



Charlotte Fuchs, BSc

Development and Evaluation of a Microelectronic Application for Biochemical Measurements

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingeneurin

Master's degree program: Biomedical Engineering

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Christian Baumgartner

Institute of Health Care Engineering with European Testing Center of
Medical Devices

Graz, February 2021

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Acknowledgement

I would like to thank Infineon Technologies Austria AG and the Institute of Health Care Engineering at TU Graz for their support, which made this master's thesis possible. I am also appreciative of the cooperation with the Austrian Institute of Technology in Vienna.

The completion of my thesis would not have been possible without the support and nurturing of my supervisors at Infineon Inge Siegl, Christian Zajc and Markus Haberler. Their unwavering support has accompanied me throughout the work. Thanks also to the other colleagues in the department (CRE) under the direction of Gerald Holweg, who have actively supported me.

Entwicklung und Evaluierung einer mikroelektronischen Applikation für biochemische Messverfahren

Kurzfassung

Ein biochemischer Sensor wandelt ein chemisches Signal in ein elektrisches Signal um. Hierfür stehen verschiedene Messverfahren zur Verfügung, die je nach Anwendung gewählt werden. In dieser Arbeit wird die elektrochemische Impedanzspektroskopie implementiert. Zur Durchführung der Messungen wird ein System-on-Chip, welches von der Infineon Technologies Austria AG entwickelt wurde, in Kombination mit einem biochemischen Sensor vom Austrian Institute of Technology in Wien verwendet. Das implementierte Messverfahren wird evaluiert und das System mit entsprechenden Ersatzschaltungen getestet, bevor die abschließenden Messungen mit dem Sensor durchgeführt werden. Die mit diesem low-power System erzielten Ergebnisse sind vergleichbar mit denen gängiger Potentiostaten. Das System könnte somit für Messungen spezifischer Biomoleküle im Speichel verwendet werden, um den Gesundheitszustand und das Stressniveau des Menschen zu überwachen.

Schlüsselwörter - elektrochemische Messung, elektrochemische Impedanzspektroskopie, biochemischer Sensor, System-on-Chip, Randles-Schaltung

Development and Evaluation of a Microelectronic Application for Biochemical Measurements

Abstract

A biochemical sensor converts a chemical signal into an electrical signal. Various measurement methods are available, which are selected according to the respective application. In this thesis the electrochemical impedance spectroscopy is implemented to drive an electrochemical measurement. A system-on-chip designed by Infineon Technologies Austria AG in combination with a biochemical sensor developed at the Austrian Institute of Technology in Vienna is used for this. The implemented measurement method is evaluated and the system is tested with equivalent circuits. Conclusively, the measurements are performed with the sensor. The results obtained with this low-power system are comparable to the results of common potentiostats. So, the system could be used to measure specific biomolecules in saliva to monitor the human health status and stress level.

Keywords - electrochemical measurement, electrochemical impedance spectroscopy, biochemical sensor, system-on-chip, Randles circuit

Contents

1	Introduction and goal of the thesis	1
2	Measurement method and procedure	5
2.1	Fundamentals of measurement method	5
2.1.1	Electrochemical impedance spectroscopy (EIS)	7
2.1.1.1	Equivalent model	8
2.1.1.2	Impedance determination and representation	8
2.1.2	Sensor specific measurement specification	10
2.2	Measurement procedure	11
2.2.1	Draft of measuring procedure	11
2.2.2	PCB design	14
2.2.3	Planned evaluation steps	15
2.2.4	Sensor measuring routine	16
3	Implementation, evaluation and testing	19
3.1	Test-chip overview	19
3.1.1	Sensor interface structure	20
3.2	Simulation environment	22
3.2.1	Signal generation	23
3.3	Evaluation of the sensor interface	26
3.3.1	Test configuration	27
3.4	Improvements	28
3.4.1	Handling high frequencies	28
3.4.2	ADC filter delay	29
3.4.2.1	Calibration function	30

3.4.3	Additional moving average filter	31
3.4.4	Calibration of shunt resistors	35
3.5	Implementation of the measurement procedure	36
3.5.1	Initialization	36
3.5.2	Measurement	37
3.5.3	Data analysis	40
3.6	Evaluation of the implemented parts	41
3.6.1	Output signal	41
3.6.2	Current measurement	42
3.6.3	Signal processing	44
3.7	Test results	45
3.7.1	Different resistive and capacitive loads	45
3.7.2	Equivalent circuit (Randles circuit)	49
3.8	Sensor measurements	50
3.8.1	Setup	50
3.8.2	Comparative measurements	51
4	Discussion	55
4.1	Implemented method	55
4.2	Adaptations to improve measurement	56
4.2.1	Additional implementation tools	57
4.2.2	Calibration function	58
4.2.2.1	ADC filter structure	58
4.2.3	Moving average filter	59
4.3	Results and their accuracy	60
4.3.1	Noise overlay	60
4.3.2	Torn-out points	61
4.3.3	Low frequency region	63
4.3.4	Comparative measurements	65
5	Conclusion	67
6	Future work	69
6.1	PCB design	71

List of Figures

2.1	Two-electrode system for electrochemical measurements	6
2.2	Three-electrode system with potentiostat for electrochemical measurements	6
2.3	Simplified presentation of an electrochemical biosensor	7
2.4	Schematic of the Randles circuit	8
2.5	Nyquist plot examples	9
2.6	Visualization of the parameter of the Randles circuit on an impedance plot	10
2.7	Electrode layout on glass	10
2.8	Initial block diagram of measurement procedure	13
2.9	PCB layout of the electrode configuration	14
2.10	Electrodes on PCB connected to the test-chip board	15
2.11	Measuring routine with sensors	17
3.1	Simplified block diagram of the test-chip	20
3.2	Schematic of a resistor ladder	21
3.3	Schematic of two resistor ladders in series	21
3.4	Schematic of the DAC register	21
3.5	Reduced representation of the DAC structure in the simulation	23
3.6	Reduced representation of the DAC structure in the implementation	24
3.7	Sine wave generation	25
3.8	Representation of the simulation environment	25
3.9	DAC output verification	26
3.10	Test setup	27

3.11	Part of the assembler implementation	29
3.12	ADC filter delay according to moving average filter	30
3.13	Calibration signal and threshold detection	31
3.14	Determination of threshold current	31
3.15	Current response of ohmic load and ohmic-capacitive load in low frequency region	32
3.16	Extract from the moving average filter implementation	32
3.17	Measured current response with and without additional mov- ing average filter	33
3.18	Block diagram of the final measurement procedure	34
3.19	Schematic of the circuit for shunt resistor determination	35
3.20	Initialization steps of the system	37
3.21	Signal stored in 32-bit array	38
3.22	Current response of a high frequency signal	39
3.23	Sequence of the implemented measurement	40
3.24	Applied signal with 40mV amplitude at 18Hz	41
3.25	Applied signal with 40mV amplitude at 2.33kHz	41
3.26	Schematic of the circuit to determine the current response	43
3.27	Oscilloscope output for current determination	43
3.28	Comparison of cross-correlation calculated in Matlab versus on chip	44
3.29	Measured impedance before the countermeasures	45
3.30	Measured impedance after the countermeasures	45
3.31	Two measurements taken with the same load	46
3.32	Nyquist plot for $40k\Omega$ resistor parallel a $470nF$ capacitor	47
3.33	Nyquist plot for $40k\Omega$ resistor parallel a $4.7\mu F$ capacitor	47
3.34	Nyquist plot for $40k\Omega$ resistor parallel a $10\mu F$ capacitor	47
3.35	Comparison PalmSens and test-chip with a load of $10k\Omega$ re- sistor parallel a $470nF$ capacitor	48
3.36	Comparison PalmSens and test-chip with a load of $30k\Omega$ re- sistor parallel a $4.7\mu F$ capacitor	48
3.37	3-component measurement setup	48
3.38	Measurement results with 3 components	48

3.39	Schematic of the recreated equivalent circuit	49
3.40	Nyquist plot of the equivalent circuit measured with PalmSens	50
3.41	Nyquist plot of the equivalent circuit measured with the test-chip	50
3.42	Setup for EIS measurement on PCB	51
3.43	Blank measurement of sensors on PCB with test-chip	52
3.44	Comparison of blank measurement with test-chip and Autolab (potentiostat)	52
3.45	Comparative measurements of the sensors on PCB with test- chip and Autolab (potentiostat)	53
4.1	CIC decimation filter structure	58
4.2	Structure of a three-stage CIC decimation filter	58
4.3	Applied signal with 10mV amplitude and noise overlay	60
4.4	Applied signal with 10mV amplitude at 470Hz	60
4.5	Current response in low frequency region	61
4.6	Schematic of the circuit for bit stream representation	62
4.7	Representation of 1-bit stream	62
4.8	Representation of 1-bit stream with oscillation	62
4.9	Impedance plots with different frequency compensation elements	63
4.10	Current response for different frequencies	64
4.11	Representation of current recession	65
6.1	Representation of a possible use case	70
6.2	Depiction of the potential measurement setup	70
6.3	Front of a possible demo design	71
6.4	Back of a possible demo design	71
A1	Implemented cross-correlation function	79
A2	Implementation of calibration function	80
A3	Implementation of the additional moving average filter	81

Abbreviations

PC personal computer	1
EIS electrochemical impedance spectroscopy	2
AIT Austrian Institute of Technology	3
IgG immunoglobulins g	3
WE working electrode	6
RE reference electrode	6
CE counter electrode	6
SAM self-assembled monolayer	6
WEs working electrodes	10
dc direct current	11
ac alternating current	11
IDE integrated development environment	11
PCB printed circuit board	11
NFC near field communication	14
DAC digital-to-analog converter	15
ADC analog-to-digital converter	15
SIF sensor interface	15
UART universal asynchronous receiver transmitter	16
PBS phosphate-buffered saline	16
RAM random-access memory	20
GPIO general purpose input/output	20

VHDL very high speed integrated circuit hardware description language	
VDD supply voltage	24
DMA direct memory access	28
LDO low-dropout regulator	36
FSR full scale range	38
CPU central processing unit	57
ISR interrupt service routine	57
CIC Cascaded-Integrator-Comb	58

Chapter 1

Introduction and goal of the thesis

Nowadays the number and variety of biochemical sensors is high, like their field of application. The distribution ranges from food monitoring [1], [2] to commonly known chemical "tongue" applications [3], [4]. The distribution of the variety of biochemical sensors also includes monitoring of the water quality [5] up to medical and health applications[6], [7]. Whereby this thesis focuses on the latter ones from the health sector.

All above mentioned applications need a sensor to measure any kind of quantity. As it is written at Thevenot et al.[8], chemical sensors usually consist of a chemical recognition system(receptor) in series with a physicochemical transducer. The biochemical receptor is responsible for the specificity of the sensor. That is the ability to detect only one specific analyte [9]. Most of these sensors are selective for a particular analyte, where the receptor should provide a high degree of selectivity [8]. It is described at Zisper [9] as detection of components in an unknown mixture. The transducer transfers the output signal from the receptor to, mostly, an electrical one, but it also works vice versa, to stimulate the sensor with an electrical signal [8].

This recorded signal is processed by an electrical circuit subsequently. For the processing either a personal computer (PC) [4] or a small stand alone circuit[7] is used. Where the last could be connected wirelessly or wired to

the PC or to a smartphone for further, more complex signal processing. Also, the data output or graphic illustration of the data is often accomplished over these terminals. In a number of new applications smartphones are used as a display and processing unit, which leads to so-called smart applications dedicated for layman-use at home[10], [11]. Especially in the medical and health sector they are strongly present.

Without any control, no electrical measurement could happen. So, the task is to control the sensor with the connected circuit to get results, which can be used for further calculations or predictions. Furthermore, there is some kind of logic hardware and a program/firmware needed, to transform the measured signals into output data. A variety of electrochemical measurement methods can be implemented for this purpose. The focus of this work is on the electrochemical impedance spectroscopy (EIS), which has already been used in other applications.

Electrochemical immunosensors can determine potentially harmful substances in drinking water, which was done by Boffadossi et al.[12]. In this application the concentration of microcystin-LR, one of the most common and hazardous cyanotoxins, is monitored. The used measurement method is EIS.

A system especially for EIS measurements was designed by Gu et al. [13]. It has low power consumption, particularly designed for wearables or implants. The low-frequency and low-amplitude stimulation corresponds approximately to the present work.

Another application [14] uses functionalized gold electrodes in a microfluidic system to determine cytokines and provide information about the immune response. The measurements were also performed using EIS.

An international team of researchers developed an electrochemical immunosensor to detect tumour necrosis factor- α (TNF- α) cytokine in human saliva, what may be used for monitoring heart failure patients [15]. This application also uses functionalized gold electrodes with a similar structure to the ones used in the present work. Also, the same measurement method, EIS, was used.

Motivation

This thesis combines two parts from two companies to build one application. On the one hand there is the newly developed test-chip from Infineon Technologies Austria AG. On the other hand there is the sensor, designed by the Austrian Institute of Technology (AIT). The sensor was designed in course of a doctoral thesis [16]. These two parts are electrically connected. The description of the used firmware and its implementation is the main part of this thesis. Through this firmware the system runs and enables further signal processing.

The two hardware parts, the test-chip and the sensor, fit very well together. The test-chip from Infineon is specially designed for electrochemical measurements and the sensor requires the electrochemical impedance spectroscopy as measurement method. Using this measurement method a sinusoidal signal is applied to the electrodes and the electrical response of the system under test is measured [17]. The applied signal is varied in frequency, what leads to a measured impedance spectrum. This spectrum is either plotted or used for further calculations.

The elaborated system builds a demonstrator to show the functionality and the possible field of application of the newly developed test-chip.

Another aspect is the possible medical use of the application. The results from the measurements should help to track the stress level and the health status overall. It will be designed for layman-use at home, so only a smartphone is needed in addition.

Goal of the thesis

The aim of this master thesis is to perform an electrochemical measurement, electrochemical impedance spectroscopy. A system-on-chip designed by Infineon in combination with a sensor, which is developed and produced at the Austrian Institute of Technology in Vienna is used. It enables to determine the impedance between the electrodes with the implemented method and can thus estimate the quantity of specific biomolecules. These biomolecules can be cortisol or immunoglobulins g (IgG) in saliva later. With this measure-

ments the human health status and stress level can be monitored. In further consequences it could enable burnout prevention.

Chapter 2

Measurement method and procedure

In this chapter of the thesis the characteristics of the measurement method, electrochemical impedance spectroscopy, and the concept of the implemented measurement procedure are covered. Moreover, the planned evaluation steps and the PCB design are described. Chapter 2 finds its conclusion by the description of the measuring routine for the sensor.

2.1 Fundamentals of measurement method

The method, which is implemented, is the electrochemical impedance spectroscopy. It belongs to the electrochemical measurement methods, which further include the cyclic voltammetry, the differential pulse voltammetry and the normal pulse voltammetry. All the measurement methods have in common that an electrical signal is applied and the response of the system is measured. The difference between the methods is the signal form and whether a voltage or a current is applied or measured. In the present work a sinusoidal voltage is applied and the current response is measured to meet the specifications set by the sensor from the AIT.

There are three electrode arrangements used for electrochemical measurements. Either a two-electrode system, which is shown in Figure 2.1, a

three-electrode system, which can be seen in Figure 2.2, or even a system with four electrodes [17]. The three-electrode system will now be discussed in more detail, as it is part of this application. It is used in combination with a potentiostat, which is able to perform the common measurement methods described at the beginning of this section. The main task of the potentiostat is to apply a voltage to an electrochemical cell and simultaneously measure the current response [18]. The voltage is applied between the working electrode (WE) and the reference electrode (RE) [17]. The counter electrode (CE) establishes a desired cell input voltage between WE and RE [18]. The current is simultaneously measured at the WE.

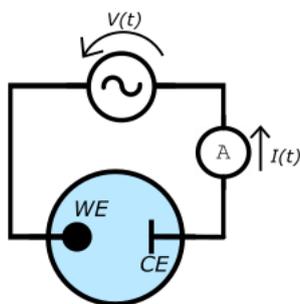


Figure 2.1: Two-electrode system for electrochemical measurements [17]

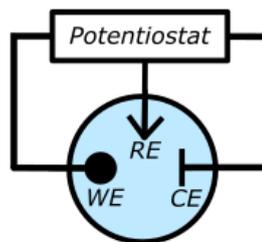


Figure 2.2: Three-electrode system with potentiostat for electrochemical measurements

The sensors used for electrochemical measurements are biosensors with an electrochemical transducer, which usually are chemically modified electrodes coated with a biochemical film [8]. Usually electrodes with solid metal surfaces like gold, copper or silver are used because on this surface thiol groups spontaneously form a so-called self-assembled monolayer (SAM) [19]. These chains can easily be functionalized with various groups to act as biosensors. Typically antibodies or antigens are used, to which the analytes bind. The design of such a sensor structure is shown in Figure 2.3 in a simplified way.

The signal is applied via the functionalized electrodes, which then serve both as exciter and transducer. The power supply is provided by the connected circuit, which in turn receives its energy from a battery or the con-

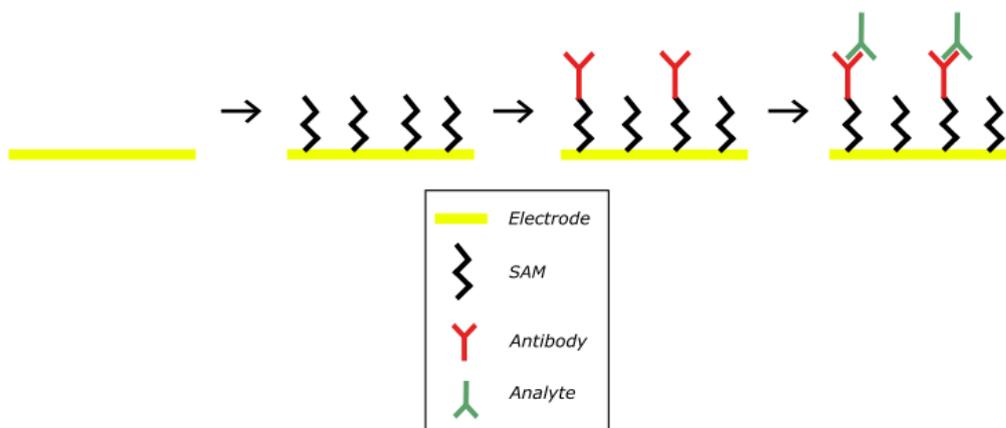


Figure 2.3: Simplified presentation of an electrochemical biosensor

nected terminal device. But there are also applications which produce their own energy like in the project from Chen et al. [20].

Measurable is the change of one or more solution properties by either a formation of a product or consumption of a reactant [21]. Thus, it is formed a redox reaction at the electrode's surface. Therefore, the interaction changes the measured impedance and allows conclusions to be made regarding the analyte concentration. The measurement result is the selective quantity/semi-quantity of an analyte [8].

The used sensor design works according to the antibody-analyte-principle. More information about the explicit electrode design and measuring characteristics can be found in Section 2.1.2.

