Developing a capacitive biosensor towards an embedded system

Mattias Jönsson

Department of Electrical and Information Technology LTH
Department of Biotechnology LTH
Nohau solution

Advisor: Joakim Larsson, Martin Hedström, Viktor Öwall

June 17, 2009

Printed in Sweden
E-huset, Lund, 2009

Abstract

This report describes the steps to build a capacitive measuring system of a biosensor and how previous systems have worked. The method of measuring capacitance is called "Step response of an RC circuit". This means that a potential step is put on the biosensor and the charging current over the biocell can be measured. Out of the measured current wave the capacitance can be calculated. Today it exist two different systems, the original from the 90s which is reasonably stable but are not user friendly and the source code is not available so it is a locked system and it is not possible to add new features or modify. The second system is written in Labview and is user friendly but not as stable as the original system. Both systems are designed for a lab environment but there is a need for a new system that is stable, user friendly and small. The motivation of this project is to make a small portable system that could be used outside the lab environment and implemented in industry.
The new system was developed on a Linux platform to make it more cost efficient and new hardware to handle the analog signal was designed to get full control. An investigation of which platform to use was made so it would be possible to build an embedded system based on the new system in the future.

0.1 Acknowledgment

There are several people I want to give credits to in this project and that is Martin Nilsson from the department Electrical and Information technology from Lund university (LU) for his help in construction and building the analog interface box between data acquisition hardware and the biosensor. I want to thank my supervisor Joakim Larsson from Nohau of the support I received in the technical area. At last I want to give credits to Martin Hedström and Kosin Teeparuksapun from the department of biotechnology from LU that helped me understand the biochemical part of this master thesis.

Table of Contents

Chapter 1

Project specification

1.1 Background

A new sort of biosensor was developed in Lund that has the ability to change capacitance value when it gets exposed to a known substance. The big advantage of this biosensor is that it can detect a very small concentration of the known substance, and that in an online manner in a chemical process. However to make this sensor useful, a capacitive measuring system was needed. The first functional measuring system was built in Lund in the 90:s and there was an article written in 1999 [1]. The basic idea that has been used is a potentiostatic step to get a current response. This means that a pulse is put over the cell and a current response is measured by the system. The capacitance can then be calculated out of this current response. The first capacitive system that was built were based on an old platform that are not widely used today, the *Keithley 575 measurement and control system*. The software was programmed in Pascal using an PC 386MHz. The old system had some problems:

1. Sensitivity towards outside disturbance
2. No ability to change the software
3. Old software interface
4. Not user friendly

The strength of this system is that it is much more stable and reliable then the newer Labview version.
A new system was developed in Thailand based on the old Keithley system. This system was built on a *National Instrument platform*. The hardware that has been used is a PCI-6221 DAQmx (68-pin) card and an interface hardware that amplifies the signal from the current response and produces a down scaled potentiostatic step from the NI card. The program that has been used to control the system was programed in LabView. Labview is a graphical programming interface that are specially constructed to quickly and easily put up a measurement system. It was on this LabView model that this Master thesis is based upon.

1.2 Job description

The main goal with this project was to make a system that was more independent from labview and make it more robust against outside disturbance. To make this happen the project was divided according to following parts:

- Read in documentation of old systems
- Decipher Labview model and improve the solution
- Transform program to ANSI C
- Build new hardware interface
- Test the new system
- Construct temperature measurement
- Investigate embedded system implementation
- Build the embedded system

1.3 Target group

US Food and Drug Administration (FDA) have announced a new directive about better processing control of the manufacturing of drugs. The term of technology that can be used to solve some of this problem are called Process Analytical Technology (PAT) and are best described in FDA's guide lines to the pharmaceutical industry *"PAT can be defined as a system for designing, analyzing, and controlling pharmaceutical manufacturing through the measurement of critical quality and performance parameters"* [7]. It is in this area that the biosensor comes to use. Because it has the potential to become a sensor tool that can analyze the batch continuously of contamination of different sorts and give warnings under the processing steps of drugs. Because of this new directive, new demands is put on the pharmaceutical industries and thereby they will need new processing methods and new analyzing tools to manufacture drugs.

