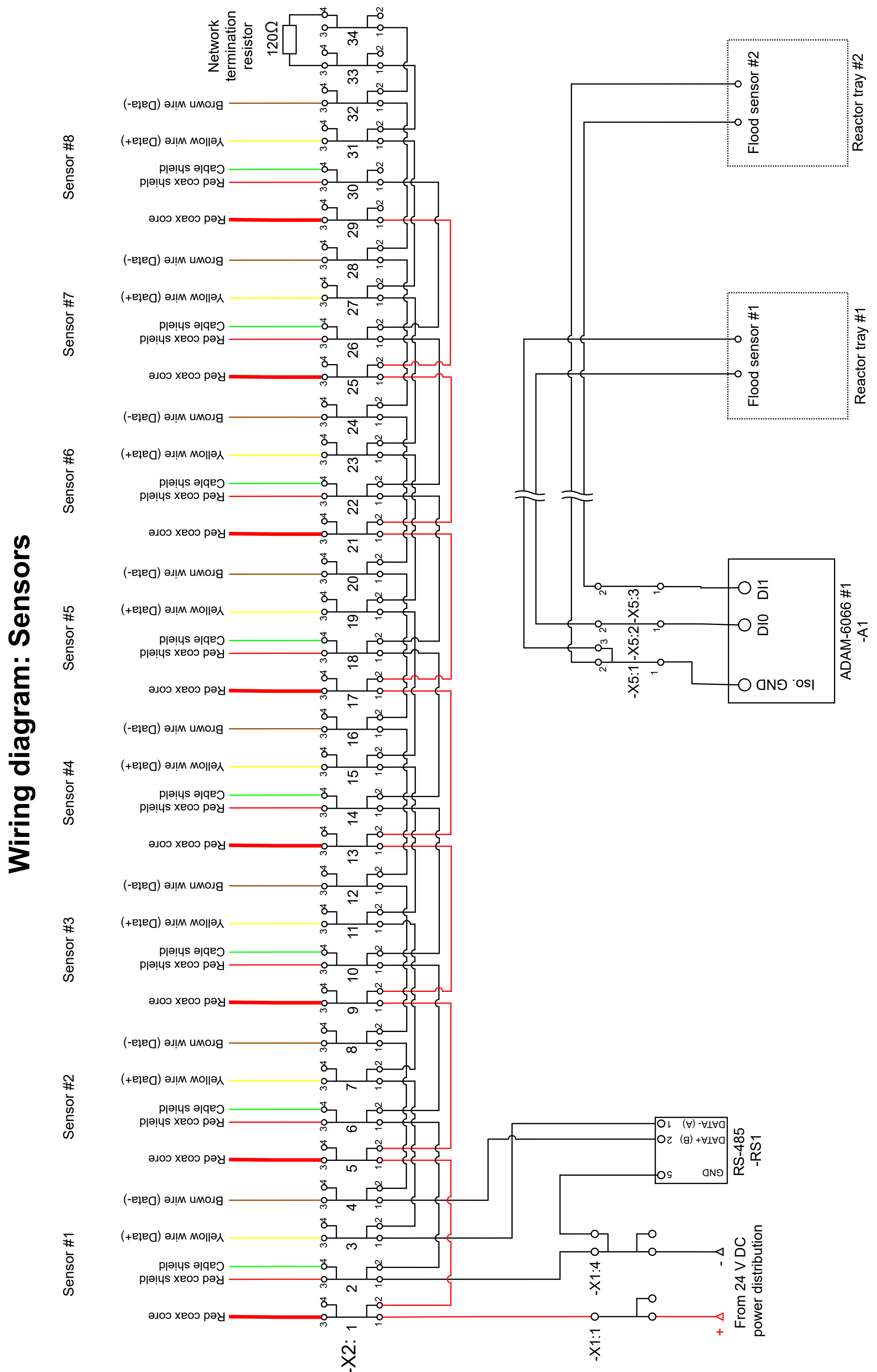


Figure 3.29: Wiring diagram of the sensors.

4. Communication Interface

This chapter will go in-depth on the communication interface between the different components in this system. All communication will go through a connected computer that operates as the master device.

The connected components use different types of digital communication interfaces. It differs in both the data transmission and the messaging protocol. The different data transmissions are RS-232, RS-485, and Ethernet, which use ASCII, MODBUS RTU, and MODBUS TCP, respectively, as their messaging protocols. These communication interfaces, or protocols, are explained in both their functionality and usage. Since the RS-232 and RS-485 are more used than Ethernet in this project, and having wired both the RS-232 and RS-485, these two protocols will be more explained. We will not go further into details about the Ethernet protocol, as we only use the MODBUS TCP/IP communication protocol. The MODBUS TCP/IP will, however, be further discussed.

Regardless of communication protocols, all devices are connected to the computer through USB-A via two USB hubs. As a result, some devices need adapters or converters to change their original assigned physical interface. Figure 4.1 shows how 12 devices are connected through the two USB hubs before two cables connect the hubs to the computer. All connected devices have their own dedicated USB socket, which will make their assigned COM port in the computer constant. Having constant COM ports will also make it easier for the control program that has been made for this project. A scenario where this will be beneficial is when the user connects new devices to the control cabinet. The program will then know what COM port the new devices have been assigned.

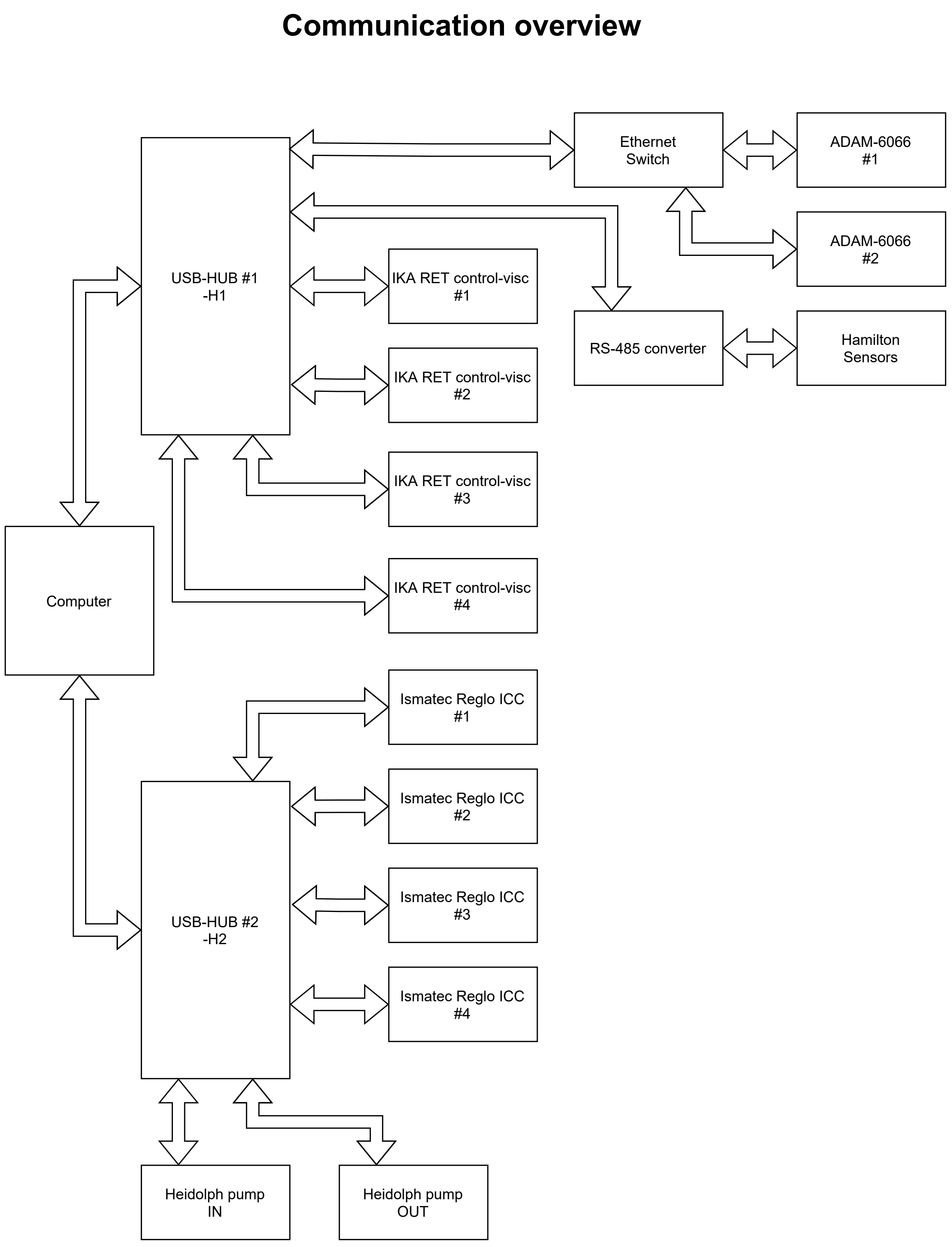


Figure 4.1: An overview on how all devices are connected to the computer.

4.1 Communication protocols

There are three data transmission schemes in use for this project, which are RS-232, RS-485, and Ethernet. All these transmission schemes build upon information and data being transferred between two or more connected devices.

Both RS-232 and RS-485 are serial communication schemes, where data transmission occurs over a single communication line, sending one bit of data at a time.

4.1.1 RS-232 Serial line

There are three types of devices in this project that use the RS-232 communication protocol, which are the Heidolph pumps, the Ismatec Reglo pumps, and the IKA RET control-visc. These support the use of either a virtual communication port via USB or through a DE-9 connector, where both Ismatec Reglo pumps and IKA RET control-visc will use a virtual communication port via USB while the Heidolph pumps only support communication through a DE-9 connector.

RS-232 is a standard protocol for serial communication used in industries and commercial products that allow communication between two devices. The RS-232 is limited to only having a one-to-one connection with a master and slave [34].

Some devices may support "daisy-chaining" multiple slaves together, allowing more devices to be connected through the same line. Being that the RS-232 originally are limited to two devices connected together, addressing several slaves through the "daisy-chain" is not possible. The commands or data sent through the "daisy-chained" devices are sent to all the connected devices, excluding separate communication with the connected slaves. This is the case with Ismatec's Reglo pumps used in this project; see section 3.4.1. Even though some devices may support the use of this "daisy-chain", it will not be used in this project, so this will not be further explained.

The RS-232 protocol defines the interface between a data terminal equipment (DTE) and a data communications equipment (DCE). An example of a DTE can be a computer, and an example of a DCE can be a modem [35].

RS-232 wiring

The RS-232 standard designates 25 lines, or wires, for the interface between the DTE and DCE. Even though the RS-232 specifies the use of a DB-25 connector, these connectors are rarely used. In modern designs, virtually no RS-232 applications need more than nine wires, leading to the use of a DE-9 connector instead of the DB-25 connector. This is also the case for the devices in this project. The unused signals, when using nine lines, are intended for use with synchronous modems, secondary transmission channels, and selecting transmission speeds on dual-rate modems [34].

There are three essential lines when communicating over RS-232. Some applications may only need these three lines (or even two, if the signals are intended for one-way transmission) for communication. These essential lines are TX, RX, and SG [34].

- **TX**: The TX line is where the data are transmitted from the DTE to the DCE.
- **RX**: The RX line is where the DTE receives the data from the DCE.
- **SG**: SG is the signal ground.

RS-232 Data transmission

For all digital communication, the transmitting data are based on a bitstream of the logic levels '0' and '1'. The RS-232 logic levels are defined as positive and negative voltages [34]. For the RS-232 protocol, the transmitting device and the receiving device support a slight difference in the voltage levels. The transmitting device uses a driven voltage between 5 V and 15 V to represent the logic '1' and a voltage between -5 V and -15 V to represent the logic '0'. In comparison, the receiving device accepts a voltage between 3 V and 15 V to represent the logic '1' and a voltage between -3 V and -15 V to represent the logic '0'. The difference in the voltage level of transmitted data and an accepted received data results in a noise margin of 2 V at the receiving end. The logic level of a received signal between -3 V and 3 V is undefined in the RS-232 protocol [36].

The RS-232 protocol supports cable lengths up to about 15 to 20 meters depending on the cable capacitance [36]. Therefore, an RS-232 receiver can be placed on the end of a relatively long cable, resulting in a received voltage that has attenuated or is affected by noise. To compensate for this, the receiver accepts smaller voltage levels than the transmitting device sends [34].

For two connected devices to be able to communicate, they need to be configured to the same characteristics, such as baud rate and message frame format. An RS-232 message format may consist of a start bit, multiple data bits, an optional parity bit, and stop bits. Figure 4.2 shows a typical message frame.

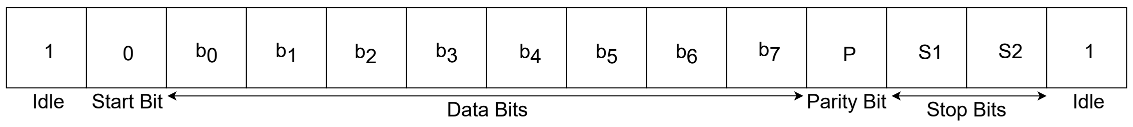


Figure 4.2: A typical RS-232 messaging frame.

When the communication is idle, i.e., there are no data transmitting between the connected lines, the logic level is constantly '1' until the transmitting device sends a start bit with the logic level of '0'.

4.1.2 RS-485 Serial line

The Hamilton sensors used in the project support communication over an RS-485 network. The RS-485 protocol is, as the RS-232 protocol, a serial data communication standard. The RS-485 communication protocol is a data-transmission scheme that offers a robust solution for transmitting data [37]. Compared to the RS-232 protocol's one-to-one connection, the RS-485 standard supports 32 'standard' transceivers connected to the same network. Alternatively, with high-impedance transceivers, the RS-485 network can support the use of 256 transceivers. When connecting multiple devices to the same network, the devices are "daisy-chained" together. Creating a bus that the master device communicates with. In addition, the RS-485 protocol allows for higher transmission speeds at longer distances than the RS-232 protocol.

Usual options for an RS-485 network are either MODBUS RTU or ASCII. For this project, the MODBUS RTU is preferred.

RS-485 Wiring

The interface bus may be designed for full-duplex or half-duplex data transmissions, where full-duplex transmissions require a "four-wire" setup (two signal pairs) and half-duplex transmissions only need a "two-wire" setup (one signal pair) [38]. Since the Hamilton Sensors used in this project, from section 3.2.1, only support a "two-wire" network, the report will not go further in-depth on the "four-wire" network.

The "two-wire" RS-485 network requires two signal wires and a common ground to function. The two signal wires are usually referred to as A and B. During data transmission, differential voltages between the A and B outputs determine the logic state of the data. To avoid signal reflections along the data lines, 120 Ω resistors are be placed across the differential lines at both ends.

Data transmission

The RS-485 data transmission is, as for the RS-232 protocol and all other digital communications, based on a bitstream of the logic levels '0' and '1'. These logic levels are determined by the differential voltages between the A and B wire (A - B). During transmission, both voltage levels are referenced to the common ground.

When the transmitting device sends data, the differential output of the two wires is limited to a minimum of 1.5 V with a maximum differential voltage of ± 6 V [34]. A differential output between -1.5 V and -6 V represents the logic level '1', and a differential output between +1.5 V and +6 V represents the logic level '0'. For the receiving device, however, it detects differential inputs down to 200 mV. A detected differential voltage of -200 mV or lower represents the logic '1', and a differential voltage of +200 mV or higher represents the logic '0' [34]. The difference in the allowed transmitted outputs and received inputs add up to a noise margin of 1.3 V, a sufficient margin for reliable data transmission even under severe signal degradations [39]. The RS-485 differential signals already cancel most noise [34].

4.2 ASCII

The devices that use RS-232 in this project communicate through ASCII, which is the preferred messaging protocol when using the RS-232 protocol. Other transmission protocols, like RS-485, may also use ASCII if desired.

ASCII is a standard table for digital representation of upper- and lowercase Roman letters, numbers, and special control characters, where each character has its own unique binary string. The ASCII standard is originally used seven bits to represent all characters in the table, adding up the total of 128-character set to choose from. Even though the official ASCII standard only uses seven bits, most computers use a full byte(eight bits) when sending an ASCII character. With this extra bit, many companies have made their own nonstandard extension of the ASCII standard, doubling the number of character sets to chose from. From the original 128-character set into a 256-character set [40].

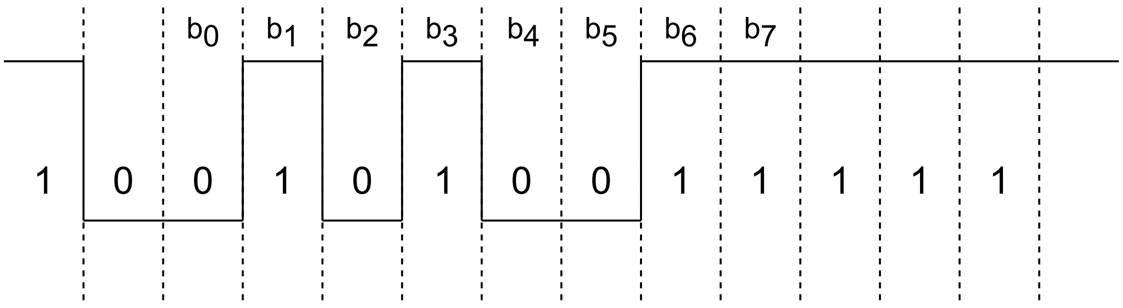


Figure 4.3: The ASCII character S.

Figure 4.3 shows how the ASCII character 'S' is represented in an RS-232 messaging frame. The bits b_0 to b_7 add up to the following binary string: 0101 0011.

4.3 MODBUS

This project uses two MODBUS protocols: the MODBUS RTU and the MODBUS TCP/IP protocols. The Hamilton sensors use the MODBUS RTU, while the ADAM-6066 modules use the MODBUS TCP/IP.

MODBUS is an application messaging protocol that provides master/slave communication between devices on different types of buses or networks. The MODBUS RTU is based on serial transmission, while the TCP/IP uses Ethernet.

4.3.1 MODBUS RTU

The MODBUS RTU protocol is a Master-Slave protocol, where only one master device at a time is connected to the bus. This protocol allows for up to a maximum of 247 slave devices to be connected to the bus [41]. Communication is always between the master device and a slave device. The slave nodes cannot communicate with each other. When communication happens between the master node and a slave node, the master device must initiate the communication. The initiation can only start one MODBUS transaction at a time. This MODBUS transaction consists of two messages: a request from the master device and a reply from the slave device. The request message is sent to the slave device with its unique slave address. Each slave must have an individual slave address to be addressed independently from the other slave devices.

The initial state of the slave devices connected to the bus is idle, meaning there are no pending requests. All slave devices stay idle until a pending request occurs from the master device. Then the slave checks the message information before it performs the action requested in the message. If the slave device detects no error in the received frame, a reply must be sent back to the master. This reply may be that the required action is completed or an error has occurred. If the slave device detects an error in the received frame, no reply is returned to the master.

MODBUS command structure

The MODBUS application protocol defines a simple Protocol Data Unit(PDU) independent of the underlying communication layers [42]. The MODBUS PDU consists of a function code field and then a data field that is transmitted. When mapping the MODBUS protocol on some specific buses or networks, in this case to the RS-485 protocol, it introduces some additional fields to the PDU and forms the Application Data Unit(ADU) [42]. These additional fields are the address field and Cyclical Redundancy Check(CRC). An example of the difference between PDU and ADU is shown in the image below.

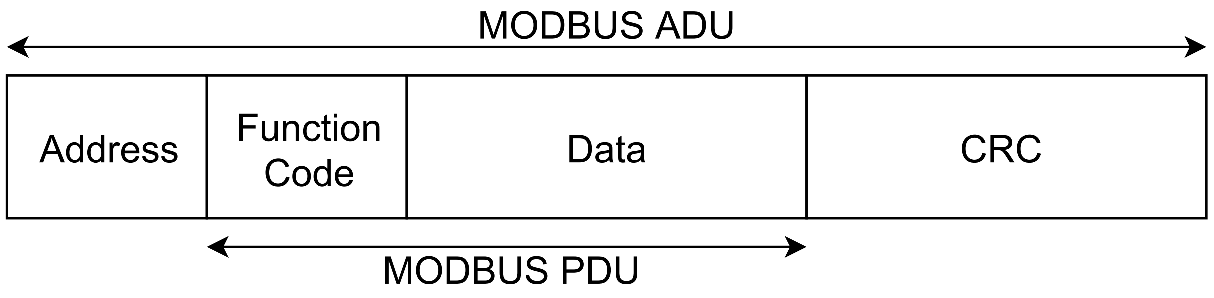


Figure 4.4: A MODBUS ADU command.

- **Address field**: The address field contains only the slave address. If the message is sent from the MODBUS master, it contains which slave it wants to communicate with. If the message is sent from a slave unit, it contains which slave address the message comes from.
- **Function code**: The function code indicates to the server what action to perform.
- **Data**: The data field contains the request or response parameters if there are any.
- **Cyclical Redundancy Checking (CRC)**: This field is the result of a "Redundancy Checking" calculation that is performed on the message contents.

MODBUS Data transmission

The MODBUS Serial Line protocol has two different options for serial transmissions; the RTU mode and the ASCII mode. [41] These modes determine how information is packed into the message fields and then decoded by the receiving device. This transmission mode must be the same for all devices connected to the bus. The Hamilton sensors used in this project communicate on the MODBUS serial line using the Remote Terminal Unit(RTU) mode. Here, each 8-bit byte in a message contains two 4-bit hexadecimal characters. The main advantage of using RTU mode instead of ASCII mode is better data throughput for the same baud rate due to a greater character density for the RTU mode.

The total MODBUS RTU message frame has a maximum size of 256 bytes. Four of these bytes are always used by the address-, function code- and CRC fields, where both the slave address and function code fields use 1 byte and the CRC field uses 2 bytes. The rest, 252 bytes, contains the data. The data field may use anywhere between 0 up to 252 byte(s), depending on which data is being transmitted. Figure 4.5 shows how a successful RTU message frame may look like.

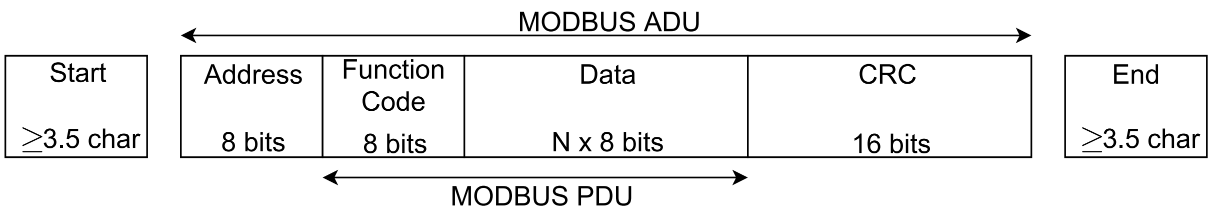


Figure 4.5: RTU message frame.

Both the transmitting and receiving devices need to have the same known beginning and ending point to separate the message frames. This is shown in figure 4.5, as the two blocks "Start" and "End". The message frames are separated by a silent interval of at least 3.5 character times in RTU mode, which is shown in both the "Start" and "End" boxes from figure 4.5. As frames are being transmitted over the serial line, the interval separating the frames are limited to a minimum length. Figure 4.6 show three valid frames sent after one another. The silent interval lengths are only limited by a minimum length, and, as the figure shows, frames separated with a silent interval of 4.5 characters are also valid.

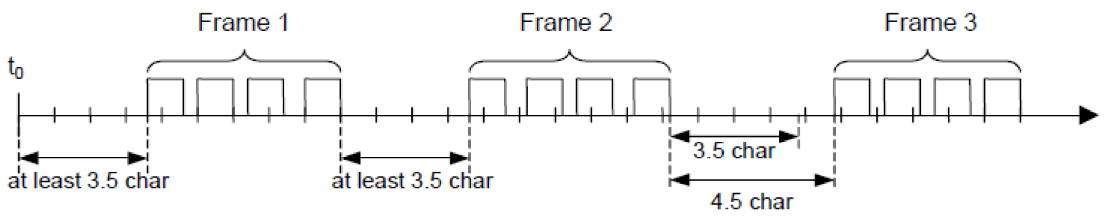


Figure 4.6: Successful frames with silent interval [41].

In figure 4.6, all three frames are considered complete. However, the frames also have their boundaries to work within. As the message frames are split by a silent interval, the characters within the frame are also split by an interval. The difference is that while a minimum interval limits the interval between frames, a maximum interval limits the interval between characters. The maximum silent interval within a message frame is 1.5 characters. The entire message frame must be transmitted as a continuous stream of characters. If not, the message will be discarded by the receiver. If the receiver reads a silent interval of more than 1.5 characters times between two characters, the message will be discarded as the message is declared incomplete. Figure 4.7 shows both a successful frame and an unsuccessful frame. The frame to the left, frame 1, does not have any silent interval longer than 1.5 characters and is considered a successful message. Frame 2, on the other hand, has a silent interval longer than 1.5 and is therefore considered unsuccessful and is discarded.

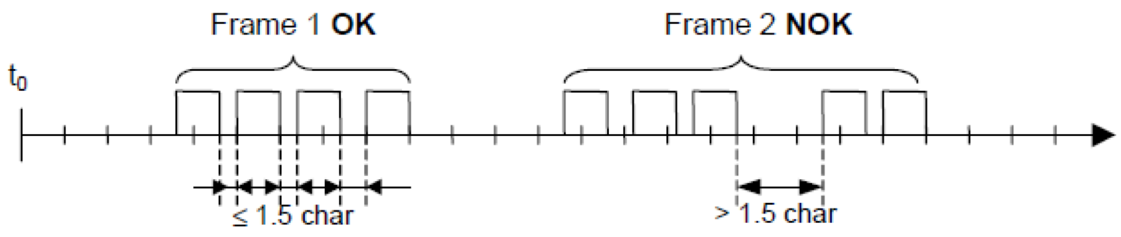


Figure 4.7: One successful and one unsuccessful data transmission of a message frame [41].

Function codes

The Arc Sensors uses only a subset of function codes from the MODBUS protocol. This subset uses the following three function codes from MODBUS: #3 Read Holding Registers, #4 Read Input Registers, and #16 Write Multiple Registers [43].

- **#3 Read Holding Registers**: This function code is used to read the contents of a contiguous block of holding registers. The requesting message specifies the starting address of the desired register and the number of registers. In the PDU, register addresses start at zero resulting in addresses 1-16 are instead addressed as 0-15, continuing so throughout the number of register addresses. The response message packs the register data as two bytes per register. The first byte contains the high order bits, and the second byte contains the low order bits [43].

- **#4 Read Input Registers**: This function code reads from 1 - 125 contiguous input registers in a remote device. The requesting message specifies the starting address of the desired register and the number of registers. In the PDU, register addresses start at zero resulting in addresses 1-16 are instead addressed as 0-15. Continuing so throughout the number of register addresses. The response message packs the register data as two bytes per register. The first byte contains the high order bits, and the second byte contains the low order bits [43].

- **#16 Write Multiple Registers**: This function code is used to write a block of contiguous registers to a remote device. The requested written values are then specified in the request data field. The data is packed as two bytes per register. If there are no errors, the response message returns the slave address, function code, starting address, and the quantity of registers written [43].

MODBUS RTU Error Messages

The MODBUS standard comes with some predefined exception codes, or error codes, for some particular cases. These codes occur if the slave receives the message without a communication error but cannot handle it. This might be a request to read a non-existent register, for example. This exception code will then be sent back to the client, informing it of the error that just occurred and its nature [42]. While the MODBUS standard has nine predefined exception codes, the Hamilton Arc Sensors only use the following four [43].

