



Technische Universität München

Lehrstuhl für Bioverfahrenstechnik

Semesterarbeit

Development of a Low-Cost Electrical Conductivity Meter for Liquids

Sebastian Plamauer

Matrikelnummer: 3609702

10th August, 2016

Betreuer: M.Sc. Timm Severin

Prof. Dr.-Ing. Dirk Weuster-Botz

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Ort, Datum

Unterschrift

Abstract

In the light of problems surrounding the use of fossil fuels, photobioreactors emerge as a cost effective option to produce climate-neutral and renewable bio-fuels. To study and improve those reactors simulations and experiments are necessary. To enable the measurement and analysis of the flow in the reactor this thesis develops a low-cost electrical conductivity meter to measure changes of salinity in a flowing liquid. This allows for tracking saltwater pulses in a freshwater flow, indicating the movement of the volumes. The resulting system was successfully tested and generated usable data.

Contents

1. Introduction	7
2. Objectives	9
2.1. Spacial and Time Resolution	9
2.2. Electrical Conductivity Resolution	12
2.3. Cost	12
2.4. Usability	12
2.5. Requirements	13
3. Background	14
3.1. Theoretical Background	14
3.2. Market Research	17
4. Design	19
4.1. System Design	19
4.2. Electrodes	21
4.3. Matrix Switches	23
4.4. MinieC Interface	26
4.5. Microcontroller	29
4.6. Carrier Board	30
4.7. Embedded Software	31
4.8. OpenSalinity GUI	33
4.9. Data Conditioning	34
4.10. Bill of Materials	35
4.11. List of Software	36

5. Results	38
5.1. Verification	38
5.1.1. Spacial Resolution	38
5.1.2. Electrical Conductivity Resolution and Range	41
5.1.3. Cost	43
5.1.4. Deployability in the Bioreactor	43
5.1.5. Usability	45
5.2. Validation	45
6. Conclusion	56
7. Outlook	57
A. Appendix	58
A.1. Instruction Manual	58
A.2. Algorithms	61
B. Abbreviations	63
C. List of Figures	64
D. List of Tables	65
E. Bibliography	66

1. Introduction

Fossil fuels play a large role in the world's energy production and as energy sources for transportation vehicles. Especially for vehicles, their high energy density and liquid form at room temperature has enabled the construction and operation of cars and airplanes otherwise not possible. At the same time, there are fundamental problems in using fossil fuels as energy sources. Be it oil, coal or gas, the formation of these resources is a natural process spanning millions of years. This means that once the current reserves are depleted, they will not be refilled. Even without depleting all the reserves, the depletion of the easily reachable reserves means that ever greater effort has to be taken to find and use less accessible fields, using methods with severe impact on the environment. In addition to that, burning fossil fuels for energy also releases a slew of gases into the atmosphere, of which CO₂ is the most prominent. CO₂ is a greenhouse gas and as such influences the world's climate tremendously.

In light of those problems, renewable energy sources are needed. For electricity production there are some attractive technologies already in use that directly or indirectly generate power using the sun. For vehicles, using electrical power can be problematic. The electricity has to be stored in batteries, and even with the higher efficiency of electrical motors, the lowered energy density reduces range drastically. Batteries also have to be charged, as opposed to be refilled like tanks, which takes substantially more time. While for some vehicles, like cars, this technology still can be made feasible, it is much less suitable for airplanes.

For these applications a more direct replacement for fossil fuels has to be found. A possible solution are bio-fuels. Bio-fuels are generated from biomass, for example plants. An important requirement for the growth of biomass in order to produce bio-fuel is to avoid competition over land and resources with food production. A low price is also important to make the product economically viable.

One technology which potentially fits these needs are open photobioreactors (PBR)

growing microalgae. These algae produce lipids, which can be processed to fuel. To improve and optimize these reactors experiments and models are needed. A computational fluid dynamics (CFD) simulation models the flow in the reactor. In order to validate this simulation, experiments measuring the flow conditions in the reactor have to be conducted. To do so, a sensor system needs to be developed. The option explored in this thesis is to measure the stream's conductivity. The next chapter will describe the objectives and criteria of this system in detail.

2. Objectives

The goal of this project is the development and test of a low-cost electrical conductivity meter for liquids to be used as an aid to measure and analyze the flow in a photobioreactor.

The method chosen beforehand was to measure the fluids' electrical conductivity, which can be changed easily by adding water with differing salt concentrations. Commercially available conductivity meters are built to measure with high accuracy in order to obtain information about a liquid's absolute salinity and are relatively expensive. In our case, however, the meter does not need to create high-accuracy absolute measurements, but measure a relative change allowing to distinguish different liquids by their salinity. However, this needs to happen very fast and at a lot of different points in the stream. The more positions measured, the more complete the picture of the flow becomes. Therefore, the cost per sensor has to be low, to not put a restraint on the total number of points that can be measured.

The actual flow analysis is not part of this work, but rather the creation of a tool to make it possible. As such, the system needs to be designed to be easily usable by people without deep knowledge of the underlying technology.

The following sections describe and detail the requirements the sensor system has to fulfill in order to meet the objectives. These requirements translate the objectives into discrete and verifiable units, serving as the base for development and benchmark for the later performance analysis.

2.1. Spacial and Time Resolution

The spacial resolution ds and the time resolution dt decide the granularity of the flow image. Both resolutions are connected by the velocity v of the stream flowing over the

sensor as shown in equation (2.1).

$$v = \frac{ds}{dt} \quad (2.1)$$

With both resolutions connected, only one of them has to be defined.

The information gathered by the sensor has to be granular enough to enable the verification of the simulation. As it is not possible to derive a hard number from that, the following list shows different deliberations to establish a first estimate.

- The granularity of the simulation is determined by the mesh size and is in the order of about 1 mm to 5 mm. A resolution better than that would be unnecessary, making this the lower limit.
- The data collected by the sensor system shows salinity of the liquid passing a certain position over time. By extracting the same data from the simulation, a graph can be compiled comparing the measured to the simulated values. The flow becomes visible when comparing data from different positions. The first sensor to show a spike in salinity is further up the stream than sensors spiking later. The distance between two sensors has to be big enough for a given velocity of the stream so that the time delay can be clearly seen. At a flow speed of v in the order of 1 m/s, a resolution in the order of centimeters would yield time differences of centiseconds. Whether or not this is sufficient depends on noise and delay in the sensor system, however centiseconds is a conservative estimate likely to be achievable.
- The geometry of the inlet as shown in figure 2.1 also influences the needed resolution. It has to be an order of magnitude smaller than the basins characteristic length, in this case the width of 850 mm, to enable gathering data about the flow conditions within.

Derived from all that, a spacial resolution in the order of centimeters is chosen as requirement.

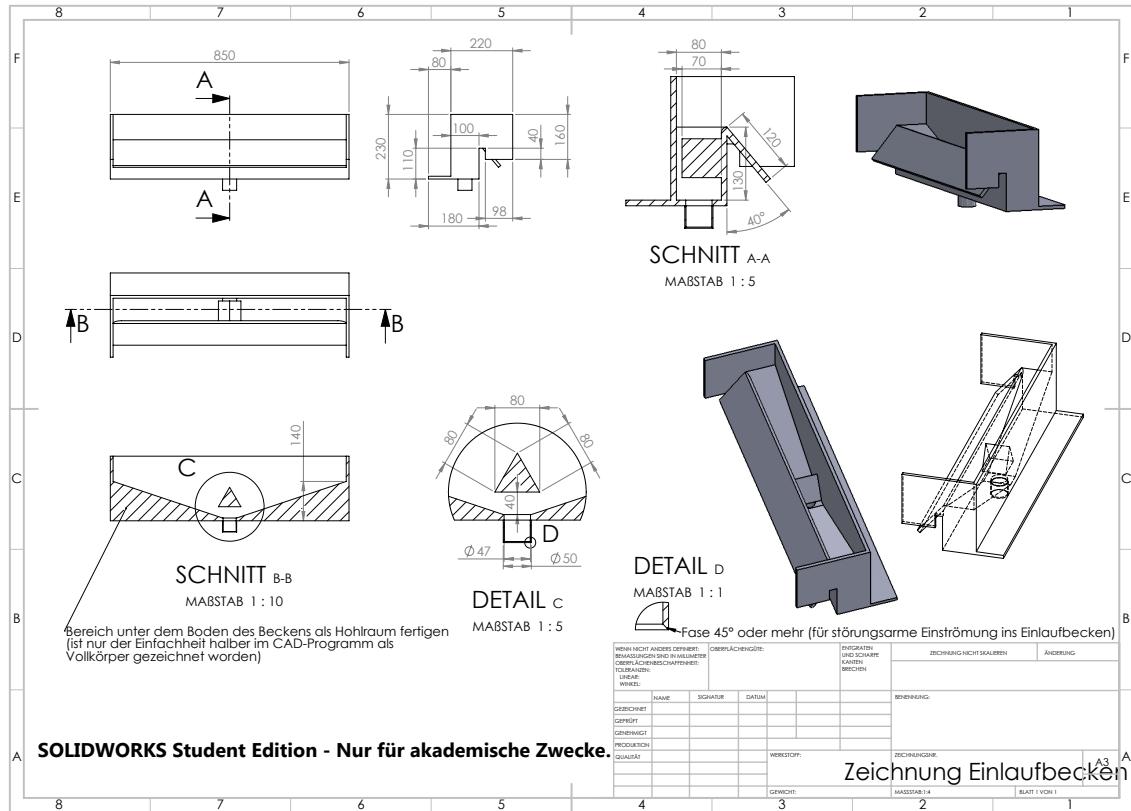


Figure 2.1.: The inlet basin through which the water is fed back from the bottom of the reactor to the top.

2.2. Electrical Conductivity Resolution

To measure the arrival of the new water stream after switching the water feed, the system has to be able to distinguish between water with different salinity. The water used normally in the reactor is tap water with a salinity of about 0.2 %. The salinity of the added saltwater can be chosen freely. Water with a salinity of about 5 % is easily available in the facilities and offers a sensible choice. Assuming a reactor with 65 l in circulation and an added saltwater impulse of 5 l, the resulting salinity after perfect homogenization would be approximately 0.5 %. The system has to be able to clearly distinguish between all those salinities. The sensors sensitivity therefore shall be better than a change of 0.1 % salinity with a range from 0 to 5 %.

2.3. Cost

The more sensors used, the more points in the stream can be measured and the better the resulting image of the stream. Therefore, the cost per sensor has to be low enough to not be prohibitive to adding more sensors. The cost of a high quality lab conductivity meter is in the range of €1000 and was set as the goal of maximum cost for the sensor system. The number of sensors needed to cover all interesting regions of the bioreactors is about 40. A maximum cost per sensor of €25 results from those figures.

2.4. Usability

The sensor system is meant to be used in the algae reactors of the algae cultivation center at the Ludwig Bölkow Campus. It has to be possible to easily mount and remove the system to and from the reactor without having to dismantle it.

The system also has to be easy to use, so it can be helpful to the researchers working on the reactor. It has to work reliably and act according to expectations of the users. The chance of handling errors that lead to loss of data has to be minimized. All operations have to be documented in a minimal set of written instructions, so the system can still be used even if the designer is not available.

2.5. Requirements

Table 2.1 concludes and displays all requirements alongside the methods of verification to be used.

Table 2.1.: Requirements

Nr.	Requirement	Verification
1	The system shall have a spacial resolution in the order of 10 mm.	Test, Inspection
2	The system shall have a sensitivity of 0.1 % salinity.	Test
3	The system shall have a range from 0 to 5 % salinity.	Test
4	The cost per sensor shall be less than €25.	Analysis
5	The system shall be deployable in the algae reactor.	Demonstr.
6	The system shall be usable with a minimal set of written instructions.	Test, Review

3. Background

In this chapter a theoretical background of electrical conductivity measurement will be established. After that, a market research takes a look at the commercially available solutions of sensor systems and parts that could be used to develop one.

3.1. Theoretical Background

The following section is translated and summarized from Tränkler et al. [2015] and Gevatter [2000].

The conductivity κ is the ability of a certain volume of a substance to conduct electricity. It is measured in the unit S/m and is a specific parameter normalized to the length and cross section of the volume. Conductivity in a liquid depends on ions as charge carrier and can therefore be used to measure its concentration.

To better understand conductivity, an equivalent circuit diagram, shown in figure 3.1, of an electrode system shown in figure 3.2, can be used. This method is called conductometry and in this form a two-electrode-cell is used. In this measurement configuration, two electrodes are submerged in the liquid to be measured and a voltage is applied. The resistance R of the substance is determined by measuring the potential drop U_R given a constant voltage U , current I and internal resistance R_i .

