



Bioimpedance Monitoring of Cell Cultures

Digital Bioimpedance Acquisition System based on the
Red Pitaya Instrumentation Board

Magna Karina Morales Nordgard
Master's Thesis, Spring 2018



Contents

List of Figures	v
1 Introduction	1
1.1 Objectives	2
1.2 Thesis structure	2
I Conceptual Framework	3
2 General Impedance: Background Theory	4
2.1 Impedance and Admittance	4
2.1.1 Resistance and Conductance	6
2.1.2 Reactance	7
3 Bioimpedance: Background Theory	8
3.1 Permittivity and Polarization	8
3.1.1 Relaxation	9
3.1.2 Dispersion	10
3.1.3 Dielectric Response of Biological Tissues	12
3.2 Equivalent Circuit Models	14
3.2.1 Debye Models	14
3.2.2 Constant Phase Element (CPE)	15
3.2.3 Cole Models	17
3.3 Instrumentation	18
3.3.1 Electrodes and Electrode Configurations	18
3.3.2 Operational Amplifiers	19
3.3.3 General architecture of a bioimpedance measurement system	21
4 Cell Cultures	22
4.1 The Human Cell	22
4.2 Impedance Monitoring of Cell Cultures	25
II Application Development	26
5 Red Pitaya STEMlab Platform	27
5.1 Overview	27
5.2 Hardware Specifications	28
5.2.1 Fast Analog IOs	29
5.2.2 Extension Connectors	30

5.2.3	Official LCR meter application	31
6	Software Development	34
6.1	Application Development using NGINX server	34
6.2	Development with Red Pitaya API	37
6.3	Remote Control with SCPI server	38
6.3.1	Signal Generation	38
6.3.2	Signal Acquisition	39
7	Embedded System Development	42
7.1	Development Tools and Project Set-up	42
7.2	Xilinx IP	45
7.2.1	Envisioned System Overview	45
7.2.2	General Purpose I/O (GPIO) Core	46
7.2.3	Direct Digital Synthesizer (DDS) Core	48
7.2.4	Cascaded Integrator-Comb (CIC) Core	48
7.2.5	Block Random Access Memory (BRAM)	49
8	Red-Pitaya Based Bioimpedance Acquisition System	50
8.1	Electrodes	50
8.2	Front End Prototype	51
8.3	Acquisition Scripting in MATLAB	52
8.3.1	Signal Generation and Acquisition	52
8.3.2	Demodulation	53
8.3.3	Script Structure	54
III	Results and Future Work	56
9	Results	57
9.1	Measurements on Passive Components	57
9.1.1	Test RC Network	57
9.1.2	Gamry Dummy Cell	59
9.2	Cell Measurements	62
9.2.1	Cell culture preparation	62
9.2.2	Measurements on Plasma Treated Electrodes	65
9.2.3	Measurements on Cells	66
10	Conclusions and Future Work	71
10.1	Front-End	71
10.1.1	Proposed Improvements	72
10.2	Scripts	76
10.3	Embedded System Development	76
Appendices		76
A	Embedded Development Tools	78
A.1	Installing Vivado Design Suite	78
B	MATLAB Scripts	79
B.1	Generate	79

B.2 Acquire	80
B.3 Demodulation	81
B.4 Main Script	82
C Front-End Schematics	84
Bibliography	89

List of Figures

2.1.1 Example geometry of an ideal resistor	6
2.1.2 Geometry of a parallel plate capacitor	7
3.1.1 Single relaxation-time dielectric dispersion. From [9].	10
3.1.2 Single relaxation-time Wessel plot. From [9].	11
3.1.3 Resultant dispersion from overlapping relaxations. From [9].	11
3.1.4 Idealized dispersion regions. From [5].	12
3.1.5 Ion concentration and charge distribution at the electrical double layer. From [9].	12
3.2.1 Debye Model Variants.	14
3.2.2 a) Permittivity Wessel Plot for the series dielectric model (SDM). b) Effect of a conductance in parallel with the SDM. c) Conductivity Wessel Plot for the series conductor model (SCM). d) Effect of a capacitor in parallel with the SCM. From [5].	15
3.2.3 Immitance of an isolated CPE. Notice that the CPE symbols are different from the standard resistance and capacitance symbols. From [5].	16
3.3.1 Operational Amplifier Equivalent Circuit. Modified from [14].	19
3.3.2 Operational Amplifier Circuits. From [5].	20
4.1.1 Generalized human cell showing the principal organelles. From [17].	22
4.1.2 Diagram of the Cell Membrane. From [19].	23
4.1.3 Equivalent circuit model for a cell. From [18].	24
5.2.1 STEMlab board. From [23].	28
5.2.2 Jumper settings for input voltage range. From [23].	29
5.2.3 Extension connectors. From [23].	30
5.2.4 Extension Board for LCR meter.	31
5.2.5 LCR Meter Panel.	31
5.2.6 Histogram of capacitance and of a 1 uF film capacitor at 100 Hz.	32
5.2.7 Histogram of capacitance and phase of a 1 uF film capacitor at 10 kHz.	33
6.1.1 Screenshot of Red Pitaya Dashboard	34
6.1.2 Red Pitaya Backend with NGINX. From [23].	35
6.3.1 SCPI application in the Red Pitaya dashboard	38
7.1.1 Script dependencies	44
7.2.1 System Architecture	45
7.2.2 GPIO customization window	46
7.2.3 GPIO architecture block diagram. From [31].	47
7.2.4 Simplified DDS Block Diagram. From [32].	48

8.1.1 Carbon Microelectrodes	50
8.2.1 Analog Front End Schematic with Trans-impedance Amplifier	51
8.3.1 Flow diagram of the code structure used for cell culture monitoring.	55
9.1.1 Test Circuit	57
9.1.2 RC Test Circuit Semilogarithmic Plot	58
9.1.3 RC Test Circuit Wessel Plot	58
9.1.4 EIS Dummy Cell Schematic Diagram. From [36].	59
9.1.5 Gamry Dummy Cell Measurement Set-up	60
9.1.6 Gamry Dummy Cell Semilogarithmic Plot	61
9.1.7 Gamry Dummy Cell Wessel Plot	61
9.2.1 Electrode Chip	62
9.2.2 Electrode holder with one electrode chip in socket.	63
9.2.3 Assembled electrode holder.	63
9.2.4 Plasma Treated Electrode Semilogarithmic Plot	65
9.2.5 Coated Electrode Wessel Plot	66
9.2.6 Initial measurement on cells with low excitation voltage	67
9.2.7 Measurement with higher excitation voltage	68
9.2.8 Two measurements taken about 19 hours apart.	69
9.2.9 Real component of impedance. Measurements taken about 19 hours apart.	70
10.1.1Front-End Revision Proposal.	72
10.1.2Simulated responses using the AD8642-based prototype design, and the LTC1028-based revised design. Calculated values added for reference.	75

Chapter 1

Introduction

The Bioimpedance Group at the University of Oslo is involved in a collaboration between various European universities called TRAINING4CRM, where the aim is to train PhD students to perform interdisciplinary research, in particular within the field of cell-based regenerative medicine [1]. This branch of research deals with methods that can be used to restore the functionality of tissues whose behaviour has been altered due to trauma or chronic pathological conditions [2]. The technological goals of the TRAINING4CRM project include developing an implantable device to restore brain functions that are lost through neurodegenerative disorders such as Parkinson's disease. Such disorders are caused by gradual changes in the structure of the cells or neurons in the brain, as well as loss of function or death in these cells [1]. In order to restore the impaired functionality, dead cells could potentially be replaced by new cells [3], and this is the essence of what the implantable device should facilitate.

Stem cells are cells that are capable of changing from one cell type to another, a process that is known as differentiation. Embryonic stem cells (ESCs) are capable of differentiating into all types of cells, thus they may conceivably be used to replace the damaged neurons in the brain of patients suffering from neurodegenerative diseases. However, knowledge of the techniques by which the differentiation process of the ESCs can be controlled is insufficient, and this challenge must be addressed and overcome in order to be able to employ this method in disease treatment [3]. Additionally, one of the objectives of the TRAINING4CRM project is to genetically modify the implanted stem cells so that functions such as neurotransmitter release may be controlled optogenetically, meaning that control is enacted by stimulating the cells with light.

Tailored instrumentation is necessary to control the differentiation process of the cells and the light source of the implant, as well as to sense and confirm that the cells respond as expected. The use of bioimpedance measurements in this context may prove to be a useful tool in tracking the development of the cell culture in the implant, and potentially also to detect neurotransmitter release.

This master's project was proposed as a pilot study under the TRAINING4CRM project, with the aim of developing a bioimpedance measurement system capable of obtaining impedance spectra from cell cultures in the frequency range from 100 Hz to 100 kHz, and engage in research of the potential relevance of bioimpedance measurements in the context of cell culture research, and, in particular, cell-based regenerative medicine.

Additionally, the need to develop compact and inexpensive measurement systems for use in general bioimpedance research is an important motivation to test the capabilities of novel development platforms within the Bioimpedance Group. Therefore a promising development board that was unfamiliar to the group was selected to be surveyed and used as the baseline for the developed instrument, with the objective of assessing the capabilities and development alternatives presented by the platform.

1.1 Objectives

- Develop an instrument for measurement of complex electrical impedance in the frequency range 100 Hz to 100 kHz using the Red Pitaya STEMLab platform.
- Investigate the development approaches that can be used to develop instrumentation based on the Red Pitaya STEMLab platform.
- Present a theoretical framework that explains the origin of parameters that are observable through bioimpedance measurements of cell cultures.
- Perform measurements on cell cultures to demonstrate the ability of the developed instrument to detect changes in important parameters such as cell proliferation.

1.2 Thesis structure

This thesis is divided in three parts. The first part discusses the theoretical framework of the project, with a focus on bioimpedance theory and cell culture applications. The second part discusses the Red Pitaya platform, the various approaches that can be used in order to develop instruments with said platform, and presents the design that was tested at the end of the project. Finally, the third part presents the obtained results and suggests how to proceed with future work.

Part I

Conceptual Framework

Chapter 2

General Impedance: Background Theory

This chapter discusses the definitions of impedance and admittance as they are used when describing linear electrical circuits. Linearity in a circuit refers to the ability of the circuit to preserve the shape of the applied signal [4]. This means that when the input of a linear circuit is a sinusoidal of a certain frequency, the output will also be a sinusoidal of the same frequency, but the amplitude and the phase of the sinusoidal may vary.

2.1 Impedance and Admittance

Electrical impedance is the measure of the total opposition that a circuit presents to the passage of electric current when a voltage is applied. This can be expressed in the form of Ohm's Law:

$$Z = \frac{v}{i} = [\Omega] \quad (2.1)$$

where v is the applied voltage and i is the current.

The electrical impedance of a circuit is composed of a real part, given by the resistance R of the circuit, and an imaginary part, composed of both the inductive and capacitive reactance X of the circuit. Total electrical impedance may then be defined in Cartesian form as:

$$Z = R + jX = [\Omega] \quad (2.2)$$

Admittance is defined as the inverse of impedance, and is the measure of how easily a circuit allows applied current to flow.

$$Y = \frac{1}{Z} = \frac{i}{v} = [S] \quad (2.3)$$

Admittance is similarly composed of a real and an imaginary part:

$$Y = G + jB \quad (2.4)$$

Where G stands for conductance, the real part of the admittance, and B stands for susceptance, the imaginary part of the admittance.

The admittance variables G and B, may be found if the impedance variables R and X are known, using:

$$G = \frac{R}{R^2 + X^2} \quad (2.5)$$

$$B = -\frac{X}{R^2 + X^2} \quad (2.6)$$

And inversely:

$$R = \frac{G}{G^2 + B^2} \quad (2.7)$$

$$X = -\frac{B}{G^2 + B^2} \quad (2.8)$$

The impedance of a circuit may also be expressed in the equivalent exponential form as:

$$Z = |Z|e^{-j\theta} \quad (2.9)$$

Where $|Z|$ is the magnitude and θ is the phase difference between the voltage signal and the current signal in the time domain.

$$|Z| = \sqrt{R^2 + X^2} \quad (2.10)$$

$$\theta = \arctan\left(\frac{X}{R}\right) \quad (2.11)$$

$$R = |Z|\cos(\theta) \quad (2.12)$$

$$X = |Z|\sin(\theta) \quad (2.13)$$

Table 2.1.1 gives a summary of the names, symbols and units of the various components of impedance and admittance. The names of the units Ω and S are Ohm and Siemens respectively.

Table 2.1.1: Designations of impedance and conductance

	Name	Symbol	Unit
Real Impedance	Resistance	R	Ω
Impedance of an ideal inductor	Inductive Reactance	X_L	Ω
Impedance of an ideal capacitor	Capacitive Reactance	X_C	Ω
Total Impedance	Impedance	Z	Ω
Real Admittance	Conductance	G	S
Imaginary Admittance	Susceptance	B	S
Total Admittance	Admittance	Y	S

2.1.1 Resistance and Conductance

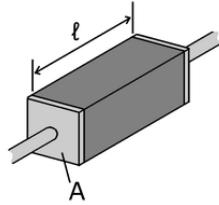


Figure 2.1.1: Example geometry of an ideal resistor

The resistance of a component depends on two factors, the resistivity of the material, and the geometry of the object. The resistivity ρ is a material constant given in Ohm-meters [$\text{Ohm} \cdot \text{m}$]. Assuming a conductor of uniform material and cross section such as the one in figure 2.1.1, we can compute the resistance from equation 2.14.

$$R = \rho \frac{l}{A} \quad (2.14)$$

Similarly the conductance of such a component can be computed:

$$G = \sigma \frac{l}{A} \quad (2.15)$$

Where σ is the conductivity given in Siemens per meter [S/m]. Notice that resistivity and conductivity are reciprocals $\rho = 1/\sigma$.

2.1.2 Reactance

Reactance is the imaginary part of the impedance of a component or circuit. This definition doesn't say much about the nature of reactance, and it is better to define reactance as the **opposition** of a circuit element **to a change** in current or voltage due to that element's inductance or capacitance. Changing the frequency of the signal that is applied to the circuit element is equivalent to changing *the rate of change* of the signal, thus reactance is a frequency dependent electrical property.

Capacitive Reactance

Capacitive reactance is the measure of the opposition to the change of voltage across an element, and it is given by:

$$X_C = -\frac{1}{\omega C} = -\frac{1}{2\pi f C} = [\Omega] \quad (2.16)$$

Where ω is the angular frequency $\omega = 2\pi f$ and C is the capacitance of the component.

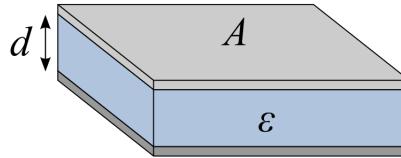


Figure 2.1.2: Geometry of a parallel plate capacitor

For an ideal parallel plate capacitor such as the one depicted in figure 2.1.2, the capacitance is given by:

$$C = \epsilon \frac{A}{d} = \epsilon_r \epsilon_0 \frac{A}{d} = [F] \quad (2.17)$$

Where $\epsilon = \epsilon_r \epsilon_0$ is the permittivity of the material between the plates. ϵ_0 is a constant, the permittivity of vaccum, and ϵ_r is the relative permittivity (also known as dielectric constant) of the material.

The permittivity of a material is the measure of its ability to resist an electric field. In general, materials with high relative permittivity (absolute permittivity expressed as a ratio of the materials permittivity relative to the permittivity of vaccum) are denoted dielectric materials. We may classify a material as a dielectric when the displacement current is larger than the in-phase current [5]. This is true when $\omega C > G$ is satisfied.

Inductive Reactance

Inductive reactance is the measure of the opposition to the change of current across an element with changing voltage, and it is given by:

$$X_L = \omega L = 2\pi f L = [\Omega] \quad (2.18)$$

When discussing bioimpedance we seldomly need to consider inductive reactance, therefore this topic will not be discussed further.

Chapter 3

Bioimpedance: Background Theory

The concept of bioimpedance refers to the impedance of a biological material, and describes the passive electrical properties of a said substance.

In bio-materials the charge carriers are ions dissolved in the intra- and extracellular liquid environments [5], and for current to flow these ions must be displaced within the tissue. It can be proven that these electrolyte solutions behave as conductors and follow Ohm's law [5]. On the other hand, biological cell membranes are dielectrics with bound charges, and display both high resistivity and high capacitance [6].

These effects combine when biological tissue is excited by an alternating signal, and give rise to complex behaviour that is dependent on the frequency of the aforementioned signal. The measured frequency-dependent bioimpedance data can be used to derive information about the tissue's physiological conditions and structure [7].

3.1 Permittivity and Polarization

When the definition of capacitive reactance was discussed in chapter 2, the concepts of permittivity and dielectric materials were introduced.

Permittivity was defined as a material constant that gives the measure of a material's ability to resist the formation of an electric field, and materials with high relative permittivity were classified as dielectric materials.

Bio-materials are complex heterogeneous materials, and will display both resistive and reactive behaviour to varying degrees depending on their composition. Discrete components in real life also display mixed behaviour, but these effects are rather small because the components are manufactured with heterogeneous materials specifically selected to display the desired behaviour.

For example, a parallel-plate capacitor will have a homogeneous dielectric material between its plates. The material will dissipate a small amount of electromagnetic energy, and therefore the capacitor will display a small conductive behaviour. The dissipation of electromagnetic energy in dielectric materials is known as dielectric loss, and in practical electronic applications

it is modelled as a small resistance in series with the capacitor, and is termed equivalent series resistance (ESR) [8].

To better understand the behaviour of heterogeneous materials such as biologic tissue, we need to understand the mechanics that influence permittivity.

When an electric field is imposed on a given material, the material will experience a local disturbance (shift from average equilibrium positions) of the charge distribution in the volume. The volume will experience *electric polarization*.

Polarization in homogeneous materials happens through three mechanisms:

1. **Electronic Polarization:** When an electric field is applied to the material, the electron cloud of individual atoms is displaced with respect to their nucleus. These atoms become so-called induced dipoles, and these will orient themselves in the direction of the local electric field.
2. **Orientational Polarization:** The already existing permanent dipoles experience torque, which causes rotational movements, when they are influenced by an electric field.
3. **Ionic Polarization:** Re-positioning of ions relative to each other, which entails that the ions migrate and substance is displaced.

Polarization is dependent on the time it takes for the charges to rearrange themselves within the material. If the material is excited with an AC signal, we will observe that with a lower frequency, the polarization displayed by the material reach a higher level than it would with a high frequency excitation signal. The reason is that the charges will have more time available per half-period to rearrange themselves and reach maximum polarization.

3.1.1 Relaxation

The momentary lag in the permittivity of a given material, caused by the delay of the polarization in the material in the presence of a changing electric field, is known as **dielectric relaxation** [5]. We may also say that a significant change in dielectric properties over a frequency range is a dielectric dispersion [9].

Relaxation occurs in the time domain, and is observed by applying a step function excitation signal. The time it takes for the system to reach a new equilibrium after the step increase or decrease of the electric field is known as the **relaxation time**. This time is dependent on the polarization mechanism: electronic polarization has the fastest relaxation times, whereas the orientational polarization of large organic molecules such as proteins can be much slower [5].

The response of a medium with a single relaxation process is a first-order process, described by the Debye equation:

$$D(t) = D_\infty + \frac{D_0 - D_\infty}{1 - e^{-t/\tau}} \quad (3.1)$$

Where D_∞ is the flux density at $t=0+$ and D_0 is the static value obtained after a time $t \gg \tau$ [10].

The flux density D is related to the permittivity ϵ and the electric field E according to:

$$\mathbf{D} = \epsilon \mathbf{E} \quad (3.2)$$

Note that both **D** and **E** are vector fields (indicated by the bold typesetting), which means they have a value for magnitude and direction in every point in space.

A heterogeneous material will exhibit the various polarization mechanisms in accordance with its composition. These mechanisms will have different relaxation times. Additionally, homogeneous materials present interfacial polarization effects that arise from induced charge accumulation at the interface between different materials [9]. Interfacial polarization exhibits the longest mean relaxation times, which may be in the order of seconds. These interfacial effects play a significant role in the frequency dependence of the bioimpedance of complex materials [10].

3.1.2 Dispersion

The frequency dependence of permittivity (in accordance with relaxation theory) is known as **dielectric dispersion** [5]. The permittivity response in the frequency domain of a medium with a Debye type relaxation is given by:

$$\varepsilon = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + j\omega\tau} \quad (3.3)$$

where ε_∞ is the dielectric constant at frequencies much higher than the characteristic frequency $1/(2\pi\tau)$, ε_s is the very low frequency (static) dielectric constant, ω is the angular frequency $2\pi f$, and τ is the relaxation time [9].

The consequence of equation 3.3 is that we must regard permittivity as a *complex value*, which in practice means that the material has a degree of dielectric loss that causes energy dissipation. This loss is caused by relaxation mechanisms, and independent of free charge conduction.

The dielectric constant (ε') and the loss factor (ε'') can be derived from equation 3.3:

$$Re[\varepsilon] = \varepsilon' = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + \omega\tau} \quad (3.4)$$

$$Im[\varepsilon] = \varepsilon'' = (\varepsilon_s - \varepsilon_\infty) \frac{\omega\tau}{1 + \omega^2\tau^2} \quad (3.5)$$

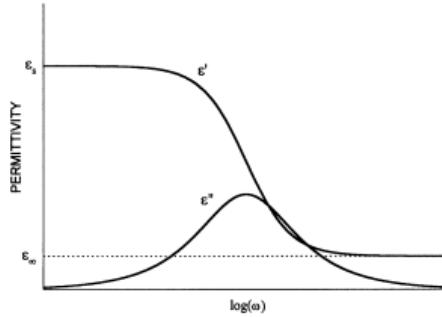


Figure 3.1.1: Single relaxation-time dielectric dispersion. From [9].

Figure 3.1.1 shows a plot of both the dielectric constant and the loss factor against the logarithm of angular frequency. We see that the real permittivity falls from a static value ε_s at low frequencies to a lower value ε_∞ at high frequencies, while the complex permittivity peaks at the characteristic frequency.

Another way to represent complex permittivity data is by plotting ε'' against ε' in a Wessel plot. For a medium with a single relaxation time, the Wessel plot will be a perfect semi-circle, as shown in figure 3.1.2.

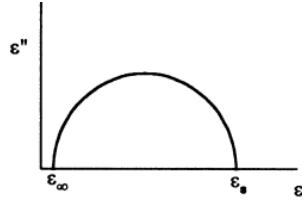


Figure 3.1.2: Single relaxation-time Wessel plot. From [9].

For dielectric properties with multiple relaxation times, the relative complex permittivity may be written as:

$$\varepsilon = \varepsilon_\infty + \frac{\Delta\varepsilon_1}{1 + j\omega\tau_1} + \frac{\Delta\varepsilon_2}{1 + j\omega\tau_2} + \dots \quad (3.6)$$

where ε_∞ is the dielectric constant at very high frequencies, $\Delta\varepsilon_i = (\varepsilon_{si} - \varepsilon_\infty)$ is the dielectric increment for the i th dielectric relaxation, and τ_i is the i th relaxation time [9].

Materials with multiple relaxation mechanisms, may exhibit overlapping dispersions. In the graphical representation of such a case, the frequency response of ε' will be sharper, while the peaking of ε'' will be more gradual and less defined. The Wessel plot will appear flattened, no longer a semicircle.

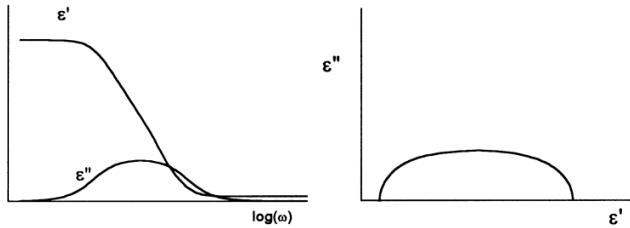


Figure 3.1.3: Resultant dispersion from overlapping relaxations. From [9].

Since single relaxation-time responses are rare, measured data is often fitted with empirical models that take into account a distribution of time constants (DRT). One such approach is to use the Cole-Cole equation:

$$\varepsilon = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + (j\omega\tau)^{1-\alpha}} \quad (3.7)$$

where α is an adjustable parameter indicating the DRT. Data from biological materials can often be fitted to the above equation with $\alpha = 0.3$ to 0.5 , indicating a broad spectrum of relaxation times [9].

3.1.3 Dielectric Response of Biological Tissues

Biological tissues typically exhibit dielectric properties that are characterized by three major dispersions: α , β and γ , and sometimes also a smaller dispersion δ . These are shown in figure 3.1.2.