Since this thesis deals with the implementation of the electrochemical impedance spectroscopy, the fundamentals are explained in the following section.

2.1.1 Electrochemical impedance spectroscopy (EIS)

This measurement method is characterized by applying a sinusoidal voltage or current to a sample and vary the excitation signal in frequency, starting with the highest frequency down to the lowest. Simultaneously the current response or voltage drop is measured to determine the impedance of the system. The measured impedance represents the interaction of different re-

sistive and capacitive components. These components result from the sensor including the biofilm, which is attached to the electrodes and changes the double layer capacitance [22].

2.1.1.1 Equivalent model

The Randles circuit, illustrated in Figure 2.4, approximates the electrode-electrolyte-relation best for biosensors.

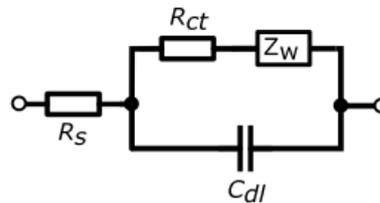


Figure 2.4: Schematic of the Randles circuit

In Figure 2.4 R_s describes the solution resistance. The charge transfer resistance R_{ct} reflects the electron transfer between electrode and redox probe. C_{dl} is the double layer capacitance, which represents the double layer effect at the electrode surface. The Warburg impedance Z_w describes the charge diffusion near the electrode surface. [23]

2.1.1.2 Impedance determination and representation

The impedance represented by the equivalent model is calculated from the applied and measured current/voltage. Therefore, the maximum amplitude of the applied signal and the measured one as well as the phase shift φ between the two signals need to be determined. Equation 2.1 demonstrates the determination of an impedance at a certain frequency by applying a voltage and measuring the current response. This and more fundamentals can be found in [24].

$$Z = \frac{V_0}{I_0} \cdot (\cos(\varphi) + j\sin(\varphi)) \quad (2.1)$$

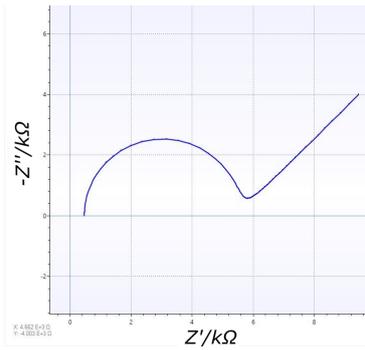
The current response has to be measured for all applied frequencies, so that the impedance can be calculated according to Equation 2.1. The established impedances can then be plotted either in a Bode plot or more popular for EIS measurements a Nyquist plot. The negative imaginary part ($-Z''$) is plotted over the real part (Z'), which is illustrated in the following figures.

Figure 2.5a shows the Nyquist plot of a RC-parallel circuit. The impedance is calculated as follows:

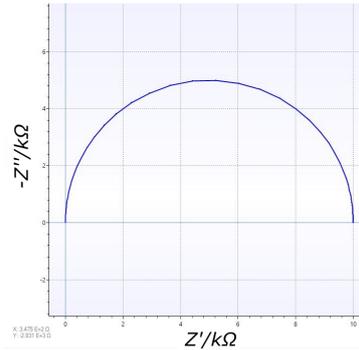
$$Z = \frac{R}{1 + j\omega RC} \quad (2.2)$$

with $\omega = 2\pi f$. At high frequencies ($\omega \rightarrow \infty$) the bottom term goes to infinity and the impedance tends towards zero. At lower frequencies ($\omega \rightarrow 0$) the lower term becomes 1 and the impedance equals R.

Figure 2.5b the Nyquist plot of a Randles circuit.



(a) Nyquist plot of a RC-parallel circuit



(b) Nyquist plot of a Randles circuit

Figure 2.5: Nyquist plot examples

The parameters of the Randles circuit can be recognized from the Nyquist plot. The starting point of the impedance plot at the left is R_s . Followed by a semicircle formed by the C_{dl} in combination with the R_{ct} . The latter one can be determined by reading the decisive point and subtract R_s . In this example R_s is disappearing small. After the notch an increasing straight line with a gradient of approximately 45° follows, which represents Z_w . All parameters are marked in an impedance plot in Figure 2.6.

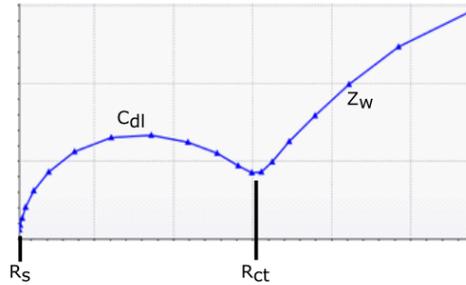


Figure 2.6: Visualization of the parameter of the Randles circuit on an impedance plot

2.1.2 Sensor specific measurement specification

The measurement has been developed especially for the system, consisting of the test-chip from Infineon and the sensor from AIT, so some specifications have to be considered. Measurement specifications have to be met depending on the used electrode layout. The layout is illustrated in the following Figure 2.7. On the left-hand side are the connections to the potentiostat and these connections are identified as follows. Number 1 indicates the reference electrode (RE), 2 the counter electrode (CE) and 5 to 9 the working electrodes (WEs). Thus, the RE and CE of all sensors are interconnected, only the WEs are controlled individually. In this case one sensor consists of the RE, the CE and a WE each.

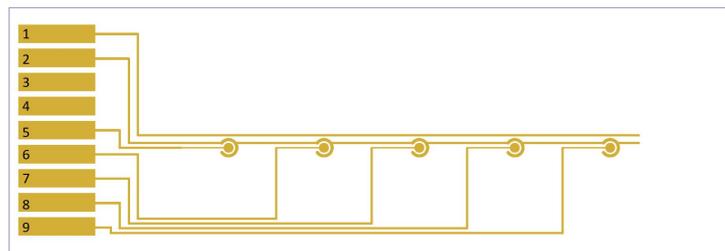


Figure 2.7: Electrode layout on glass [16]

The electrodes are made of gold, only the RE is coated with an Ag/AgCl layer. The Ag/AgCl is applied in the form of a paste on the gold wires. The gold wires of the WEs and CE encourage the creation of a SAM that gets activated in a further step to bind antibodies. The antibodies enable

the measurement due to binding specific molecules. The preparation steps of gold electrodes can be found in the doctoral thesis [16].

The measurement specifications include a direct current (dc) offset and a superimposed alternating current (ac) component. For the measurement itself a dc offset of 160mV is applied between the WE and RE to start the redox reaction at the WE [25]. This reaction forms an electric current flow and a current response is measured at the WE. After a short waiting time, which needs the system so settle, it is superimposed with a sinusoidal signal. This signal has an amplitude of about 5mV - 10mV. The applied frequencies should be in the range of 10kHz to 1Hz. All the specifications and further information can be found in the dissertation of J. Schrattenecker [16].

The current response is measured as simultaneously as possible with the sinusoidal stimulation. So, the response of each sensor can be determined this way. Within this system it is only possible to measure one sensor after the other.

2.2 Measurement procedure

This section describes the concept of the measuring procedure, including all necessary steps to determine the impedance of the connected circuit or sensor. The implementation was performed in the integrated development environment (IDE) from Eclipse Foundation 2019-12 version 4.14. Also the printed circuit board (PCB) design and the planned evaluation steps are part of this section. Finally, the routine of sensor measurements is described.

2.2.1 Draft of measuring procedure

Before the actual measurement takes place, some pre-functions have to be called. At the beginning the used frequencies are determined in a fixed interval with a desired number of frequency points specified. Initially the dc offset is applied to the system, between the WE and RE. It forms an electric current flow to measure the current response of the sensor. The current response at the WE is measured as soon as the sinusoidal signal is applied.

The measured data is stored in an array.

Then the raw data is used in the signal processing part, which includes the following. First the cross-correlation is determined, to define the phase shift. Second the maximum of the applied and measured signal is determined. These parameters are obtained for the calculation of the impedance, which is shown in Equation 2.1. This is done for every frequency point and repeated several times at the same frequency. The results of the repeated measurements are averaged and thus improve accuracy. The last computation on chip is the impedance calculation.

Any further steps, like plotting, are done externally. For this thesis the tool Matlab, version R2017a, from The MathWorks Inc is used, where the data is imported, in some cases further processed and plotted. The impedance can be represented in different ways, either as Nyquist or Bode plot. A specific application is to use the gained impedance values to determine the health state or track the fitness level. The whole measurement method, which is going to be implemented is shown in Figure 2.8.

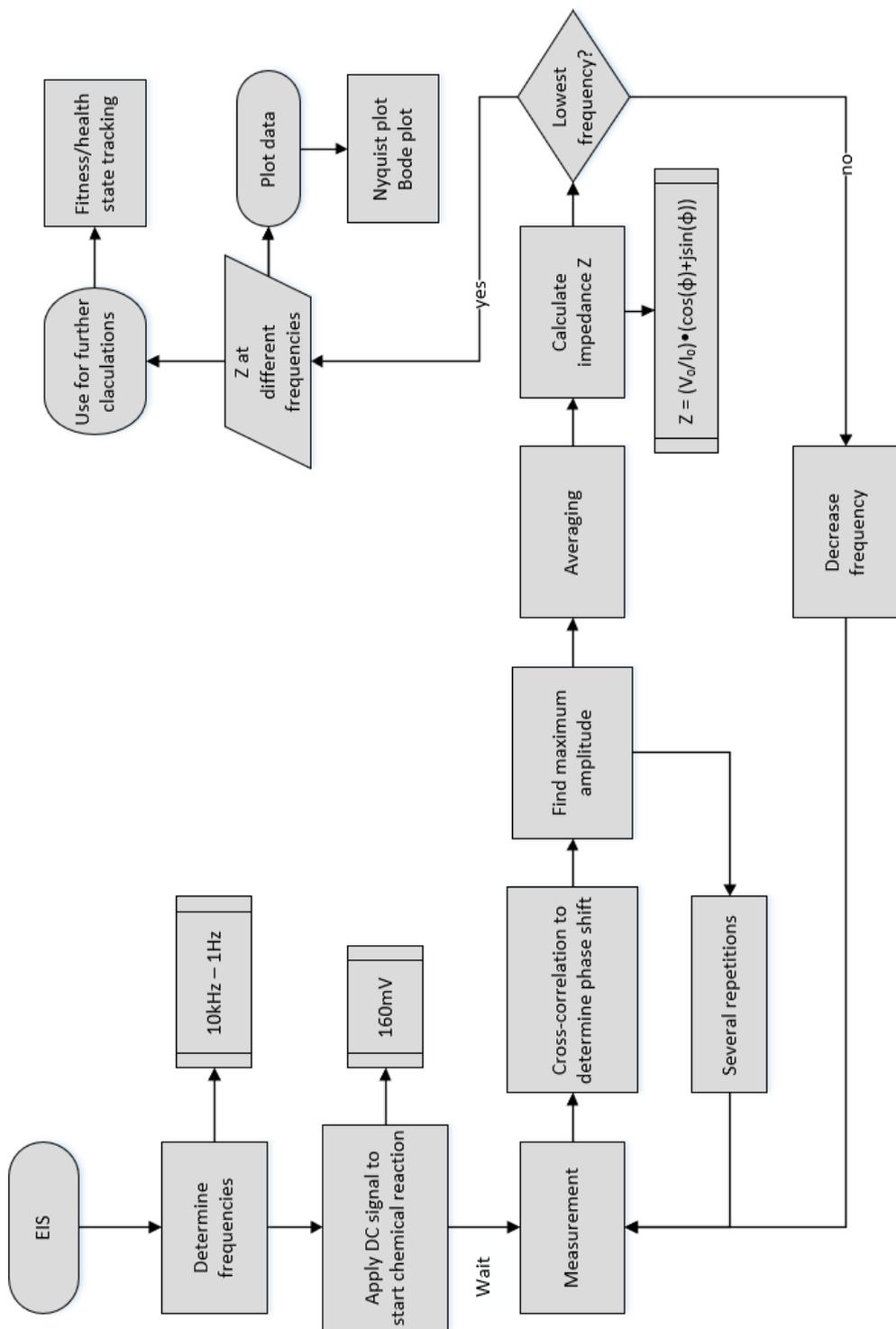


Figure 2.8: Initial block diagram of measurement procedure

2.2.2 PCB design

The outcome of this thesis should be a demonstrator, including the electrodes together with the chip and additional components, like the near field communication (NFC) antenna, on one circuit board. Therefore, the sensor has to be manufactured and tested on PCB, initially in the same size and shape as the developed one at the AIT, see Figure 2.7. So, it is possible to verify and test it in the same device and the only varied condition is the surface texture. The PCB is designed in Autodesk Eagle, CadSoft Computer GmbH, version 9.2.

For this purpose, the electrode layout developed at the AIT is transferred to PCB, which is shown in Figure 2.9. Both layouts use gold for the electrodes' surface and the conductor lines are of same thickness. The only difference is the carrier surface of the gold electrodes, which could lead to different results, as the experiments showed in the doctoral thesis of [16].

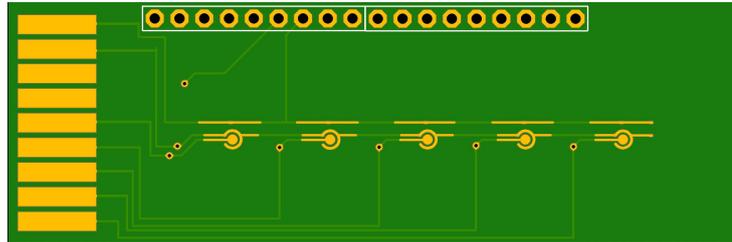


Figure 2.9: PCB layout of the electrode configuration

The PCB layout has the same dimensions as the original design on glass. So, it is possible to test it in the same structure. Additionally, it has a connector strip at the top for easy connection to the test-chip board. The combination of the gold electrodes on PCB and the test-chip is shown in Figure 2.10.



Figure 2.10: Electrodes on PCB connected to the test-chip board

2.2.3 Planned evaluation steps

As already mentioned, the chip was just designed and manufactured, so the basic functionalities have to be tested first. In the context of this work only the evaluation of the parts necessary for the measurement is mentioned. Therefore, testing the signal generation with the digital-to-analog converter (DAC) and the measurement using potentiostat and analog-to-digital converter (ADC) is going to be described in the following Chapter 3. Moreover the tests for verification of the software are described in that chapter.

There are several functions which can be tested without any hardware, like the cross-correlation, maximum detection and signal array generation. These functions were already implemented and tested before the chip arrives, by using test data. Matlab is used for the evaluation, as these functions are available and easy to apply. Like `xcorr()` [26] is used for the cross-correlation function or `max()` [27] to determine the maximum in an array.

But there are also functions, like the measurement itself, which hardly or cannot be tested without the hardware. Firstly, the right initialization of the sensor interface (SIF), including DAC and ADC, is checked. Then the signal generation and output is tested, also the different modes of the DAC are checked. The verification of the applied signal is simple because it is measurable at the electrodes' output with an oscilloscope. In a further step the measurement via the potentiostat and ADC is tested. A dc signal is

applied and the measured signal is sent via universal asynchronous receiver transmitter (UART) to the PC. For this purpose, another program is used, the HTerm version 0.8.4, which is a RS232-interface. The evaluation of the measured signal is more complicated with changing signals like the sinusoidal signal. For this part of the evaluation Matlab is used to make the results graphically visible and easier to identify and validate. The data is sent to the PC via the UART interface again and inserted in Matlab for further processing. This enables the measurement relevant parts to be tested.

2.2.4 Sensor measuring routine

The test setup includes the functionalized sensor from the AIT, the test device used by AIT for all previous measurements and a measurement unit. The measurement units are the test-chip and Autolab (PGSTAT12, Eco Chemie B.V., Netherlands). Furthermore, some measurement solution, which enables the electrochemical measurement, and sample fluids, which contain the analyte to be determined, are needed.

After the preparation the sensor can be stored in the fridge at 4°C for some days. To reactivate the sensor, a phosphate-buffered saline (PBS) buffer is put on the electrodes for 30 minutes. The buffer is washed off and replaced by the measurement solution hexaammineruthenium(III), which leads to a redox reaction and as a result to a current flow. Subsequently, the blank measurements are taken to test the functionality. These results can be used as reference for result processing in the end. The blank results are subtracted from the measurement results to get rid of the circuit induced impedance. In the next step the measurement solution is replaced by the sample fluid with different concentration in each measurement chamber. The concentrations are 10ng/ml, 100ng/ml, 1µg/ml, 10µg/ml and 100µg/ml. These fluids rest for another 30 minutes. During this time the analytes bind to the antibodies. After the specified period the rest of the sample fluid is washed off to avoid interferences in the measurement. Now the measurement solution is put on again and the measurements are taken alternately with the test-chip and Autolab. The whole routine is shown in figure 2.11.

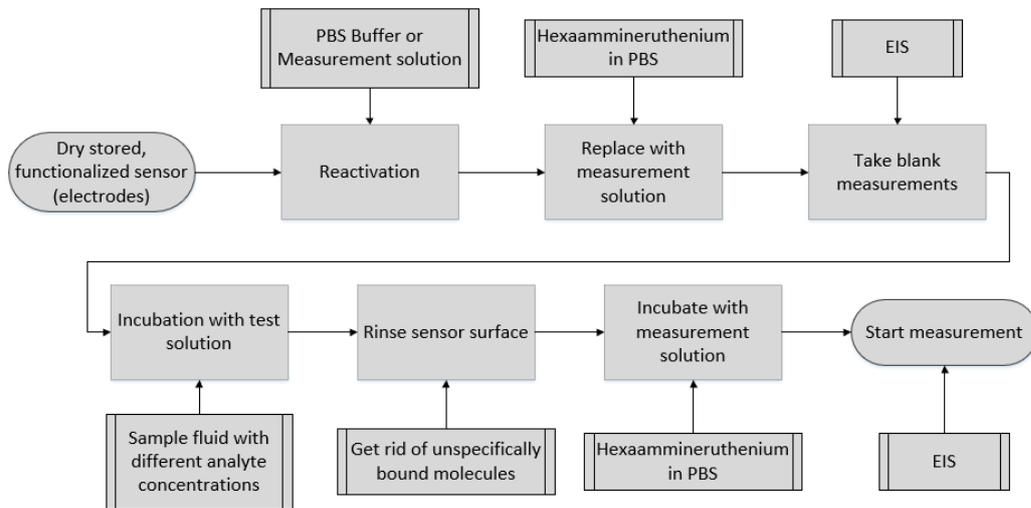


Figure 2.11: Measuring routine with sensors

The measured current flow depends on the number of bounded analytes. The more analytes are bound, the bigger is the impedance. The impedance also depends on the applied frequency regarding the electrode behaviour. The applied voltage and measured current is used to calculate the impedance, like in Equation 2.1. The typical graph of real part to the negative imaginary part of the impedance looks like in figure 2.5b.

