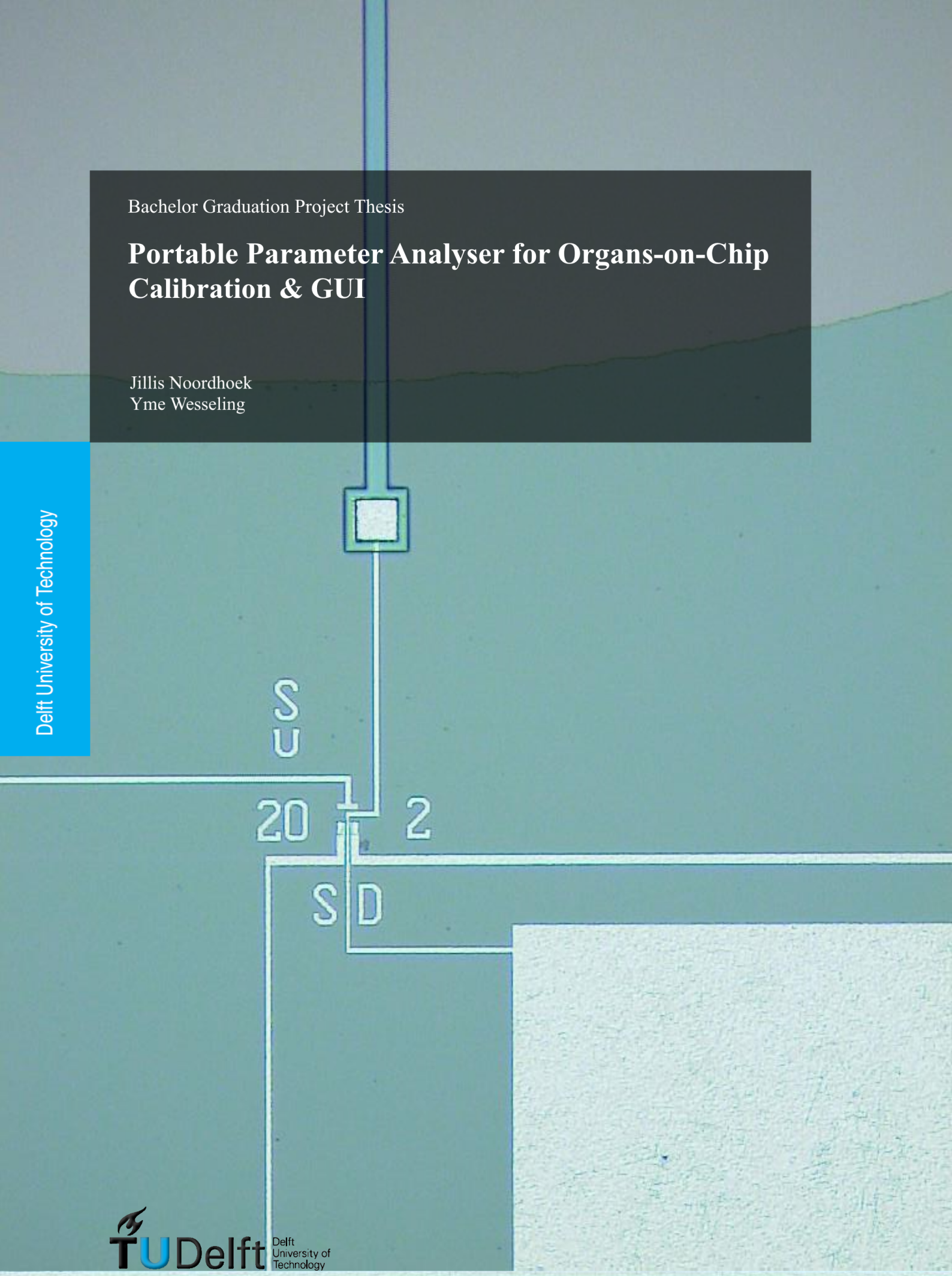
Bachelor Graduation Project Thesis

Portable Parameter Analyser for Organs-on-Chip Calibration & GUI

Jillis Noordhoek
Yme Wesseling

Delft University of Technology



TU Delft Delft University of Technology

Challenge the future

TU Delft

Calibration of Portable Parameter Analyser for Organs-on-Chip

Bachelor Thesis

Jillis Noordhoek (4567951)
Yme Wesseling (4569318)

Supervisors
Dr. Massimo Mastrangeli
Dr. Marco Spirito
Hande Aydogmus
Carmine De Martino

Thesis committee
Dr. Nuria Llombart-Juan
Dr. Massimo Mastrangeli
Dr. Marco Spirito
Carmine de Martino

Tuesday, 30th June, 2020
Version: July 23, 2020

UNDER EMBARGO UNTIL 1st May, 2021

DELFT UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRICAL ENGINEERING, MATHEMATICS AND COMPUTER SCIENCE

ELECTRICAL ENGINEERING PROGRAMME

Summary

Until now parameter analysers for Organ-on-Chips (OoCs) have needed bulky multi-probe setups that do not fit in biological research labs. For this reason a bachelor graduation project was proposed to get one of the sensors designed by the Electronic Components, Technology and Materials group (ECTM) out of the lab and into the hands of potential end users. This thesis is one of three from that project, and describes the calibration and user interface components of the portable parameter analyzer that is developed for the OoC sensor.

First, an analysis is performed on the amplifier design that was given with the sensor. The analysis showed that the biggest sources of error in the overall gain are the offset and gain error, while non-linearity was not significant. Therefore, a two-point calibration method was deemed sufficient for the amplifier calibration. It is performed by taking two reference voltages as input of the amplifier, and measuring the corresponding output. With those points the actual gain and offset voltage can be calculated and corrected in measurements.

Because of circumstances it was not possible to test in a lab environment whether the amplifier and the two-point method would meet the requirements. Therefore a second calibration method is proposed, the 'sweep' method. For each input voltage step the corresponding output voltage is measured. This mapping can be stored in memory, and any future measurement can be looked up to find the correct voltage. The sweep method can also be used with a slight modification of the current hardware in order to simply plot the gain of the amplifier, to verify that it is linear as intended.

Because the portable parameter analyzer is operated remotely, there was a need to develop a communication protocol on top of the Bluetooth link, in order to allow for parallel development of the GUI and embedded software on the analyzer. Once the communication between GUI and analyzer was defined, it was also possible for the other group to calculate the power consumption of the communication module.

Finally, a Graphical User Interface (GUI) needs to be developed that can interact with the analyzer (connect, change settings, retrieve data, etc.) and it should display and store the measured data. A framework called Qt [1] is chosen for developing the GUI, and a graphical design was made. Two modules are implemented in the GUI: A Bluetooth scanner to connect to the analyzer, and a way to plot data from the analyzer.

Preface

It should be noted that the circumstances around this bachelor graduation project are different from previous years due to COVID-19. The groups felt that being limited to communication over voice and video channels diminished the amount of teamwork and interaction with the supervisors.

Nonetheless, we would like to thank our supervisors for guiding us through this project and creating a friendly online environment to work in. They set up such a unique project to work on, and later worked round-the-clock, even answering questions and giving valuable feedback during the weekends.

Contents

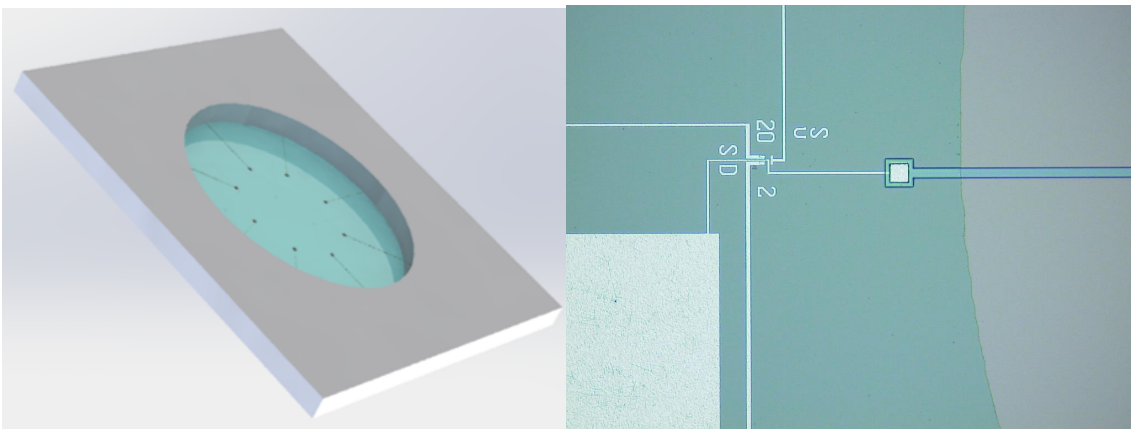
Chapter 1

Introduction

Looking at the future, it is hard to see humanity ever testing medicine on animals or humans. Aside from a seemingly endless debate on the ethics of animal testing [4], using animals for testing is not as simple as is commonly thought and can fail to produce meaningful results [5]. More and more evidence is showing that the use of Organ-on-Chip (OoC) technology could be a viable alternative to traditional models used in drug or cosmetics testing [6–8]. OoC technology enables rapid testing while mimicking dynamic biological environments. However, at this stage, many of the OoC technologies are tested in the research environment without the end-user in mind. The measurement setups are complicated and hard to transport, due to being connected to power supplies and bulky measurement equipment [8].

At the Electronic Components, Technology and Materials (ECTM) group, work has been proceeding to develop an OoC sensor. The design is now at a stage where a solution can be developed to take the sensor out of the initial research environment, into a complete product that incorporates the sensor.

The sensor in question consists of eight Floating Gate Field Effect Transistors (FGFET) designed to measure the changes in electrochemical charges in a (bio)chemical solution. As can be seen in Figure 1.1, the floating gates of the FGFETs extend into the sensing area, such that charges accumulated on the gates will modulate the gate-source voltage and therefore change the FGFET threshold voltage and drain current [15].



(a) (b)

Figure 1.1: a) Model of the sensor showing the 8 floating gates extending into the sensing area. b) Close-up made during the fabrication process, showing one of the FGFETs with the floating gate extending to the right of the figure [15].

The goal is to incorporate this sensor in a measurement device hereafter referred to as Portable Parameter Analyzer (PPA). The the end-user should be able to operate the PPA with limited training and without the use of expensive peripheral equipment with redundant functionality or complexity. Such a PPA will

need to be capable of measuring very small currents in the sub-micro ampere range while operating in human-body conditions, such as temperature, humidity and CO2 levels.

1.1 Project description

In order to get the sensor and amplifier design from ECTM into the hands of biomedical researchers, an electrical engineering bachelor graduation project was proposed. The project is divided into three parts, each to be researched and developed by two persons. The first team will model the sensor and assess its specifications and areas of potential improvement. The third team will research the power requirements of the system and write code for the embedded micro-controller. Finally, the subject of this thesis is the work from the second group. We will research what is necessary in order to get a calibrated measurement. We will also develop the (graphical) user interface.

1.2 State-of-the-art review

As stated before there is a need for portable parameter analysers and it is difficult to find similar projects. The only other similar project found successfully demonstrated current measurements below 1 nA [9]. Unfortunately, this paper does not describe any calibration methods and does not go into detail of the design choices that were made.