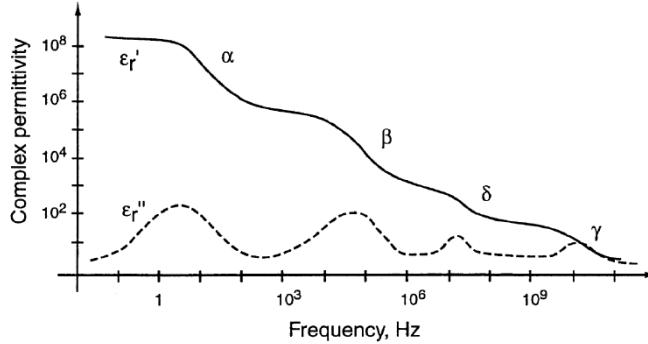


Figure 3.1.4: Idealized dispersion regions. From [5].

In 1957, H.P. Schwan described the low frequency α , radio frequency β and microwave frequency γ dispersions and observed that different relaxation mechanisms were behind them [10].

α -dispersion

Although α -dispersion was observed early by Schwan in 1957, the mechanisms behind this type of dispersion are often unclear [11]. The dispersion grouping has been based on the frequency range (from below 1 Hz, up to 100 kHz), rather than knowledge of the mechanisms that cause it.

Schwan proposed that a plausible cause for α -dispersion could be counter-ion relaxation effects near the membrane surfaces. A counter-ion is understood to be the ion that accompanies an ionic species in order to maintain electric neutrality. At boundaries such as cell-membranes, ions from the extracellular liquid will be gradually bound by the charge distribution at the interface to the cell membrane, forming a counter-ion layer or electrical double layer.

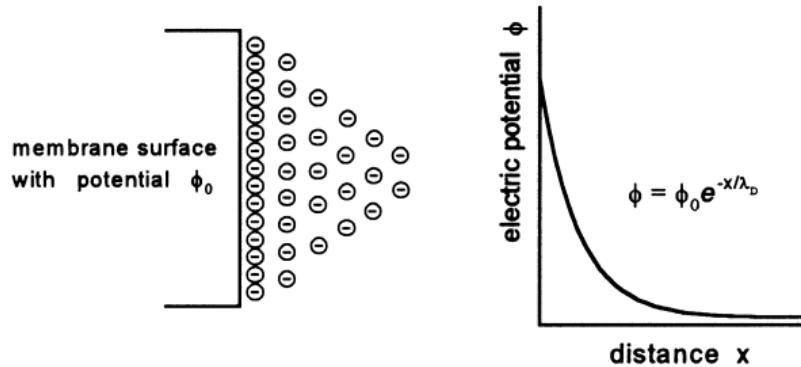


Figure 3.1.5: Ion concentration and charge distribution at the electrical double layer. From [9].

The counter-ions might be able to move laterally along the surface, but not transversely across it, in the presence of an external electric field. When the electric field is switched off, the reestablishment of the original counter-ion atmosphere (charge distribution) will be diffusion controlled [10], leading to a relaxation with a relatively long time constant. In the frequency domain this will be a dispersion in the α -frequency range.

β -dispersion

Biological tissues usually have membrane systems that extend to other tissues and block ion movement. Under the influence of an external electric field, ions will accumulate at these membrane surfaces, charging the interfaces. This effect is a Maxwell-Wagner type interfacial relaxation process, that is present in all systems where electric current must pass an interface between two different dielectrics. The conductance of the first dielectric that the current passes will contribute to the build-up of charge at the interface to the second dielectric [9].

Cell suspensions typically exhibit a large β dispersion, due to the interfacial effect at the interface between the phospholipid cell membrane and the intra- or extracellular solution [10].

δ -dispersion

Many molecules found in biological tissue are polar, like water and proteins. These permanent dipoles will be randomly oriented in the absence of an external electric field, but reorient in the presence of an external electric field. Other molecules will become induced dipoles and align to the electric field.

The orientation of polar molecules requires time and causes a Debye type relaxation as described by equation 3.3. Due to molecular in-homogeneity, shape variance, and not only one type of molecule but several types of molecules undergoing relaxation processes, there will be a distribution of time constants [10]. The case will be better described by equation 3.6. Typical characteristic frequencies are 15-20 GHz for water, 400-500 MHz for amino-acids.

γ -dispersion

The dispersion of polar proteins, by the same mechanism described under δ -dispersion, appears generally around the 1-10 MHz range. The magnitude of the dielectric decrement is smaller than for typical β dispersions and appears at the tail of the frequency range of the β dispersion but before the typical δ dispersion range.

3.2 Equivalent Circuit Models

The electrical properties of cells and biological tissue can be modelled using various types of equivalent circuits. The equivalent circuits outlined below account for various levels of complexity in the frequency response of the variables that are attempted modeled, as well as if the behaviour is mainly that of a conductor or a dielectric.

3.2.1 Debye Models

Debye models consist of three ideal components and are suitable to describe the presence of a *single* relaxation process. Figure 3.2.1 shows the four possible configurations of ideal components that are used as Debye Models.

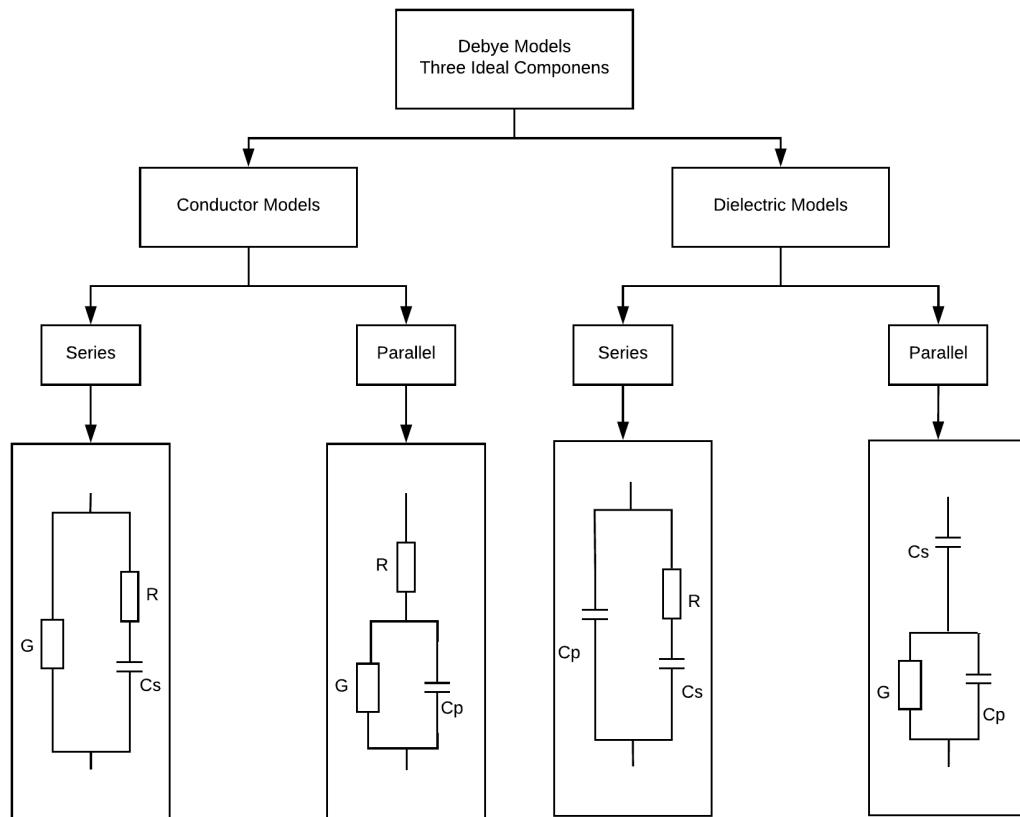


Figure 3.2.1: Debye Model Variants.

Both dielectric models will produce a semicircle in the dielectric plane, like the one shown in figure 3.2.2.a. The conductor models will also produce a semicircle, but in the admittance plane seen in figure 3.2.2.b, with real (σ') and imaginary (σ'') conductivity on the axes rather than real and imaginary permittivity.

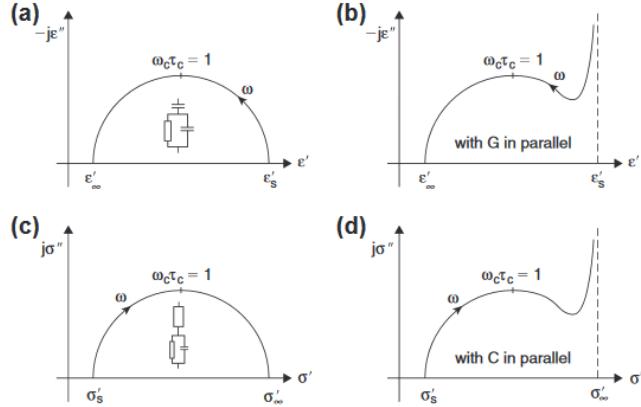


Figure 3.2.2: a) Permittivity Wessel Plot for the series dielectric model (SDM). b) Effect of a conductance in parallel with the SDM. c) Conductivity Wessel Plot for the series conductor model (SCM). d) Effect of a capacitor in parallel with the SCM. From [5].

Figure 3.2.2 also displays the effects of adding a conductance in parallel to the SDM (b) and the effects of adding a capacitor in parallel to the SCM (d). These additions disturb the semicircles at low frequencies.

The parallel dielectric model in particular is suitable for describing a single relaxation process in a dielectric as described by the Debye equation presented in sections 3.1.1 and 3.1.2, and it can be defined by one unique time constant [5].

The series conductor model has been used as a model for cells, where C_s represents the cell membrane capacitance, R represents the intracellular liquid and G represents the extracellular liquid [5]. The model is however lacking in that living cells display more than one dispersion mechanism, therefore more complex models are necessary to reflect their behaviour.

3.2.2 Constant Phase Element (CPE)

The constant phase element (CPE) is a model of a component that is frequency dependent, but as its name suggests, the phase between the real and imaginary components is kept constant [5]. The use of a CPE in the electrical equivalent models of biological tissue will produce a better agreement with the displayed characteristics, since biological tissue typically displays a distribution of time constants. The CPE is suitable in cases where the distribution of time constants is logarithmically symmetrical [10].

The CPE may be modeled as either a frequency dependent admittance or impedance, and is in practice to be understood as a frequency dependent combination of a resistance and a capacitance.

The admittance of a CPE is given by:

$$Y_{CPE} = Q_0(\omega\tau)^m \quad (3.8)$$

$$Q_0 = G_{\omega=1} + jB_{\omega=1} \quad (3.9)$$

$$\varphi_{CPE} = \frac{B_{\omega=1}}{G_{\omega=1}} \quad (3.10)$$

The τ and m parameters, in combination, scale the frequency dependence of the CPE. The value of m must be restricted to $0 \leq m \leq 1$ for the Y_{CPE} to increase with frequency in accordance with tissue, although the definition of a CPE itself does not require m to be restricted. If m is 0, the CPE behaves as an ideal resistor, whereas if m is 1, it behaves like an ideal capacitor.

The complex admittance plot of the Y_{CPE} does not discriminate between a series or parallel arrangement of the variable components. The same applies to the complex impedance plot of the Z_{CPE} . This is illustrated in figure 3.2.3.

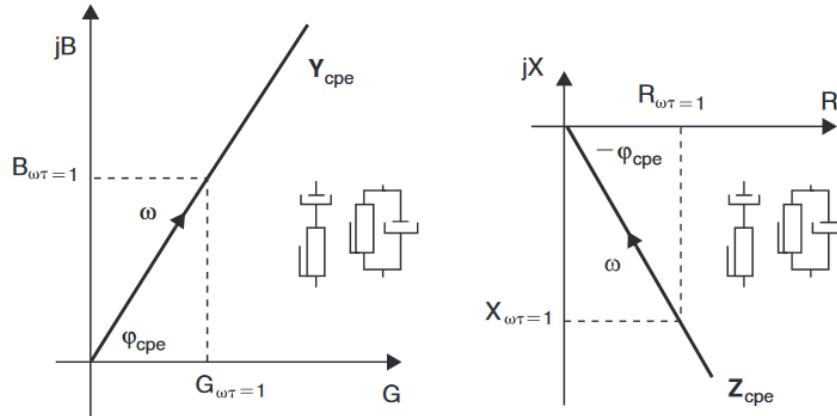


Figure 3.2.3: Immittance of an isolated CPE. Notice that the CPE symbols are different from the standard resistance and capacitance symbols. From [5].

In many electrolytic systems there is also a correlation between the exponent m and the phase angle, according to Fricke's Law [10]. For such systems the CPE model may be modified to include this dependency, according to equations 3.11 and 3.12, where the exponent used is α instead of m .

$$Y_{CPE_F} = G_{\omega=1}(j\omega\tau)^\alpha = \omega^\alpha\tau^\alpha G_{\omega=1}(\cos\frac{\alpha\pi}{2} + j\sin\frac{\alpha\pi}{2}) \quad (3.11)$$

$$Z_{CPE_F} = R_{\omega=1}(j\omega\tau)^{-\alpha} = \omega^{-\alpha}\tau^{-\alpha}R_{\omega=1}(\cos\frac{\alpha\pi}{2} - j\sin\frac{\alpha\pi}{2}) \quad (3.12)$$

The constant phase φ is then $\alpha\frac{\pi}{2}$, unless α too is frequency dependent. If this is the case, then the CPE_F will not be in agreement with the Kramers-Kronig transforms. These transforms are typically used as a tool to analyze experimental data consistency, by checking the relationship between the real and imaginary parts of an immittance or of permittivity [5].

3.2.3 Cole Models

The CPE_F on its own is not enough to describe the response of tissue, it must be used in conjunction with other components. We may substitute the capacitors C_s and C_p from the Debye conductor models (shown in figure 3.2.1) with a CPE_F .

If we substitute the capacitor from the Debye parallel conductor model with a CPE_F , and add a conductance in parallel, we obtain a model that can be described using the *Cole_Z* equation:

$$Z = R_\infty + \frac{R_0 - R_\infty}{1 + (j\omega\tau_Z)^\alpha} \quad (3.13)$$

R_∞ is the resistive part of the impedance at very high frequencies, R_0 is the impedance at 0 Hz, and τ_Z is the inverse of the characteristic frequency ω_c [12], which corresponds to the apex of the arc in the impedance Wessel diagram [5]. The parameter τ can more intuitively be understood as being the mean relaxation time in the distribution of the time constants involved [10].

The Cole equation is an empirical equation which produces a suppressed semicircle in the impedance plane, as is usually the case in measurements from biological materials. Using the relationship in 3.14 it is possible to find the resistance and the reactance as a function of ω [12].

$$j^\alpha = \cos(\alpha\frac{\pi}{2}) + j \sin(\alpha\frac{\pi}{2}) \quad (3.14)$$

$$R = R_\infty + \frac{(R_0 - R_\infty)(1 + (\omega\tau)^\alpha \cos(\alpha\frac{\pi}{2}))}{1 + 2(\omega\tau)^\alpha \cos(\alpha\frac{\pi}{2}) + (\omega\tau)^{2\alpha}} \quad (3.15)$$

$$X = -j \frac{(R_0 - R_\infty)(\omega\tau)^\alpha \sin(\alpha\frac{\pi}{2})}{1 + 2(\omega\tau)^\alpha \cos(\alpha\frac{\pi}{2}) + (\omega\tau)^{2\alpha}} \quad (3.16)$$

Similarly, if we substitute the capacitor from the Debye series conductor model with a CPE_F , we obtain a model that can be described using the *Coley* equation:

$$Z = G_0 + \frac{G_\infty - G_0}{1 + (j\omega\tau_Y)^{-\alpha}} \quad (3.17)$$

One of the consequences of empirical nature of the Cole equations is that they are suited to describe the response of tissue, but not to explain the causes of the described response [5]. Nonetheless, they are broadly used due to their simplicity [10].

A single Cole equation is suitable to model a single electrical dispersion that displays a distribution of relaxation times. For heterogeneous tissues with distinct dispersions over a broader frequency range, each distinct dispersion should be described with an individual Cole equation.

3.3 Instrumentation

3.3.1 Electrodes and Electrode Configurations

The electrode is the interface between the biological material and the instrumentation electronics, where the ionic current in the material is transformed into electronic current through electrochemical reactions. Both the geometry and the material of the electrodes chosen for a given set-up are significant to how the electrodes influence the measurement.

A single electrode in contact with the material is called a half-cell. On its own, a half cell is not enough to carry current or measure a potential; at least two electrodes are necessary to close the electric circuit [5].

In the bioimpedance literature, we distinguish between current carrying and pick-up electrode pairs. In order to close the circuit when injecting a current, we need two electrodes, hence the electrode pair is current carrying. Similarly, in order to measure a voltage or potential difference, we need to measure the difference between two electrodes, hence the electrode pair is used to pick-up a voltage.

Electrodes through which current flows are polarized and will display electrode polarization impedance. Ideally, pick-up electrodes should not be polarized, but they may still display polarization impedance as a result of endogenic phenomena [5]. Polarization impedance is not a property of the tissue, but of the electrode-tissue interface. It is additional to the impedance of the tissue and is thus in many cases undesirable. However, when monitoring two-dimensional cell cultures, the polarization impedance is an interesting phenomenon, as will be discussed in chapter 4.

Two-electrode measurements, also known as two-terminal measurements, rely on both electrodes acting as both a current-carrying and a pick-up pair simultaneously [13]. These measurements will be heavily influenced by the electrode polarization impedance, as the dualistic role of the electrodes presupposes that current flows through the pick up pair. This is also the measurement configuration chosen for the project.

Other electrode configurations include three- and four-electrode measurements (or three-/four-terminal measurements). Four-electrode measurements have two electrode pairs, where one is acting exclusively as current carrying and the other is acting exclusively as pick-up. Such a configuration is useful to avoid contribution from the electrode polarization impedance or otherwise to target a specific volume in the sample [13].

In three-electrode measurements one of the electrodes acts both as current-carrying and pick-up electrode. Results from measurements done with a three-electrode set-up will include the polarization impedance of this dual-role electrode in addition to the transfer impedance of a tissue volume close to this electrode with decreasing contribution from areas farther away from it [5].

3.3.2 Operational Amplifiers

The choice of electrode configuration directly influences what instrumentation will be needed to perform the measurements. A key component in all electrode configurations is the operational amplifier, a high gain amplifier with two inputs.

In general, it is useful to approach the design of circuits containing an operational amplifier by assuming the amplifier is ideal. The characteristics of an ideal operational amplifier are as follows:

1. $A = \infty$, the gain is infinitely large.
2. $v_0 = 0$ when $v_1 = v_2$, there is no offset voltage.
3. $R_d = \infty$, the input impedance is infinitely large.
4. $R_O = 0$, the output impedance is zero.
5. The bandwidth is unlimited, with no phase shift.

The equivalent circuit for an operational amplifier is shown in figure 3.3.1

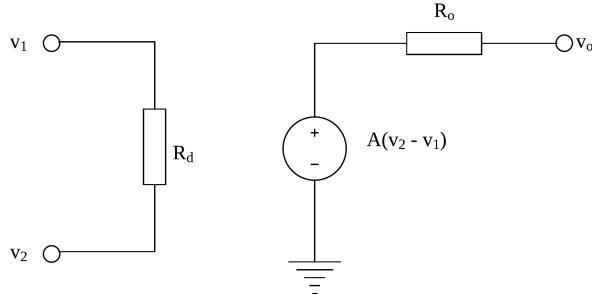


Figure 3.3.1: Operational Amplifier Equivalent Circuit. Modified from [14].

The differential voltage at the inputs v_1 and v_2 is multiplied by the gain of the amplifier, A , to generate an output voltage v_o . The current flowing to the output terminal must pass through the output resistance R_o .

When we apply the assumptions for an ideal amplifier, we arrive at the conclusion that no current flows into the input terminals v_1 and v_2 , because the input resistance R_d is infinite, thus impeding current flow. Also, because the gain of the ideal amplifier is infinite, both inputs v_1 and v_2 must be at the same voltage in order to avoid the situation where multiplying the input voltage difference by infinity would yield an infinite output voltage [14].

Figure 3.3.2.a shows the operational amplifier symbol, as well as the positive and negative power supply rails, which are usually omitted from the schematics.

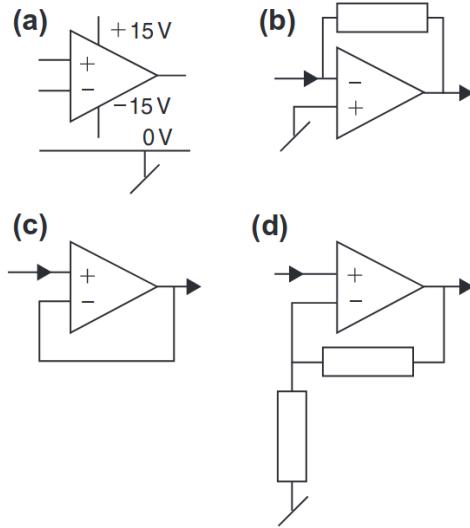


Figure 3.3.2: Operational Amplifier Circuits. From [5].

In instrumentation applications, the operational amplifier must be used with feedback to yield interesting results. The most basic configurations for an operational amplifier are the voltage follower (shown in figure 3.3.2.c), and the non-inverting amplifier (shown in figure 3.3.2.d).

The voltage follower is used as a voltage buffer: the positive input voltage appears at the output, since both input terminals must be at the same voltage level, and the negative input terminal is connected to the output terminal.

Assuming the operational amplifier is ideal, the non-inverting configuration in figure 3.3.2.d provides the following output voltage:

$$v_o = \left(1 + \frac{R_f}{R_1}\right)v_1 \quad (3.18)$$

Where R_1 is the resistor connected to the circuit ground, and R_f is the feedback resistor.

The circuit in figure 3.3.2.b is a simple transimpedance amplifier, that is, a current-to-voltage converter. It accepts an input current i_{in} on the negative input of the operational amplifier, and produces an output $v_o = A \cdot i_{in}$, where A is the gain of the amplifier in volts per ampere [15].

Assuming that the operational amplifier is ideal, with no input offset current, then the gain of the amplifier is determined by the feedback resistance R_f .

$$v_o = -R_f \cdot i_{in} \quad (3.19)$$

The gain is negative due to the choice of reference direction. The gain magnitude is also the sensitivity of the conversion, because it determines the amount change exhibited by the output voltage v_o in response to a change in the input current i_{in} . For a sensitivity of $1V/mA$, a feedback resistor $R_f = 1k\Omega$ is required [15].

Transimpedance amplifiers are typically used in systems where the excitation signal is a voltage, and the response is a current that needs to be converted to a voltage value in order for the system to acquire the response of the sample.

3.3.3 General architecture of a bioimpedance measurement system

Conceptually, a bioimpedance measurement system must consist on three parts:

1. The electrode interface to the sample.
2. A signal conditioning front-end.
3. A back-end responsible of generating the excitation signal, and obtaining and storing the in-phase and quadrature components of the measured variables.

The signal conditioning solution is linked to the selected electrode configuration. Some considerations that may need to be taken into account are the maximum allowed current through the sample, electrode size and placement, and whether the excitation is a voltage or a current. Typically it will include some sort of differential amplifier to measure the voltage drop across the sample and a voltage-to-current or a current-to-voltage converter (such as a transimpedance amplifier) depending on the nature of the excitation signal.

Chapter 4

Cell Cultures

4.1 The Human Cell

Cells are the basic unit of structure and function in the body, and come in a many different shapes and sizes. The majority of the cellular functions are performed by subcellular structures known as organelles, and the amount and type of organelles in a cell depends on the type of cell. Although no particular cell type can be deemed to be "the typical cell", figure 4.1.1 shows a conceptual illustration of a slice of a general human cell, with some of the organelles they most commonly posses [16].



Figure 4.1.1: Generalized human cell showing the principal organelles. From [17].

The cell membrane, depicted in figure 4.1.1, is a thin barrier of approximately 7 to 15 nm that separates the intracellular liquid from the extracellular environment [14]. This membrane

is composed primarily of phospholipids, which have a polar (hydrophilic) and a non-polar (hydrophobic) end. Since the environments on either side of the cell membrane are aqueous, the phospholipids form a double layer to prevent the hydrophobic tails of the molecules to be in contact with the liquid, while the polar hydrophilic heads are exposed to the liquid on both surfaces.