Chapter 3

Implementation, evaluation and testing

This chapter describes the implementation steps, parts of the evaluation and testing. Before the implementation of the measurement method, the basic functionality of the chip is tested. During the implementation further testing is done and thus the implementation is adjusted. Finally, the measurements are performed with the fully implemented measurement procedure and the provided sensors.

3.1 Test-chip overview

Figure 3.1 shows a simplified block diagram of the new test-chip. It uses an ARM Cortex(R)-M0 microcontroller, which is the core of the system and operates at a frequency of 24MHz. The system is extended with a sensor interface (SIF), which is especially designed for electrochemical measurements. It contains a DAC [28] for signal generation, a potentiostat for driving the biochemical sensor and an ADC [29] for measuring the current response of the used sensor. The potentiostat is the connection between the electrodes and the chip. That is illustrated by the remaining two blocks in the SIF representation, the On-Chip Sensor in combination with the Sensor MUX for switching between the different WEs. Also used for the application is the

random-access memory (RAM) for temporary storage of results. The UART with some of the 16 general purpose input/output (GPIO) pins are needed to output additional data to an oscilloscope or transfer it to the computer.

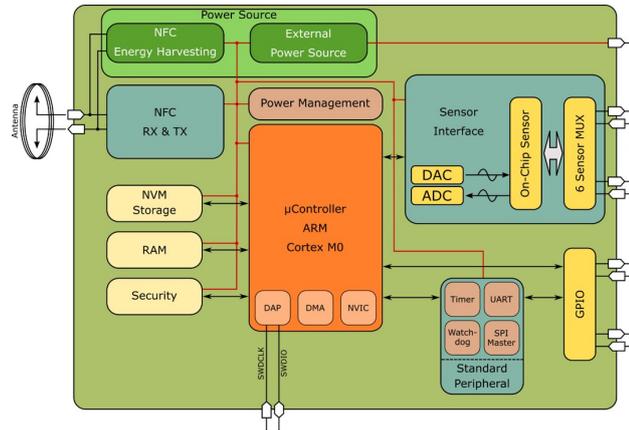


Figure 3.1: Simplified block diagram of the test-chip
 Source: Infineon Technologies Austria AG, 2019, internal documentation, unpublished

For quick commissioning of the system the firmware development was already started prior tape out of the test-chip. To test the software without the actual hardware, a simulation environment was used.

3.1.1 Sensor interface structure

The SIF includes the DAC, ADC, electrodes management and all other parts associated to the potentiostat.

The DAC is responsible for the signal output. The structure of the DAC offers different operation modes [28]. It consists of two resistor ladders, see Figure 3.2, in series, whereby the second can be deactivated. Each ladder has a resolution of 8 bit. If the ladders are combined in series the resolution is 16 bit. These ladders are called stages in further course. The first and second stage has two taps, where the required voltage can be taken from. The whole structure is illustrated in Figure 3.3. Furthermore, a distinction can be made between dc and ac mode. This implementation uses the so-called 16-bit ac mode. In this configuration both outputs of the DAC are used, one for the

RE and the other one for the WE.

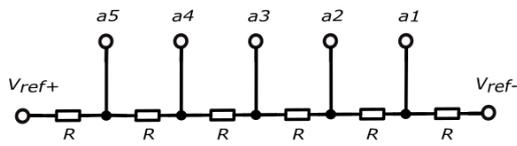


Figure 3.2: Schematic of a resistor ladder

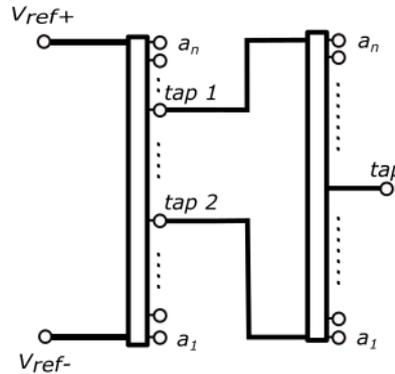


Figure 3.3: Schematic of two resistor ladders in series

The data values are stored in a 32-bit array. Bits 1-8 represents the first stage first tap, bits 9-16 set the second tap in the first stage. The upper word (bits 17-32) controls the taps of the second stage accordingly. This structure is loaded into the appropriate register for output and is shown in Figure 3.4. Only one tap in the second stage is used at a time, depending on the mode, either dc or ac mode.

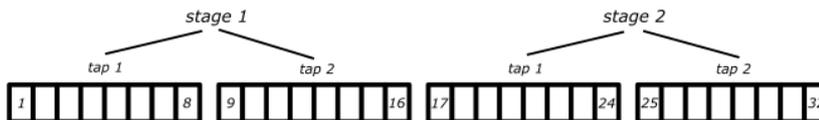


Figure 3.4: Schematic of the DAC register

The ADC is used with the highest resolution of 15 bit and in continuous mode. So, it is continuously measuring and storing the values in the corresponding register. The values have to be picked up at certain time intervals, depending on the frequency of the applied sinusoidal signal. The picked-up values are stored in an array for further use.

The potentiostat provides two different measurement methods: The current mirror mode, where the current response is directly measured, and the shunt mode, where the current response is measured in terms of a voltage

drop on a shunt resistor. The voltage drop on the shunt resistor should be in the range from 600mV to 100mV. As soon as the voltage drop becomes bigger, the next smaller shunt resistor is selected, and vice versa. The current mirror mode works with a similar principle but so-called stages instead of resistors. There are 4 stages ranging from low μA to 1mA. If the measured current is outside the input range of the ADC, the stages in the potentiostat are reconfigured to ensure proper current acquisition.

Both available methods were tested. For high capacitance loads, the current mirror mode was not able to drive the required current and thus delivered inaccurate results. That is why the shunt mode is chosen for the following application. The shunt mode is available for four of the six electrodes, WE1 to WE4. To cover a wide range of possible results WE1 was chosen for the measurements. WE1 provides the biggest available current input range, from $1\mu\text{A}$ to 1mA. The used WE1 includes eight different shunt resistors, from $1\text{k}\Omega$ to $3\text{M}\Omega$.

The electrodes need to be activated separately. Except the CE, which is already initialized with the activation of the potentiostat. Each electrode, whether it is a WE or the RE, has to be configured. Then the input channel is chosen. There are four possible input sources for the potentiostat. These are the two DAC outputs, the reference voltage (1.5V) or the output of the positioning block. After initialization, the selected signal can be applied to the electrode connections.

3.2 Simulation environment

The used simulation environment is provided by Cadence Design Systems, Inc. The entire digital part of the system-on-chip is developed by the usage of several hardware description languages like Verilog, System Verilog and VHDL. With this implementation the digital part of the entire system on chip can be simulated with NCSim. The graphical interface SimVision monitors the simulated signals. The graphical interface supported the development and evaluation of the code on the virtual system. Due to the simulation of the test-chip's characteristics it was possible to detect some mistakes in the

hardware implementation, which was directly fixed by the digital designer.

In order to verify also some analog components in the digital environment, it was required to introduce analog behavioural models in the simulation environment. One example is the DAC module for the SIF, which is required to perform the analog signal generation.

3.2.1 Signal generation

Regarding the specifications set by the AIT, as discussed in Section 2.1.2, an output signal with a 160mV dc offset between the WE and RE superimposed by a sinusoidal signal with 5mV - 10mV amplitude has to be generated.

The RE is set to ground and the WE is to tap 160mV above the RE. This is set in the first stage. Also, the required difference of 10mV is set in the first stage between the two taps. This leads to a voltage of 155mV at the first tap and 165mV at the second tap in the first stage of the DAC. The whole settings are demonstrated in Figure 3.5, where the second stage is tapped in the middle to achieve 160mV at the electrode output.

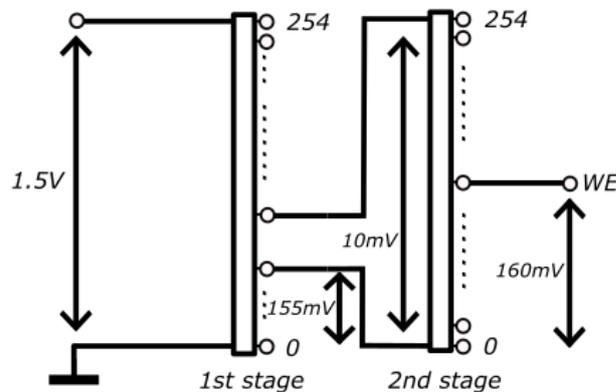


Figure 3.5: Reduced representation of the DAC structure in the simulation

The configuration of the first stage is changed in the following implementation compared to the simulation. The simulation is only a mapping of the analog blocks up to a certain extent, thus the analog system can show a slightly different behaviour. So, the RE is set to the upper end of the first resistor ladder and the WE 160mV beneath it. This is necessary to get the

electrode potential in the middle of the supply voltage (VDD). Otherwise the CE would not be able to regulate the potential.

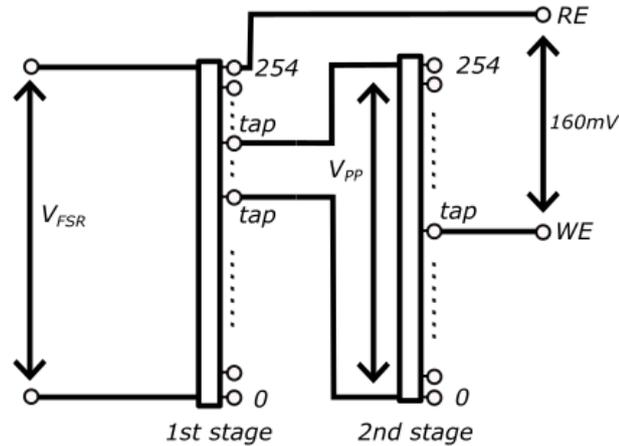


Figure 3.6: Reduced representation of the DAC structure in the implementation

In the second stage the values for the sinusoidal signal are applied with a resolution of 8 bit. The signal for the simulation is stored in an array of 200 values for one sine period, where array values range from 0 to 254. This array can be found in the appendix.

In further sequence the sinusoidal signal is generated from a fixed array, which is also included in the appendix. The array holds only the values for a quarter period of a whole sine of 200 points, therefore 50 points. The values range from 0 to 254, in hexadecimal notation 0x0 to 0x7f. The hexadecimal notation is used throughout the whole program for any numbers.

The generation of the whole sine wave takes place in four steps as shown in Figure 3.7. Step I is the initial array, in step II the values get mirrored on the y-axis. Step III mirrors the initial values at the x-axes and the last step mirrors the values generated at step II at the x-axes. This procedure was chosen because it was the easiest way to generate a sinusoidal signal with different lengths. Depending on the frequency the points in the array are reduced, which means for higher frequencies less points. For frequencies above 3kHz only a quarter of the 50 points are considered for the signal formation.

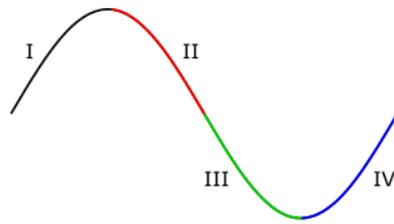


Figure 3.7: Sine wave generation

The required signal values are written into the output register, where they get send value by value with the applied frequency. It ranges from 10kHz to 1Hz, which is gained by varying the interval between the individual output points. So, a sinusoidal signal varied in frequency is visible at the output.

In the simulation environment it is possible to monitor several different signals, like different clock signals, output signal and so on, at the same time. This is helpful in analysing possible inconsistencies. Figure 3.8 shows a possible simulation output.

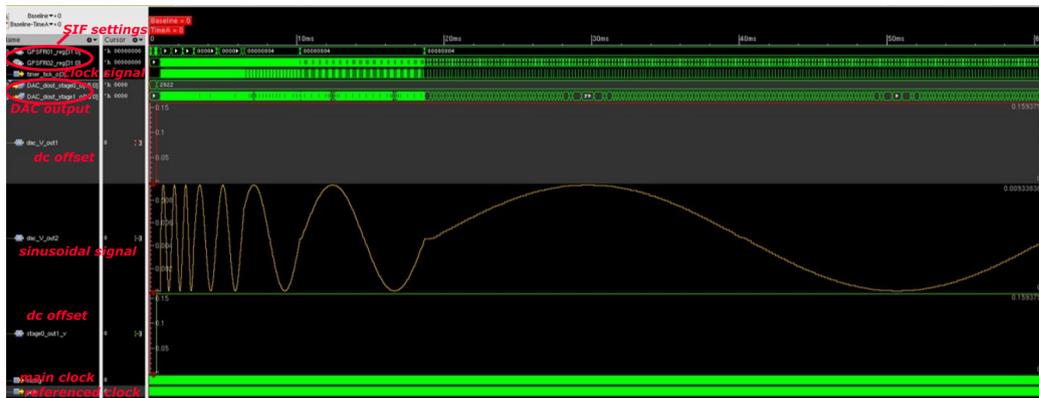


Figure 3.8: Representation of the simulation environment

The first two lines represent the register for general information about the SIF including the configuration settings. Followed by the clock signal, which is changed with frequency. The next two vectors contain the information of the first and second stage of the DAC output signal. The offset is represented by the following red and in the next but one green line, which is hardly recognizable. The sinusoidal signal is in the middle, the yellow graph. It

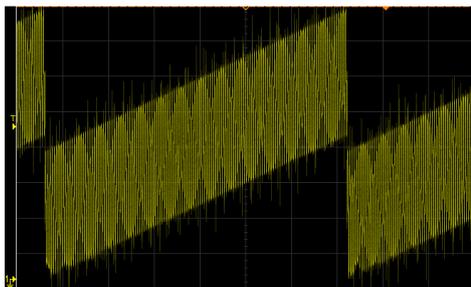
represents already different frequencies. Finally, the main and a referenced main clock is shown in the two last lines. It is only an extract of the available signals, which could be displayed in the program.

The signal generation part was tested in this way, until the test-chip arrived and the evaluation started.

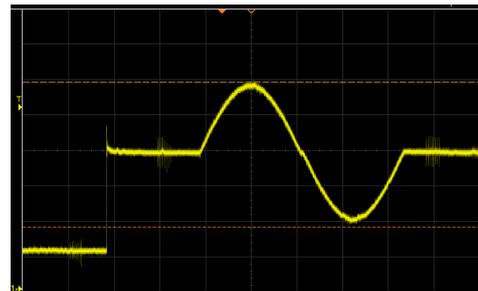
3.3 Evaluation of the sensor interface

The two main parts of the SIF were tested. These include the DAC and the ADC, which are connected to the electrodes by the potentiostat.

In order to test the DAC, the output signal via the electrode was recorded. For that the signal generation was implemented and the verified configuration of the SIF was used. The correctness of the shape of the signal was verified via an oscilloscope. To do so, different kind of signals are applied in different DAC modes. Figure 3.9a shows ramping up the first stage in combination with ramping up and down in the second stage at each point of the first stage. With these two routines combined the whole 16-bit resolution was tested. In Figure 3.9b a dc offset can be seen and the beginning of a sinusoidal signal applied over the offset.



(a) 16-bit ramping with superimposed fast ramping



(b) DC offset with superimposed sinusoidal signal

Figure 3.9: DAC output verification

After the successful evaluation of the DAC, the data measurement was tested. This is more difficult than the signal generation because the recorded signal cannot directly be measured with the oscilloscope. Therefore, the

measured data was read out via UART and displayed in HTerm. Or the data was stored, further processed and plotted in Matlab for easier and faster verification. Thus, the first tests showed the required signal quality and characteristics.

3.3.1 Test configuration

The used DAC test setup includes the test-chip board, a debugging-tool and the oscilloscope. The board and the debugging-tool, the black magic probe, are shown in Figure 3.10. The oscilloscope was attached between the WE(yellow) and RE(brown)/CE(red) in the picture. As long as no real sensor has been used, RE and CE are shorted.

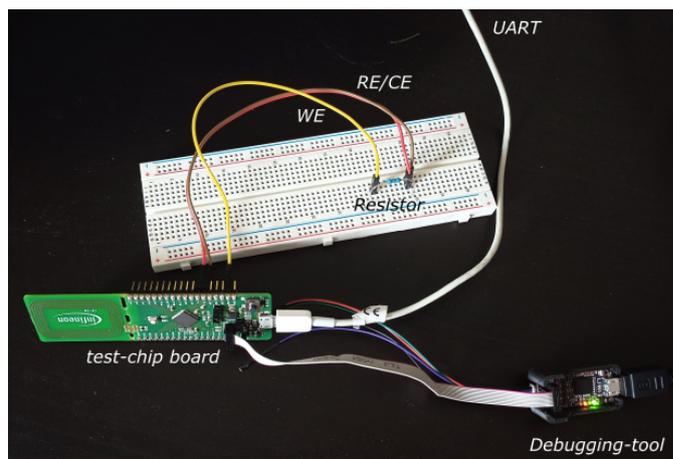


Figure 3.10: Test setup

For the current measurement the setup shown in Figure 3.10 was used. The black magic probe and an additional UART interface are connected to the PC. The measured data were transferred via the UART interface to the PC for further evaluation. The already tested DAC signal was applied to measure the current at the resistor. The measured current response is either immediately displayed in HTerm or stored and graphically represented in Matlab.

3.4 Improvements

During the first tests, unexpected behaviour of the test-chip occurred. One of them was related to transmission of data points at high frequencies. Another one dealt with the delay of the measured data points at the ADC. This could be associated with the functionality of the internal moving average filter. In higher frequency regions, it leads to a lack of nearly one period. This could be eliminated by the introduction of a calibration function. Due to a noisy signal in the lower frequency region an additional filter was implemented in the firmware. For further improvements of the results the real values of the shunt resistors in the potentiostat were determined for more precise calculations.

3.4.1 Handling high frequencies

The system seemed to work fine for high frequencies with the DAC alone but in combination with the ADC it reached its limits. Although, the core clock frequency is 24MHz, the clock cycles are getting tight. A sinusoidal signal with a frequency of up to 10kHz should be generated with at least 25 support points. This corresponds to a transmission frequency of 250kHz. It is approximately a hundredth of the clock frequency and worked fine for the data output only. But this is not enough time to call additional functions for receiving and storing the measured values. However, the specification for the measurement method requires quasi simultaneous excitation and measurement. Overcoming this issue was a major challenge to meet in this thesis.

Three possible countermeasures were tested to overcome the timing issue: the interrupt, the direct memory access (DMA) controller and the use of assembler code. Since the use of the last mentioned variant yielded the most promising results, it is used in this application. It solves the timing problem when both parts, applying and measuring a signal, are active at high frequencies.

Assembler Using assembler leads to efficient code, which executes quickly by working directly with the resources of the system used [30]. If assembler

code is embedded in C-code it is called in-line assembler, which is mostly used for performance-sensitive parts [31]. It is possible to write high efficiently code but only if used correctly, with the understanding of the architecture of the microprocessor.