Chapter 2
Theory

This chapter is intended to give the reader a deeper understanding of some important topics in this project. Hence, here is described the theory of how the biocell works, how the new algorithm of calculating the capacitance value works and how the Finite Impulse Response (FIR) filters taps are designed.
Figure 2.1 [6] represent a complete measuring cycle and the different steps. In the first step a buffer solution is driven trough the system so that a steady baseline is obtained. Then a new solution is added to the system with some contamination which will affect the baseline. Difference between the two baselines can be directly connected to the concentration contamination that we are analysing for. Out off this difference in capacitance, it can be decided how much of the contamination it was in the solution that was analysed.

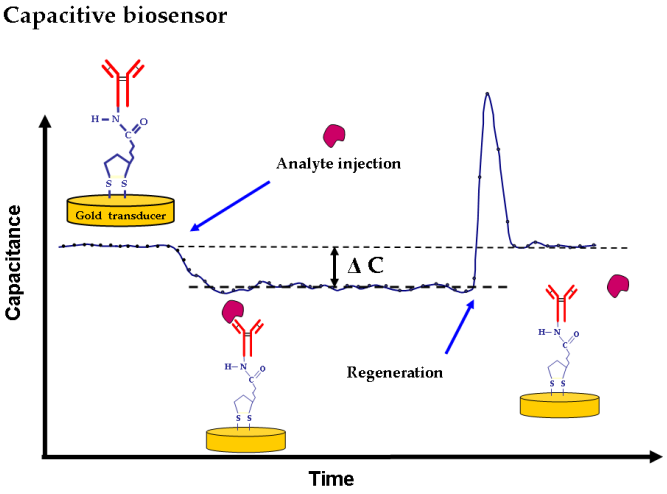

Figure 2.1: Baseline progress

2.1 The Biocell

In this section the biocell will be studied so that a deeper understanding on how the whole system works is gained.
Figure 2.2 describes how the biocell is chemically constructed. A gold surface is placed on a no-conducting surface made of glass or silicone. On the gold surface an insulating monolayer is placed that have the ability to bind the sensing "anchor" molecule. After this procedure the sensing molecule is added. The sensing molecule has the ability to only bind a special type of molecules. The different layers contribute to a total capacitance between 200-400nF.
The contribution of the analyte is quite small in comparison to the rest of the layers in worst case only 0.5nF. Therefore a good measurement method of the capacitance is required.
The Biosensor is then put in a flow cell and the measurement can begin[6]

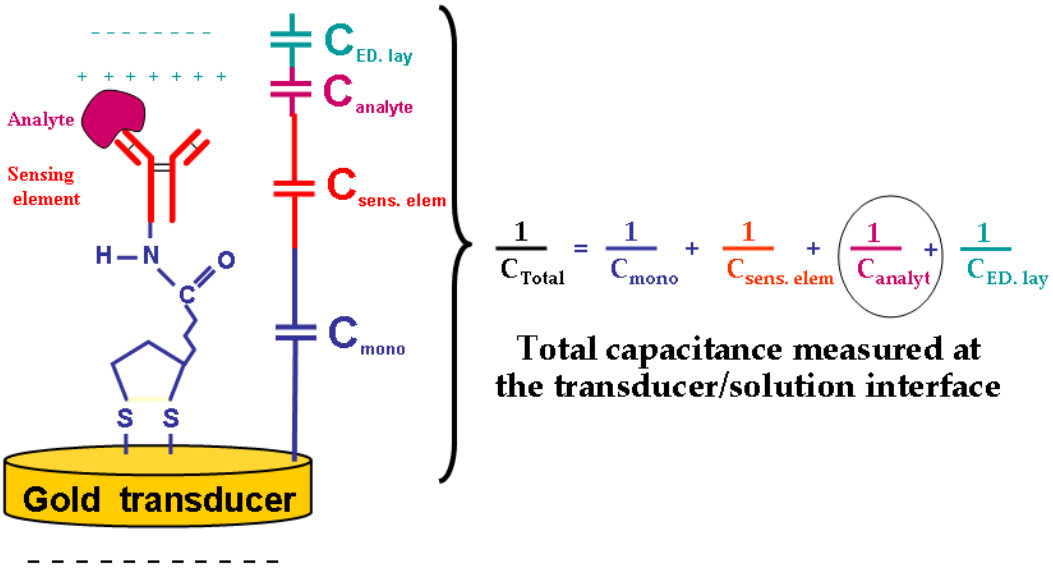

Figure 2.2: A simplified picture of how the biocell is constructed on a molecular level