The intrinsic electrical conductance of the lipid double layer is very low, it is not permeable to ions, and thus it is considered a dielectric. Since this dielectric is surrounded by the conducting intra- and extracellular liquids, it behaves as a capacitance [18]. This capacitance is linked to a dispersion in the β -range.

Molecules such as cholesterol, carbohydrates and proteins are found embedded in the cell membrane. Some of these molecules facilitate ion exchange between the interior and the exterior of the cell. We can regard these transport molecules as conductances in parallel with the lipid double layer capacitance.

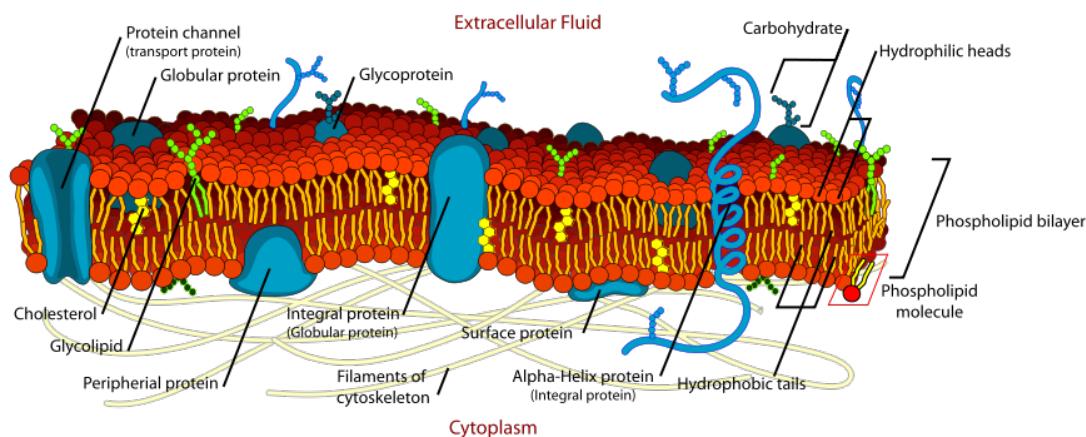


Figure 4.1.2: Diagram of the Cell Membrane. From [19].

Some of the organelles within the cell possess membranous systems of their own. These membranes are also lipid double layers, and they also separate conductive electrolytes. Thus they may also be regarded as capacitances, albeit of lower value, and it has been suggested that the γ -dispersion phenomenon may be linked to the dispersion caused by these capacitances.

Electrically the cell membrane is akin to a leaky capacitor. The lipid double-layer is a dielectric material that insulates the inside of the cell chemically, and thus also electrically, from the exterior and that acts as a charge separator. However, the membrane possesses ion channels that allow some ions to flow out, hence the leakage [14]. The capacitance of the cell membrane is commonly estimated to be approximately $1 \mu F/cm^2$ [5].

A common electrical equivalent model of the cell is shown in figure 4.1.3.

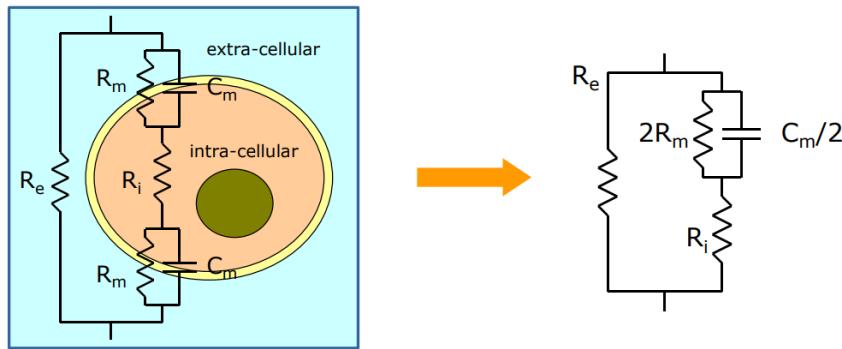


Figure 4.1.3: Equivalent circuit model for a cell. From [18].

The model on the left is explanatory in nature, it shows how injected current may either flow through the conducting medium electrolyte that surrounds the cell (represented by R_e), or otherwise flow across the cell. If the current flows through the cell, it enters through the cell membrane, which is a parallel combination of the capacitance of the lipid double layer (represented by C_m) and the conductance of the transport channels embedded in the membrane (represented by R_m). The current must pass through the conducting intracellular liquid (represented by R_i) and flow out by crossing the cell membrane ($R_m||C_m$) a second time [18].

The circuit can be simplified to yield the equivalent circuit on the right. This circuit describes a single Debye dispersion (assuming standard passive components) in parallel with a conductance, which does not take into account the likelihood of a distribution of time constants due to homogeneity. However, by replacing the membrane capacitance with a CPE, the distribution of time constants is accounted for, as the model now can be described with help of the ColeZ equation presented in section 3.2.2.

4.2 Impedance Monitoring of Cell Cultures

The traditional way to characterize cell cultures is through biochemical techniques that observe changes in gene and protein expression. Many of these techniques are destructive to the cells, which hinders the tracking of a population of cells undergoing biological processes over time.

Alternatively, cell cultures may be tracked over time through non-destructive assessments of their physical state. For instance, when cells proliferate, the increased amount of cells is reflected in a general increase in mass. When cells differentiate, their morphology changes. Changes in morphology can be changes in size, shape, or structural features of the cells in question [20].

The physical changes that the cells display when they proliferate or undergo differentiation have an impact in their electrical properties, a reason why observed changes in their electrical properties can be correlated to biological phenomena, with the advantage that this is a nondestructive method.

Cells may be cultured two-dimensionally on a flat structure, or three-dimensionally. 2D cultures do not emulate the conditions prevailing in living tissue faithfully in every respect. The cell morphology in particular becomes more flattened than it would in living tissue. Cells display greater mobility, proliferate more, and their natural cell-to-cell interactions are disturbed [3]. It is believed that 3D cell cultures may enable better reproduction of factors related to tissue organization, while facilitating the collection of spatially distributed data [13]. However, 2D cell cultures still express specialized functions, which is why they are still an advantageous way to examine the behaviour of individual cell types.

The correct terminology for cell growth on electrodes is adhesion. Adhesion in general refers to the way cells form mechanical contacts with their environment; in tissue this may be their anchoring to other cells or to the extracellular matrix. In cell culture applications where cells are grown on electrodes, it is necessary for them to properly adhere to the electrodes in order for them to express specialized cell functions [3].

2D cell cultures are particularly well suited to study the development of cell-electrode (also known as cell-substrate) interactions. The reason for this is that the measured impedance increases as cell membranes adhere to the electrodes and the free electrode area decreases. In particular, the real part of the impedance increases, as there is less space where free ionic conduction may take place. The capacitance is directly proportional to the covered area, as illustrated by equation 2.17, but reactive impedance is inversely proportional to the value of the capacitance (eq. 2.16), and thus the imaginary component decreases [3].

Having chosen the physical arrangement of the cell culture, we may choose to conduct measurements at a given frequency, and monitor how the impedance at the chosen frequency changes over time. This is known as electrical cell-substrate impedance sensing (ECIS), and is in fact the first technique that was applied to demonstrate the potential that impedance measurements offer in providing real-time quantitative data regarding cell-substrate interactions [21].

Alternatively, we may sweep the excitation signal over a given frequency range to obtain impedance spectra. This approach will of course provide more information regarding the interaction of the cell culture with the electrode surface over time and its response to external factors such as the introduction of cellular effectors or chemicals that affect cell morphology.

Part II

Application Development

Chapter 5

Red Pitaya STEMlab Platform

5.1 Overview

The Red Pitaya STEMlab platform is an open-source software instrumentation platform that can be configured for a wide range of measurement tasks. The hardware of the platform is proprietary, and the PCB itself has been given the official name *STEMlab* by the Red Pitaya company, although it is usual for users to call the STEMlab board *a red pitaya*. The platform will henceforth be referred to as *Red Pitaya STEMlab*, *STEMlab* or simply *Red Pitaya* interchangeably.

The central component of the Red Pitaya is a Xilinx Zync 7010 system-on-chip (SoC) device. The defining feature of this device is that it contains a dual core ARM Cortex-A9 processor and a traditional Field Programmable Gate Array (FPGA). The ARM processor is capable of running full operating systems [22], and the Red Pitaya uses this feature to run a custom Linux operating system which is installed on an SD card.

Additionally, the STEMlab board features two ADCs and two DACs which operate at 125 MSPS. These are the interface between the SoC processing system and the Red Pitaya fast analog I/Os, and it is this unique combination of data-acquisition and processing components that gives the Red Pitaya its versatility as an instrumentation tool.

The STEMlab is available in two versions, STEMlab 125-10 and STEMlab 125-14. The main difference between these versions is the technical specification of the high-frequency inputs and outputs; the 125-10 version features 10-bit ADC and DAC resolutions, whereas the 125-14 version offers 14-bit ADC and DAC resolutions [23].

In order to achieve better measurements, the STEMlab 125-14 was chosen for this master project, and further discussion of hardware will refer to the STEMlab 125-14 hardware specification.

STEMLAB 125-14	
Processor	Dual Core ARM Cortex A9
FPGA	Xilinx Zynq 7010
RAM	512MB
BRAM	3.8MB
System memory	Micro SD (4-32GB)
Console	Micro USB
Power connector	Micro USB
Power consumption	5V, 2A max

Table 5.1.1: Basic Specifications

5.2 Hardware Specifications

This section gives an overview of the hardware features and specifications of the STEMlab Red Pitaya platform. Figure 5.2.1 shows the top view of the Red Pitaya and identifies the position of the components and features of the board.

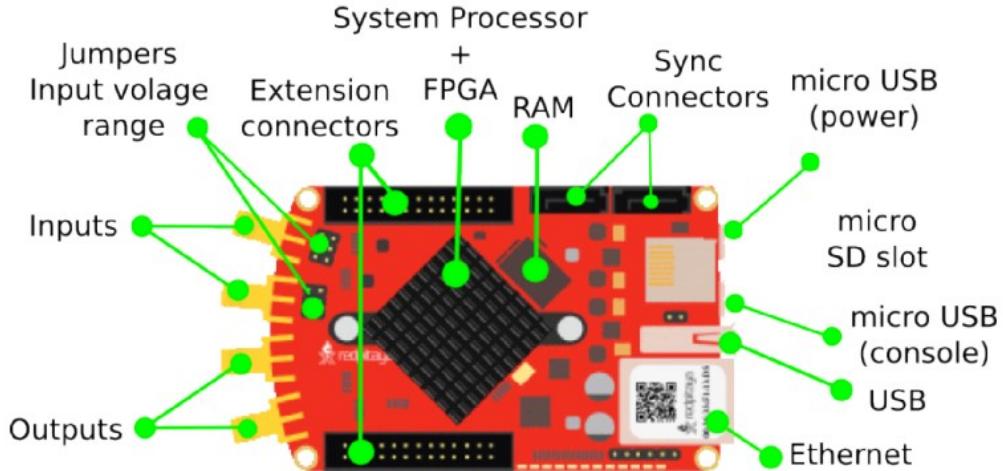


Figure 5.2.1: STEMlab board. From [23].

The board is powered with a 5V/2A micro USB power supply connected at the outer micro USB port. The inner micro USB port can be used for console communication, but normally all communication with the Red Pitaya system can be done through the Ethernet connection.

Connection between host PC and STEMlab

The standard way of connecting and using the Red Pitaya is by connecting the board to the LAN network through a router with DHCP settings enabled, using the depicted Ethernet port. **This is not possible at the University of Oslo** due to networking regulations which forbid connecting any devices other than PCs with UiO custom installations of either Windows or Linux to the wired network. Therefore, to connect the Red Pitaya to a PC in the university premises it is necessary for the PC to have an extra Ethernet card installed, so as to connect the board directly to this extra Ethernet card.

Furthermore, the PC needs to have software installed to perform DHCP server functions, otherwise the Red Pitaya needs to be configured with a static IP address. Typically it will be easier to install and configure DHCP server on a Linux based PC than on a Windows PC. Furthermore most development tools for the Red Pitaya are documented only for Ubuntu. For this project a personal laptop was set up with Ubuntu 14.04.03 LTS due to the added flexibility of being an administrator of the development machine.

5.2.1 Fast Analog IOs

The on-board ADC IC is the Linear Technology LTC2145-14 and the DAC IC is the Analog Devices AD9767ASTZ. Both datasheets are referenced as [24] and [25] respectively. The chips sit under the heatsink along with the Zync SoC.

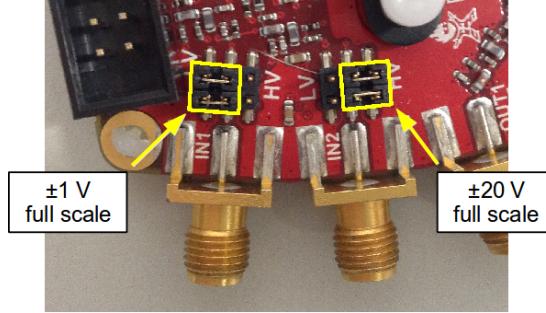


Figure 5.2.2: Jumper settings for input voltage range. From [23].

Table 5.2.1 gives an overview of the RF IO specifications. The voltage range of the RF inputs can be set by changing the position of jumpers in front of the input SMA connections to the positions shown in figure 5.2.2. Additionally, it is possible to connect the two central pins with a single jumper to achieve high input impedance setting. This is required when taking impedance measurements using the LCR meter and impedance analyzer.

RF Inputs		RF Outputs	
RF input channels	2	RF output channels	2
Sample rate	125 MS/s	Sample rate	125 MS/s
ADC resolution	14 bit	DAC resolution	14 bit
Input impedance	1MOhm/10pF	Load impedance	50 Ohm
Full scale voltage range	\u00b11V (LV) \u00b120V (HV)	Voltage range	\u00b11V
Max. Input voltage range	30V	Output slew rate	200V/us
Input ESD protection	Yes	Short circuit protection	Yes
Overload protection	Protection diodes	Connector type	SMA

Table 5.2.1: RF IO specifications [23].

5.2.2 Extension Connectors

Figure 5.2.3 shows the pinout of the extension connectors of the Red Pitaya. Extension connector E1 provides 16 digital IOs, one of them (DIO0_P) can be used as an external trigger for data acquisition. Extension connector E2 includes pins for I2C, UART and SPI communication, as well as the possibility to interface an external ADC clock and analog outputs and inputs. The analog outputs and inputs refer to the DACs and ADCs embedded in the Zync system and are utterly inferior, with a sampling rate of 100 kS/s and 12 bit resolution. The analog inputs have a voltage range of 0 to 3,5 V and the outputs have a range of 0 to 1,8 V [23].

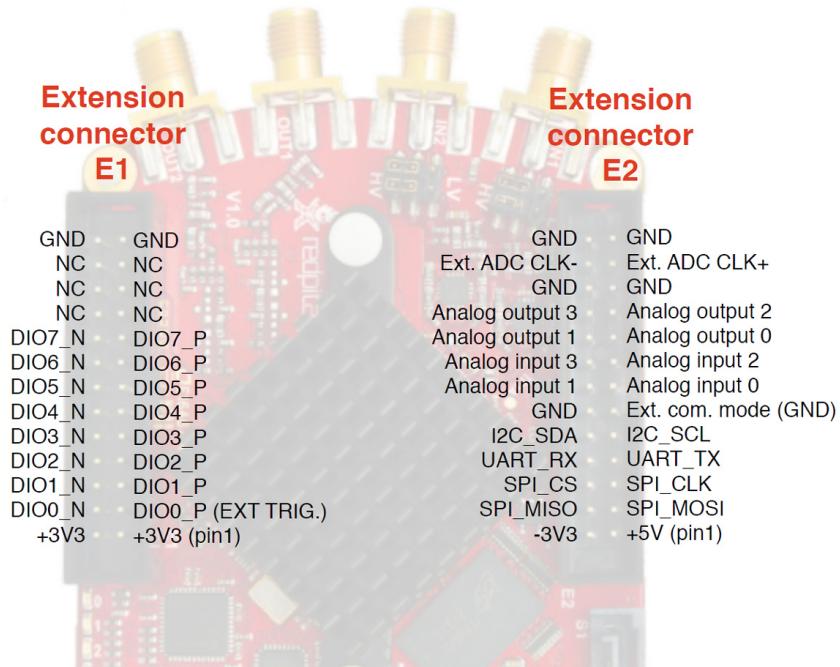


Figure 5.2.3: Extension connectors. From [23].

5.2.3 Official LCR meter application

The Red Pitaya company sells firmware and hardware for an official LCR meter application. The aforementioned hardware is an extension board, whose design is also proprietary. The extension board is shown in figure 5.2.4.

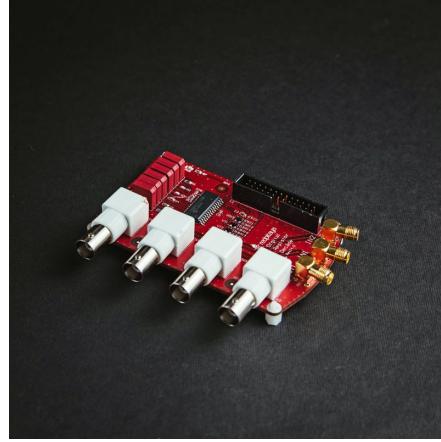


Figure 5.2.4: Extension Board for LCR meter.

An LCR Meter is a type of electronic test equipment that is used to measure the inductance (L), capacitance (C), and/or resistance (R) of a component. These devices are not designed for impedance spectroscopy, but to test components against their specifications, and will typically only have a few test frequencies to choose from [5].

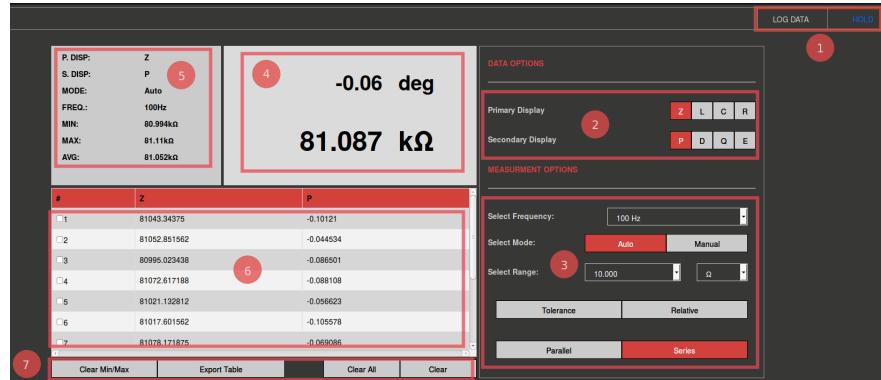


Figure 5.2.5: LCR Meter Panel.

The LCR meter can be used to take measurements of impedance data at 4 fixed frequencies: 100 Hz, 1 kHz, 10 kHz and 100 kHz. It is therefore not capable of doing frequency sweeps and not suitable for this project, however it is still interesting to look at the performance of the application at these frequencies to have an idea of what can be achieved through tailored development.

The LCR meter can be used to log measured data and export it to .csv format. The data can then be furthered processed in MATLAB.

Measurement of Capacitance at 100 Hz

Figure 5.2.6 shows two histograms displaying the results obtained when measuring a 1 uF film capacitor at 100Hz, and logging for 203 samples.

The mean value of the capacitance is $1.094 \mu\text{F}$ with a standard deviation of $1.43 \cdot 10^{-9}$. The mean value of the phase is -90.139 degrees and the standard deviation is 0.06.

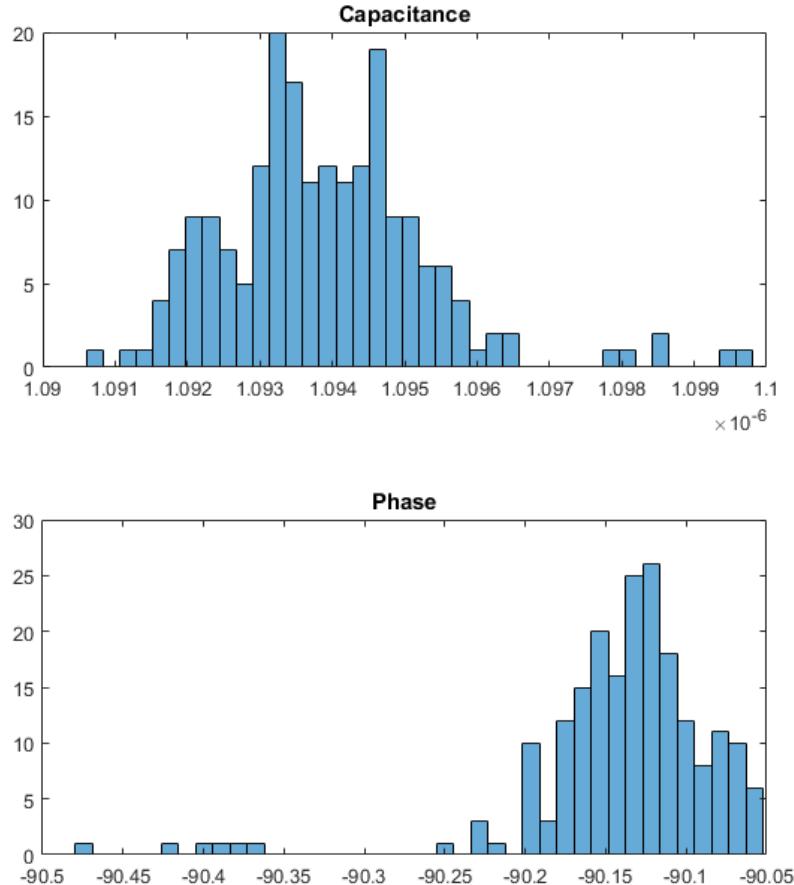


Figure 5.2.6: Histogram of capacitance and of a 1 uF film capacitor at 100 Hz.

Measurement of Capacitance at 10 kHz

At 10kHz, the performance of the measurement is significantly degraded. With 200 samples, the capacitance is measured to have mean value of $1.894 \mu\text{F}$ with a standard deviation of $3.5 \cdot 10^6$. The phase is measured to have a mean value of -84.015 degrees but with a standard deviation of 31.72.

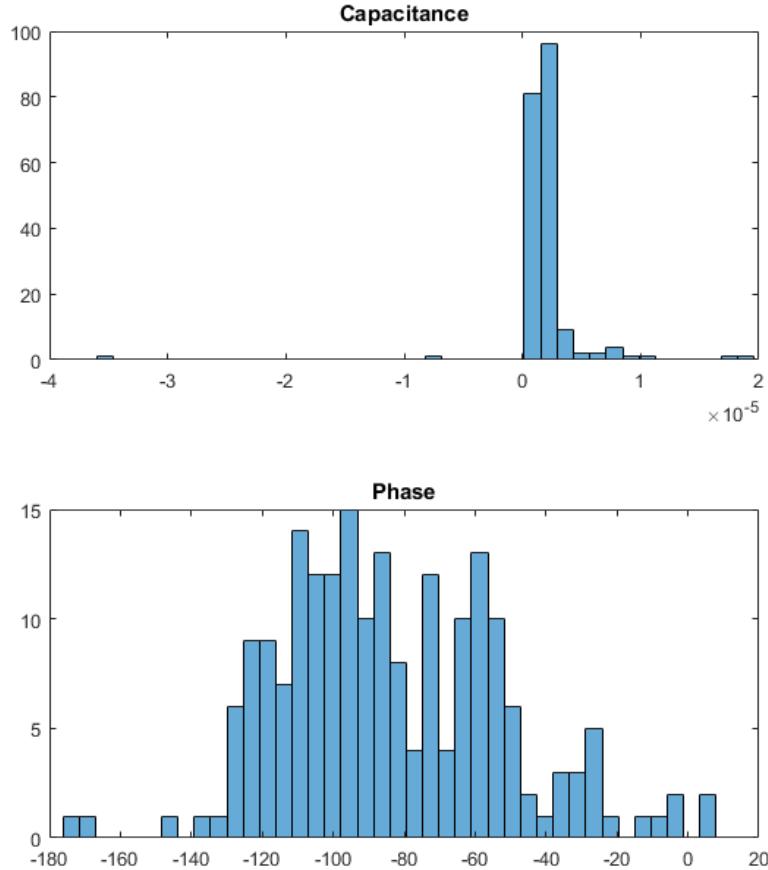


Figure 5.2.7: Histogram of capacitance and phase of a $1 \mu\text{F}$ film capacitor at 10 kHz.