The coding construction has the following appearance [32]:

```
asm(
assembler template
: output operands (optional)
: input operands (optional)
: list of clobbered registers (optional)
);
```

It is used to control the program regarding timings at any time. In this application it is possible to minimize the delay between the DAC output and the measurement by the potentiostat and ADC as well as operating it at certain frequencies. Even the cycles consumed by the program itself must be taken into account, see Figure 3.11.

```
" MOV r7, %[count]           \n\t// Load number of delay cycles
" SUBS r7, r7, #36           \n\t// #36 compensation for overhead in assembler program (DAC & ADC meas)
"delayLoop:                 \n\t
" SUBS r7, r7, #10           \n\t// 10 cycles => 1x delay loop (SUBS and BGT)
" BGT delayLoopR           \n\t
```

Figure 3.11: Part of the assembler implementation

Therefore, the timing critical part of the DAC and ADC is written in assembler, but none of the other parts, because they are not timing depended.

After the improvement the highest required frequency of 10kHz was reached and the results for lower kHz frequencies got improved. The signal at 5kHz and below can be output with 50 data points instead of only 25, which makes it more precise. Below 1kHz the signal consists of 200 data points. That would not be possible without the enhanced implementation.

3.4.2 ADC filter delay

The already mentioned delay of the measured signal is shown in the following Figure 3.12.

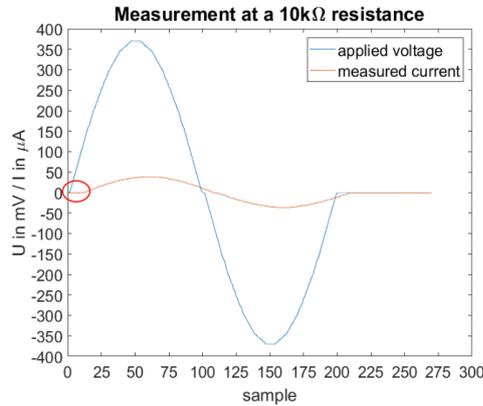


Figure 3.12: ADC filter delay according to moving average filter

The moving average filter of the ADC could be identified as the source of the delay. The filter averages over several input values, whereby every time step the last value drops out and the next one is considered. In the end a new value is written in the register with a frequency of 37.5kHz. Hence, if the input signal frequency is high, an observable delay in the output signal occurs, as shown in Figure 3.12. With decreasing frequencies the delay times get smaller or disappear completely. Furthermore, the register update frequency of 37.5kHz is also the reason why the measured signal at high frequencies also has a step pattern, which is shown in Figure 3.22. In contrast to the step pattern, the delay at the beginning would falsify the measurement. Thus, some compensation in firmware was done to distinguish the delay and avoid corrupt measurement results.

3.4.2.1 Calibration function

The calibration of the delay time is done by just applying a step function to the system, which is shown in Figure 3.13. Whereby a resistor of 10 to 20kΩ is connected between the WE and the RE/CE, because it maps the expected impedance values closest. The data points are applied with the same frequency as they would be applied in the sinusoidal signal.

In this function a threshold is calculated due to the used resistor. First the used resistor is determined, then the maximum occurring current. This

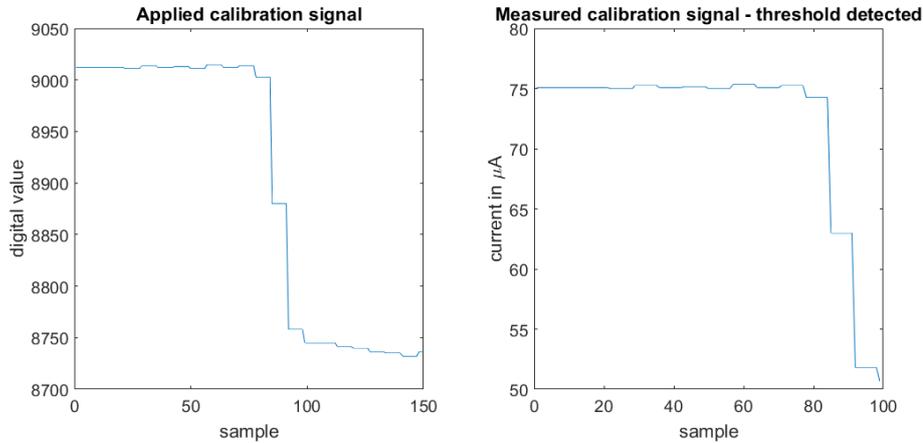


Figure 3.13: Calibration signal and threshold detection

current is increased by the factor determined by the resistor to achieve the most accurate results. This is implemented as follows, see Figure 3.14. In the end this determined maximum current, the threshold, is compared to the measured current.

```
// Determination of external resistance to determine threshold voltage
resistance = (1200 - (100 * 4.7)) / adc_current_sensor_mA;
float max_current_mA = (float) (1200 - ((100 + step) * 4.7)) / (float) resistance;
float factor = (0.01 / resistance) * 10000;
max_current_mA += factor;
```

Figure 3.14: Determination of threshold current

Whenever the threshold is reached the delay is identified, compare Figure 3.13. In case there are some disturbances on one measurement, this is done three times for every frequency. Then the mean value of the three results is calculated to get more precise results. The results are stored in an array for the use during the actual measurement. The implementation can be found in the appendix, Figure A2.

3.4.3 Additional moving average filter

During evaluation and testing it was found that applying a capacitive load leads to a noisier current signal in the low frequency region, what is shown in Figure 3.15.

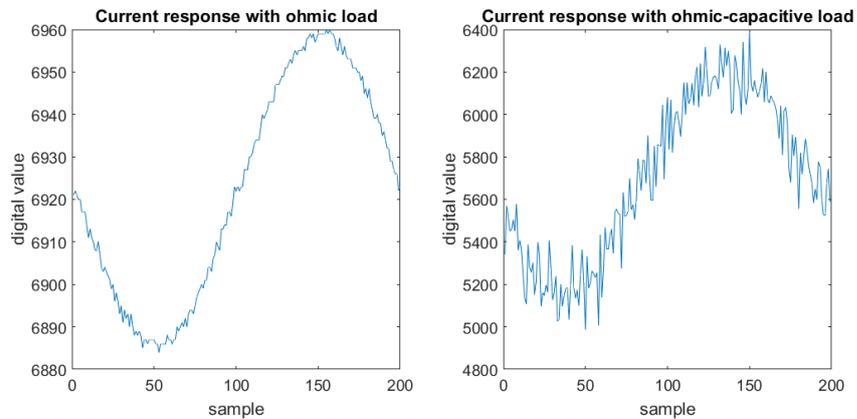


Figure 3.15: Current response of ohmic load and ohmic-capacitive load in low frequency region

```

for (uint16_t i = adc_start - (samples / 2) + 1;
    i < adc_start + sinus_signal_length - (samples / 2) + 1; i++)
{
    int average_res = (int) (((int) adc_data[i + samples] - (int) adc_data[i]) / samples);
    average = average_res + average;
    adc_signal[pos] = average;
    pos++;
}

```

Figure 3.16: Extract from the moving average filter implementation

Therefore, an additional moving average filter was implemented, which averages over 4 to 16 points depending on the frequency of the sinusoidal signal. For higher frequencies less data points are averaged than for lower ones. Whenever a new data point is considered the last one drops out. This is implemented as demonstrated in Figure 3.16. The last entry is subtracted from the next considered data point. The newly determined average is added to the previous one, which contains the data points in between. The filtered signal is stored in an array. The whole implementation is in the appendix, Figure A3.

The effect on the results is shown in Figure 3.17. The spikes are reduced and thus also the detected maximum in the array, which in turn leads to better results.

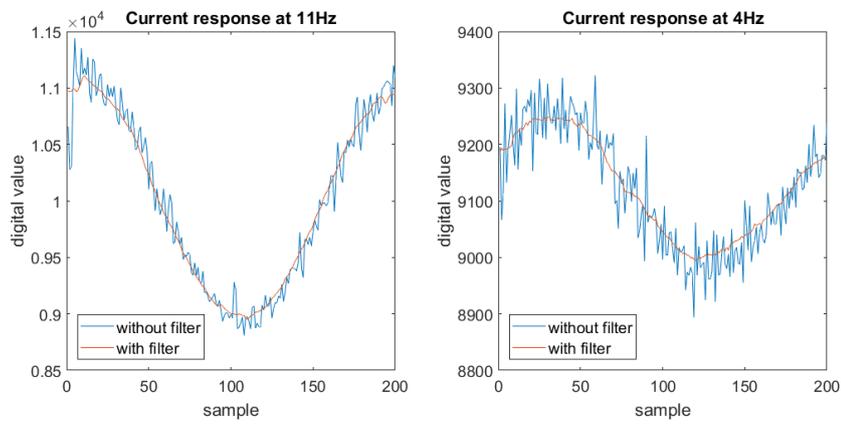


Figure 3.17: Measured current response with and without additional moving average filter

After all the adaptations the improved measurement procedure is illustrated in Figure 3.18, where the improvements are highlighted.

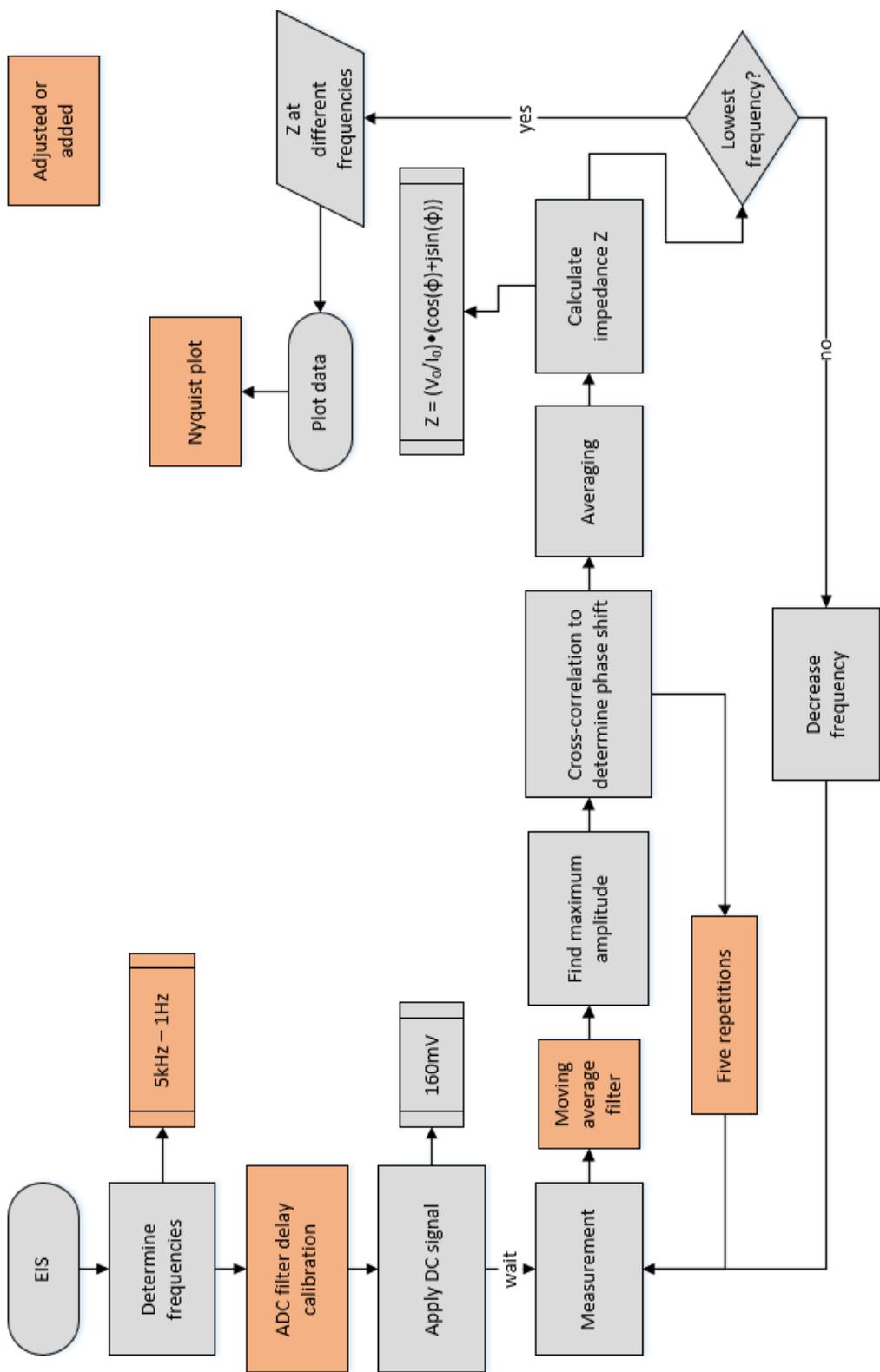


Figure 3.18: Block diagram of the final measurement procedure

3.4.4 Calibration of shunt resistors

If the shunt mode is chosen for the measurements, the shunt resistors could be calibrated to achieve more accurate results. The resistors have a deviation from the nominal value due to manufacturing tolerances. This is determined by identifying the ADC offset and subsequently measuring every single shunt resistor.

The offset (V_{offset}) was determined by just configuring the SIF and the ADC, which ran in continuous mode, and measured without any input signal. There were taken several values and the average of them. The ADC on the used test-chip has an offset of $425\mu V$. Then every shunt resistor (R_{shunt}) of WE1 was determined with the circuitry depicted in Figure 3.19.

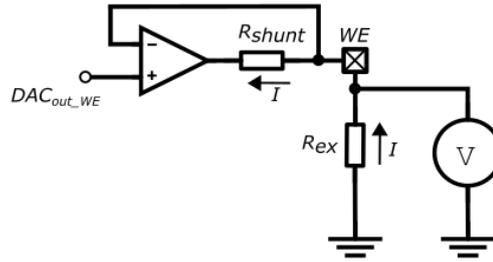


Figure 3.19: Schematic of the circuit for shunt resistor determination

The voltage output at the WE (V_{WE}) is set to 1.5V. Then the external resistor (R_{ex}) is chosen to build a voltage drop at the R_{shunt} of about 600mV, because this area offers the most accurate results.

$$R_{ex} = \frac{V_{WE}}{\frac{600mV}{R_{shunt}}} \quad (3.1)$$

The external resistor has already a tolerance, which has to be measured before and taken that value for further considerations. The voltage drops at the R_{ex} was additionally measured (V_{meas}) to take the non-idealities of the applied voltage (1.5V) into account. The current is calculated with these

measured quantities as follows:

$$I = \frac{V_{meas}}{R_{ex}} \quad (3.2)$$

The current stays the same in the circuit, also over the shunt resistor.

$$R_{shunt} = \frac{V_{WE} - V_{offset}}{I} \quad (3.3)$$

As an example, if the value of R_{shunt} would be $1k\Omega$, R_{ex} has to be $2.5k\Omega$. The closest available resistor has $2.7k\Omega$, with a measured value of $2.692k\Omega$, and the measured voltage (V_{meas}) drop is $616mV$. This leads to a shunt value of 1105Ω and a variance of 10%.

The corrected values are stored internally and used for the conversion from the digital value of the voltage drop at R_{shunt} to the measured current value in mA.

3.5 Implementation of the measurement procedure

The implementation is based on the concept presented in Section 2.2.1. The already presented concept is extended by an initialization part at the beginning. Only small adjustments were made to improve the procedure, as it is shown in Figure 3.18.

3.5.1 Initialization

The first settings, including the chip initialization and start up as well as the initialization of the RAM and UART, will not be discussed in detail in this work. After the main power switch for the SIF and the low-dropout regulator (LDO) is switched on, the initialization of the SIF starts. First, the way of measuring the current response is set, which is in this case the shunt mode. After that, the DAC and ADC are configured and activated. Thereafter, the required electrodes, RE, CE and needed WEs, are chosen and configured. For the beginning only one WE, the RE and CE are used.

They have to be activated and the input channel has to be chosen. The WE and the RE get their signals from the DAC outputs. The whole initialization part is illustrated in Figure 3.20.

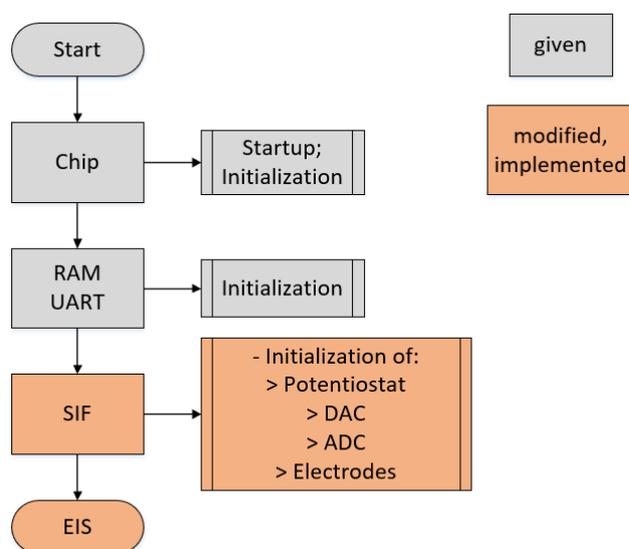


Figure 3.20: Initialization steps of the system

3.5.2 Measurement

Before the actual measurement starts, there are some pre-functions called, the frequency determination and the calibration function.

The applied frequencies are stored in an array as described below.

uint32_t freq[] = {19 frequencies from 5kHz to 1Hz}

The highest frequency (10kHz) was not taken into account as this did not add value to the measurement but brought uncertainty. Thus a fixed array was used with the following values in Hertz: 5000, 3793, 2336, 1438, 886, 546, 336, 207, 127, 78, 48, 29, 18, 11, 7, 4, 3, 2, 1. In the low frequency region more values were taken, because it is the more meaningful part in the graph and in further consequence for the calculations.

Firstly, the reload value is determined, which is the number of clock cycles to overcome before the next data point is applied. The calculation is depicted

in Equation 3.4.

$$reload = \frac{\frac{clockFrequency}{appliedFrequency}}{numDataPoints} \quad (3.4)$$

For higher frequencies it is a smaller number, like 96 for a 5kHz signal, and it becomes bigger with lower frequencies, 1200 for a 100Hz signal. With the generated reload value, the ADC filter delay is determined. For this, the calibration function is analysing the delay time with the use of a step function, see Section 3.4.2.1. The measured delay times are stored in an array.

The full scale range (FSR) of the DAC input is 1.2V, which is the potential of the RE. The potential of the WE is 160mV beneath the RE potential. So, the first stage of the DAC is centric around 1.040V. The first tap is at 1.035V and the second one at 1.045V. To apply the dc offset, the middle value of the second stage is selected. This value is applied to the system and followed by a few seconds pause before the execution of the code continues.

The following part is repeated for every single frequency, including the signal processing and the final calculations.