However, there are some areas of research that relate closely. For example, [10] describes some of the challenges of amplifier designs for biopotential readout circuits for portable acquisition systems. Such as: low-noise, high Common-Mode Rejection Ratio (CMRR), low power dissipation and configurable gain. Additionally, it is also regarding a wireless system.

In case calibration has to be done with small currents, it is possible to generate small currents in the pico-ampere range, assuming a known capacitance [11]. As seen in this paper the setup is bulky, so if regular re-calibration is necessary a smaller form-factor might be desired.

1.3 Thesis outline/synopsis

First, an overview is given of the requirements and the current system overview. Then, the differential amplifier design is characterized, and the types of errors that should be calibrated are determined. Two potential calibration methods are discussed, of which one method is later implemented and the results discussed. Before going to the discussion and conclusion of the thesis, an overview is given for the communication and user interface design.

Chapter 2

Programme of Requirements

The following requirements were agreed upon together with the supervisors of the project.

2.1 System requirements

1. Must operate within an incubator at 37°C
2. Must operate within an incubator with a CO_2 concentration of 5%
3. Must be resistant to the humidity of the incubator (IP59)
4. Battery-life must last at least 6 days
5. Must sense and amplify the drain current of the FGFET.
6. Must control the drain-source and gate-source voltages of the FGFET.
7. Must sense in the range from 0 to 0.5 mA and control in the order of mV
8. Must sense with a resolution of 1 µA or better.
9. Must be capable of real-time data read-out
10. Must visualize the drain current vs drain voltage (I_D vs V_{DS}) characteristics
11. Shall provide a comprehensible GUI for biologists, see the GUI requirements in 2.4

2.2 System trade-off requirements

1. Minimize power consumption
2. Minimize heat dissipation
3. Minimize noise
4. Maximize current and voltage resolution
5. Minimize volume

2.3 Calibration routine

1. The calibration routine should facilitate a reference measurement as a ground for further measurements
2. The calibration routine should find and cancel the amplifier error
3. The calibration routine should find the sensitivity of the sensor

2.4 Graphical user interface

1. The GUI should show the real-time evolution of the drain current against the drain voltage
2. The GUI must show the real-time evolution of the drain current over time
3. The GUI must show the calibration status
4. There should be a button to initiate calibration routines
5. There should be a button to start measurement routines
6. The data should be logged in a readable format every minute
7. There should be an input field to set a specific measurement period
8. There should be an input field to set a measurement interval
9. There should be an input field to set a specific gate-source voltage
10. There should be an input field to set a specific drain-source voltage
11. The GUI should work on Windows XP and up
12. The GUI should not interfere with other operations running on the computer

Chapter 3

System overview

The goal of the system is to provide a remote readout of a small sensor array to the user. This means that there is a sensing stage, a measurement stage, a communication stage and finally a user interface. With data-processing throughout the stages. Initially a development version of the system is made to try out the system design, and the calibration and measurement routines. Figure 3.2 shows an overview of this system, which uses wired communication and lab supplies to power the system.

The final version of the portable parameter analyser should be able to operate in the environment of the end-user. It will consist of several modules, such as the sensor, an ADC/DAC, a wireless communication module, a GUI and a battery to power it all, as can be seen in Figure 3.2.

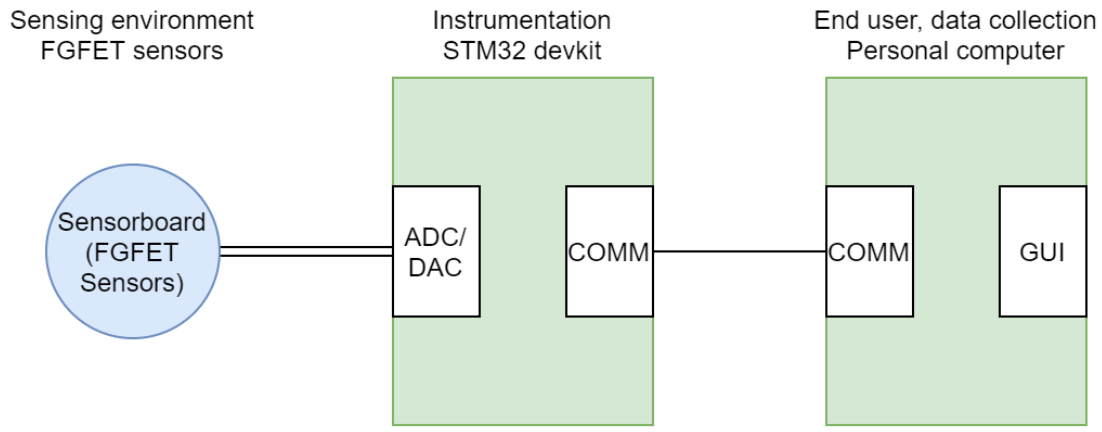


Figure 3.1: An overview of the top level design of the portable parameter analyser, development version

The sensor

Each sensor consists of the FGFET (from the sensing IC), connected to a current sensing resistor on the circuit board of the parameter analyzer. The FGFETs are biased using Digital-to-Analog Converters (DACs), and the drain current is then measured using the current sensing resistor, which is connected to a differential amplifier which in turn is connected to a Analog-to-Digital Converter (ADC).

ADC/DAC

The DAC provides the gate-source and drain-source voltages for biasing the FGFET. The ADC measures the amplified voltage over the current sensing resistor. The output of the ADC is read by the microproces-

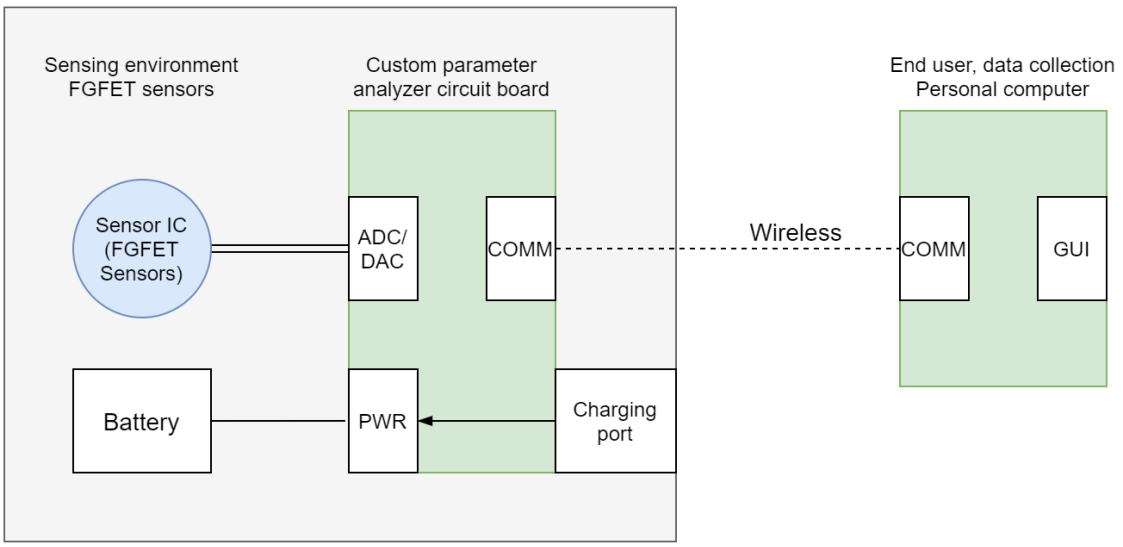


Figure 3.2: An overview of the top level design of the portable parameter analyser

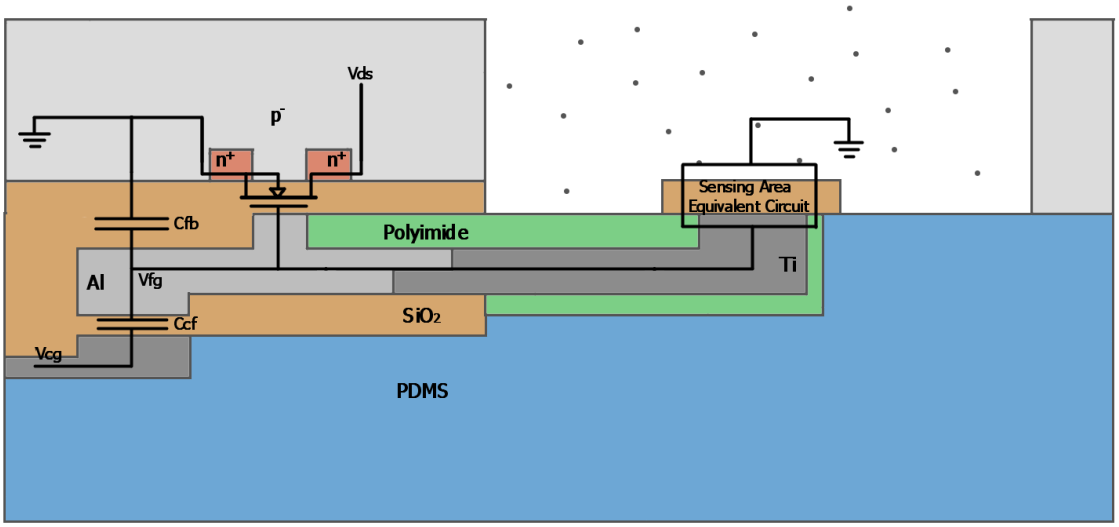


Figure 3.3: Cross-section of the floating gate sensor [3]

sor, which sends it to the communication module so that the data is sent to the user interface for further processing. On the development version, both the ADC and DAC work on 12-bit digital codes.

Communication module

Initially communication is achieved using a wired serial connection between the micro-controller and the computer with the user interface. In the final version, Bluetooth will be used in order to get wireless communication at low power consumption.

Graphical user interface

The graphical user interface will provide the end-user with measurement data in a comprehensible way. It will also give the end-user various input options, such as measurement length, sample rate and levels

for the drain-source and gate-source voltage. The graphical user interface will allow a number of simple measurement functions to be executed on the measurement data.

Power supply

The power supply consists of a rechargeable battery, which is connected to the circuit board. This battery will provide power to the ADC, DAC, differential amplifier and communication module. A rechargeable battery is used to enable the total parameter analyser to function completely within a closed environment, without any wires connecting to the outside world. Batteries and wireless operation are necessary for two reasons: First, to allow the analyzer to work in different setups with minimal modifications, such as a pressure-sealed connector on the incubator. Secondly, to remove possible noise interference that is often transmitted or enabled by wired communication/power supply.

3.1 Measurement method

A change in the surface charge on the sensing region of the FGFET will correspond to a change in the drain current through the FGFET. However, a change in the biasing of the FGFET could also cause the drain current to change. Therefore, to monitor the evolution of the charge in the sensing region the drain current has to be measured at regular intervals at constant biasing settings.

If the range of the drain current to be measured is 0 to 1 mA and a resolution of at least 1 µA as per the requirements, the current range should be converted and amplified to a voltage range of 0 to 3.3 V using a 32.67 Ω resistor. The 3.3 V range is necessary to make optimal use of all the bits in the ADC, and the 32.67 Ω resistor combined with the 101 times gain of the differential amplifier ensures that the current of 1 mA is mapped to exactly 3.3 V in ideal conditions.