- **(0x01) - Illegal function**: The function code received in the query is not an allowable action for the slave. This may be because the function code is only applicable to newer devices and was not implemented in the unit selected. It could also indicate that the slave is in the wrong state to process a request of this type, for example, because it is unconfigured and is being asked to return register values [42].
- **(0x02) - Illegal data address**: The data address received in the query is not an allowable address for the slave. More specifically, the combination of the reference number and transfer length is invalid. For a controller with 100 registers, the PDU addresses the first register as 0 and the last one as 99. Suppose a request is submitted with a starting register address of 96 and a quantity of registers of 5. In that case, this request will fail with Exception Code 0x02 "Illegal Data Address," since it attempts to operate on registers 96, 97, 98, 99, and 100, and there is no register with address 100 [42].
- **(0x03) - Illegal data value**: A value contained in the query data field is not an allowable value for the slave. This indicates a fault in the structure of the remainder of a complex request, such as the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program since the MODBUS protocol is unaware of the significance of any particular value of any particular register [42].
- **(0x04) - Slave device failure**: This code means that an unrecoverable error has occurred in the slave while it was attempting to perform the requested action [42].

If any of these exceptions occur, the response message will consist of two fields; the function code and the exception code. The first field is the function code and will, in this case, reveal if this is an exception response. In normal responses, the function code of the reply will be the same as the request. However, it will be precisely 80 hexadecimal higher than the request function code in an exception case. This is because all function codes are below 80 hexadecimal. Meaning, the MSB (most significant bit) of the request function codes are zero, while in an exception response, the MSB is raised to one [42].

Modbus RTU Cyclical Redundancy Checking

The Cyclical Redundancy Checking (CRC) field contains the error check bytes of the MODBUS message, where the value in this field is used to check the content of the entire message. The error check bytes must always be present to comply with the MODBUS protocol [44]. This value is a 16-bit binary value implemented as two 8-bit bytes and is a result of a CRC calculated by the transmitting device. The receiving device then has to recalculate the CRC value during receipt of the message and compare it to the value received in the CRC field. If these two values are not equal, an error results, and the message should be discarded. For more information on how the CRC value is calculated, see appendix E.1.

4.3.2 MODBUS TCP/IP

The MODBUS TCP(Transmission Control Protocol)/IP(Internet Protocol), used by the ADAM-6066, is a variant of the MODBUS RTU transmitted with a TCP (a networking standard) wrapper over Ethernet instead of over a serial line [45]. Instead of using a slave-ID, like the MODBUS RTU, it uses IP addresses and a port number (502) to identify the devices on a network.

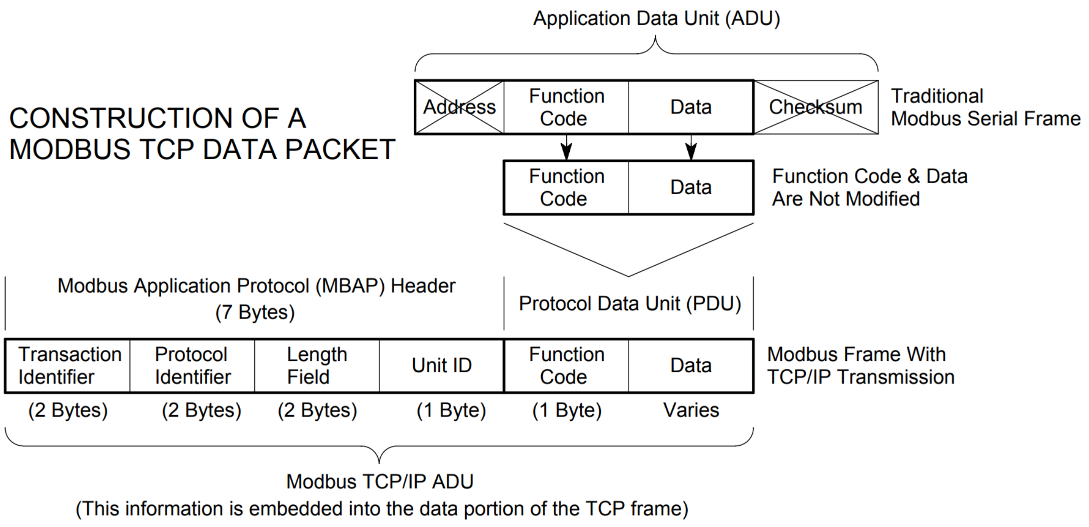

Application Data Unit (ADU)
CONSTRUCTION OF A MODBUS TCP DATA PACKET
Address
Function Code
Data
Checksum
Traditional Modbus Serial Frame
Function Code
Data
Function Code & Data Are Not Modified
Modbus Application Protocol (MBAP) Header (7 Bytes)
Protocol Data Unit (PDU)
Transaction Identifier
Protocol Identifier
Length Field
Unit ID
Function Code
Data
Modbus Frame With TCP/IP Transmission
(2 Bytes)
(2 Bytes)
(2 Bytes)
(1 Byte)
(1 Byte)
Varies
Modbus TCP/IP ADU
(This information is embedded into the data portion of the TCP frame)

Figure 4.8: Construction of a MODBUS TCP data packet [46].

Figure 4.8 describes how the MODBUS TCP/IP ADU is constructed from the previously described MODBUS application data unit (ADU) and protocol data unit (PDU), see section 4.3.1, `MODBUS command structure`. As figure 4.8 illustrates, the function code and data are the same as in MODBUS RTU, which is described earlier. The function codes used to turn ON/OFF air pumps and read flood sensor values from the two ADAM-6066 are the following two from MODBUS: #2 Read Discrete Inputs and #15 Write/Force Multiple Coils.

- **#2 Read Discrete Inputs**: This function code reads the states of the discrete inputs. The states are binary, which means the state is either 1 for ON or 0 for OFF [42].

- **#15 Write/Force Multiple Coils**: Writing the value of coils to a binary value, 1 for ON or 0 for OFF [42].

From the figure 4.8, a MODBUS application protocol header is used to make the MODBUS TCP/IP ADU.

MODBUS Application Protocol (MBAP) Header

This header is used to identify the MODBUS application data unit. It provides some differences compared to the MODBUS RTU ADU that is used on the serial line [47].

The address field is replaced by a Unit identifier (Unit ID) in the MBAP header used to communicate via single IP address devices like routers and bridges. They support multiple independent MODBUS end units [47].

All MODBUS requests and responses are designed in such a way that the recipient can verify that a message is finished [47]. Four function codes where the MODBUS PDU has a fixed length, the function code alone is sufficient. The data field includes a byte count for function codes carrying a variable amount of data in the request or response [47].

Additional length information is carried in the MBAP header to allow the recipient to recognize message boundaries even if the message has been split into multiple packets for transmission when on MODBUS TCP [47]. There is a minuscule chance of undetected corruption to request or response messages because of the existence of explicit and implicit length rules and the use of CRC-32 error check core (on Ethernet) [47].

The MBAP header consist of 4 fields: **Transaction Identifier**, **Protocol Identifier**, **Length Field** and **Unit ID**:

- **Transaction identifier**: The transaction identifier is of length 2 bytes and is an identification of a MODBUS Request/Response transaction.

- **Protocol identifier**: The protocol identifier is of length 2 bytes and is used for intra-system multiplexing. This value is 0 by default when using MODBUS protocol.

- **Length field**: The length field is of length 2 bytes and is a byte count of the following fields, which is the Unit ID and data fields (Function code and Data).

- **Unit ID**: The unit ID is only used for routing purposes when communicating with a slave device on a MODBUS serial line sub-network. The device is then communicated with through a gateway between an Ethernet TCP/IP network and a MODBUS serial line, and it is 1 byte long [47]. As there are no direct links between the Ethernet TCP/IP network and the serial lines, the unit ID is not used.

The MBAP header is a total of 7 bytes long.

MODBUS TCP/IP uses a predefined port through which the MODBUS/TCP ADU is sent. This port is 502 and is reserved for MODBUS communications.

4.4 Hamilton sensor communication

As mentioned in section 3.2.1, the Hamilton sensors use a "two-wire" RS-485 MODBUS for digital communication. With the use of RS-485, all sensors can be "daisy-chained" together to the same serial line. This allows for an easier setup and communication interface with the sensors.

USB to Serial RS-485 converter

To connect the sensors to the controlling computer, an additional component is needed to convert the physical interface of the sensors. As stated in chapter 3, it is preferred to connect all devices through USB-A ports. Thus a USB-A to Serial RS-485 conversion is used.

Figure 4.9: Exsys's USB-A 2.0 to 1 x RS-232/422/485 [48].

Exsys's USB 2.0 to 1 x RS-232/422/485, shown in figure 4.9, provides the desired conversion with a data transfer rate from 300 baud to 921,6 Kbaud [48]. Since all devices that communicate over RS-485 are "daisy-chained" together, they will use the same communication line. Thus, only one USB-A to Serial RS-485 converter is needed for this project. Also, since the Hamilton sensors support a two-wire communication line, only three terminals are used on the converter. The three terminals are R-, R+ and signal ground.

The USB-A to Serial RS-485 converter supports various types of transmission schemes, such as RS-232, RS-422, RS-485 2-wire, and RS-485 4-wire. With the converter's DIP switch, the desired transmission scheme is selected. The three pins, M0, M1, and M2, determines the transmission mode. Figure 4.10 shows the selected settings for the used Exsys's USB 2.0 to 1 x RS-232/422/485. These settings correspond to an RS-485 2-wire transmission mode. In addition to the transmission mode selected, the termination pin, pin 4, is switched ON. When the termination pin is switched to ON, the converter connects a 120 Ohm resistor between the two data wires. The restoring four pins, pin 5 to 8, are dedicated to a bias resistor but are not used for this project. For more information regarding Exsys's USB 2.0 to 1 x RS-232/422/485, see the USB to RS-485 converter manual in appendix B.5.

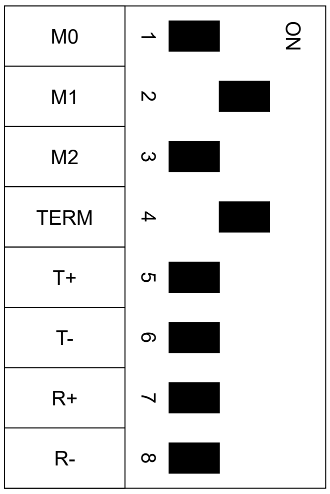


Figure 4.10: Dipp-switch settings for the USB to Serial RS-485 converter

Sensor connection

Figure 4.11 shows how the sensors are connected to the power supply and the USB to Serial RS-485 converter. As stated earlier, a 2-wire transmission scheme is used for the sensors where the sensors are "daisy-chained" together. Thus, only Data - and Data + are in use. In addition to the data wires, each sensor must be connected to the 24 V power supply. As the figure shows, the USB to Serial RS-485 converter uses the same common ground as the sensors. For a more detailed wiring diagram of this is connected, see figure 3.29.

When the sensors are "daisy-chained" together, they are connected to the Data - and Data + wires in series with the other connected sensors. One 120 Ohm resistors are placed at both ends of the data lines to avoid signal reflection in the data lines. Figure 4.11 only shows the resistor placed after the last connected sensor. The other resistor is placed within the converter, as earlier stated.

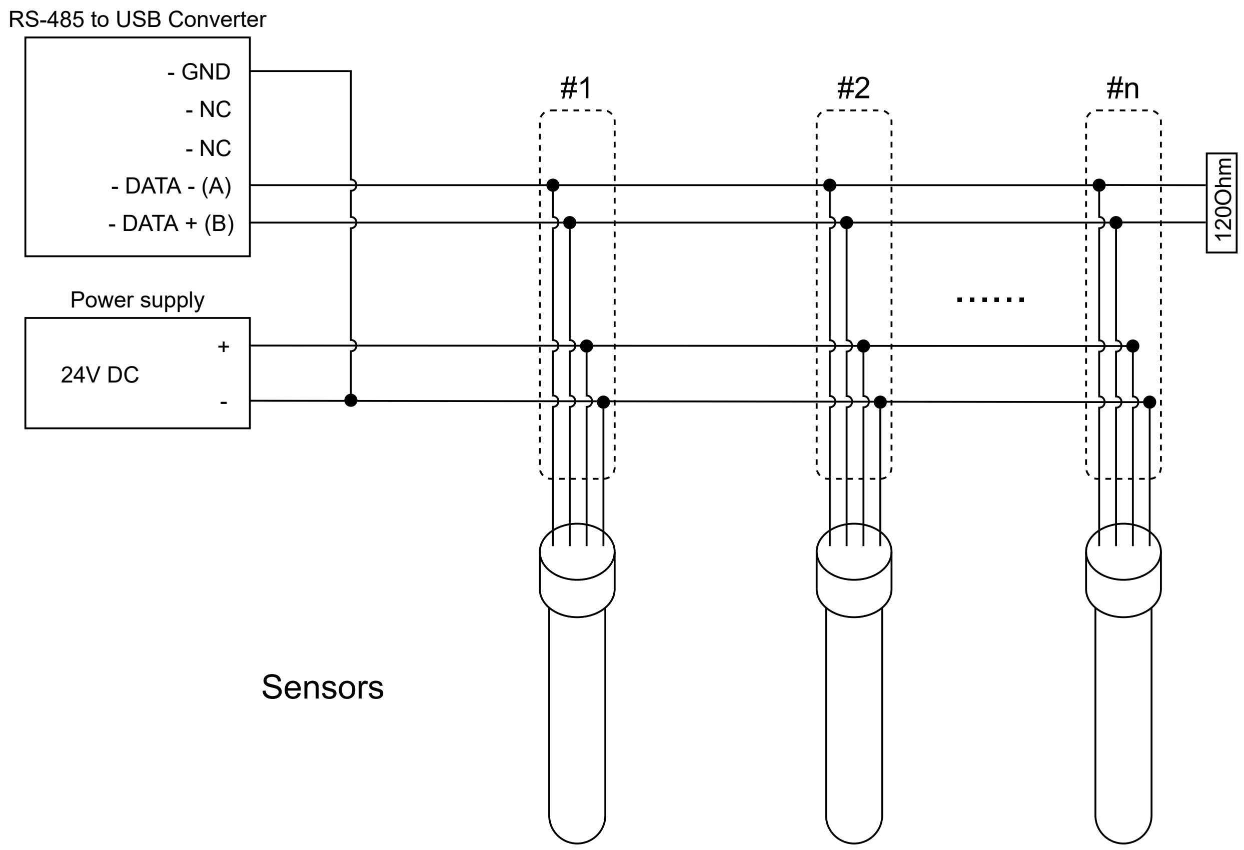


Figure 4.11: Illustration of how the daisy chained sensors are connected to the converter.

Original, the digital interface supports up to 31 sensors connected to the same RS-485 network. However, the setup for this project is limited to eight sensors. The limitation is not for the sensor's communication interface but for the control cabinet and the amount of SBR's it supports.

Command syntax

As previously stated, the Hamilton sensors use the MODBUS RTU communication protocol. Even though the sensors support many different commands with its three MODBUS function codes, we will focus on the three most used commands for this project. Remember that communication on the MODBUS RTU protocol consists of a request from the master device and then a reply from the slave device. Meaning that to receive data, the master must request it first. For more information on different commands and function codes than what is presented, see the Visiferm programming manual and EasyFerm programming manual in appendix B.1.

The three most used commands are reading the measured DO level -, pH level -, and temperature of the fluids. All three commands use MODBUS function code #3, "Read Holding Register". During data transmission, the data within the reply command should either be transformed to a different data type or interpreted as a hex value. If the data are to be transformed, the following data types are used; floating-point value, ASCII (character) value, and decimal value. For all messages defined in both sensor's programming manuals, it is clearly stated which data formats are used for each register.

Since both the DO measurement and pH measurement are placed at the same channel and register for their respective sensor, the request command for both are the same. Both values are placed at the primary measurement channel 1 (PMC1) for their respective sensor, placed at register 2090. Reading the measurement value of PMC1 for the EasyFerm sensors will give you the pH measurement while reading the measurement value of PMC1 for the VisiFerm will give you the DO measurement.

To successfully request the measurement value of PMC1, the request must consist of a slave address, the function code, the starting address for the register, the number of registers, and the calculated CRC. A request command in HEX-format sent to a device with slave address 01 will look like this: **0x 01 03 08 29 00 0A 16 65**

- **0x 01**: The slave address of the sensor (decimal "01").
- **0x 03**: The Function code, "Read Holding Registers".
- **0x 08 29**: The starting address of the register (decimal "2089").
- **0x 00 0A**: The number of registers, or length, to read (decimal "10").
- **0x 16 65**: The calculated CRC.

By requesting the measurement value of PMC1 from a VisiFerm sensor, you request the measured DO value. The sensor may reply with the following message in HEX-format: **0x 01 03 14 00 10 00 00 7B C4 41 A8 00 00 00 00 00 00 00 00 CF 8D 42 7B C0 30**

• **0x 01**: Slave address (decimal "01").	
• **0x 03**: "Read Holding Registers".	
• **0x 14**: Data-Byte count (decimal "20").	
• **0x 00 10**: Register 1 (hex).	Physical unit - low register
• **0x 00 00**: Register 2 (hex).	Physical unit - high register
• **0x 7B C4**: Register 3 (floating point).	Measurement value PMC1 - low register
• **0x 41 A8**: Register 4 (floating point).	Measurement value PMC1 - high register
• **0x 00 00**: Register 5 (hex).	Status - low register
• **0x 00 00**: Register 6 (hex).	Status - high register
• **0x 00 00**: Register 7 (floating point).	Min allowed value - low register
• **0x 00 00**: Register 8 (floating point).	Min allowed value - high register
• **0x CF 8D**: Register 9 (floating point).	Max allowed value - low register
• **0x 42 7B**: Register 10 (floating point).	Max allowed value - high register
• **0x C0 30**: CRC	

By extracting the received data from the sensor, we get the following information.

- Registers 2 and 1 provide the physical unit of the measured value. The hex string 0x 00 00 00 10 represents the physical unit %-vol.
- Register 4 and 3 provides the measurement value of PMC1. The hex string 0x 41 A8 7B C4 transformed to floating-point figure results in the decimal value 21.06043.
- Register 6 and 5 provides the status of the sensor. The string 0x 00 00 00 00 shows that there is no warning or error.
- Register 8 and 7 provides the minimum allowed value. The hex string 0x 00 00 00 00 transformed to floating-point figure results in the decimal value 0.
- Register 10 and 9 provides the maximum allowed value. The hex string 0x 42 7B CF 8D transformed to floating-point figure results in the decimal value 62.95269.

By requesting the measurement value of PMC1 from a EasyFerm sensor, you request the measured pH value. The sensor may reply with the following message in HEX-format: **0x 01 03 14 10 00 00 00 CD 0C 40 80 00 00 00 00 00 00 00 00 00 00 41 60 77 0D**

• **0x 01**: Slave address(decimal "01").	
• **0x 03**: "Read Holding Registers".	
• **0x 14**: Data-Byte count (decimal "20").	
• **0x 10 00**: Register 1 (hex).	Physical unit - low register
• **0x 00 00**: Register 2 (hex).	Physical unit - high register
• **0x CD 0C**: Register 3 (floating point).	Measurement value PMC1 - low register
• **0x 40 80**: Register 4 (floating point).	Measurement value PMC1 - high register
• **0x 00 00**: Register 5 (hex).	Status - low register
• **0x 00 00**: Register 6 (hex).	Status - high register
• **0x 00 00**: Register 7 (floating point).	Min allowed value - low register
• **0x 00 00**: Register 8 (floating point).	Min allowed value - high register
• **0x 00 00**: Register 9 (floating point).	Max allowed value - low register
• **0x 41 60**: Register 10 (floating point).	Max allowed value - high register
• **0x 77 0D**: CRC	

By extracting the received data from the sensor, we get the following information.

- Registers 2 and 1 provide the physical unit of the measured value. The hex string 0x 00 00 10 00 represents the physical unit pH.
- Register 4 and 3 provides the measurement value of PMC1. The hex string 0x 40 80 CD 0C transformed to floating-point figure results in the decimal value 4.02503.
- Register 6 and 5 provides the status of the sensor. The string 0x 00 00 00 00 shows that there is no warning or error.
- Register 8 and 7 provides the minimum allowed value. The hex string 0x 00 00 00 00 transformed to floating-point figure results in the decimal value 0.
- Register 10 and 9 provides the maximum allowed value. The hex string 0x 41 60 00 00 transformed to floating-point figure results in the decimal value 14.

The last command that is frequently used is reading the temperature. This request may be sent to both sensors, as both sensors support temperature measurements, where the request command will be the same. The temperature measurements of both sensors are placed at the Primary Measurement Channel 6 (PMC6), which uses register 2400. When requesting the temperature, the following command, in hex format, is used for a slave with address 01:
0x 01 03 09 69 00 0A 16 4D

• **0x 01**: The slave address of the sensor (decimal "01").
• **0x 03**: The Function code, "Read Holding Registers".
• **0x 09 69**: The starting address of the register (decimal "2409").
• **0x 00 0A**: The number of registers, or length, to read (decimal "10").
• **0x 16 4D**: The calculated CRC.

Both sensors will then reply with the same command structure. The sensors may reply with the following message in HEX-format: **0x 01 03 14 00 04 00 00 2A E0 41 D1 00 00 00 00 00 00 C2 2**

• **0x 01**: Slave address (decimal "01").
• **0x 03**: "Read Holding Registers".
• **0x 14**: Data-Byte count (decimal "20").

• **0x 00 04**: Register 1 (hex).	Physical unit - low register
• **0x 00 00**: Register 2 (hex).	Physical unit - high register
• **0x 2A E0**: Register 3 (floating point).	Measurement value PMC6 - low register
• **0x 41 D1**: Register 4 (floating point).	Measurement value PMC6 - high register
• **0x 00 00**: Register 5 (hex).	Status - low register
• **0x 00 00**: Register 6 (hex).	Status - high register
• **0x 00 00**: Register 7 (floating point).	Min allowed value - low register
• **0x C2 20**: Register 8 (floating point).	Min allowed value - high register
• **0x 00 00**: Register 9 (floating point).	Max allowed value - low register
• **0x 43 02**: Register 10 (floating point).	Max allowed value - high register

• **0x 70 E5**: CRC

By extracting the received data from the sensor, we get the following information.

- Registers 2 and 1 provide the physical unit of the measured value. The hex string 0x 00 00 00 04 represents the physical unit °C.

- Register 4 and 3 provides the measurement value of PMC6. The hex string 0x 41 D1 2A E0 transformed to floating-point figure results in the decimal value 26.14594.

- Register 6 and 5 provides the status of the sensor. The string 0x 00 00 00 00 shows that there is no warning or error.

- Register 8 and 7 provides the minimum allowed value. The hex string 0x C2 20 00 00 transformed to floating-point figure results in the decimal value -40.

- Register 10 and 9 provides the maximum allowed value. The hex string 0x 43 02 00 00 transformed to floating-point figure results in the decimal value 130.

For more information regarding the Hamilton sensors communication interface, see the programmers manuals in appendix B.1.

4.5 Ismatec Reglo ICC communication

As stated in section 3.4.1, the Ismatec Reglo pumps will communicate over USB instead of RS-232. When using USB instead of RS-232, a slight change occurs in the command syntax between the controlling device and the Relgo pump. Using a USB connection between the pump and the controlling computer, the connection between them is a point-to-point link, meaning no other devices are connected to the same communication line. This leads to the addressing field representing the individual pump channels instead of the different connected pumps as if a "daisy-chained" RS-232 is used.

Ismatec Reglo ICC connection

As the Ismatec's will use USB connections to connect to the computer, the connection setup is pretty straightforward. Each pump will need one USB wire connected to the USB hubs inside the control cabinet. Table 4.1 shows which USB port the Ismatec pumps should be connected to. The COM ports to each device are saved as constant values inside the program, and therefore all devices must be connected to their assigned USB port.

Device	Description	USB port
Ismatec Reglo ICC Pump 1	Ismatec pump 1 is connected to reactor 1	USB hub 2, port 3
Ismatec Reglo ICC Pump 2	Ismatec pump 2 is connected to reactor 2	USB hub 2, port 4
Ismatec Reglo ICC Pump 3	Ismatec pump 3 is connected to reactor 3	USB hub 2, port 5
Ismatec Reglo ICC Pump 4	Ismatec pump 4 is connected to reactor 4	USB hub 2, port 6

Table 4.1: Overview of where to connect the Ismatec pumps

Command syntax

The Ismatec pumps use predefined ASCII commands for their communication interface. We will now go through the most used action commands and response commands used in this project.

Since USBs are used to communicate with the pumps, you have a point-to-point link and thus no other pump connected to the same communication line. This leads to the addressing field now represents the individual pump channels instead of the different connected pump as if a "daisy-chained" RS-232 is used. A typical command with channel addressing will be 1H. 1 indicates which channel address and H is the following command. In that message case, channel 1 will start to pump.

- **Addressing**: The addressing field will represent which channel to communicate with. Since the Ismatec pumps in this project have four channels, the addressing field will be either 1, 2, 3, or 4.
- **Ismatec commands**: The computer and the control program communicates with the Ismatec pumps throughout the process. Table 4.2 shows the most used commands. The following commands are without addressing. If any of these commands are to be sent to channel 1, then the command will consist of the address followed by the command.

Command	**Function**	**Operation**	**Response**
H	Start pumping	Set	*
I	Stop pumping	Set	*
xD	Get pump direction	Get	J(CW) or K(CCW)
J	Set rotation direction to clockwise	Set	*
K	Set rotation direction to counter-clockwise	Set	*
S	Gets the current speed setting in RPM	Get	Fractional Type 1
S	Set RPM flow rate in RPM mode (0.01 RPM) in Discrete Type 3	Set	*
f	Get current volume/time flow rate(mL/min)	Get	Volume Type 1
f	Set RPM flow rate in volume/time mode (mL/min) in Volume Type 1	Set	Volume Type 1

Table 4.2: Overview of the most used Ismatec commands.

- **Data types**: The Ismatec Reglo Pumps uses a variety of different data types for message definitions. Here are some of the general data types that are used for this project, from the commands and responses of table 4.2, and how they are defined in the Reglo ICC Operating Manual B.2:
    1. Boolean:
        - Width: 1
        - Format: A single character indicating True or False.
            * 0 = False

    * 1 = True

2. Direction:
   - Width: 1
   - Format: A single character indicating direction.
     * J = Clockwise
     * K = Counter-clockwise
3. Discrete Type 3:
   - Width: 6
   - Range: 0 to 999999
   - Format: Six characters in base 10. The value is right-justified. Unused digits to the left are zeros.
4. Volume Type 1:
   - Width: 7
   - Units: mL
   - Format: mmmmEse - Represents the scientific notation of m.mmm x 10se. For example, 1.200 x 10-2 is represented with 1200E-2. Note that the decimal point is inferred after the first character.
     * mmmm - 4 character mantissa
     * E - "E" character
     * s - Sign for exponent (+ or -)
     * e - Single digit exponent
5. Fractional Type 1:
   - Width: Variable
   - Format: XXXX.XX - The width of the integer portion of the value is variable. The decimal point and two digits to the rigth of the decimal point are always provided.

- **Status response**: A single ASCII character indicates the execution status of the requested command. Here are the 4 different status responses:
  - *(0x2a): The command was executed successfully.
  - #(0x23): The command was not executed successfully.
  - -(0x2d): A negative response used for some special commands. The meaning differs for command to command, and should be checked in the command definitions for context.
  - +(0x2b): A positive response used for some special commands. The meaning differs for command to command, and should be checked in the command definitions for context.