The resistance R is

$$R = \frac{U_R}{I} \quad (3.1)$$

The inverse of the resistance R is the conductance G

$$G = \frac{I}{U_R} \quad (3.2)$$

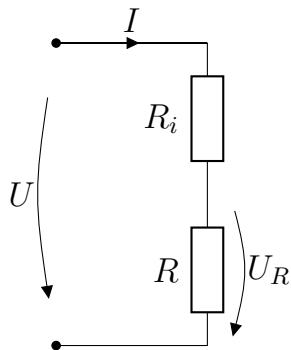


Figure 3.1.: The equivalent circuit diagram shows the voltage drop U_R over the resistance of the fluid relative to the internal resistance R_i , driven by the applied voltage U .

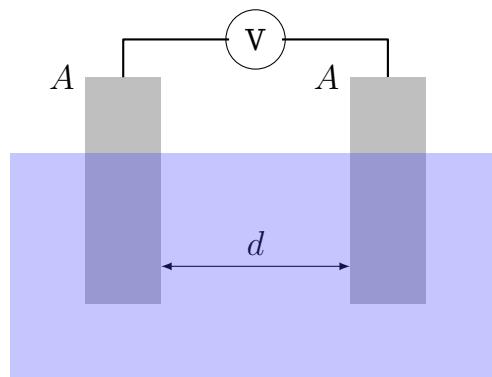


Figure 3.2.: Two electrodes with the area A are submerged in the solution with the distance d between them. A voltmeter is used to measure the voltage drop.

The cell constant C describes the geometry of the sensor

$$C = \frac{d}{A} \quad (3.3)$$

where d is the distance between and A the surface area of the electrodes.

Considering the cell constant C finally yields the conductivity

$$\kappa = G \cdot C \quad (3.4)$$

In electrolytes, the electrical conduction is a result of mass transfer, where ions are carrying the charges. If the measurement is conducted with direct current, this mass transfer leads to changes in the measured solution and the electrode surface, negatively impacting the measurement. Furthermore, polarization effects create additional resistance, leading to lower than actual results. To avoid this, alternating current is used. The fast, periodical swap of polarity eliminates the net mass flow and its effects. Polarization is a result of the current flowing through the electrode, thereby its effects can be minimized by minimizing this current. One method to do this is replacing the two-electrode-cell with a four-electrode-cell. This separates the current flow from the potential measurement by using one electrode pair to apply the current, and a separate pair to measure the potential drop.

The electrolyte's temperature affects the mobility of the ions and thereby also has a big influence on the conductivity. It has to be taken into account when comparing two measurements. If the temperature T is known, the conductivity at that temperature κ_T can be normalized to a reference temperature T_{ref} using equation (3.5), resulting in the reference conductivity κ_{ref} .

$$\kappa_{ref} = \kappa_T \frac{1}{1 + C_T \cdot (T - T_{ref})} \quad (3.5)$$

The temperature coefficient C_T assumes a linear correlation and is only valid in a narrow temperature range. For bigger ranges the denominator can be replaced with

a polynomial using higher order coefficients, resulting in equation (3.6).

$$\kappa_{ref} = \kappa_T \frac{1}{\sum_{i=0}^n C_{Ti} \cdot (T - T_{ref})^i} \quad (3.6)$$

3.2. Market Research

Conductivity meters can be readily bought and range from prices of over €1000 for lab equipment [Fisher Scientific 2016a] to €100 for simple field water quality monitors [Fisher Scientific 2016b]. Even cheaper water quality testers from no-name manufacturers can be found for as little as €10 from online vendors.



Figure 3.3.: Fisher Scientific™ Traceable™ Salinity Meter Pen [Fisher Scientific 2016b]

Figure 3.3 shows a field conductivity meter from a quality manufacturer as example for the typical traits most available solutions share:

- a single electrode pair
- a display to report measurements
- designed to perform singular reads with high quality

While field conductivity meters have the electrodes integrated to form one compact device, the more expensive lab equipment often has the electrodes attached to the device with a cable, allowing for a more flexible use.

A different option was found in the MinieC I2C eC interface from Sparky's Widgets [Edwards 2016]. The MinieC interface is an Open Hardware project providing a conductivity sensor that easily interfaces with a microcontroller, licensed under Creative Commons [Creative Commons 2016]. This license allows to study the original design files and build modified versions. Ready-made boards can be purchased for about €25. The design approach of this solution is also more modular than that of the more professional equipment. The MinieC offers an interface to connect external electrodes and can be controlled and read by a microcontroller.

Table 3.1 shows a comparison of the strengths and weaknesses of the different systems with respect to the use case in this project. Both the lab and field conductivity meters offer complete solutions with high accuracy. They do however lack the modularity of the MinieC as well as the machine interface, both of which are more important for the intended purpose than the accuracy. The complete solutions they offer simply are solutions for a different case than ours, while the MinieC is positioned as a flexible building block to be used in whatever way needed. While it means that more work has to be done in designing a system around it, this is exactly the trait that allows us to build a device fit for our tasks.

Table 3.1.: Feature comparison table of different conductivity sensor solutions. Strengths and weaknesses are indicated by + and - signs.

Feature	Lab Conductivity Meter	Field Conductivity Meter	MinieC
Price	--	-	+
Modularity	-	--	++
Completeness	++	++	-
Interface	-	-	++
Accuracy	++	+	-

4. Design

Figure 4.1 shows the device developed in this thesis. The following section describes the design of the sensor system and all components involved.

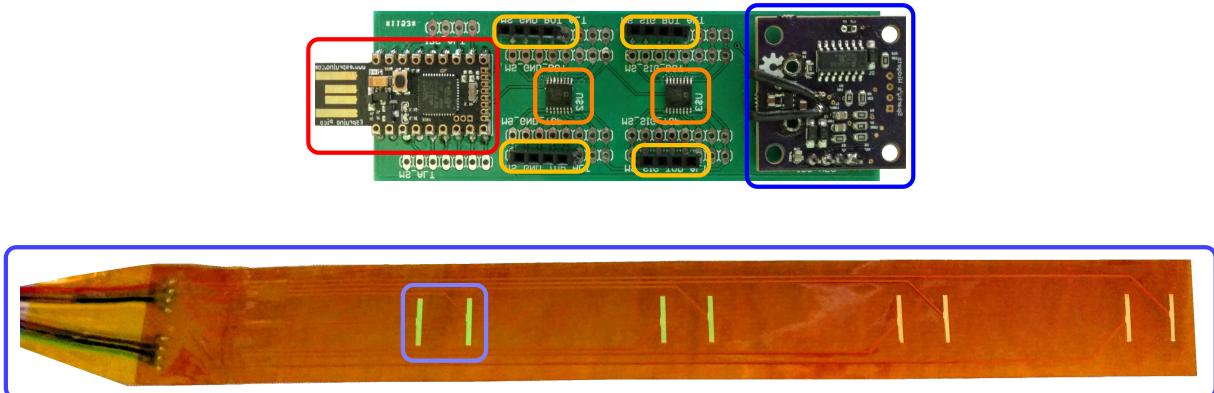


Figure 4.1.: The developed sensor system. The microcontroller and USB connection (—) control all parts and connect to the host PC. The Matrix Switches (—) connect the MinieC (—) to the sensor strip (—) containing the sensors (—) via the connectors (—).

4.1. System Design

The system as shown in Figure 4.2 is designed as follows:

On the user-facing side there is a personal computer (PC). This PC runs software that visualizes a live data stream and provides a control interface for the sensor system. It is connected via USB to a microcontroller. This microcontroller controls the MinieC interface via the Inter-Integrated Circuit (I2C) communication protocol, reads the measurement data from it and sends it to the PC. It also controls the matrix switches via the Serial Peripheral Interface Bus (SPI). The MinieC Interface provides the signal (SIG) and ground (GND) between which the resistance is measured. Both lines are connected to one matrix switch each. These matrix switches are able to connect one

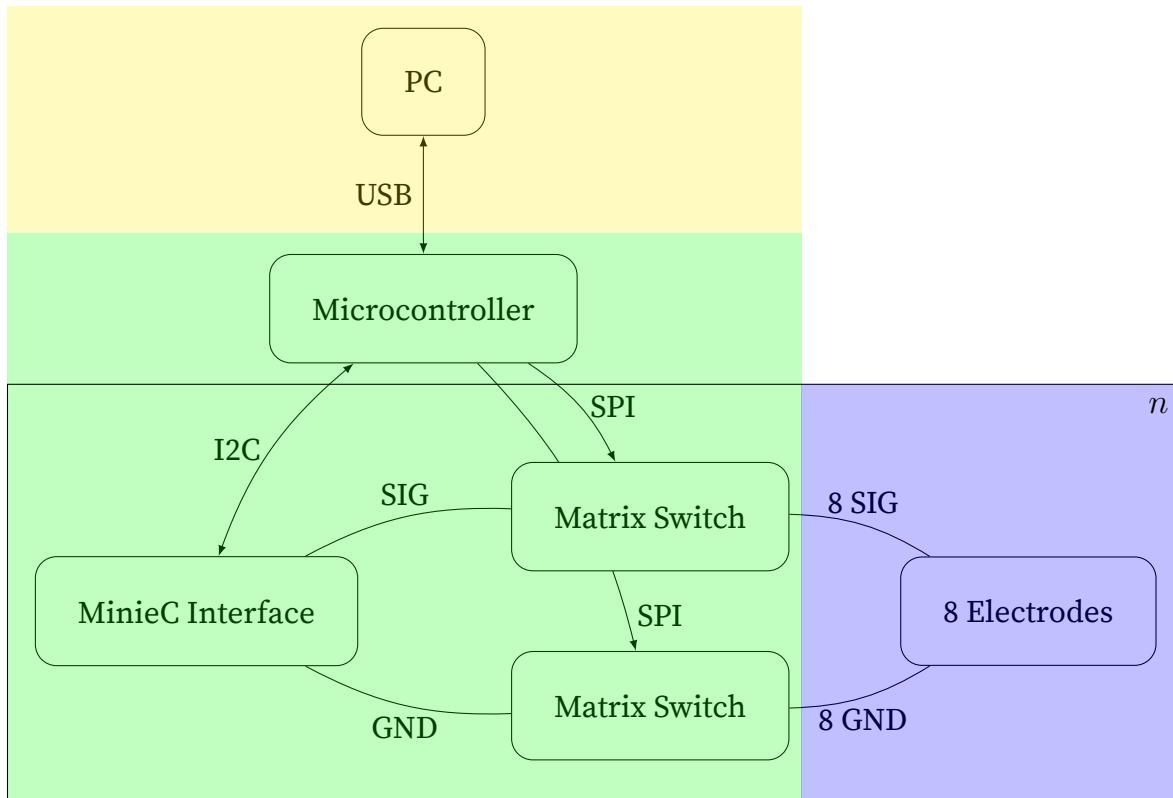


Figure 4.2.: System Design - The yellow area marks the user facing side of the system, the green area is the hardware that is mounted on the carrier board and the blue area marks the sensors deployed in the water stream. The rectangle marks the part of the system that is repeated n times to get to the needed number of sensors.

input to 8 different outputs. On each of those outputs, one electrode is connected. The matrix switches can thereby connect the MinieC interface to one of 8 electrode pairs at a time.

To add further sensors to the system, two strategies are possible:

- by chaining up multiple stages of matrix switches, the number of electrodes can be increased eightfold with each stage
- by connecting another MinieC Interface with its own set of matrix switches to the microcontroller, 8 more electrodes can be added with each of these subsystems

The first option results in lower cost per added electrode, as the MinieC interface is reused. However, with a system like that, all electrodes have to be read in serial,

while with the second option each MinieC interface can be read in parallel, resulting in a higher sample rate. Practically the second option is also easier to achieve. For the first option, a board with 18 matrix switches and 80 connections is needed, while the second option only uses 2 switches per board resulting in 16 connections. The simpler board greatly reduces complexity and can also be made smaller.

4.2. Electrodes

The electrode pairs are the actual sensors in contact with the fluid to be measured. Their material and geometry influence the measuring range of the system. According to Tränkler et al. [2015] a cell constant C of 1 enables a range from $2 \cdot 10^3 \mu\text{S}/\text{cm}$ to $10 \cdot 10^3 \mu\text{S}/\text{cm}$. For a water temperature of 18°C this corresponds to salinities of 0.12 % and 0.66 %, which would be lower than those stated in the requirements. Still, a cell constant C of 1 is the best compromise for our purpose. Tränkler et al. [2015] also do not specify the material's influence on the range and also do not specify how the usable range is defined, leaving room for a possibly higher than anticipated range. First bench tests confirmed this assumption.

Figure 4.3 shows the geometry of the sensor. To achieve a cell constant C of 1, a width w of 1 mm, height h of 10 mm and distance d of 10 mm were chosen.