In these ideal conditions the resolution requirement would be fulfilled. Because the range is perfectly mapped, the resolution can be taken as $\frac{1mA}{2^{12}} = 244\,\mathrm{nA}$ since a 12-bit ADC is used.

However, in the real world there are a number of factors that will limit the resolution, such as:

- Error in the resistance value (resistor tolerances, trace resistance)
- Error in the gain of the amplifier
- The effective resolution of the ADC
- Error in the supply rails
- Noise
- Offset voltages

From table A.1 it can be seen that the input-referred noise voltage of the amplifier is about 12 nV/$\sqrt{\mathrm{Hz}}$. That means if the resolution and current sensing resistor is known, an estimation can be made on the minimum sampling time in order to remove influences of (thermal) noise. At 37 °C and using a resistor of 32.67 Ω, the noise may not exceed the resolution of 244 nA (or 7.97 uV if the resolution is converted to a voltage resolution using the 32.67 Ω resistor). So the total input-referred noise would be:

$$V_n = V_{n,res} + V_{m,amp} = \sqrt{4k_bTBR} + 12 \cdot 10^{-9}\sqrt{B} = (\sqrt{4k_bTR} + 12 \cdot 10^{-9})\sqrt{B} \tag{3.1}$$

Then

$$B < (\frac{7970}{12.75})^2 = 394\,\mathrm{kHz} \tag{3.2}$$

Due to the small bandwidth of the signals of interest (i.e. the biological processes operate on a larger timescale) the noise can easily be filtered either with software filtering (running average) or a hardware low-pass filter.

3.2 Amplifier

Before the start of this project, a design for an instrumentation amplifier was made at the ECTM group, to be used in the portable parameter analyzer. Figure 3.4 shows the schematic overview of the components that are used to do the measurement, and Figure A.1 shows the given schematic design for the instrumental amplifier (In-Amp). This instrumentation amplifier will amplify the voltage over the current sensing resistor with a factor of 101 to increase the accuracy of the final measurement. Care was taken during the design to obtain a flat frequency response at low frequencies as can be seen from Figures A.2 and A.5.

To verify that this design would be suitable, multiple simulations were done in order to create a simplified 'datasheet'. This datasheet can be seen in appendix A. The next chapter will use the information from the amplifier in order to come up with calibration methods that could remove non-linearity and offset errors.

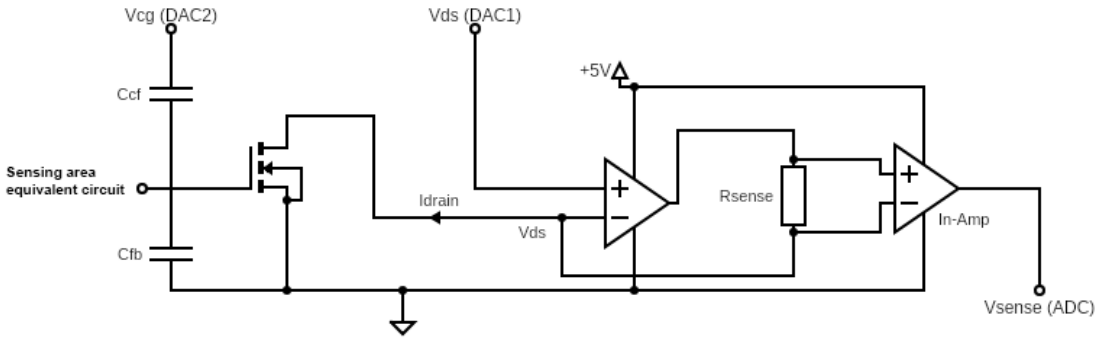


Figure 3.4: Schematic overview of the measurement system [2]

Chapter 4

Calibration methods

Production imperfections, tolerances and drift over time can affect the specifications of the system, resulting in inaccurate measurements. To improve the accuracy of the measurements a form of calibration is needed.

First there will be a look at how much error there is in the gain of the amplifier, then a simple calibration method is proposed in order to counter the error. Additionally, Section 4.3 describes a calibration method that might not be necessary, but can be used in order to simply verify that the gain of the amplifier is as expected.

4.1 Amplifier error

During the analysis of the given amplifier, simulations showed that the amplifier and sensing resistor combination can induce a measurement error of a maximum of 186 LSB, which is only taking into account variation in the resistors, not the OP-Amps used in the differential amplifier. Figure 4.1 shows a Monte Carlo simulation which varies the resistor values in the design using a 1% tolerance, showing that calibration is necessary to correct the measurements and reduce the error in the system. This Monte Carlo simulation was performed in Advanced Design Systems [12] using the circuit shown in Figure A.1.

However, the resistors are not unique in their susceptibility to imperfections that can arise during the production process. For example, the variety in the offset voltage of the operational amplifiers used in the differential amplifier. The offset voltage can vary between -1 and 2 μV between different operational amplifiers [13]. This combined with a number of other possible variables such as variations in value of the resistors can have an effect on the gain of the differential amplifier and could induce a voltage offset in the final measurement.

A Monte Carlo simulation iterating over different resistor values and OP-amp offset voltages was done, this showed that the gain is mostly constant over the range of input voltages. The offset voltages were modeled using a voltage source in series with the inverting input of every Op-amp. These sources were varied according to the distribution shown in the datasheet [**?**]. The error due to non-linearity is shown in Figure 4.2. As the maximum error due to non-linearity is smaller than the resolution of the ADC (2.1 μV $<<$ 0.81 mV) it can be ignored.

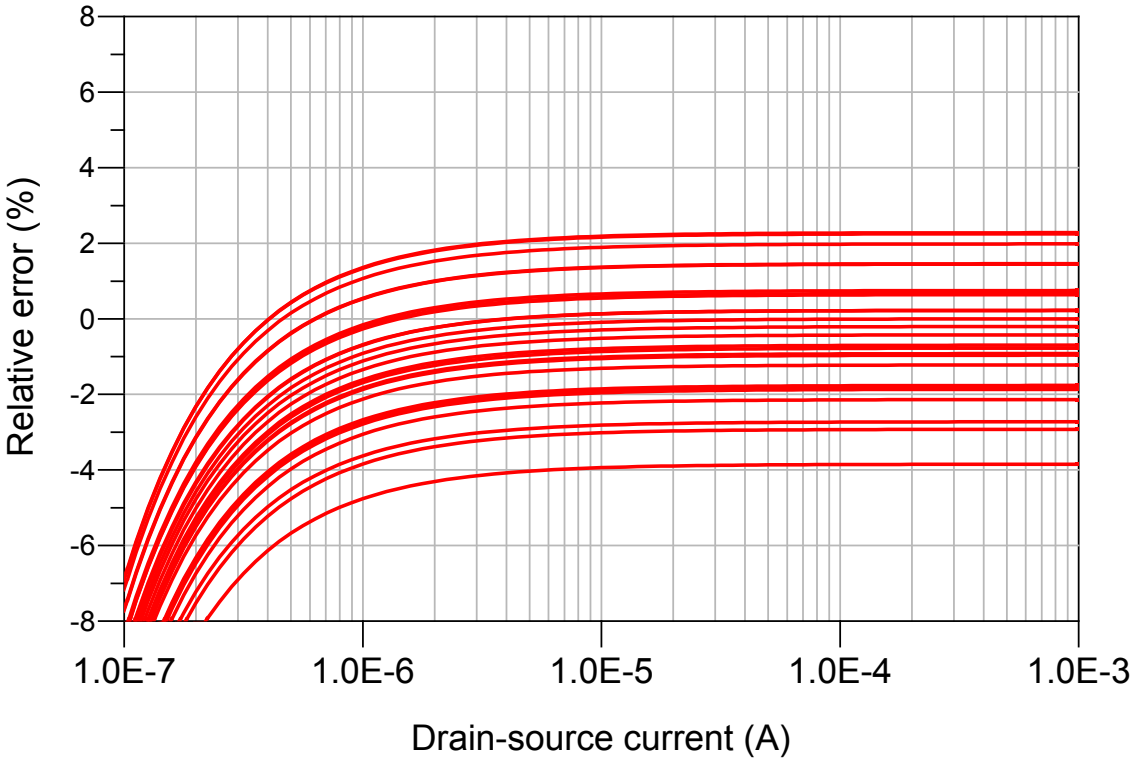


Figure 4.1: Monte-Carlo iterations showing the amplifier error relative to the drain-source current

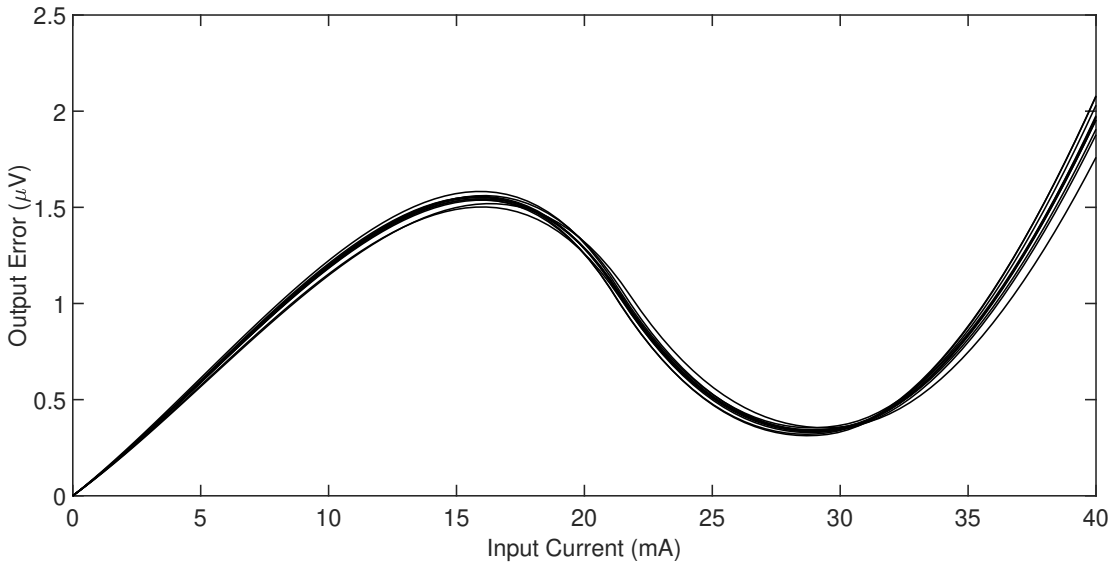


Figure 4.2: Monte-Carlo iterations showing the amplifier error with gain and offset error compensated

4.2 Two point method

Due to the negligible non-linearity described in Section 4.1 it is possible to extract a function that finds the output voltage for any input voltage from only two measurements. The main advantages of this approach is that it is quick and can be performed completely by hardware on the development board, without the use of external sources or measurement equipment.
These measurement points should ideally be located at each end of the input voltage range to obtain the gain with the highest amount of precision. However, if the second measurement point is in a region where clipping occurs, the two point method fails to provide good results, as the clipping is essentially a large non-linearity. Therefore, care must be taken that the measurement points are chosen to be in the linear region of the amplifier gain.