For more information regarding the digital communication interface of the Ismatec Reglo ICC pumps, see the Reglo ICC Operating Manual in appendix B.2.

4.6 Heidolph communication

We will now go through how the Heidolph pumps are connected to the computer and how they communicate. As earlier stated in section 3.4, the Heidolph pumps use the RS-232 protocol for their digital interface, where the physical communication connection is a DE-9 connector. For the pump to meet our desires for USB connections between all devices and the computer, a converter is needed.

USB to Serial RS-232 converter

Modern computers do not support a DE-9 connector directly, and neither does the control cabinet made for this project. To connect these pumps to the computer, a USB-A to Serial converter is used. Even if other devices communicate over RS-232, they do not use a DE-9 connector, so the converter is not needed for these devices. Therefore, the Heidolph pumps are the only devices that will require conversion.

Trendnet's USB to Serial Converter, shown in figure 4.12, gives the desired conversion from USB-A to a DE-9 connector [49]. The converter can supply a data transfer rate of up to 6 Mbps, which covers the needs for the Heidolph pump communication. Since there are two of the Heidolph pumps in this project, two USB-A to Serial RS-232 converter are used.

Figure 4.12: USB to Serial converter [49].

Connection

Now that the physical communication connection is changed to USB, the connection setup is pretty straightforward. Each pump will first use the USB to Serial RS-232 converter before the USB connector is plugged into the USB hubs inside the control cabinet. Table 4.3 shows which ports on the USB hubs the Heidolph pumps should be connected to. The COM ports to each device are saved as constant values inside the program, and therefore all devices must be connected to their assigned USB port.

Device	Description	USB port
Heidolph filling pump	Heidolph filling pump is connected to all the reactors through solenoid valves	USB hub 2, port 1
Heidolph emptying pump	Heidolph emptying pump is connected to all the reactors through solenoid valves	USB hub 2, port 2

Table 4.3: Overview of where to connect the Heidolph pumps.

Command syntax

The Heidolph pumps use predefined ASCII commands for their communication interface. We will now go through the most used action commands and response commands used in this project.

- **Heidolph action commands**: The computer and the control program communicates with the Heidolph pumps throughout the process. Table 4.4 shows the most used commands.

Command	Function	Operation	Response

LED3?	Ask for status of speed rpm light	Get	LED = 0000\r\n = OFF
			LED = 0001\r\n = ON

LED4?	Ask for status of speed rpm light	Get	LED = 0000\r\n = OFF
			LED = 0001\r\n = ON

LED5?	Ask for status of speed rpm light	Get	LED = 0000\r\n = OFF
			LED = 0001\r\n = ON

LED6?	Ask for status of speed rpm light	Get	LED = 0000\r\n = OFF
			LED = 0001\r\n = ON

DSP?	Ask for the displayed value at the pump	Get	DSP=xxx\r\n
TA2!	Start/Stop pump	Set	-
TA3!	Change direction of rotation	Set	-
SDZ=xxxx!	Set rotation rpm: xxxx = 0000 - 9999	Set	-

Table 4.4: Overview of the Heidolph commands.

- **Heidolph response commands**: The following commands shown in table 4.5 are the status commands for the Heidolph pumps.

Response	Explanation
OK\r\n	Command successful
ERROR\r\n	Command failed
PO\r\n	Dosing completed

Table 4.5: Overview of the Heidolph commands.

For more information regarding the digital communication interface of the Heidolph pumps, see the Heidolph operation manual in appendix B.2.

4.7 IKA RET control-visc communication

We will now go through how the IKA RET control-visc are connected to the computer and how they communicate. As earlier stated in section 3.5.1, the IKA RET control-visc uses the RS-232 protocol for their digital interface, where the physical communication connection is a USB.

Connection

As the IKA RET control-visc uses USB connections to connect to the computer, the connection setup is pretty straightforward. Each mixer will need one USB wire connected to the USB hubs inside the control cabinet. Table 4.6 shows which USB port the IKA RET control-viscs should be connected to. The COM ports to each device are saved as constant values inside the program, and therefore all devices must be connected to their assigned USB port.

Device	Description	USB port
IKA RET control-visc 1	IKA RET control-visc 1 is connected to reactor 1	USB hub 1, port 3
IKA RET control-visc 2	IKA RET control-visc 2 is connected to reactor 2	USB hub 1, port 4
IKA RET control-visc 3	IKA RET control-visc 3 is connected to reactor 3	USB hub 1, port 5
IKA RET control-visc 4	IKA RET control-visc 4 is connected to reactor 4	USB hub 1, port 6

Table 4.6: Overview of where to connect the IKA RET control-visc.

Command syntax

The RET control-visc comply with the recommendations of NAMUR to communicate with the computer. The following commands are used for this project to communicate with the IKA RET control-visc:

NAMUR Command	Description	Available X values
IN_PV_X	Current value reading	X = 1;2;3;4;5;7;80;90
IN_SP_X	Set target value reading	X = 1;2;3;4;5;7;80;90
OUT_SP_X	Setting of target value to n	X = 1;2;4;7;54;55;56
RESET	Switch off the device function	
START_X	Swithcing ON device funcitons	X = 1;2;4;5;7;80;90
STOP_X	Swithcing OFF device funcitons	X = 1;2;4;5;7;80;90

Table 4.7: Overview of where to connect the IKA RET control-visc.

The abbreviations used above are defined as:
- **n**: variable value, floating-point number
- **X = 1**: Medium temperature (external temperature sensor)
- **X = 2**: Hot plate temperature
- **X = 3**: Hot plate safety temperature
- **X = 4**: Speed
- **X = 5**: Viscosity trend
- **X = 7**: Heat transfer medium temperature
- **X = 80**: pH value
- **X = 90**: Weight value
- **X = 54**: Error 5 response time in seconds (180 <= n <= 1200)
- **X = 55**: Intermittent mode cycle time in seconds (10 <= n <= 600)
- **X = 56**: Intermittent mode, duration of interruption in seconds (5 <= n <= 60)

For more information regarding the IKA RET control-visc and its communication interface, see IKA RET control-visc manual in appendix B.3.

Controlling the available operations

This project uses two of the IKA RET control-visc operations, which are stirring and weighting. These two operations are controlled as follows:

- **Stirring**: The stirring function may be controlled by a simple ON/OFF command or an intermittent mode. The intermittent mode gives the opportunity to run the stirring cyclical by selecting both a desired run and stop time. The stirring module also comes with a stirring bar decoupling safety, where the stirring function stops in order for the user to pick up the bar. The system then returns to its previously set speed once the bar is reconnected. In the case of decoupling again within 3 minutes, the maximum speed is reduced to 100 rpm, and if there should be four consecutive decouplings, the error message ER 41 is displayed. In this case, heating is switched off, and testing conditions should be reconsidered.

- **Weighting**: When active, the weighting function measures the weight placed upon the IKA RET control-visc. When the weighting function is started, the scale sets the current weight as zero and measures relative to this value. Ideally, the stirring and heating should be turned off during weighing due to the inaccuracies generated by the movements, but it is possible to run all simultaneously.

4.8 ADAM-6066 communication

We will now go through how the ADAM-6066 modules are connected to the computer and how they communicate. As earlier stated in section 3.8.1, the ADAM-6066 uses the MODBUS TCP/IP protocol for their digital interface, where the physical communication connection is Ethernet.

USB to Ethernet adapter

Because the ADAM-6066 connects to a controlling device by Ethernet, a USB to Ethernet adapter is used to open a communication line between the controlling device and the two ADAM-6066. Trendnet's TU2-ET100 USB to fast Ethernet adapter, shown in figure 4.13, gives the desired conversion from USB-A 2.0 to a 10/100 Mbps Ethernet port [50]. Another option is to use a network card, but by using an adapter, the computer does not require a network card with Ethernet ports. The current computer used to run the program does not have Ethernet ports available.

Figure 4.13: Trendnet TU2-ET100 USB to Ethernet adapter [50].

Connection

The two ADAM-6066 modules are connected with the computer in an Ethernet switch, creating a network between the computer (through the adapter and USB hub) and the two modules. Figure 4.14 illustrates how the modules switch and the adapter is connected together.

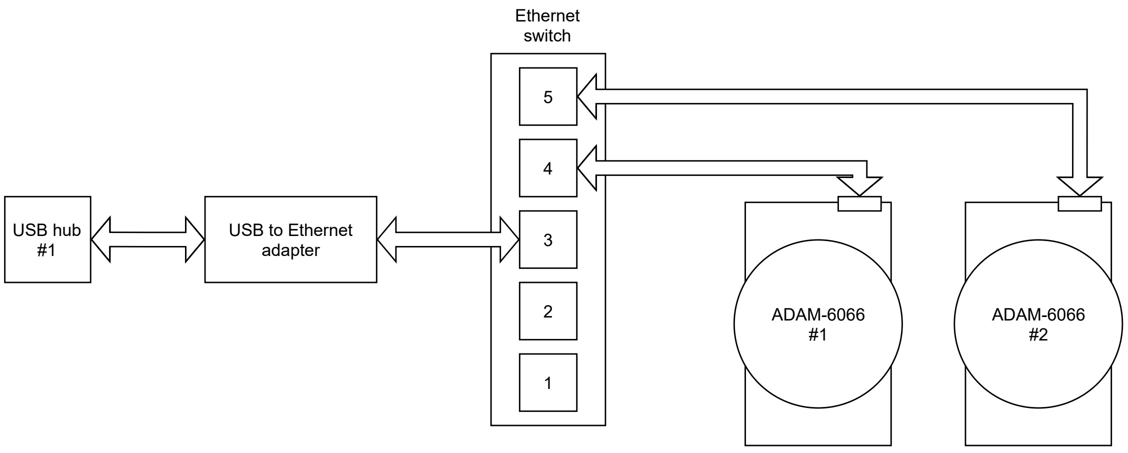


Figure 4.14: Illustration of how the modules, switch, and adapter are connected.

The modules are configured through a software provided by Advantech, ADAM/APAX. Net Utility. Here they are assigned a static IP address each. The IP addresses are: **169.254.14.156** for ADAM-6066 #1 and **169.254.14.157** for ADAM-6066 #2, and as mentioned earlier the port used is **502**. For more information about the setup, see the operation manual in appendix B.4 The software is only used to assign the modules with a suitable IP address. The rest is handled in LabView through MODBUS commands.

Command syntax

As mentioned, the ADAM modules use MODBUS messages in hex to operate and perform the actions. The modules communicate with the computer in the command-response form [51]. The modules will be in listen mode when no data is not transmitted.

A couple of examples of the two function codes are listed to better understand how the PDU structure sends messages to a module. The examples are slightly modified from the ADAM-6000 user manual [51]. The examples only look at the PDU (i.e., without MBAP header):

#2 Read Discrete Inputs: Want to read the status of discrete inputs (input 1 $\rightarrow$ 8) on the ADAM-6066 #1 module. Assume that input 2 and 3 is ON and the rest is OFF.
By sending message:
- 01 02 00 01 00 06

Where:
- 01 is the station address.
- 02 is the function code.
- 00 is the start address high byte.
- 01 is the start address lowbyte.
- 00 is the requested number of input high byte.
- 06 is the requested number of input low byte.

If only input 1 and input 2 is ON the recieving message is:
- 01 02 01 60

The format of the recieving message is:
- 01 - Station address
- 02 - Function code
- 01 - Byte count
- 60 - Data.

The data, 60 in hex, represent 0110 0000 in the binary format, which tells us that input 2 and 3 are ON (= 1).

#15 Write/Force Multiple Coils This function code sets the relays to either be ON (=1) or OFF (=0). i.e., writing coils in a sequence. Function code 15 is 0F in hex.
We want to write a series of 2 coils starting at address 00017 (11 in hex) in the ADAM-6066 #1 module, which is the relays 1 and 2.
By sending the request message:
- 01 0F 00 11 00 02 02 03 00

The structure is:
- 01 - Station address.
- 0F - Function code.
- 00 - Start address high byte.
- 11 - Start address low byte.
- 00 - Requested number of coil high byte.
- 02 - Requested number of coil low byte.
- 02 - Byte count.
- 03 - Force data high byte.
- 00 - Force data low byte

The data contents are two bytes: 03 00 (hex), equivalent to 0000 0011 0000 0000. The binary bits map to the addresses in the following manner:

This tells us that setting the addresses 17 and 18 bits to 1, the first two relays switches.

Bit:	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Address (000XX):	24	23	22	21	20	19	18	17	-	-	-	-	-	-	26	25

Table 4.8: Table of how the binary bits are mapped to the addresses in the module [51].

A normal response is:
- 01 0F 00 11 00 02

Where:
- 01 - Station address.
- 0F - Function code.
- 00 - Start address high byte.
- 11 - Start address low byte.
- 00 - Requested number of coil high byte.
- 02 - Requested number of coil low byte.

0x02 describes that the requested number of forced coils is 2.

5. Control program

Often in SBR processes, it is desired to run multiple sequences with the same growth conditions to observe the growth of the microorganisms. To produce these same growth conditions for a desired number of times, one needs to control values like the volume of liquid at the different stages and dissolved oxygen level. These control values need to be processed through the connected computer.

The whole process is controlled through the connected computer using a control program developed in LabView, a visual programming language. The computer communicates with all devices connected to this system through this control program. The user can adjust control parameters, control and observe the process within the control program. It is possible to study the measured and estimated data as visual graphs or in an excel spreadsheet as the process runs.

The developed program can be explained in two parts, the graphical user interface(GUI) and the program operations. The GUI is what the user sees and how to operate it, and the program operations are what happens behind the graphical interface when the program is running.

The control program has mainly been developed using our own intuition. However, some libraries and available code have been used to help develop the program. These libraries are listed in appendix G.

5.1 Graphical user interface

The GUI is designed to give the user a good and clear overview of all measured variables and incoming data. The control program may control up to four reactors running at the same time. When all reactors are active, the user may get lost in all the incoming data. Therefore it is essential to present a clear overview of which devices are active, the status of each reactor, and all incoming data.

The program uses in total 11 tabs to separate all information, which are six process tabs and five device tabs. With the use of these tabs, the user can easily navigate through the program. Each tab gives the necessary information of the selected tab. The process tabs consist of measurements and control parameters of each reactor, and the device tabs will consist of information regarding the selected device type.

The six process tabs consist of one main tab, four SBR tabs, and one Excel log tab. The main tab is dedicated to a total overview of all the active reactors and devices. The four SBR tabs are dedicated to the four possible reactors, providing the user with more specific information regarding the selected SBR. Within the SBR tabs, the user may choose freely between control methods for the reaction stage of the process, in addition to setting desired control parameters. The last process tab, the Excel log tab, contains all the information regarding the control program's logging function.

The five device tabs are dedicated to all the connected components in which the control program operates. These are the Ismatec pumps, Heidolph pumps, the IKA RET control-visc, the Hamilton sensors, and the Adam-6066 I/O modules. The device tabs will consist of information regarding the selected device type and not a single device. If the user selects the Imsatec tab, information regarding all four possible Ismatec pumps will be provided.

5.1.1 Main-tab

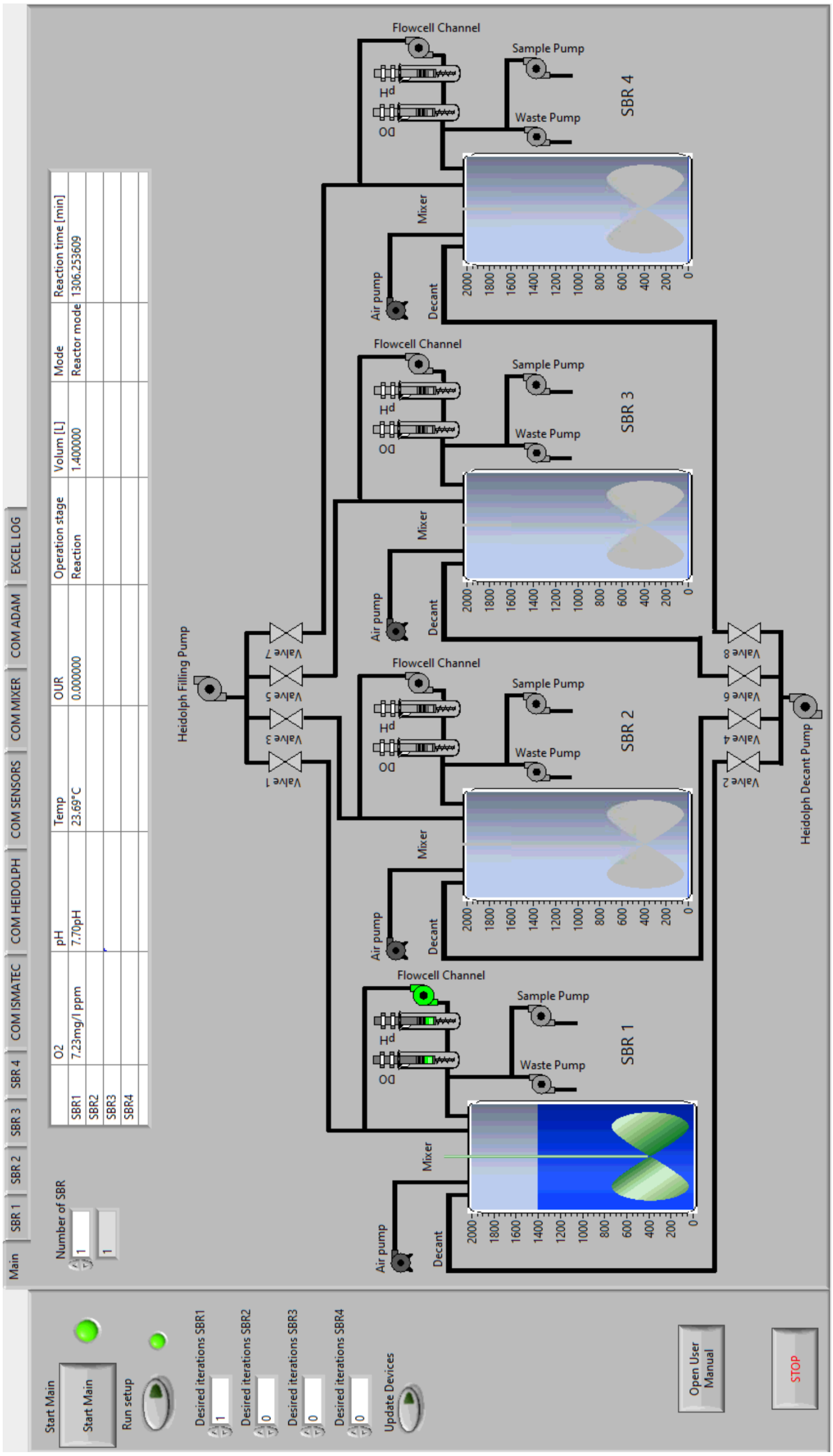
The "Main"-tab, shown in figure 5.1, is meant to provide the user the most important data regarding the running process and has the ability to set desired control parameters that affect the whole process, such as the amount of SBR-processes to run. This will give the user an overview of how the process is going for each reactor without reading through too much information. If the user wants more information regarding a specific reactor or device, the other tabs will provide this.

The most relevant data for each reactor is provided inside a table placed at the top of the tab. The table consists of four rows, one row for each reactor. Each row will provide information such as the three sensor measurement values(DO, pH, and temperature), OUR, operation stage, volume, the control mode, and the reaction time. If the control mode is set to OUR mode, the last estimated OUR value is shown in the table. If the control mode is not set to OUR mode, this column will be set to N/A.

The bottom area of the tab, in figure 5.1, shows an image representing all reactors with their corresponding pumps, mixers, sensors, and valves. This area provides the user with an overview of which of these devices that are active for any given time.

The pumps, mixers, and valves will change color whenever the status of the device change. When these devices are active, the color changes to green, and when the devices are not active, the color is set to grey. In addition to the device status, the current volume in each reactor is also shown.

The box placed on the left-hand side of the tab, separated from the main tab, will always be present regardless of which tab is selected. This box contains multiple buttons and control parameters. The buttons are a start button of the "Main"-sequence, the run setup button, an update devices button, an open user manual button, and the stop button. In addition, in the middle of the box, the user may select desired iterations for each reactor.

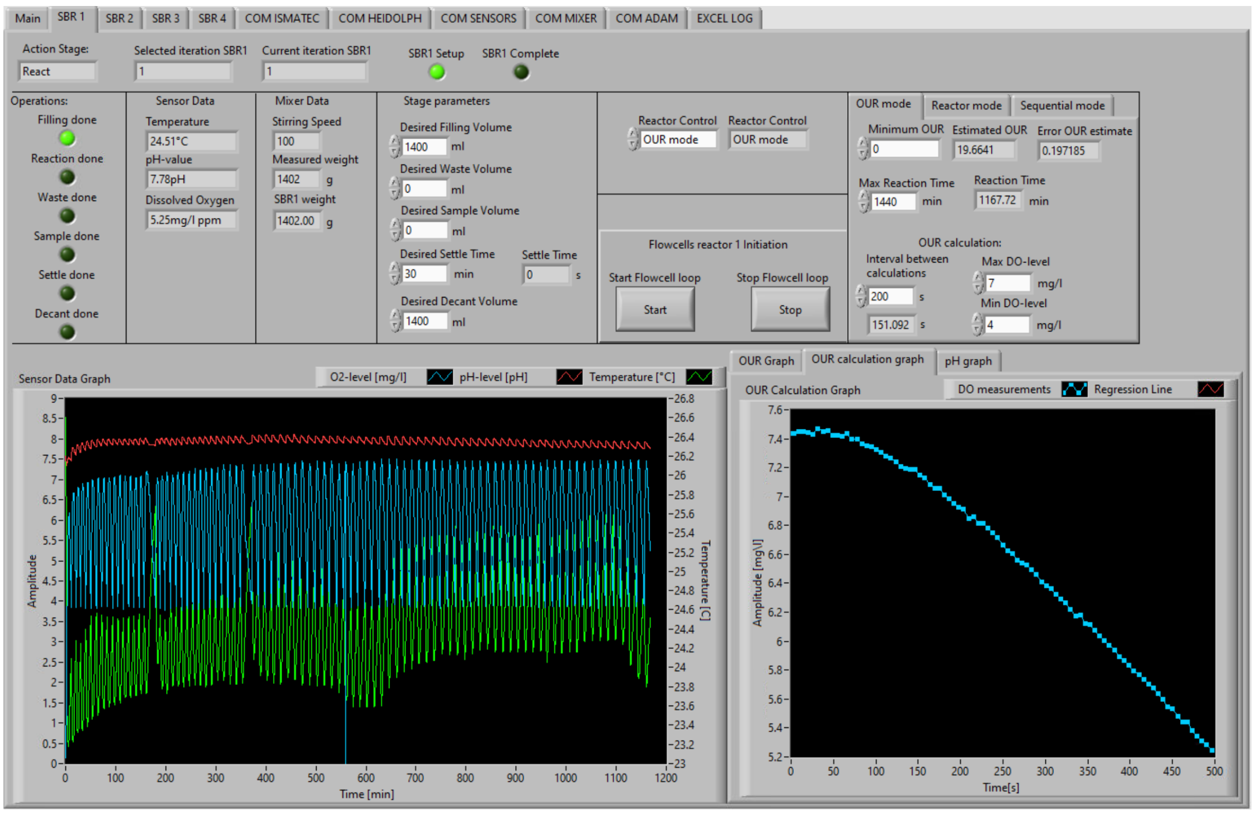


	O2	pH	Temp	OUR	Operation stage	Volum [L]	Mode	Reaction time [min]
SBR1	7.23mg/l ppm	7.70pH	23.69°C	0.000000	Reaction	1.400000	Reactor mode	1306.253609
SBR2								
SBR3								
SBR4								

Figure 5.1: Overview of main tab.

5.1.2 SBR tabs

The measured data are sorted into *SBR* tabs, where each tab relates to the different reactors in use. When in one of the *SBR* tabs, all information belongs to the selected reactor. More specific information regarding the selected reactor is given to the user in these tabs and the possibility to set desired control parameters. The *SBR* 1 - 4 tabs will look the same but with different measurements and control parameters. Figure 5.2 shows one of these tabs.



Action Stage: React
Selected iteration SBR1: 1
Current iteration SBR1: 1

Temperature: 24.51°C
pH-value: 7.78pH
Dissolved Oxygen: 5.25mg/l ppm

Stirring Speed: 100
Measured weight: 1402 g
SBR1 weight: 1402.00 g

Desired Filling Volume: 1400 ml
Desired Waste Volume: 0 ml
Desired Sample Volume: 0 ml
Desired Settle Time: 30 min
Settle Time: 0 s
Desired Decant Volume: 1400 ml

Reactor Control: OUR mode
Reactor Control: OUR mode

Minimum OUR: 0
Estimated OUR: 19.6641
Error OUR estimate: 0.197185
Max Reaction Time: 1440 min
Reaction Time: 1167.72 min
Interval between calculations: 200 s
151.092 s
Max DO-level: 7 mg/l
Min DO-level: 4 mg/l

Figure 5.2: Overview of control settings and measured variables for reactor 1. Notice how all measurements plotted in the *Sensor Data Graph* oscillate. This is a result of the OUR calculation when the flow cell loop is switched off.

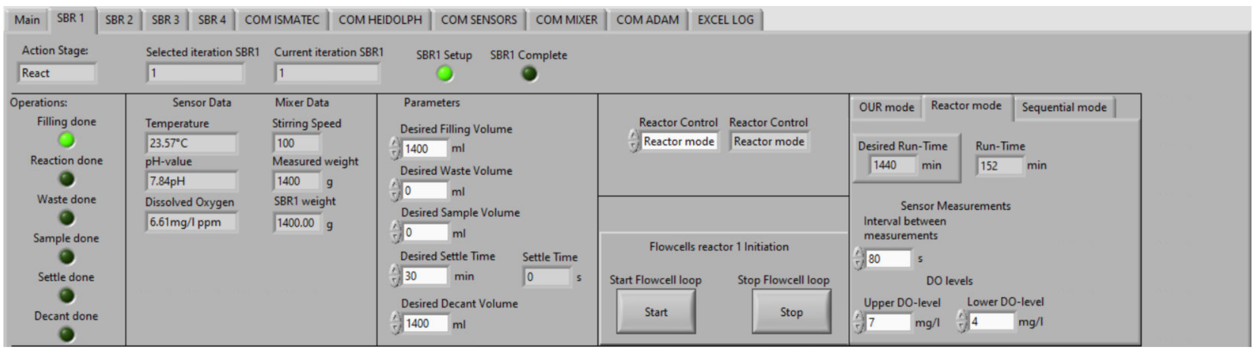
The *SBR* tabs can be split into an upper half and a lower half. The upper half consists of information such as individual measurements, control parameters, and control options. The lower half contains multiple graphs in which the sensor measurements are plotted.

The upper half is split into groups to separate measured data, control parameters, and options and is shown in figure 5.3. At the top of the tab, the current action stage, the selected iterations, and the current iteration is shown, along with indications showing if the setup was successful and if the process is complete. Beneath this, seven different boxes contain the following information:

- **Operations**: This box provides the user with indications of which stages that are completed in the current iteration. If a stage is complete, a green light will appear next to the completed stage.