The sinusoidal signal is generated as described in Section 3.2.1. Within this function the reload value is determined as described before. The signal is superimposed to the dc offset and loaded into the DAC output register. The two signals have to be combined in a 32-bit array, like in Figure 3.4. For this, the offset in combination with the amplitude is stored in the first 16 bits. The sinusoidal signal is stored in the last 8 bits. Thus, the entire signal is stored in an array and can be output value by value. This function is shown in Figure 3.21.

```

/* Generation of the complete output signal - DC signal with overlaid sinusoidal signal */
void _signalArrayGeneration(uint32_t *signal_array, uint16_t *raw_input_signal, uint16_t raw_input_signal_length,
    uint16_t stage0_out1, uint16_t stage0_out2)
{
    for (uint16_t signal_pos = 0; signal_pos < raw_input_signal_length; signal_pos++)
    {
        *(signal_array + signal_pos) = (((raw_input_signal[signal_pos] & 0xFF) << 24) | (0x0 & 0xFF) << 16)
        | ((stage0_out2 & 0xFF) << 8) | ((stage0_out1 & 0xFF) << 0));
    }
}

```

Figure 3.21: Signal stored in 32-bit array

Subsequently, the superimposed sinusoidal signal oscillates around the

middle value of 1.040V. This is part of the signal transmission, which is written in assembler code, due to timing issues. In contrast to the majority of the code which is written in C for simpler readability. When starting the sinusoidal signal, the measured values of the ADC are picked up from the result register. This happens as simultaneously as possible, only a few CPU clock cycles are between the applied and measured value.

Three to four periods of the sinusoidal signal are applied in a row for each frequency, because the system needs time to settle. The settling takes one period for high frequencies, thus down to 1kHz there are applied four periods. Data for further processing is only collected during one period in all cases. This period is cut out one period plus the ADC delay shifted backwards from the whole measured signal. Figure 3.22 shows a current response in the high frequency region. It can be noticed that the first period has a smaller amplitude than the following one, which represents the settling. So, the signal used for further processing and calculations will be cut out from sample point 50 to 100.

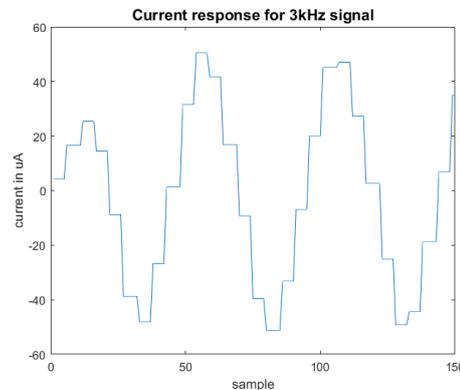


Figure 3.22: Current response of a high frequency signal

This signal is first used to determine the best measuring range, like described in Section 3.1.1. It is adjusted, before a measurement with a new frequency starts, so the range adjustment will not affect the measurement itself. Subsequently, the measurement is done in the best available range and achieves the most accurate results. Figure 3.23 shows this sequence.

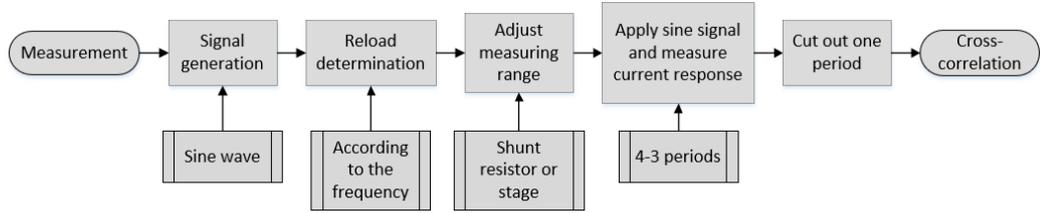


Figure 3.23: Sequence of the implemented measurement

3.5.3 Data analysis

After the whole signal transmission and receiving part, the signal processing starts.

Firstly, the maximum of the measured current is determined, by a simple maximum detection function. This function sweeps through the array and detects the biggest number, of which the position is returned. The maximum voltage is already determined directly after the calibration. This value should not change during the whole measurement procedure and is the same for every frequency.

Thereafter, the operating point bias must be removed from both signals, voltage and current. For this, the mean value of the signals is calculated. The calculated mean value is subtracted from every measured data point to get a signal that is centered around the x-axis. That is necessary to apply the cross-correlation to the signal. The cross-correlation function was implemented on the basis of [33] and can be found in the appendix, Figure A1.

To determine the phase shift, the position of the maximum in the cross-correlated signal is detected. With that position the phase shift is calculated like in the following equation:

$$phi = \frac{corrMaxPos \cdot 360^\circ}{signalLength} \quad (3.5)$$

All necessary values are gathered, so the impedance can be calculated like shown in Equation 2.1. The impedance is divided into real (Equation 3.6), and imaginary part (Equation 3.7). These parameters are transferred

via UART to the PC and finally plotted in Matlab.

$$Re(Z) = \frac{voltageMax}{currentMax} \cdot \cos(phi) \quad (3.6)$$

$$Im(Z) = \frac{voltageMax}{currentMax} \cdot \sin(phi) \quad (3.7)$$

3.6 Evaluation of the implemented parts

The implemented functions have to be evaluated to make sure that they work properly. In case an error occurs, it is easier to find the cause of the problem. Throughout the implemented function, it can be determined if the error is connected to the output signal or the measured one or even emerges from a mistake in the calculations.

3.6.1 Output signal

The generated output signal was evaluated by measuring it with the oscilloscope at the electrode's output. The signal has two characteristics, which were checked in the following. On one hand, the signal form, the dc offset combined with the sinusoidal signal, had to be observed. On the other hand, the frequency had to be checked for correctness.

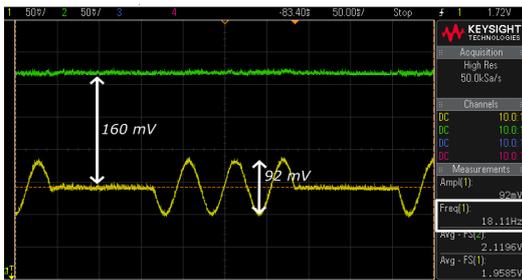


Figure 3.24: Applied signal with 40mV amplitude at 18Hz

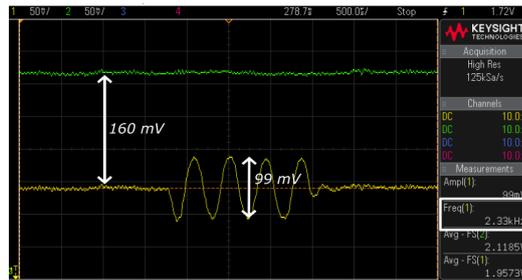


Figure 3.25: Applied signal with 40mV amplitude at 2.33kHz

Figure 3.24 and 3.25 show examples of such an evaluation measurement. The green graph represents the RE/CE and the yellow one the WE. The dc

offset is measured between the green and yellow line and is exactly 160mV. The selected frequencies were compared to the oscilloscope's results, which are outlined in the figures. They are identical to the set ones. But a variation could be found in the peak-to-peak value. The signal was applied with an amplitude of 40mV, which should result in a peak-to-peak of 80mV. However, as marked in Figure 3.24, 92mV were measured. This could occur because of the noise, which cannot be observed in this picture due to the high-resolution mode selected on the oscilloscope. Moreover, the varied number of periods is visualized. Three periods were applied for low frequencies and four periods for higher ones.

3.6.2 Current measurement

The complexity of the signal evaluation increases, due to the fact that it cannot be measured with the oscilloscope directly. Two independent measuring methods are available. The first one sends the measured values via UART to the PC. For further processing or plotting, the values are imported in Matlab. Additionally, the current values can be calculated from the imported digital values. This requires the adjusted shunt resistor and the settings of the ADC. As an example I would like to mention: if the ADC input range is $\pm 1.5 \mu\text{A}$ (FSR is 3.0), the bit width is 15 bits (a factor of 0x4000) and the shunt resistor is 29.66k Ω , the digital value of 10780 corresponds to a sensor current of:

$$currentmA = \frac{\frac{adcFSR}{adcBitwidth} \cdot digitalValue - \frac{adcFSR}{2}}{potRshunt}$$

$$\frac{\frac{3}{0x4000} \cdot 10780 - \frac{3}{2}}{29.66} = 15.7\mu A \quad (3.8)$$

The second method is to measure the voltage drop at an additional resistor in series to the circuit, e.g. resistor parallel capacitor, and calculate the current through that shunt resistor. The whole measurement circuit is shown in Figure 3.26.

Figure 3.27 shows the oscilloscope output at a frequency of 546Hz. The

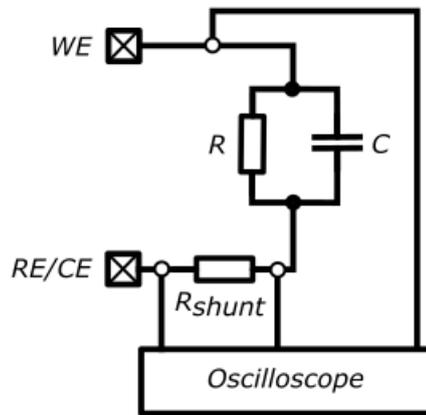


Figure 3.26: Schematic of the circuit to determine the current response

voltage drop was calculated from the green graph, which represents the node at the circuit side, and the blue graph, which is the RE/CE output. The dc offset is 160mV between these two lines. The yellow graph was measured at the WE output. Beside the voltage drop also the phase shift is identifiable, which is highlighted at the right bottom in Figure 3.27.

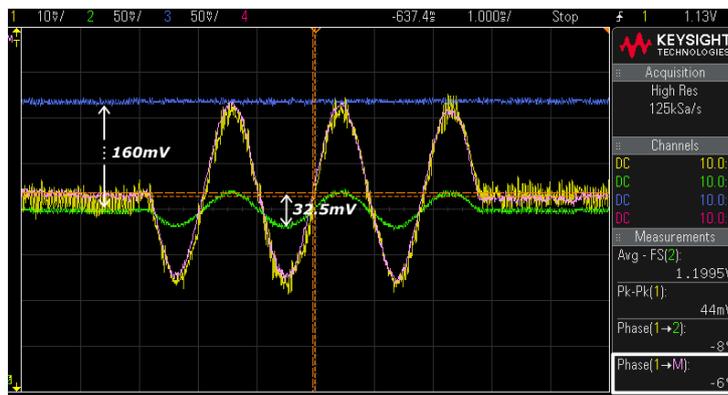


Figure 3.27: Oscilloscope output for current determination

To determine the maximum current value, the peak voltage in the green graph is measured and added to the dc offset. In this example it results in 176.25mV. This is divided by the value of the shunt resistor (10k Ω) and results in 17.6 μ A.

Although, both methods determined the maximum current at the same

frequency, slightly different results are obtained. This could occur because of measurement inaccuracies. The shunt resistor was not measured and was thus ideally accepted with $10\text{k}\Omega$ for the current calculation. The shunt resistor could also influence the behaviour of the circuit.

3.6.3 Signal processing

The entire processing chain, including maximum detection and cross-correlation, was tested and verified with Matlab. First, the whole process was verified with test data. In the next step the raw data was transferred via UART from the test-chip to the PC and processed in Matlab with the available functions. The interim results from the individual functions and the final result, the real and imaginary part of the impedance, was compared to the output of the written C program. The results of the test-chip and Matlab are so similar that only the test-chip signal is visible in Figure 3.28. Thus, the functionality of the code was proven.

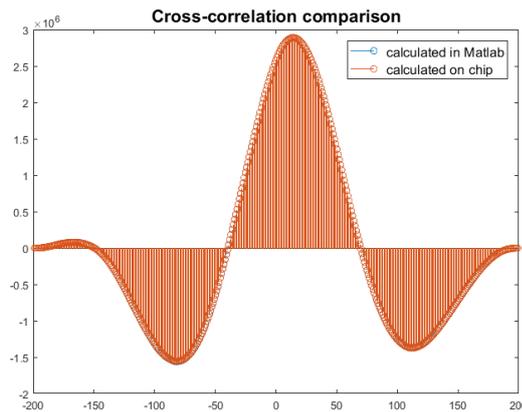


Figure 3.28: Comparison of cross-correlation calculated in Matlab versus on chip

3.7 Test results

The first measurements just included a resistor parallel a capacitor. Furthermore, the impedance was varied by changing resistor and capacitor sizes. Where the capacitances from 470nF to 10 μ F were used. In further consequence additional parts were added and, in the end, an equivalent circuit was used. The impedance results were plotted in Matlab with the negative imaginary part ($-Z''$) over the real part (Z'). During the first measurements a phenomenon was observed, which could not be handled and appeared also in the following measurements. Some torn-out points appeared in the low frequency region. The appearance of this torn-out points could be related to the big capacitances used, since the test-chip was designed for 50nF maximum.

3.7.1 Different resistive and capacitive loads

The first figures, 3.29 and 3.30, show the comparison between before and after the improvements. Figure 3.29 shows the measurement result before the implemented filter and the calibrated shunt resistors, but with calibrated ADC filter delay. Figure 3.30 is based on data after the improvements.

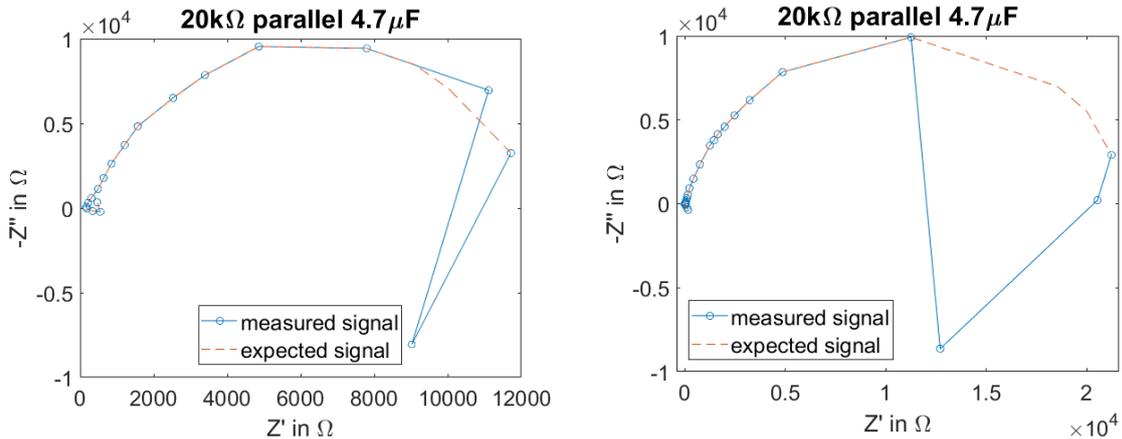


Figure 3.29: Measured impedance before the countermeasures

Figure 3.30: Measured impedance after the countermeasures

The absolute values are better in the second case, nevertheless the heavy torn-out points in the low frequency region occur in both pictures. The

dashed line indicates the expected curve shape. This curve shape corresponds to the ideal mapping of the impedance of the respective electrical circuit in a Nyquist plot. As already shown in Section 2.1.1.2, the expected Nyquist plot of a RC-parallel circuit is a half-circle starting in the origin. Possible reasons for deviations from that expected curve will be discussed in the following Chapter 4.

Beside the torn-out points the measurements taken are reproducible, which shows Figure 3.31. The measurements were taken with a $20\text{k}\Omega$ resistance parallel a $2.2\mu\text{F}$ capacitance. The deviations between expected and measured appear at the same frequencies and with the same magnitude.

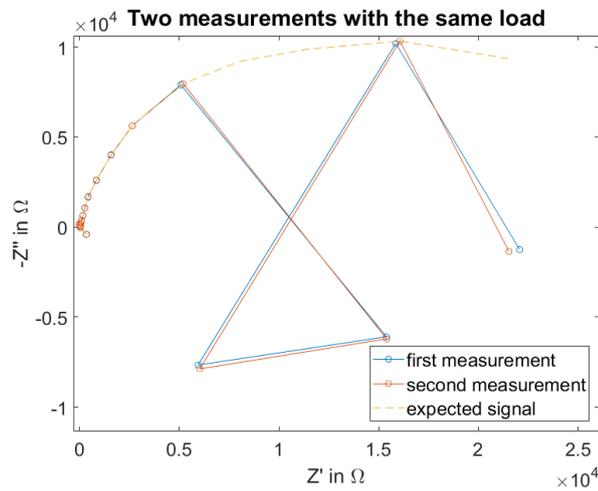


Figure 3.31: Two measurements taken with the same load

The following measurements were taken with the same resistor but different capacitors. Figure 3.32, 3.33 and 3.34 display both, the measured signal and the expected one. It may be observed that the bigger the impedance, the less it looks like a half-circle but more like a quarter of a circle, what was expected.

For the reason of evaluation the next measurements were taken with the potentiostat from PalmSens (PalmSens4) in combination with the according software (PSTrace5). The same settings were chosen as implemented on the test-chip.

The comparison between the results from the test-chip and the poten-

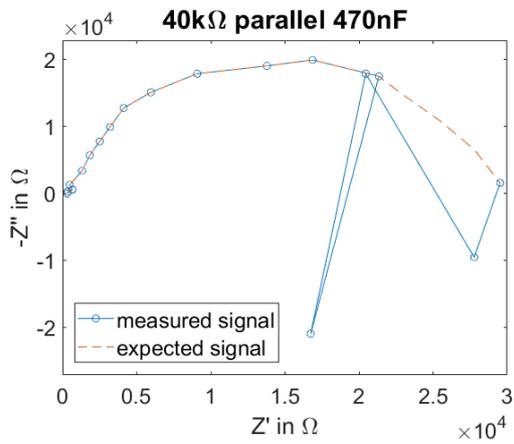


Figure 3.32: Nyquist plot for $40k\Omega$ resistor parallel a $470nF$ capacitor

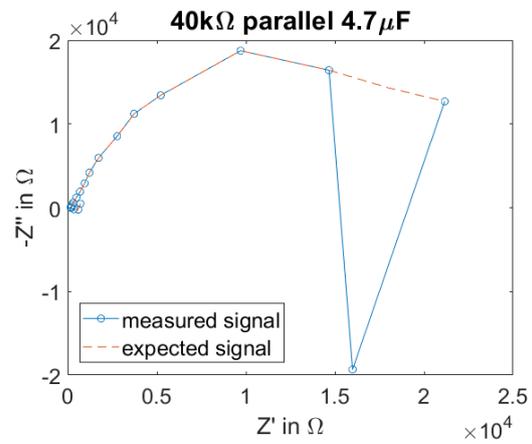


Figure 3.33: Nyquist plot for $40k\Omega$ resistor parallel a $4.7\mu F$ capacitor

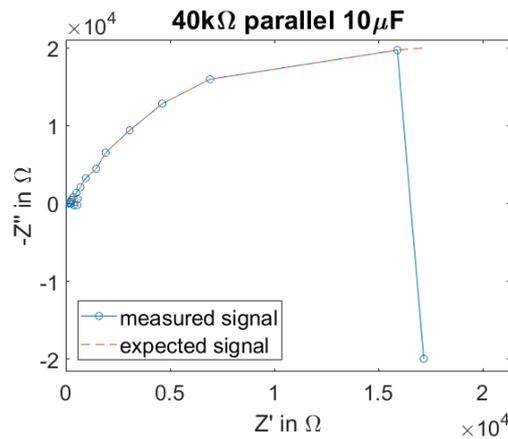


Figure 3.34: Nyquist plot for $40k\Omega$ resistor parallel a $10\mu F$ capacitor

tiostat from PalmSens are shown in Figure 3.35 and 3.36. The results with the higher impedance are more similar to those results with the smaller one, which may depend on the measurement settings of the test-chip. In case the shunt resistor is not chosen as optimal as possible, the measurement is imprecise. The torn-out points do not come from this, but may have a different origin, which will be discussed.