Chapter 6

Software Development

6.1 Application Development using NGINX server

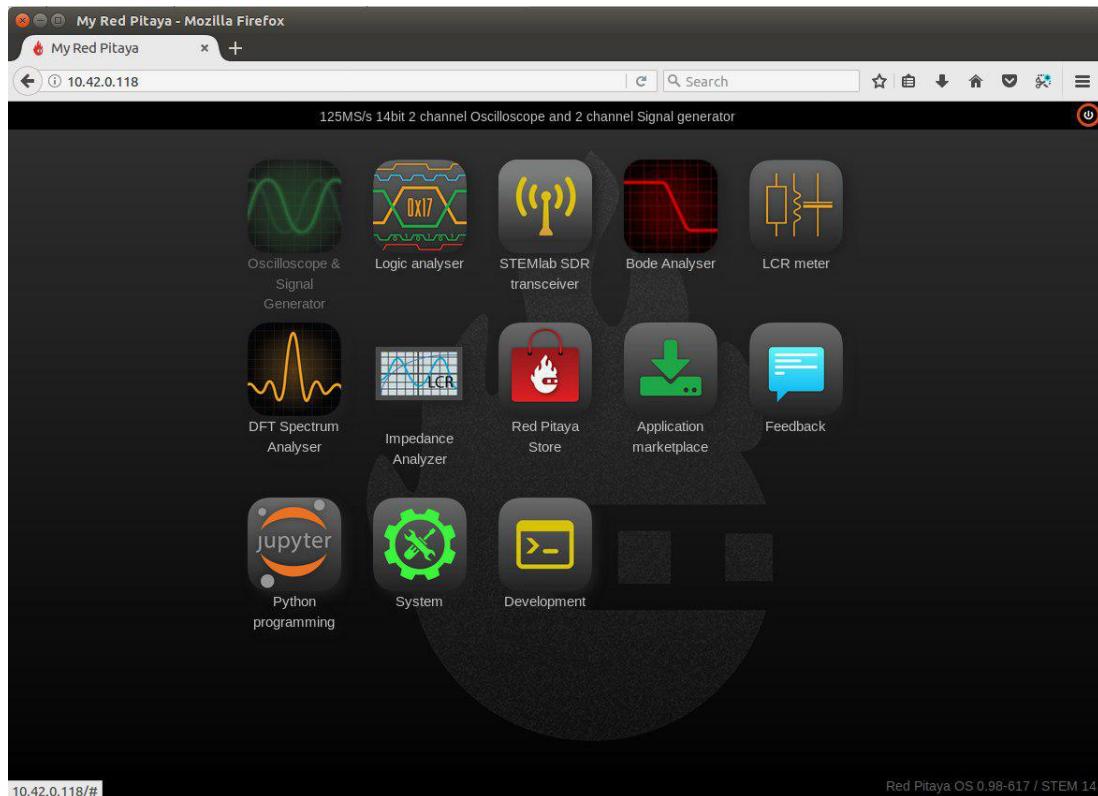


Figure 6.1.1: Screenshot of Red Pitaya Dashboard

The Red Pitaya board runs an Ubuntu Linux build that is booted from the SD card. The Red Pitaya team has developed the system aiming for development to be directed towards making applications with GUI interfaces in a browser, or Web UIs. Figure 6.1.1 shows the Red Pitaya dashboard as seen through the web browser.

This front-end variant relies on an NGINX server that serves as the bridge between the back-end of the board and the Web UI front-end. The NGINX server is used because it allows for modules to be loaded in runtime without need to restart the system, taking care of jobs such as loading the right FPGA image, as well as the necessary APIs, for a given application [23].

The Red Pitaya documentation [23] includes a tutorial called "*Creating first app*" at subsection 3.2.1 that gives a concise overview of the different file sources that must be included to make a software application with Web UI for the Red Pitaya. The current subsection is based on the aforementioned tutorial, but attempts to structure the information in a way that gives a better overview of the application project structure.

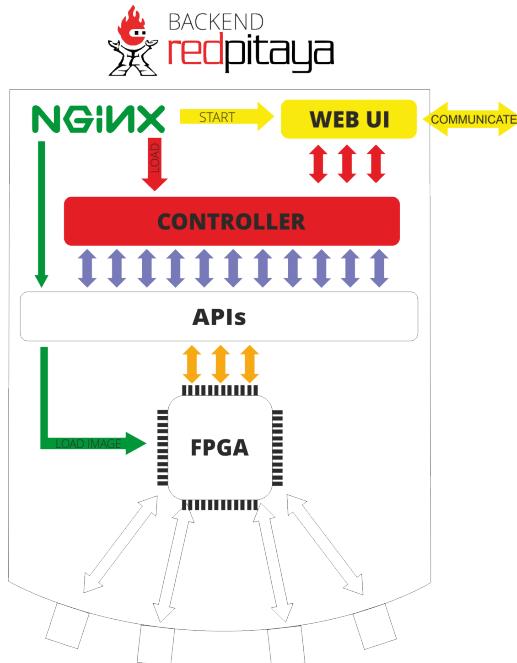


Figure 6.1.2: Red Pitaya Backend with NGINX. From [23].

Whenever the icon of an application is clicked on on the Red Pitaya Dashboard (see figure 6.1.1), the NGINX server will:

- Open the user interface.
- Load the associated FPGA image.
- Load the application controller.

All files necessary for NGINX applications to work are found under `/opt/redpitaya/` in the RedPitaya filesystem. This location can be navigated to through a terminal that has established SSH connection to the Red Pitaya. Eg. `$ ssh root@10.42.0.118` with default password "root".

The source files required for any given application must be located at `/opt/redpitaya/www/apps`. The contents of a plausible template application at this location are outlined below.

1. `css/`
 - `style.css`
2. `info/`
 - `icon.png`
 - `info.json`
3. `js/`
 - `app.js`
 - `jquery-2.1.3.min.js`
4. `src/`
 - `main.cpp`
 - `main.h`
 - `Makefile`
5. `fpga.conf`
6. `index.html`
7. `MAKEFILE`

The `style.css` file (1) and the `index.html` (6) file produce together the structure and layout of the web-UI.

The files under subfolder `/info` (2) provide the application icon and basic information in JSON format about the application respectively. This basic information is the name displayed below the icon in the Red Pitaya dashboard and the description that appears at the top of the screen when hovering above the icon with the mouse pointer.

The files under subfolder `/js` (3) are JavaScript files that establish WebSocket connection with the Red Pitaya. The Websocket protocol is used to enable persistent TCP¹ connection between server (in this case, the NGINX server) and client (the Web UI) so that data can be exchanged at any time without the server needing a prompt from the client to pass data. This protocol therefore allows for dynamic content updates [27].

The back-end files are found under subfolder `/src` (4), and is the C/C++ application that interacts with the Red Pitaya hardware and peripheral. These source files are compiled into the controller conceptually depicted in figure 6.1.2. Several mandatory functions that are called by the NGINX server are listed in the Red Pitaya documentation [23] section 3.2.1.2.3, including functions to initialize the application, set parameters, update signals, and exit the application.

The `fpga.conf` (5) file defines which FPGA image is loaded when an application is started. The FPGA image must be located in `/opt/redpitaya/fpga`.

¹TCP stands for Transmission Control Protocol and it is a network protocol that lets two hosts connect and exchange data streams, also guaranteeing delivery of data and packets in the order they are sent [26].

6.2 Development with Red Pitaya API

The Red Pitaya API can be used to control generation and acquisition of signals from software, as well as the on board LEDs and the GPIO.

The API may be used in an application developed using the NGIX server flow described in the previous section. This approach has been discarded as a suitable approach for this project due to the fact that it presupposes knowledge of web-development that is not compatible with the projects background in electronics development.

It is also possible to use the API directly in C programs that can be compiled on the operating system of the Red Pitaya and made executable.

In order to use this approach the right FPGA image must be loaded using the command line:

```
$ ssh root@10.42.0.118  
$ cat /opt/redpitaya/fpga/fpga_0.94.bit > /dev/xdevcfg
```

To copy the C program file to the Red Pitaya OS, from the command line of the Ubuntu development PC:

```
$ scp name.c root@10.42.0.118:/tmp/name.c
```

Then, to compile, make executable and run:

```
$ cd /tmp  
$ gcc -o name name.c  
$ ./name
```

6.3 Remote Control with SCPI server

The API functions have equivalent functions described for use with the SCPI server, allowing for the same software control functionality described in the APIs for generation and acquisition of signals. The SCPI server commands can be called from programs such as MATLAB or Labview. MATLAB establishes connection to the Red Pitaya through the TCP interface. When the SCPI server is running, the NGINX server must be disabled, as both are not compatible.

In order to enable the SCPI server, the following commands must be run on the red pitaya OS:

```
$ systemctl stop redpitaya_nginx
$ systemctl start redpitaya_scpi
```

Alternatively, the SCPI server can be started by navigating to the SCPI server application in the Red Pitaya dashboard.

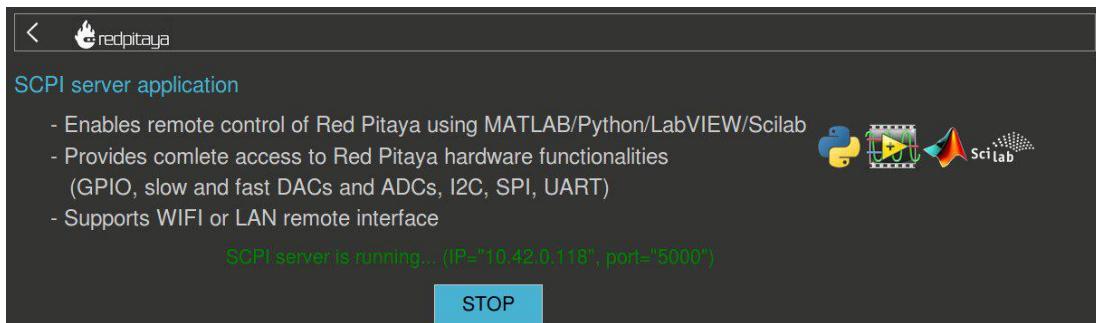


Figure 6.3.1: SCPI application in the Red Pitaya dashboard

The following subsections will discuss how to use the SCPI server functions with MATLAB.

6.3.1 Signal Generation

Listing 6.1 is a basic example showing how to set up continuous signal generation at the RF output of the Red Pitaya.

```

1  %% Define Red Pitaya as TCP/IP object
2  IP= '10.42.9.118';                                % IP of Red Pitaya
3  port = 5000;                                       % Port of Red Pitaya
4  tcpipObj=tcpip(IP, port);                         % Declare TCP/IP Object
5  %% Open connection with Red Pitaya
6  fopen(tcpipObj);                                  % Open Connection to SCPI server
7  tcpipObj.Terminator = 'CR/LF';                     % Set terminator value
8
9  %% Set Output Signal Parameters
10 fprintf(tcpipObj,'GEN:RST');                      % Reset generator to default settings
11 fprintf(tcpipObj,'SOUR1:FUNC SINE');               % Set function type
12 fprintf(tcpipObj,'SOUR1:FREQ:FIX 2000');          % Set frequency
13 fprintf(tcpipObj,'SOUR1:VOLT 1');                  % Set amplitude
14 fprintf(tcpipObj,'OUTPUT1:STATE ON');              % Set output to ON
15 %% Close connection with Red Pitaya
16 fclose(tcpipObj);
```

Listing 6.1: Basic Signal Generation Example [23].

In order for MATLAB to establish connection to the SCPI server, a TCP/IP client must be set up with the correct IP address and port of the Red Pitaya board. The default port is 5000, but both the IP address and port number can be found by navigating to the SCPI server application in the Red Pitaya dashboard, shown in figure 6.3.1.

In line 7, the terminator value for the communication between the TCP/IP client and the SCPI server is configured. When performing an fprintf function, all occurrences of \n are replaced with the configured terminator value. In order to pass Red Pitaya specific SCPI commands, fprintf is used in lines 10 through 14. There are no explicit instances of \n in the example, however, in MATLAB, the default format for fprintf inserts an \n at the end of every call to the function. This default line terminator is then replaced by the new terminator configuration.

Since the terminator is set to CR/LF, the terminator is a carriage return followed by a line feed, the corresponding code is \r\n.

The SCPI commands specific to signal generation are called in lines 10 through 14. Notice that all SCPI commands are passed through the MATLAB TCP/IP object as strings. The possible function type parameters are: SINE, SQUARE, TRIANGLE, SAWU, SAWD, PWM, ARBITRARY, with SINE as default.

Other SCPI commands are available to enable for example pulse-width-modulated output with programmable duty cycle or triggered signal generation, and can be found in the Red Pitaya documentation.

6.3.2 Signal Acquisition

Listings 6.2 through 6.5 demonstrate how to achieve on-trigger signal acquisition.

```

1 %% Define Red Pitaya as TCP/IP object
2 IP= '10.42.9.118'; % IP of Red Pitaya
3 port = 5000; % Port of Red Pitaya
4 tcpipObj = tcpip(IP, port); % Declare TCP/IP Object
5 tcpipObj.InputBufferSize = 16384*32; % Declare Buffer Size
6 %% Open connection with Red Pitaya
7 fopen(tcpipObj);
8 tcpipObj.Terminator = 'CR/LF';
9 %% Flush Buffers
10 flushinput(tcpipObj);
11 flushoutput(tcpipObj);

```

Listing 6.2: Signal Acquisition Example Part 1

The TCP/IP object is set up in the same way as for signal generation, but additionally a buffer size attribute must be declared (line 5). The depth of the Red Pitaya buffer is 16384, and the word length for signal acquisition is 32 bit. The buffers should be flushed before acquisition is started.

Decimation	Sampling Rate	Time Scale
1	125 MS/s	131.072 us
8	15.6 MS/s	1.049 ms
64	1.9 MS/s	8.389 ms
1024	122 kS/s	134.22 ms
8192	15.2 kS/s	1.074 s
65536	7.6 kS/s	8.590 s

Table 6.3.1: Decimation Factors for Signal Acquisition

The acquisition configuration parameters are set in listing 6.3. The decimation parameter set in line 4 defines the sampling rate, and should be selected according to the Nyquist-Shannon sampling theorem in order to avoid aliasing. The possible decimation values and their corresponding sampling rates are given in table 6.3.1.

For long acquisition intervals it may be desirable to turn averaging on. This will enable the calculation of the average of the samples, improving the signal-to-noise ratio but impairing detection of high frequency peaks that may appear. To do this, the string passed in line 6 must be changed to 'ACQ:AVG ON'.

In line 12 the gain is configured to match the physical jumper settings for high or low voltage (see figure 5.2.2).

```

1 % Reset acquisition to default settings
2 fprintf(tcpipObj,'ACQ:RST');
3 % Set decimation vale (sampling rate) in respect to you acquired signal frequency
4 fprintf(tcpipObj,'ACQ:DEC 1');
5 % Disable averaging
6 fprintf(tcpipObj,'ACQ:AVG OFF');
7 % Set trigger level
8 fprintf(tcpipObj,'ACQ:TRIG:LEV 0');
9 % Set trigger delay to 0 samples
10 fprintf(tcpipObj,'ACQ:TRIG:DLY 0');
11 % Set gain to high voltage (HV) or low voltage (LV)
12 fprintf(tcpipObj,'ACQ:SOUR1:GAIN HV');
```

Listing 6.3: Signal Acquisition Example Part 2

The trigger options are set in lines 8 and 10 of listing 6.3. The trigger level must be specified in milivolts. The trigger delay (t_{dly}) determines how the buffer fills with samples in relation to the trigger. A trigger of 0 will set the triggering sample to the center of the buffer, and the resulting buffer content will have $\text{buffer_depth}/2 = 16384/2 = 8192$ samples acquired before the trigger condition was met and the same amount samples acquired after the trigger condition was met.

The way this is achieved is as follows: As soon as acquisition starts, the circular buffer will start filling with samples regardless of whether a trigger condition is met or not. When the trigger condition is met, a countdown will start from $t_{dly} + \text{buffer_depth}/2$. When the counter reaches 0, acquisition is stopped and data can be read from the buffer.

Acquisition is started by passing the 'ACQ_START' command. A pause is then required to fill the buffer with fresh samples. The minimum necessary length of the pause can be determined from the time scale corresponding to the decimation factor.

Acquisition must be started before configuring the trigger source. In line 6 of listing 6.4, acquisition is configured to trigger on channel 1, positive edge. The available parameter options for trigger configurations are:

- NOW - Triggers Immediately
- CH1_PE\CH1_NE - Channel 1, Positive or Negative Edge
- CH2_PE\CH2_NE - Channel 2, Positive or Negative Edge
- EXT_PE\EXT_NE - External Trigger, Positive or Negative Edge
- AWG_PE\AWG_NE - Trigger from Generator, Positive or Negative Edge

```

1 %% Start Acquisition Functionality
2 fprintf(tcpipObj,'ACQ:START');
3 pause(1)
4
5 % Set trigger to Source 1, Positive Edge
6 fprintf(tcpipObj,'ACQ:TRIG CH1_PE');
7 % Wait for trigger detection
8 while 1
9     trig_rsp=query(tcpipObj,'ACQ:TRIG:STAT?')
10    if strcmp('TD',trig_rsp(1:2))
11        break
12    end
13 end

```

Listing 6.4: Signal Acquisition Example Part 3

The loop in lines 8 to 13 will prevent the script from attempting to read the buffer before the trigger condition is met. The status of the trigger is read at every iteration of the loop on line 9, and the returned string is subsequently compared to '*TD*', which stands for *trigger detected*. The alternative state is '*WAIT*'. When the string comparison succeeds the loop breaks, and the script may continue to reading the data from the buffers.

```

1 % Format Read Data
2 fprintf(tcpipObj,'ACQ:DATA:UNITS VOLTS');
3 fprintf(tcpipObj,'ACQ:DATA:FORMAT ASCII');
4
5 % Read data from buffer in string format
6 signal_str_1=query(tcpipObj,'ACQ:SOUR1:DATA?');
7 signal_str_2=query(tcpipObj,'ACQ:SOUR2:DATA?');
8
9 % Convert values to numbers.
10 signal_num_1=str2num(signal_str_1(1,2:length(signal_str_1)-3));
11 signal_num_2=str2num(signal_str_2(1,2:length(signal_str_2)-3));
12
13 %% Close TCP/IP connection
14 fclose(tcpipObj)

```

Listing 6.5: Signal Acquisition Part 4

Finally in listing 6.5 the buffer data is collected. First the data format is set in lines 2 and 3. The available units are RAW and volts VOLTS, and the available formatting is FLOAT and ASCII. The default format is FLOAT, but since the values are returned as strings through the TCP/IP client it is more convenient to have them returned as ASCII characters and converted to numbers using the MATLAB str2num function. This function converts a character array to a numeric matrix and is well suited to translate the acquired values to a MATLAB array for further processing. The string containing the values starts always with a '{' character and finishes with three blank spaces and a '}', which is why these are removed from the acquired string when they are passed to the str2num function. ø

Chapter 7

Embedded System Development

The versatility of the STEMlab board comes from the programmable Zynq System-on-Chip and its on-board processing resources. In the context of this project, it was desirable to modify the firmware of the SoC to better take advantage of the processing capability of the system. In particular, the flexibility to load new FPGA binaries during run-time was investigated. A promising application of this approach could be to provide hardware acceleration for the calculation of bioimpedance data.

Embedded system development using the Xilinx design tools is highly modularized, focusing on an intellectual property (IP) design flow. This means that the tool is optimized to work with reusable design blocks that are interconnected at the project level. The Red Pitaya company provides little support for project-based design on the Zynq-SoC FPGA. Modifications to the firmware can be done by editing their source files, however, modifying the source files is only practical when performing small changes. To make significant changes, like adding or removing complex IP to the FPGA architecture, it is desirable to use the project GUI.

The implementation of custom firmware proved to be too time consuming for the scope of this project, however, a significant amount of time was devoted to exploring the development environment. The results of this investigation are layed out in this chapter, focusing on IP that is useful to produce an embedded system optimized for processing impedance measurement data.

7.1 Development Tools and Project Set-up

The development tool necessary to produce binaries for the Zynq-SoC FPGA is the Vivado Design Suite. Since the STEMlab board is a third party PCB, it is not directly supported by Vivado and therefore special consideration must be taken to include the necessary modules at the project creation time to ensure that the board will function correctly.

Instructions on how to install the development tools on a development machine running Ubuntu 16.04 LTS are included in the appendixes.

The Red Pitaya company has a Github [28] repository providing the sources they use to build the first stage bootloader ELF file, the memory test ELF file, the Linux device tree and the FPGA bit file.¹ This sources are tailored to regenerate the existing design using a Makefile

¹ELF file stands for "Executable and Linkable Format".

included in the repository under the /fpga subfolder. Modifying just one part of the system using this approach is difficult without in-depth knowledge of the whole system. For this reason, community users Pavel Denim [29] and Anton Potocnik [30] have created a development approach that permits working on a project level in the Vivado GUI.

A powerful feature of the Red Pitaya is that the NGINX server enables loading the FPGA image during runtime. Therefore FPGA development can be pursued without having to deal with the other parts of the system that require more in-depth knowledge of Linux systems. Binaries for an FPGA image can be generated with Vivado, transferred to the Red Pitaya Linux system using a SSH connection, and subsequently be loaded from the OS.

The approach devised by the above mentioned collaborators allows one to create projects in Vivado that contain the necessary configuration for the STEMlab boards and also make it possible to add or remove IP from a project in a transparent way. The general folder setup is outlined below:

```

1. /cfg
    1.1 - clocks.xdc
    1.2 - ports.tcl
    1.3 - ports.xdc
    1.4 - red_pitaya.xml
2. /cores
    2.1 - /axis_red_pitaya_adc_v_1_0
        2.1.1 - axis_red_pitaya_adc.v
        2.1.2 - core_config.tcl
    2.2 - /axis_red_pitaya_dac_v_1_0
        2.2.1 - axis_red_pitaya_dac.v
        2.2.2 - core_config.tcl
3. /projects
    3.1 - Sample_Project
        3.1.1 - /client
            - client.m
        3.1.2 - /server
            - server.c
        3.1.3 - basic_red_pitaya_bd.tcl
        3.1.4 - block_design.tcl
4. /scripts
    4.1 core.tcl
5. make_cores.tcl
6. make_project.tcl

```

Directory nr. 1, /cfg, contains configuration files that are used to set up the clocks and the IO, including the correct standard for the various IO of the SoC chip. All the files contained in this folder must be included and should not be edited.

Under directory nr. 2, /cores, a single subdirectory must be created for each new IP created for the Red Pitaya. Suitable examples can be the IP used to interface the on-board ADCs and DACs. Inside this subdirectory we include the core specific register transfer level (RTL) files and a tcl file that sets-up the core.

Directory nr. 3 is the /projects folder. Each project is given a unique name, that must be the same name as the subdirectory for higher level tcl scripts to be able to find the right file location. The contents of this directory are flexible, but the basic_red_pitaya_bd.tcl file must always be included. This file sets up the most basic parameters, such as the Zynq processing system and the physical pinout interfaces. The block_design.tcl file can be modified to load and connect any additional RTL that we wish to add to the project. RTL files must be located in the project subdirectory along with the tcl files. Verilog files will be added automatically, however, vhdl files should be listed separately.

```
# Load any additional Verilog files in the project folder
set files [glob -nocomplain projects/$project_name/*.v projects/$project_name/*.sv]
if [llength $files] > 0
add_files -norecurse $files
# Load additional VHDL files
add_files -norecurse [glob -nocomplain projects/$project_name/my_file.vhd]
```

The make_project.tcl file (6) is used to set the project name and subsequently run the tcl file that builds the project, block_design.tcl. The script dependencies for creating a project are outlined in figure 7.1.1.

```
set project_name "Sample_Project"
source projects/$project_name/block_design.tcl
```

The make_cores.tcl (5) will look through the /cores directory (2.) and run the tcl script that configures each individual core. The IP cores will then be added to the Vivado library. This file must be run from the Vivado command line separately before running make_project.tcl. Otherwise only those cores which have been explicitly created in either block_design.tcl or basic_red_pitaya_bd.tcl files will be compiled and added to the GUI core library.