- **Sensor data**: This box provides the user with the sensor measurements. The sensor data are only updated when the program receives new readings from the sensors, which will only be when the reactor is in the reaction stage.
- **Mixer data**: This box provides the user with the current stirring speed and measured weight from the magnetic mixer. The stirring speed will only be updated during the reaction, waste, and sample stage, while the measured weight will be updated during the fill, waste, sample, and decant stages.
- **Stage parameters**: This box provides the user with control parameters of the different stages. Here the user may adjust the filling, waste, sample, and decant volume. Also, the settling time is selected here, along with an indication of how long the reactor has been settling.
- **Control mode**: This box provides the user with three different control methods to choose from; OUR mode, reactor mode, and sequential mode.
- **Filling the flow cell**: This box is dedicated to the filling of the flow cells. At startup, the user might need to fill the flow cells. This is only needed if there is any air inside the flow cells. If the user needs to fill the flow cells, the connected Ismatec pump will need to run in the opposite direction than in normal operation. Here, the user may start and stop the filling.
- **Parameters regarding the selected control mode**: This box consists of three different tabs for the three different control methods. If a new control method is selected, the corresponding tab will be selected by the program. Inside these tabs, the user may select the desired control parameters of the selected control method. The three different control method tabs are shown in figure 5.4.
  - **OUR mode**: When selecting the OUR mode, the user is provided a control tab containing control parameters. This mode is controlled by both time and DO measurements. The user can freely adjust the minimum OUR, the maximum reaction time, the interval between OUR estimations, and the upper and lower DO level during OUR estimations. The minimum OUR and the maximum reaction time control the duration of the reaction stage. The first of these parameters to be reached will cancel the reaction stage and move onto the next stage. The interval between calculations and DO levels control the OUR estimations. The estimations will start by either the interval duration or the maximum DO level is reached. The estimation sequence will then last until the DO level reaches the minimum DO level set by the user. The OUR-mode tab is shown in figure 5.4a.
  - **Reactor model**: When selecting the reactor mode, the user is provided a new control tab containing control parameters. These parameters are the upper and lower DO level in the reactor, the interval between measurements, and desired reaction time. When the lower DO level is reached, the air pump starts and supplies the reactor with air, and the DO level will start to increase. When the upper DO level is reached, the air pump is turned off. The interval between measurements will control how often the fluid loop containing the flow cell will circulate. The Reactor-mode tab is shown in figure 5.4b.
  - **Sequential mode**: When selecting the sequential control option, the user is provided a new control tab containing control parameters. The sequential mode

is similar to the reactor mode, where both have the desired reaction time that controls the duration of the reaction stage and intervals between measurements. The difference comes with the DO control, where the sequential mode controls the air supply by desired intervals instead of measured DO levels. The Sequential-mode tab is shown in figure 5.4c.



Action Stage: React
Selected iteration SBR1: 1
Current iteration SBR1: 1
SBR1 Setup ●
SBR1 Complete ●

Operations:
Filling done ●
Reaction done ●
Waste done ●
Sample done ●
Settle done ●
Decant done ●

Sensor Data
Temperature: 23.57°C
pH-value: 7.84pH
Dissolved Oxygen: 6.61mg/l ppm

Mixer Data
Stirring Speed: 100
Measured weight: 1400 g
SBR1 weight: 1400.00 g

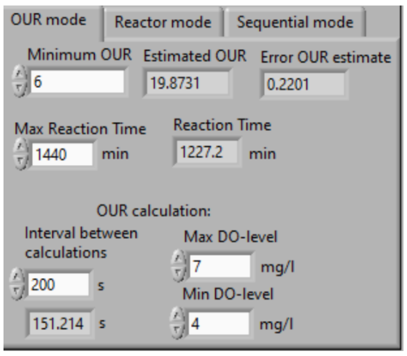
Parameters
Desired Filling Volume: 1400 ml
Desired Waste Volume: 0 ml
Desired Sample Volume: 0 ml
Desired Settle Time: 30 min
Settle Time: 0 s
Desired Decant Volume: 1400 ml

Reactor Control: Reactor mode
Reactor Control: Reactor mode

Flowcells reactor 1 Initiation
Start Flowcell loop
Stop Flowcell loop

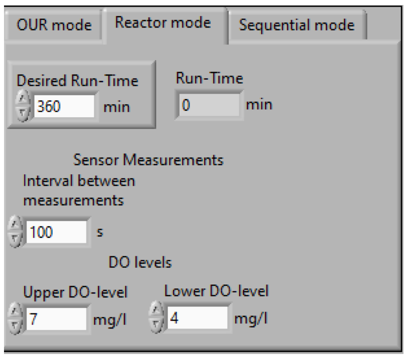
Desired Run-Time: 1440 min
Run-Time: 152 min
Sensor Measurements
Interval between measurements: 80 s
DO levels
Upper DO-level: 7 mg/l
Lower DO-level: 4 mg/l

Figure 5.3: Overview of the upper area of the SBR pages



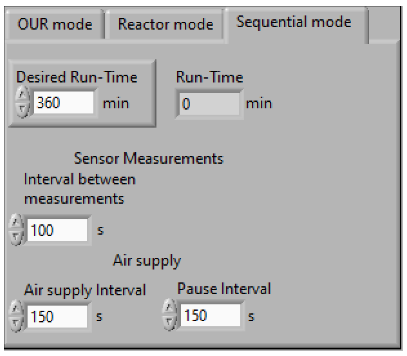
Minimum OUR: 6
Estimated OUR: 19.8731
Error OUR estimate: 0.2201
Max Reaction Time: 1440 min
Reaction Time: 1227.2 min
OUR calculation:
Interval between calculations: 200 s
151.214 s
Max DO-level: 7 mg/l
Min DO-level: 4 mg/l

(a) OUR mode tab.



Desired Run-Time: 360 min
Run-Time: 0 min
Sensor Measurements
Interval between measurements: 100 s
DO levels
Upper DO-level: 7 mg/l
Lower DO-level: 4 mg/l

(b) Reactor mode tab.



Desired Run-Time: 360 min
Run-Time: 0 min
Sensor Measurements
Interval between measurements: 100 s
Air supply
Air supply Interval: 150 s
Pause Interval: 150 s

(c) Sequential mode tab.

Figure 5.4: The three control methods tabs.

The lower half of the *SBR* tabs are, as mentioned, dedicated to multiple graphs and are shown in figure 5.5. This area consists of four graphs that show the development of the sensor measurements and are only updated during the reaction stage. The main graph that will be active for all the control methods is the *Sensor Data Grap*, which shows the development of all the sensor measurements; DO level, pH level, and temperature. These values are plotted against the relative time since the start of the reaction stage.

The three other graphs are placed in three different tabs, placed at the right-hand side of the lower half. The user may toggle freely between the tabs throughout the operation. These graphs are only updated when the OUR mode is chosen. The three graphs are OUR development, OUR calculation, and pH. The three graphs are shown in figure 5.6.

The first of the graphs, the OUR development, are shown in figure 5.6a. Here, every estimated OUR value is plotted against the relative time of the reaction stage. The user can then see how the OUR develops throughout the reaction stage.

The following graph, the OUR calculation graph, is shown in figure 5.6b. This graph shows how the DO level change during the OUR estimations and is only updated during this sequence. This data is the foundation for the estimation of the OUR value. When the measured DO level goes below the minimum DO level set by the user, the graph is updated with the calculated linear regression line. This graph is cleared every time before the OUR estimation sequence start.

The last graph, the pH graph, is shown in figure 5.6c. As for the OUR calculation graph, this graph shows how the pH level change during the OUR estimations and is only updated during this sequence. This graph is also cleared before the start of the OUR estimation sequence.

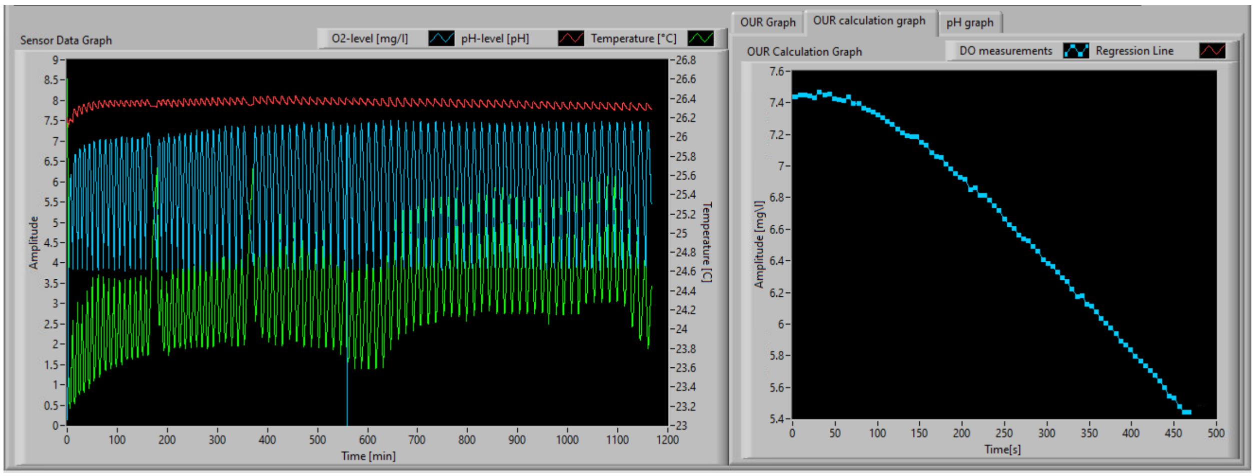
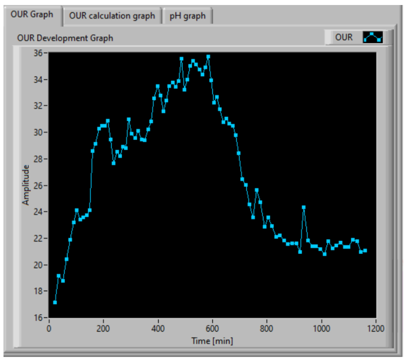
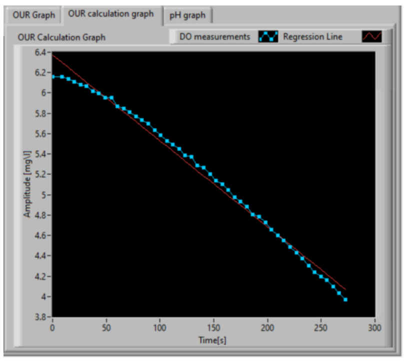


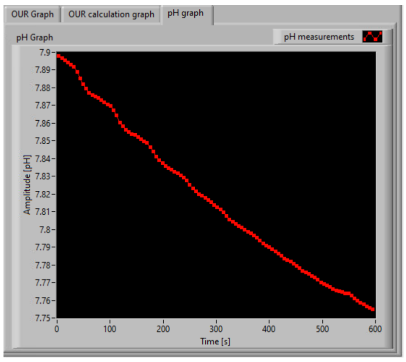
Figure 5.5: Overview of the bottom area of the SBR pages.



(a) OUR development graph.



(b) OUR calculation graph.



(c) pH graph.

Figure 5.6: The three control methods tabs.

5.1.3 Device communication tabs

The remaining six tabs are dedicated to the different devices that are connected to the computer, in addition to an Excel log tab. Here, parameters such as COM-ports, pump speed, rotational speed, among other things, may be selected. More specific information regarding the selected device can also be found here. In addition, the user may also send direct messages to the individual components for debugging.

COM Ismatec tab

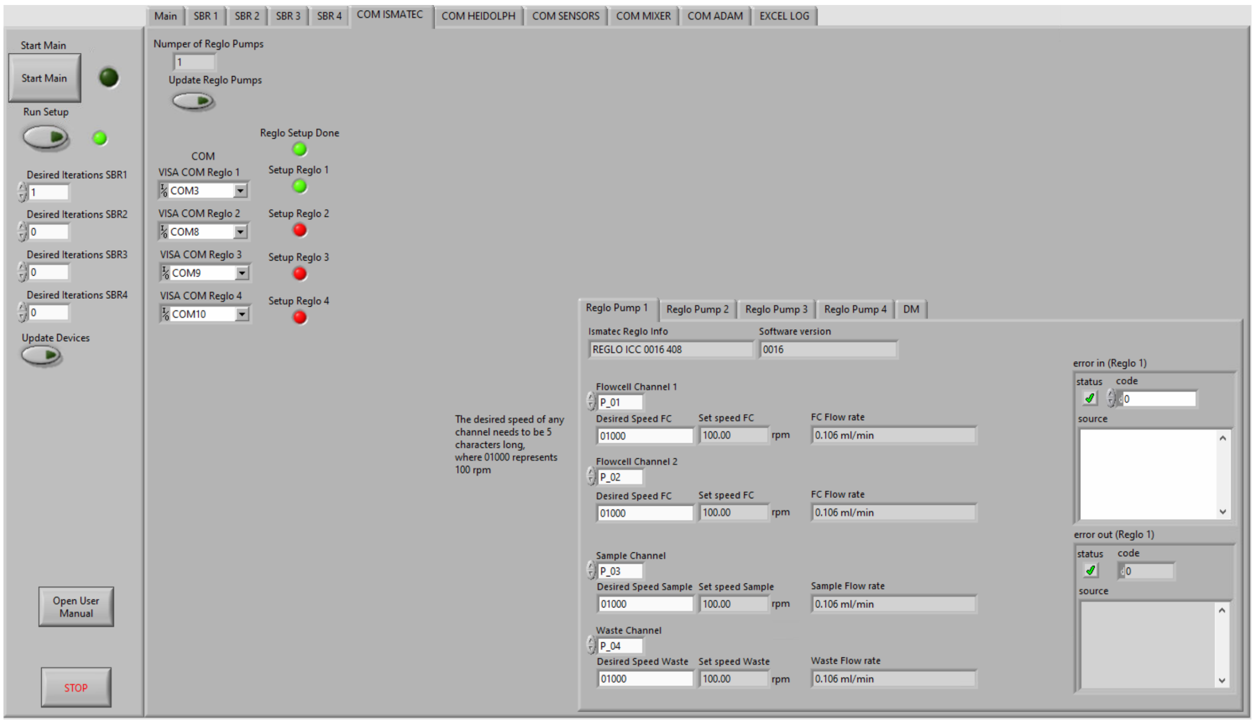
This tab, shown in figure 5.7, is dedicated to the Ismatec Reglo pumps and contains all parameters regarding the Ismatec Reglo pumps. Inside the tab, general information is found on the left-hand side of the page and an additional page containing five tabs is found on the right-hand side of the page. The numbering that has been given to the different indicators, tabs, and COM-ports shows which reactor the data belongs to, which means that the *VISA COM Reglo 1*, *Reglo Pump 1* tab, and *Setup Reglo 1* all belong to the reactor 1 setup.

The general information consists of the selected COM-port for each connected pump, indications showing if the setup was successful, and the number of Ismatec Reglo pumps needed for the desired amount of active reactors. The COM-ports dedicated to each pump are already set as a default value inside the program. Meaning every time the program starts, the correct COM-port is assigned to the pumps. The COM-port assigned to the Ismatec pump 1 must correspond to the pump physically connected to reactor 1. This is, of course, the same for the other three COM-ports. Even if it should not be needed, the user may select other COM-ports for the pumps. In addition to this information, a button is provided to the user. If the user wants to update the Ismatec pumps with new parameters without running the whole main setup sequence, this button can be pressed. The program will then re-configure the Ismatec pumps.

When the program runs its setup sequence, it configures the number of Reglo Pumps shown in *Number of Reglo Pumps.* If this number is two, the program configures Reglo 1 and 2 during setup. If this setup is complete, the two indicators, *Setup Reglo 1* and *Setup Reglo 2* lights up green. These two indicators need to be green before the main sequence can start.

The additional page on the right-hand side of the *COM Ismatec* tab is made to separate data between the four possible Ismatec pumps. Four of the pages are named after which reactor the pump corresponds to, as *Reglo Pump 1-4*. Each of these tabs will only contain data corresponding to its pump. The fifth page is dedicated to individual communication, giving the user the possibility to send direct messages to the desired pump and channel. This can be used to debug the pumps or check other values that are not shown on the other four pages.

The four pages, *Reglo Pump 1-4*, contains both controls and indicators. The controls are parameters that the user may change if desired. If not, these are set to default values when the program starts. The control parameters are each channel's pump speed and which channel is assigned to which operation. After a setup or update of the Ismatec Reglo pump has been completed, the indicators will show the actual speeds set within the pump with the corresponding flow rate next to the speed. This is to ensure that the desired parameters are indeed set. Also, the software version of the pumps will be shown along with some basic info about the pumps.



Numper of Reglo Pumps: 1

Desired Iterations SBR1: 1
Desired Iterations SBR2: 0
Desired Iterations SBR3: 0
Desired Iterations SBR4: 0

COM
VISA COM Reglo 1: COM3
VISA COM Reglo 2: COM8
VISA COM Reglo 3: COM9
VISA COM Reglo 4: COM10

Reglo Setup Done
Setup Reglo 1
Setup Reglo 2
Setup Reglo 3
Setup Reglo 4

The desired speed of any channel needs to be 5 characters long, where 01000 represents 100 rpm

Ismatec Reglo Info: REGLO ICC 0016 408
Software version: 0016

Flowcell Channel 1: P_01
Desired Speed FC: 01000 Set speed FC: 100.00 rpm FC Flow rate: 0.106 ml/min

Flowcell Channel 2: P_02
Desired Speed FC: 01000 Set speed FC: 100.00 rpm FC Flow rate: 0.106 ml/min

Sample Channel: P_03
Desired Speed Sample: 01000 Set speed Sample: 100.00 rpm Sample Flow rate: 0.106 ml/min

Waste Channel: P_04
Desired Speed Waste: 01000 Set speed Waste: 100.00 rpm Waste Flow rate: 0.106 ml/min

error in (Reglo 1)
status: ✓ code: 0
source

error out (Reglo 1)
status: ✓ code: 0
source

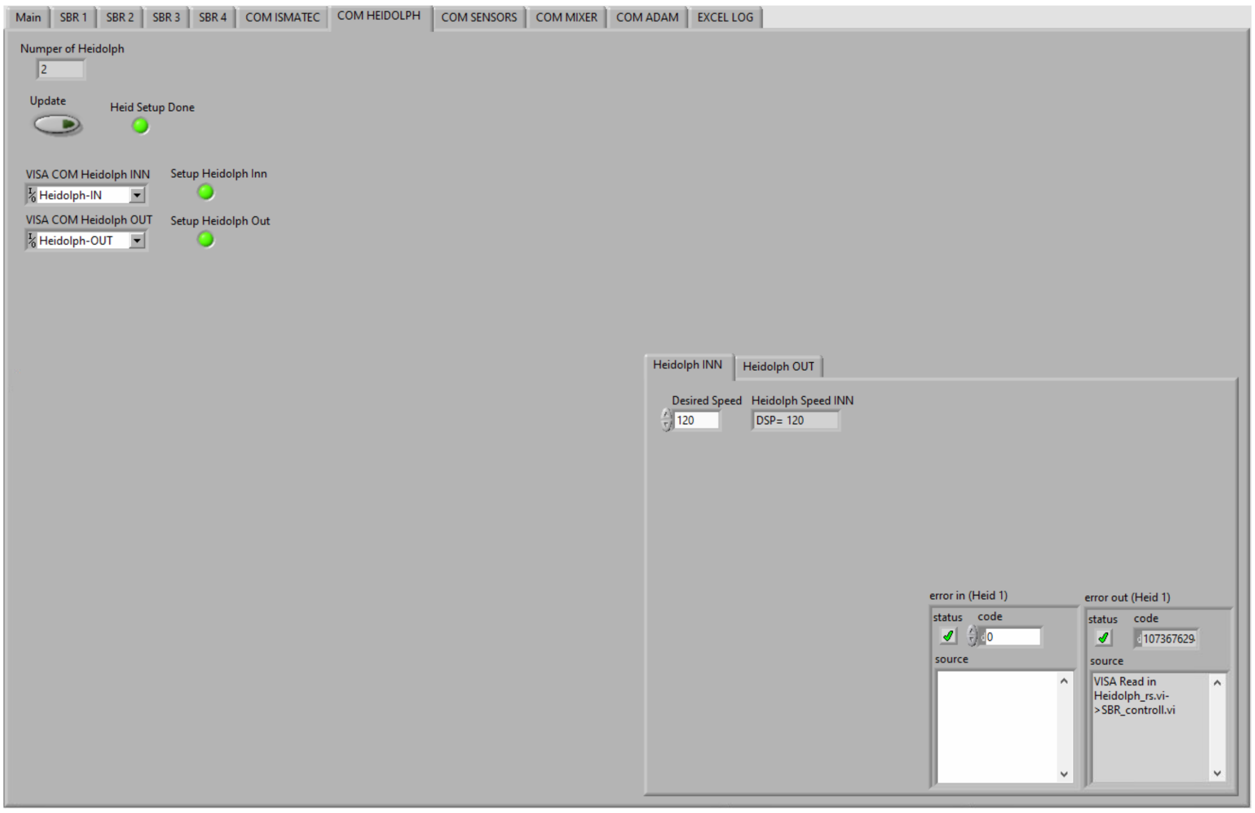
Figure 5.7: Overview of communication settings for the Reglo pumps.

COM Heidolph tab

This tab, shown in figure 5.8, is dedicated to the Heidolph pumps and consists of all parameters regarding the Heidolph pumps. The setup of the page is similar to the *COM Ismatec* tab, where general information is found on the left-hand side of the page and an additional page containing two tabs is found on the right-hand side of the page.

The general information consists of the selected COM-port for each connected pump, indications showing if the setup was successful, and the number of Heidolph pumps needed for the desired amount of active reactors. The number of Heidolph pumps will always be two. The COM-ports dedicated to each pump are already set as a default value inside the program. Even if it should not be needed, the user may select other COM-ports for the pumps. In addition to this information, a button is provided to the user. If the user wants to update the Heidolph pumps with a new desired speed without running the whole main setup sequence, this button can be pressed. The program will then re-configure the Heidolph pumps.

The additional page on the right-hand side of the *COM Heidolph* tab is made to separate data between the two Heidolph pumps. Inside each of the two tabs, the desired speed may be selected by the user. If not, this is set to a default value. When the Heidolph pumps are configured in the setup, the actual assigned speed is shown next to the desired speed.



Numper of Heidolph: 2

VISA COM Heidolph INN: Heidolph-IN

VISA COM Heidolph OUT: Heidolph-OUT

Desired Speed: 120
Heidolph Speed INN: DSP= 120

error in (Heid 1)
status: ✓ code: 0
source:

error out (Heid 1)
status: ✓ code: 107367629
source: VISA Read in Heidolph_rs.vi->SBR_controll.vi

Figure 5.8: Overview of communication settings for the Heidolph pumps.

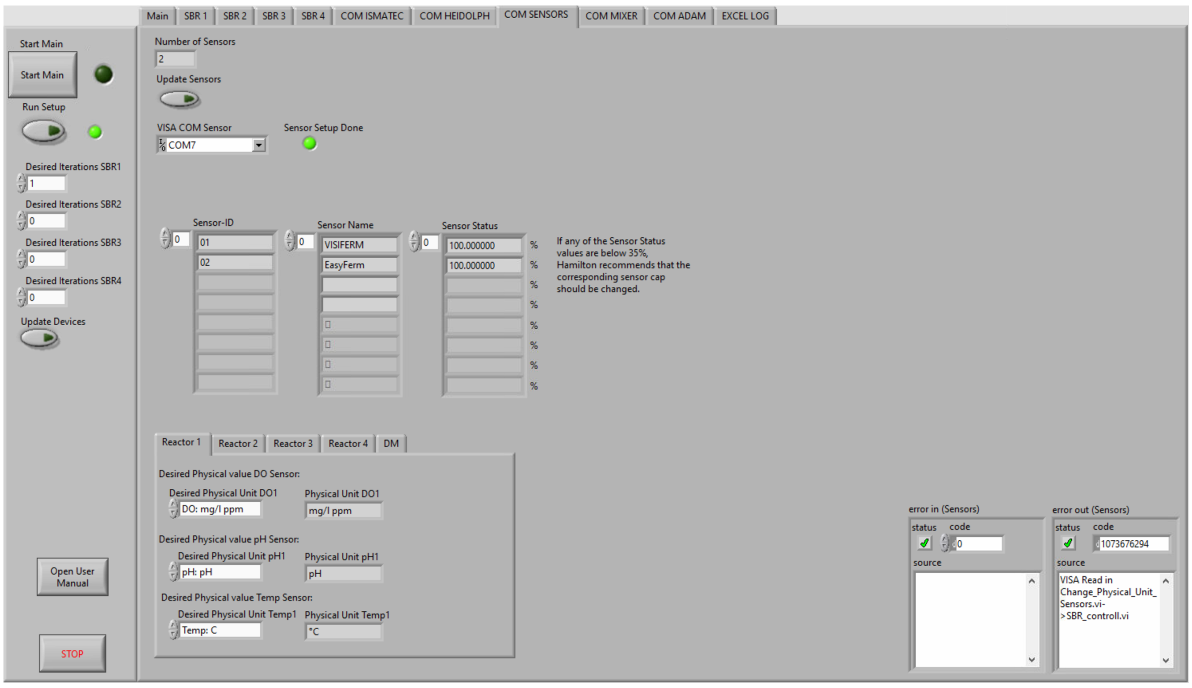
COM Sensors tab

This tab, shown in figure 5.9, is dedicated to the Hamilton sensors and contains information regarding the Hamilton sensors. The page consists of general information, the connected sensor's ID, name, and status, and an additional page containing the physical unit of the measurements from the sensors.

The general information presented at the upper left corner of the page is similar to the other device pages, showing the number of sensors, the assigned COM-port, and an indicator that shows if the setup was successful. The COM-port dedicated to the sensors is already set as a default value inside the program. Even if it should not be needed, the user may select another COM-port for the sensor. In addition to this information, an update button is provided to the user. If the user wants to re-configure the sensors without running the whole main setup sequence, this button can be pressed.

The three arrays presented beneath the general information contain the connected sensor's ID, name, and status. These arrays are updated during the setup sequence, where the length of the arrays is equal to the number of senors. The sensor's ID is actually the slave address used during communication between the sensor and the computer. The sensor's name is important for the program, as it tells which sensor it communicates with. The program can then send the correct request command to the sensors. The last array is the status array, which contains the sensor cap quality in percentage. If this percentage value is less than 35 %, Hamilton recommends that the sensor cap should be replaced.

Beneath this, the additional page is presented. The page consists of five different tabs, where four of these are dedicated to each reactor. The user is presented with three control fields inside these reactor tabs, where the desired physical unit of each measured value can be selected. These are set as default values but may be changed if the user desires. The available units of measurement for the temperature are degrees Celsius, degrees Fahrenheit and Kelvin. This is actually the only change in the physical unit that the program fully supports. Even though there are four available units of measurement for the DO level, the program only supports DO measurements in mg/l ppm. The other three available units of measurement are %-sat, %-vol, and ug/l ppb. An extension of the program should include the support of the different physical units of the DO measurements. The only available unit of measure for the pH value is pH. The last tab of the page is dedicated to direct messaging. Here, the user may communicate with each sensor individually and request other information that is not presented inside the program. This will be useful for debugging the sensors.



Number of Sensors: 2

VISA COM Sensor: COM7

Desired Iterations SBR1: 1
Desired Iterations SBR2: 0
Desired Iterations SBR3: 0
Desired Iterations SBR4: 0

Sensor-ID	Sensor Name	Sensor Status
01	VISIFERM	100.000000 %
02	EasyFerm	100.000000 %

If any of the Sensor Status values are below 35%, Hamilton recommends that the corresponding sensor cap should be changed.