2.2 The flow cell

In this section the flow cell system will be studied. In this project a three-electrode flow cell has been used. In figure 2.3, a simplified three-electrode flow system is presented. The function of the flow cell is quit simple, the auxiliary node and working node are placed at opposite sides in the cell and the reference is placed

close by the working node. Between the auxiliary and working node the buffer liquid is passing. The benefit with this type of cell compared to an electrode cell

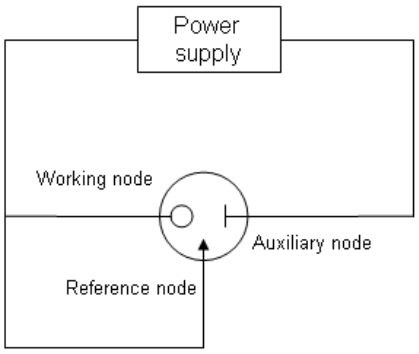

Figure 2.3: Three Electrode flow cell

that only has two nodes, working and auxiliary node is the third reference node that can be used to scale the resistance in the fluid that was use. Hence, it is very important to put the reference node as close as possible towards the working node reference [3].
A simple model on how an OP-amplifier can be used to compensate for voltage drop in the solution between auxiliary and working node are presented in figure 2.4. The OP-amplifier put out current into the cell that is represented by two resistances, R1 and R2. R1 is the resistance in the solution and R2 represent the resistance in the biocell at time t=0 when only the resistive part is active. The OP-amplifier put out a current so that the voltage over R1 is the same as on the OP-amplifiers plus in port. This coupling scale the solution resistance and make the measurement better.

2.3 Finite Impulse Response (FIR) filter

In the Labview model a low pass filter was implemented. The user had the option between a 3rd order Infinite Impulse Response (IIR) filter with the topology Bessel or a Finite Impulse Response (FIR) filter with 29 taps. Both had a cutoff frequency of $5kHz$. The filter first implemented was the FIR filter since the developer was well familiar with the FIR filter architecture and had implemented this sort of filter before. It was chosen to implement the FIR filter first and then if there was time over the IIR filter would be implemented.
The intention was to make the same FIR as in the Labview model. However there was no information in Labview how it calculated the filter taps. Since the sample frequency is adjustable it was necessary to implement filter taps that was not precalculated but that was directly calculated in the program. Therefore, the method to calculate the taps was transversal filter dimensioning through inverse Fourier

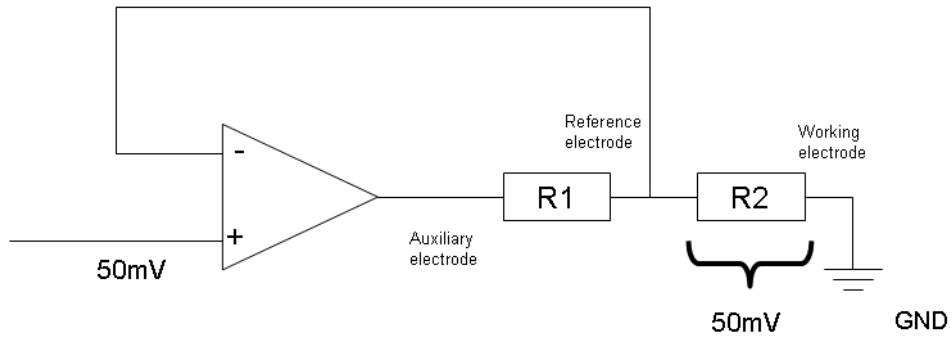

Figure 2.4: Simple model of an OP-circuit with a biocell at time t=0

transform [4]. In figure 2.5 a Bode diagram that describes the filter response. The conclusion of the diagram is that the filter taps gives a cutoff frequency of $5Khz$ The filter taps were tested and it was shown that the filter was filtering away important frequencies in the current response curve. The FIR filter have been tested in Matlab with sampled curve values from the new designed system. In figure 2.6 the curve with the lower amplitude represent the filtered signal and the one with higher amplitude represent the signal before the filtering. The conclusion is that this filter removes too much of the relevant data. It was decided to keep the FIR filter design in the program as an option so that in the future better filter taps could be developed.
The model that was used and tested in Matlab was implemented in ANSI C.