The input voltage is defined as the voltage across the measurement resistor induced by the drain current. The output voltage is the output of the amplifier as measured by the ADC. From these two measurement points a first order polynomial function in the form of Equation 4.1 can be computed. The gain G can be calculated by dividing the difference in output voltage by the difference in input voltage between the two measurements as in Equation 4.2. Then the offset voltage is equal to the difference between the gain multiplied by the input voltage and the measured output voltage as shown in Equation 4.3.

$$V_{out} = G \cdot V_{in} + V_{offset} \tag{4.1}$$

$$G = \frac{V_{out,2} - V_{out,1}}{V_{in,2} - V_{in,1}} \tag{4.2}$$

$$V_{offset} = V_{out,2} - G \cdot V_{in,2} \tag{4.3}$$

By plugging the measured gain and offset voltage into Equation 4.4 the corrected output voltage can be found. The fraction calculates the input voltage which is then multiplied by the theoretical gain of the amplifier to obtain the corrected output voltage.

$$V_{out,corrected} = \frac{V_{out} - V_{offset}}{G} \cdot 101 \tag{4.4}$$

Simulation

To simulate the calibration routine a Monte Carlo simulation of the differential amplifier was done on a DC sweep over the input, taking into account variations of the resistor values and offset voltages of the operational amplifiers. This DC sweep swept the entire range of the ADC to obtain accurate output values for all input values. Both the input voltages and the output voltages were exported to MATLAB [14] where the output voltages were rounded to the resolution of the ADC.
Figure 4.3 and 4.4 show that the calibration reduces the effects of production imperfections by a substantial amount. In both figures the error is defined as the difference between the corrected output voltage and the theoretical output voltage. The jaggedness of the lines is due to the ADC as the output voltage is measured at the output of the ADC meaning that the quantization error induced by the ADC is included in the plot.
The maximum error of the simulated calibration was 1.1 mV at the output of the ADC while the maximum error of the uncalibrated output was 145 mV, however, the simulations used an ideal DAC. In reality the input voltage supplied by the DAC will contain noise and other inaccuracies such as a voltage offset and non-linearity.

By performing two point calibration on multiple temperature points it is possible to create a calibration procedure that also corrects the measurement based on the current measured temperature.

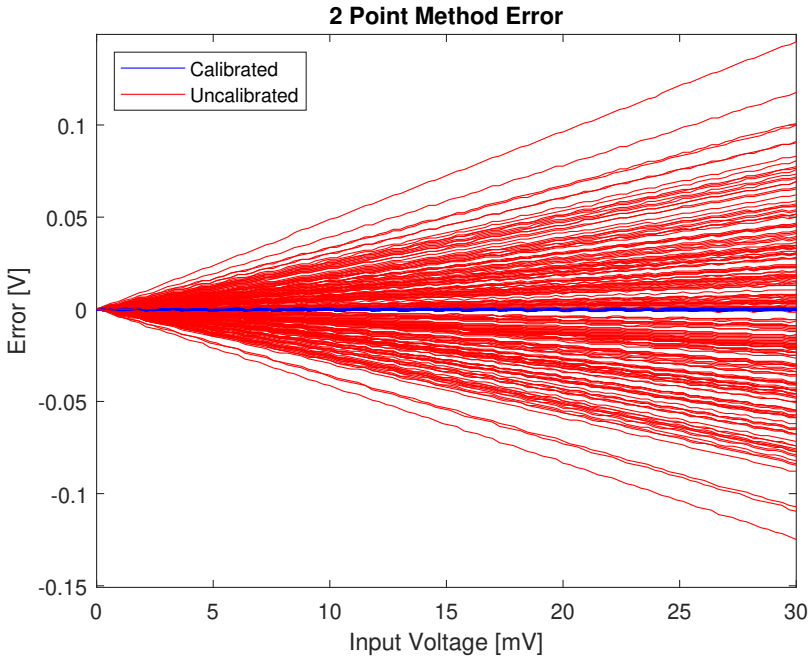


Figure 4.3: Output error after calibration of Monte Carlo simulation

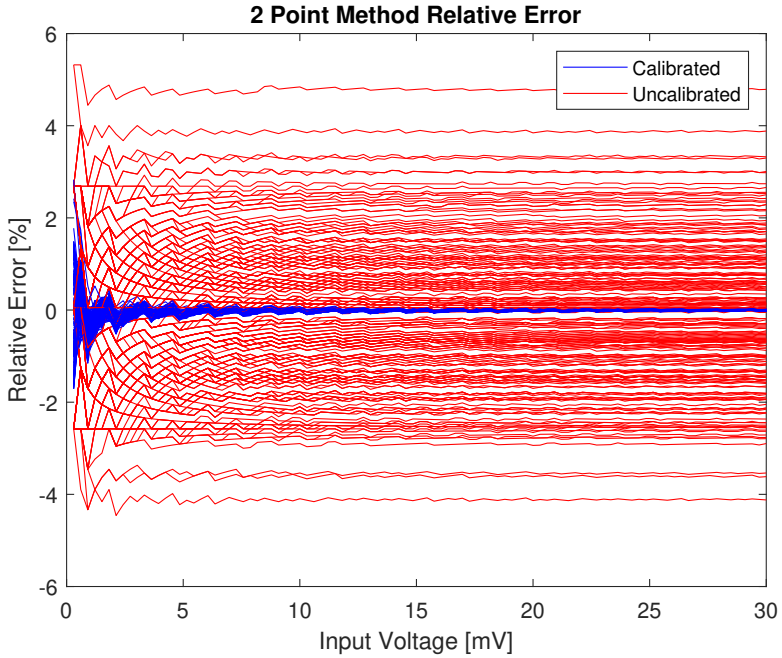


Figure 4.4: Relative output error after calibration of Monte Carlo simulation

4.3 Sweep method

A second method called the ‘sweep’ method is a brute-force approach to calibration where a DAC or similar device sweeps a known input voltage across the input of the differential amplifier. For each voltage step the corresponding output voltage is measured. Then a mapping can be stored in memory, and any future measurement can be looked up to find the correct voltage. This method is great because it removes the need to model the gain, because any non-linearities will be corrected.

The disadvantage is that this method requires more hardware. In order to sweep the input of the amplifier with good resolution and accuracy, an attenuator is necessary with a gain that is inverse of the ideal gain of the amplifier. And likely the DAC and attenuator have to be of high quality in order to provide any advantages over the Two Point method.

The Two Point method is obviously preferred because the non-linearity is not an issue according to the simulations that were done so far. However, if the practical implementation of the PPA shows during testing that non-linearity is an issue, then this method could be a viable alternative to look into for calibration.

Additionally, this method can be used with cheap hardware (i.e. the hardware currently available) just to verify that the gain is indeed of linear form, without unexpected deviations.

Chapter 5

Communication protocol

In order to enable parallel development of the PPA and GUI a communication protocol or layer on top of the Bluetooth/USB layer was necessary. By designing the communication protocol first and making all the definitions, two teams can work independently on implementing the software for the communication between the GUI and the PPA. This will both speed up development and make it easier to add features in the future. Additionally, by knowing how the communication will function, estimations can be made on the power consumption required for the communication. The approach and structure taken in this chapter are based upon Formal Methods in Communication Protocol Design [16].

5.1 Requirements

The communication protocol on top of the Bluetooth/USB layer must provide service between the GUI and PPA. With this service the GUI should be able to do the following:

- Initiate a calibration routine
- Configure measurement parameters
- Initiate a measurement routine
- Remotely turn off the PPA
- Request status, ID and measurement data

And the PPA must be able to do the following:

- Acknowledge the commands from the GUI
- Acknowledge whether a requested state transition is possible
- Return stored measurement data

5.2 Implementation

The requirements in the previous section imply that the GUI acts as a master and the PPA as slave. This in itself is already an implementation or design choice, which is justified by the possibility that in the future the GUI might connect to multiple PPA's, in which case the GUI would benefit from being the master.

Because the PPA works with measurement and calibration routines that have relatively long intervals between measurements, it was decided to have a proper state machine on the PPA. This is because when the GUI requests a measurement, the time passed means that the communication could time-out before a response can be sent. Based on this state machine the communication is defined. The GUI will work more

on an interrupt basis, because the user will want to interface with it at any time. Additionally, the GUI will not have to perform any specific actions based on the PPA state, which is why it will only keep track of the state, but a state machine will not be implemented for the GUI.

Table B.1 shows the commands that have to be implemented in order to satisfy the requirements from the previous section. It also shows the responses that are expected from the PPA. For simplicity in programming most values and commands are grouped in bytes.

Improvements

If the power consumption of the communication proves to be a problem, it could be possible to compress the data in order to decrease the amount of bytes that are sent. Currently all the commands and values are separated in bytes, often with room to spare. By putting multiple fields into one byte, less bytes will have to be sent altogether. The only disadvantage is in the increased (code) complexity, requiring the use of bitmasking in order to extract data from the received packets.

5.3 Verification

With a debugger connected to the PPA it will be possible to verify that the PPA acts according to the commands that are sent from the GUI. The only difficult part is in the scenarios that rarely happen, such as verifying that the PPA responds correctly when the GUI sends a command while a measurement is in progress. While rare due to the relatively short measurement durations, it is important to know that the PPA can handle such an event without losing data or losing track of the current state. A suggested method for testing this is to temporarily increase the rate at which the PPA measures and the rate at which the GUI communicates. Then by smart usage of breakpoints, the behaviour can be verified.

Chapter 6

Graphical user interface

In collaboration with the client a set of requirements was made for a Graphical User Interface (GUI) that could be used to remotely control and access the portable parameter analyzer from a computer or laptop. These requirements can be found in section 2.4. The most important requirement is that the GUI should display and log the evolution of the drain-source current as time progresses.

To design the GUI there are many platforms available that are all sufficient. In the end the choice was between a framework called Qt and MATLAB. Table 6.1 shows some considerations for choosing the framework. Because there is a possibility that the GUI development will be continued by a party outside of Electrical Engineering, where MATLAB might not be immediately available, Qt was chosen for its open-source aspect and free design tools.

Table 6.1: Some features for the GUI framework selection

	MATLAB	Qt
Tools for GUI design	Available, easy to use	Available, more design options available and easy to use
Maintainability	MATLAB dependencies can be vague which could make it hard to maintain	Self-contained, should be easy to upgrade and maintain
Modularity	Not focused on object-oriented programming, could become hard to maintain	Object-oriented (uses C++) so it can be made in a modular fashion
Price/Licensing	Requires licence	Open-source licences available with free tools

6.1 Toplevel overview

Figure 6.1 shows a suggested top-level design for the GUI. At this point in time it has not been implemented, but this overview could provide a good guideline for the development of the software. This section will explain each element in more detail.

StateMachine

While this element does not strictly have to be a state machine, it is the entry-point of the application and should keep the state in order to prevent the user from accidentally removing data or interfering with the PPA. This element is kind of the brain of the GUI.