Desired Physical value DO Sensor:
Desired Physical Unit DO1: DO: mg/l ppm
Physical Unit DO1: mg/l ppm

Desired Physical value pH Sensor:
Desired Physical Unit pH1: pH: pH
Physical Unit pH1: pH

Desired Physical value Temp Sensor:
Desired Physical Unit Temp1: Temp: C
Physical Unit Temp1: °C

error in (Sensors)
status ✓ code 0
source

error out (Sensors)
status ✓ code 1073676294
source
VISA Read in Change_Physical_Unit_Sensors.vi->SBR_controll.vi

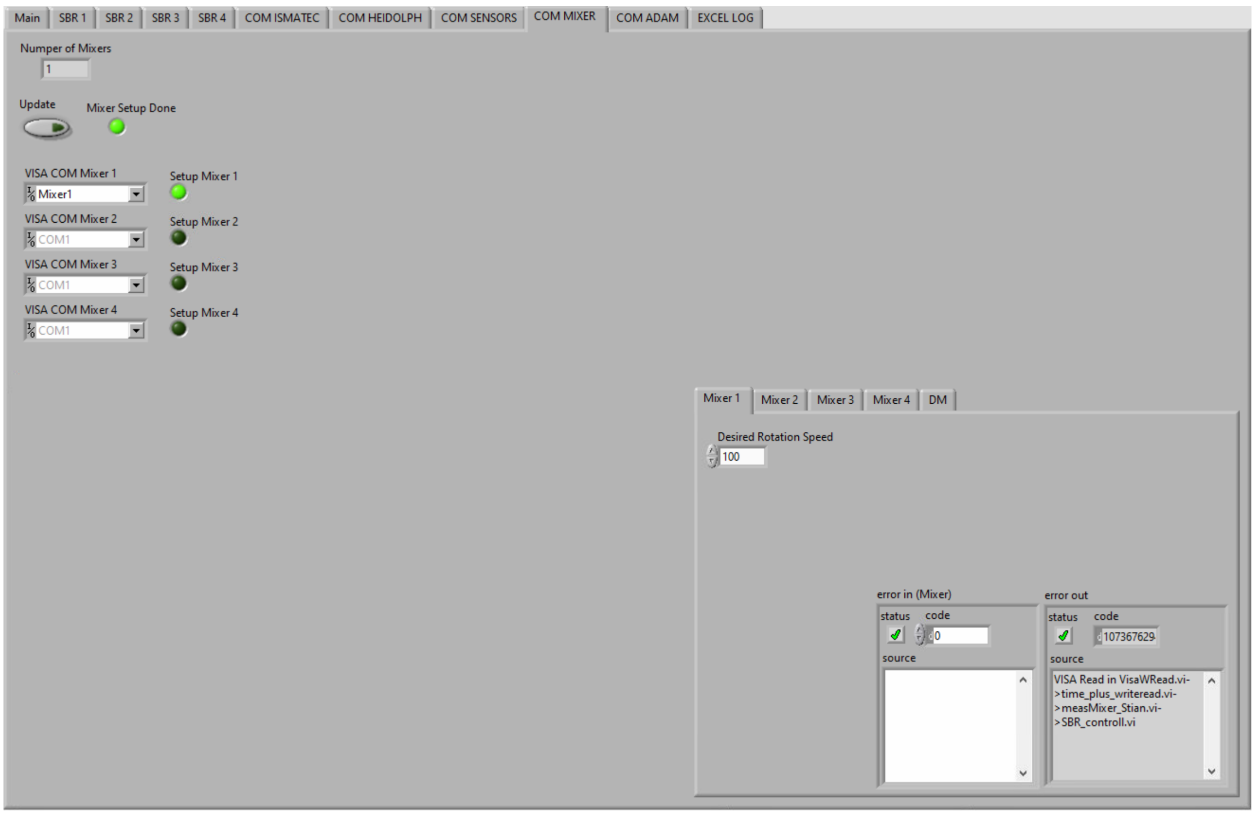
Figure 5.9: Overview of communication settings for the Hamilton sensors.

COM Mixer tab

This tab, shown in figure 5.10, is dedicated to the mixers and contains information regarding the IKA RET control-visc. The page's setup is similar to both the *COM Ismatec* and *COM Heidolph* pages, where general information is found at the upper left side of the page and an additional page containing five different tabs at the right-hand side of the page.

The general information consists of the selected COM-port for each connected magnetic mixer, indications showing if the setup was successful, and the number of mixers needed for the desired amount of active reactors. The COM-ports dedicated to each mixer are already set as a default value inside the program. Even if it should not be needed, the user may select other COM-ports for the mixers. In addition to this information, a speed update button is provided to the user. If the user wants to update the IKA RET control-visc with a new desired rotation speed without running the whole main setup sequence, this button can be pressed.

The additional page on the right-hand side of the *COM Mixer* tab is made to separate data between the four possible IKA RET control-visc. Four of the pages are named after which reactor the mixer corresponds to, as *Mixer 1-4*. Inside each of these tabs, the desired rotation speed can be selected. If not, this is set to a default value inside the program. The fifth page is dedicated to individual communication, giving the user the possibility to send direct messages to the desired mixer. This can be used to debug the mixer or check other values that are not shown on the other four pages.



Numper of Mixers: 1

VISA COM Mixer 1: Mixer1
VISA COM Mixer 2: COM1
VISA COM Mixer 3: COM1
VISA COM Mixer 4: COM1

Desired Rotation Speed: 100

error in (Mixer)
status ✓ code 0
source

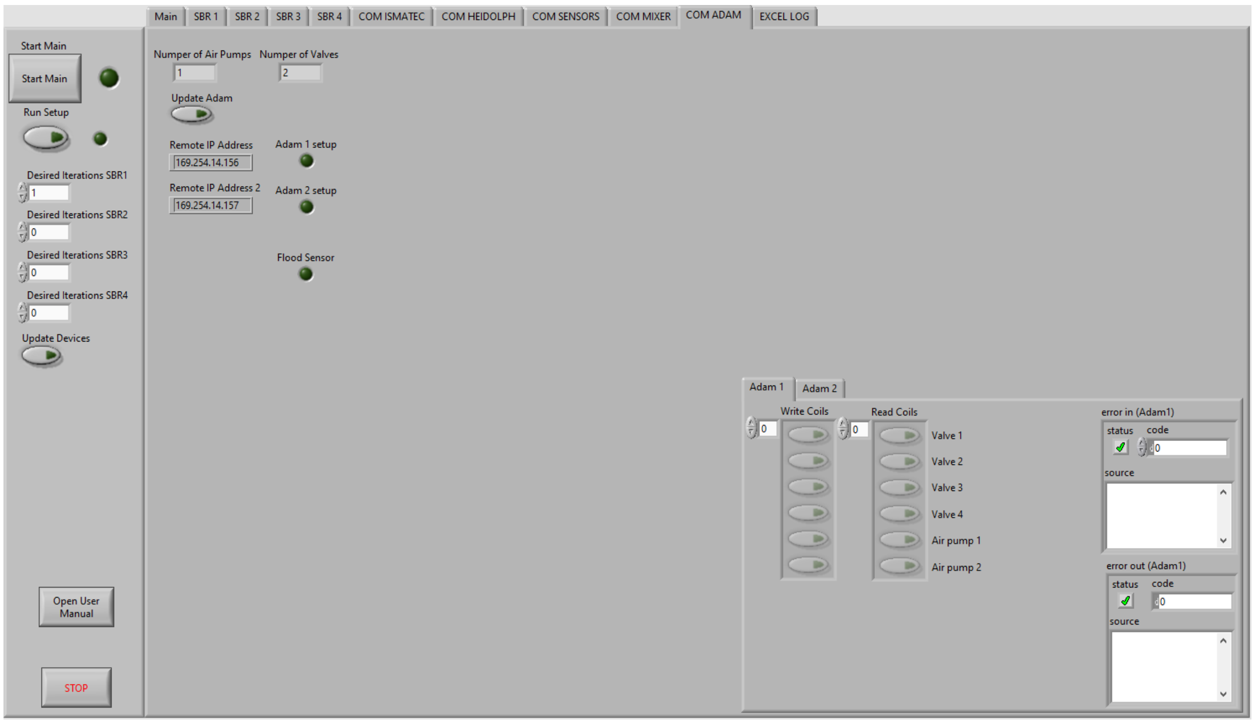
error out
status ✓ code 107367629
source
VISA Read in VisaWRead.vi->time_plus_writeread.vi->measMixer_Stian.vi->SBR_controll.vi

Figure 5.10: Overview of communication settings for the Reglo pumps.

COM Adam tab

This tab, shown in figure 5.11, is dedicated to the Adam-6066 power relay and contains information regarding this device. The page contains general information and an additional page that consists of two tabs.

The general information placed at the upper left corner of the page contains multiple indicators. Here, the number of air pumps and solenoid valves needed for the number of active reactors are shown, in addition to the remote IP address of the Adam's. The IP address for both Adams is set to a default value and is assigned at the program's start. Three lights are also provided to show the user if the setup was successful and if the flood sensor is triggered. The additional page contains which coils are active for both the Adam's. The two possible Adams are separated into two tabs.



Desired Iterations SBR1: 1
Desired Iterations SBR2: 0
Desired Iterations SBR3: 0
Desired Iterations SBR4: 0

Numper of Air Pumps: 1
Numper of Valves: 2
Remote IP Address: 169.254.14.156
Remote IP Address 2: 169.254.14.157

Write Coils: 0
Read Coils: 0

error in (Adam1)
status: ✓ code: 0
source

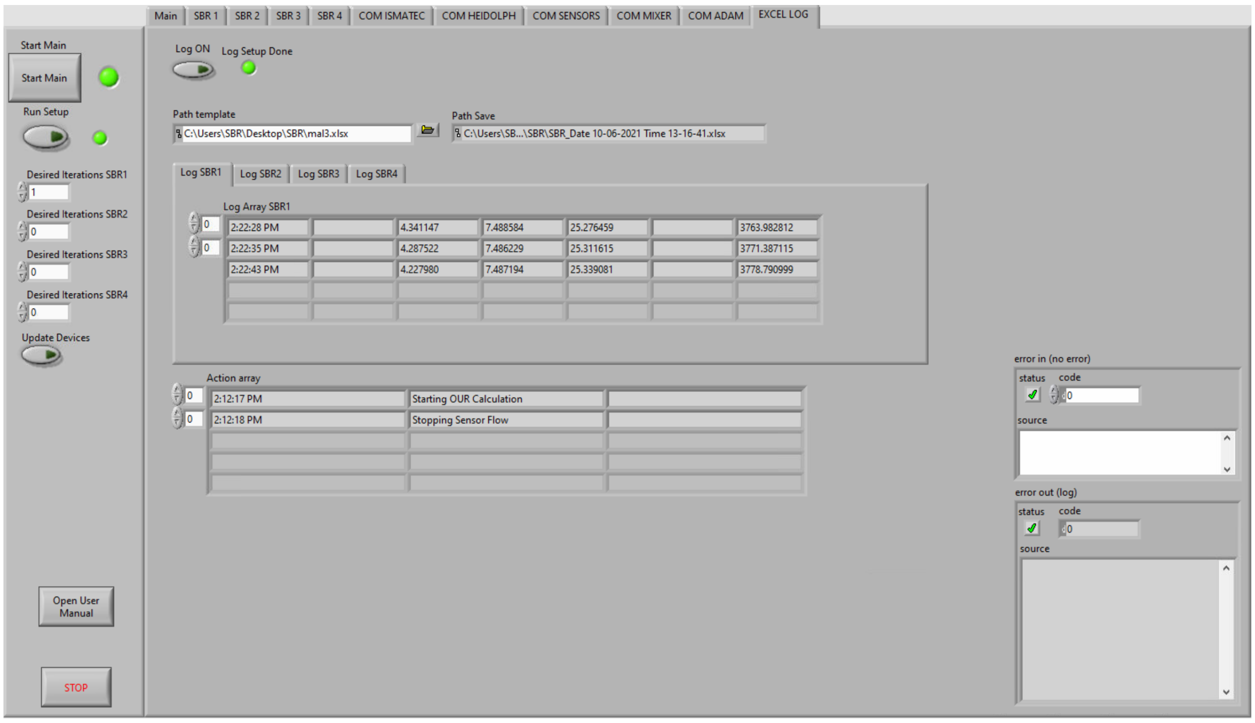
error out (Adam1)
status: ✓ code: 0
source

Figure 5.11: Overview of communication settings for the Reglo pumps.

Excel log tab

This tab, shown in figure 5.12, is dedicated to the logging function. All the data logged to the Excel file are first placed within multiple arrays and are written to Excel when the array's length reaches five. There are five different arrays, where four of these are placed within a new page placed at the middle of the "Log" tab. These four arrays consist of the sensor measurements, where the four arrays correspond to the four reactors. The fifth array is the action array, which is placed just beneath the measurement arrays. All actions inside the program, either if the user or the program does it, are logged into this array.

The tab also shows which template is used for the Excel file, at the *Path Template*. Next to this, the saved file name is shown, in addition to the path of the saved Excel file.



Desired Iterations SBR1: 1
Desired Iterations SBR2: 0
Desired Iterations SBR3: 0
Desired Iterations SBR4: 0

Path template: C:\Users\SBR\Desktop\SBR\mal3.xlsx
Path Save: C:\Users\SB...\SBR\SBR_Date 10-06-2021 Time 13-16-41.xlsx

Log Array SBR1 (index 0, 0)

2:22:28 PM		4.341147	7.488584	25.276459		3763.982812
2:22:35 PM		4.287522	7.486229	25.311615		3771.387115
2:22:43 PM		4.227980	7.487194	25.339081		3778.790999

Action array (index 0, 0)

2:12:17 PM	Starting OUR Calculation	
2:12:18 PM	Stopping Sensor Flow	

error in (no error)
status: ✓ code: 0
source:

error out (log)
status: ✓ code: 0
source:

Figure 5.12: Overview of communication settings for the Reglo pumps.

5.2 Program operations

This section will go through what happens behind the user's graphical interface and how the program itself is set up.

At the program's startup, default values are written to all the control parameters, except the stage parameters. These values are constant values that are saved within the program and are written to the control parameters every time the program starts. This ensures that the main sequence is not started without correct configuration and important parameters set, such as desired pump speeds and rotation direction, among other things. These default values can, of course, be changed by the user after the program has started if desired.

In addition to process safety, it also makes it easier for the user. It can be time-consuming and difficult to select the correct COM-ports and configuration settings for all the connected devices at every startup. The user only needs to set the desired stage parameters for each reactor before the setup, and the main sequence can start.

Once the program has started, its functionality can be split up into five individual operations. These operations are setup, flow cell initializing, main sequence, individual communication, and user log. Only the first four operations mentioned, setup, flow cell initializing, main sequence, and individual communication, are operations the user can operate. While the last, the user log operation, is something the program does by itself as long as the program is running.

5.2.1 Setup

The program is made with two different methods of setting up communication and control parameters for the different devices used in the process; main setup and individual setup.

- **Main Setup**: The main setup covers everything that needs to be done before the main sequence can start. This operation can start by either pressing the *Run Setup* button or, if the setup is not yet completed, by pressing the *Start Main* button. This operation goes through every connected device and configures it with the selected control parameters. The setup goes through the different devices in the following order:

  1. **Assign correct device numbers**: This first step is to assign the correct number of each device that is required for the number of reactors that are selected. These values tell the program how many of each device to configure in the setup.
  2. **Log file setup**: The log file setup opens an Excel template stored locally at the computer. After this, all five log arrays are created with the correct size and length. These arrays accept five logged strings before it is written to the Excel file. When the log file setup is complete, the program writes *Log Setup Done* to the action log array with a time stamp.
  3. **Hamilton Sensor setup**: This step opens a communication line with the selected amount of connected sensors through the assigned COM-port. It also checks for both the sensor ID and sensor name for each connected sensor. This will be used to distinguish the sensors and ensure that the correct sensor will be

used for each measurement(DO sensor to measure DO and pH sensor to measure pH). The program then checks the sensor cap status of each connected sensor and configures the sensor measurements with the desired physical unit. When this is successfully done, *Sensor Setup Done* is written to the action log array with a time stamp. It also writes the selected physical units of measure to the data log array.

4. **IKA RET control-visc setup**: The mixer setup opens communication with the connected mixers through the assigned COM-ports and then starts the scale. The weighting function takes a few seconds to start up. When this is successfully done, *Mixer Setup Done* is written to the log file with a time stamp.
5. **Heidolph pump setup**: This opens communication with the two connected Heidolph pumps through the assigned COM-ports. After this, it changes the display to *Speed rpm* and assigns the desired speed and pump direction. It also double-checks that the desired speed is set within the pumps. When this is successfully done for both pumps, *Heidolph Setup Done* is written to the log file with a time stamp. If there occurs an error in one of the pumps, it is written to the action log array which of the pumps are having problems.
6. **Ismatec Reglo pump setup**: This opens communication with the selected amount of connected Reglo pumps through the assigned COM-ports. The program then assigns the pump channels to the correct operation, with the desired speed and correct pump direction for each channel. Then it double-checks that the desired speed and direction are successfully sent to the pump and asks for the flow rate for each channel. It also checks the software version and name for each connected pump. When this is successfully done, *Reglo Setup Done* is written to the action log array with a time stamp.
7. **Adam setup**: This opens communication with the selected number of active Adam-6066 through the assigned IP address. If the number of active reactors is above two, both Adams are needed and will then be configured. Once the communication line is opened, all the relays are shut since none of the solenoid valves or air pumps should be active at this stage. When Adam 1 is successfully set up, *Adam 1 Setup Done* is written to the action log array with a time stamp. If Adam 2 is active and successfully set up, *Adam 2 Setup Done* is also written to the action log array.
8. **Successful setup**: This last step checks if all the stages of the setup were complete. If this is the case, and the setup was complete, the main sequence is ready to begin.

- **Individual setup**: The individual setup allows the user to re-configure each device type individually. If any new parameters are set to just one device, instead of running the whole *Main setup* sequence, an individual setup can be used to write the newly set control parameter. This operation can also be used for debugging one single component without needing to run the main setup each time. The individual setup runs the same operations regarding the selected device as the *Main setup* does.

5.2.2 Initialize flow cells

The initialize flow cell operation is intended for readying the flow cell pairs connected to the SBR. At the program startup, the fluid volume inside the flow cells might be low or empty. This may be because air has migrated into the flow cells due to long idle periods or because the fluids have been pumped out during an earlier process. Ideally, when the main sequence is running, the flow cells should be filled up. The initialization may be done individually for all the flow cell pairs. Pairs, meaning the two flow cells connected to the same reactor.

Inside every SBR page in the LabView program, a box is dedicated to this operation corresponding to the selected reactor. When the *Start* button is pressed, the connected Ismatec pump will run the flow cell channels the opposite way as intended when running the main sequence.

The initialization will need a bit of manual help from the user to fill the flow cells. First, the user must unscrew the top screw of the flow cell that contains the pH sensor, opening the seal. When wastewater fluids are pumped into the flow cell, the air inside will migrate out the top of the flow cell.

After loosening the top screw of the flow cell, the fluid level inside will start to rise. At this step, there might be spilled wastewater. When the flow cell starts to overflow, the top cap of this flow cell can be tightened again. Now that the first flow cell is filled up, the flow cell containing the DO sensor can be filled. This is done by loosening the flow cell from the griper that holds it and then rotate the flow cell so that the outlet is pointing upwards. The excess air inside the flow cell will then be transported into the reactor as wastewater starts to fill the flow cell. When there is no excess air left inside the second flow cell, the initialization is complete. The flow cell containing the DO sensor should be placed back into its gripper and on top of the IKA lab disc.

The *Stop* button inside the dedicated flow cell initialization box should now be pressed. This will stop the fluid circulation and re-configure the Ismatec pump with the correct speed and channel direction.

For more information on how to fill the flow cells with wastewater before starting the main sequence, see the user manual in appendix H.

5.2.3 Main sequence

The main sequence consists of all the necessary stages to complete an SBR process and is started by pressing the *Start Main* button in the LabView program. The main sequence then starts if, and only if, the main setup sequence has been done and was successfully completed. This ensures that all devices are set up properly before the process can begin. If this is fulfilled, the main sequence will run through the following program stages; idle, fill, react, waste, sample, settle, and decant. These seven stages will now be explained.

Idle stage

This stage is the first stage for each reactor. Here, all selected control values and methods are written to the program. Desired iterations are also checked here. After one iteration, the reactor is set back to the idle stage. If the desired iterations are greater than one, a new process is started.

Only one reactor at a time may proceed to the next stage, the filling stage. There is only one pump dedicated to filling all reactors, and thus only one reactor can operate in this stage at a time. If there are multiple active reactors, the other reactors will be held at the idle stage until the first reactor has completed the filling stage. Once the first reactor goes onto the next stage, the second reactor may proceed to the filling stage, and so on for the other active reactors.

Filling stage

The next stage is the filling stage, which will be the first stage where things happen physically. With the use of solenoid valves, fluids are directed into the desired reactor by the filling pump.

The following device operations are done at the filling stage:

- **Heidolph filling pump**: The Heidolph filling pump will start at 120 rpm and transport substrate into the reactor.

- **Adam-6066 & Solenoid valves**: For the fluids to be transported to the correct reactor, the corresponding solenoid valve for filling the reactor is opened by the Adam-6066 module.

- **RET control-visc**: The IKA RET control-visc starts the scaling function to ensure control over the filling volume.

At this stage, the measured weight of the added fluids is the most important value and will be the controlling variable. Once the filling volume reaches 50 ml below the desired filling volume, the speed of the Heidolph pump is lowered to 40 rpm. This is to ensure that the desired weight is reached without overfilling the reactor. Once the setpoint volume is reached, the filling stage of this reactor is completed. Before the program moves on to the next stage, the operating devices are stopped. The Heidolph pump is stopped, the opened solenoid valve is closed, and the weighting function of the RET control-visc is stopped.

Reaction stage

The next stage for the reactor is the reaction stage, which will be the longest. There are three different control methods of this stage, which the user selects. The three control methods are OUR mode, reactor mode, and sequential mode.

Even though there are three different modes, there are some similarities between the control methods. The same devices will be used in all the modes, where the difference is in how they are controlled with active and inactive periods. The following devices will be active throughout the process, regardless of which mode is selected;

- **IKA RET control-visc**: The IKA RET control-visc will start rotation with the desired speed at the beginning of this stage and be active throughout the stage.
- **IKA lab disc**: The magnetic stirrer discs placed beneath the flow cells will be active throughout the process.

The other devices in use at this stage are the Ismatec Reglo ICC pumps, the Hamilton sensors, and the air pumps. These devices will be controlled differently for the different control methods.

- **OUR mode**: When the OUR mode is selected, the process will be controlled by OUR estimations. Before the main sequence start, the user has selected control parameters such as minimum OUR, maximum reaction time, the interval between calculations, and upper and lower DO levels inside the flow cells. The user may change these control parameters at any time during the process.

  The reaction stage can be split into two different phases when the OUR mode is selected; OUR estimation and circulation. The program will jump between these two phases until the reaction stage is complete. The Ismatec pump and the Hamilton sensors will operate differently in these two phases, while the air pump will be constant ON throughout the process.

  The following operations will happen when the reaction stage is at its circulation phase:

  - **Ismatec Reglo ICC pump**: The Ismatec pump will start the two channels that are assigned to the fluid loop that contains the flow cells. Now, the fluids inside the flow cells will be replaced with the wastewater inside the reactor.
  - **Hamilton sensors**: The two Hamilton sensors that are connected to the reactor will measure at an interval of 5 seconds. Since the air pump is on and the Ismatec pumps circulate the reactor's wastewater, the DO level inside the flow cells will increase. The measured DO level, pH level, and temperature are plotted into the *Sensor Data Graph.*
  - **Control**: Two parameters control this phase of the reaction stage, the maximum DO level and the maximum interval between calculation. The circulation phase is completed when one of the two control parameters is reached. This is if either the time past is longer than the maximum set interval between calculations or if the measured DO level inside the flow cell is greater than the maximum DO level. When the circulation phase is complete, the reactor may proceed to the OUR estimation phase.

The following operations will happen when the reaction stage is at its OUR estimation phase:

- **Ismatec Reglo ICC pump**: The Ismatec pump will stop the two running channels.
- **Hamilton sensors**: The two Hamilton sensors that are connected to the reactor will continue to measure at the same frequency. Since the fluid circulation has now stopped, the DO level of the wastewater inside the flow cells will decrease due to no air supply and growth of the bacteria. The measured DO level, pH level, and temperature will continue to be plotted into the *Sensor Data Graph*. In addition, the development of the measured DO level and pH level will be plotted into separate graphs, the *OUR Calculation Graph* and the *pH graph*.
- **Control**: The OUR estimation phase will be controlled by one control parameter, which is the minimum DO level set by the user. Once the measured DO level inside the flow cell is less than this value, the OUR can be estimated. The estimation is based on the *OUR Calculation Graph* data, where linear regression is used on this data. The result of the linear regression is the estimated OUR and will be added to the *OUR development graph*. Once the OUR has been estimated, the OUR calculation phase is complete.

After each OUR estimation, the program checks this value against the minimum allowed OUR. If the estimated OUR is above the minimum OUR, the reactor will go back to the circulation phase. However, if it is below, the reactor will proceed to the next stage of the main sequence, the waste stage.

If the estimated OUR does not go below the minimum allowed OUR, the maximum reaction time will control the reaction stage. When the time elapsed since starting the reaction stage reaches the maximum reaction time, the reactor will proceed to the waste stage. Regardless of any estimated OUR values or which of the two phases the program currently operates in.

- **Reactor mode**: If the reactor mode is selected, the reaction stage is controlled by elapsed time. The user has selected a reaction time before the main sequence started, which will be the control parameter for the duration of the reaction stage. The other control parameters that will control this mode are the interval between measurements and upper and lower DO levels. The user may change these control parameters at any time during the process.

  The program will run through the following operations when the reactor mode is selected:

  - **Hamilton sensors**: The interval between measurements will control when measurements are to be carried out. The sensors will then measure the DO level, the pH level, and the temperature of the fluid content of the flow cells at these intervals. The sensors will take five measurements, with five seconds between each measurement. Once the five measurements are done, the sensors stay idle until the interval time has elapsed again. All the sensor measurements are plotted into the *Sensor Data Graph*.

  - **Ismatec Reglo ICC pump**: Before the sensors measure the fluid content of the flow cells, the reactors' wastewater is circulated into the flow cells. The circulation will start a quarter of the selected interval time before the sensor measurements are done, or by a minimum of 20 seconds before. The Ismatec Reglo ICC pump will then start its two channels assigned to the fluid loop containing the flow cells. Once the five sensor measurements are complete, the pump stops.
  - **Air pump**: The air pump is controlled by the two parameters; upper DO level and lower DO level. When the measured DO level is below the lower DO level, the air pump starts. The air pump will stay ON until the measured DO level is greater than the upper DO level.

  The program will continue with the operations that are just described until the desired reaction time is reached. When the time elapsed since the reaction stage start is greater than the desired reaction time, the reactor will proceed to the next stage, the waste stage.

- **Sequential mode**: If the sequential mode is selected, the reaction stage is controlled by elapsed time. The user has selected a reaction time before the main sequence started, which will be the control parameter for the duration of the reaction stage. The other control parameters that will control this mode are the interval between measurements, air supply interval, and pause interval. The user may change these control parameters at any time during the process.

  The program will run through the following operations when the sequential mode is selected:

  - **Hamilton sensors**: The interval between measurements will control when measurements are to be carried out. The sensors will then measure the DO level, the pH level, and the temperature of the fluid content of the flow cells at these intervals. The sensors will take five measurements, with five seconds between each measurement. Once the five measurements are done, the sensors stay idle until the interval time has elapsed again. All the sensor measurements are plotted into the *Sensor Data Graph*.
  - **Ismatec Reglo ICC pump**: Before the sensors measure the fluid content of the flow cells, the reactors' wastewater is circulated into the flow cells. The circulation will start a quarter of the selected interval time before the sensor measurements are done, or by a minimum of 20 seconds before. The Ismatec Reglo ICC pump will then start its two channels assigned to the fluid loop containing the flow cells. Once the five sensor measurements are complete, the pump stops.
  - **Air pump**: The air pump is controlled by the two parameters; air supply interval and pause interval. The reactor will receive air supply at the duration of the air supply interval value, and then the air supply will stop at the duration of the pause interval value.