In the next step a resistor was added in series to the used RC-element to come closer to the equivalent circuit. The used circuit is shown in Figure 3.37, where the components are already named like in the Randles circuit.

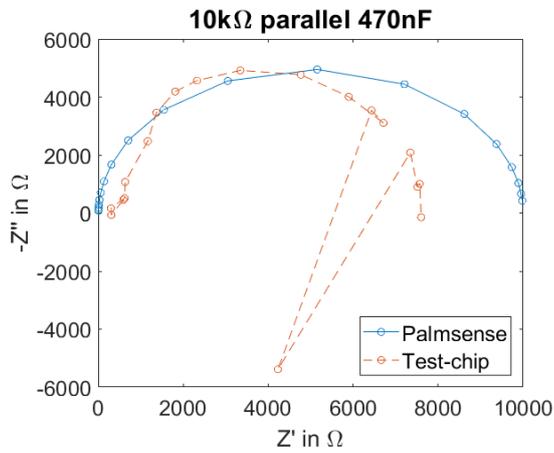


Figure 3.35: Comparison PalmSens and test-chip with a load of $10k\Omega$ resistor parallel a $470nF$ capacitor

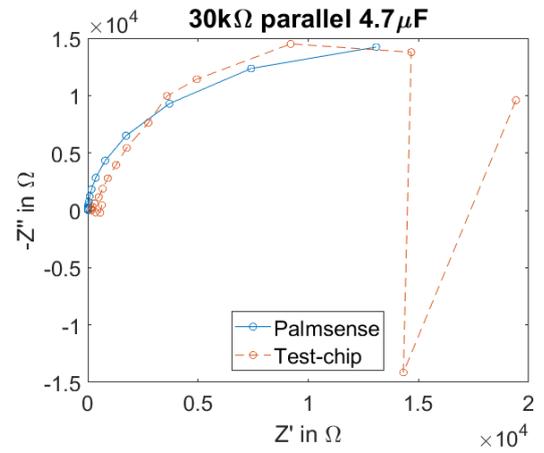


Figure 3.36: Comparison PalmSens and test-chip with a load of $30k\Omega$ resistor parallel a $4.7\mu F$ capacitor

Figure 3.38 illustrates the measurement results with different resistors, which leads to a different size of the semicircle.

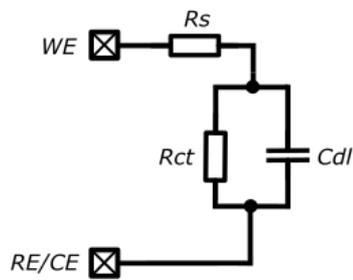


Figure 3.37: 3-component measurement setup

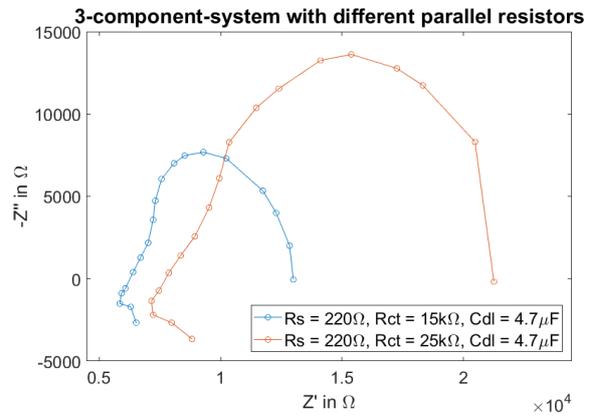


Figure 3.38: Measurement results with 3 components

From Figure 3.38 it can be concluded that the bigger the resistor, which is parallel to the capacitor, the bigger the semicircle gets. The series resistor (R_s) should indicate the shift of the starting point on the x-axis, which should be the same in both measurements and around 200Ω . But in this case, it gets influenced by the R_{ct} variation, what leads to different starting points

on the x-axis.

3.7.2 Equivalent circuit (Randles circuit)

To prepare the measuring circuit for the measurements with the sensor, an equivalent circuit (Randles circuit) was used first. The model, shown in Figure 2.4, had to be modified to be built just by resistors and capacitors. Therefore, the fitting tool of the PalmSens software, PStRace, was used. It was evaluated that four RC-elements in series represent the desired curve shape of the current response best. The component values were chosen to simulate the results from the doctoral thesis [16]. The used circuit including component values is shown in Figure 3.39.

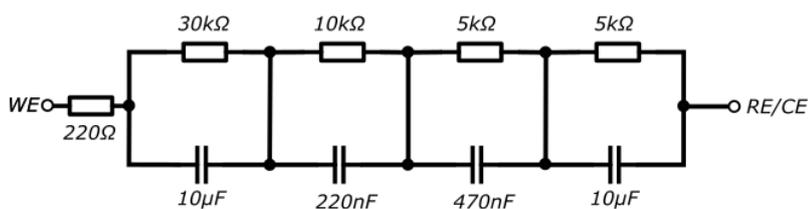


Figure 3.39: Schematic of the recreated equivalent circuit

This circuit was measured with the test-chip and as a reference with the potentiostat from PalmSens. In this process the developed firmware had to prove itself. The equivalent circuit was placed between WE and RE/CE like in the tests before. The results look alike as Figure 3.40 and 3.41 show.

At first there are no major torn-out points visible in the measurement results. At second glance the notch could probably be interpreted as such because it deviates from the PalmSens measurement.

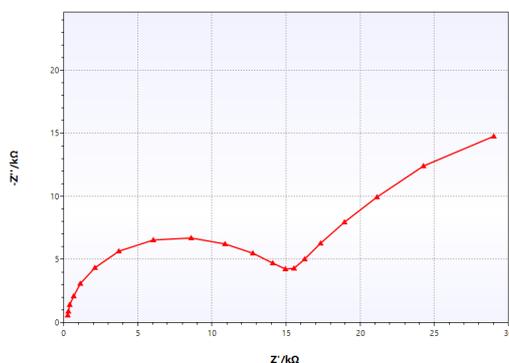


Figure 3.40: Nyquist plot of the equivalent circuit measured with PalmSens

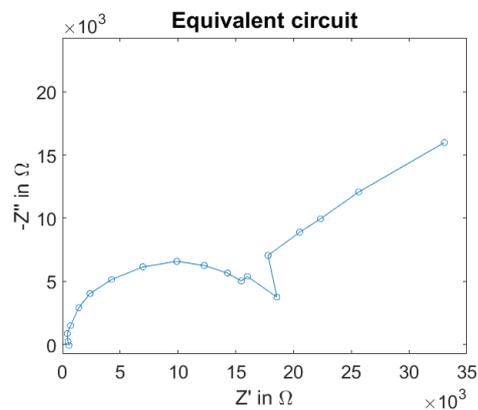


Figure 3.41: Nyquist plot of the equivalent circuit measured with the test-chip

3.8 Sensor measurements

In conclusion, the sensors on PCB were tested with the test-chip and the results are compared to the results from a potentiostat produced by Autolab.

3.8.1 Setup

This biosensor was first tested on PCB in the course of this work. For this purpose, the PCB was tested with the device, which was used by AIT for all previous measurements of the sensor on glass. Thus, the measurements were alternately carried out with Autolab and the test-chip. This allowed the results of the two measurement systems to be compared.

The measurement setup is demonstrated in Figure 3.42. The PCB is located on top of the measuring device. It includes the five measurement chambers, which are separated by sealing rings. The screws make sure that the pressure is the same on every spot of the board. On the left-hand side of the box is the connector to the electrodes, which leads into the box at the upper side. On the bottom side are the electrode connectors to the measurement unit, like in this case the Autolab or the test-chip. There is a connector for each WE, RE and CE. But the RE is connected through all chambers, as it is the CE. So only the WE's connector has to be changed

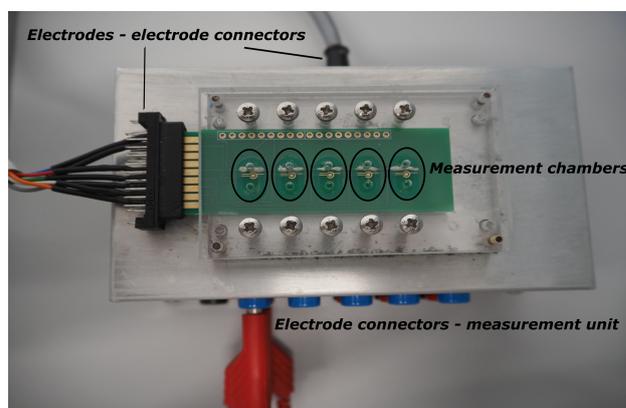


Figure 3.42: Setup for EIS measurement on PCB

to change from one measurement chamber to the next one. The RE and CE can stay the same. In this use case the RE and CE were not short-circuited.

3.8.2 Comparative measurements

The measuring routine included two measurement cycles. The first measurements were the so-called blank measurements, which test the functionality of the sensors. They were followed by the ones with the bounded analyte. The measurement procedure followed the process described in Section 2.2.4. The measurements were taken alternately with the test-chip and the potentiostat from Autolab. Finally, the results are compared.

Blank measurements These measurements show the behaviour of the sensor without any analyte bound. Only the functionalized electrode with the measurement solution are measured. Figure 3.43 shows the results of the blank measurements with the test-chip for all five measurement chambers. The different chambers provide similar results, only 5% deviation in the range from 2.3kHz to 10Hz. But the imaginary part of chamber 1 at 3.8kHz deviates 30% from the middle value, which is the biggest outlier. The results in the low frequency region deviate not that strongly from the middle value, only about 10%. Furthermore, the results from each chamber are compared to the Autolab results. The comparison for chamber 2 is shown in Figure

3.44. There are big differences in the high frequency region, but below 1kHz the deviation from the test-chip results to the results from Autolab are less than 5%. The typical curve shape, like in Figure 3.41, can be seen a little.

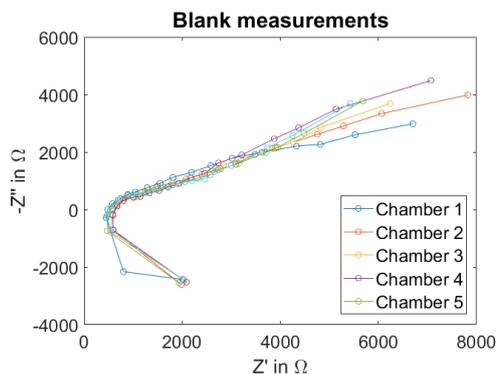


Figure 3.43: Blank measurement of sensors on PCB with test-chip

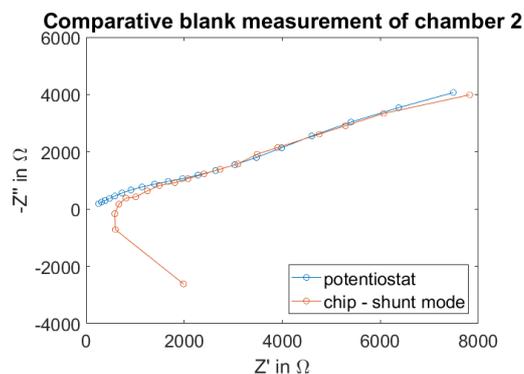


Figure 3.44: Comparison of blank measurement with test-chip and Autolab (potentiostat)

Analyte measurements In the second turn, the measurements were performed with a different analyte concentration in each chamber. For the individual concentrations see Section 2.2.4. First, each chamber was measured once with the potentiostat from Autolab and afterwards with the test-chip. The test-chip measurements were taken in both modes, the current mirror mode and the shunt mode. The results for two different concentrations are shown in Figure 3.45.

Comparing the results from the test-chip and Autolab, they are very similar. Especially the impedances between 800Hz and 4Hz deviate only 5%. In the low frequency region is a deviation visible, more pronounced in the current mirror mode with a deviation of up to 20%. That was expected due to a matching error [34]. The shunt mode measurement points in the high frequency region deviate more from the other two measurement methods, which is shown in Figure 3.45. A reason for this behaviour could be the relatively high current flowing because of the low impedance of the sensor.

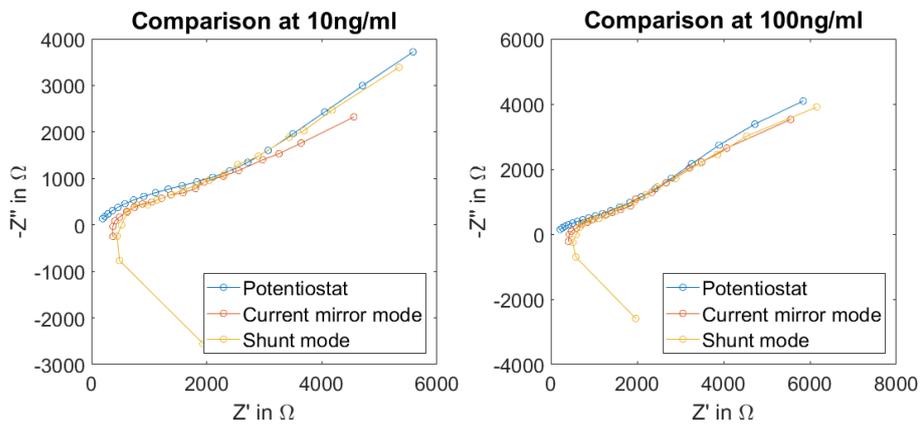


Figure 3.45: Comparative measurements of the sensors on PCB with test-chip and Autolab (potentiostat)

Chapter 4

Discussion

This section describes and discusses decisions and challenges faced during implementation and evaluation phase. But also, the results of the measurements are considered.

4.1 Implemented method

The first concept of the implemented method was created, while the test-chip was still in design and production phase, see Figure 2.8. This concept was a representation of the ideal process, which was adjusted after the first evaluation steps. However, the basic structure remained the same, although the initialization part was added, also the calibration function and a filter implementation. The adjustments were made to further improve the results of the measurement.

The added calibration function first used the same sinusoidal signal as the measurement, but then got reduced to a simple step function. It is a leaner function that requires less resources. Since the calibration function was used to determine the delay of the ADC filter, a certain time delay has to be detected. The reason for the delay is described in Section 4.2.2. It is not necessary to apply a sinusoidal signal, but a dc offset is sufficient in this case. Because this is determined before the start of the measurement, the time delay for every frequency is stored in an array for future use. These

values have to be evaluated individually for each chip.

Initially, the idea was to generate the frequencies dynamically. But this does not guarantee that the same frequencies are used in the measurements and in consequence the results are not reproducible. Thus, a fixed array is used to obtain reproducible data.

To determine the impedance at the respective frequency points, the phase shift is needed. Two possibilities were considered to determine the phase shift between two sinusoidal signals: the cross-correlation function and the Fourier Transformation. Both can be evaluated with Matlab using the Matlab functions, `xcorr()` [26] or `fft()` [35]. The cross-correlation function was chosen because the Fourier Transformation would have been more complex to implement in C code. The only disadvantage of the cross-correlation is the large amount of memory required. This was solved with additional memory that was activated. The test-chip has three times 4kB RAM. Whereby each memory block must be activated separately to save unnecessary energy when not in use. Due to the cross-correlation and other structures like arrays the required memory is high. So, the whole 12kB RAM is activated in this application.

The detection of the maximum of the applied and measured signal as well as the impedance calculations were already tested in Matlab with test data. Both functions were taken over to the C program and worked fine from the beginning. Only the order of the individual functions was changed during the implementation phase. The maximum detection in the applied signal is shifted to the beginning because it does not change during the whole measurement. Where the maximum detection in the measured signal is placed after the signal acquisition.

4.2 Adaptations to improve measurement

The implemented method was modified to obtain better measurement results. The first step was to overcome the restriction in the high frequency region. Furthermore, the filter stages of the ADC are taken into account and a second moving average filter was implemented to improve the results at

low frequencies. In the course of this chapter the potential causes for the torn-out points are also discussed.

4.2.1 Additional implementation tools

In the exploration of the possibilities to overcome the high frequency restriction two more possibilities besides assembler code were tested. They were tested to improve the execution of time critical parts.

Interrupt An interrupt occurs whenever a timer reaches a certain value. The interrupt calls the so called interrupt service routine (ISR), which contains the part to execute. After the execution of the ISR the program returns to the main part, where it was interrupted.

An interrupt can be generated in three different ways. One way is to interrupt by an external hardware, which is called external or hardware interrupt. Another form of the interrupt is the software interrupt, which is triggered by software. Or the program is interrupted by an internal event, which is called an internal interrupt. The last one occurs whenever an exception or abnormal condition occurs. [36]

The software interrupt was tested in this application. But the interrupt latency was too long, which is the time between the generation of an interrupt request and the start of the ISR [37]. It takes 14 cycles which is too long for this task and thus not usable in this application.

DMA controller Here is no interrupt needed but the DMA controller shifts data from external/internal to external/internal memory or from one register to another while the central processing unit (CPU) is handling its basic processes [38]. When the core comes to the position in the program where the data is needed, it is already there for further processing. Also, an interrupt could indicate the presence of the data and makes the program jumping there [38].

The downside, one DMA controller can only handle one action. To be applicable for this scenario two DMA controllers are needed, one for the

output and one for the input. But the test-chip has only one, thus the DMA controller is not functional for this application. Although it is almost ten times faster than the interrupt.

Therefore the data transmission is written in assembler code. It is the tool with the most precise timing.

4.2.2 Calibration function

The calibration function was added after the first evaluation steps, whereby the time delay was detected. The origin of the time delay at the beginning of each measurement is the structure of the ADC filter. So, it is described in the following section, for better understanding.

4.2.2.1 ADC filter structure

As already mentioned in Section 3.4.2, the filter decimates the measured data points by a factor of 64. The output of the ADC has a frequency of 37.5kHz. This frequency results from the ADC clock frequency (2.4MHz) and a series of filter stages. The structure of these filter stages are a series of two Cascaded-Integrator-Comb (CIC) decimation filters. The structure of a simple decimation filter is shown in Figure 4.1.