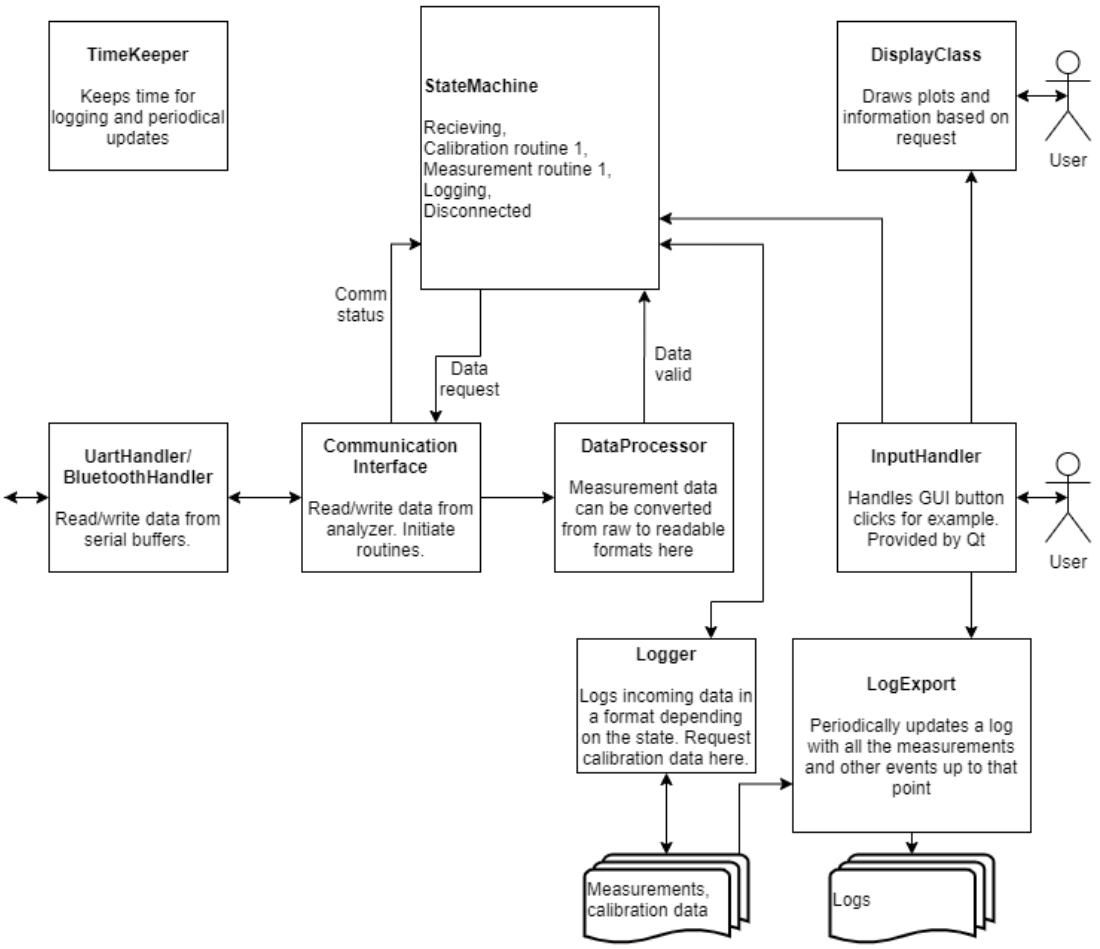


Figure 6.1: An overview of the top level design of the GUI

DisplayClass

When the user requests data to be displayed or when the application starts, the GUI should be updated to display the data in a readable format. Qt provides the interface to the GUI elements, therefore the DisplayClass is more of an implicit element in the overview.

InputHandler

Handles the inputs of the user such that when buttons are pressed the appropriate functions are executed, such as exporting data, displaying data and requesting data from the PPA. This element is also a part of Qt.

TimeKeeper

Keeps the time in order to add timestamps to logged items. Provides functions to translate the timestamp format from the PPA messages into a readable format. Timekeeper functions can be called from anywhere where necessary.

Communication Interface

Provides an interface for communication protocols to get or send data. Additionally this element will detect if an incoming packet is a data packet from the PPA, and sends it to the DataProcessor to be processed.

DataProcessor

Converts and sorts data from the PPA into readable formats. The functionality depends on specifications from Chapter 5.

UartHandler/BluetoothHandler

Provides the interface to communication protocols over the air or wire. The implementation depends a lot on the protocol that is used.

Storage

Comprised of the classes Logger and LogExport, these elements provide functions for logging (storing data and events with timestamps) and retrieving the log files such that the user can do further data processing on the measurements.

6.2 Implementation

A mock-up of the GUI design was made (as seen in Figure 6.2 using Qt, which means that the application can launch, but is not functional yet. Only the Bluetooth was implemented, such that a PPA can be connected to the GUI. (todo; implement communication with ppa)

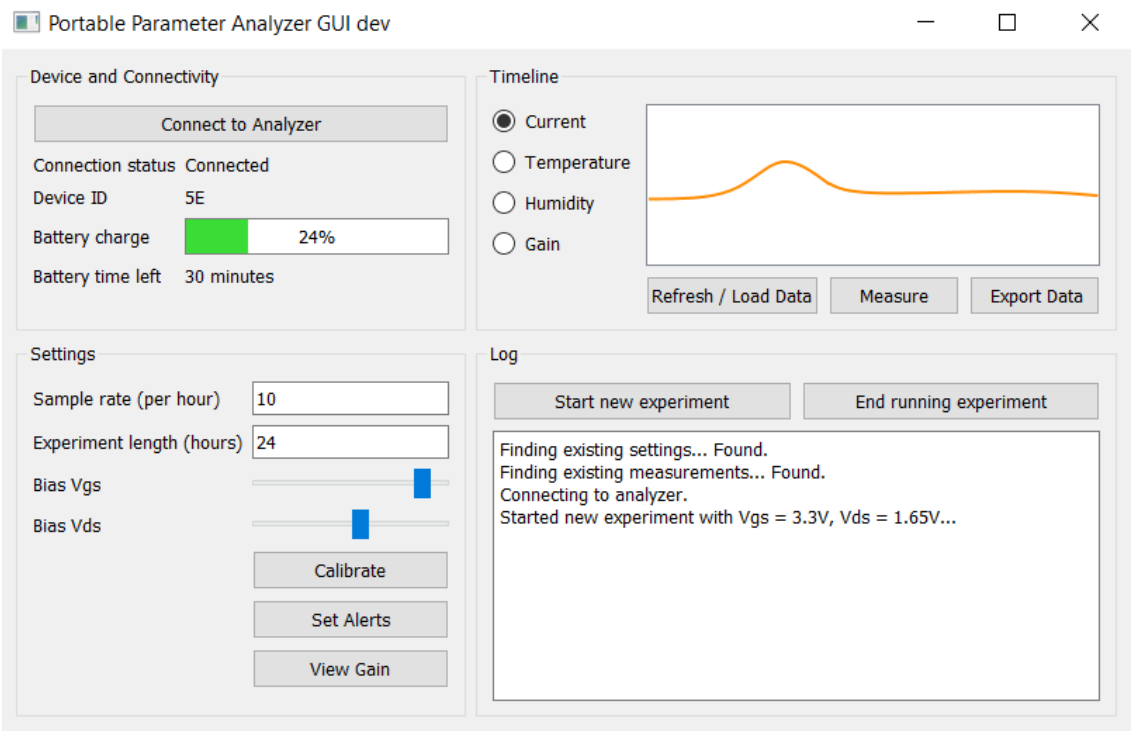


Figure 6.2: Mock-up of the GUI design

Chapter 7

Calibration Results

The following results were achieved using the setup depicted in Figure 7.1. It is expected that this setup will have more noise than the final system due to the wired connection with the computer and the long wires used to connect different components.

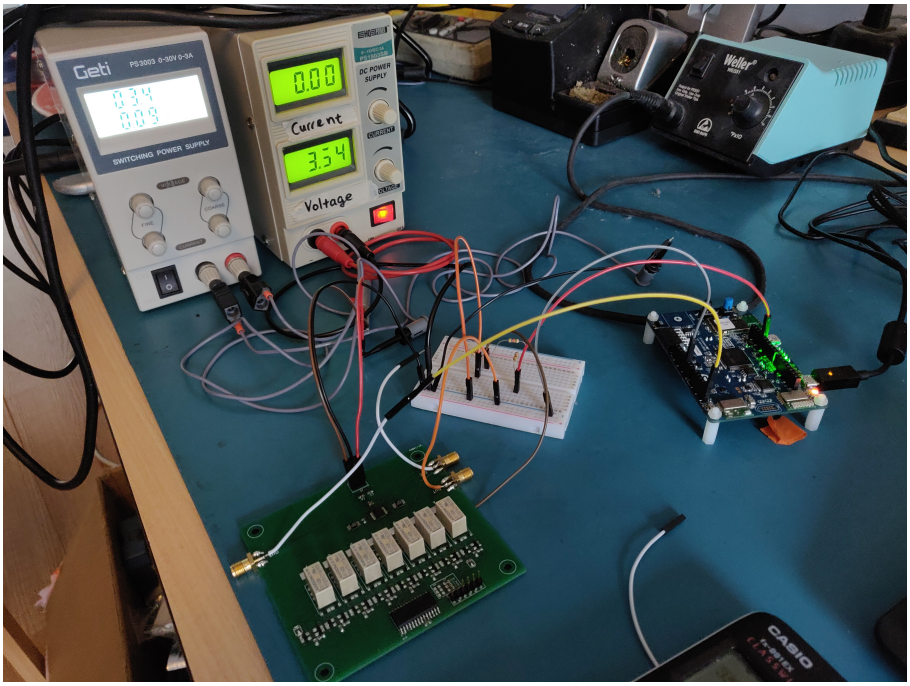


Figure 7.1: Measurement setup for the two-point method

7.1 DAC/ADC accuracy test

To determine the magnitude of the inaccuracies injected by the DAC and ADC on the development board, a test was done by sweeping the full range of the DAC over the full range of the ADC. This is possible as both the DAC and ADC use the same have the same reference voltage and resolution. The results are shown in Figure 7.2 where the error is defined as the difference between the voltage at the output of the ADC and the ideal output of the ADC.

The peaks at the beginning and end of the measuring range are from clipping, which was later found to be caused by the DAC. While this slightly reduces the measuring range and therefore the resolution, it is not

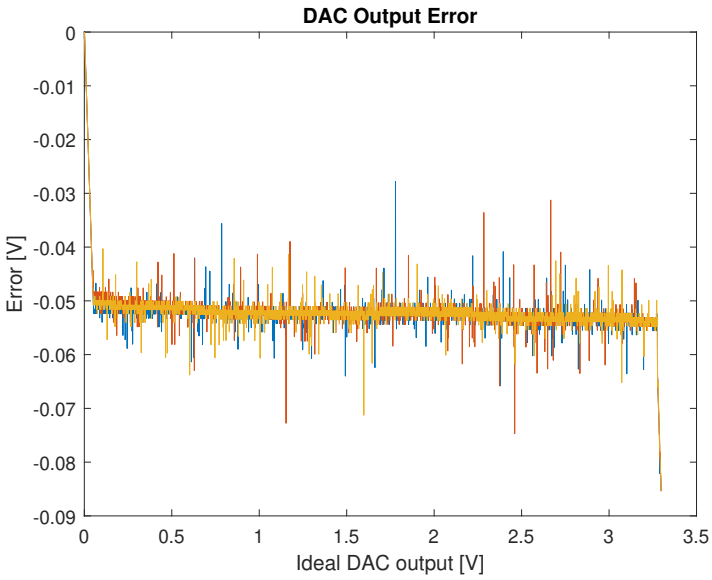


Figure 7.2: The output error as measured with the ADC

expected to be significant, as the operating point will not be at the endpoints of the range. The maximum peak-to-peak noise is around 40 mV, corresponding to roughly 20 uA of the drain-source current. In this test the non-linearity is in the order of 2.4 mV.