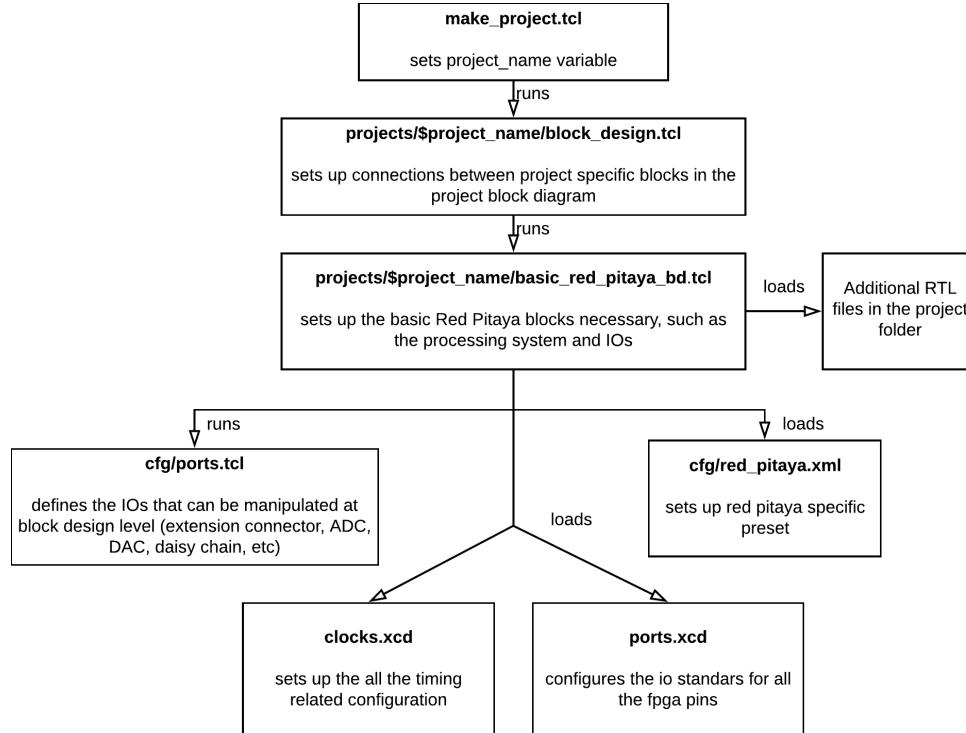


Figure 7.1.1: Script dependencies

7.2 Xilinx IP

The Vivado IDE is tailored for an IP-centric design flow. That means that the general project flow should focus on creating and verifying sub-system IP that is then added to the Vivado IP library and interconnected to yield the desired system result. The Vivado installation includes a significant IP library, with customizable IP that can be programmed into the FPGAs to reduce development time. Most of the Vivado IPs use the AXI4 communication bus protocol, which is the protocol used by both Xilinx soft processors and ARM hard processors.

AXI4 stands for Advanced eXtensible Interface, and it is a bus protocol originally developed for microcontrollers by ARM. The AXI4 protocol is part of the ARM AMBA® 3.0 open standard, and may be used in a memory-mapped or streaming applications. In the context of Xilinx ZYNQ devices, it provides a specification for movement of data between IP.

7.2.1 Envisioned System Overview

Figure 7.2.1 shows and overview of all the components that must be in place in order to realize a bioimpedance measurement system. To the left we see the signal conditioning front-end and the device under test (DUT), which in this case is the electrodes and the cells. These are external to the STEMlab board and thus not part of the embedded system, but having them in the will aid the understanding of the signal flow. The embedded system proper is seen on the right. From the signal conditioning front-end, we generate and sample using the on-board fast DACs and ADCs. Reusable IP to control these components is available from Pavel Denim [29], and shown in blue. The physical interface of the signals from the front end to the embedded system is color-coded gray.

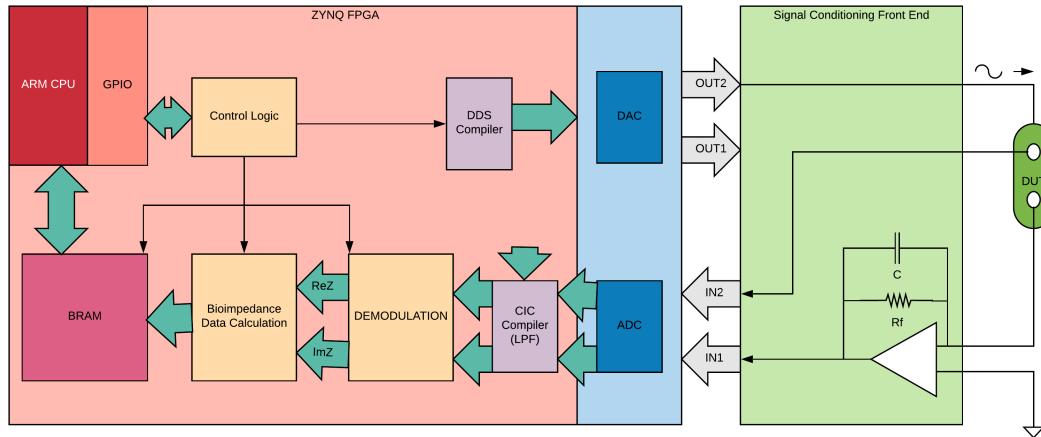


Figure 7.2.1: System Architecture

The hard ARM processor is shown in red. The general purpose input-output is shown in light-red, and is an interface between the software running on the processor and the programmable logic in the FPGA. The GPIO is a memory mapped interface, and is provided as configurable IP from Xilinx. This component provides registers that can be written to or read from the CPU in order to communicate with the hardware.

The blocks shown in light purple are components that can be accomplished using existing Xilinx IP. The pink block represents the temporary storage of the system, and it is accomplished through a mix of Xilinx IP and original IP. Blocks that must be designed and verified specifically for the project are shown in light orange. The green arrows indicate that data transfer must be performed using the AXI4 bus interface.

The general idea is to use a direct digital synthesizer in conjunction with one of the on-board fast DACs in order to generate the excitation signal. The excitation will be a voltage, and cause a current to flow through the device under test. This current will subsequently be converted to a voltage by the transimpedance amplifier and sampled with one of the fast ADC inputs. The excitation is measured at the remaining ADC input. Both the excitation and the response are subsequently low-pass filtered using the Cascaded Integrator-Comb (CIC) compiler block.

In the demodulation block, both the in-phase and quadrature components of the excitation and response are extracted and passed to a block that calculates the complex impedance values. The values are subsequently stored in the temporary memory and passed to the processor.

7.2.2 General Purpose I/O (GPIO) Core

The GPIO core implements a 32-bit general-purpose input/output slave interface to the AXI bus. It supports the AXI4-Lite interface, which differs from the full AXI4 bus in that all transactions are of burst length 1, meaning that only one beat or data word is transferred per transaction.

The core can be implemented with one or two channels, with variable length from 1 to 32 bits. The customization window is shown in figure 7.2.2.

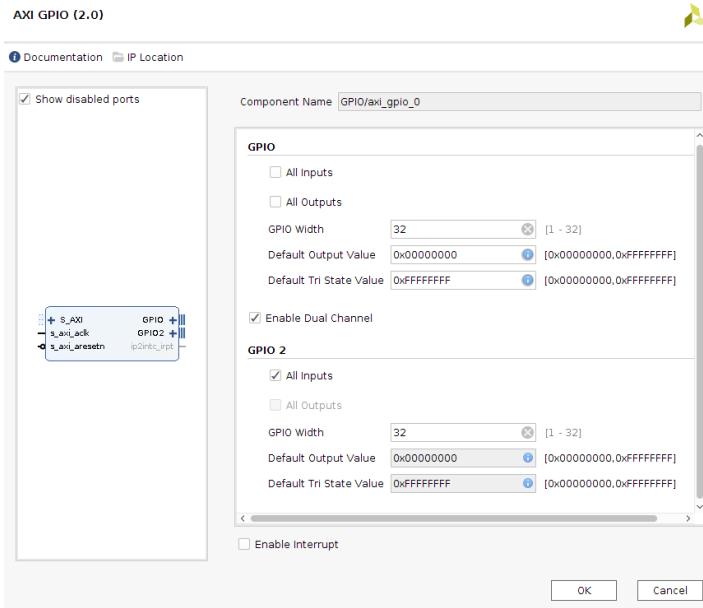


Figure 7.2.2: GPIO customization window

In figure 7.2.2, the core is configured with two 32-bit wide channels, GPIO and GPIO2. The former has both an input and an output register, whereas the latter is only configured to be an input. The architecture of the GPIO is shown in figure 7.2.3 to elucidate how I/O functionality is achieved in the core.

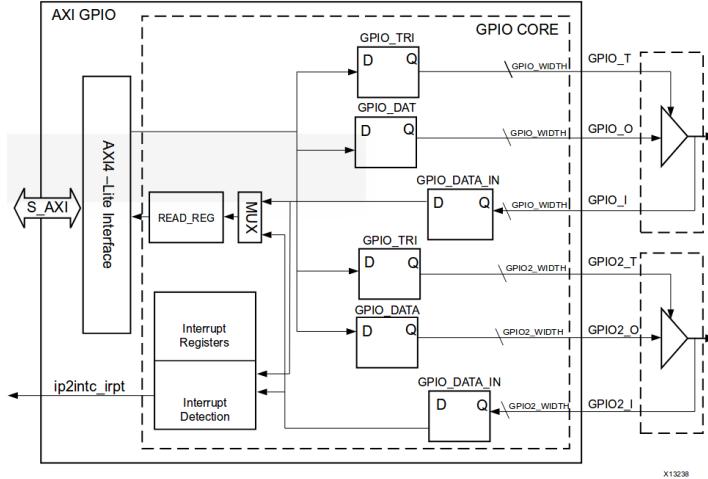


Figure 7.2.3: GPIO architecture block diagram. From [31].

The ports are configured for input or output by enabling or disabling the tri-state buffers to the right, which are the interface to the rest of the hardware. They can also be configured to generate interrupts when their inputs change value [31].

In the processing system, the core is memory mapped to a memory location and a range to make it accessible to software. Typically an application requiring communication to and from the processor will include a server program that is run on the board and a client program that communicates with the server.

The GPIO is a fundamental block of the system because it is a relatively simple interface between the processor and the control logic in the FPGA. It can be used to issue the start command for acquisition or to signalize to the processor that acquisition is complete and data needs to be read from memory.

7.2.3 Direct Digital Synthesizer (DDS) Core

The DDS core can be used to synthesize a sinusoids in conjunction with the on-board digital to analog converters. The method by which the core generates the input values for the DAC is based on a look-up table scheme.

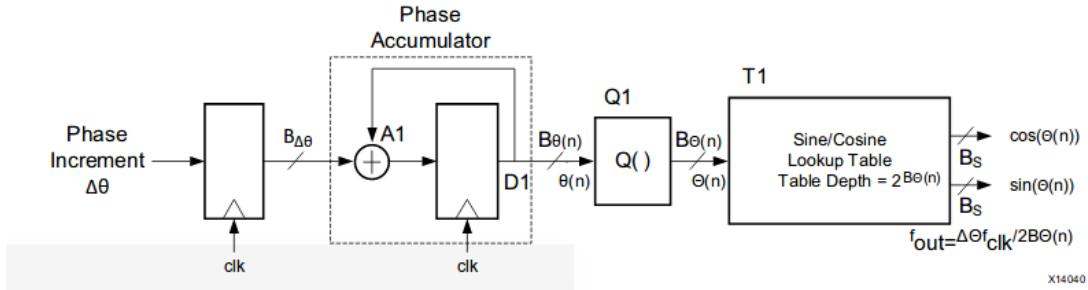


Figure 7.2.4: Simplified DDS Block Diagram. From [32].

A phase increment value is assigned to each clock cycle, and the phase accumulator computes a slope based on this increment. The slope is linear and corresponds to a ramp when plotted in time. Each point in the slope is mapped to sampled sinusoid values in the look-up table by the quantizer Q1. The quantizer takes the output from the phase accumulator and outputs the correct address input to the look-up table. When the ramp reaches its maximum value, a whole period of the sinusoidal signal has been achieved.

In the standard mode of operation, the output frequency is given by:

$$f_{out} = \frac{f_{clk} \Delta\theta}{2^{B_{\theta(n)}}} \quad (7.1)$$

Where f_{out} is the system clock frequency, $\Delta\theta$ is the phase increment value parameter, and $B_{\theta(n)}$ is the number of bits. With the Red Pitaya, the parameters f_{out} and $B_{\theta(n)}$ must be set to 125MHz and 14 bits respectively. Therefore, the output frequency is determined by the phase increment parameter.

The DDS core can be configured to have up to 16 channels that output a fixed frequency sinusoidal, or else using the AXI4-stream interface the phase increment value can be dynamically updated to achieve programmable frequency.

7.2.4 Cascaded Integrator-Comb (CIC) Core

The CIC filter core has been successfully implemented by Pavel Denim [29] to perform low-pass filtering in a Vector Network Analyzer (VNA) application based on the Red Pitaya. VNAs are typically used in radio-frequency applications to measure the network parameters of circuits, and involve measurement of complex impedance. For this reason, it has been highlighted as a suitable option to perform the required low-pass filtering of acquired input data, however, due to time constraints in the project, it has not been possible to examine the design.

7.2.5 Block Random Access Memory (BRAM)

Block Random Access Memory (BRAM) is RAM that is built from embedded memory blocks in the FPGA part of the SoC chip. Xilinx provides an IP, the Block Memory Generator Core, to instantiate BRAM in the FPGA. The BRAM may be instantiated as a single- or dual port RAM or ROM. The ports are AXI4 and AXI4-Lite compliant and independent, but both access a shared memory space. A dual-port BRAM block can be utilized to build a simple first-in-first-out (FIFO) memory, for example. The instantiated BRAM functions as a slave, and a hierarchy must be built around it to control reading and writing operations. This can be achieved through a Xilinx BRAM controller IP block, however communication with the BRAM from software using this approach has not been achieved.

Pavel Denim [29] has written several IP cores that can be combined to access the BRAM. These cores are used in several of his projects, which gives the advantage that it is possible to examine the software to infer how to communicate with the memory from the CPU. These projects are sparsely commented, so it is highly recommended to work through the example projects written by Anton Potočnik [33] before attempting to include or modify the source material developed by Pavel Denim.

Chapter 8

Red-Pitaya Based Bioimpedance Acquisition System

Initially, attempting to achieve a flexible and efficient embedded system solution with the Red Pitaya was favoured, in particular due to lack of access to cell culture equipment at the University of Oslo and due to lack of access to information regarding the electrodes chosen for the Training4CRM project.

These conditions hindered the possibility to address the design of a front-end solution at an early stage in the project, in particular because electrode specifications were first received three months before the project deadline. At this time, an opportunity to conduct cell measurements at the Technical University of Denmark (DTU) was also presented, for which a front-end prototype printed circuit board was quickly developed and used in conjunction with a programmatic acquisition approach on the Red Pitaya using the SCPI server interface.

8.1 Electrodes

The electrodes used at the Technical University of Denmark were pyrolytic carbon electrodes with silver contacts, as shown in figure 8.1.1. These electrodes were designed to be interfaced with Electrical Impedance Spectroscopy (EIS) and Cyclic Voltammetry instruments.

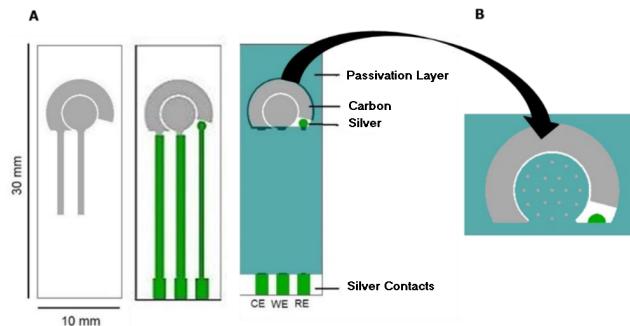


Figure 8.1.1: Carbon Microelectrodes

The system was developed to perform two-electrode bioimpedance measurements, therefore the reference electrode was not used. In EIS applications, the working electrode is designed to be the electrode at which the excitation is applied, therefore this was also the electrode used to apply the excitation signal during testing at DTU. The counter electrode was used to collect the response.

The working electrode is covered with an insulating passivation layer except for 19 circular openings of $250\mu\text{m}$, as shown in figure 8.1.1. The purpose of this is to increase the sensitivity in the working electrode and decrease the contribution of the counter electrode, because it increases the current density and thus the sensitivity of the electrode [34].

8.2 Front End Prototype

A simple custom front-end prototype for two-electrode measurements was manufactured. The front end features an AD8642-based transimpedance amplifier with variable feedback impedance. The AD8642 was selected due to its very low input bias current of maximum 1pA. Five traces in parallel, each with space for a single surface mount device (SMD) component, were added to the PCB layout. These traces may be included or excluded from the feedback using jumpers in order to achieve variable feedback impedance. The envisioned purpose of the variable feedback impedance was to be able to adjust the sensitivity and perform phase correction during prototype testing at DTU, however this was not possible due to lack of access to soldering equipment for SMD components.

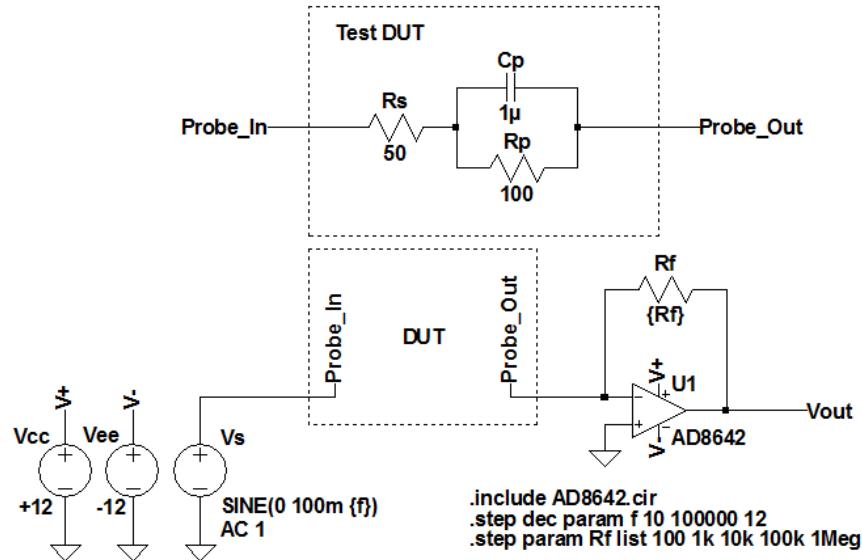


Figure 8.2.1: Analog Front End Schematic with Trans-impedance Amplifier

8.3 Acquisition Scripting in MATLAB

8.3.1 Signal Generation and Acquisition

In order to be able to monitor the proliferation of cell cultures, it is necessary to perform measurements at a regular interval during a suitably long period of time (in the order of days). Although single frequency measurements can be suitable to assess the progress of cell adhesion [21], an approach with frequency sweeps between 10 Hz and 100kHz was selected in order to obtain full impedance spectra at each measurement interval.

In order to provide the frequency sweep functionality, the scripts for signal generation and acquisition described in sections 6.3.1 and 6.3.2 were modified and declared as two MATLAB functions named generate and acquire. This function-based approach has earlier been implemented by Albert Ruiz Vargas et. al [35], whose work was used as a starting point, and who also improved the efficiency of the system by recompiling the SCPI server to include equivalent functions for use in a Red Pitaya based instrument tailored for gastroenterological bioimpedance spectroscopy.

The declarations for the generate and acquire functions follow below:

```
function void = generate(IP,port,chan,amp,freq,type,state, DC)
function [exc, resp] = acquire(IP,port,DEC,fc)
```

Both functions take the IP and port of the Red Pitaya to set up TCPI communication.

The generate function is a void function whose purpose is to configure the signal generator to produce the required stimuli. In practice it is only necessary to change the freq parameter, as the channel (chan), amplitude (amp), type, state and DC levels are all kept constant during the sweeps.

The acquire function returns two vectors, exc and resp. Both on-board acquisition channels are used, in order to obtain a measurement of the stimulus signal in vector exc, and a measurement of the voltage at the output of the transimpedance amplifier in vector resp.

As discussed in section 6.3.2, correct signal acquisition presupposes correct set-up of the decimation parameter DEC. Additionally, there is an input parameter fc, that must be set to the frequency at which measurements are being conducted. Both the DEC and the fc are used to compute the amount of buffer samples that must be read to obtain a whole of the acquired signal, using the calculation below.

```
fs = clk_f/DEC;
samps_T = fix(fs/fc);
```

In the first line, the sampling frequency fs of the system is calculated by dividing the system clock clk_f, which runs at 125MHz, by the current decimation factor. This sampling frequency is in line 2 divided by the current measurement frequency fc in order to obtain the number of samples samps_T that must be obtained in order to get a full cycle of the acquired signals, and the result of the division is rounded toward zero using the MATLAB fix function. This variable gives the amount of samples to read from the buffer, depending on the number of cycles that must be obtained at the given frequency to perform demodulation.

8.3.2 Demodulation

The demodulation function used to obtain the value of the impedance is given in listing 8.1.

```

1  function [ Impedance ] = demodulation(R_f,fc,DEC,resp,exc)
2
3  clk_f = 125000000;           % System Clock Frequency
4  fs = clk_f/DEC;             % Sampling Frequency
5  samps_T = fix(fs/fc);       % Number of Samples per Cycle
6
7  %% Demodulation Factors
8  n = 1:length(resp);
9  cos_mult = cos(2*pi*n/samps_T)';
10 sin_mult = -sin(2*pi*n/samps_T)';
11
12 %% Demodulation
13 rV_I = exc .* cos_mult';      % In-phase component of excitation
14 rV_Q = exc .* sin_mult';      % Quadrature component of excitation
15
16 rI_I = -(resp / R_f) .* cos_mult';    % In-phase component of response
17 rI_Q = -(resp / R_f) .* sin_mult';    % Quadrature component of response
18
19 %% Mean Value Calculation
20 mean_rV_I = mean(rV_I);
21 mean_rV_Q = mean(rV_Q);
22 mean_rI_I = mean(rI_I);
23 mean_rI_Q = mean(rI_Q);
24
25 %% Impedance of the sample
26 Impedance = (mean_rV_I+i*mean_rV_Q) / (mean_rI_I+i*mean_rI_Q)
27
28 end

```

Listing 8.1: Demodulation Function

The function takes the acquired excitation and response vectors corresponding to a single frequency, and multiplies each one by a reference signal in phase and in-quadrature to extract the real and imaginary parts of every individual data point in the vectors. The mean value of each vector is then computed and subsequently used to calculate the impedance value at the measured frequency.

8.3.3 Script Structure

An overview of the structure of the code that calls the discussed functions is outlined in figure 8.3.1. The essence of the script is the inner loop, which performs a single measurement sweep. Three sweeps are conducted every 2 hours, and this is repeated 24 times in order to monitor for a period of 24 hours.