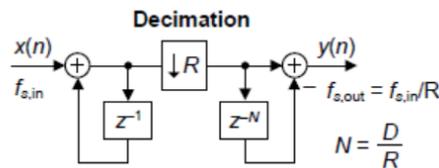


Figure 4.1: CIC decimation filter structure [39]

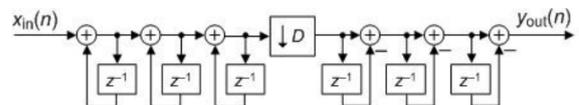


Figure 4.2: Structure of a three-stage CIC decimation filter [40]

The CIC decimation filter consists of two stages. The feed forward loop, also called comb section and the feedback loop, called an integrator. N in Figure 4.1 is typically one or two for high sample rate ratios. [39]

In this application a three-stage decimation CIC filter is used, which is shown in Figure 4.2. $\downarrow D$ represents the down sampling by an integer D , where

all samples are discarded except for the D th sample [40]. Two of these filters are in series, whereby the first CIC filter has a D of 32 and the second one has a D of 2. Together an appropriate anti-aliasing property for the desired frequency of 10kHz is achieved. With an ADC clock frequency of 2.4MHz and the following filter stages, an output frequency of 37.5kHz is generated.

The output frequency leads to the delay in the measured signal. The delay is more prominent at higher frequencies because the system has to settle down first. At frequencies below 1kHz possibly only one old value is picked up instead of a new one. That happens if the value in the register has not changed but it is already read out. The delay for each signal frequency, due to the filter stages, could be calculated. But in this application, it is easier to measure it. For this, the calibration function was added at the beginning of the measurement.

4.2.3 Moving average filter

This additional moving average filter was added to get rid of the noise in the low frequency region. The data points are further updated with a frequency of approximately 37.5kHz in the ADC register. But not every value is fetched, some are skipped, which leads to an obviously noisy output signal.

Another contribution to noise is the applied load. The measured signal is noisier with a capacitive component involved. The added noise could either occur because of interferences with components on the test-chip or just because of the passive component itself. The constant charging and discharging of the capacitor can cause oscillations. Thus, this filter is primarily used with capacitive loads and decreasing frequencies. The added filter averages over more values if the signal gets noisier, so at lower frequencies, as described in Section 3.4.3. The figure for the number of values to be averaged was determined by means of testing.

4.3 Results and their accuracy

The first results with a higher amplitude than requested looked very promising, but under the given specifications the results were no longer as accurate. However, the system can impress with reproducibility. Every measurement that was repeated had very similar or nearly identical results. There were some improprieties observed like the torn-out points in the Nyquist plots in Section 3.7, which came along with phase shifted and amplitude varied current responses. This happens due to sensor loads that were outside the specifications of the used test-chip. This loads were used to approximate the behaviour of the sensor.

4.3.1 Noise overlay

The accuracy decreased with decreasing amplitude of the sinusoidal signal. This can be explained with noise overlay, which ultimately obscures the signal. Figure 4.3 and 4.4 show the applied signal with an amplitude of 10mV measured with an oscilloscope, the green line represents the signal at the RE/CE and the yellow one at the WE. The noise at the WE is even half of the amplitude of the signal.

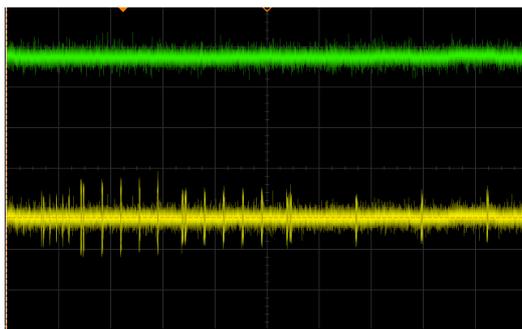


Figure 4.3: Applied signal with 10mV amplitude and noise overlay

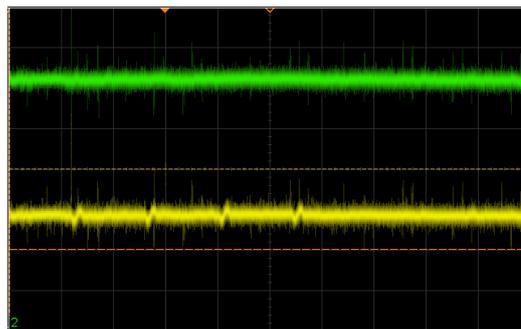


Figure 4.4: Applied signal with 10mV amplitude at 470Hz

The measured current response with a smaller amplitude is distorted even more by the noise. This leads to an error in the final calculation because the applied signal cannot be measured and has to be assumed to be ideal. In

contrast to the measured one, which is influenced by the noise.

Another aspect could be the set measuring range. As described in Section 3.5.2 there are different shunt resistors in the shunt mode and stages in the current mirror mode to best fit to the measurement region. If the system measures in the lower region of each stage or resistor, it is more error prone, because of the bigger error due to the smaller signal.

4.3.2 Torn-out points

The torn-out points result from a wrongly measured current response, which is shown in Figure 4.5. The current response at 2Hz (yellow graph) is phase shifted and has a lower amplitude than the others. It should be between the orange (3Hz) and the violet (1Hz) graph. Whenever this phenomenon is observed, the calculated impedance at that frequency is wrong and results in a torn-out point in the Nyquist plot.

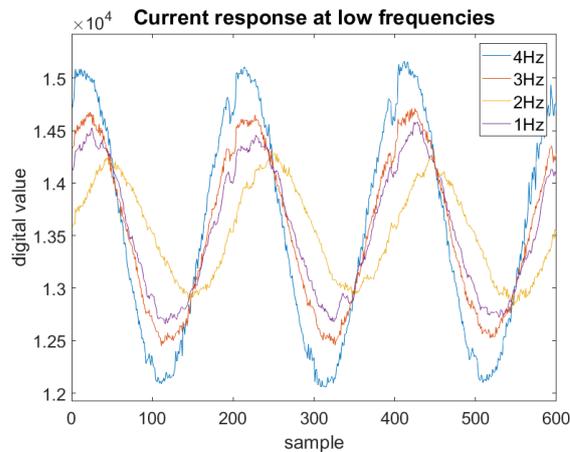


Figure 4.5: Current response in low frequency region

This behaviour was analysed using the 1-bit stream at the input of the ADC filter. The stream of zeros and ones was converted into a continuous signal with an external low pass filter. The bit stream was output via a GPIO pin and low pass filtered to measure it with the oscilloscope, this is shown in Figure 4.6. The left side represents the measurement circuit between the electrodes and the right circuit the bit stream recording. In between would

be the test-chip with its outputs.

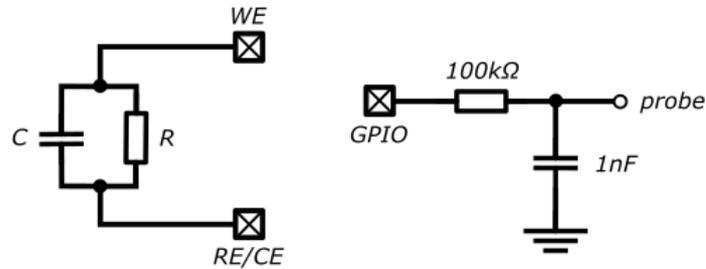


Figure 4.6: Schematic of the circuit for bit stream representation

The following Figures 4.7 and 4.8 were captured with the configuration shown in Figure 4.6. The blue graph represents the filtered bit stream, the green one the applied voltage at the WE and the yellow one the applied voltage at the RE/CE. The pink line is the extracted difference between the potentials at the electrodes to show the phase shift between the applied sinusoidal signal (green) and the measured signal (blue).

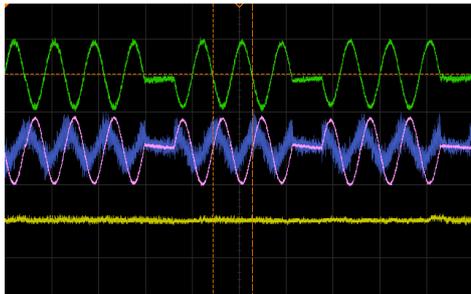


Figure 4.7: Representation of 1-bit stream

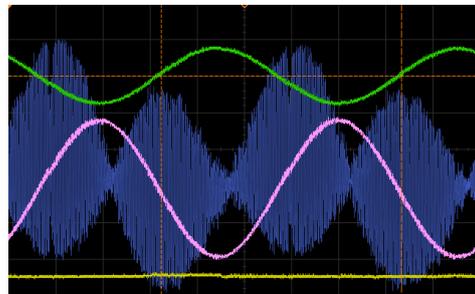


Figure 4.8: Representation of 1-bit stream with oscillation

Figure 4.7 shows the desired signal, in contrast to Figure 4.8, which reflects some kind of oscillation. This oscillation may be the cause for the incorrectly measured signal due to the big loads beyond test-chip specifications. The frequency compensation of the potentiostat has to be chosen appropriately depending on the load condition. With proper configuration of the hardware, the torn-out points can be removed. Since this is not in the

scope of this work, it was not further investigated. But two measurements with different frequency compensation settings are depicted in Figure 4.9.

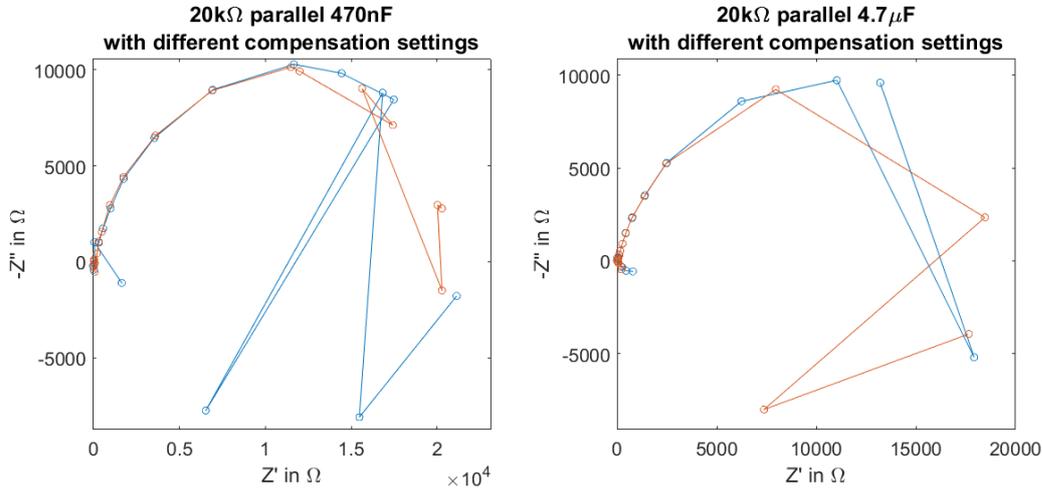


Figure 4.9: Impedance plots with different frequency compensation elements

Also, different loads between the electrodes led to different torn-out points in the Nyquist plot. This could already be observed in the results in Section 3.7. This leads to the assumption that the frequency compensation parts interact with the applied load between the electrodes. However, this phenomenon did not occur in the simulation and has to be investigated in more detail to be understood. This behaviour is important for the accuracy of the determination of the analyte concentration, since the significance is in the low frequency region. If this measuring system should come to execution, this should be solved.

4.3.3 Low frequency region

An additional reason for the problems in the low frequency region could be the decreasing current in this frequency range. Figure 4.10 shows the current responses for all the different frequencies measured with the equivalent circuit. It can be observed that the current below 100Hz is nearly a hundred times smaller than in the kHz-region. Furthermore, the signal is noisier, which makes it even more prone to errors.

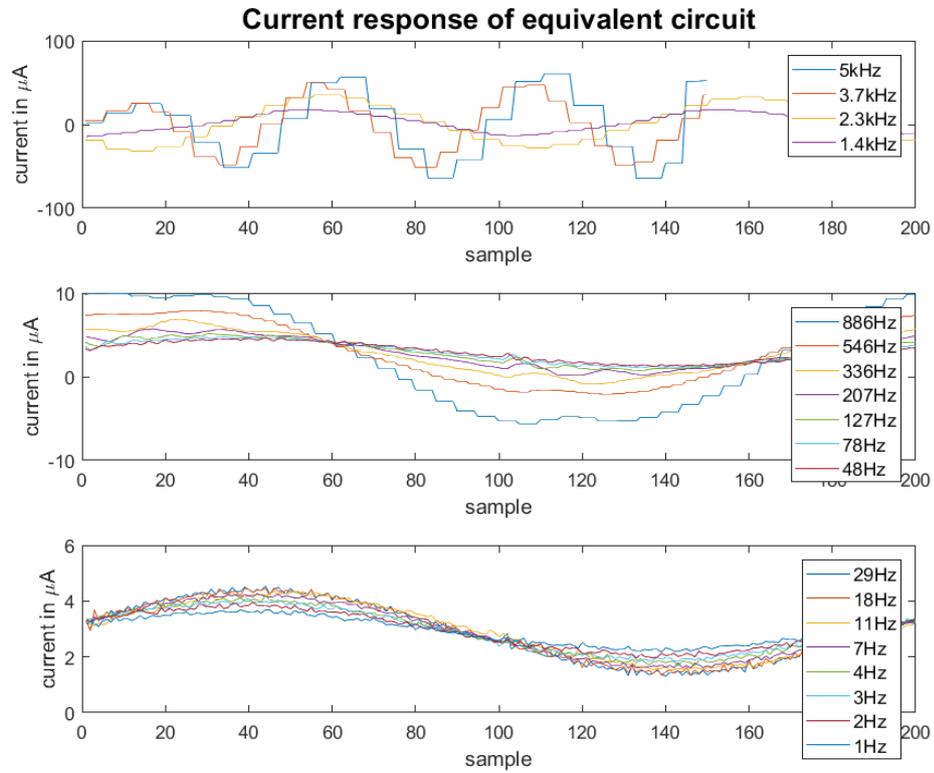


Figure 4.10: Current response for different frequencies

In addition to the decreasing amplitude, the signal shape can be seen in Figure 4.10 for the different frequencies. The difference between the high frequencies to the lower ones is immense. At frequencies above 800Hz steps are visible in the signal, which results from a decreasing number of sampling points for increasing signal frequencies. The same value is picked up from the register several times before it changes. A solution might be to only picking up a value whenever a new one is written into the register.

An additional phenomenon is visible in Figure 4.5, the recess just before the peak, which appears most likely in the low frequency region. This comes from a short unaltered applied signal, which is shown in Figure 4.11. It always happens around the peak points of the current signal, where the applied voltage has its turning points. This could be another reason for inaccurate results and is solved with the implemented moving average filter. The filter smooths the recess but does not make it disappear if it is too

pronounced.

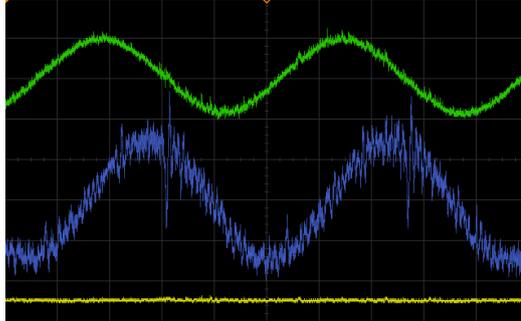


Figure 4.11: Representation of current recession

4.3.4 Comparative measurements

There were two additional potentiostats used other than the test-chip, to verify the results. For measurements of the RC-elements or the equivalent circuit the potentiostat from PalmSens was used. Comparing the results from PalmSens to the results gained with the test-chip, they describe the same curve shape, except for the torn-out points. However, the impedance values differ from each other. The difference is bigger with a smaller load. But the test-chip is just 4x4mm and the dimensions of the housing of the PalmSens are 15.7x9.7x3.5mm [41]. Not only the housing is bigger, also the computing power. A possible reason for the behaviour of the test-chip could be the wrongly chosen shunt resistor or a phenomenon described in the sections above.

The Nyquist plots of the measured sensors look almost the same for the test-chip and the potentiostat from Autolab, compare to Figure 3.45. But the Autolab [42] is also bigger, with more computing power and better resolution.

The low impedance of the sensor comes on the one hand from the low number of pinholes, which occur in the SAM [19]. This makes the charge transfer resistance small. And on the other hand, it comes from a favourable electron transfer between the electrolyte and the electrode.

The low impedance implies good conductivity, which is disadvantageous for the shunt mode, especially in the high frequency region. With lower

frequencies the results get better. That may be again a problem with the selected shunt resistor. On the other hand, the current mirror mode provides good results, also at low impedance. So, the measurement method has to be adjusted due to the load to ensure the system works properly.

Chapter 5

Conclusion

The electrochemical measurement method for the application, which includes the test-chip developed by Infineon Technologies Austria AG and the sensor designed by the AIT, was implemented and successfully tested. The measurement method, the electrochemical impedance spectroscopy, was assumed by the sensor. In addition, there were certain requirements, like the dc offset and the sinusoidal signal, to be met. The implementation itself was adapted to the test-chip.

The concept presented at the beginning of the work was adjusted due to improvements of the measurement method. The measurement procedure stated at the beginning was extended by initialization steps and further functions. One of them was the calibration before the actual measurement to integrate the ADC filter delay. Another one was the implementation of a moving average filter due to the noisy signal in the low frequency region. By means of these improvements, suitable results were achieved.

The individual steps were evaluated and, in the end, the entire system was tested, before the sensor was measured. The results are reproducible and similar to the comparative measurements of the PalmSens. Except the torn-out points, which presumably occur because of the large load outside the specification limits of the test-chip. But these large loads are a precursor to the equivalent circuit and subsequently the sensor. Therefore, they were nevertheless used for the tests. The measurements with the equivalent circuit

were the preliminary stage to the sensor measurements, which were made at the final stage of the work at the AIT in Vienna. The sensor measurements were taken with the gold electrodes on PCB. The results from the test-chip are very close to the ones taken with the Autolab potentiostat. This proves the functionality of the implemented measurement method in combination with the test-chip and the sensor.

This low-power sensor system is designed to perform an EIS measurement. It was shown that the results of this system are comparable to the results of conventional potentiostats, which are bigger and more powerful.

In further consequence, it could be used to measure the concentration of specific biomolecules like cortisol in saliva to monitor the health status. A possible setup for this use case is presented in the following chapter. It includes a corresponding circuit board design and the according measuring sequence.

Chapter 6

Future work

There is the possibility to create a demonstrator with the test-chip and the electrode design from the AIT [16]. It could work as a smart application in combination with an NFC-capable smartphone. The NFC-capable smartphone would supply the circuit with energy and does the post processing including visualization of the measurement results.

The possible measurement cycle would be kept simpler than in the measurements taken in this work, see Figure 2.11, and would look like shown in Figure 6.1. The sensor would be taken out of the fridge, where it can be stored several days. The measurement procedure would start by applying the test solution, for instance saliva diluted with water. After 30 minutes of incubation the surface has to be rinsed, which can already be done with the measurement solution. The last step before the measurement would be that the sensor is put into the measurement solution and power the circuit with the smartphone. Then the initialization would take place followed by the EIS measurement as shown in Figure 3.18. The calibration could be done prior to delivery, where the array with the calibration values is stored on chip.