7.2 Amplifier characteristics

To measure the amplifier response over the full input-range, the output of the DAC is attenuated. In this case a voltage divider with an attenuation factor of 93.98 was used. If the gain of the amplifier were near the theoretical gain 101, the output voltage of the amplifier would go above 3.3 V at some point during the measurement. However, the gain was found to be 53.49, with an output offset of -54.7 mV. This means that the whole input range of the amplifier was able to be measured. It is currently unclear why the gain is not 101, as the gain resistor was measured to be 1 kΩ.

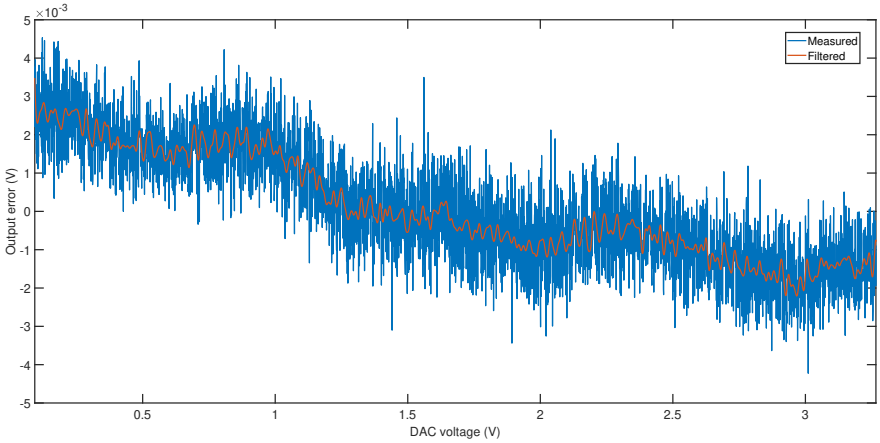


Figure 7.3: Uncalibrated error measurement of the system with ADC, amplifier and DAC. The gain and offset error are approximated and removed

7.3 Two-point method

Figures 7.4 and 7.5 show that the two-point method can do an error correction that is significant, with a maximum error of 1.4 mV after calibration. However, the error before calibration is huge due to an unknown defect that caused the gain of the differential amplifier to be around 53 instead of the designed gain of 101. With the two-point method and the measurement setup shown in Figure 7.1 the gain was found to be 53.37 V/V and the offset -51.8 mV.

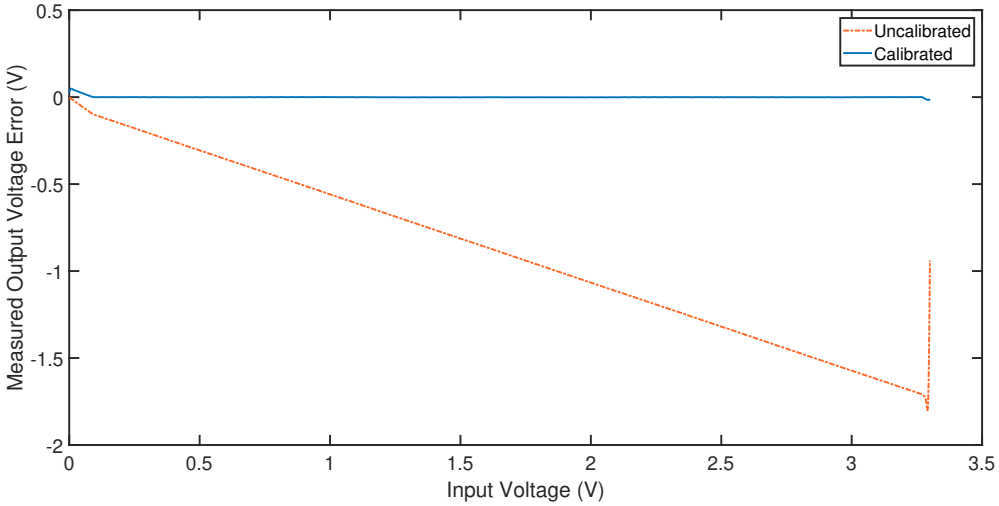


Figure 7.4: Error in the output measurement before and after calibration.

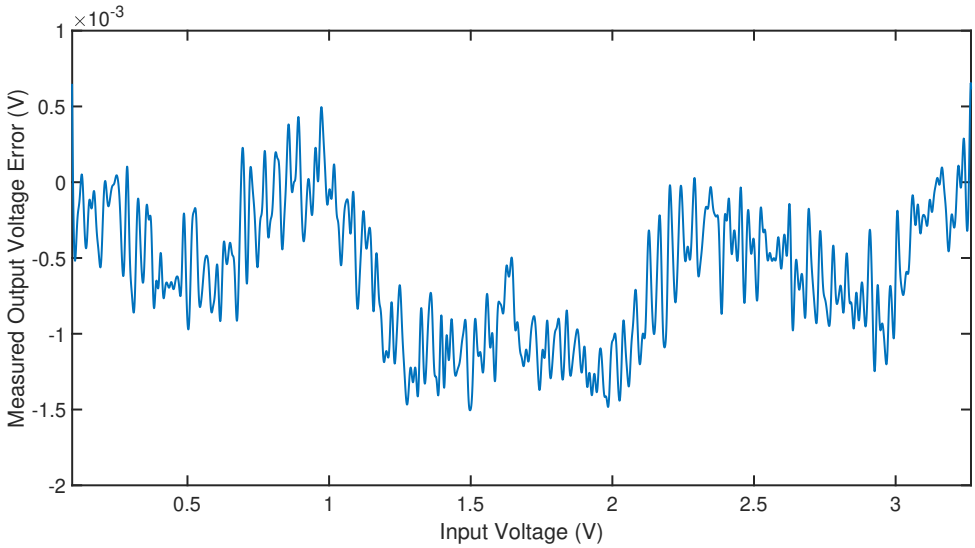


Figure 7.5: Error in the output measurement after calibration.

Chapter 8

Discussion

Few results were obtained due to the lack of a physical demonstrator. Initially, a simulation of the proposed calibration method was done. This resulted in a maximum error of 1.1 mV at the output of the ADC (out of the 3.3 V range), while the maximum error of the uncalibrated output was 145 mV. However, these simulations were done with a perfect DAC to provide the input voltages. Later, measurements were made using the DAC on the development board to generate reference voltages for the two-point calibration method. The measurements were noisier than expected, but not to the extent that they were useless. The calibration method seems like it can improve the measurement accuracy as it had a maximum error of 1.5 mV, but this was hard to prove without a good reference measurement. It is difficult to make a good reference setup because there are many elements that need to be known: Errors inserted by the DAC and ADC or reference voltages and the attenuation.

Chapter 9

Conclusion

The first step was to carry out an analysis of the given differential amplifier. While it was not possible to perform testing in the lab, the simulation results looked promising and a datasheet was made for the amplifier.

Then work proceeded on attempting to estimate the amount of error that the amplifier and other elements in the system would contribute. It is impossible to account for all the variations, but with Monte-Carlo simulations that took into account OP-Amp offset voltages and resistor variations it was safe to say that the two-point method for calibration would be sufficient. Initial measurements that were done later confirmed this.

As the whole system could not be tested in a lab environment, a second calibration method was proposed. The sweep method would be able to account for non-linearities (not just gain error and offset) in the gain, at the cost of extra hardware. However, the method could also be used to simply verify that the amplifier is working correctly, as it sweeps the full input range of the amplifier.

For the user interface a communication specification was developed that would allow the GUI and PPA software to be developed independently, while also acting as a form of documentation. From the specification it was also possible for the other group to estimate how much power would be needed for communication. There is still room to compress data, but that was deemed not necessary for now.

With the communication specification and PPA states in mind it was possible to start designing a GUI that could remotely operate the PPA. First a toplevel diagram was made for the software back-end, and then the GUI design was mocked up using Qt.

9.1 Recommendations and future work

In order to verify that the two-point calibration method works, it is recommended to do reference measurements with lab equipment. Additionally, the usage of the DAC should be reconsidered because the DAC seemed to add a lot of error in the system. Other options would be reference voltage ICs or a high-quality DAC. Finally, measurements or simulations should be done to measure the effects of temperature fluctuations on the current measurements.

To increase the ease of use of the portable parameter analyser a different calibration could be implemented in addition to the two point method, that would relate the measured current to electrochemical quantities in the sensing area. This calibration would convert the change in drain current to a change in pH, for example.

Because the sensor only measures changes in the surface charge on the sensing region, it is impossible to measure the pH without information on the volume and possible presence of charge carriers other than hydrogen ions. These other charge carriers affect the measured surface charge but do not affect the pH of the solution. This means that the pH of the solution to be measured must be known in advance.

Another complication of this approach is the fact that the surface charge density is not always directly proportional to the pH of the solution [17]. Possible causes for inaccuracy when using this calibration are changes in pH due to temperature [18] and changes in pH due to changes in hydrogen ion activity as hydrogen activity does affect the pH [19] but not the measured drain current.

Bibliography

[1] Qt. (5.15), Qt Project, The Qt Company. Available: https://www.qt.io/

[2] E. Albayrak and M. Mens, "Portable Parameter Analyser for Organs-on-Chip: Power Budget Analysis", Bachelor's thesis, Delft University of Technology, June 2020.

[3] M. van der Maas, K. dos Reis Vezo, "Portable Parameter Analyser for Organs-on-Chip: Charge Sensor Model", Bachelor's thesis, Delft University of Technology, June 2020.

[4] P. Germain, L. Chiapperino, G. Testa, "The European politics of animal experimentation: From Victorian Britain to 'Stop Vivisection'", Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences, vol. 64, p. 75-87, 2017, doi: https://doi.org/10.1016/j.shpsc.2017.06.004

[5] A. Knight, "Systematic Reviews of Animal Experiments Demonstrate Poor Contributions Toward Human Healthcare", Reviews on Recent Clinical Trials, vol. 3, p. 89-96, 2008, doi: https://doi.org/10.2174/157488708784223844

[6] Z. Wang, R. Samanipour, K-i. Koo, K. Kim, "Organ-on-a-Chip Platforms for Drug Delivery and Cell Characterization: A Review", Sensors and Materials, vol. 27, no. 6, p. 487-506, 2015,

[7] B. Zhang, M. Radisic, "Organ-on-a-chip devices advance to market", Lab on a Chip, vol. 17, no. 14, p. 2395-2420, June 2017, doi: 10.1039/C6LC01554A

[8] M. Mastrangeli, S. Millet, T. ORCHID partners, and J. van den Eijnden-van Raaij, "Organ-on-chip in development: Towards a roadmap for organs-on-chip", ALTEX, 2019.