  The program will continue with the operations that are just described until the desired reaction time is reached. When the time elapsed since the reaction stage start is greater than the desired reaction time, the reactor will proceed to the next stage, the waste stage.

Waste stage

The next stage for the reactor is the waste stage. The waste stage is controlled by one control parameter, which is the desired waste volume.
The program will run through the following operations at the waste stage:

- **IKA RET control-visc**: The RET control-visc will continue stirring at the same speed, and the scale function will remain active. The weighting module will measure the outgoing fluids. Having both the stirring and weighting modules active at the same time will result in some noise on the weight measurements due to movement from the magnetic stirrer.

- **Ismatec Reglo ICC pump**: The Ismatec pump will be used to transport the wastewater out of the reactor with one assigned channel. This channel will start at the desired speed selected by the user and run until the desired volume is pumped out of the reactor.

When the desired volume is pumped out of the reactor, the waste stage is complete. The reactor will proceed onto the next stage, the sample stage.

Sample stage

The next stage for the reactor is the sample stage. The sample stage is similar to the waste stage, where the desired sample volume will control the stage.

The program will run through the following operations at the sample stage:

- **IKA RET control-visc**: The RET control-visc will continue stirring at the same speed as it did in the reaction sequence. In addition to the stirring function being active, the weighting module will also be active to control the volume of fluids that are pumped out. Having both these operations active at the same time will result in some noise on the weight measurements due to movement from the magnetic stirrer.

- **Ismatec Reglo ICC pump**: The Ismatec pump will be used to transport the wastewater out of the reactor with one assigned channel. This channel will start at the desired speed selected by the user and run until the desired volume is pumped out of the reactor.

When the desired volume is pumped out of the reactor, the sample stage is complete. The reactor will proceed onto the next stage, the settle stage.

Settle stage

The next stage for the reactor is the settling stage. This stage is controlled by one parameter set by the user, the desired settle time. This stage will mainly be a waiting stage, where the organisms inside the reactor sink to the bottom of the reactor. This is to ensure that the decanted fluids, at the next stage, are mainly processed wastewater.

At the start of this stage, all devices are set to inactive. Then the program waits until the time elapsed since the start of the stage is greater than the desired settle time. When the desired settle time is reached, the program proceeds to the decanting stage.

Decanting stage

The decant stage is the finishing stage of the main sequence. This stage will be similar to the filling stage but with the opposite pump direction. In addition, as it is for the filling stage, only one reactor may be in this stage at a time. This is because there is only one pump dedicated to the decanting of all the reactors. However, this will rarely be a problem since there will already be a delay between the reactors from the filling stage. With the use of solenoid valves, the correct reactor is emptied. The decanting stage is controlled by one parameter, the desired decant volume.

The following device operations are done at the decanting stage:

- **Heidolph emptying pump**: The Heidolph emptying pump will start at 120 rpm and transport processed wastewater out of the reactor.
- **Adam-6066 & Solenoid valves**: For the fluids to be transported out of the correct reactor, the corresponding solenoid valve for emptying the reactor is opened by the Adam-6066 module.
- **RET control-visc**: The IKA RET control-visc measures the weight of the reactor and will have control over the decanted volume.

At this stage, the measured weight of the decanted fluids is the most important value and will be the controlling variable. Once the decanting reaches 50 ml above the desired decant volume, the speed of the Heidolph pump is lowered to 40 rpm. This is to ensure that the desired weight is reached without emptying the reactor too much. Once the setpoint volume is reached, the decanting stage of this reactor is completed. Before the program moves on to the next stage, the operating devices are stopped. The Heidolph pump is stopped, the opened solenoid valve is closed, and the weighting function of the RET control-visc is stopped. Once the decanting stage is complete, the reactor is sent back to the idle stage.

Flow chart

A simplified flow chart of the main process is illustrated in figure 5.13. The different control methods are illustrated in figure 5.14, figure 5.15, and figure 5.16. This overview gives a visual insight into how the process described earlier works and how the program runs.

SBR PROCESS FLOW CHART

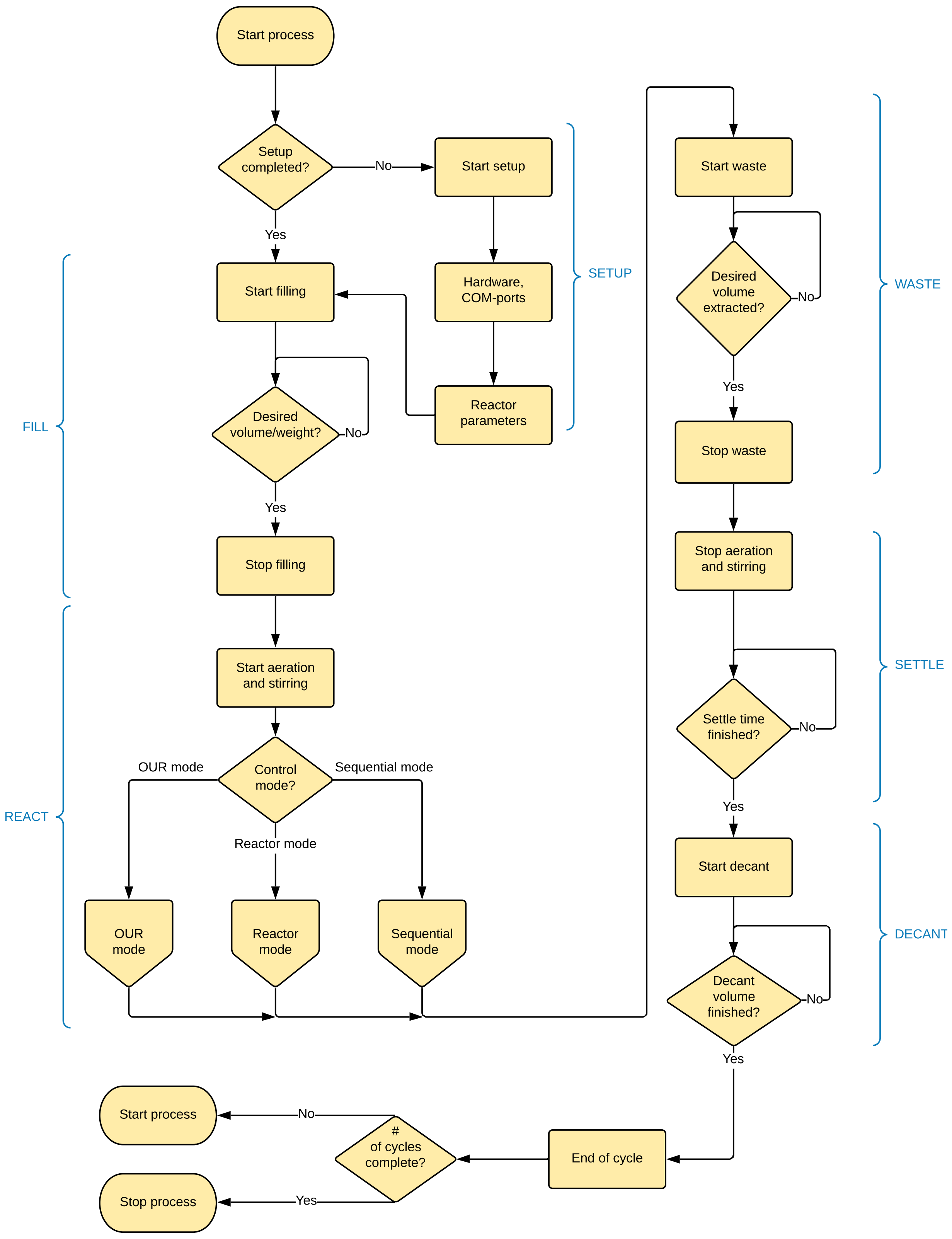


Figure 5.13: Main flow chart of the SBR process.

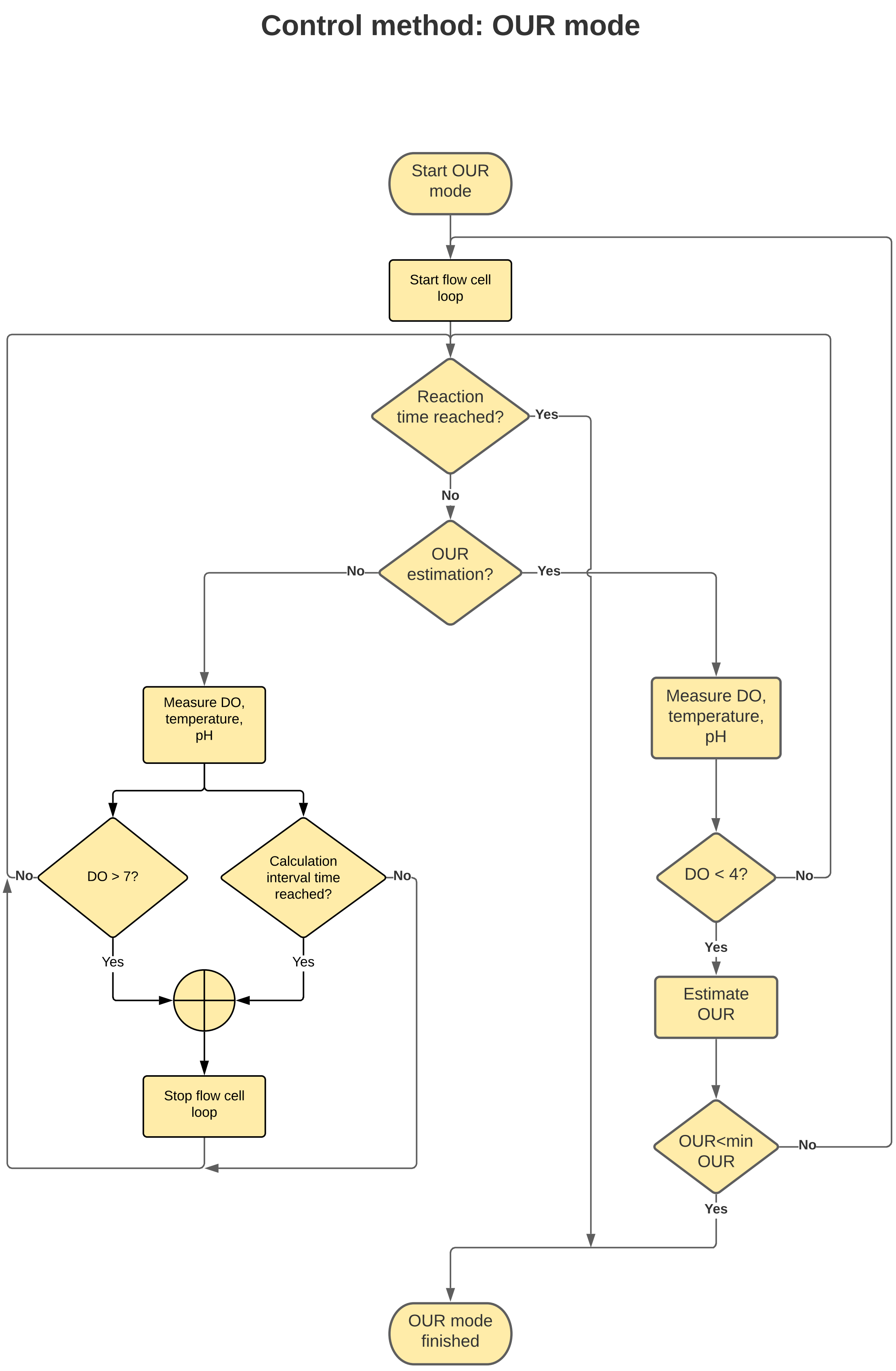


Figure 5.14: Control method - OUR mode flow chart.

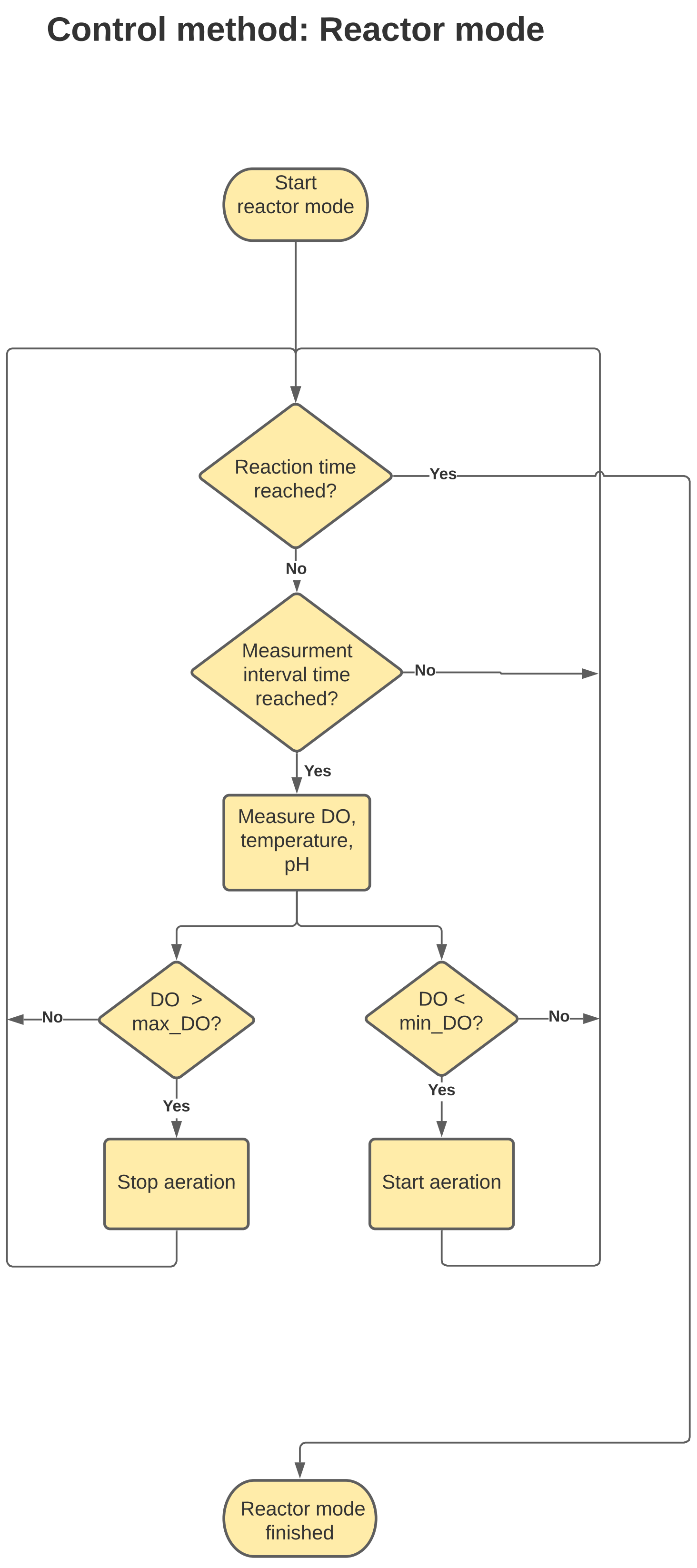


Figure 5.15: Control method - Reactor mode flow chart.

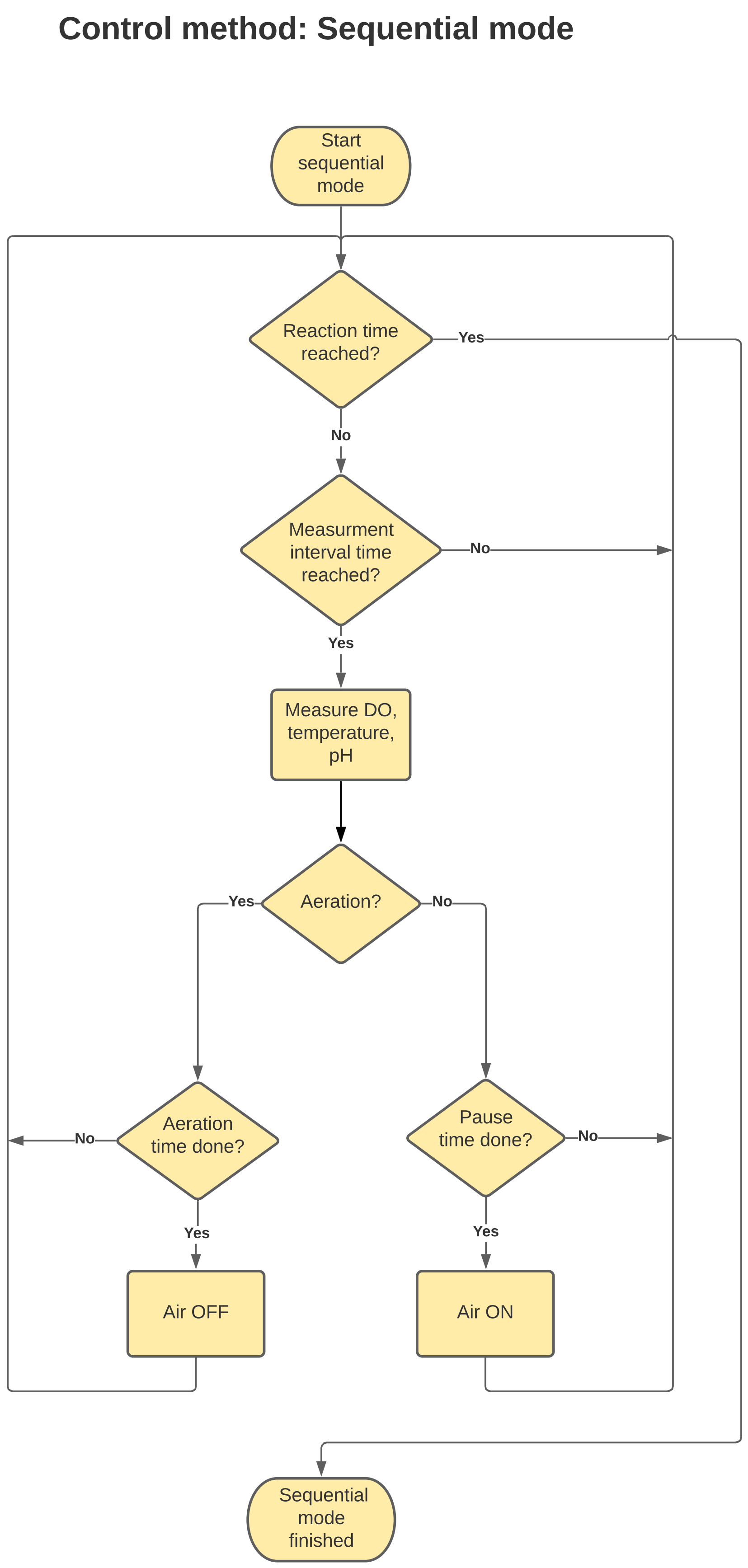


Figure 5.16: Control method - Sequential mode flow chart.

5.2.4 Individual communication

Within the program, the user will have the possibility to communicate with each device individually. This can be useful for setting parameters individually for each device or for debugging. Individual communication will only be available when the main sequence is not running. Only three of the connected devices support the individual communication operation. These are the Ismatec pumps, the IKA RET control-visc, and the Hamilton sensors.

Within the device tabs for each of these devices, there is a DM tab. These tabs consist of the parameters that are needed to communicate with a device. For the user to successfully communicate with a device, the correct COM port needs to be assigned, followed by a command in which the device supports. Most of the commands are already made and are available for the user to send. For the Hamilton sensors DM tab, an additional field gives the user the possibility of manually writing in the command string.

5.2.5 Log

The LabView program contains a log-part in which measured values and actions are logged throughout the program into an Excel spreadsheet. The Excel spreadsheet consists of five tabs, four for the four possible reactors and one tab for the different actions done.

The logging operation is initialized at the start of the setup sequence and saved to the computer. The given name of the Excel file consists of *SBR* followed by the date and time in which the setup started. An example of the file name will be: *SBR Date 10-06-2021 Time 16-42-17.xlsx.* If autosave is turned on, as it should, the file will be saved continuously throughout the process. The file is also saved at the end of the process.
The measured values tab will mainly be used during the reaction phase for each reactor. Here, the DO level, pH level, and temperature value will be logged to the Excel tab. This allows the user to study the data from the process further.

In addition to the measured values, all actions are logged to the Excel file. Both the actions done by the user and by the program will be logged and timestamped. Examples of user actions can be starting the main sequence, running the setup sequence, or changing control parameters such as different volumes or the number of reactors in use. Examples of program actions can be the what stage the different SBR processes goes into or actions done by a device such as waste pump start/stop or stirring start/stop. These logged actions allow debugging after a process is done in case something went wrong. Possible errors can both be user or program fault, and enlightening this can show issues that need further development in the program.

6. Verification

Internal verification has been carried out to check that the control program, the user interface, and the physical setup function as intended. The following tests are done with biological sludge gathered from another active SBR at the biological lab. The substrate used for all tests is peptone, which is also given to us by the biologists at the laboratory. The peptone is mixed with water, resulting in a COD of 400 mg/l.

Since we only have one full SBR setup, the verification tests are limited to this. Therefore, we are not able to test the functionalities of multiple active SBR. The control program is, however, developed to support the use of four active SBR.

An explanatory video of the SBR process with OUR control mode is made and can be found in a appendix I. Also, a user manual is made, where the user can read how to use the program and configure and calibrate Hamilton sensors, see appendix H.

Physical setup for verification test

Figure 6.1 shows how our physical setup looks like at the biology laboratory. The physical setup for one full SBR consists of two Heidolph pumps, one IKA RET control-visc, one Duran reactor, one Ismatec Reglo ICC pump, one EHEIM air pump, two flow cells containing two Hamilton sensors, the control cabin, and the solenoid valve assembly box.

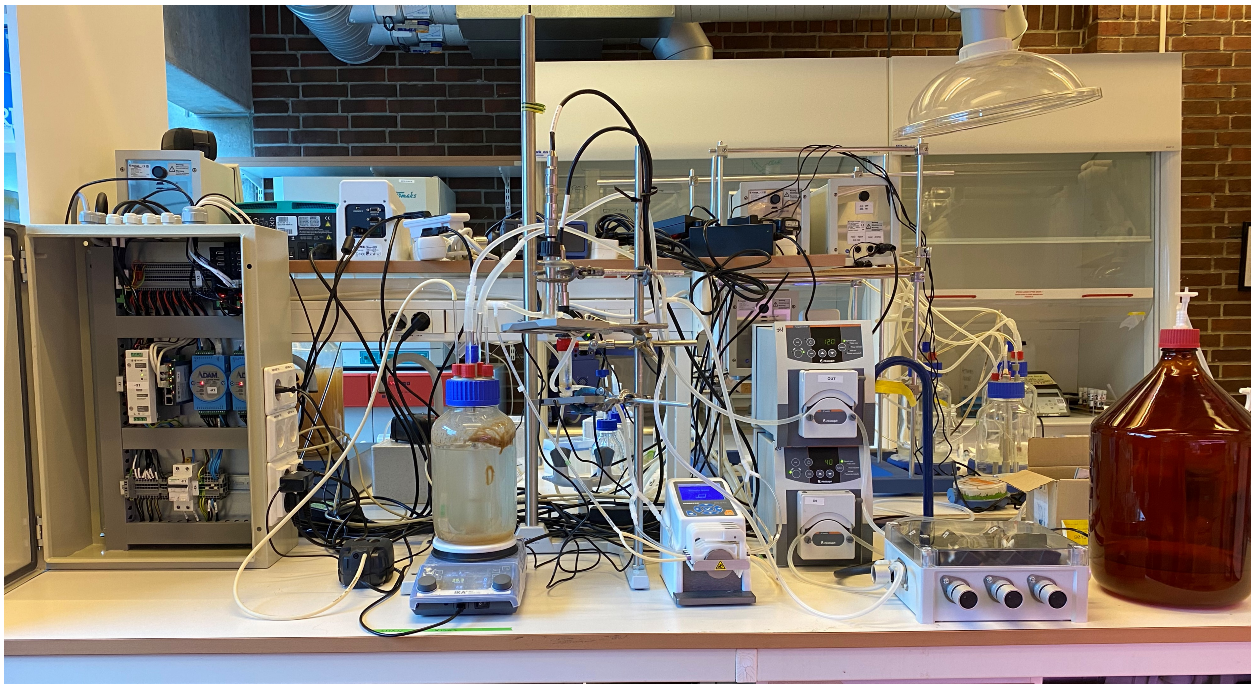
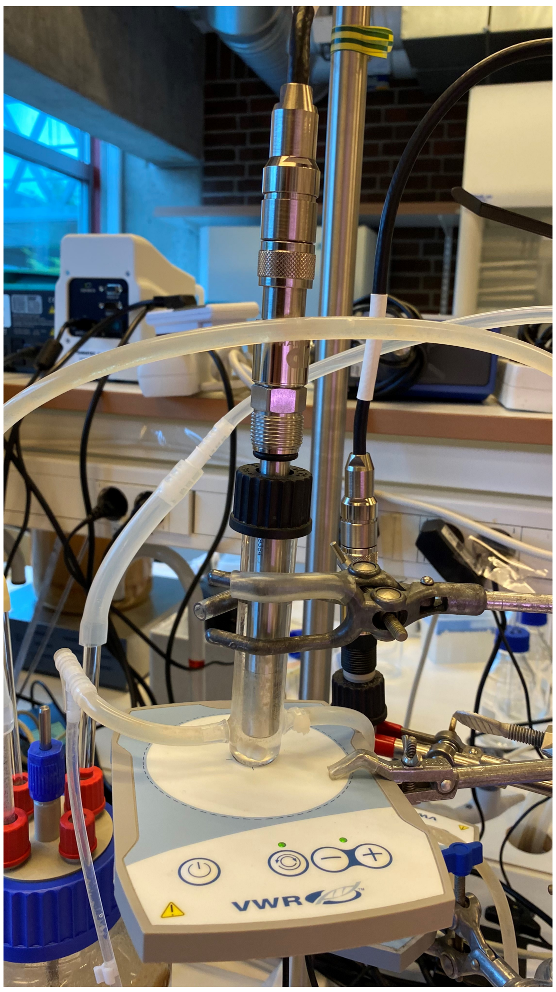


Figure 6.1: Image of a full reactor setup at the biology laboratory.

The figures 6.2a and 6.2b show how the Hamilton sensors are mounted into the flow cells and the smaller mixers mounted under the flow cells.


(a) DO sensor placed within a flow cell.