2.4 Line fitting

Instead of only trying to find a better filter, a second approach to the problem was made to use more data points from our sampled data and make a mean evaluation. Thereby to get ride of irregular noise that could make the measurement differ a lot between measurements on the same capacitance.
It was decided to make an exponential line fitting on the curve because of the need to make the measurement more consistent. A line fitting was also made in the Labview model but it was only implemented by a graphical block function and it made several different curve fitting attempts and then decided which was the best fit. The method that was chosen in the new program is a commonly used method called "Least square method"[5]. The formula below describe the method in short terms were the exponential curve formula 2.1 have been linearized formula 2.2.

$$y = Ae^{-t\cdot\tau} \tag{2.1}$$

$$ln(y) = ln(A) - t\cdot\tau \tag{2.2}$$

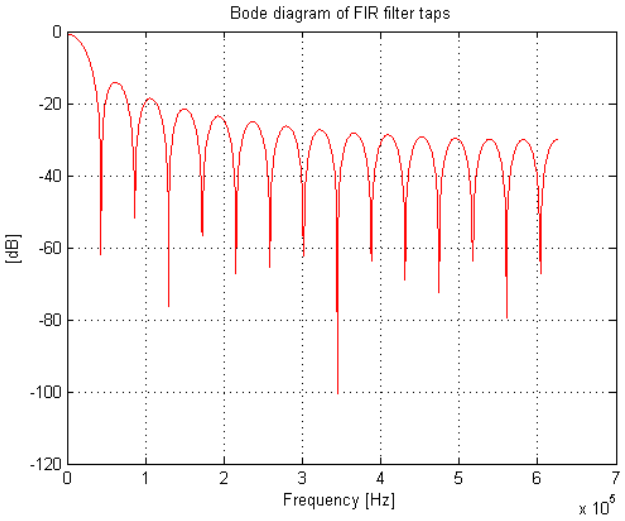

Figure 2.5: Bode diagram of low pass FIR filter taps with cut-off frequency $5kHz$

$$\tau = \frac{1}{RC} \tag{2.3}$$

After this is done the sampled data and the formula constants is put up in a matrix system as in formula 2.4. The system is then solved so that the constants A and τ become known. Now that the constants are known it is easy to calculate the capacitance value out of formula 2.1.

$$[B] \cdot [A, \tau] = [Y] \tag{2.4}$$

2.5 New calculation algorithm of capacitance

Since of the difference between the measured capacitance values of the same dummy cell was too big, a new calculation algorithm was constructed and compared to the old one that was used in the first C program. The new calculation algorithm can be described in a few steps

1. Extract curve form
2. Exponential line fitting
3. Calculate resistance value
4. Calculate capacitance value

The algorithm begins with extracting a curve between the peak value and $1/3$ of the peak value. This is done because when the signal strength decrease, the noise will change the curve more and make the curve fitting unreliable. The user