[9] S. Park, D. Jeon, Y. Moon, I. Park and G. Kim, "Web-drive based source measure unit for automated evaluations of ionic liquid-gated MoS2 transistors", Review of Scientific Instruments, vol. 90, no. 12, p. 124708, 2019. Available: 10.1063/1.5111724.

[10] R. F. Yazicioglu, C. Hoof, R. Puers, "Biopotential Readout Circuits for Portable Acquisition Systems", Springer Science & Business Media, 16 okt. 2008

[11] L. Callegaro, V. D'Elia and B. Trinchera, "A Current Source for Picoammeter Calibration," in IEEE Transactions on Instrumentation and Measurement, vol. 56, no. 4, pp. 1198-1201, Aug. 2007, doi: 10.1109/TIM.2007.900128.

[12] PathWave Advanced Design System, (2015). Keysight Technologies

[13] Analog Devices, "55 V, EMI Enhanced, Zero Drift, Ultralow Noise, Rail-to-Rail Output Operational Amplifiers," ADA4522-1/ADA4522-2/ADA4522-4, May 2015

[14] MATLAB R2018b, (2018). MathWorks

[15] H. Aydogmus et al., in preparation (2020)

[16] G. Bochmann and C. Sunshine, "Formal Methods in Communication Protocol Design," in IEEE Transactions on Communications, vol. 28, no. 4, pp. 624-631, April 1980, doi: 10.1109/TCOM.1980.1094685.

[17] P.-W. Chen, C.-Y. Tseng, F. Shi, B. Bi, & Y.-H. Lo, "Measuring Electric Charge and Molecular Coverage on Electrode Surface from Transient Induced Molecular Electronic Signal (TIMES)." Scientific Reports, 9(1). 2019, doi:10.1038/s41598-019-52588-6.

[18] A.K. Covington, "Definition of pH Scales, Standard Reference Values, Measurement of pH and Related Terminology" in Pure & Applied Chemistry, vol. 57, no. 3, pp. 531-542, 1985.

[19] I. Feldman, "Use and Abuse of pH Measurements" in Analytical Chemistry, vol. 28, no. 12, pp. 1859-1866, December 1956, doi: 10.1021/ac60120a014.

Appendices

Appendix A

Amplifier Simulations

For measuring very small currents an ADA4522-based instrument amplifier design was given. This datasheet contains simulations and specifications of the design, done in order to get a better understanding of the design and its performance. These specifications will become important when the calibration methods are designed.

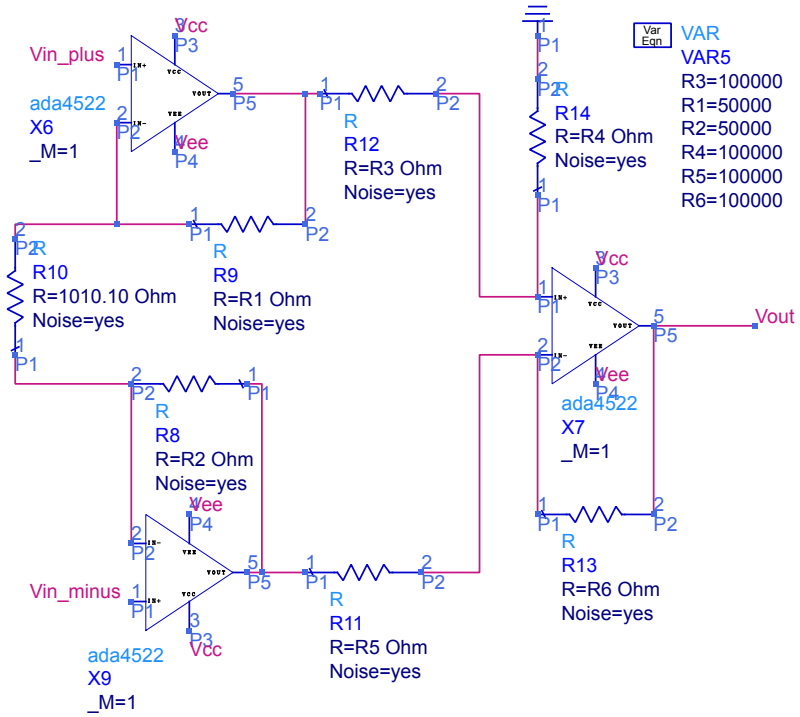


Figure A.1: The schematic design of the ADA4522-based amplifier

The simulations are run with the following general conditions applied.

Condition	Value
Vcc	15 V
Vee	-15 V
Temperature	37 °C
Output	Floating

The following table summarizes the specifications derived from the figures and calculations in the remainder of this document.

Parameter	Conditions	Value	Units
INPUT			
Offset Voltage, RTI	$V_{cm} = 0V$	0.10573	µV
Impedance, Differential	f = 10 Hz	$341.2 \parallel 219.9$	$G\Omega \parallel pF$
Impedance, Common-mode	f = 10 Hz	$235.6 \parallel 528.4$	$M\Omega \parallel pF$
Common-Mode Rejection	$V_{cm} = \pm 1V, \Delta R_s = 0\Omega$		
	G = 100	151.6	dB
NOISE VOLTAGE, RTI	$G = 100, R_s = 0\Omega$		
f = 10 Hz		12	$nV/\sqrt{Hz}$
f = 10 kHz		12	$nV/\sqrt{Hz}$
f_B = 1 to 100 Hz		1.2	µV
GAIN			
Gain Equation		$1 + (100\ k\Omega/R_g)$	
Gain error		TBD	
Gain vs Temperature		TBD	
Nonlinearity		TBD	
FREQUENCY RESPONSE			
Bandwidth, -3 dB	G = 100	28	kHz
Rise Time	$V_o = 0.1V$	10.2	µs
Settling Time 0.01%		46	µs

Table A.1: General specifications of the given differential amplifier

A.1 Frequency Response

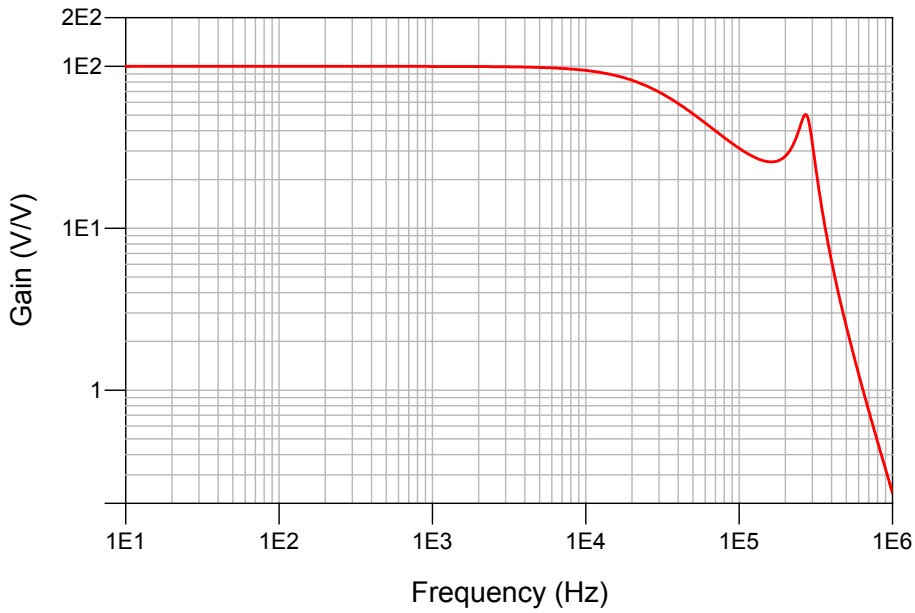


Figure A.2: Differential gain versus frequency of the differential amplifier

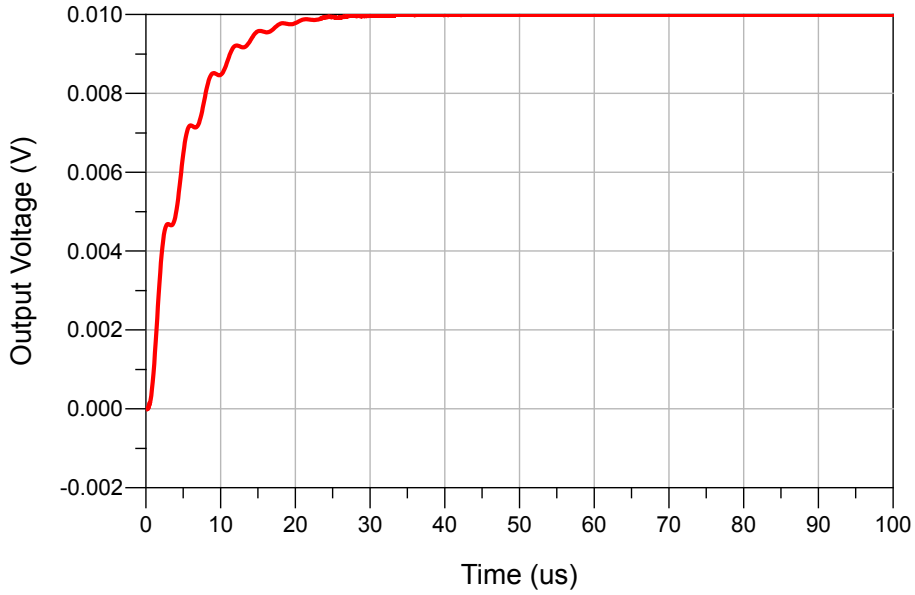


Figure A.3: Small signal step response. No output load.

A.2 Gain

The gain is as seen in Figure A.2. The figure below shows the output error when the ideal gain is 100. From this, the gain error and non-linearity can be determined. However, at this stage the component tolerances are not yet taken into account so the gain error is insignificant. The gain error and non-linearity will be evaluated when the component tolerances are known.

It appears there are some discontinuities in the figure, for which the cause is not known. Fortunately, these appear outside of the expected measurement range.

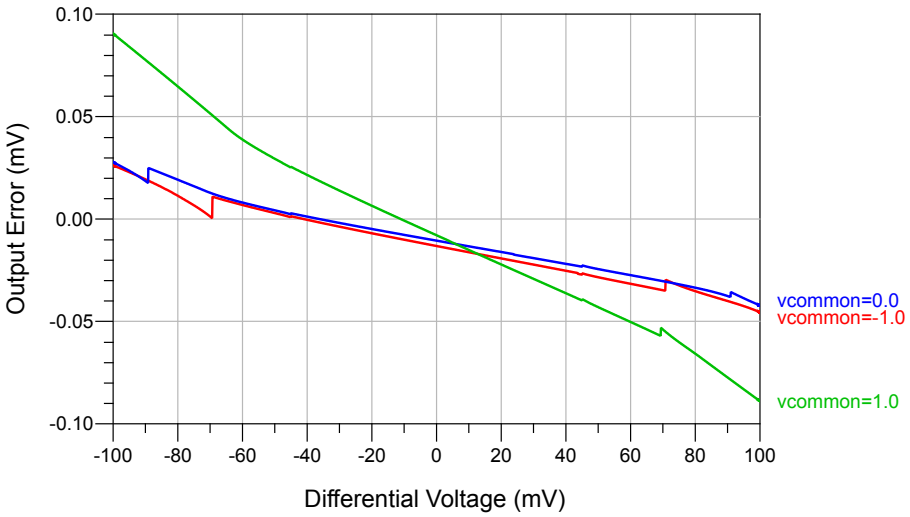


Figure A.4: Output voltage minus the ideal output voltage of 100 times the input voltage.