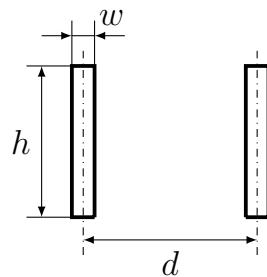


Figure 4.3.: Two rectangular electrodes form an electrode pair.

As a first proof-of-concept a sensor array was built, containing multiple electrode pairs on a strip, pictured in Figure 4.4. A 50 mm wide and 250 mm long band of Kapton adhesive tape served as the base. Four electrode pairs made from 0.2 mm platinum

wire were arranged equidistant on the strip. Eight 0.4 mm enameled copper wires were run along the tape to connect each electrode to the left end of the strip, from which insulated cables run to the carrier board. After soldering the joints, two smaller strips of tape were used to cover the wiring, exposing only the electrodes to fluid.

First tests with this sensor array showed the viability of the concept, however a simple look at it shows the inherit problems: instead of a uniformly flat strip with minimal influence on the flow, the assembly forms several irregularities. Soldering 0.2 mm platinum wire to 0.4 mm enameled copper wire on a piece of adhesive tape per hand also did not result in clean solder joints. While, after the experience of the first array, the second array turned out a bit cleaner, the fundamental problem remains: it is a tedious manufacturing process resulting in a low quality product.



Figure 4.4.: A handmade sensor strip. The short, vertical silver lines are the platinum electrodes, the darker horizontal lines are the copper wires, running to the cables attached on the left.

As an alternative to these handmade strips, industrially produced flexible printed circuit boards (flex-PCB) were identified. Flex-PCBs are constructed similar to the handmade arrays described above. They also use Kapton as base, on which a copper coating is applied and partially removed by etching to form the conducting paths. On top, another layer of Kapton is applied, with cutouts in the places where the copper is supposed to be exposed. The exposed copper is then plated with ENIG (Electroless nickel immersion gold) to protect the copper from oxidation and provide the landing pads for electrical components to be soldered on.

For our purpose those exposed and plated landing pads can be used as electrodes, being nicely embedded in a FlexPCB that also runs the wiring up to an interface from where cables can be run. Using the FlexPCB itself as cable is not viable due to the high cost per area. Instead, cables are soldered directly to the PCB and silicone is used to



Figure 4.5.: The sensor strip design to be implemented as flexible PCB. The red lines mark the copper layer, the green line is the outline of the board.



Figure 4.6.: A finished sensor strip assembled with cables and waterproofing on the left.

create a waterproof seal around the connection.

The final design for the sensor strip, as shown in Figure 4.5, implemented as a flexible PCB consisted of four electrode pairs spaced 50 mm apart. The strip is 25 mm wide and 220 mm long. Figure 4.6 shows one of the 16 pieces manufactured by LEITON.

4.3. Matrix Switches

The matrix switches are an essential part of the system, enabling the use of multiple sensors with a single MinieC Interface, thus lowering the cost per sensor. The part used is an Analog Devices ADG738 . It is an 8-channel CMOS analog matrix switch controlled via a 3-wire serial interface. The following information is taken from the data sheet provided by the manufacturer Analog Devices [2016].

Figure 4.7 shows the functional block diagram. The switch has one drain pin (D) and 8 source pins (S1..S8). Despite the naming of drain and source, the internals provide simple switches between the drain and each source pin. The switches work in both directions without any restriction on the signal other than a maximum current of 120 mA, which far exceeds our needs. By sending control commands via the SPI interface, each of the 8 internal switches can be turned on and off individually.

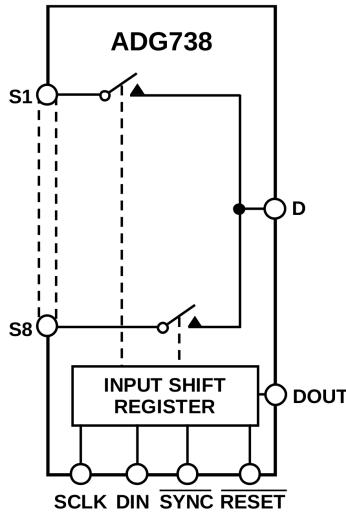


Figure 4.7.: The functional block diagram of the ADG738 matrix switch. Pin (D) is connected to the outputs (S1 .. S2) via switches, that are controlled by SPI-input on (DIN). (SCLK) is the driven by the clock signal, (SYNC) is the chip select pin and (RESET) breaks the connection of all switches. (DOUT) provides the signal of (DIN) to enable daisy-chaining.

The example timing diagram in Figure 4.8 describes the data transmission process. The microcontroller sends one byte of data to the matrix switch. Each of the 8 bits of this byte controls one switch. The first bit controls the first switch and so on. If the bit is 1, the switch is closed, if 0, the switch is open. To send the byte, first the synchronization pin (SYNC) has to be pulled low from its usual high level. A clock signal is provided to the clock pin (SCLK). At each falling edge of the clock signal, the data input (DIN) is read - where high leads to a 1-bit and low to a 0-bit. After 8 cycles, SYNC is pulled high again marking the end of data transmission with a full byte transferred. After that, the switches immediately take their instructed states with switching times in the order of 100 ns. The complete process of transferring one byte and switching to the new state takes about 520 μ s.

Multiple matrix switches can be controlled at once by daisy-chaining the data output pin (DOUT) of the first device to DIN of the second one, and so forth. Both SYNC and SCLK are connected to the same bus. This assures that all matrix switches are set in the same state and at the same time which is important for our case to synchronize

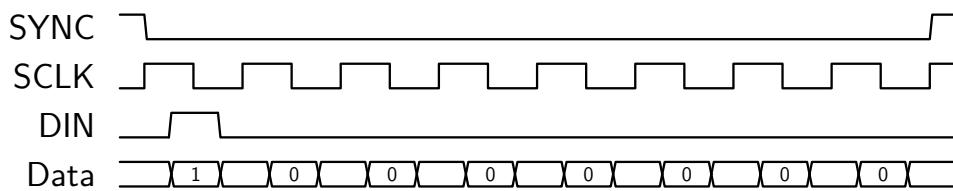


Figure 4.8.: Example Timing Diagram - Data input starts when (SYNC) is pulled low. On falling edges of (SCLK), (DIN) is read. If (DIN) is high at that moment, the bit is 1, if low it is 0. After eight bits are transferred, (SYNC) is pulled high again and the data transfer is finished. The resulting state is a closed first switch while all other switches are open.

the two used ADG378 parts.

4.4. MinieC Interface

The MinieC Interface contains the electronics to perform the resistance measurement. It contains several parts used to generate an electrical signal, run it through a circuit in which the liquid to be measured serves as a resistor and measure the voltage drop over it.

The first part is a charge pump voltage inverter. Its purpose is to generate a negative output voltage from a positive input. This negative voltage and the positive voltage are needed to drive a Wien bridge oscillator. This oscillator outputs a sine wave voltage oscillating between the positive and negative input voltage. This whole first stage's purpose is to generate an alternating current to be used in the measurement, avoiding the polarization effects described before.

The AC signal provided by the first stage is then fed into an operational amplifier (opamp). An opamp is a part that has two input signals and one output, where the output is proportional to the difference of the input signals. In our case, the opamp's output is pulled to ground via a voltage divider as shown in Figure 4.14. The first resistor R_i in the divider is fixed, while the second one R is the liquid to be measured. The output voltage of this divider is depended on the liquid's resistance. This voltage is then fed back into the second input of the OpAmp. Via this feedback, the output (OUT) of the opamp is now the input signal (SIG) modulated in amplitude by the resistance of the liquid.

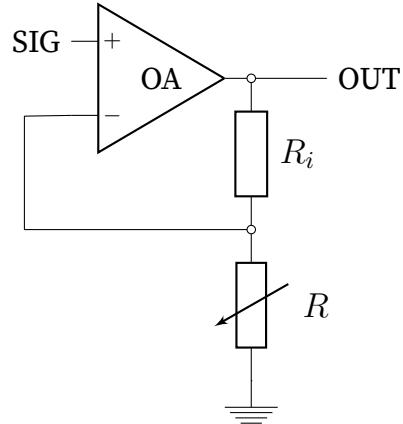


Figure 4.9.: The opamp (OA) modulates the input signals (SIG) amplitude proportional to the variable resistance R in the voltage divider to generate the output (OUT).

In the last step, the modulated output is amplified in two stages, filtered, and then measured by an analog-digital-converter (ADC). The ADC measures the voltage and provides the measurement to the microcontroller via I2C.

First tests of the interface surfaced an interesting problem. The graph displayed in figure 4.10 shows the response to an immediate switch from the AC input signal to zero. The first red line marks the moment of the switch, the second line marks the moment when the response reached the voltage zero. The red dot marks the moment when the response reached the midpoint between the old and new input. It took 40 ms for the response to follow the input. This slow response time is not acceptable in our system because it prohibits the rapid switching between sensors needed for the fast sampling of data. The slow response would smear the measurement over all sensors and it would not be possible to measure a difference between them. Waiting for that amount of time between each read of a sensor would reduce the time resolution to a level where the desired information can no longer be extracted from the data. The cause of this behavior is the filter placed before the ADC. As the output of the opamp is an alternating current, a diode and a filter capacitor are used to generate a direct current signal proportional to the amplitude of the AC signal. This is done so the sample rate of the ADC does not have to match the frequency of the signal, which makes it easier to use when fast sampling rates are not necessary.

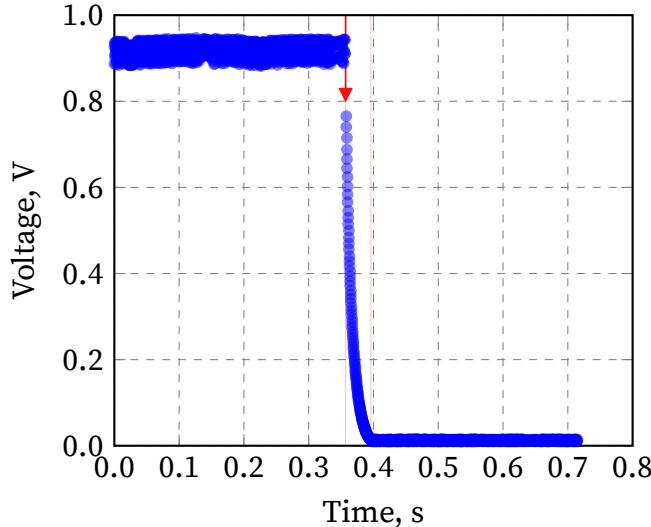


Figure 4.10.: The blue graph shows the measured voltage with a resistance of 0. At the first red line, a switch breaks the connection, making the resistance infinite. The voltage drops slowly towards zero and reaches the new value after 0.04 s, marked by the second red line.

Figure 4.11 shows the response to the same switching as before, but with the filter capacitor removed. The diode is still in place and removes the negative voltages from the output. Without the capacitor however, the signal oscillates with the same 1666 Hz as the Wien bridge oscillator provides. Because the sample rate of the ADC is not as fast as the oscillation, it measures at random moments on the sine wave, resulting in an output that moves between zero and the maximum amplitude. The red line again marks the moment of the switch and it can be clearly seen that the response delay is now gone. The oscillating behavior that was electrically filtered before is now visible and has to be addressed in the data analysis which will be described later in the Section 5.2 Validation.

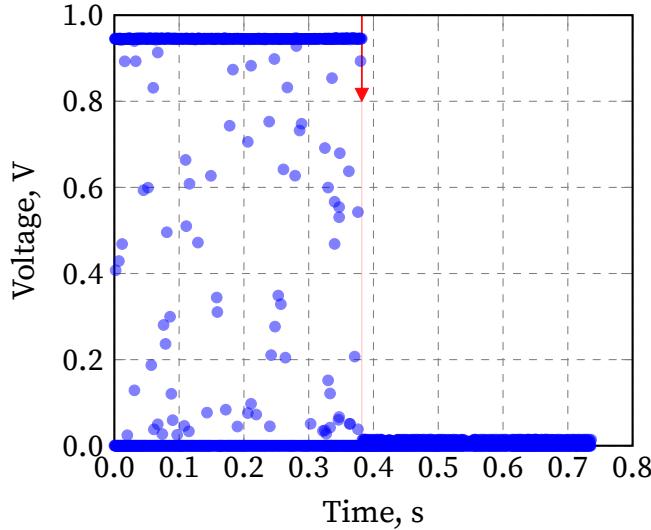


Figure 4.11.: The scatter plot shows the signal oscillating between 0 and the voltage at a resistance of 0. The red line marks the breaking of the connection. The signal follows without delay.