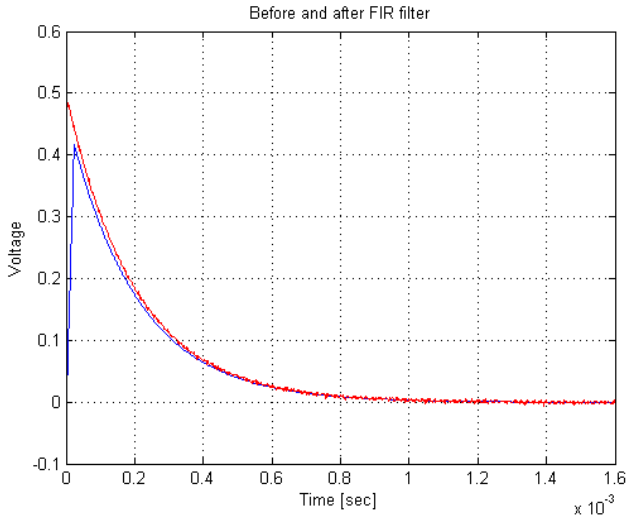

Figure 2.6: Before and after FIR filter blue before filter red line after filter

has the option to chose how many values after the peak value that should be valid for further calculation. Next step is to make an exponential line fitting on the extracted values to get rid of irregular noise. Out of the line fitting both the amplitude A and the damping τ become known in formula 2.1. The resistance is then calculated out of that the capacitance acts as a conductor at time zero. At time zero in formula 2.2 only the term $ln(A)$ remains and we have the current that runs trough the resistance and the voltage that are put over the resistance is the potential step. The resistance is then calculated with the help of the formula 2.5. Now it is only to use the formula $\tau = 1/R \cdot C$ to get the capacitance value.

$$R = \frac{V_{pulse}}{ln(A)} \tag{2.5}$$

2.6 Validate C-code algorithm

It was necessary to make a validation between the Matlab model and the C-code algorithm. Therefore, a test bench was made in C to test only the C file that handle computation and saving the data to file.
A curve that was sampled and saved with the new measuring system with the function "save waveform to file". The sampled array of values was then tested in the C programed test-bench to see if it produced the same value as the Matlab model that the C program was based on. The τ value and X value matched with the Matlab models value. The result of this match can be found in tabell below.

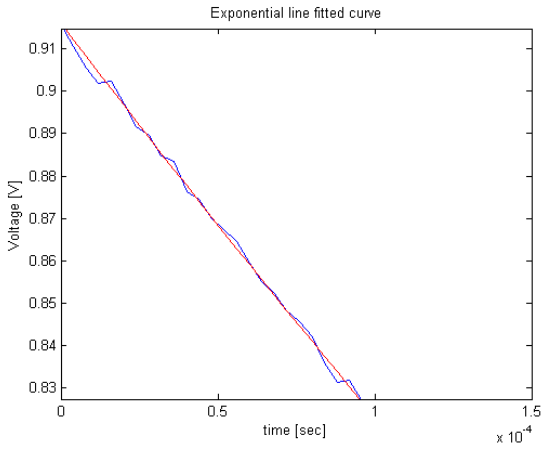

Figure 2.7: Before and after the curve been line fitted. Straight line after line fitting

Platform	*X*	τ	*Capacitance*$[nF]$	*Resistance*$[\Omega]$
C	-0.087608705942812795	-1069.8770689760202	342.513848	2728.902293
Matlab	-0.08760870594281	-1069.8770689760202	342.513848	2728.902293

This test shows that the C-code is at least valid down to 11th decimal towards the Matlab model which is good enough for this design.

Chapter 3

System description of previous systems

There was some improvement in the labview version compared to the old Pascal version of the Biocell measurement system.

- User friendlier
- Easy to change software
- Easy to add new features in program

There was some features that should be looked upon and be improved in the new design.

- Reduce noise interference
- Get measured value closer to the theoretical value

3.1 Mode of procedure

The first step in this project was to get an understanding how the labview program worked. One of the problems was that it did not exist any documentation of the labview system, figure 3.1. Therefore, quit a large amount of time had to be spent on finding out the ”inner working” of the system. It was decided in the beginning of the project that the Department of Electrical and Information Technology (EIT) from Lund University would make a copy of the analog hardware ”black box” between the National Instrument card and the biocell from the previous labview system. This would make it possible to work in parallel with Lund University and Nohau. A virtual signal was simulated and applied to the in signal in the system to be able to run and test the program. The tool ”create analog signal” was used to create a virtual channel with an analog signal. The formula 3.1

$$V_{out} = A \cdot e^{-t/(400 \cdot 10^{-6})} \tag{3.1}$$

describes a fading exponential curve. The formula was developed empirically to get a good capacitance value from the Labview program.