Additionally, between the acquisition and demodulation steps, the acquired data was detrended in order to remove DC offsets caused by the offset voltage of the AD8642, and vectors acquired at a frequency below 1kHz were filtered in order to smooth out some of the low frequency noise.

```
1 % Remove Offsets
2 resp_dt = detrend(resp);
3 exc_dt = detrend(exc);
4
5 % Filter Low Frequency Noise
6 if freqs(c) < 10^3
7     resp_filt = sgolayfilt(resp_dt,7,107);
8     exc_filt = sgolayfilt(exc_dt,7,107);
9 else
10    resp_filt = resp_dt;
11    exc_filt = exc_dt;
12 end
```

Listing 8.2: Filtering

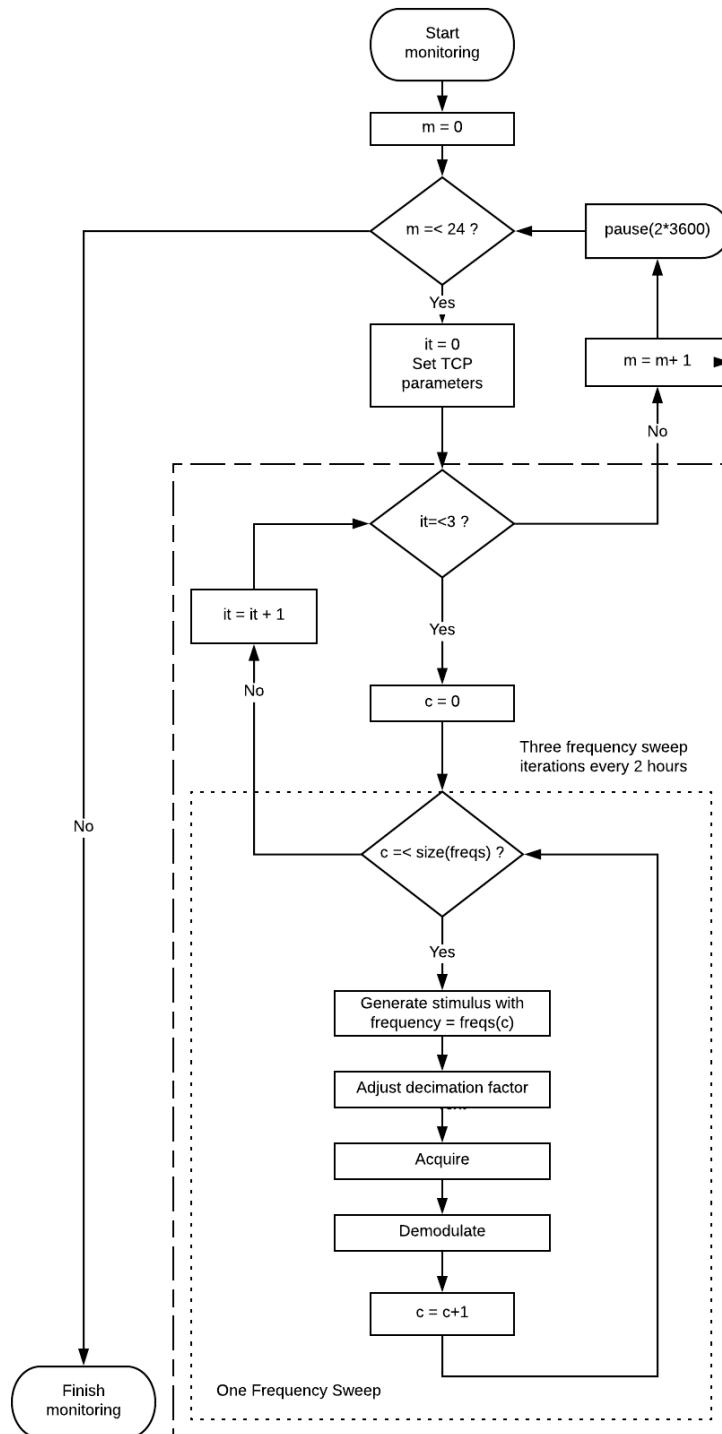


Figure 8.3.1: Flow diagram of the code structure used for cell culture monitoring.

Part III

Results and Future Work

Chapter 9

Results

9.1 Measurements on Passive Components

9.1.1 Test RC Network

Measurements were conducted on a simple RC network at the University of Oslo after the PCB front-end was manufactured in order to verify the both the assembly and the software. The test circuit has the same configuration as the parallel Debye model for a conductor.

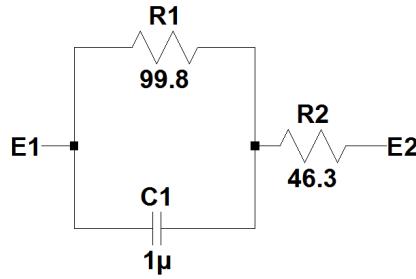


Figure 9.1.1: Test Circuit

At low frequencies, the absolute value of the impedance is given by the sum of R_1 and R_2 , which is expected to be about 146.1Ω . As the capacitive reactance drops with increasing frequency, the impedance of the parallel combination of C_1 and R_1 becomes small, and the value of the impedance is determined by the value of R_2 .

The measurements were conducted with an excitation voltage of 500 mV and in the frequency range between 100 Hz and 100 kHz. Results were plotted in a semilogarithmic plot (figure 9.1.2) and in the complex impedance wessel plot (figure 9.1.3).

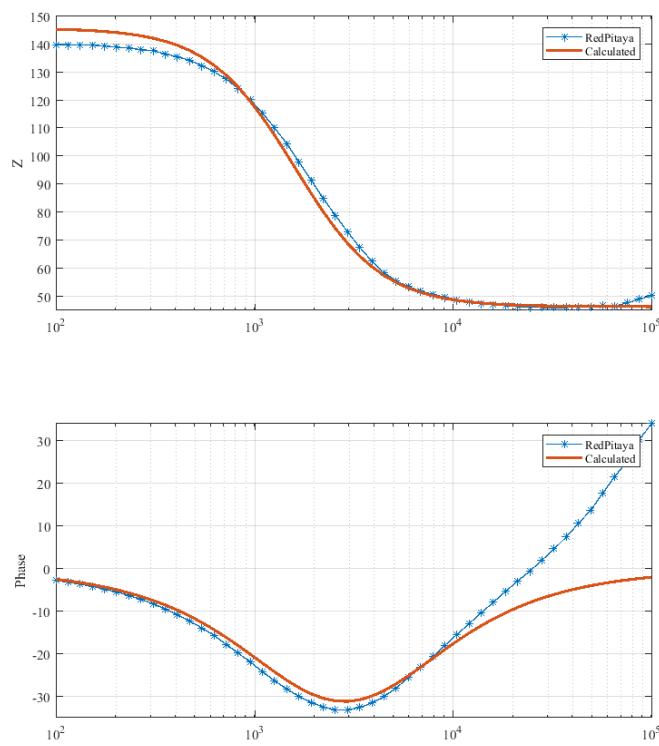


Figure 9.1.2: RC Test Circuit Semilogarithmic Plot

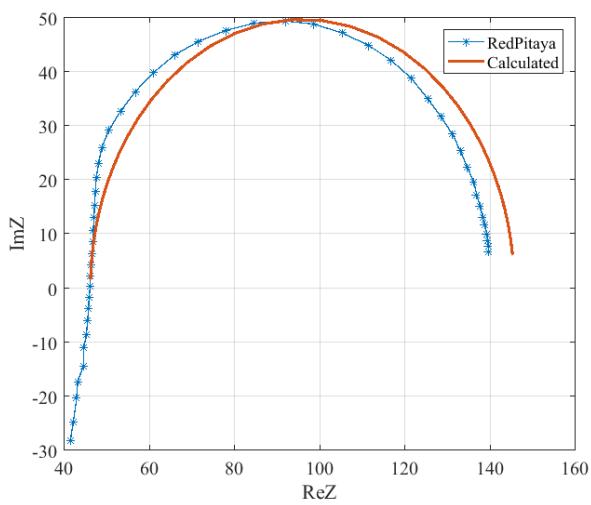


Figure 9.1.3: RC Test Circuit Wessel Plot

9.1.2 Gamry Dummy Cell

At the Technical University of Denmark, measurements were conducted on a PCB intended for potentiostat calibration in order to compare the results obtained with the Red Pitaya with results obtained with a dedicated potentiostat. The circuit is shown in figure 9.1.6. Considering that the Red Pitaya System measures across the nodes labeled WORKING and COUNTER, the circuit becomes very similar to the RC circuit in figure 9.1.1, R1 is $3.01\text{ k}\Omega$ and R2 is $200\text{ }\Omega + 47\text{ }\Omega = 247\text{ }\Omega$.

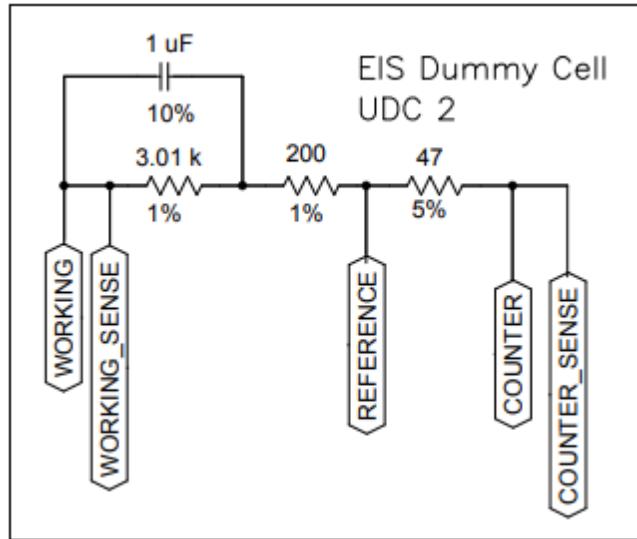


Figure 9.1.4: EIS Dummy Cell Schematic Diagram. From [36].

At this point it became apparent that it was desirable to operate with a broader frequency range and lower excitation voltage than what had been designed for. Lowering the excitation voltage to 10 mV and increasing the frequency range to go from 10 Hz to 100 kHz presented challenges with measurement consistency and increased inductive artifacts. Measurements between 10 Hz and 100 Hz picked up a lot of noise, which made it necessary to implement extra noise filtering in the MATLAB script to obtain meaningful results.

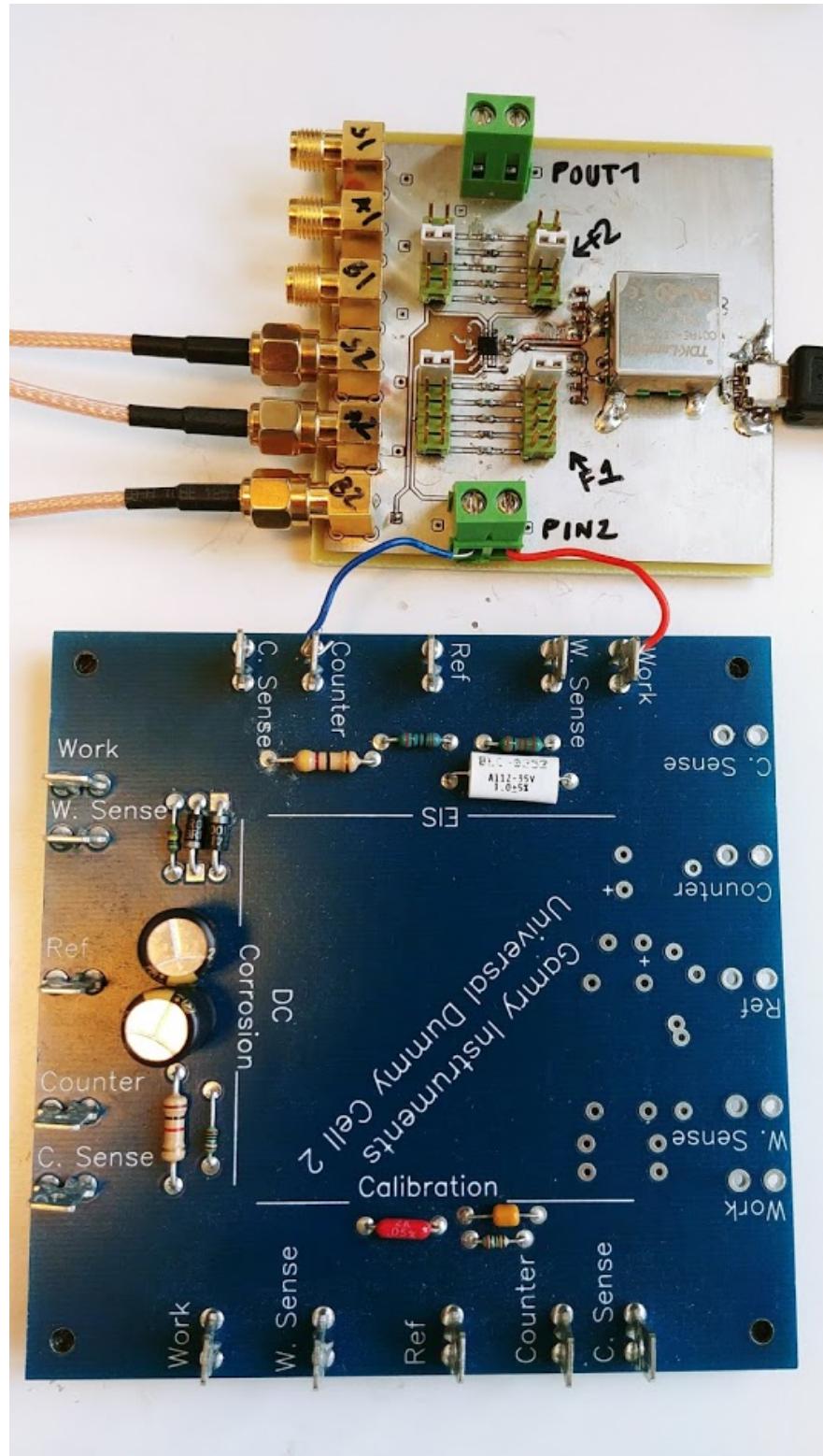


Figure 9.1.5: Gamry Dummy Cell Measurement Set-up

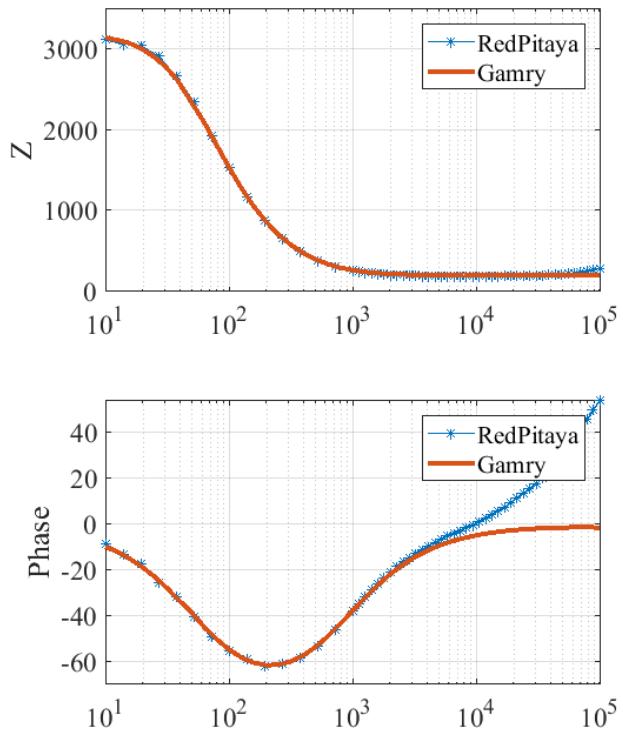


Figure 9.1.6: Gamry Dummy Cell Semilogarithmic Plot

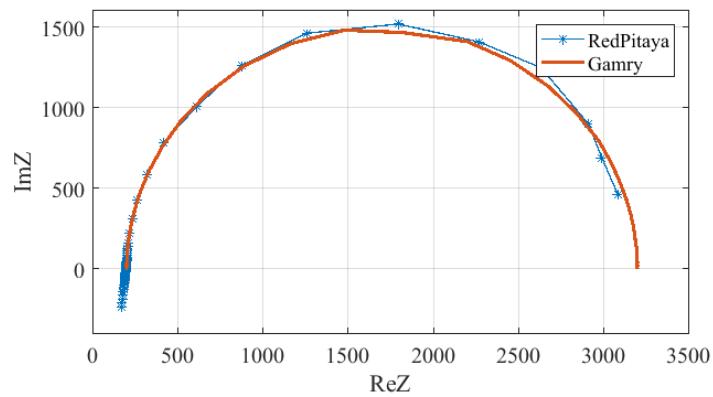


Figure 9.1.7: Gamry Dummy Cell Wessel Plot

9.2 Cell Measurements

Measurements were conducted on rat pheochromocytoma (PC-12) cells. These cells are tumorous cells from the adrenal gland of the kidney, but they express nerve growth factor and produce both dopamine and norepinephrine [37], which makes it possible to use them as a neuronal model. They are less cumbersome to culture than human ventral mesencephalic neural stem cells (hVM1), another cell line that was available in Copenhagen, so it was a natural first step to verify measurements on a PC-12 culture before attempting to measure hVM1 cells.

9.2.1 Cell culture preparation

Both the electrodes and the electrode chamber must undergo a series of procedures in order to prepare the cell culture before impedance measurements can be taken. Christin Schülke, an early stage researcher affiliated with the TRAINING4CRM project, was in charge of performing these procedures.

In this subsection an overview of the preparation procedures will be presented.

1. Oxygen/Air plasma treatment of electrode chips



Figure 9.2.1: Electrode Chip

The first step was to plasma treat the electrode chips (shown in figure 9.2.1) for about one and a half minutes. The main goal of plasma treatment is to remove contaminants from the surface of the electrodes, however in the particular case of carbon electrodes it also serves the purpose of aiding cellular attachment. Carbon is an hydrophobic material, which negatively impacts cellular adhesion since cell environments are aqueous. Plasma treatment introduces polar groups on the surface of the electrodes, which increases the hydrophilicity of the electrodes.

2. Assembly of electrode holder

The electrode holder was cleansed with ethanol before the electrode chips where placed in their sockets. Afterwards, the silicon rings were wiped with ethanol and rinsed with H_2O before being placed in their sockets.

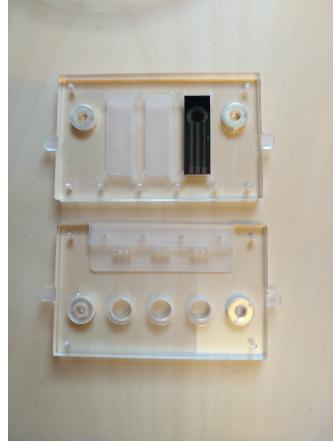


Figure 9.2.2: Electrode holder with one electrode chip in socket.

The assembled electrode holder is shown in figure 9.2.3.

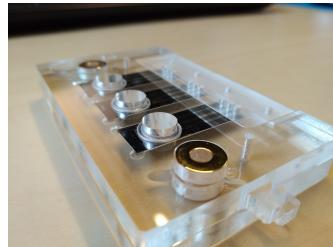


Figure 9.2.3: Assembled electrode holder.

After assembly, the tightness of the setup was verified by pipeting H_2O into the electrode wells and checking for leaks. Subsequently, the casing was sterilized for 15 minutes with 500mM sodium hydroxide ($NaOH$). Following sterilization the set-up was rinsed from $NaOH$ three times with phosphate buffered saline (PBS).

3. Coating

The next step after sterilization is to coat the electrode surface with extracellular matrix protein. In this case, the commercial Geltrex® was used. Extracellular matrix is necessary for cells to adhere and communicate, and Geltrex® has proven to be an effective coating for use with PC-12 cells, which otherwise are poorly adherent leading to low impedance measurement response [21]. Electrode coating was performed for a duration of two hours at 37°C.

4. Subculturing of cells

In order to seed cells in the wells of the electrode set-up, cells must be subcultured from a routine culture that is constantly grown in a cell culture flask. This cell culture flask is kept in an incubator at 37°C and 5% CO_2 concentration.

Subculturing, also known as passaging or splitting, is a procedure that must be repeated at a regular interval to avoid confluence of the cells. Confluence is reached when the cells have proliferated enough to reach the maximum amount of cells that can be held in the flask, and leads to a lack of cell culture medium that results in cell death if the cells are not passaged in a timely manner. Cell culture medium provides the essential nutrients and hormones that the cells need in order to grow, as well as an appropriate chemical environment.

The subculturing procedure is outlined below:

- Medium is removed from the cell culture flask.
- The cells are washed with PBS.
- The cells are detached from the cell culture flask. This is done by adding a mixture of Trypsin and Ethylene Diamine-Tetra-acetic Acid (EDTA). Trypsin is an enzyme that degrades the proteins that cause cell adhesion. EDTA is a chelating agent, and enhances the effect of the trypsin enzyme.
- The cell suspension is transferred into a new tube.
- The tube is centrifuged, causing the cells to clump in the bottom of the tube. The solution above the cells (supernatant) is then discarded.
- The cells are resuspended in fresh medium.

After passaging the cells were counted, and about 300 000 cells were seeded per well in 550 μL medium.

9.2.2 Measurements on Plasma Treated Electrodes

After obtaining satisfactory results with the Gamry Dummy Cell, measurements on plasma treated electrodes were attempted. It is standard procedure to measure the impedance of the electrodes before performing measurements on cells to perform normalization: the amount of cells that are seeded on electrodes is highly variable, therefore the impedance values can be higher or lower at the start of each monitoring cycle and need to be normalized with respect to the electrode surface. The aspect that is interesting when monitoring a culture is not necessarily the impedance value itself, but whether the real part of the impedance increases (indicating a larger amount of cells adhering to the electrode surface) or decreases (indicating that already adhered cells are dying) over time.

Obtaining spectra from the electrodes proved to be challenging due to the low excitation voltage and the size small size of the electrodes.

An attempt was made at increasing the sensitivity of the transimpedance amplifier by switching the $1k\ \Omega$ feedback resistor with a $10k\ \Omega$ feedback resistor, however this decreased the bandwidth of the amplifier dramatically, and in particular the apparent inductive effects exhibited by the phase response became very prominent.

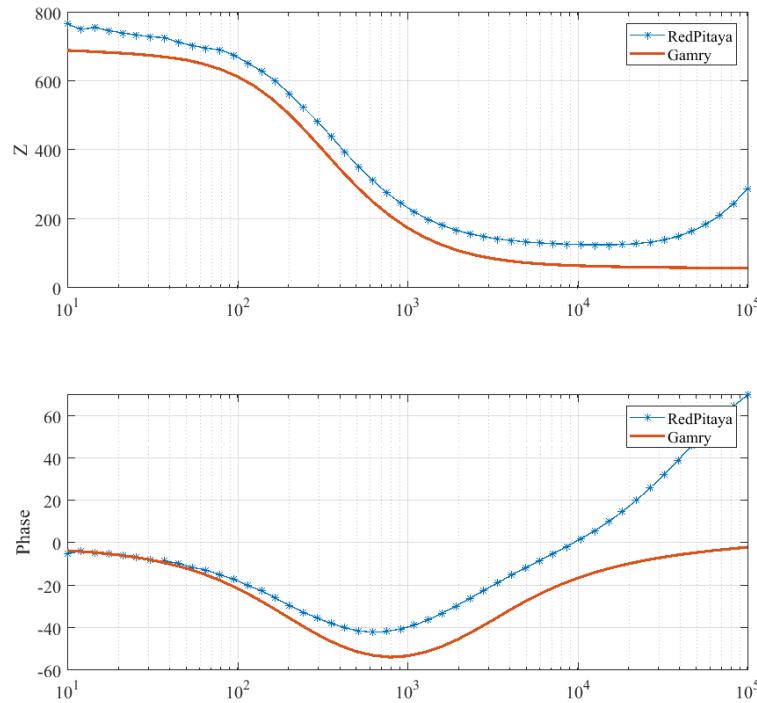


Figure 9.2.4: Plasma Treated Electrode Semilogarithmic Plot

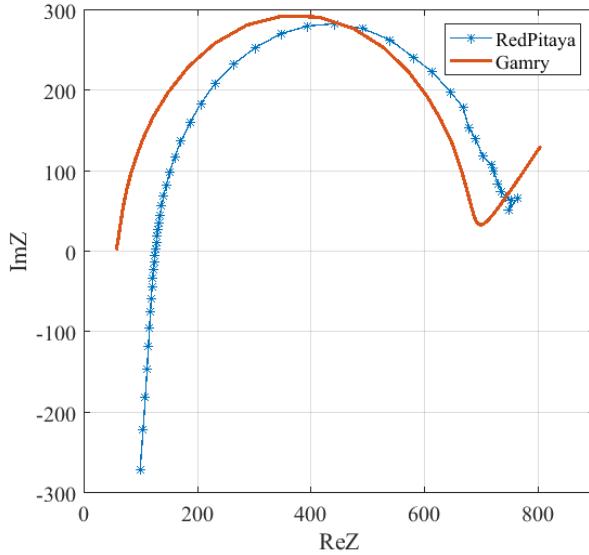


Figure 9.2.5: Coated Electrode Wessel Plot

9.2.3 Measurements on Cells

In order to be able to monitor the development of a cell culture, measurements must be conducted regularly during the monitored time interval. In order for cells to be able to survive and proliferate, they are placed in an incubator that has optimal temperature and $C0_2$ concentration, $37^\circ C$ and 5% CO_2 , respectively. The incubator itself must be in a clean room to avoid contaminants in the cell culture, and the electrodes should not be manipulated after being placed in the incubator.

The nature of the set-up poses challenges regarding sterilization of the instrument set-up, as well as revision of connections to the electrodes in the incubator, and in general once a measurement is started it is difficult to make corrections on the fly. Given that the goal was to monitor a cell culture over at least three days, and the time in Copenhagen was limited, there was only enough time to seed one well and attempt to monitor the development of the cell culture in said well. The measurements were started with the same parameters that were used to obtain the electrode impedance spectra, meaning a low excitation and a $10\ k\Omega$ feedback resistor. Once preliminary results were fetched (shown in figure 9.2.6), it became apparent that the measurement had failed. For this reason, it was decided to increase the excitation voltage to 500 mV five hours after the electrodes had been placed in the incubator, to see if that would produce a meaningful spectrum, but it was not possible to change the feedback resistance. The result after this modification is shown in figure 9.2.7.