The measurement results could be used for further medical examination. As mentioned in the beginning this sensor was developed to track the stress level. This is done by measuring the quantity of IgG or cortisol in saliva. The notch of the impedance plot, the so-called charge transfer resistor, is tracked for this purpose. The results, either the plots of the impedance or

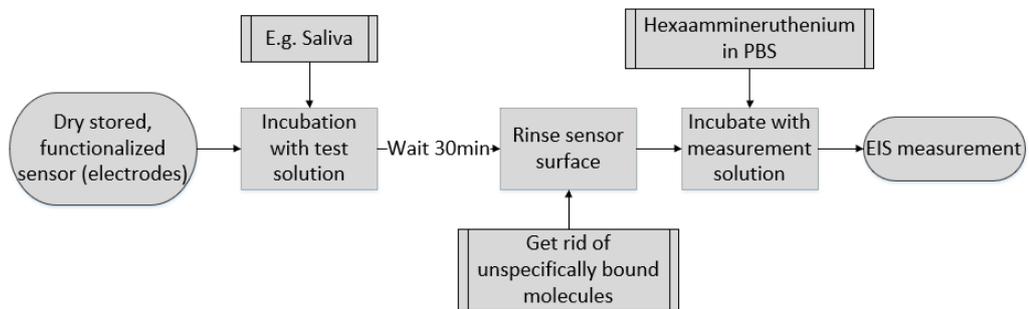


Figure 6.1: Representation of a possible use case

the concentration of the investigated analytes, could be displayed on the smartphone in the end. It could also act as a warning system, which alert if a certain value of analyte is reached or overstepped.

For this kind of application, a different circuit board compared to the tests done in this work would be needed. The idea is to put it in a cup together with the measurement solution. So, everyone is able to do it on her/his own at home. The routine would just start by putting the board into the cup and fill in the test solution and proceed like shown in Figure 6.1. For the power supply an NFC-capable smartphone will be held to the bottom of the cup, where the board will be located in the cup. That is pictured in Figure 6.2.

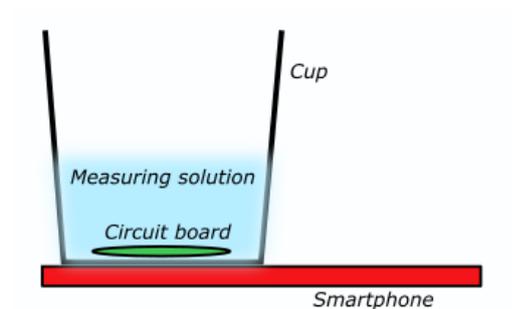


Figure 6.2: Depiction of the potential measurement setup

6.1 PCB design

The following figures show a potential design of a demonstrator, which has the shape of a round tile. Figure 6.3 shows the top of the board with the test-chip, some supporting structures and the NFC antenna at the edge surrounding the circuit. Figure 6.4 demonstrates the bottom, where the electrodes are located and the NFC antenna is also visible. The electrodes have the same dimensions and material as in the PCB design described in Section 2.2.2. There are two electrodes to align the results and make them more accurate.

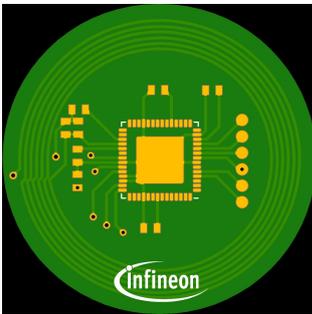


Figure 6.3: Front of a possible demo design

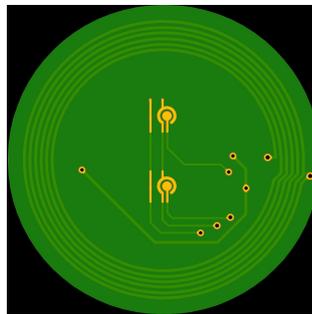


Figure 6.4: Back of a possible demo design

This application could help people monitor and improve health, especially related to burnout.

Bibliography

- [1] A. Poghosian, H. Geissler, and M. J. Schöning, “Rapid methods and sensors for milk quality monitoring and spoilage detection,” *Biosensors and Bioelectronics*, vol. 140, p. 111272, 2019.
- [2] S. Jiang and Y. Liu, “Gas sensors for volatile compounds analysis in muscle foods: A review,” *TrAC Trends in Analytical Chemistry*, vol. 126, p. 115877, 2020.
- [3] Z.-H. Chen, Q.-X. Fan, X.-Y. Han, G. Shi, and M. Zhang, “Design of smart chemical ‘tongue’ sensor arrays for pattern-recognition-based biochemical sensing applications,” *TrAC Trends in Analytical Chemistry*, vol. 124, p. 115794, 2020.
- [4] P. K. Kundu and M. Kundu, “A comparative study on tea samples classification by e-tongue using cross correlation based pca, pca and shamon’s nonlinear mapping method,” in *2012 7th International Conference on Electrical and Computer Engineering*, 2012, pp. 490–493.
- [5] Y. Jiang, X. Yang, P. Liang, P. Liu, and X. Huang, “Microbial fuel cell sensors for water quality early warning systems: Fundamentals, signal resolution, optimization and future challenges,” *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 292 – 305, 2018.
- [6] T. Arakawa, T. Aota, K. Iitani, K. Toma, Y. Iwasaki, and K. Mitsubayashi, “Skin ethanol gas measurement system with a biochemical gas sensor and gas concentrator toward monitoring of blood volatile compounds,” *Talanta*, vol. 219, p. 121187, 2020.

- [7] P. Pirovano, M. Dorrian, A. Shinde, A. Donohoe, A. J. Brady, N. M. Moyna, G. Wallace, D. Diamond, and M. McCaul, “A wearable sensor for the detection of sodium and potassium in human sweat during exercise,” *Talanta*, vol. 219, p. 121145, 2020.
- [8] D. R. Thevenot, K. Tóth, R. A. Durst, and G. S. Wilson, “Electrochemical biosensors: Recommended definitions and classification,” *Pure and Applied Chemistry*, vol. 71, no. 12, pp. 2333 – 2348, 1999.
- [9] L. Zipser, “Selectivity of sensor systems,” *Sensors and Actuators A: Physical*, vol. 37-38, pp. 286 – 289, 1993, proceedings of Eurosensors VI.
- [10] D. Xu, X. Huang, J. Guo, and X. Ma, “Automatic smartphone-based microfluidic biosensor system at the point of care,” *Biosensors and Bioelectronics*, vol. 110, pp. 78–88, Jul. 2018.
- [11] A. C. Sun, C. Yao, V. A.G., and D. A. Hall, “An efficient power harvesting mobile phone-based electrochemical biosensor for point-of-care health monitoring,” *Sensors and Actuators B: Chemical*, vol. 235, pp. 126 – 135, 2016.
- [12] M. Boffadossi, A. Di Tocco, G. Lassabe, M. Pirez-Schirmer, S. N. Robledo, H. Fernández, M. A. Zon, G. González-Sapienza, and F. J. Arévalo, “Development of an impedimetric immunosensor to determine microcystin-LR. New approaches in the use of the electrochemical impedance spectroscopy was used in determining to determine kinetic parameters of immunoreactions,” *Electrochimica Acta*, vol. 353, p. 136621, Sep. 2020.
- [13] J. Gu, H. Yao, K. Wang, B. Parviz, and B. Otis, “A 10 μ A on-chip electrochemical impedance spectroscopy system for wearables/implantables,” in *2014 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov. 2014, pp. 309–312.
- [14] B. Berger, K. Kurabayashi, and M. Nasir, “Investigation into the use of electrochemical impedance spectroscopy for cellular functional im-

- munophenotyping,” in *2016 IEEE SENSORS*, Oct. 2016, pp. 1–3, iSSN: null.
- [15] L. Barhoumi, A. Baraket, F. G. Bellagambi, G. S. Karanasiou, M. B. Ali, D. I. Fotiadis, J. Bausells, N. Zine, M. Sigaud, and A. Errachid, “A novel chronoamperometric immunosensor for rapid detection of TNF- α in human saliva,” *Sensors and Actuators B: Chemical*, vol. 266, pp. 477–484, Aug. 2018.
- [16] J. D. Schrattenecker, “Electrochemical detection of fitness-relevant biomolecules in human saliva,” Ph.D. dissertation, University of Technology Wien, 2020.
- [17] M. Grossi and B. Riccò, “Electrical impedance spectroscopy (EIS) for biological analysis and food characterization: a review,” *Journal of Sensors and Sensor Systems*, vol. 6, no. 2, pp. 303–325, Aug. 2017.
- [18] P. M. Levine, P. Gong, R. Levicky, and K. L. Shepard, “Active cmos sensor array for electrochemical biomolecular detection,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 8, pp. 1859–1871, 2008.
- [19] A. Lasia, *Electrochemical Impedance Spectroscopy and its Applications*. Springer-Verlag, 2014.
- [20] Y. Chen, W. Ji, K. Yan, J. Gao, and J. Zhang, “Fuel cell-based self-powered electrochemical sensors for biochemical detection,” *Nano Energy*, vol. 61, pp. 173 – 193, 2019.
- [21] W. Clarke and P. D’Orazio, “Chapter 9 - electrochemistry,” in *Contemporary Practice in Clinical Chemistry (Fourth Edition)*, fourth edition ed., W. Clarke and M. A. Marzinke, Eds. Academic Press, 2020, pp. 159 – 170.
- [22] T. Kim, J. Kang, J.-H. Lee, and J. Yoon, “Influence of attached bacteria and biofilm on double-layer capacitance during biofilm monitoring by electrochemical impedance spectroscopy,” *Water Research*, vol. 45, no. 15, pp. 4615 – 4622, 2011.

- [23] S. Fujishiro, H. Ohnuki, H. Wu, D. Tsuya, and H. Endo, “Regenerable myoglobin biosensor based on protein g immobilized on interdigitated electrodes,” *Japanese Journal of Applied Physics*, vol. 59, no. SC, p. SCCA05, nov 2019.
- [24] G. Instruments. (2020) Basics of electrochemical impedance spectroscopy. [Online]. Available: <https://www.gamry.com/application-notes/EIS/basics-of-electrochemical-impedance-spectroscopy/>
- [25] J. D. Schrattenecker, R. Heer, E. Melnik, T. Maier, G. Faflek, and R. Hainberger, “Hexaammineruthenium (II)/(III) as alternative redox-probe to Hexacyanoferrat (II)/(III) for stable impedimetric biosensing with gold electrodes,” *Biosensors and Bioelectronics*, vol. 127, pp. 25–30, Feb. 2019.
- [26] MathWorks. (2021) Cross-correlation. (Date last accessed 26.01.2021). [Online]. Available: <https://de.mathworks.com/help/matlab/ref/xcorr.html>
- [27] ——. (2021) max. (Date last accessed 26.01.2021). [Online]. Available: <https://de.mathworks.com/help/matlab/ref/max.html>
- [28] N. Sailer, “A floating differential digital to analog converter for an integrated electrochemical measurement system,” Master’s thesis, University of Technology Graz, 2020.
- [29] M. Haberler, I. Siegl, C. Steffan, and M. Auer, “Mismatch reduction techniques for current-mirror based potentiostats,” in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2020, pp. 99–103.
- [30] T. Wilmshurst, “Chapter 4 - starting to program – an introduction to assembler,” in *Designing Embedded Systems with PIC Microcontrollers (Second Edition)*, second edition ed., T. Wilmshurst, Ed. Boston: Newnes, 2010, pp. 75 – 106.

- [31] C. Arbone, B. Ditu, S. Craciun, and D. Badea, “Model-driven inline assembler generator for retargetable compilers,” in *2013 19th International Conference on Control Systems and Computer Science*, 2013, pp. 71–76.
- [32] B. B. Rao. (2020) Inline assembly for x86 in linux. IBM. (Date last accessed 26.01.2021). [Online]. Available: <http://oldlinux.org/Linux.old/docs/asm-inline/asm-inline.pdf>
- [33] “TIME-DOMAIN CROSS-CORRELATION FUNCTION - File Exchange - MATLAB Central,” (Date last accessed 25.01.2021). [Online]. Available: <https://de.mathworks.com/matlabcentral/fileexchange/63337>
- [34] I. Siegl, M. Haberler, and C. Steffan, “Matching Considerations for Bidirectional Current Mirrors,” in *2019 Austrochip Workshop on Microelectronics (Austrochip)*, Oct. 2019, pp. 65–70.
- [35] MathWorks. (2021) fft. (Date last accessed 29.01.2021). [Online]. Available: <https://de.mathworks.com/help/matlab/ref/fft.html>
- [36] X. Fan, “Chapter 4 - interrupts,” in *Real-Time Embedded Systems*, X. Fan, Ed. Oxford: Newnes, 2015, pp. 85 – 117.
- [37] D. Ibrahim, “Chapter 18 - interrupts in cortex-m4 based microcontrollers,” in *Arm-Based Microcontroller Multitasking Projects*, D. Ibrahim, Ed. Newnes, 2021, pp. 347 – 369.
- [38] D. J. Katz and R. Gentile, “Chapter 3 - direct memory access,” in *Embedded Media Processing*, ser. Embedded Technology, D. J. Katz and R. Gentile, Eds. Burlington: Newnes, 2006, pp. 75 – 101.
- [39] R. Lyons. (2020) A beginner’s guide to cascaded integrator-comb (cic) filters. (Date last accessed 23.10.2020). [Online]. Available: <https://www.dsprelated.com/showarticle/1337.php>

- [40] ——. (2018) Two easy ways to test multistage cic decimation filters. (Date last accessed 26.12.2020). [Online]. Available: <https://www.dsprelated.com/showarticle/1171.php>
- [41] PalmSens. (2021) Palmsens4 - compact, versatile and powerful. (Date last accessed 23.01.2021). [Online]. Available: <https://www.palmsens.com/product/palmsens4/>
- [42] A. T. Group. (2021) Autolab - modular electrochemical instruments. [Online]. Available: https://www.artisanng.com/info/Metrohm_Autolab_microAutolabII_PGSTAT12_PGSTAT30_PGSTAT100_Datasheet.pdf

Array of sine values in simulation

```
uint8_t sine = { 127,131,135,139,143,147,151,154,158,162,166,170,174,177,
181,185,188,192,195,199,202,205,208,211,214,217,220,223,225,228,230,233,235,237,
239,241,243,244,246,247,248,250,251,252,252,253,254,254,254,255,255,255,254,254,
254,253,252,252,251,250,248,247,246,244,243,241,239,237,235,233,230,228,225,223,
220,217,214,211,208,205,202,199,195,192,188,185,181,177,174,170,166,162,158,154,
151,147,143,139,135,131,129,125,121,117,113,109,105,102,98,94,90,86,82,79,75,71,
68,64,61,57,54,51,48,45,42,39,36,33,31,28,26,23,21,19,17,15,13,12,10,9,8,6,5,4,4,
3,2,2,1,1,0,1,1,2,2,3,4,4,5,6,8,9,10,12,13,15,17,19,21,23,26,28,31,33,36,39,42,
45,48,51,54,57,61,64,68,71,75,79,82,86,90,94,98,102,105,109,113,117,121,125,127 };
```

Array of sine values in implementation

```
uint8_t sine = { 0, 0x3, 0x7, 0xb, 0xf, 0x13, 0x17, 0x1b, 0x1f, 0x23, 0x27, 0x2b,
0x2e, 0x32, 0x36, 0x39, 0x3d, 0x40, 0x44, 0x47, 0x4a, 0x4d, 0x50, 0x53, 0x56,
0x59, 0x5c, 0x5f, 0x61, 0x64, 0x66, 0x69, 0x6b, 0x6d, 0x6f, 0x71, 0x72, 0x74,
0x76, 0x77, 0x78, 0x79, 0x7b, 0x7b, 0x7c, 0x7d, 0x7e, 0x7e, 0x7f, 0x7f, 0x7f };
```

Parts of the implementation

```
void MW_CROSS_CORRELATION_Calc(int *first_signal, int *sec_signal, uint16_t length, int *cross_corr)
{
    uint16_t maxlag = length - 1;
    int temp = 0;
    int i = 0;
    uint16_t index = 0;
    uint16_t j = 0;
    for(i = -maxlag; i <= maxlag; i++)
    {
        for(j = 0; j < length; j++)
        {
            int k = j + i;
            if(k >= 0 && k < length)
            {
                temp = temp + (first_signal[j]*sec_signal[k]);
            }
        }
        cross_corr[index] = temp;
        index++;
        temp = 0;
    }
}
```

Figure A1: Implemented cross-correlation function following [33]

```

uint16_t delay_loop = 0;
for (uint8_t loop = 0; loop < 3; loop++)
{
    EWOMIS_DAC_SetDataOutValue(DAC, stage0_out1, stage0_out2, 0x0, 0x7F);
    EWOMIS_ROMFW_DelayMs(5);
    // #####
    // ## Transceive data
    memset(adc_data, 0, sizeof(&adc_data));
    _transmitData(length, dac_data, adc_data, reload);

    // #####
    // Determination of current signal for further calculations
    float adc_current_sensor_mA = 0;
    float adc_volt_V = EWOMIS_ADC_ContConvFil_ConvertDigToVolts(adc_data[0], bitwidth);
    float adc_measured_V = EWOMIS_ADC_RecalcVScaling(adc_volt_V, fsr_gain);
    adc_current_sensor_mA = EWOMIS_POT_WE_RecalcVScalingRShunt(EWOMIS_POT_WE1, adc_measured_V, rsel);

    // Determination of external resistance to determine threshold voltage
    resistance = (1200 - (100 * 4.7)) / adc_current_sensor_mA;
    float max_current_mA = (float) (1200 - ((100 + step) * 4.7)) / (float) resistance;
    float factor = (0.01 / resistance) * 10000;
    max_current_mA += factor;
    for (delay_loop = 0; delay_loop < length; delay_loop++)
    {
        float adc_volt_V = EWOMIS_ADC_ContConvFil_ConvertDigToVolts(adc_data[delay_loop], bitwidth);
        float adc_measured_V = EWOMIS_ADC_RecalcVScaling(adc_volt_V, fsr_gain);
        adc_current_sensor_mA = EWOMIS_POT_WE_RecalcVScalingRShunt(EWOMIS_POT_WE1, adc_measured_V, rsel);
        if (adc_current_sensor_mA < max_current_mA)
        {
            break;
        }
    }

    delay_pos += (delay_loop - delay) - 1;
}
delay_pos = (delay_pos / 3);
return delay_pos;

```

Figure A2: Implementation of calibration function

```

// #####
//moving average filter - should only be used with capacitive load
uint8_t samples = 0;
uint32_t average = 0;

if (freq_pos < 6)
    samples = 4;
if (freq_pos < 8)
    samples = 8;
else
    samples = 16;

// starting point of moving average
for (uint16_t start_loop = adc_start - (samples / 2); start_loop < adc_start + (samples / 2);
    start_loop++)
{
    average += adc_data[start_loop];
}
average /= samples;

for (uint16_t i = adc_start - (samples / 2) + 1;
    i < adc_start + sinus_signal_length - (samples / 2) + 1; i++)
{
    int average_res = (int) (((int) adc_data[i + samples] - (int) adc_data[i]) / samples);
    average = average_res + average;
    adc_signal[pos] = average;
    pos++;
}

```

Figure A3: Implementation of the additional moving average filter