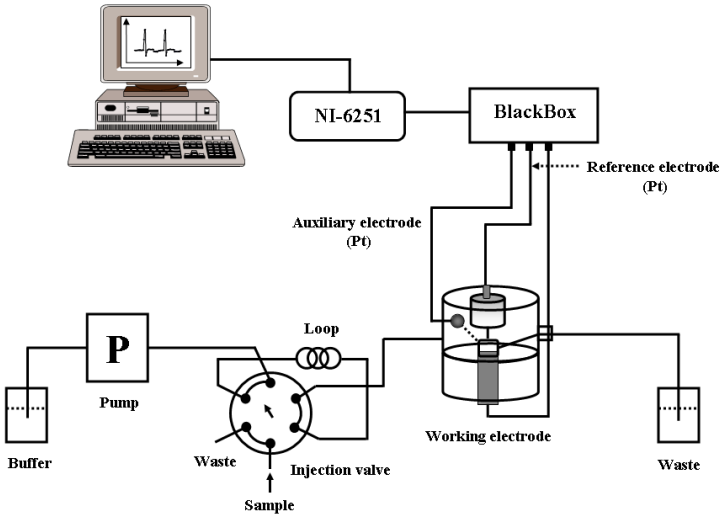

Figure 3.1: Overview of system

The Labview tool "Create Analog Signal (LabVIEW SignalExpress)" was chosen because it was intuitive to use and it was possible to change sample rate and sample block size. By probing the signal lines in the labview program and use the function highlight execution the program was decoded. At this stage there was some issues about some of the calculation in the program and some of the outgoing signals. The first problem was that the program was equipped with unknown constants in the calculation of the capacitance. These constants had nothing to do with the theoretical method of "Step response of an RC circuit" that was chosen to calculate the capacitance value.

$$I_{max} = MaxValue+ \mid \frac{dV\backslash dt}{sampelRate} \cdot MaxValueTimeStamp+ \mid \cdot 1.3 \qquad (3.2)$$

- MaxValue = the maximum value in the in sampled current answer vector
- dV\dt = slope of the current vector
- MaxValueTimeStamp = time stamp of the maximum value

$$Cn = (\frac{5 \cdot \Delta \cdot 2 \cdot 10^5}{10^6 \cdot R(ln(\frac{I_{max}}{I_{12}}) - ln((\frac{I_{max}}{I_{11}})) \cdot SR} \qquad (3.3)$$

$$+\frac{5 \cdot \Delta \cdot 2 \cdot 10^5}{10^6 \cdot R(ln(\frac{I_{max}}{I_{22}}) - ln((\frac{I_{max}}{I_{21}})) \cdot SR}) \cdot \frac{1}{2}$$

- Delta = number of steps between the calculation points I_{t12} and I_{t11} from I_{max} vector
- R = calculated resistance value
- I_{max} = peak in Imax vector
- I_{t11} = 10 values after max point in Imax vector
- I_{t21} = 15 values after max point of Imax vector
- I_{t12} = 20 values after max point in Imax vector
- I_{t22} = 25 values after max point of Imax vector
- SR = sample rate

The formula 3.2 is a way to estimate theoretical top value. The constants in this formula was not documented before and it could not be mathematically or numerically proven why they had chosen the constants. It was decided not to be investigate futher. Formula 3.3 that calculate the capacitance value uses three constants 5, $2 \cdot 10^5$ and 10^5. This three fixed values have nothing to do with the theoretical formula 3.4 but the three values counter each other out in the formula 3.3. The theory that is applied can be found in [2].

$$C = \frac{-t}{R \cdot ln(\frac{I_{max}}{I_0})} \tag{3.4}$$

To understand the system better a study of the signals was made on the labview outgoing signal with an oscilloscope. When the "black box" was mapped through reverse engineering a real connection could be made between the hardware and software. It was possible to see what the software generated signals were doing in the hardware. After that the system was tested with a known dummycell (see figure 3.3) and some conclusion could be made.
The system did not behave as expected, it measured the double capacitance value of the dummycell. It was decided to check for the error in the following steps.