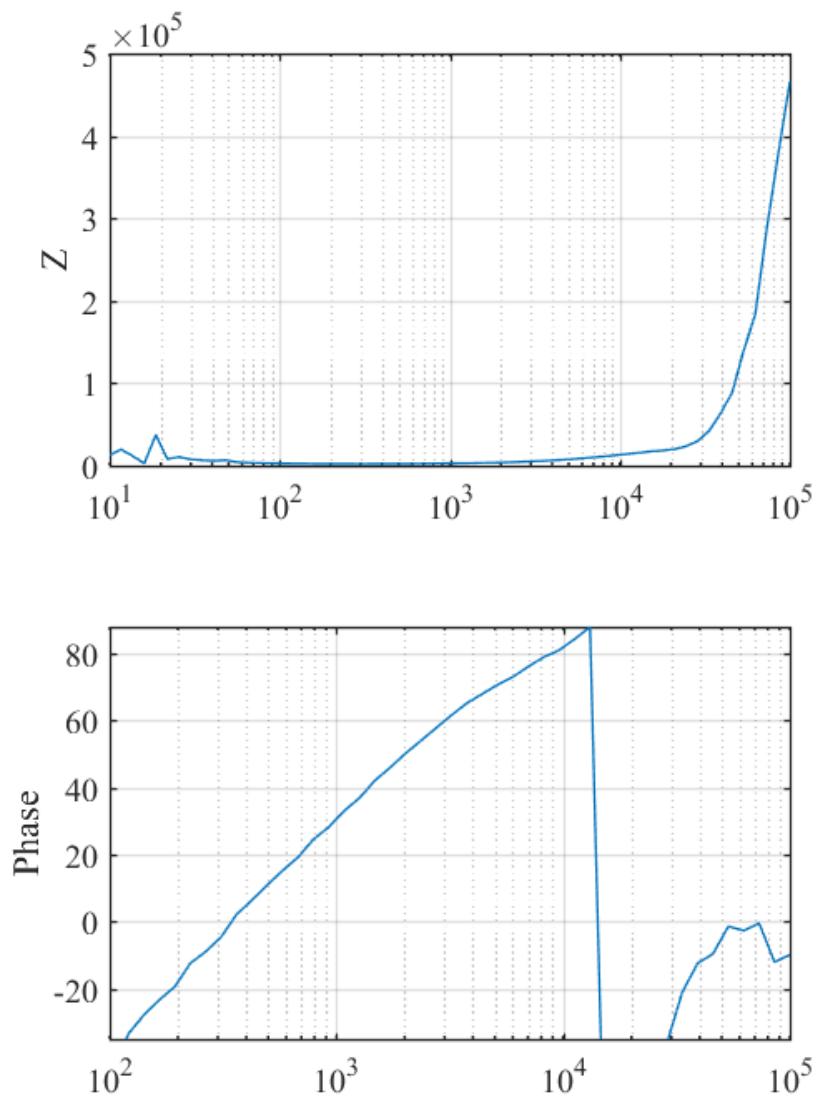


Figure 9.2.6: Initial measurement on cells with low excitation voltage

With the increase of the excitation signal, it was possible to obtain a spectrum ranging from $19.6\text{ k}\Omega$ and -85 degrees phase shift at low frequencies to $152\text{ }\Omega$ and 10 degrees phase shift and at high frequencies.

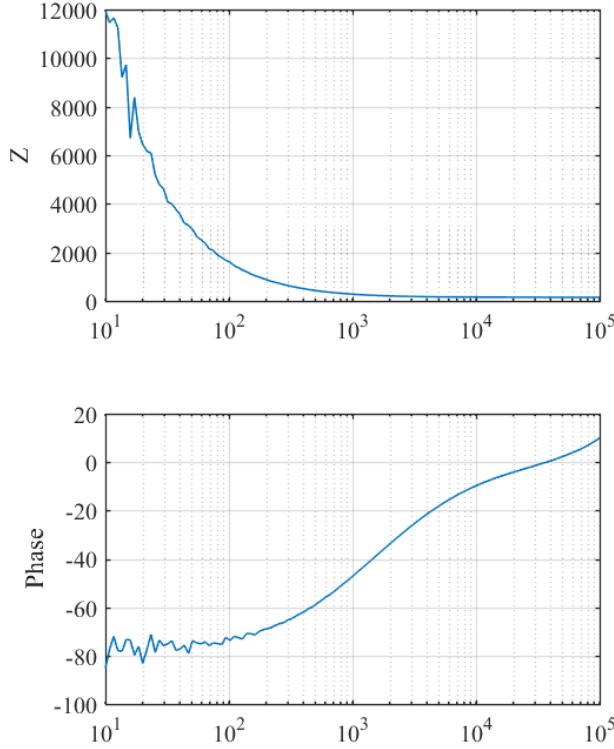


Figure 9.2.7: Measurement with higher excitation voltage

At frequencies below 100 Hz, the system still picked up a lot of noise, therefore the frequency range was reduced to the original 100 Hz to 100 kHz. By the second day of monitoring, Matlab had lost connection to the UiO license provider, and had stopped taking measurements. It was decided to take a new measurement at that point, about 19 hours after the measurement had been acquired, expecting to prove the death of the cells due to the high input voltage. Cell apoptosis would be evidenced by a decrease in the real component of impedance, because when fewer cells adhere to and cover the electrode surface there is more space for free ionic conduction to take place.

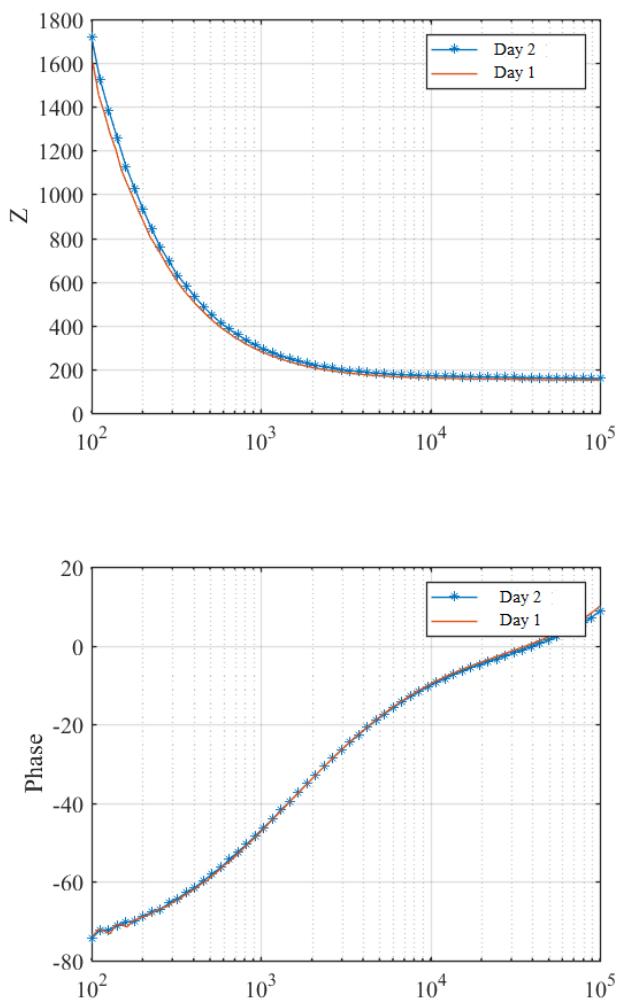


Figure 9.2.8: Two measurements taken about 19 hours apart.

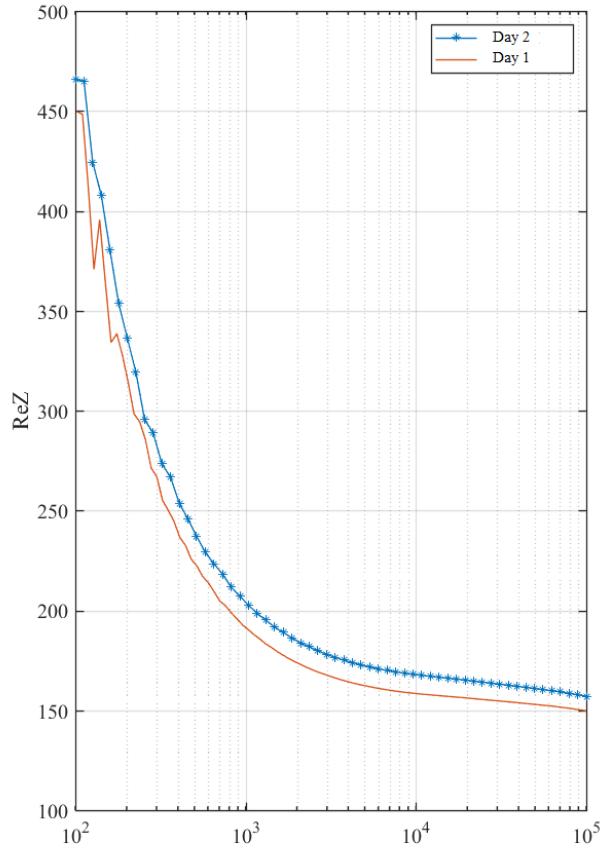


Figure 9.2.9: Real component of impedance. Measurements taken about 19 hours apart.

Surprisingly, the measurement taken the 21st of February showed an increased real component of impedance value, by about 10Ω . This increase may or may not be negligible with regards to cell adhesion, and it raises the question of whether cell monitoring should have been continued for a longer period of time. An alternative explanation is that the measurements taken on the previous day may have harmed the cells that had adhered to the electrodes prior to the increase of the excitation signal amplitude, but that other cells had had the opportunity to adhere and grow on the electrodes in the 19 hour interval where no measurements were conducted.

It would have been useful to be able to measure simultaneously with professional equipment in order to corroborate the results, but it was not physically feasible to move such equipment into the clean room.

Chapter 10

Conclusions and Future Work

10.1 Front-End

The front-end PCB that was employed to take measurements at DTU was developed in a very short time-frame, and therefore there are many weaknesses to address in a potential new revision of the board.

The design of the front-end didn't take into consideration that in order to avoid sustained electroporation of the cells (leading to the death of the cells), only very low current must pass through them. For this reason the excitation voltage had to be lowered considerably to perform measurements without risking cell damage. The voltage was lowered digitally, from the DAC, to 10 mV. In order to take full advantage of the on board DAC, a revision of the front end should be made where the full scale output of the board is lowered through analog components to a suitable value. There is no agreement in the research group about what the upper value of the current should be.

The measurements also displayed very significant inductive effects. Some of these effects may be accounted for through the self-inductance of the system, which is estimated to about $1\mu H/m$, however, selection of an operational amplifier with higher gain bandwidth product will improve the performance at higher frequencies significantly.

The input offset voltage of the transimpedance amplifier added a significant DC component to the measured response. This DC component was removed digitally, but a new revision of the board should aim for as low as possible input offset voltage due to the excitation signal amplitude required.

The resolution of the on board ADC's was also not fully taken advantage of. The jumper that defines the input voltage range was set to $\pm 20 V$, but with an amplifier supply of $\pm 12 V$ and the sensitivity that produced the better results ($1V/mA$), it is clear that much better response resolution is attainable.

Finally, the front-end included its own power supply that was required an extra micro-usb connection. This meant one more wire that needed to be sterilized and brought into the clean-room. The problem can be avoided by powering the amplifier through the power pins of the Red Pitaya extension connectors (shown in figure 5.2.3) and dimensioning the PCB such that it can be stacked directly on top of the Red Pitaya. For a dual supply amplifier the supply range would therefore be $\pm 3 V$.

10.1.1 Proposed Improvements

An alternative design of the front-end is presented in this subsection, that addresses the above mentioned weaknesses of the prototype used in Denmark.

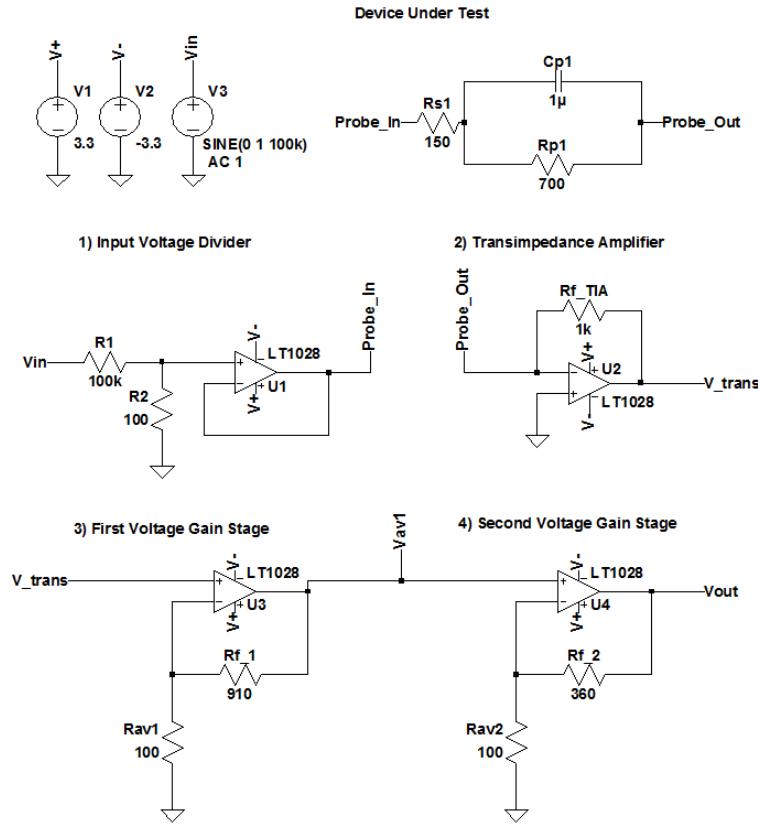


Figure 10.1.1: Front-End Revision Proposal.

This design features four LT1028 operational amplifiers. This is a low noise amplifier with a higher gain-bandwidth product, lower input offset voltage, but as a trade-off, a higher input bias current.

	AD8642		LTC1028			
	Typ	Max	Typ	Max		
Input Offset Voltage	50	1.5	mV	20	80	uV
Input Bias Current	-	260	pA	30	180	nA
Input Offset Current	-	65	pA	18	100	nA
Gain Bandwidth Product	3.5	-	MHz	75	-	MHz

Table 10.1.1: Comparison of key operational amplifier design specifications

Input Voltage Divider

To address the low excitation amplitude requirement, a voltage divider followed by a voltage follower should be employed. The voltage follower is necessary to prevent loading the output.

The voltage divider reduces the amplitude of the excitation signal by a factor of 1/1001, according to eq. 10.1, thus lowering the input amplitude from 1 V_p to about 1 mV_p.

$$\frac{Probe_In}{Vin} = \left(1 + \frac{100k\Omega}{100\Omega}\right)^{-1} = \frac{1}{1001} \quad (10.1)$$

Since a specification for the true maximum current value is uncertain, a necessary consequence of this project is that the current tolerance of different cell lines must be researched in order to avoid excessive attenuation of the excitation signal in future revisions.

Transimpedance Amplifier

The transimpedance amplifier has a sensitivity of 1V/mA. The gain of the transimpedance amplifier is kept low to avoid narrowing the bandwidth unnecessarily. For an AC input current of 20μA, the output will be about 200mV.

Voltage Gain

Two stages non-inverting amplifiers are used to raise the output of the transimpedance amplifier to a value suitable for sampling at the input of the Red Pitaya. Using two amplifier stages will also help avoid narrowing the bandwidth.

The total gain (eq. 10.4) is low enough to ensure that an impedance as low as 50Ω at 100kHz still will produce an output with an amplitude lower than 1V_p, making it possible to use the low-voltage scale setting at the Red Pitaya input.

$$Av_1 = \frac{V_{av1}}{V_{trans}} = 1 + \frac{Rf_1}{Rav_1} = 1 + \frac{910\Omega}{100\Omega} = 10.1 \quad (10.2)$$

$$Av_2 = \frac{V_{out}}{V_{av1}} = 1 + \frac{Rf_2}{Rav_2} = 1 + \frac{360\Omega}{100\Omega} = 4.6 \quad (10.3)$$

$$Av_{tot} = Av_1 \cdot Av_2 = 10.1 \cdot 4.6 = 46.46 \quad (10.4)$$

One might want to consider raising the gain of the amplifier stages if the impedance at 100 kHz proves to increase much above 750 Ω, a value that would produce an output swing of about 300mV with the given component values. Rav2 can be switched with a 560 Ω resistor to raise the voltage gain of the second stage to 6.6, and the total gain to 66.66. This will not degrade the performance in the frequency range of interest, but the tradeoff is that the impedance at 100 kHz should be at least 100 Ω to avoid clipping at the input. This is higher than recorded when measuring the prepared electrodes, and therefore a lower gain was chosen as a starting point for the new design. It is desirable to have the highest amplitude possible within 1V_p in order to take as much advantage as possible of the Red Pitaya ADC, so the correct gain configuration should be investigated further.

Impedance Response Simulation

The expected impedance response has been simulated with values similar to those measured for the plasma treated electrode in figure 9.2.4, by performing an AC analysis of the proposed design with the device under test shown in figure 10.1.1. In order to do this we must first derive an expression for the impedance as a function of V_{in} and V_{out} .

$$Z = \frac{V_{DUT}}{I_{DUT}} \quad (10.5)$$

V_{DUT} is simply the excitation signal after the voltage division:

$$V_{DUT} = \frac{1}{1001} \cdot V_{in} \quad (10.6)$$

I_{DUT} as a function of output voltage V_{out} is given by:

$$\begin{aligned} I_{DUT} &= \frac{V_{trans}}{-Rf_{TIA}} \\ &= \frac{V_{out}}{Av_{tot}} \cdot \frac{1}{-Rf_{TIA}} \\ &= \frac{V_{out}}{46.46} \cdot \frac{1}{-1k\Omega} \\ &= -2.152 \cdot 10^{-5} \cdot V_{out} \end{aligned} \quad (10.7)$$

Thus, the impedance as a function of V_{in} and V_{out} is given by:

$$\begin{aligned} Z &= \frac{V_{DUT}}{I_{DUT}} \\ &= \frac{\frac{1}{1001} \cdot V_{in}}{-2.152 \cdot 10^{-5} \cdot V_{out}} \\ &= -46.42 \cdot \frac{V_{in}}{V_{out}} \end{aligned} \quad (10.8)$$

The resultant frequency response is shown in figure 10.1.2, together with the simulated response using the AD8642 prototype configuration. In particular, the phase response using the revised design shows a significant improvement of the phase response in the frequency range between 10 kHz and 100 kHz, reducing the apparent inductive effects that were seen in the experimental results.

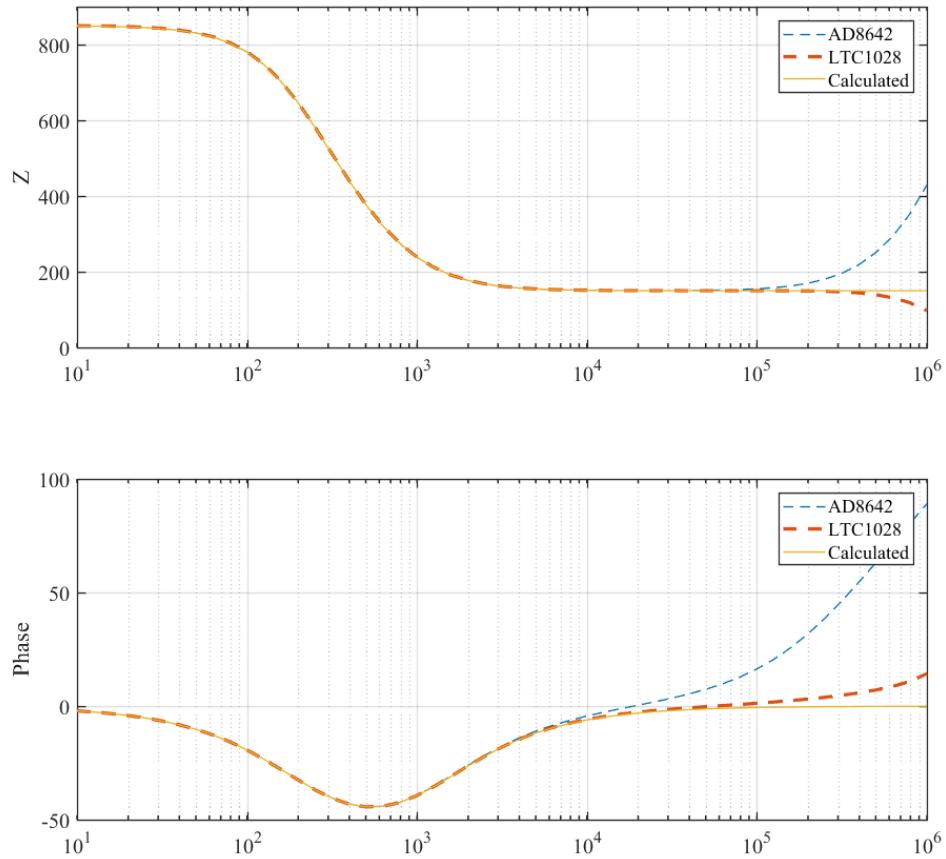


Figure 10.1.2: Simulated responses using the AD8642-based prototype design, and the LTC1028-based revised design. Calculated values added for reference.

10.2 Scripts

The software used for measurement in this project can be further improved by making modification of measurement parameters more transparent. An example would be to create a function that takes the measurement frequency as an input and returns the best suited decimation factor. By doing this, it would be much easier to extend the bandwidth of the measurement, or else increase or decrease the number of frequency points at each measurement sweep.

A very serious problem that became apparent in Denmark was that Matlab would lose connection with the license server, interrupting the monitoring cycle. It would be better to limit the Matlab script to perform a single acquisition cycle, and implement the monitoring functionality through a python/bash script. The purpose of the aforementioned script would be to launch Matlab and run the single acquisition cycle periodically.

Finally, a revision of the front-end would make it necessary to modify the demodulation function according to the attenuation and/or gain settings of the electronics surrounding the transimpedance amplifier.

10.3 Embedded System Development

The applicability of the Red Pitaya in design of high-bandwidth bioimpedance measurement systems is high due to the specifications of the fast on-board DACs and ADCs and the flexibility to develop tailored signal processing solutions in the programmable logic of the Zynq-SOC. The quality of the acquisition using API functions does not seem to correspond to that specification, and debugging is difficult because the firmware of the FPGA is not documented, therefore it is not trivial to improve the API functions to better take advantage of the on-board resources.

Highly skilled users such as Pavel Denim are able to realize complex systems that take full advantage of the specifications by modifying the firmware and designing IP to suit their needs. However, in order to do so it is important to allocate enough time to understand the design flow of Zynq-devices prior to attempting to create a tailored design on a board that is not supported or documented by Xilinx. A better approach would therefore be to first become proficient in IP-flow design with the help of a well documented and vendor-supported development board such as the Zedboard (produced by Digilent INC), and then porting the design to the Red Pitaya board.

Once proficiency with IP-flow design in Vivado is attained, and once the designer has sufficient experience with packaging and creating IP to be incorporated in a system based on AXI4-bus data movement, the approach suggested in chapter 7 can be used to build a project adapted to the Red Pitaya. It is recommended to design and verify the IP with the help of a development board before incorporating it into the Red Pitaya. It is also recommended that a solid working knowledge of Embedded Linux is in place before attempting to create firmware for the Red Pitaya with the intent of producing a small bioimpedance instrument.

Appendices

Appendix A

Embedded Development Tools

A.1 Installing Vivado Design Suite

For embedded system development, it is necessary to have an installation of Vivado on a development machine capable of ssh connection with the RedPitaya board.

To install Vivado on a development machine running Ubuntu 16.04 LTS, download the Linux self-extracting web installer for the latest Vivado HLx WebPACK edition. At the time of writing the latest version was 2017.4. The following bash commands must subsequently be run from the download directory in order to start the installer:

```
$ chmod +x Xilinx_Vivado_SDK_Web_2017.4_1216_1_Lin64.bin  
$ sudo ./Xilinx_Vivado_SDK_Web_2017.4_1216_1_Lin64.bin
```

Make sure to include Zynq-7000 SoC devices during the installation.