4.5. Microcontroller

The microcontroller has to be able to control the functions of the sensor nodes attached to it, read the data from them, log it and serve it to the user-interface. They usually are programmed in C or C++, however in recent years other options emerged. One of those is MicroPython [2016], which is an implementation of the Python 3 programming language designed to run on microcontrollers. As a scripting language Python is a vastly easier language to work with than C/C++, and helps to speed up the development process. Scripting languages are also often familiar to the targeted users of this device from usage for data processing and visualization. Using MicroPython allows us to design a system where it is more likely that the people using it are able to quickly understand the code, enabling them to improve or adapt it for alternative purposes. It does however limit our choice of hardware to supported platforms and it requires more powerful and thus more expensive micro-controllers. But as the system only requires one microcontroller to drive a very large amount of sensors, the added cost is relatively small and outweighed by the benefits of the better usability.

For prototyping, a development board named "Espruino Pico" was chosen. It is a

very small and simple board that provides the electrical boilerplate to use a microcontroller without needing to deal with the lowest level of electronics. It provides a stable power supply and a USB connection to a host PC for programming and to export the collected data.

4.6. Carrier Board

The carrier board is a simple printed circuit board (PCB) implementing all parts described above. Figure 4.12 shows the assembled board. On the left, the Espruino Pico board can either be soldered directly to the PCB or plugged in using pin header connectors. Above and beyond used pins are replicated on through-holes. This allows for easy connection of measurement equipment to debug the system during development, but can also be used later to connect carrier boards together. Only one would carry a microcontroller and control the other connected boards.

In the middle there are two matrix switches. They can either be directly soldered on as in this image or they can be soldered to an adapter board that is again mounted to the carrier with pin header connectors in the inner rows of through-holes. The outer through-holes are where the connectors for the cables to the sensors are mounted.

The MineeC is located on the right-hand side. It can also be either directly soldered or used with pin header connectors. Additionally, two wires have to be run to the carrier board because of a misalignment of ports due to a design flaw.

The design is tailored for use as a prototype. That means that everything is made bigger than necessary, which allows for easier modifications. All used pins are replicated on additional through-holes, making them accessible for measuring equipment like an oscilloscope or multimeter. It is also kept modular, so that each part can be swapped out separately. A later redesign would integrate all boards into one and try to reduce size, which in turn saves money on PCB manufacturing. However, the prototype showed no critical design errors and is fully functional, so the next design step is only necessary when more boards are needed.

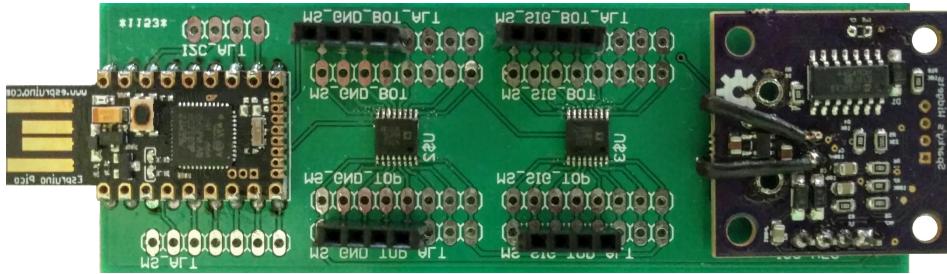


Figure 4.12.: The assembled carrier board with all parts directly soldered on.

4.7. Embedded Software

The embedded software is the program running on the microcontroller. As already described in Section 4.5 Microcontroller, MicroPython is used to implement this program.

As soon as the system is powered up, it starts listening for the Start and Stop commands from the OpenSalinity GUI running on the host PC. Communication happens via a serial connection on the USB port. Once a Start command is received, polling of the sensors is started and the data is delivered to the PC, where it is captured and stored. In order to poll the sensors, the program has to control the switches and the ADC. It first switches both matrix switches to a certain electrode pair and then reads the ADC value for it. After that it switches to the next pair and thus cycles through all connected sensors. Each ADC read is accompanied by a time stamp for the read. During all this the device keeps listening for signals from the GUI in an asynchronous fashion. A stop signal can be received at any time and is executed immediately.

The data is sent in a simple format described in the Listing 4.1. One line contains timestamps and values for all n connected sensors separated by one whitespace. The line ends with the newline character.

Listing 4.1: The data format contains timestamps and values separated by a whitespace.

```

1 <time 1> <value 1> <time 2> <value 2> ... <time n> <value n> \n
2 <time 1> <value 1> <time 2> <value 2> ... <time n> <value n> \n
3 ...

```

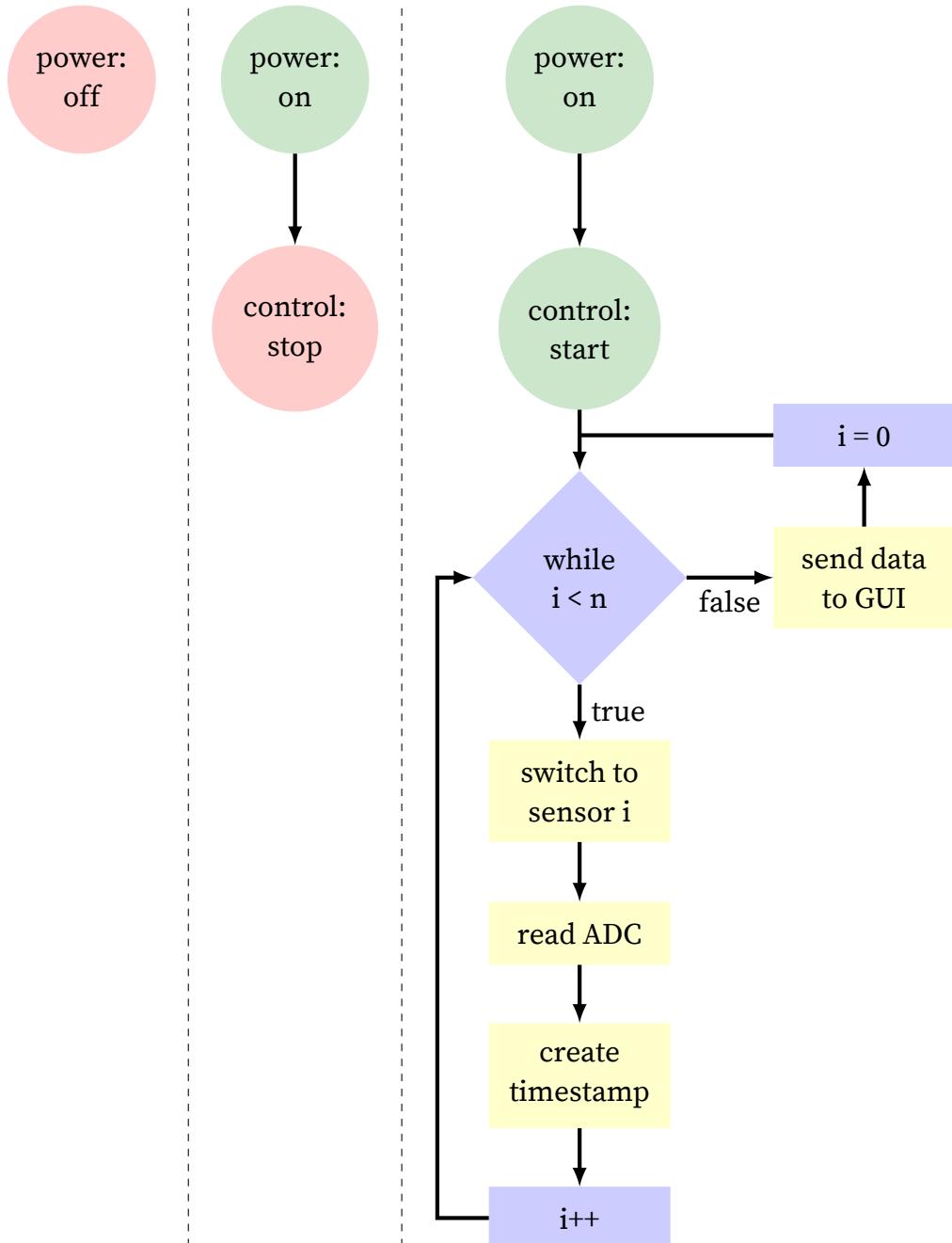


Figure 4.13.: Embedded Software Flow Diagram - If the power state of the device is "on" and the control state is "start" data is collected. The control state is set by a signal sent from the GUI and can be changed asynchronously. Data collection is done in a while-loop that runs as long as the iterator i is smaller than the number of sensors n . In the loop, first the Matrix Switches are set to the i -th sensor, then the ADC is read and afterwards a timestamp is created. When the cycle through all sensors is finished the data is sent to the GUI. Then the loop is started again. This happens until either the control state switches to "stop" or the power state to "off".

4.8. OpenSalinity GUI

The OpenSalinity GUI is a Graphical User Interface (GUI) designed to simplify and aid the usage of the sensor system. It provides following functionality:

- Storing all sensor data to a file.
- Choosing a file to which the data is stored.
- Starting and Stopping the data capture.
- Visualizing the data.

The Save button allows to create the file to be written to and offers a default file name containing the date and time of the creation, helping to keep the data logs in order. Once a file to save to is chosen the data capturing can be started. The live visualization is a bar graph showing each sensor's current measurement, allowing to monitor the ongoing experiment. In addition to size, the bars are also color coded, shifting from red to blue with increasing salinity.

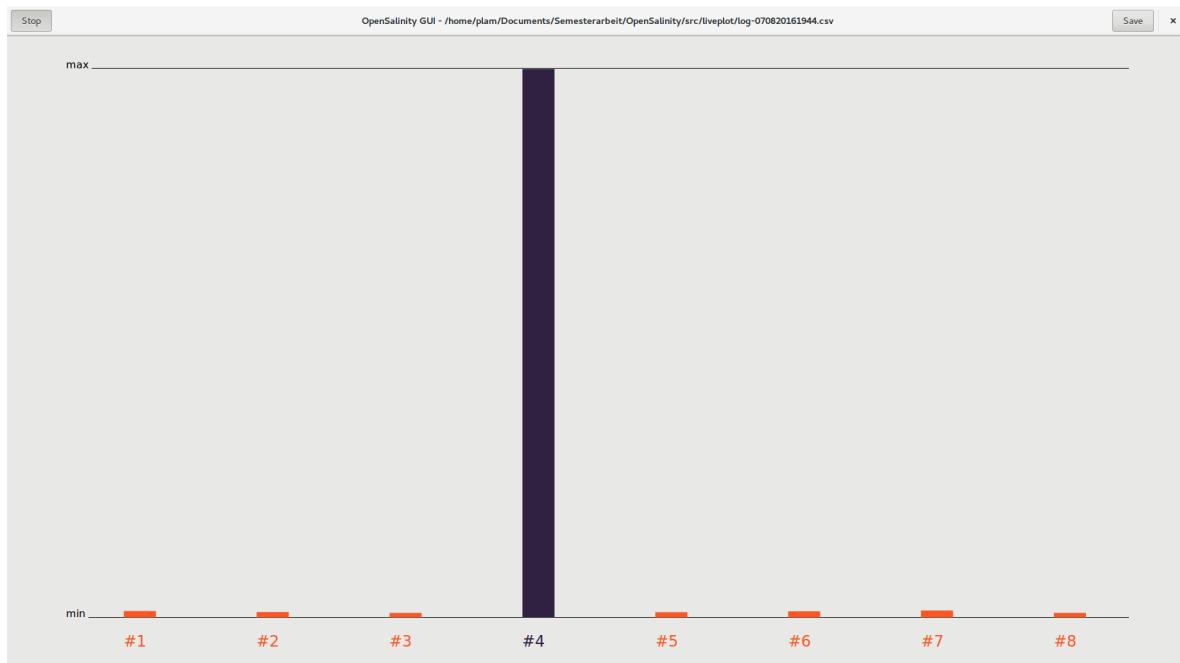


Figure 4.14.: OpenSalinity GUI - in this example sensor #4 measures a resistance of zero (shorted with a wire), all others a infinite resistance (air gap). The Start/Stop button is on the top left, the Save button on the top right. The path to the chosen file is displayed in the header bar.

The software again is written in Python, using GTK+ as GUI toolkit and pyserial to communicate with the microcontroller. It was developed and tested on a Linux-based operating system, however, due the nature of the used programming language and toolkits it is cross-platform and can be run on Windows and OSX also.

In addition to or as replacement of the GUI, a set of standard command line tools can be used. These tools allow for a redundant capturing of the sensor data and for an alternative visualization. Usage of those is less intuitive but they are more robust and faster than the GUI. A detailed manual on how to use those tools as well as the GUI is provided in the Appendix.

4.9. Data Conditioning

Before the data can be analyzed it has to be conditioned to deal with certain peculiarities of the embedded software. This is done in post-processing rather than during

the data capturing to minimize overhead and retain high sampling rates.