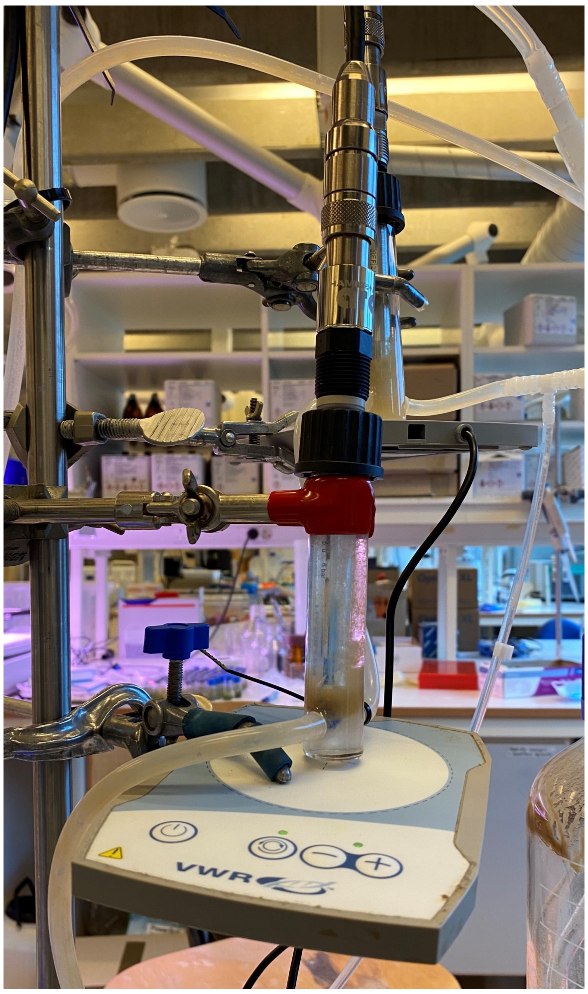

(b) pH sensor placed within a flow cell.

Figure 6.2: Hamilton sensors placed into flow cells.

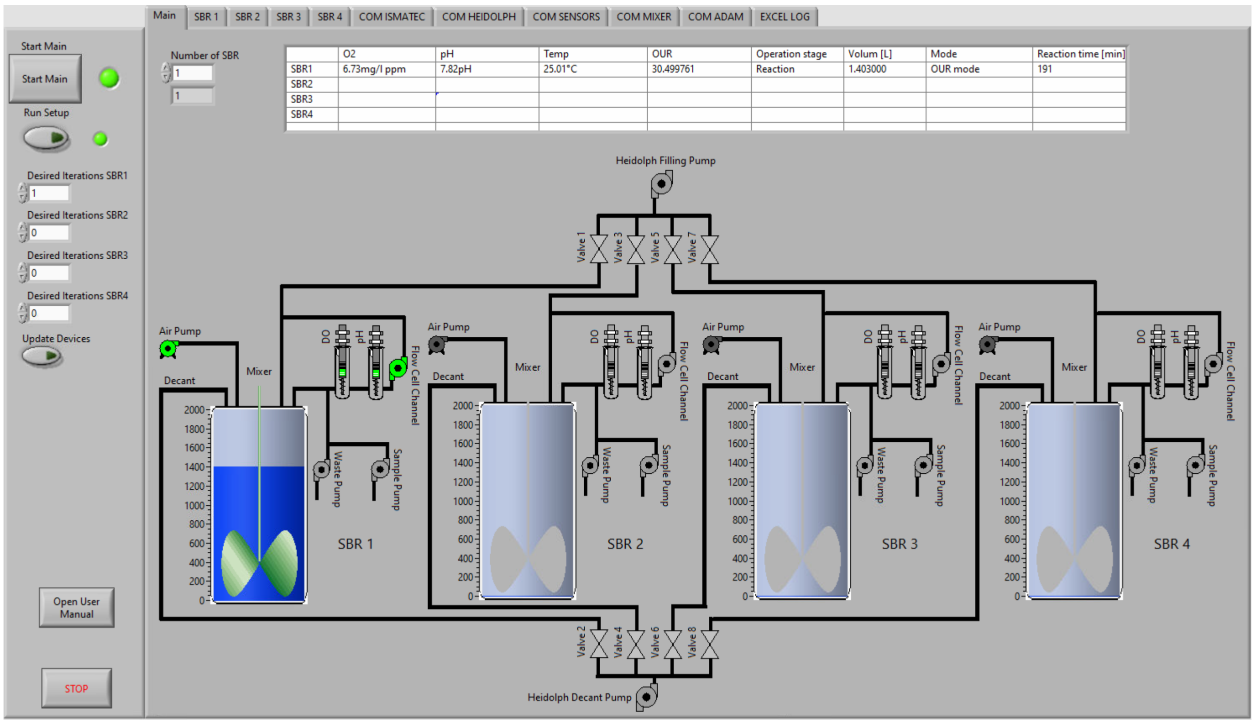
6.1 Main tab

This verification step will cover the visual part of the program, the GUI. This Main tab is intended to be the tab that is used the most. It was desired that the user gets enough information regarding all the active processes and devices to have a good overview of the process.

Figure 6.3 show the Main tab during the reaction stage of a process, running only one reactor. The figure clearly shows the active devices. When comparing reactor one to the other three reactors, it is easy to see that the magnetic stirrer, the flowcell loop, and the sensors are active. In addition, we see that the reactor has been filled with 1400 ml. This is also checked physically, where the presented information matches with the activity of the components.

The table at the top of the tab shows the last measured data from the sensors, which stage the reactor is in, the filled volume, the control method of the reaction stage, and the duration of the reaction stage.

As a result of this verification test, we see that the main tab works as intended when running one reactor.



	O2	pH	Temp	OUR	Operation stage	Volum [L]	Mode	Reaction time [min]
SBR1	6.73mg/l ppm	7.82pH	25.01°C	30.499761	Reaction	1.403000	OUR mode	191
SBR2								
SBR3								
SBR4								

Figure 6.3: *Main*-tab during reaction mode.

6.2 Control methods

Multiple control methods for the reaction stage of the reactors have been made, which provides the user with more options when running an SBR process. These control methods are OUR mode, reactor mode, and sequential mode. Verification tests have been carried out to ensure that the three control methods work and do change the control of the reaction stage. These verification tests will only look at how the control program behaves and not the biological aspect of the process.

6.2.1 OUR mode

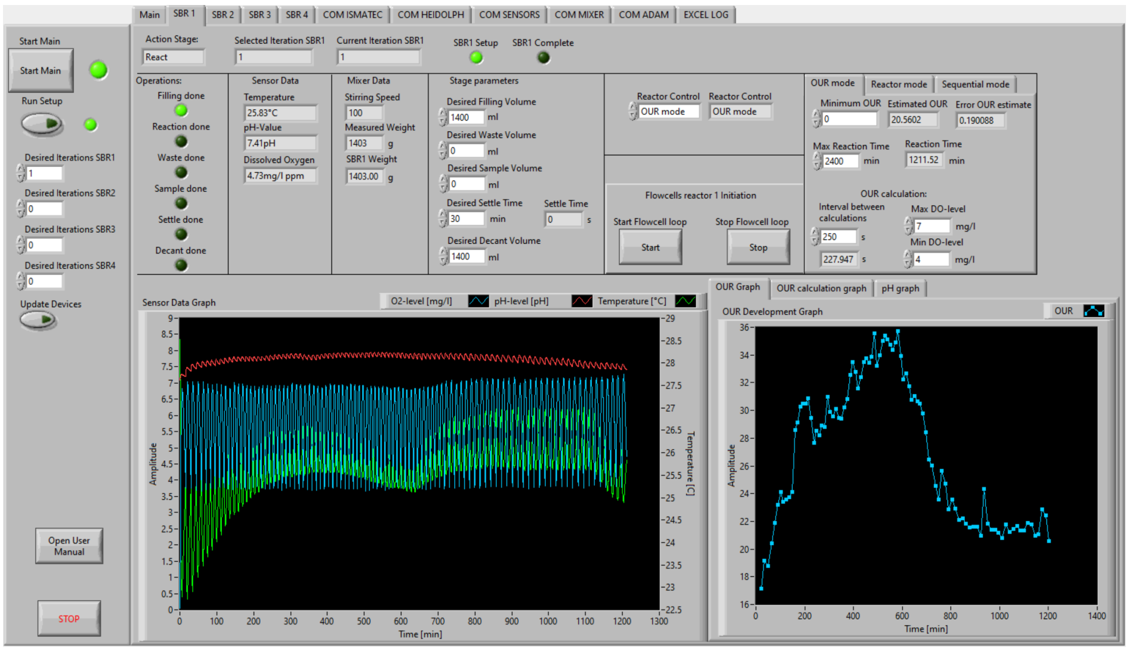
Figure 6.4 show the *SBR1* tab during a process. Here, OUR mode is selected as the control method. The data provided in this tab tells us that the reactor has been filled with 1400 ml of substrate and has been running for 1212 minutes.

The control parameters for the control method are set as follows:

- **Minimum OUR = 0**
- **Max Reaction time = 2400 min**
- **Interval between calculations = 250 s**
- **Max DO-level = 7 mg/l**
- **Min DO-level = 4 mg/l**

When the minimum OUR is set to 0, the reaction stage is only controlled by the max reaction time, which in this case is 2400 min. In addition, in the scenario from figure 6.4, the interval between calculations was not controlled by the set control parameter of 250 seconds. The indicator block below the interval between calculation control shows that the interval only lasted 228 seconds. This is a result of the DO level rising above the max DO level, 7 mg/l in this case.

When you look at the *Sensor Data Graph*, you see that all the measured values vary quite a lot. This is because measurements are done when the fluid loop is circulating and when it is not circulating. This affects all the measurements and resulting in the oscillation you see in the plotted data. Further, the sensor data graph shows that the control parameters controlling the DO levels seem to be working as intended. The plotted blue line shows the measured DO levels, where the DO level oscillates between the user's two control parameters; max DO level and min DO level. When the DO level rises, it stops right after the 7 mg/l mark. It overshoots a bit since the DO level keeps rising a little bit even if the fluid loop is inactive. When the DO level drops, it stops at the 4 mg/l mark.



Action Stage: React
Selected Iteration SBR1: 1
Current Iteration SBR1: 1
SBR1 Setup
SBR1 Complete

Desired Iterations SBR1: 1
Desired Iterations SBR2: 0
Desired Iterations SBR3: 0
Desired Iterations SBR4: 0

Operations: Filling done, Reaction done, Waste done, Sample done, Settle done, Decant done

Sensor Data
Temperature: 25.83°C
pH-Value: 7.41pH
Dissolved Oxygen: 4.73mg/l ppm

Mixer Data
Stirring Speed: 100
Measured Weight: 1403 g
SBR1 Weight: 1403.00 g

Stage parameters
Desired Filling Volume: 1400 ml
Desired Waste Volume: 0 ml
Desired Sample Volume: 0 ml
Desired Settle Time: 30 min
Settle Time: 0 s
Desired Decant Volume: 1400 ml

Reactor Control: OUR mode
Reactor Control: OUR mode

Flowcells reactor 1 Initiation
Start Flowcell loop
Stop Flowcell loop

Minimum OUR: 0
Estimated OUR: 20.5602
Error OUR estimate: 0.190088
Max Reaction Time: 2400 min
Reaction Time: 1211.52 min
OUR calculation:
Interval between calculations: 250 s
227.947 s
Max DO-level: 7 mg/l
Min DO-level: 4 mg/l

Figure 6.4: *SBR1*-tab during OUR mode.

The figures 6.5 and 6.6 show how the DO levels and pH levels develops during a calculation phase. Both these values are decreasing during this phase, as one should expect. This is a result of the bacteria consuming the DO and substrate, resulting in lower levels of both the DO level and pH level in the wastewater. Figure 6.5 also shows that the calculation phase stops when the measured DO level gets below 4 mg/l. After the calculation phase is done, the red line is plotted in the *OUR Calculation Graph*. The red line is a result of linear regression and fits the measured DO levels well. The red regression line in figure 6.5 is the result of an estimated OUR value of 22.4.

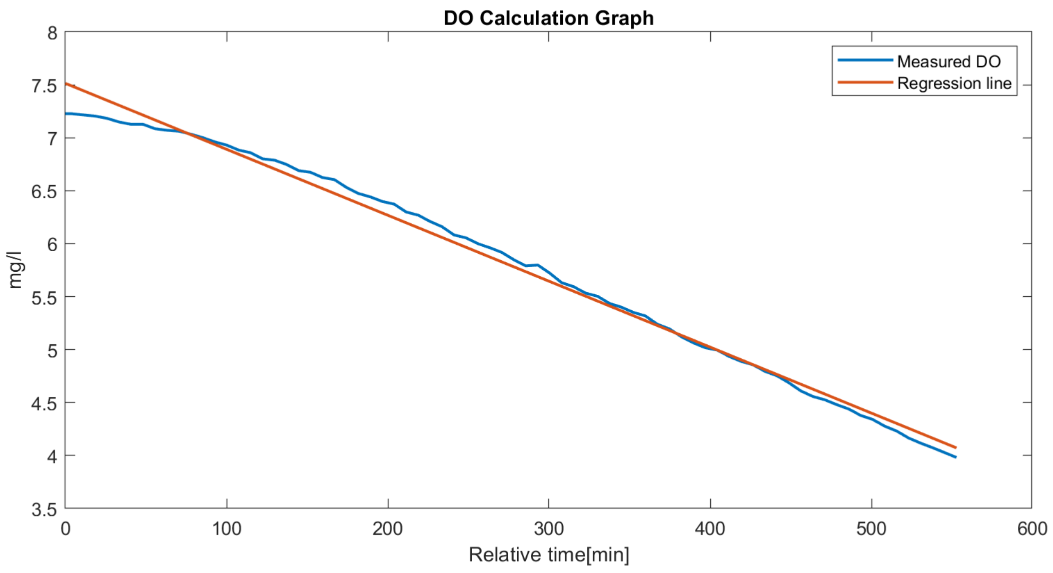


Figure 6.5: DO development inside flowcells during calculation phase. The plotted data is gathered form the Excel log.

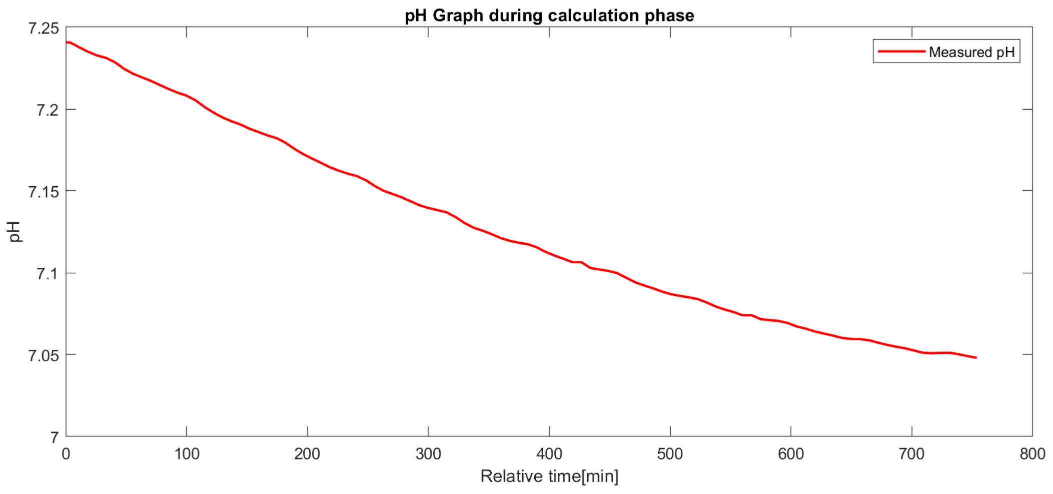


Figure 6.6: pH development inside flowcells during calculation phase. The plotted data is gathered form the Excel log.

After each OUR estimation, the estimated OUR is added to the *OUR Development Graph*. This OUR development data is shown in figure 6.7. The result is as expected, where the estimated OUR increases at the start of the process and started to drop after a while as the bacteria consume the substrate.

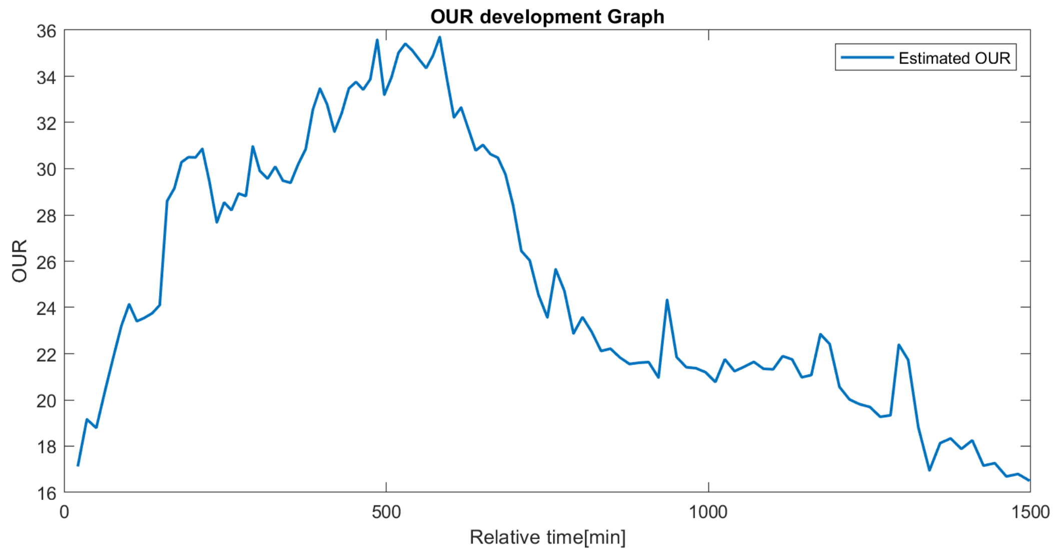


Figure 6.7: Estimated OUR development during a test done over 43 hours. The plotted data is gathered form the Excel log.

6.2.2 Reactor mode

Figure 6.8 show the *SBR1* tab during a process. Here, reactor mode is selected as the control method. The data provided in this tab tells us that the reactor has been filled with 1400 ml of substrate and has been running for 1141 minutes.

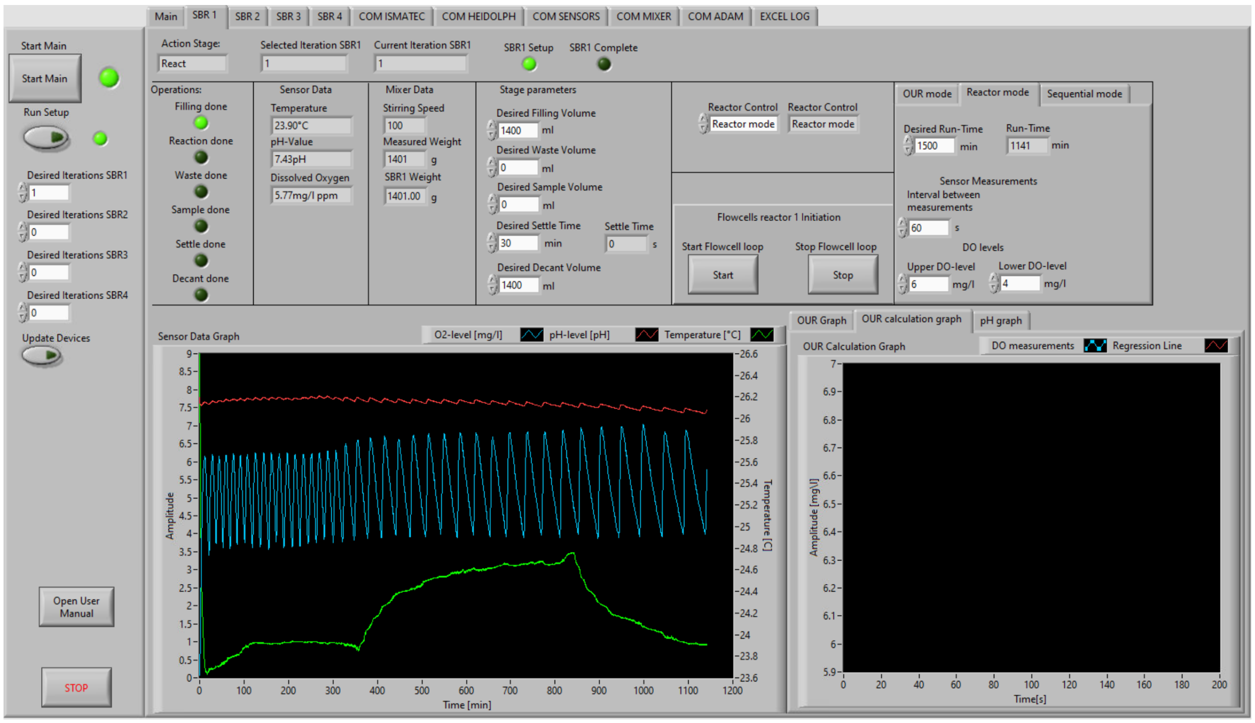
The control parameters for the control method are set as follows:

- **Desired Run time = 1440 min**
- **Interval between measurements = 80 s**
- **Upper DO-level = 6 mg/l**
- **Lower DO-level = 4 mg/l**

Looking at the *Sensor Data Graph*, where the blue line shows the DO measurements, the control parameters controlling the DO-level works as intended. The DO measures stay between the user's two control parameters: upper DO-level and lower DO-level. When the DO-level rises, it stops around the 6 mg/l mark which is the selected upper DO-level. When the DO level drops, it stops around the 4 mg/l mark which is the selected lower DO-level. Oscillations occur because the air supply is turned ON and OFF when reaching the upper and lower DO-level. This behavior is as desired.

The *Sensor Data Graph* also shows that the duration in which the DO-level drops increases. This results from lower levels of substrate in the wastewater, thus a lower growth rate for the bacteria, which is as expected. The amount of substrate also affects the pH level of the wastewater. Looking at the Sensor data graph, you see that the measured pH level slowly decreases. This is because the added substrate has a higher pH level than the treated wastewater. When the bacteria consume this, the pH level decreases.

The last measurement is the temperature. From the sensor data graph, you can see that the temperature changes during the process. The time of day, sunlight, and room temperature will affect the measured temperature, and therefore it varies throughout the process.



Action Stage: React
Selected Iteration SBR1: 1
Current Iteration SBR1: 1

Desired Iterations SBR1: 1
Desired Iterations SBR2: 0
Desired Iterations SBR3: 0
Desired Iterations SBR4: 0

Sensor Data
Temperature: 23.90°C
pH-Value: 7.43pH
Dissolved Oxygen: 5.77mg/l ppm

Mixer Data
Stirring Speed: 100
Measured Weight: 1401 g
SBR1 Weight: 1401.00 g

Stage parameters
Desired Filling Volume: 1400 ml
Desired Waste Volume: 0 ml
Desired Sample Volume: 0 ml
Desired Settle Time: 30 min
Settle Time: 0 s
Desired Decant Volume: 1400 ml

Reactor Control: Reactor mode
Reactor Control: Reactor mode

Desired Run-Time: 1500 min
Run-Time: 1141 min
Sensor Measurements
Interval between measurements: 60 s
DO levels
Upper DO-level: 6 mg/l
Lower DO-level: 4 mg/l

Figure 6.8: *SBR1*-tab during a reaction stage, with reactor mode as control method.

6.2.3 Sequential mode

Figure 6.9 show the SBR1 tab during a process. Here, sequential mode is selected as the control method. The data provided in this tab tells us that the reactor has been filled with 1400 ml of substrate and has been running for 734 minutes.

The control parameters for the control method are set as follows:

- **Desired Run time = 1500 min**
- **Interval between measurements = 40 s**
- **Air supply interval = 450 s**
- **Pause interval = 100 s**

Even though the air supply interval and pause interval was set as mentioned above at the time of figure 6.9, they were not this value throughout the reaction process. They were initially set to the following:

- **Air supply interval = 300 s**
- **Pause interval = 200 s**

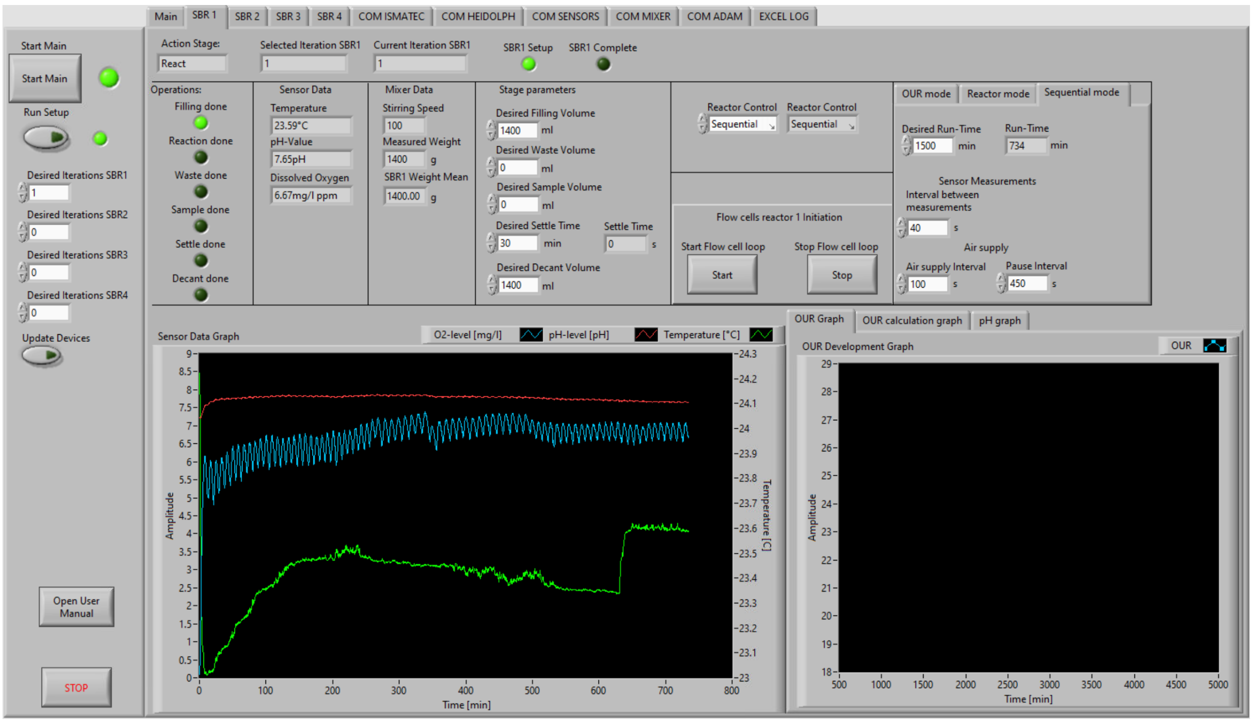
These values were changed to the first-mentioned values after the reaction stage had lasted about 300 minutes.

The sequential control method is designed to have on and off regulation of the air supply, which will result in DO oscillations throughout the reaction stage. The *Sensor Data Graph* in figure 6.9, where the blue line shows the DO measurements, clearly shows that the air supply has been turned on and off during the process. The drop in DO level that occurs after 300 minutes is a result of adjusting the parameters. The DO level kept rising, going above or desired oscillation area, which caused us to adjust the parameters a bit.

The temperature measurements shown in the *Sensor Data Graph* change a bit during the process. This results from the time of day, sunlight, and room temperature will affect the measured temperature, and therefore it varies throughout the process. However, the suddenly steep rise at approximately 630 minutes of reaction time results from the user manually removing some of the sludge inside the flow cells.

The last value plotted in the *Sensor Data Graph* is the pH level. As we can see, this measurement slowly decreases through the process. This is as expected since the bacteria consume the substrate, which contains a higher pH level than the wastewater initially does.

Since air supply control, with its supply and pause intervals, is the primary control of this method we can determine that the control method functions as intended.



Action Stage: React
Selected Iteration SBR1: 1
Current Iteration SBR1: 1

Desired Iterations SBR1: 1
Desired Iterations SBR2: 0
Desired Iterations SBR3: 0
Desired Iterations SBR4: 0

Sensor Data
Temperature: 23.59°C
pH-Value: 7.65pH
Dissolved Oxygen: 6.67mg/l ppm

Mixer Data
Stirring Speed: 100
Measured Weight: 1400 g
SBR1 Weight Mean: 1400.00 g

Stage parameters
Desired Filling Volume: 1400 ml
Desired Waste Volume: 0 ml
Desired Sample Volume: 0 ml
Desired Settle Time: 30 min
Settle Time: 0 s
Desired Decant Volume: 1400 ml

Reactor Control: Sequential
Reactor Control: Sequential

Desired Run-Time: 1500 min
Run-Time: 734 min
Sensor Measurements
Interval between measurements: 40 s
Air supply
Air supply Interval: 100 s
Pause Interval: 450 s

Figure 6.9: *SBR1*-tab during a reaction stage, with sequential mode as control method.