Running the following bash command on the development machine will install additional libraries that are necessary for the Vivado installation to work properly:

```
apt-get install libxft2 libxft2:i386 lib32ncurses5
```

The permissions of the Xilinx installation folders need to be changed as well:

```
$ sudo chmod 777 -R /opt/Xilinx/  
$ sudo chmod 777 -R /.Xilinx/
```

Finally, the following lines must be added to the .bashrc file in the development machine:

```
# Xilinx  
. /opt/Xilinx/Vivado/2017.4/settings64.sh  
export XILINXD_LICENSE_FILE=/home/username/.Xilinx/Xilinx.lic  
# End Xilinx
```

Appendix B

MATLAB Scripts

B.1 Generate

```
1 function void = generate(IP,port,chan,amp,freq,type,state,DC)
2
3 tcpipObj=tcpip(IP, port);
4 fopen(tcpipObj);
5 tcpipObj.Terminator = 'CR/LF';
6 fprintf(tcpipObj,['SOUR' num2str(chan) ':FUNC ' type]);
7 fprintf(tcpipObj,['SOUR' num2str(chan) ':FREQ:FIX ' num2str(freq) ]);
8 fprintf(tcpipObj,['SOUR' num2str(chan) ':VOLT ' num2str(amp)]);
9 fprintf(tcpipObj,['SOUR' num2str(chan) ':VOLT:OFFS ' num2str(DC)]);
10 fprintf(tcpipObj,['OUTPUT' num2str(chan) ':STATE ' state]);
11 fclose(tcpipObj);
12
13 end
```

B.2 Acquire

```

1 function [resp, exc]=acquire(IP,port,DEC,fc,c)
2 % Constants
3 samps = 16834;
4 clk_f = 125000000;
5 % Samples per period calculation
6 fs = clk_f/DEC;
7 samps_T = fix(fs/fc);
8 % TCPI Object Config
9 tcpipObj=tcpip(IP, port, 'Timeout', 20);
10 tcpipObj.InputBufferSize = samps*32;
11 tcpipObj.OutputBufferSize = samps*32;
12 fopen(tcpipObj);
13 tcpipObj.Terminator = 'CR/LF';
14 % Remove all data in buffers
15 flushinput(tcpipObj);
16 flushoutput(tcpipObj);
17 fprintf(tcpipObj,'ACQ:RST');
18 % Set decimation value
19 fprintf(tcpipObj,['ACQ:DEC ' num2str(DEC)]);
20 % Enable averaging
21 fprintf(tcpipObj,'ACQ:AVG ON');
22 % Set trigger level
23 fprintf(tcpipObj,'ACQ:TRIG:LEV 0');
24 % Set gain (depends on amplifier gain whether HV or LV is needed for SOUR1)
25 fprintf(tcpipObj,'ACQ:SOUR1:GAIN HV');
26 fprintf(tcpipObj,'ACQ:SOUR2:GAIN LV');
27 % Set delay
28 fprintf(tcpipObj,'ACQ:TRIG:DLY 8192');
29 pause(1) % Wait for data writing
30 fprintf(tcpipObj,'ACQ:START');
31 %fprintf(tcpipObj,'ACQ:TRIG CH2_PE');
32 fprintf(tcpipObj,'ACQ:TRIG NOW'); % Triggers immediately
33
34 while 1
35     trig_rsp=query(tcpipObj,'ACQ:TRIG:STAT?');
36
37     if strcmp('TD',trig_rsp(1:2)) % Read only TD
38
39         break
40
41     end
42
43 %% Read data from buffers, 8 periods each
44 fprintf(tcpipObj,'ACQ:DATA:UNITS VOLTS');
45 fprintf(tcpipObj,'ACQ:DATA:FORMAT ASCII');
46 signal_str=query(tcpipObj,'ACQ:SOUR1:DATA? 0, ' num2str(8*samps_T));
47 signal_str_2=query(tcpipObj,'ACQ:SOUR2:DATA? 0, ' num2str(8*samps_T));
48 % Convert values to numbers
49 signal_num=str2num(signal_str(1,2:length(signal_str)-3));
50 signal_num_2=str2num(signal_str_2(1,2:length(signal_str_2)-3));
51 % Reset to default values
52 fprintf(tcpipObj,'ACQ:RST');
53 fclose(tcpipObj)
54 % Return acquired vectors
55 resp    = [signal_num];
56 exc     = [signal_num_2];
57 end

```

B.3 Demodulation

```

1 function [ Impedance ] = demodulation(R_f,fc,DEC,resp,exc)
2
3 clk_f = 125000000; % System Clock Frequency
4 fs = clk_f/DEC; % Sampling Frequency
5 samps_T = fix(fs/fc); % Number of Samples per Cycle
6
7 %% Demodulation Factors
8 n = 1:length(resp);
9 cos_mult = cos(2*pi*n/samps_T)';
10 sin_mult = -sin(2*pi*n/samps_T)';
11
12 %% Demodulation
13 rV_I = exc .* cos_mult'; % In-phase component
14 rV_Q = exc .* sin_mult'; % Quadrature component
15
16 rI_I = -(resp / R_f) .* cos_mult'; % In-phase component
17 rI_Q = -(resp / R_f) .* sin_mult'; % Quadrature component
18
19 %% Mean Calculation
20 mean_rV_I = mean(rV_I);
21 mean_rV_Q = mean(rV_Q);
22 mean_rI_I = mean(rI_I);
23 mean_rI_Q = mean(rI_Q);
24
25 %% Current through the sample
26 I = abs(mean_rI_I + i*mean_rI_Q)
27
28 %% Impedance of the sample
29 Impedance = (mean_rV_I+i*mean_rV_Q)/(mean_rI_I+i*mean_rI_Q)
30
31 end

```

B.4 Main Script

```

1 %% Perform 3 measurements every two hours, 24x
2
3 for m=1:24
4   for it =1:3
5     %% Set Parameters to Connect to SCPI server
6     IP = '10.42.0.118';
7     port = 5000;
8     chan = 1;
9     amp = 0.01;
10    type = 'SINE';
11    state = 'ON';
12    DC = 0;
13
14    R_f = 9.92*10^3;
15    freqs = logspace(1,5,60);
16
17
18    %% One Measurement Loop
19    for c = 1:size
20      % Red Pitaya Generate Utility
21      generate(IP,port,chan,amp,freqs(c),type,state, DC);
22
23      % Adjust decimation to frequency
24      if 0 < c < 20
25        DEC = 1024;
26      elseif 20 < c < 40
27        DEC = 8;
28      elseif 40 < c
29        DEC = 1;
30      end
31
32      fc=freqs(c);
33
34      % Red Pitaya Acquisition Utility
35      [exc, resp] = acquire(IP,port,DEC,fc);
36
37      % Remove Offsets
38      resp_dt = detrend(resp);
39      exc_dt = detrend(exc);
40
41      % Filter Low Frequency Noise
42      if freqs(c) < 10^3
43        resp_filt = sgolayfilt(resp_dt,7,107);
44        exc_filt = sgolayfilt(exc_dt,7,107);
45      else
46        resp_filt = resp_dt;
47        exc_filt = exc_dt;
48      end
49
50      disp('Frequency =')
51      disp(freqs(c))
52
53      impedance(c) = demodulation(R_f,fc,DEC,resp_filt,exc_filt);
54      phase(c) = atand(imag(impedance(c))/real(impedance(c)));
55
56    end

```

Listing B.1: Main Script Part 1

```

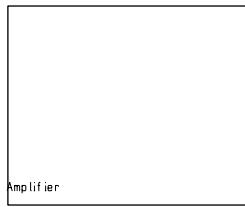
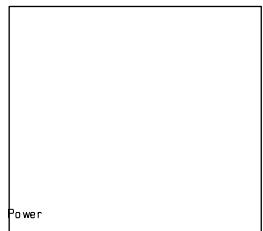
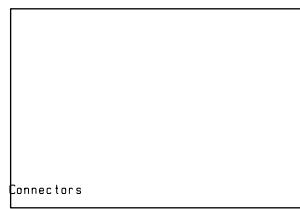
1    %% Plotting and Saving exc
2    timestamp = datestr(now,'mm.dd-HH:MM');
3    %path = '/home/magnamite/exc/Experiment/Monitor_1/_'
4    %path = '/home/magnamite/exc/Experiment/Normalization/chip_'
5    filename = [path 'cables_' timestamp '.dat']
6    newexc = horzcat(freqs',real(impedance)',imag(impedance)',phase')
7    csvwrite(filename, newexc)
8    figure
9    subplot(3,2,1)
10   semilogx(freqs, abs(impedance))
11   ylabel('Z')
12   grid on
13   subplot(3,2,2)
14   semilogx(freqs, phase)
15   ylabel('Phase')
16   grid on
17   subplot(3,2,3)
18   semilogx(freqs, real(impedance))
19   ylabel('ReZ')
20   grid on
21   subplot(3,2,4)
22   semilogx(freqs, imag(impedance))
23   ylabel('ImZ')
24   xlabel('f')
25   grid on
26   subplot(3,2,5)
27   plot(real(impedance), -imag(impedance))
28   ylabel('-ImZ')
29   xlabel('ReZ')
30   grid on
31   saveas(gcf,[filename '.png'])
32   clc
33   clear all
34   close all
35   okay=1;
36 end;
37 pause(2*3600)
38 end;

```

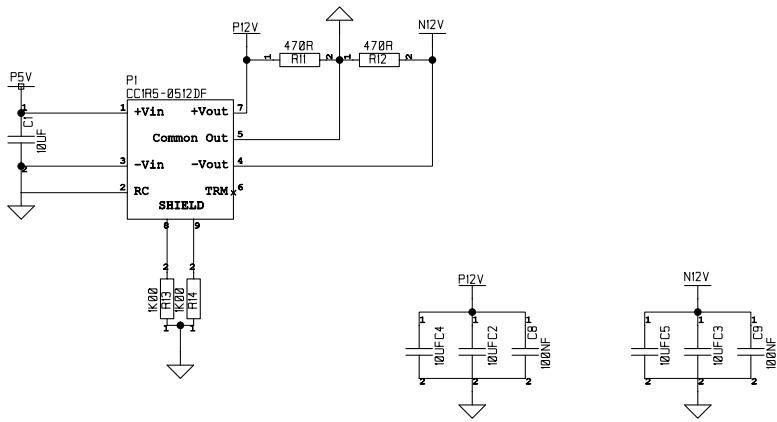
Listing B.2: Main Script Part 2

Appendix C

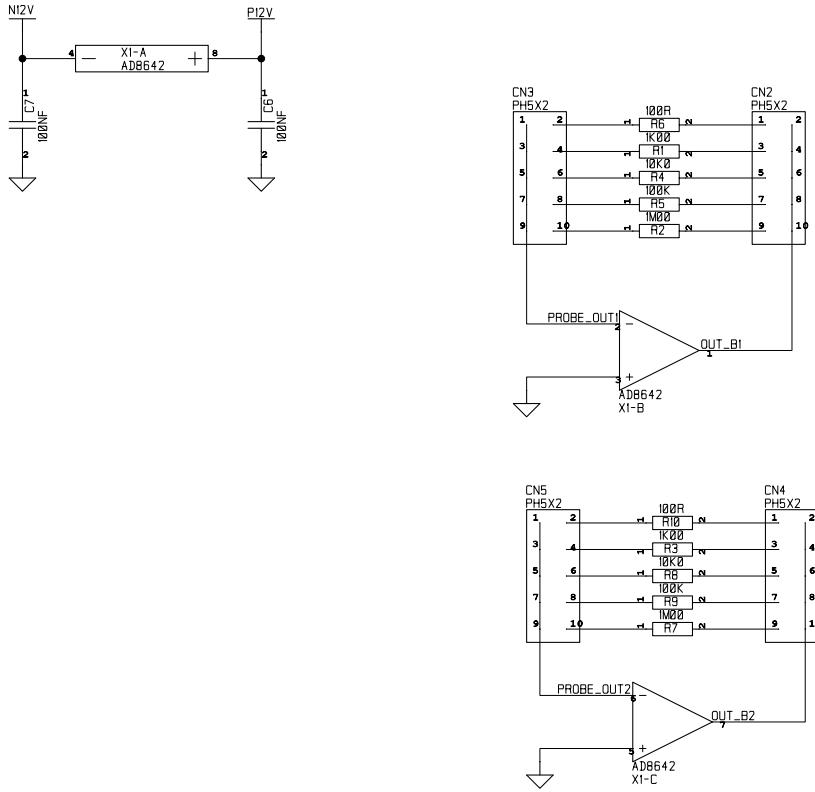
Front-End Schematics



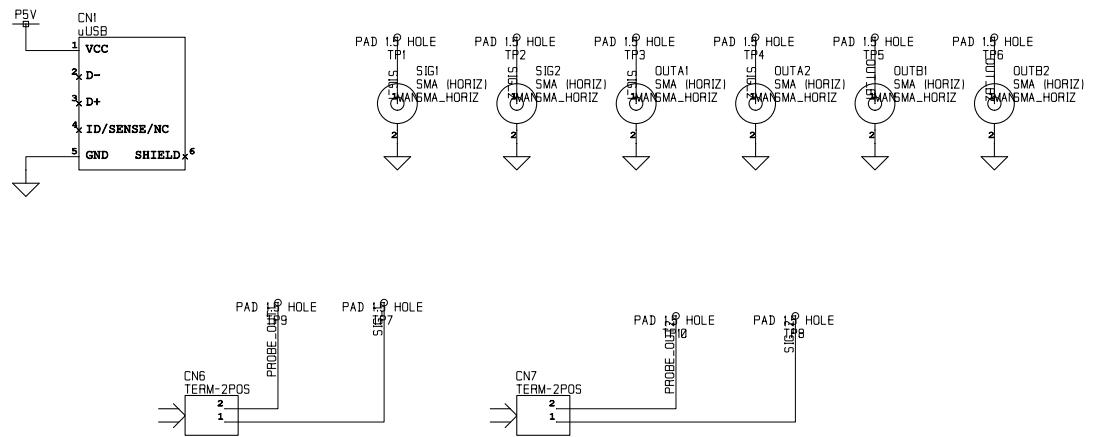
PROJECT:	RP_Frontend
SHEET NAME:	TOP
DATE:	24.01.2018
SHEET:	1 / 4
USER:	mknordga
ELAB. Dept. of Physics	UNIVERSITY OF OSLO



PROJECT:	RP_Frontend
SHEET NAME:	Power
DATE:	24.01.2018
SHET#:	2 / 4
USER:	mknordga
ELAB:	Dept. of Physics UNIVERSITY OF OSLO



PROJECT:	RP_Frontend
SHEET NAME:	Amplifier
DATE:	24.01.2018
SHFT:	3 / 4
USER:	mknordga
ELAB:	Dept. of Physics UNIVERSITY OF OSLO



PROJECT:	RP_Frontend
SHEET NAME:	Connectors
DATE:	24.01.2018
Sheet:	4 / 4
User:	mknordga
ELAB, Dept. of Physics	UNIVERSITY OF OSLO

Bibliography

- [1] The TRAINING4CRM Project. *About*. URL: <http://www.training4crm.eu/about-training4crm> (visited on 15/05/2018).
- [2] Ranjeet Singh Mahla. ‘Stem Cells Applications in Regenerative Medicine and Disease Therapeutics’. en. In: *International Journal of Cell Biology* 2016 (2016), pp. 1–24. ISSN: 1687-8876, 1687-8884. DOI: [10.1155/2016/6940283](https://doi.org/10.1155/2016/6940283). URL: <http://www.hindawi.com/journals/ijcb/2016/6940283/> (visited on 15/05/2018).
- [3] A. Heiskanen and J. Emnéus. ‘Monitoring of Cellular Dynamics with Electrochemical Detection Techniques’. In: *Applications of Electrochemistry and Nanotechnology in Biology and Medicine I*. Ed. by Noam Eliaz. New York, NY: Springer New York, 2011, pp. 1–104. ISBN: 978-1-4614-0347-0. DOI: [10.1007/978-1-4614-0347-0_1](https://doi.org/10.1007/978-1-4614-0347-0_1). URL: https://doi.org/10.1007/978-1-4614-0347-0_1 (visited on 25/03/2018).
- [4] Adel S. Sedra and Kenneth C. Smith. *Microelectronic Circuits*. Oxford University Press, 2011. ISBN: 9780199738519.
- [5] Sverre Grimnes and Ørjan G. Martinsen. *Bioimpedance and Bioelectricity Basics*. Elsevier Ltd., 2015. ISBN: 9780124114708.
- [6] C. Canali et al. ‘An impedance method for spatial sensing of 3D cell constructs – towards applications in tissue engineering’. In: *The Analyst* 140.17 (2015), pp. 6079–6088. ISSN: 0003-2654, 1364-5528. DOI: [10.1039/C5AN00987A](https://doi.org/10.1039/C5AN00987A). (Visited on 19/03/2018).
- [7] Jin Keun Seo et al. ‘Effective Admittivity of Biological Tissues as a Coefficient of Elliptic PDE’. In: *Computational and Mathematical Methods in Medicine* 2013 (2013), pp. 1–10. ISSN: 1748-670X, 1748-6718. DOI: [10.1155/2013/353849](https://doi.org/10.1155/2013/353849). (Visited on 19/03/2018).
- [8] Henry W. Ott and Henry W. Ott. *Electromagnetic compatibility engineering*. Hoboken, N.J: John Wiley & Sons, 2009. 843 pp. ISBN: 978-0-470-18930-6.
- [9] W. Kuang and S. O. Nelson. ‘Low-frequency dielectric properties of biological tissues: a review with some new insights’. In: *Transactions of the ASAE* 41.1 (1998), pp. 173–184. DOI: [10.13031/2013.17142](https://doi.org/10.13031/2013.17142). (Visited on 20/03/2018).
- [10] Ørjan G. Martinsen, Sverre Grimnes and Herman P. Schwan. *Interface Phenomena and Dielectric Properties of Biological Tissue*. In: *Encyclopedia of Surface and Colloid Science*. Vol. 20. New York, 2002, pp. 2643–2652.
- [11] Sverre Grimnes and Ørjan G. Martinsen. ‘Alpha-dispersion in human tissue’. In: *Journal of Physics: Conference Series* 224 (1st Apr. 2010), p. 012073. ISSN: 1742-6596. DOI: [10.1088/1742-6596/224/1/012073](https://doi.org/10.1088/1742-6596/224/1/012073). (Visited on 25/03/2018).
- [12] D. Ayillon, F. Seoane and R. Gil-Pita. ‘Cole equation and parameter estimation from electrical bioimpedance spectroscopy measurements - A comparative study’. In: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Sept. 2009, pp. 3779–3782. DOI: [10.1109/IEMBS.2009.5334494](https://doi.org/10.1109/IEMBS.2009.5334494).
- [13] Chiara Canali et al. ‘Bioimpedance monitoring of 3D cell culturing—Complementary electrode configurations for enhanced spatial sensitivity’. In: *Biosensors and Bioelectronics*

- 63 (Jan. 2015), pp. 72–79. ISSN: 09565663. DOI: [10.1016/j.bios.2014.07.020](https://doi.org/10.1016/j.bios.2014.07.020). (Visited on 06/03/2018).
- [14] John G Webster and John W Clark. *Medical instrumentation: application and design*. English. Hoboken, NJ: John Wiley & Sons, 2010. ISBN: 978-0-471-67600-3.
- [15] Sergio Franco. *Design with operational amplifiers and analog integrated circuits*. 3rd ed. McGraw-Hill series in electrical and computer engineering. New York: McGraw-Hill, 2002. ISBN: 978-0-07-112173-6.
- [16] Stuart Ira Fox. *Human physiology*. 10th ed. Boston: McGraw-Hill Higher Education, 2008. ISBN: 978-0-07-294613-0.
- [17] Wikimedia Commons. *File:Animal Cell Unannotated.svg* — Wikimedia Commons, the free media repository. 2006. URL: https://commons.wikimedia.org/wiki/File:Animal_Cell_Unannotated.svg (visited on 26/04/2018).
- [18] Antoni Ivorra Cano. ‘Contributions to the measurement of electrical impedance for living tissue ischemia injury monitoring’. PhD thesis. Barcelona: Universitat Politècnica de Catalunya. Departament d’Enginyeria Electrònica., Feb. 2005. URL: <http://hdl.handle.net/10803/6333> (visited on 01/05/2018).
- [19] Wikimedia Commons. *File: Cell membrane detailed diagram en.svg* — Wikimedia Commons, the free media repository. 2007. URL: https://commons.wikimedia.org/wiki/File:Cell_membrane_detailed_diagram_en.svg (visited on 27/03/2018).
- [20] Karen Low et al. ‘Physico-electrochemical Characterization of Pluripotent Stem Cells during Self-Renewal or Differentiation by a Multi-modal Monitoring System’. In: *Stem Cell Reports* 8.5 (May 2017), pp. 1329–1339. ISSN: 22136711. DOI: [10.1016/j.stemcr.2017.03.021](https://doi.org/10.1016/j.stemcr.2017.03.021). URL: <http://linkinghub.elsevier.com/retrieve/pii/S2213671117301273> (visited on 27/03/2018).
- [21] Christer Spegel et al. ‘Chip Based Electroanalytical Systems for Cell Analysis’. In: *Electroanalysis* 20.6 (Mar. 2008), pp. 680–702. ISSN: 10400397, 15214109. DOI: [10.1002/elan.200704130](https://doi.org/10.1002/elan.200704130). (Visited on 06/03/2018).
- [22] Louise H Crockett et al. *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*. Strathclyde Academic Media, 2014. ISBN: 9780992978709.
- [23] Red Pitaya. *Red Pitaya Documentation, Release 0.97*. URL: <http://redpitaya.readthedocs.io/en/latest/> (visited on 02/11/2017).
- [24] *14 bit, 125 Msps Low Power Dual ADCs*. LTC2145-14. Linear Technology. 2012. URL: <http://cds.linear.com/docs/en/datasheet/21454314fa.pdf> (visited on 03/11/2017).
- [25] *14-Bit, 125 MSPS Dual TxDAC+ Digital-to-Analog Converters*. AD9767. Analog Devices. 2011. URL: <http://cds.linear.com/docs/en/datasheet/21454314fa.pdf> (visited on 03/11/2017).
- [26] MDN Web Docs. *TCP*. URL: <https://developer.mozilla.org/en-US/docs/Glossary/TCPs> (visited on 15/11/2017).
- [27] MDN Web Docs. *Websockets*. URL: <https://developer.mozilla.org/en-US/docs/Glossary/WebSockets> (visited on 15/11/2017).
- [28] Red Pitaya. *Github Source Repository*. URL: <https://github.com/RedPitaya/RedPitaya> (visited on 07/05/2018).
- [29] Pavel Denim. *Github Source Repository*. URL: <https://github.com/pavel-demin/red-pitaya-notes> (visited on 07/05/2018).
- [30] Anton Potočnik. *Github Source Repository*. URL: https://github.com/apotocnik/redpitaya_guide (visited on 07/05/2018).
- [31] Xilinx. *AXI GPIO v2.0 LogiCore IP Product Guide*. URL: https://www.xilinx.com/support/documentation/ip_documentation/axi_gpio/v2_0/pg144-axi-gpio.pdf (visited on 07/05/2018).

- [32] Xilinx. *DDS Compiler v6.0 LogiCore IP Product Guide*. URL: https://www.xilinx.com/support/documentation/ip_documentation/dds_compiler/v6_0/pg141-dds-compiler.pdf (visited on 07/05/2018).
- [33] Anton Potočnik. *Red Pitaya FPGA tutorials*. URL: <http://antonpotocnik.com/?cat=29> (visited on 07/05/2018).
- [34] Y. Mohamed Hassan et al. ‘Pyrolytic carbon microelectrodes for impedance based cell sensing’. en. In: *ECS Transactions* 72.1 (May 2016), pp. 35–44. ISSN: 1938-6737, 1938-5862. DOI: [10.1149/07201.0035ecst](https://doi.org/10.1149/07201.0035ecst). URL: <http://ecst.ecsdl.org/cgi/doi/10.1149/07201.0035ecst> (visited on 15/05/2018).
- [35] A Ruiz-Vargas, J W Arkwright and Antoni Ivorra. ‘A portable bioimpedance measurement system based on Red Pitaya for monitoring and detecting abnormalities in the gastrointestinal tract’. In: IEEE, Dec. 2016, pp. 150–154. ISBN: 978-1-4673-7791-1. DOI: [10.1109/IECBES.2016.7843433](https://doi.org/10.1109/IECBES.2016.7843433). (Visited on 19/03/2018).
- [36] *Universal Dummy Cell Operators Manual*. Gamry Instruments, Inc. 2011. URL: <https://www.gamry.com/assets/Uploads/Universal-Dummy-Cell-Operators-Manual.pdf> (visited on 15/05/2018).
- [37] ATCC. *PC-12 (ATCC® CRL-1721™) General Information*. URL: https://www.lgcstandards-atcc.org/products/all/CRL-1721.aspx?geo_country=no (visited on 24/05/2018).