The microcontroller measures time since start up by counting clock cycles. These are converted to microseconds stored as an integer value. The microcontroller is a 32-bit architecture, however the Micropython implementation uses 2 bits for data type identification, leaving 30 bits for data. This means the counter can count from -2147483647 to 2147483647 , resulting in a wrap around roughly every 8.9 minutes. In the data conditioning phase, these wrapping times are translated to a rolling time since start-up. This is done by scanning for the first time value that is smaller than its predecessor and then adding the value of 2^{30} to all following values. This is repeated until all time values are corrected. Figure 4.15 visualizes the algorithm.

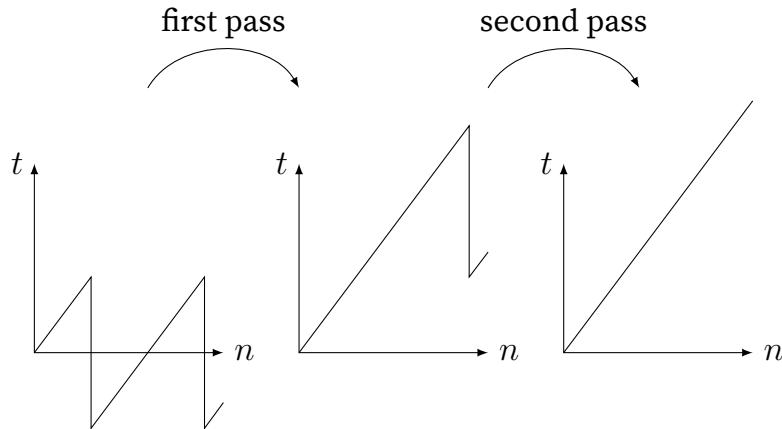


Figure 4.15.: Fixing the integer wrap of the time variable.

4.10. Bill of Materials

The bill of materials collects all used components, their price and a possible source in Table 4.1. Except for the MinieC multiple sources are available for all parts. In case it becomes unavailable, the hardware design files are provided, so they could be manufactured by a different source if need be. The Espruino Pico also comes from a small manufacturer and may not be available in the future. However, alternatives are plentiful and can be used either with simple jumper cables or by redesigning the carrier board to accommodate the different footprint.

The carrier board and sensor strip are made to order by PCB makers. Generally, any manufacturer will be able to produce the boards with the provided design files, however the size limits and corresponding prices can vary. The prices also strongly depend on lot size. The prices in the table are the prices per piece for a lot of ten carrier boards and 16 sensor strips.

Table 4.1.: Bill of Materials

Nr.	Qt.	Name	Source	Price, €/pt
1	1	Espruino Pico	watterott.com	26.95
2	1	Sparky's Widgets MinieeC Interface	sparkyswidgets.com	21.66
3	2	Analog Devices ADG738	mouser.de	3.89
4	1	Carrier Board	dirtypcbs.com	4.87
5	2	Sensor Strip	leiton.de	23.53
6	1	USB Extension Cable	amazon.de	5.49
7	1	Flat Ribbon Cable 40-pole	amazon.de	7.99

4.11. List of Software

Table 4.2 lists all software used in this project, including the version numbers. All involved software is Open Source and can be obtained freely. The packages used to analyze and plot the data are provided at the end of the list, however, the documented data format should enable the user to replace those with whatever tools he or she is most comfortable with. All software written during this project are provided under the MIT license.

Table 4.2.: List of Software

Name	Version	Source	Purpose
MicroPython	1.8.2	micropython.org	microcontroller firmware
Python 3	3.5.2	python.org	GUI and data analysis
PyGI 3	3.20	wiki.gnome.org/Projects/PyGObject	GUI toolkit
pySerial 3	3.0.1	pythonhosted.org/pyserial	host to microcontroller communication
NumPy 3	1.11.0	numpy.org	data analysis
SciPy 3	0.17.0	scipy.org	data analysis

5. Results

In this chapter the verification and validation of the developed sensor system are described. The verification is a set of tests designed to review whether or not the system meets the requirements specified in the Chapter Objectives. The validation is done by analyzing the data gathered during a number of experiments in the bioreactor.

5.1. Verification

5.1.1. Spacial Resolution

| The system shall have a spacial resolution in the order of 10 mm.

An inspection of the physical size of the sensor and their placement on the array, drawn in Figure 5.1, shows us a sensor size of 10x11 mm arranged on a grid of 25x50 mm. The theoretical minimal grid size achievable with the chosen sensor size is 11x12 mm when a minimum distance between electrodes of 1 mm is kept. This grid size is the physical limitation of the spacial resolution and by being well within the order of 10 mm meets the requirement.

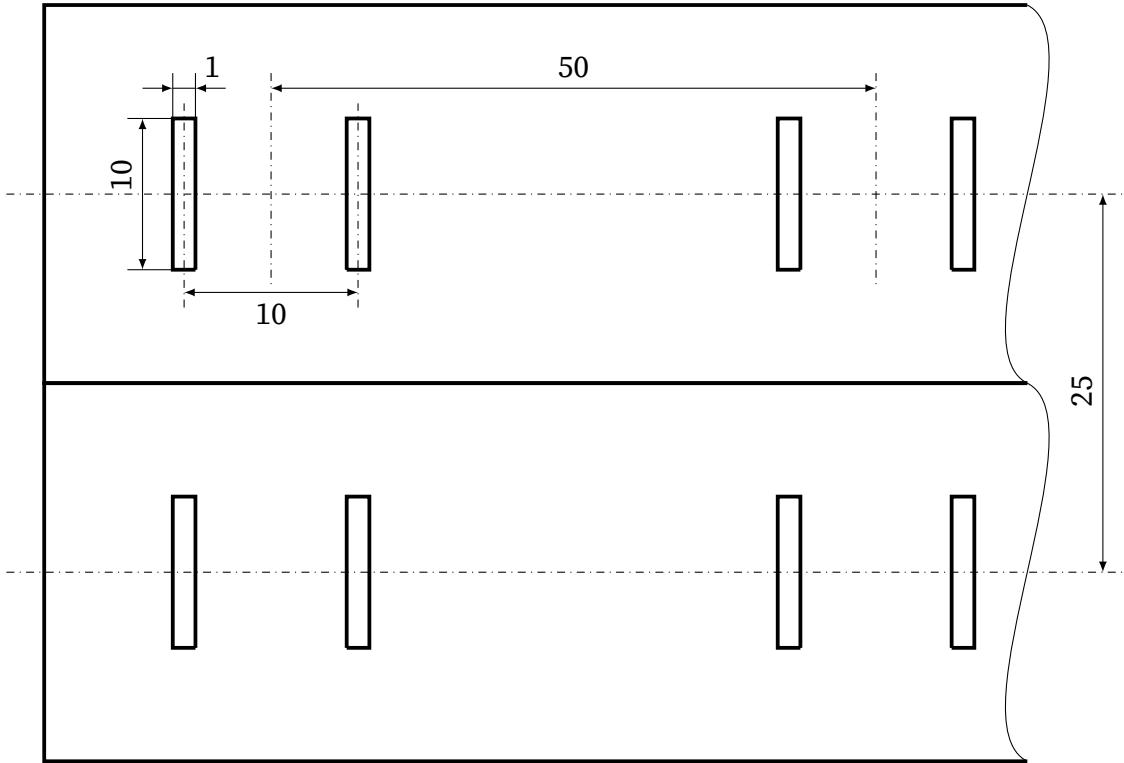


Figure 5.1.: A dimensioned drawing of the sensor grid made up by two strips placed next to each other, not drawn to scale and cut on the right side. The sensor has a size of 10x11 mm, the horizontal distance between the sensors is 50 mm and the vertical distance 25 mm.

Due to the coupling of the spacial and time resolutions via the streams velocity as shown in Equation (5.1), the grid size is not sufficient to verify the requirement.

$$v = \frac{ds}{dt} \quad (5.1)$$

A spacial resolution of 10 mm would require a time resolution of 0.01 s for a velocity of 1 m/s. However, this assumes the size l of the sensor and the time τ a measurement takes to be infinitesimal small, which in reality they are not. While the sensor measures, the flow continues and instead of measuring the conductivity of a certain volume at a certain time, an average is measured. Figure 5.2 visualizes this issue.

To accommodate for that, factors n (5.2) and k (5.3) are introduced, describing the ratio of the resolutions to the actual sizes.

$$n = \frac{l}{ds} \quad (5.2)$$

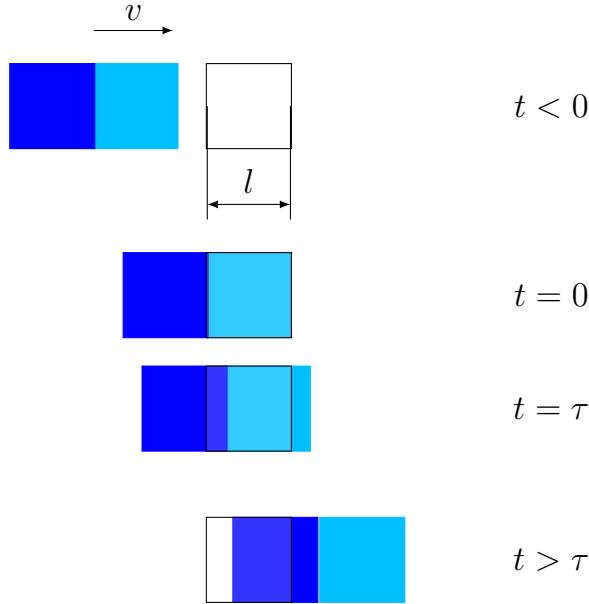


Figure 5.2.: The control volumes in blue have the same size as the sensor (black rectangle). At $t = 0$ the measurement starts and the first control volume is directly over the sensor. When the measurement ends at $t = \tau$ the control volumes have moved. The sensor is covered by a mix of the two volumes.

$$k = \frac{\tau}{dt} \quad (5.3)$$

Inserting k and n in Equation (2.1) yields:

$$v = \frac{ds}{dt} = \frac{k \cdot l}{n \cdot \tau} \quad (5.4)$$

The ratio r

$$r = \frac{k}{n} \quad (5.5)$$

is the ratio of the time it takes a control volume to enter and leave the sensor area to the time a measurement takes. Its inverse is the movement of the control volume during the measurement time and thereby the percentage by which the measured volume is bigger than the control volume. A ratio of one would mean the measured volume is two times the sensor size, thereby halving the spacial resolution. To keep the resolution close to the sensor size, a ratio r of four, increasing the measured volume by 25 % is viable. With the given sensor size and flow speed, a measurement time τ of

2.5 ms is needed.

To test if the time resolution matches this, the measurement time itself can be timed. First, the difference between two adjacent times in the data set in Listing 5.1 is calculated. The mean of the differences through the complete data set is $748 \mu\text{s}$. This time includes the $520 \mu\text{s}$ it takes to switch from one electrode pair to the next. The resulting net measurement time is $228 \mu\text{s}$, the resulting ratio r is 43.9. This means that the measurement is more than ten times faster than required, showing that the time resolution easily matches the spacial resolution. The measured volume is only 2.28 % bigger than the control volume.

Listing 5.1: An excerpt of measurement data showing three lines of data from eight sensors. The long numbers are the times at which the measurements were taken in microseconds measured from the start, the short numbers are the measured values.

```
1 975209221 21 975209961 59 975210676 15 975211397 0 975212119 0 975212840 26  
    975213554 59 975214276 9  
2 975215877 57 975216602 42 975217324 0 975218046 0 975218761 0 975219482 61  
    975220204 43 975220919 0  
3 975222520 52 975223271 0 975223993 0 975224708 0 975225429 45 975226150 53  
    975226867 0 975227583 0
```

5.1.2. Electrical Conductivity Resolution and Range

- | The system shall have a sensitivity of 0.1 % salinity.
- | The system shall have a range from 0 to 5 % salinity.

Table 5.1 and Figure 5.3 show the results of measurements of solutions with different salinities. A 10 % solution was created by adding 5 g of salt to 50 ml of demineralized water. The subsequent solutions resulted from diluting the original one by adding more water.

The measurable change of the voltage was visible at the drop from a salinity of 5 % to 2.5 %. Between 10 % and 5 % no difference was measurable at all. The upper range limit can therefore be placed between at 2.5 %, above those values the sensor is saturated, providing only the information that the salinity is above the threshold.

On the lower limit a measurement without any water resulted in a voltage of 0.02 V. This has to be considered the level of noise in the system. Above that, a clear distinction of salinities of 0.08 % and 0.16 % can be seen.