A.3 Common-Mode Rejection Ratio

The common-mode rejection ratio (CMRR) is approximated by assuming a differential gain A_d of 100 over all frequencies. From Figure A.2 it can be seen that this will hold for frequencies lower than 10 kHz.

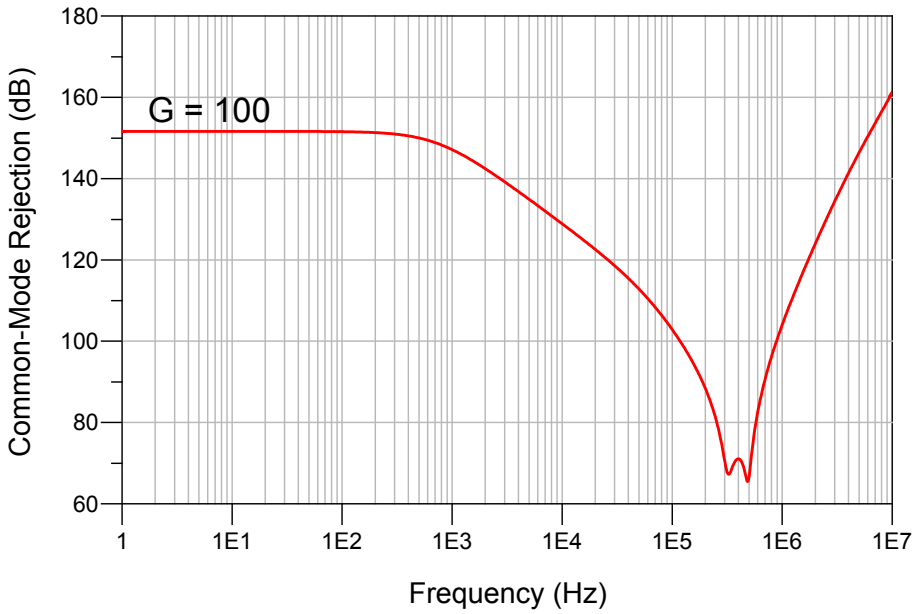


Figure A.5: Common-mode rejection ratio versus frequency. Does not take into account the gain over frequency. Perfectly balanced sources.

A.4 Offset Voltage

The offset voltage was approximated by taking the differential input voltage for which the output voltage was the closest to $0V$. This was done for various common-mode voltages.

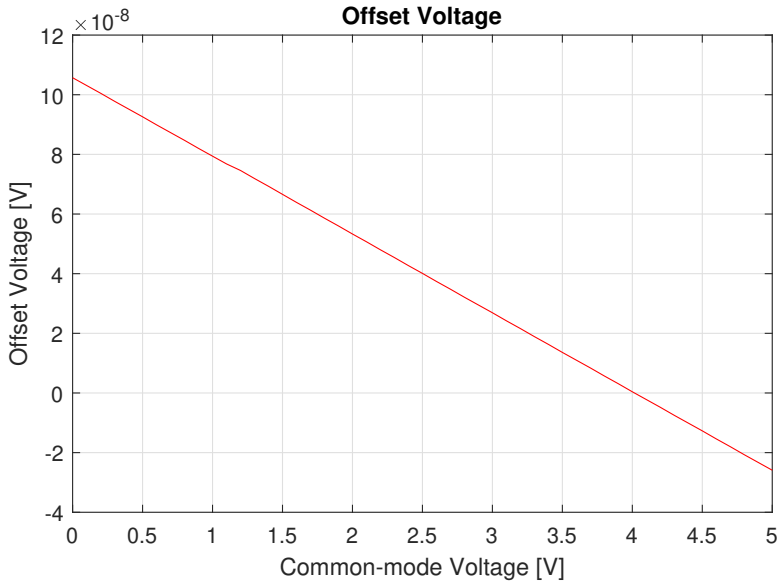


Figure A.6: Offset Voltage set out against the Common-mode Voltage

A.5 Input Impedance

The differential input impedance was measured by creating a voltage difference between the inputs of the amplifier. From this known voltage and the measured current that flowed through the inputs, the impedance was calculated.
The common mode input impedance was measured by putting an equal voltage on both of the inputs of the amplifier with respect to the ground. This created a current that flowed into the inputs, from this current and the known input voltage the common mode input impedance was found.

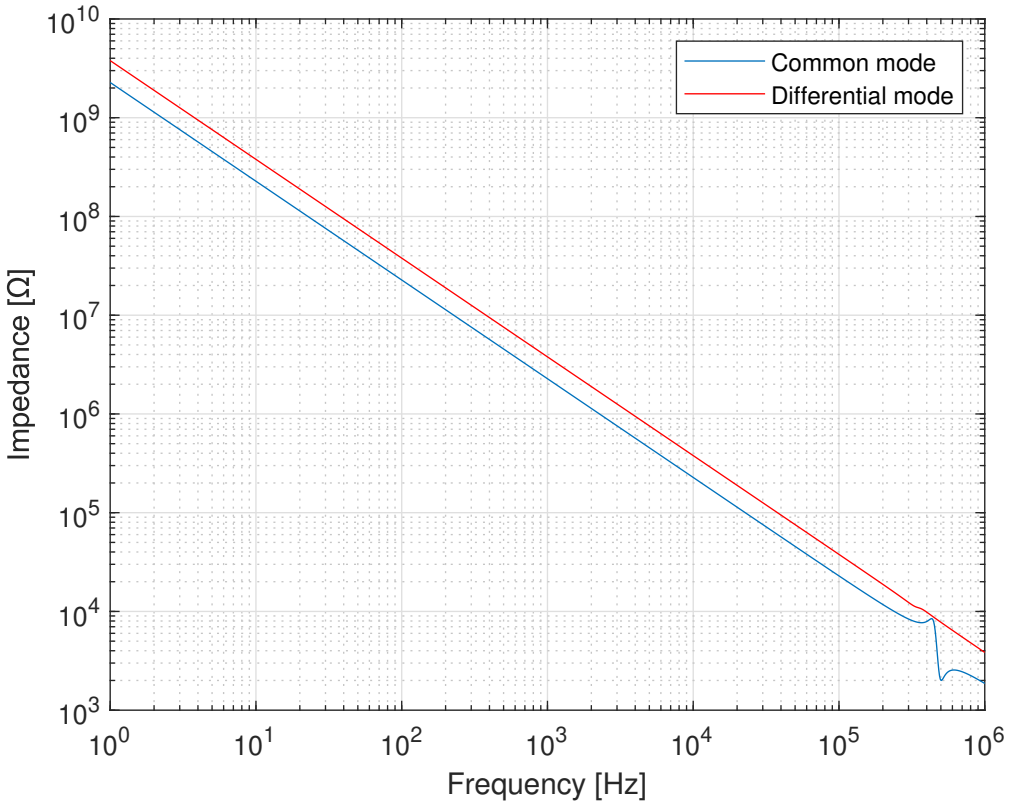


Figure A.7: The magnitude of the common mode and differential input impedances plotted against the frequency.

A.6 Noise Response

The noise is ‘measured’ at the output of the amplifier while the inputs are shorted to ground. The measured noise is divided by the gain as seen in Figure A.2 to get the input-referred noise voltage as seen in Figure A.8. It does not take into account noise from the voltage supplies.

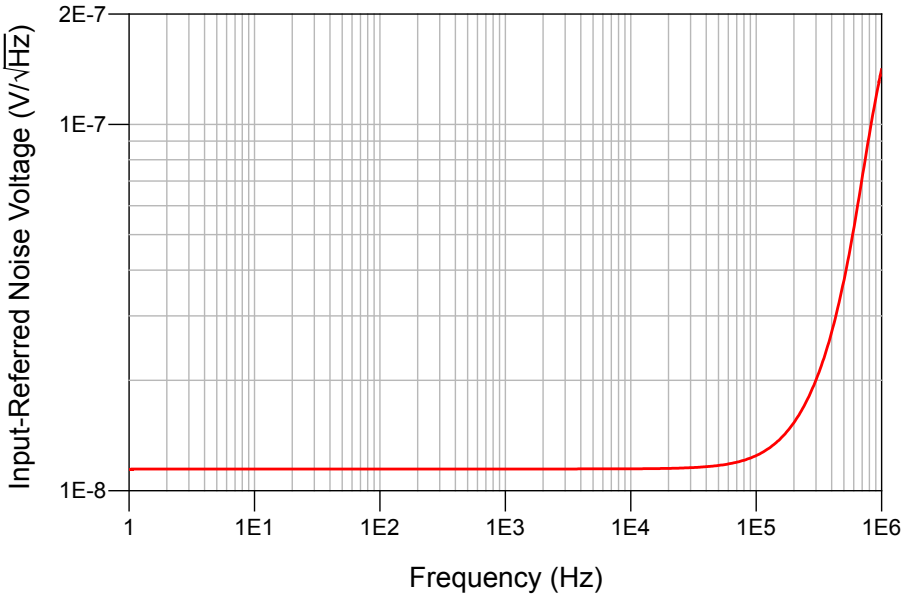


Figure A.8: Input-referred noise

Appendix B

Communication table

Requirement ID	Description	Send command	Send data	Return command	Return data
CMD_01	GUI requests PPA to go to state x (calibration or measurement method), PPA will return data according to the method descriptions in y	STATE_MOVE	state	(N)ACK_STATE_REQUEST	requestedState, errorCode
CMD_02	GUI forces PPA to go to state x	STATE_FORCE	state	ACK_STATE_REQUEST	requestedState
CMD_03	GUI requests measurement backlog	REQUEST_MEAS_ALL	-	INFO_MEASUREMENT	measurements (format required)
CMD_04	GUI requests ID/status, PPA returns ID/status	REQUEST_INFO	-	INFO_CURRENT_STATE	state, id
CMD_05	Remote shutdown (with the option to finish the current running job/measurement)	SHUTDOWN	finishJobFirst	ACK_SHUTDOWN	finishJobFirst
CMD_06	Request a single measurement, for debugging purposes	REQUEST_MEAS	-	INFO_MEASUREMENT	measurements (format required)
CMD_07	Set resistances, for debugging purposes	SET_RESISTOR	resistor, on/off	INFO_RESISTOR_SELECTION	resistor(s), on/off
CMD_08	Set bias, for debugging purposes	SET_BIAS	biasType, value	INFO_BIAS	biasVoltages

Figure B.1: Description of the commands that are sent from the GUI to the PPA. Names written in all capital letters are C++ defines that are shared between the GUI and PPA. CamelCase names are variables such as measurements and booleans.