6.3 User log

As previously mentioned, a logging function is added to the control program. This logging function stores all sensor measurements(DO, pH, and temperature) for each reactor and all action done by both the program and the user. This logging function uses an Excel file to store all the data.

The logging function works as intended. The whole Excel log from the OUR mode verification test that was carried out in section 6.2.1 can be found in appendix F.

7. Limitations and further improvements

7.1 Limitations

Due to delays of shipping when ordering new devices for our system, we have not had the possibility to verify the control program against multiple active reactors. During this project, we have only had access to one complete reactor setup. All functionalities regarding the use of one single reactor have however been tested, and successfully verified.

The control program that have been made, along with the control cabinet, are made to function with four active reactors. However, this have not been tested. As a result of this, we can not fully verify that the developed system is fully functional with four complete reactor setups connected.

The reactor trays to secure the lab for overflood are not installed due to the delays on the equipment. Thus the flood sensors are not mounted but planned for in the control program and the control cabinet. It is therefore easy to mount the flood sensors later in the future.

7.2 Further developments of the physical setup

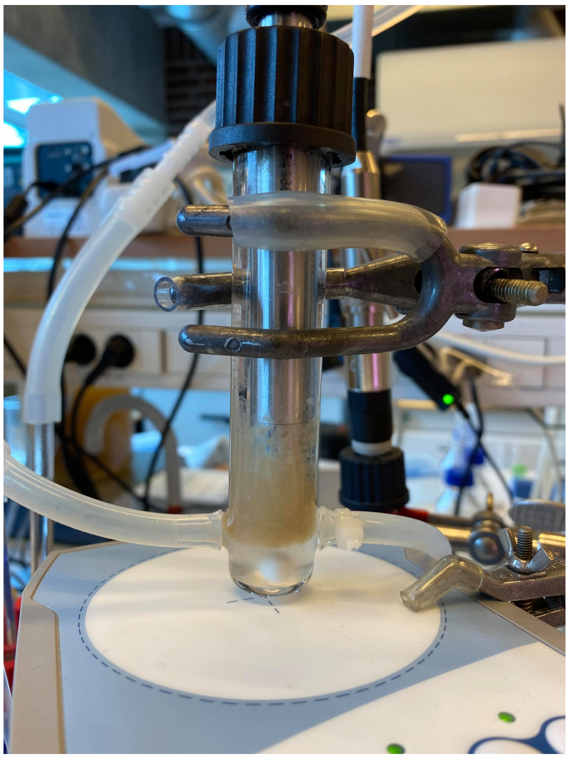
During testing and verification, we have discovered some further developments of the physical setup. Due to time limitations, we have not been able to implement this our self. However, it should be considered to further development the physical setup of the system.

Flowcell improvements

Practical tests showed trouble with sludge building up inside the flowcells during operation. When there are large enough quantities of sludge inside the flowcells, the measurements are no longer representative of the reactor's fluid. Figure 7.1a and 7.1b shows how the flowcells look after some hours of operation.

An improvement might be redesigning the flowcells. Now, both the inlet and outlet might be too small, which reduces the flow rate at the fluid loop. In addition to making both the inlet and outlet bigger, these should not be placed directly opposite each other. Instead, the inlet should be placed at the top of the flowcell with the outlet at the same place it is now. With this modification, the sludge that builds up at the middle of the flowcell will be pushed down by the circulating fluid.

In addition, by increasing the flow rate, the time it takes to replace the fluid inside both flowcells will be reduced.


(a) Sludge that builds up in the DO flowcell

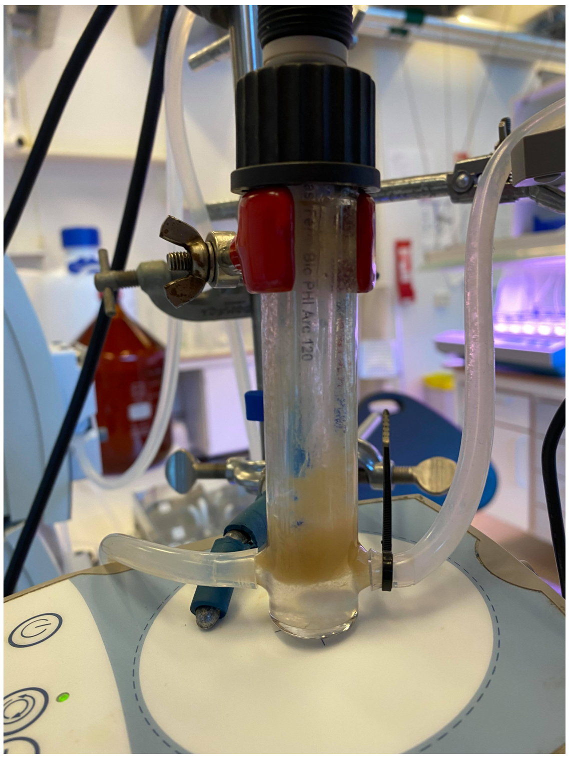

(b) Sludge that builds up in the pH flowcell

Figure 7.1: Sludge inside the flowcells.

Add a sampling carousel

Another improvement of the physical setup is adding a sampling carousel. By adding such a device, samples can be taken throughout the reaction process. The only sample option that the control program supports now is a sample at the end of the SBR process. If a sampling carousel were to be added, the control program would require improvements as well.

Add a pump for inflow/addition of a acidic or basic liquid.

Since the growth of bacteria is dependent on the pH level of the wastewater, an additional pump for inflating an acidic or basic liquid might be desired. By adding this to the setup, the user is provided with jet another control possibility. The new pump may easily be connected to the main fluid loop of each reactor.

7.3 Further developments of the user interface and control program

In addition to the physical setup, some further developments can be done for the user interface and control program.

Extend the program to support multiple physical units of measure

Today, the user is provided with the option of changing the physical units on the measurements received from the Hamilton sensors. However, if the physical unit of the DO measures is changed from mg/l ppm, the program will not function properly. This is because all the control parameters regarding the DO level uses mg/l ppm.

The extension should support all the possible physical units that the sensors support. If a new physical unit is selected for one of the measurements, all control parameters regarding this measurement should be changed to support the selected physical unit.

Add temperature control of the reactor fluid

The IKA RET control-visc used for both the weight operation and magnetic stirrer operation also supports heating. This could be added to the software to support heating of the reactor fluid.

Hamilton Sensor improvements

Additional improvements can be made regarding the Hamilton sensor interface.

Implement sensor functions in LabView

Hamilton has their own software for configuration, setup, and calibration of the sensors. All the functions that their software support has not been added to the control program developed in this project. If the sensors need to run a new calibration, the user needs to use Hamilton's software.

A potential improvement of the control program should support all the functions that Hamilton's own software supports. Then the user of our system does not need to use any third-party software.

Self addressing when new sensors are connected

Self addressing of the sensors should also be included in the control program. Now, if the user connects a new sensor to the setup, Hamilton's software is required to change the address of the sensor.

Adding a self-addressing function to our software would make it easier for the user when adding new devices to the system. The self-addressing function should be a single button for the user to press every time a new sensor is connected. If this button is pressed, the program should assign a correct address to the newly connected sensor depending on how many sensors are already connected and which reactor the new sensor belongs to.

Weight control improvements

Some improvements might also be done regarding the weight control. Today, with the used weight scale, the system tare the scale. This means that the measured weight are actually a relative weight, relative to the weight of the reactor when the setup sequence is initiated.

An improvement might be to measure the absolute zero of the reactor. This meaning the weight of the reactor when there is no fluids inside. Then, before initiating the setup sequence, the reactor should be lifted off the magnetic stirrer and placed on the bench. Now, the relative weight is actually zero. After the setup sequence is complete, the reactor can be placed back on. For every weight measurement done after this, the weight of the reactor without fluids should be subtracted from the measured weight. The result of this will be the actual weight of the total fluid volume.

Bibliography

[1] George Tchobanoglous, Mohammad Abu-Orf, Gregory Bowden, William Pfrang, Metcalf & Eddy, and AECOM. *Wastewater engineering : treatment and resource recovery.* McGraw-Hill Education, New York, 5th ed. edition, 2014.

[2] AES Arabia Ltd website. Activated sludge process. `http://www.aesarabia.com/activated-sludge-process/`. (Accessed on 08/03/2021).

[3] IWA Publishing. Sequencing batch reactor. `https://www.iwapublishing.com/news/sequencing-batch-reactor`. (Accessed on 18/02/2021).

[4] IWA Publishing. Activated sludge process. `https://www.iwapublishing.com/news/activated-sludge-process`. (Accessed on 24/03/2021).

[5] Z. Hu and D. Grasso. Water analysis | chemical oxygen demand. In Paul Worsfold, Alan Townshend, and Colin Poole, editors, *Encyclopedia of Analytical Science (Second Edition)*, pages 325–330. Elsevier, Oxford, second edition edition, 2005.

[6] VWR product page. Reactor, stirred, gls 80 thread, duran®. `https://uk.vwr.com/store/product/4631660/reactor-stirred-gls-80-thread-duran`. (Accessed on 27/04/2021).

[7] Schott website. Duran® borosilicate glass tubing. `https://www.schott.com/en-no/products/duran`. (Accessed on 27/04/2021).

[8] Hamilton Bonaduz AG. Visiferm do arc 120 h2 specification sheet. `https://www.hamiltoncompany.com/process-analytics/product-specs/49991`. (Accessed on 17/02/2021).

[9] Hamilton Bonaduz AG. Easyferm bio phi arc 120. `https://www.hamiltoncompany.com/process-analytics/product-specs/50376`. (Accessed on 17/02/2021).

[10] Yokogawa Electric Corporation website. Benefits of sensors. `https://www.yokogawa.com/special/sensing-technology/usage/benefits-of-sensors/`. (Accessed on 21/04/2021).

[11] Hamilton Bonaduz AG. ph and orp arc sensors operating instructions. `https://craft-sensors.s3.amazonaws.com/File-Uploads/624263_Manual_pH-ORP_ArcSensors_EN_LR.pdf?mtime=20190424054048&amp;focal=none`. (Accessed on 23/03/2021).

[12] EHEIM GmbH & Co. KG. Instruction manual - eheim air pump 200. `https://eheim.com/media/pdf/8f/89/1b/3701-3704_airpump_01-21.pdf`. (Accessed on 12/03/2021).

[13] Mrclab website. What is a peristaltic pump. https://www.mrclab.com/what_is_a_peristaltic_pump. (Accessed on 18/02/2021).

[14] Verderliquids BV website. How do peristaltic pumps work| technology explained. https://www.verderliquids.com/int/en/how-do-peristaltic-pumps-work/. (Accessed on 18/02/2021).

[15] Cole-Parmer Instrument Company. Reglo icc operating manual. https://pim-resources.coleparmer.com/instruction-manual/14-036-e-ismatec-reglo-icc-english-revc.pdf. (Accessed on 03/02/2021).

[16] Ismatec. Ismatec reglo peristaltic pumps data sheet. http://www.ismatec.com/download/documents/4189_Ismatec_Reglo_SS.pdf.

[17] Fischer Scientific product page. Heidolph pd 5201 pump drive image.

[18] Imlab. Ika ret control-visc manual. https://www.imlab.eu/sites/default/files/generated/bestanden/Manual-EN-RET-control-visc-Ika-Imlab.pdf. (Accessed on 05/03/2021).

[19] IKA. Data sheet lab disc white. https://www.ika.com/ika/datasheet/datasheet.php?iProduct=3907500&iLang=1&iCountry=276&filename=DataSheet_lab_disc_white.pdf.

[20] Emerson website. Pinch solenoid valve series s106 image. https://www.emerson.com/resource/image/5969410/portrait_ratio1x1/555/555/73f107514f2bd0484c97b9652934f48/KZ/asco-series-s106-solenoid-valve-jpg.jpg.

[21] ISO International Organization for Standardization. Iso - iso 868:2003 - plastics and ebonite — determination of indentation hardness by means of a durometer (shore hardness). https://www.iso.org/standard/34804.html, March 2003. (Accessed on 24/04/2021).

[22] Emerson website. Pinch solenoid valve series s106 datasheet. https://www.emerson.com/documents/automation/datasheet-pinch-solenoid-valve-series-s106-asco-en-5376674.pdf.

[23] Elko AS. Elko s1091 pr datasheet. https://proff.elko.no/getfile.php/13139447/produkter/doc/EK005744/Datasheet1511861.pdf.

[24] Elko AS. Elko s1014 pt datasheet. https://proff.elko.no/getfile.php/13126166/produkter/doc/EK004968/Datasheet1510542.pdf.

[25] Advantech website. Image of adam-6066. https://advdownload.advantech.com/productfile/PIS/ADAM-6066/Product%20-%20Photo(B)/ADAM-6066_01_B20190725162725.jpg. (Accessed on 13/06/2021).

[26] Advantech website. Adam-6066 datasheet. https://advdownload.advantech.com/productfile/PIS/ADAM-6066/Product%20-%20Datasheet/ADAM-606620180910102020.pdf. (Accessed on 18/03/2021).

[27] Advantech website. Eki-2525li industrial ethernet switch datasheet. https://advdownload.advantech.com/productfile/PIS/EKI-2525LI/file/EKI-2525LI_DS(20191015)20191016105154.pdf. (Accessed on 08/04/2021).

[28] EXSYS Vertriebs GmbH. Ex-1177hmv usb hub manual. https://www.exsys-shop.de/shopware/media/pdf/19/4d/77/handbuch_manual_ex-1177hmv.pdf. (Accessed on 22/03/2021).

[29] RS Components. Rs pro power supply specifications. https://docs.rs-online.com/26bd/A700000007350002.pdf. (Accessed on 20/05/2021).

[30] Schneider Electric. Acti 9 ic60 rcbo datasheet. https://eref.se.com/no/en/web-product-data-sheet/product-pdf/A9D37216. (Accessed on 23/03/2021).

[31] E-T-A Website. Types of mcb trip curves. https://info.e-t-a.com/ES-2017-02-IndustrialAutomation-DCurve_LP-Destination.html. (Accessed on 21/05/2021).

[32] Phoenix Contact. The ultimate question – are ferrules necessary? https://dam-mdc.phoenixcontact.com/asset/156443151564/3518fd35347bc477bc23aaef7e7bc16e/7904.pdf.

[33] Prysmian Group website. Cable app. https://no.prysmiangroup.com/cableapp.

[34] Jan Axelson. *Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems.* Lakeview Research, Madison, 2007.

[35] Anusha Electronics Hub. Rs232 protocol – basics. https://www.electronicshub.org/rs232-protocol-basics/, July 2017. (Accessed on 21/04/2021).

[36] Texas Instruments. Interface circuits for tia/eia-232-f design notes. https://www.ti.com/lit/an/slla037a/slla037a.pdf?ts=1619546460113&ref_url=https%253A%252F%252Fwww.google.com%252F, September 2002. (Accessed on 01/05/2021).

[37] Manny Soltero, Jing Zhang, Chris Cockril, Kevin Zhang, Clark Kinnaird, and Thomas Kugelstadt. Rs-422 and rs-485 standards overview and system configurations - texas instruments. https://www.ti.com/lit/an/slla070d/slla070d.pdf, May 2010. (Accessed on 03/02/2021).

[38] Thomas Kugelstadt. The rs-485 design guide - texas instruments. https://www.ti.com/lit/an/slla272d/slla272d.pdf?ts=1623524777960, February 2008. (Accessed on 11/02/2021).

[39] Thomas Kugelstadt. The rs-485 design guide - application report. https://www.ti.com/lit/an/slla272d/slla272d.pdf?ts=1621960993370&ref_url=https%253A%252F%252Fwww.google.com%252F. (Accessed on 25/05/2021).

[40] Gartner Website. Definition of american standard code for information interchange (ascii) - gartner information technology glossary. https://www.gartner.com/en/information-technology/glossary/ascii-american-standard-code-for-information-interchange#:~:text=The%20American%20Standard%20Code%20for,computer%20and%20word%20processor%20systems. (Accessed on 13/06/2021).

[41] Modbus.org. Modbus over serial line - specification and implementation guid v1.02. https://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf, December 2006. (Accessed on 15/02/2021).

[42] MODBUS IDA website. Modbus application protocol specification v1.1b. https://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf, December 2006. (Accessed on 11/02/2021).

[43] Hamilton Bonaduz AG. Visiferm programmers manual. https://craft-sensors.s3.amazonaws.com/File-Uploads/Visiferm_ProgrammersManual_ODOUM043_62417905.pdf?mtime=20180725234949&amp;focal=none. (Accessed on 17/02/2021).

[44] Tyco Electronics UK Limited Crompton Instruments. Rs485 & modbus protocol guide. http://www.cromptonusa.com/RS485_GUIDE.pdf. (Accessed on 15/02/2021).

[45] B+B SmartWorx. Modbus tcp ip at a glance. https://www.bb-elec.com/Learning-Center/All-White-Papers/Modbus/Modbus-TCP-IP-at-a-Glance.aspx. (Accessed on 31/03/2021).

[46] ABB SACE S.p.A. Communication protocol modbus-tcp quick overview. https://library.e.abb.com/public/b7cfeaae6eb0c03cc1257a4f002a5118/2CSG445026D0201%20-%20ANR-LAN%20Modbus%20TCP%20Protocol.pdf. (Accessed on 13/06/2021).

[47] Modbus Organization. Modbus messaging on tcp/ip implementation guide v1.0b. https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf. (Accessed on 20/04/2021).

[48] EXSYS Vertriebs GmbH. Ex-1309-t manual. https://www.exsys-shop.de/shopware/media/pdf/a5/bf/cc/handbuch_manual_ex-1309-t.pdf. (Accessed on 22/03/2021).

[49] TRENDnet product page. Usb to serial converter – usb to serial adapter | trendnet - trendnet tu-s9. https://www.trendnet.com/products/usb-to-serial-adapter/usb-to-serial-converter-TU-S9. (Accessed on 14/06/2021).

[50] TRENDnet product page. Usb 2.0 to fast ethernet adapter - trendnet tu2-et100. https://www.trendnet.com/support/support-detail.asp?prod=290_TU2-ET100.

[51] Ltd. Advantech Co. Adam-6000 series user manual. http://www.kgs.no/Produkter/Manual/ADAM-6000/ADAM-6000.pdf, December 2019.

Appendix

A. Specification Sheets

The specification sheets are listed as hyperlinks below.

A.1 Hamilton sensors

EasyFerm Bio PHI Arc 120 Specification Sheet
VisiFerm DO Arc 120 H2 Specification Sheet

A.2 Magnetic mixers

IKA RET control-visc Datasheet
IKA lab disc Datasheet

A.3 Peristaltic pumps

Heidolph Hei-FLOW Precision 01
Ismatec Reglo ICC data sheet

A.4 ADAM-6066 & Ethernet switch

ADAM-6066 specification sheet
EKI-2525LI Ethernet switch specification sheet

A.5 Converters & Adapters

EX 1309-T RS-232/422/485 to USB 2.0 Converter
TU-S9 USB to RS232 converter
TU2-ET100 USB to Ethernet adapter

A.6 USB hub

EX-1177HMV specification sheet

A.7 Valves

ASCO pinch solenoid valve S106

A.8 Power supply & circuit breaker

Acti 9 iC60 RCBO
RS PRO DIN Rail Power Supply

A.9 Electrical sockets

ELKO S1091 PT
ELKO S1014 PT

B. Operation Manuals

The operational manuals are listed as hyperlinks below.

B.1 Hamilton sensors

VisiFerm DO Arc programmers manual
EasyFerm pH Arc programmers manual

B.2 Peristaltic Pumps

Heidolph operation manual
Ismatec Reglo ICC operation manual

B.3 IKA RET control-visc

IKA RET control-visc manual

B.4 ADAM-6066

ADAM-6000 Series User Manual

B.5 USB to RS-485 converter

USB to RS-485 converter Manual

C. Equipment list

Equipment list of equipment used for the control cabinet

Item	Description		Quantity	Price per item NOK	Subtotal NOK	MVA (25%) NOK	Sum NOK
EX-1177HMV	Industrial USB hub, 7 USB-A		2	713	1426	356.5	1782.5
EX-1309-T	USB to RS-485 converter		1	522	522	130.5	652.5
TU2-ET100	Ethernet adapter		1	182	182	45.5	227.5
TU-S9	USB to RS-232 converter		2	129	258	64.5	322.5
RS PRO 136-8319	24V DC power supply		1	385.57	385.57	96.3925	481.9625
Schneider iC60 RCBO	Circuit breaker B16A		1	0	0	0	0
Elko S1014	Single electrical sockets		4	0	0	0	0
ELKO S1091	Double electrical sockets		2	0	0	0	0
Elko RS 2-H	Frame for Elko S1014		2	0	0	0	0
ELKO RS16 Frame	Frame for Elko S1091		2	0	0	0	0
Wago 2000-1401	Feed Through Terminal Block Grey 4-conductor		50	8.13	406.5	101.625	508.125
Wago 2000-1407	Feed Through Terminal Block Green/Yellow 4-conductor		10	29.04	290.4	72.6	363
Wago 2002-1201	Feed Through Terminal Block Grey 2-conductor		20	6.44	128.8	32.2	161
Wago 2002-1301	Feed Through Terminal Block Grey 3-conductor		30	7.81	234.3	58.575	292.875
Wago 2002-1207	Feed Through Terminal Block Green/Yellow 2-conductor		10	24.08	240.8	60.2	301
Wago 249-116	End Stop for Terminal Block		15	4.86	72.9	18.225	91.125
Wago 2000-1291	End Plate for Terminal Block		5	3.38	16.9	4.225	21.125
Wago 2000-402	Jumper Bar for Terminal Block		30	2.32	69.6	17.4	87
Wago 249-119	Adjustable Height Group Marker Carrier		10	3.28	32.8	8.2	41
RS PRO 879-3725	Grey Slotted Panel Trunking 40x40mm		4	65.528	262.112	65.528	327.64
RS PRO 467-406	Slotted DIN Rail 500x35x7.5mm		2	49.62	99.24	24.81	124.05
RS PRO 822-9748	M16 Cable Gand With Locknut		10	9.186	91.86	22.965	114.825
RS PRO 563-453	1.5A Glass Cartridge Fuse, 5x20mm Speed F		10	1.525	15.25	3.8125	19.0625
RS PRO 513-2613	Insulated Crimp Bootlace Ferrules. Bag of 500		1	123.93	123.93	30.9825	154.9125
LAPP KABEL 53015140	PG16 Cable Gland		2	12.59	25.18	6.295	31.475
LAPP KABEL 53019040	PG16 Locknut		10	4.31	43.1	10.775	53.875
LAPP KABEL 53015160	PG29 Cable Gland		1	34.06	34.06	8.515	42.575
LAPP KABEL 53019060	PG29 Locknut		1	11.56	11.56	2.89	14.45
MOLEX / FLAMAR 155220	Multicore Cable, Flamar, 12 Core, 1.5mm2		4	163.48	653.92	163.48	817.4
LAPP KABEL 0015403	Multicore Cable, Ölfex, 3 core, 2.5mm2		6	43.56	261.36	65.34	326.7
		Total NOK:			**5888.142**	**1472.0355**	**7360.1775**

D. Calculations

Power consumption SBR

Power consumption, Circuit breaker and 24V DC power supply requirement calculations

The calculations consider every item as if they draw their max current

Product	Description	Information	Power consumption per [W]	Amount	Total power consumption [W]	Ampere [A]
230V						
IKA RET control-visc	Magnetic stirrer, heater	Heat output 600W	650	4	2600	11.30
EHEIM 3702010	Air pump		4	4	16	0.07
Ismatec Reglo ICC	Small liquid pump		30	4	120	0.52
Hei-FLOW Precision 01 Mult	Bigger liquid pump		140	2	280	1.22
RS PRO DIN Rail Power Supply 136-8319	24V DC power supply output current: 5A		299	1	299	1.30
		Total:			3315 W	14.41 A

Product	Description	Information	Power consumption per [W]	Amount	Total power consumption [W]	Ampere [A]
24V						
EasyFerm	ph og temp sensor		0.15	4	0.6	0.025
Visiferm	DO sensor		1	4	4	0.167
ADAM-6066	Power relay		2.5	2	5	0.208
EKI-2525LI	Ethernet Switch		3	1	3	0.125
EX-1177HMV	USB-hub (7 socket USB 2.0, 0.5A @ 5V ea)		17.5	2	35	1.458
S10608-Z130A	Pinch solenoid valve	Assuming max two is turned on at the same time	13	2	26	1.083
		Total:			73.6 W	3.07 A

E. Communication appendix

E.1 CRC calculation

The CRC calculation is started by first pre-loading a 16-bit register to all 1's(or FFFF in hexadecimal). Then a process of applying each successive 8-bit byte of the message to the current content of the register. Only the eight bits of data in each character is used for generating the CRC value. The start-, stop- and parity bits are not used for CRC calculations.

During the calculation of the CRC, each 8-bit character is exclusive OR-ed with the register contents. Then the result is shifted in the direction of the least significant bit(LSB), with a zero filled into the most significant bit(MSB). The LSB is then extracted and examined. If the LSB are 1, then the register is exclusive OR-ed with a preset fixed value. If the LSB are 0, the process proceeds and no exclusive OR takes place.

This process is then repeated until 8 shifts have been performed. After the last shift, the next 8-bit character is exclusive OR-ed with the current value of the register. This starts the process just described again, and eight more shifts are done. The final content of the register, after all characters of the message have been applied, is the CRC value.

The MODBUS Organization [41] have a 8 step procedure for generating a CRC value:

1. Load a 16-bit register with FFFF hex (all 1's). Call this the CRC register.
2. Exclusive OR the first 8-bit byte of the message with the low-order byte of the 16-bit CRC register, putting the result in the CRC register.
3. Shift the CRC register one bit to the right (toward the least significant byte), zero-filling the most significant byte. Extract and examine the LSB.
4. If the LSB is zero;repeat step 3. If the LSB is one; Exclusive OR the CRC register with the polynomial value 0xA001.
5. Repeat steps 3 and 4 until eight shifts have been performed. When this is done, a complete 8-bit byte will have been processed.
6. Repeat steps 2 through 5 fot the next 8-bit byte of the message. Continue doing so until all bytes have been processed.
7. The final content of the CRC register is the CRC value.
8. When the CRC is appended into the message, its upper and lower bytes must be swapped so that the lo-order byte will be transmitted first followed by the high-order byte. (For example, if the CRC value is 1241 hex. 0x41 appended first as the low-order byte followed by 0x12 as the high-order byte.)

F. Excel log

A example of a logged process with OUR-mode as control method can be found by clicking the following hyperlink:
Excel log file OUR-mode

The Excel template used can be found by clicking the following hyperlink:
Excel template SBR

G. Libraries

Ismatec's own Reglo Digital LabView driver have been used to communicate with the Ismatec Reglo ICC pumps. The drivers are found by clicking the following hyperlink:
Ismatec LabView Drivers

NI Modbus Library version 1.2.1.42 have also been used to communicate with the connected devices. This is National Instruments own MODBUS library. This library can be found using the following hyperlink:
NI Modbus Library

H. User Manual

The User Manual used can be found by clicking the following hyperlink:
User manual

I. Video of SBR process

SBR Process video