Table 5.1.: Range and Sensitivity

Salinity in [%]	Voltage in [V]
10.00	0.95
5.00	0.95
2.50	0.85
1.25	0.65
0.63	0.45
0.31	0.28
0.16	0.18
0.08	0.11
Air	0.02

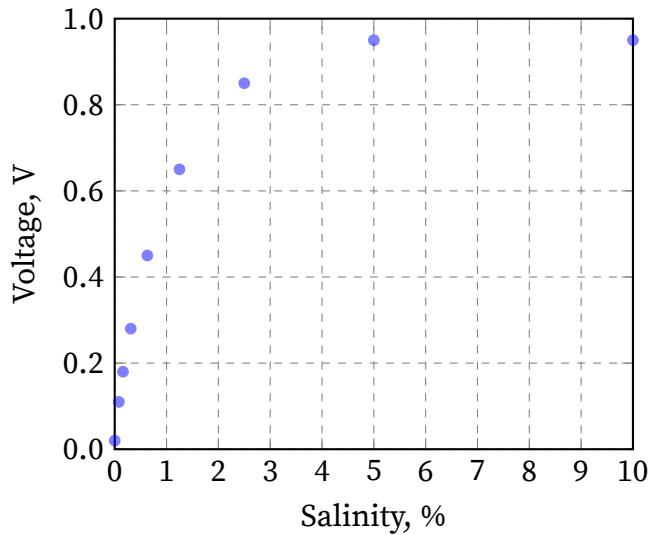


Figure 5.3.: Measured voltages over the increasing salinity show the saturation of the sensor.

While the requirement for sensitivity is clearly met, the range is just barely acceptable. For validation, the used range has to be adjusted and will show if it is sufficient.

5.1.3. Cost

| Requirement: The cost per sensor shall be less than €25

Using the bill of materials in Chapter 4.10 a total price of €121.8 for all materials for a sensor system with eight sensors can be calculated, resulting in a price per sensor of €15.23. Adding additional sensors after that would cost €11.17 per sensor, as parts like the microcontroller do not have to be duplicated.

This price does not include the host PC as the PCs already installed in the laboratory can be used, as well as personal laptops like it was done in the testing phase. The demands on the host PC are very low, so in the worst case even a single board computer like the Raspberry Pi or similar can be used.

5.1.4. Deployability in the Bioreactor

| The system shall be deployable in the algae reactor.

The deployability was demonstrated during two test campaigns on the reactor. Figures 5.4 and 5.5 show the set up during the tests. The host PC can be placed on a desk next to the reactor and be connected to the carrier board with a USB cable. The carrier board can also be placed on the desk or temporarily be mounted on the reactor. The sensor strips connect to the carrier board with long cables and are attached to the reactor with electrical tape.



Figure 5.4.: The sensor strips are mounted to the bioreactor with electrical tape. The cables run towards the carrier board.

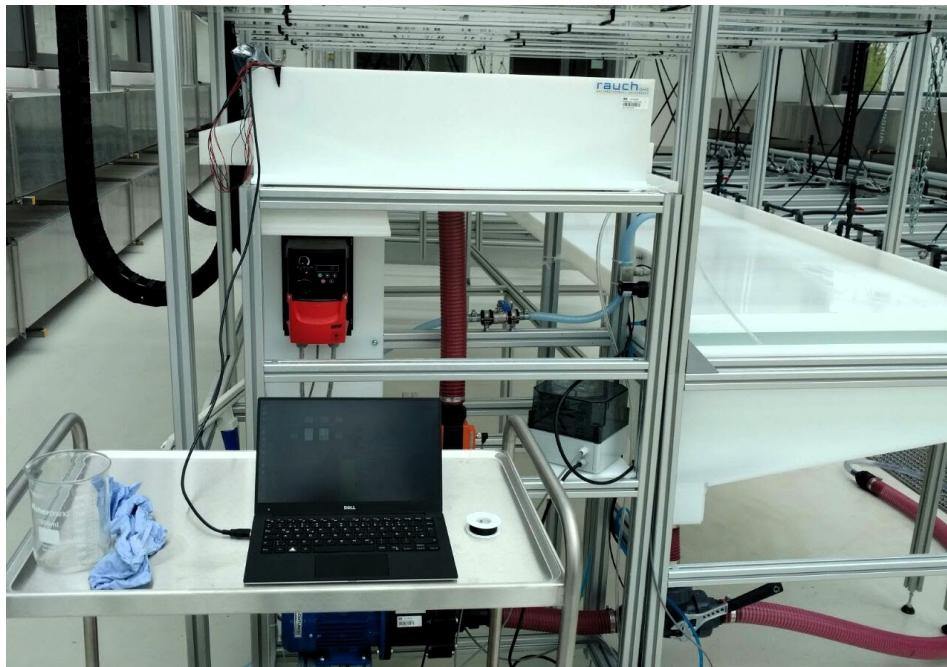


Figure 5.5.: A laptop is used as host PC. A USB cable connects to the microcontroller on the carrier board, which is mounted on the side of the inlet basin.

5.1.5. Usability

| The system shall be usable with a minimal set of written instructions.

The instruction manual for taking a series of measurements can be found in the Appendix A.1. First, an image describes the different parts of the system relevant to the user. After that a step-by-step description with images helps the user through the process. In a test, a person not familiar with the device, but generally familiar with laboratory equipment, was able to successfully conduct measurements within minutes by following the manual. However, there is to note that errors in the software occurring in certain situations can quickly destroy this ease of use. Further testing needs to be done to surface those errors so they can be fixed.

5.2. Validation

The following chapter describes a number of experiments conducted using the developed sensor system. The data is analyzed and interpreted to validate if the results are reasonable.

All experiments were done on the same day on the same bioreactor. The schematic in Figure 5.6 shows how the sensors were positioned.

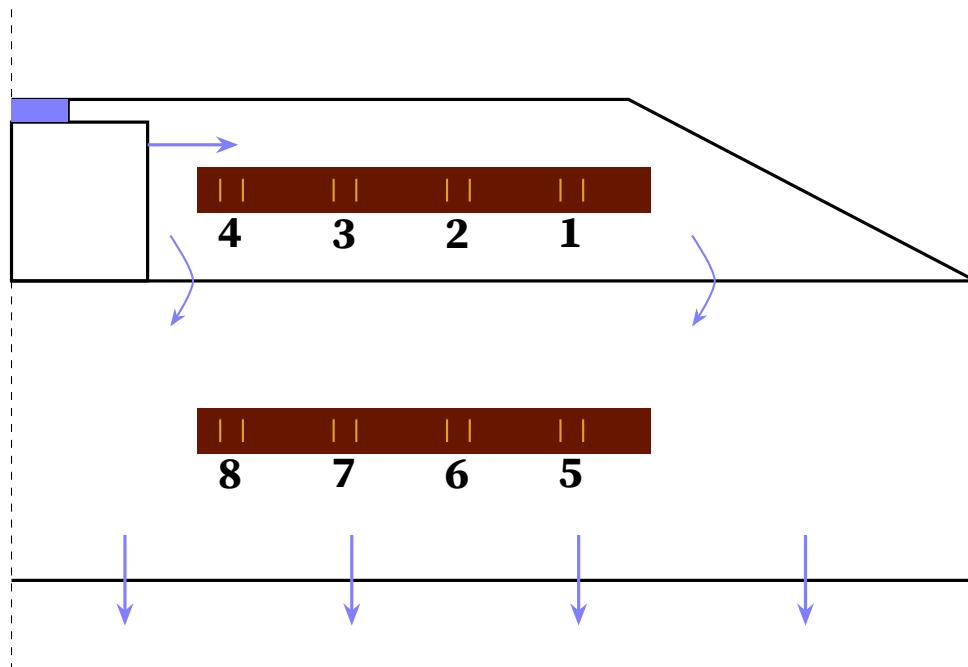
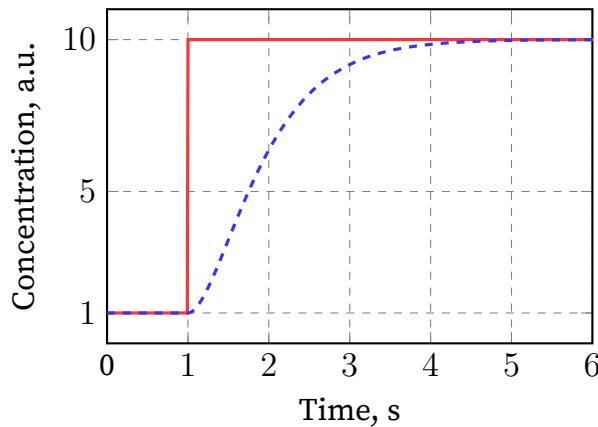


Figure 5.6.: Position of the sensors in the inlet basin [by Timm Severin].

With this configuration, two types of experiments were run: the first method uses the bioreactors plumbing to switch the feed from the collection tank to an external tank containing saltwater. Instead of pumping water in a loop, saltwater from the tank is added to the water cycle and the increase in salinity can be measured. Figure 5.7 shows the step change of the concentration in the input feed and the expected response of the system.

**Figure 5.7.: .**

]Step change of the salinity of the feed (—) and the expected response (···) [by Timm Severin].

The second method retains the original water cycle with no switching to an external feed, but adds an impulse of saltwater in the return basin instead.

In the following section a selection of the experiments will be described. The numeration equals the numeration of the collected data sets. By keeping the original notation continuity between the data files, notes and documentation can be maintained.

Test No. 2

In this test the feed switch method was used. The total amount of water in the tank was 65 l and the flow rate was $\dot{V} = 1.6 \text{ l/s}$. The saltwater in the external tank had a salinity of 1.5 %.

Figure 5.8 shows a plot of the a slice of data taken by sensor eight, located on the inner side of the weir and is used to illustrate the filter design. The aim of the filter is to create an envelope tightly following the maxima of the signal. In the first step, a reduced data set is constructed from the maximum values over small slices of the raw data. A detailed explanation of this algorithm can be found in the Section A.2 of the Appendix. The second step applies a smoothing filter developed by Savitzky et al. [1964] to create the final graph.

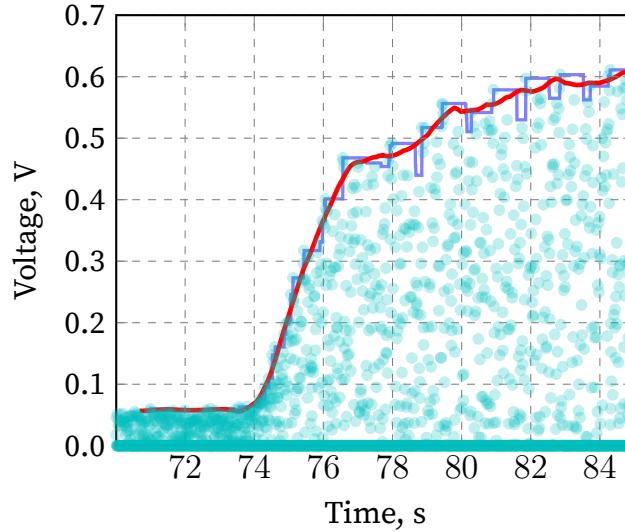


Figure 5.8.: Filter design - The scatter plot shows the raw data (—), filtered for local maxima (—) and smoothed with a Savitzky-Golay filter (—).

In Figure 5.9 the data for sensors one and eight are compared. In the left plot (a) a difference in the scale of the graphs can be seen. This scaling offset is a result of the different internal resistance along each sensor's conductive path as well as the amount of water over on the electrodes. On the weir, there is just a thin water film, while the basin is much deeper. For our comparison this scaling is unfavorably as the slopes of the signal changing over time are of interest, which are influenced by the absolute values. To counter this, all data is calibrated by calculating a mean value over an undisturbed data sample and scaling it to the first sensor. The plot (b) on the right shows the scaled data. Section A.2 in the Appendix describes the method in detail.

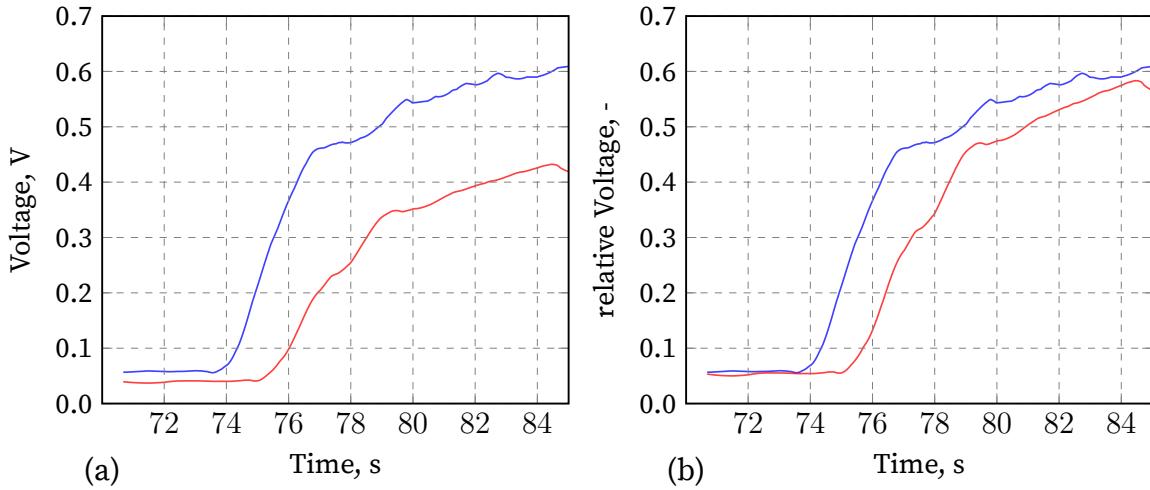


Figure 5.9.: Sensor calibration - Sensors one (— blue —) and eight (— red —) are compared uncalibrated on the left (a) and normalized on the right (b).

Figure 5.10 shows a comparison of the resulting data for sensor four and eight. Signal four changes first with signal eight trailing those changes with a delay. As sensor four is closer to the inlet, this behavior matches the expectations. The first big impulse happens directly after the feed is switched. The drop afterwards and the leveling off suggests a blending of the added saltwater with water still in the system, however this behavior is not yet fully understood.

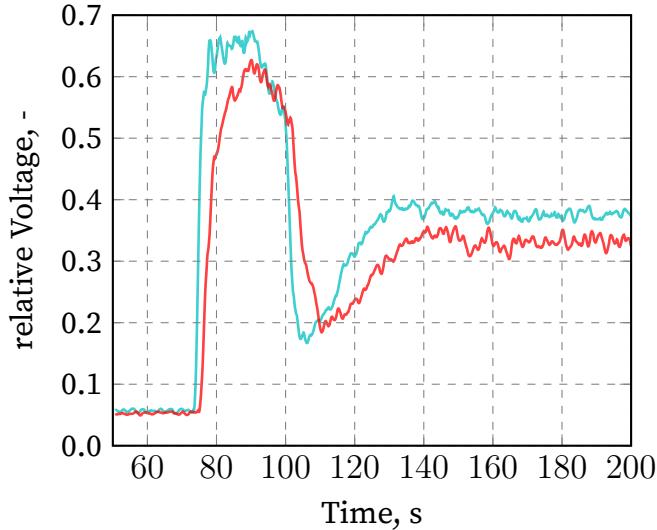


Figure 5.10.: Test No. 2 - Sensors four (—) and eight (—) are compared. Signal four which is measured further upstream always changes before the downstream signal does.

Another interesting data point is the behavior when the water flow is first switched on. Figure 5.11 shows the signals during this phase. Signal four in the inlet basin rises first, signal eight rises about $dt = 4$ s later. This is the time it takes the empty basin to fill up completely, only after which the water flows over the weir. At a flow rate of $1.6 \frac{1}{s}$, this suggests a volume of the inlet basin of around 6.4 l. The real volume calculated according to the schematic is about 5.8 l.

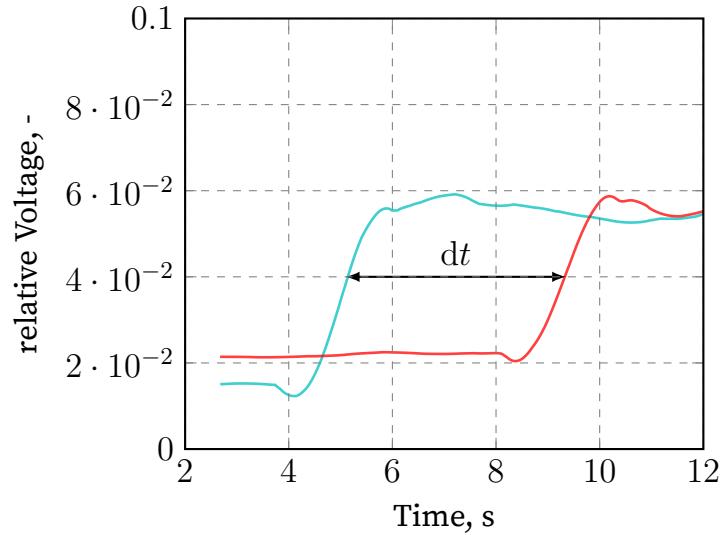


Figure 5.11.: Measurements of sensors four (—) and eight (—) during the start phase of the water flow. Signal eight rises only after the inlet basin is completely filled which takes about $dt = 4\text{ s}$.

A comparison of six different sensor outputs with data from the simulation done by Timm Severin is shown in Figure 5.12. The time segment shows the salinity increase happening directly after the feed is switched to the salt water tank. Generally, the behavior of both correspond well, with the big exception of the position one. Here, the simulation and experimental data are offset by around 2 s. The reason for that is not yet perfectly clear but it is suspected that adverse effects in the simulation are at fault.

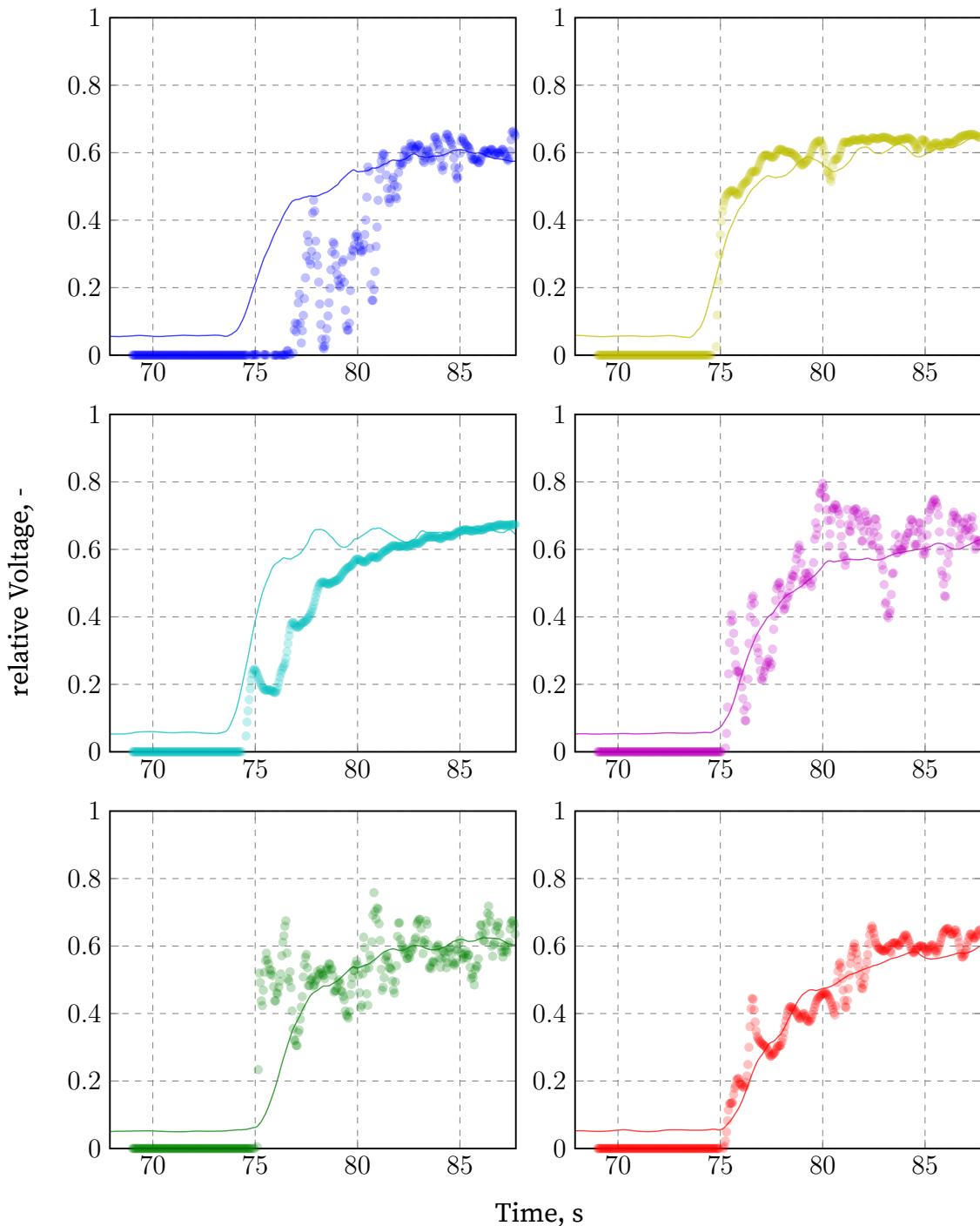


Figure 5.12.: The graphs show the data of sensor one (—), two (—), four (—), five (—), six (—) and eight (—), compared to the scatter plots of the respective simulation results. The chosen time segment shows the upwards slope of the first rise in salinity after the switching of the feed.

Test No. 3

The next test retained all specifics of the prior one, except for a higher salinity of 1.9 %. In Figure 5.13 again sensors four and eight are compared. The behavior mirrors the previous experiment with the main difference being the higher voltage measured due to the higher salinity of the water from the external tank.

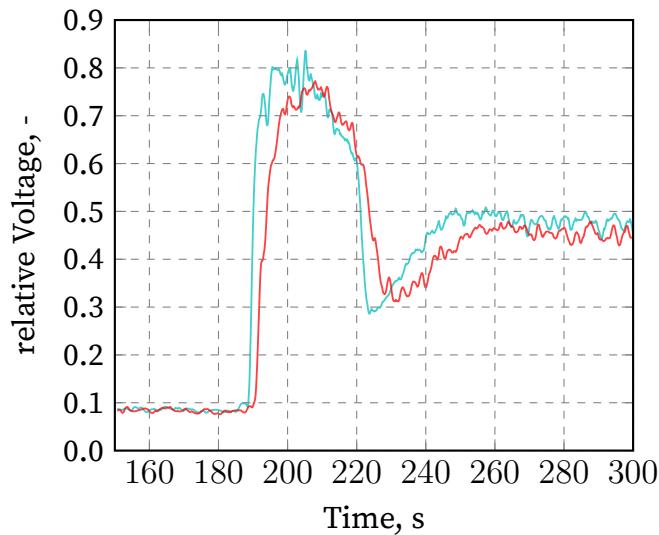


Figure 5.13.: Test No.3 - A higher salinity than before was used. Sensors four (—) and eight (—) are shown.

Test No. 4

While maintaining the previous volume and flow rate, this test used the impulse method instead of the feed switch. There were 51 of water with a salinity of 7.7 % added to the collection tank. The first peak directly follows the addition of the saltwater, after about 49 s a second peak follows. This is the time t_{cycle} it takes the water volume to completely circle through the bioreactor. The third peak is barely visible as the water and saltwater are almost completely diffused.

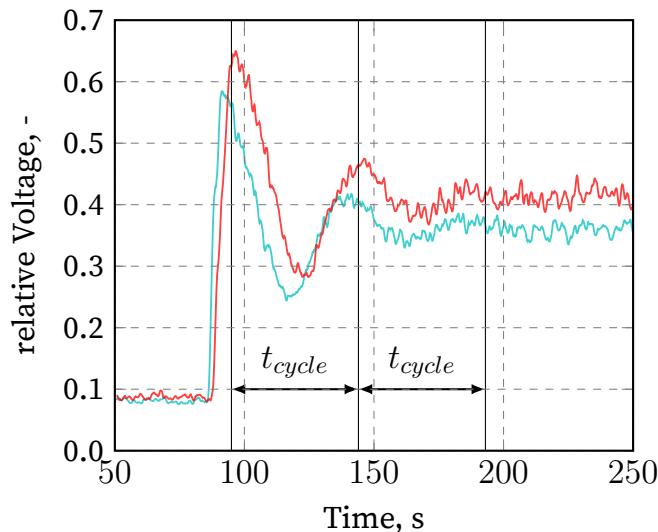


Figure 5.14.: Test No. 4 - The impulse method was used. Sensors four (—) and eight (—) are shown. The time t_{cycle} is about 49 s.

Test No. 6

Again the impulse method was used but with an increased flow rate of $2 \frac{1}{2}$ s and a lower salinity of 6.7 %. The time t_{cycle} from first to second peak is now about 41 s, as expected due to the higher flow rate. The voltages are also a little lower because of the lower salinity.

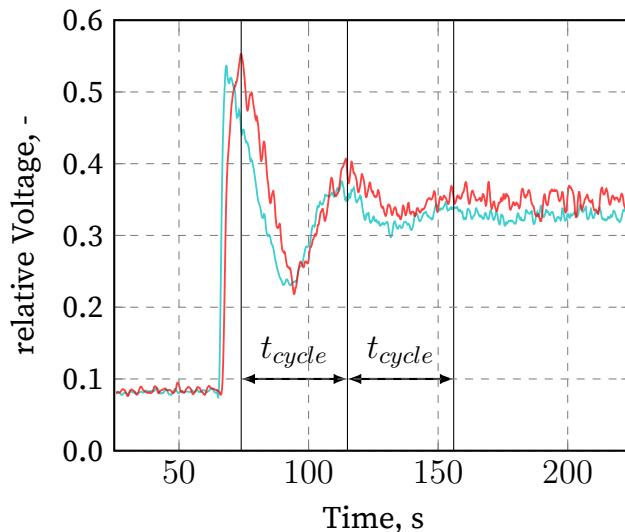


Figure 5.15.: Test No. 6 - The impulse method was used with a higher flow rate and lower salinity than before. Sensors four (—) and eight (—) are shown. The time t_{cycle} is about 41 s.

After proper filtering and scaling of the data, all conducted tests showed results consistent with the expected behavior of the bioreactor. The problem of the limited range of the sensors did not negatively impact on the measurement, as the occurring salinities were lower than initially estimated. Given the new information, a range going only as high as 2.5 % proves sufficient. The developed system provides a low-cost electrical conductivity meter for liquids suitable to be used as an aid to measure and analyze the flow in a photobioreactor and thereby succeeds in meeting the goal of the project.

6. Conclusion

The goal of the project was the development of a low-cost electrical conductivity meter for liquids to measure and analyze the flow in a photobioreactor. After a clear set of requirements regarding the spacial resolution, sensitivity, price and usability were derived from the initial goal, a summary of the theoretical background was provided. A market research identified possible solutions and surfaced a product called MinieC suitable for the task. Around this part the necessary electronics and software were developed to take measurements as well as capture and visualize the data. A suitable solution for the manufacturing of sensors made from electrode pairs was found with flexible printed circuit boards. The resulting device was tested and analyzed to review the compliance with the formulated requirements. The requirement for the measurement range was not met, but the later validation still succeeded, which means it was higher than necessary. The validation was done by conducting several experiments on the reactor and analyzing the resulting data. The behavior indicated by the measurements met the expectations and allowed to derive information about the flow conditions.

7. Outlook

Further development of the sensor system can be done to solve some of the weaknesses of the current system. Due to the nature of the MinieeC in combination with the need of a fast input respond time, the data signal is troubled by high levels of noise. While digital filtering of the captured information makes it usable, these methods always result in a loss of information. One possibility to mitigate this issue is by further increasing the sample rate and removing the diode from the MinieC that filters the negative half of the wave. Doing so would allow to measure the complete oscillating signal, instead of random samples of it. This would also enable to not only measure the amplitude of the signal but also its phase relative to the input, allowing for measuring not only the real part of the resistance, but also the imaginary part, resulting in a complete measuring of the impedance. In order to accomplish this, according to the Nyquist-Shannon sampling theorem, the sample rate would have to be at least two times the oscillation frequency of the input signal, resulting in a sample rate of 3.2 kHz. The existing hardware is capable of that, but the software would need a complete rewrite. A second option to solve this problem is using a dedicated impedance sensor like the AD5933 from Analog Devices. This sensor integrates the same measurement principle the MinieC uses on a single chip, alongside additional logic that analyzes the signal's amplitude and phase to calculate the complex impedance value. A PCB incorporating this part in a fashion compatible with the pin layout of the already developed carrier board is possible, providing a simple update strategy.

A. Appendix

A.1. Instruction Manual

Setting up the Hardware

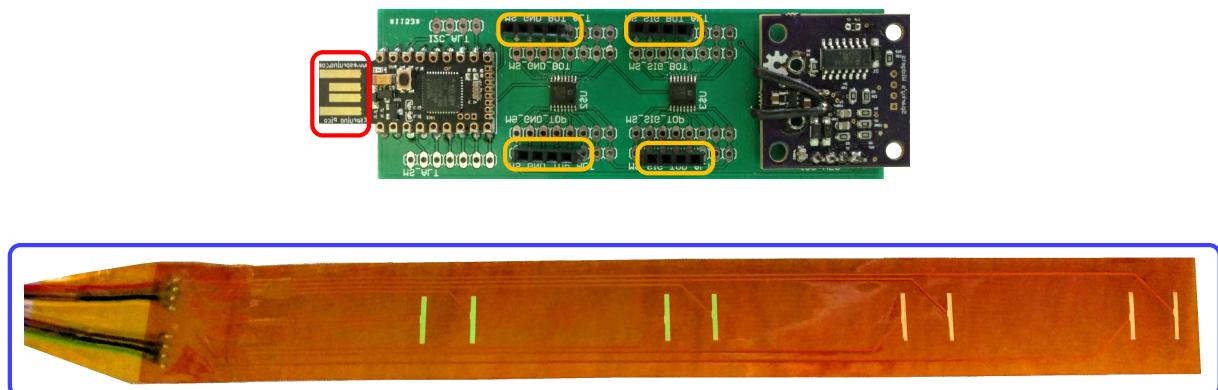


Figure A.1.: Important parts of the system:

USB connection (—)

sensor strip (—)

connectors (—)

- 1 Use electrical tape to mount the sensor strips at desired points of measurement.
Make sure not to cover the electrodes (golden parts).
- 2 Connect the cables of the sensor strip to the carrier board as shown in the picture:

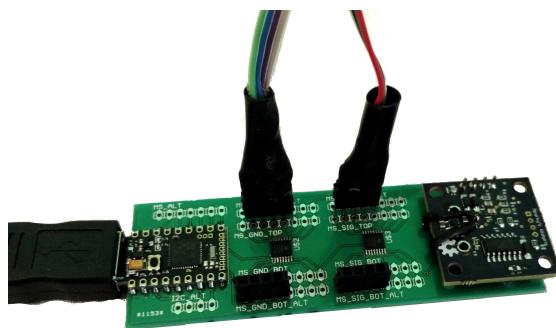


Figure A.2.: The connectors for the sensor strips.

- 3 Connect the carrier board to the host PC with the USB cable. Note the orientation in the picture:

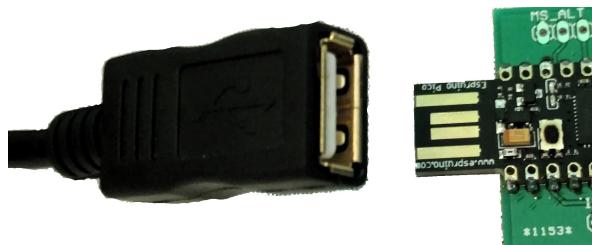


Figure A.3.: The correct USB orientation.

Wrong orientation will not cause any damage, but the system will not work.

Capturing data using the OpenSalinityGUI

- 0 Follow the three steps of "Setting up the Hardware".
- 1 Start the OpenSalinity GUI by clicking on the icon.
- 2 Click "Save" on the top right to chose a file to log the data to. If you want to use the default filename, just click "Open" in the file menu.
- 3 Start the measurement by clicking "Start".
- 4 Stop the measurement by clicking "Stop".

- 5 Repeat from step 1 for all subsequent measurements. If no new file is chosen before starting a new measurement, the data is added at the end of the previous file.

Capturing data using command line tools

- 0 Follow the three steps of "Setting up the Hardware".
- 1 Open a terminal.
- 2 Change the working directory to the directory liveplot.sh is in:

```
1      cd ~/path/to/directory
```

- 3 Execute "liveplot.sh" with a file name as argument:

```
1      ./liveplot.sh log.csv
```

- 4 To end data capturing, click into the terminal and press "Ctrl+C".
- 5 Repeat from step 1 for all subsequent measurements. If the file name is not changed, the existing file will be overwritten.

A.2. Algorithms

Rolling Maximum

The rolling maximum replaces all values with the local maxima within a slice of data. Listing A.1 shows a working implementation of the algorithm in the programming language Python.

Listing A.1: Python implementation the rolling maximum algorithm.

```

1 import numpy as np
2
3 def roll_max(vector):
4
5     for i in range(0, len(vector)):
6         try:
7             # replace each value with the maximum of the next 100 values
8             vector[i] = max(vector[i+1:i+100])
9         except:
10             pass
11
12     # shift vector 100 steps to the right
13     vector = np.insert(vector, 1, [0]*100)[-100]
14
15     return vector

```

Signal Calibration

The signal calibration scales all signals to a common reference. This common reference is the signal of the first sensor. Listing A.2 shows a Python implementation of the method.

Listing A.2: Python implementation the signal calibration.

```

1 import numpy as np
2
3 sensor_data = [sensor_1, sensor_2, ...]    # array of all sensor data
4 start = 100      # start and
5 end = 1000      # end of the slice of undisturbed data
6

```


B. Abbreviations

CFD	Computational Fluid Dynamics
PBR	Photobioreactor
PC	Personal Computer
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
GND	electrical ground
SIG	Signal
PCB	Printed Circuit Board
flex-PCB	flexible Printed Circuit Board
opamp	Operational Amplifier
OUT	Output
SCLK	Clock Signal
DIN	Digital Input
DOUT	Digital Output
D	Drain
S	Signal
SYNC	Synchronization
ADC	Analog-Digital-Converter
AC	Alternating Current
GUI	Graphical User Interface

C. List of Figures

2.1. The inlet basin.	11
3.1. The equivalent circuit diagram.	15
3.2. Two electrodes submerged in the solution.	15
3.3. Fisher Scientific™ Traceable™ Salinity Meter Pen [Fisher Scientific 2016b]	17
4.1. The developed sensor system.	19
4.2. System Design	20
4.3. Two rectangular electrodes form an electrode pair.	21
4.4. A handmade sensor strip.	22
4.5. The sensor strip design to be implemented as flexible PCB.	23
4.6. A finished sensor strip assembled with cables and waterproofing on the left.	23
4.7. The functional block diagram of the ADG738 matrix switch.	24
4.8. Example Timing Diagram	25
4.9. The opamp (OA) modulates the input signals (SIG) amplitude.	27
4.10. The blue graph shows the measured voltage with a resistance of 0.	28
4.11. The scatter plot shows the signal oscillating between 0 and the voltage at a resistance of 0.	29
4.12. The assembled carrier board with all parts directly soldered on.	31
4.13. Embedded Software Flow Diagram	32
4.14. OpenSalinity GUI	34
4.15. Fixing the integer wrap of the time variable.	35
5.1. A dimensioned drawing of the sensor grid.	39
5.2. The control volumes in blue have the same size as the sensor (black rectangle).	40

5.3. Measured voltages over the increasing salinity show the saturation of the sensor.	42
5.4. The sensor strips are mounted to the bioreactor with electrical tape.	44
5.5. A laptop is used as host PC.	44
5.6. Position of the sensors in the inlet basin [by Timm Severin].	46
5.7. Step change of the salinity of the feed and the expected response [by Timm Severin]	47
5.8. Filter design	48
5.9. Sensor calibration	49
5.10. Test No. 2	50
5.11. Measurements of sensor four and sensor eight during the start phase of the water flow.	51
5.12. The graphs show the data of the sensors compared to the scatter plots of the respective simulation results.	52
5.13. Test No.3	53
5.14. Test No. 4	54
5.15. Test No. 6	55
A.1. Important parts of the system.	58
A.2. The connectors for the sensor strips.	59
A.3. The correct USB orientation.	59

D. List of Tables

2.1. Requirements	13
4.1. Bill of Materials	36
4.2. List of Software	37
5.1. Range and Sensitivity	42

E. Bibliography

- Analog Devices (2016). ADG738/ADG739 Data Sheet. 2016.
- Creative Commons (2016). Creative Commons Attribution-ShareAlike 3.0 Unported. 2016. URL: https://creativecommons.org/licenses/by-sa/3.0/deed.en_US.
- Edwards, Ryan (2016). MinieC I2C eC interface. 2016. URL: <https://www.sparkyswidgets.com/portfolio-item/miniec-i2c-ec-interface>.
- Fisher Scientific (2016a). Fisher Scientific™ Traceable™ Conductivity, Resistivity, and TDS Meter. 2016. URL: [https://www.fishersci.com/shop/products/fisher-scientific-traceable-conductivity-tds-meter/093262](https://www.fishersci.com/shop/products/fisher-scientific-traceable-conductivity-resistivity-tds-meter/093262).
- Fisher Scientific (2016b). Fisher Scientific™ Traceable™ Salinity Meter Pen. 2016. URL: <https://www.fishersci.com/shop/products/fisher-scientific-traceable-conductivity-tds-salinity-meter-pens-2/15078202>.
- Gevatter, HJ (2000). Automatisierungstechnik 1. Meß-und Regeltechnik. 2000.
- MicroPython (2016). MicroPython project homepage. 2016. URL: <http://micropython.org/>.
- Savitzky, Abraham and Marcel JE Golay (1964). ‘Smoothing and differentiation of data by simplified least squares procedures.’ In: Analytical chemistry 36.8 (1964), pp. 1627–1639.
- Tränkler, Hans-Rolf and Leonhard M Reindl (2015). SensorTechnik: Handbuch für Praxis und Wissenschaft. Springer-Verlag, 2015, p. 1152.