1. Check in-signal to dummycell system
2. Check outgoing signals from dummycell system
3. Check software calculation

When studying the pulse signal to the dummycell, an error was found. On the pulse $18ms$ before the real pulse a $5ms$ long downward spike is generated (see figure 3.2). For some reason the National instrument are releasing the signal for 3.8ms before it puts out the new signal. The behavior stopped when the pulse stopped acting as a trigger signal. There have not been found any solution on the problem and due to lack of time this problem have only been notice. This problem is not included in this master thesis because this is a bug in National instruments hardware.

$$R = \frac{V_{in} \cdot gain \cdot sensitivity}{I_{max}} \tag{3.5}$$

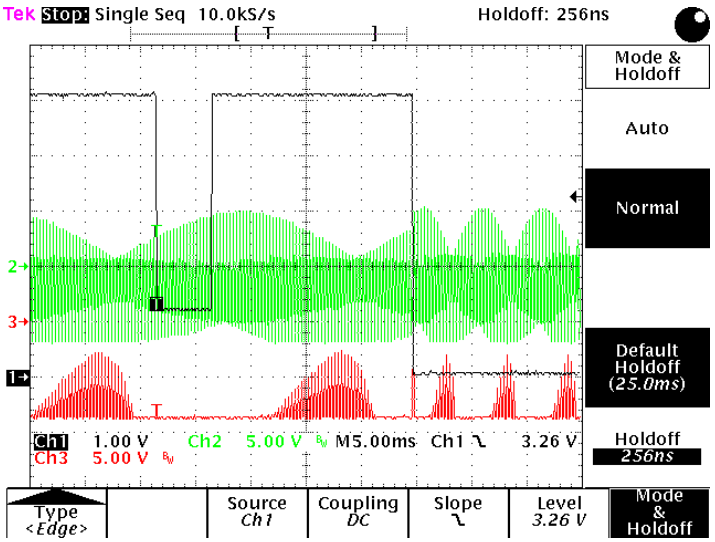

Figure 3.2: Oscilloscope picture of pulse train from National instrument card

- V_{in} = voltage on pulse
- gain = Gain chosen on the analog in signal
- sensitivity = another gain signal on different OP-amplifier

The National instrument hardware releases the signal and let it float before it put out the new signal. This spike makes the system unreliable because it is unknown if this pre-pulse charge the capacitance before the real measurement begins. After verifying, we could see that the spike disappeared when the pulse signal stopped acting as trigger of the in sampling channel. The spike disappeared when the sampling channel settings were cleared after each sampling. This change made the measuring procedure longer because the NI hardware had to re-install all the settings.
Despite this correction the program still showed the double capacitance result. When the hardware was mapped to a circuit diagram a better understanding between hardware and software could be made. It was found that the resistance that was calculated from the current response did not match up with the real resistance in the hardware. The error was that to measure the current, a voltage measurement was made over a well known resistance but the software and the resistance value in the hardware was not the same. Instead of the correct value $2kohm$ respective $4kohm$ it had $500ohm$ and $1kohm$. After closer investigation it was found that there existed two different hardware versions and four different software versions and that the version that had been presented in the beginning of the project was written to another hardware. A new software version was needed and the version *Capacitive Analyzer1.7.vi* was chosen on recommendation from the Biotechnology department LTH. In this version the resistance constants

matched. The new software was stripped off the curve fitting approximation and the low pass filter because they could not be verified in detail. The FIR low pass filters taps calculation was not documented at all and only how many taps and cutoff frequency was available. The new file was extensively tested and the result can be found in appendix A.

3.2 Analysis of existing Labview system

A deep analysis of the previous system was needed because off the lack of information and documentation. The analysis was executed in the following steeps.

1. Familiarise with labview and decipher the program
2. Test the program extensively
3. Document and study the output signals with oscilloscope

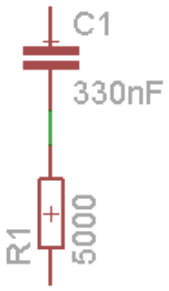

Figure 3.3: Simple dummy cell used in the beginning of the project as a reference

3.2.1 General

This study included extensive work on how the LabView architecture for the biosensor works, and how the I/O signals are configured. What should be stressed is that the technical documentation of the LabView model from Thailand was missing. Therefore, the I/O–signals had to be studied and documented using an oscilloscope. Some documentation of the hardware that transforms the signals from the LabView card (National Instrument PCI 6221) to the biosensor was found. Unfortunately the documentation was in Thai and therefore translated to English. It turned out that this document did not match the hardware. Reversed engineering was used to get all the information about the system.

3.2.2 Applications in LabView

An overview of the software GUI and functions can be found in appendix B. The program have a user friendly GUI which are easy to handle and labview is a great

program to plot curves. The curves that are presented to the user in this version is the raw current step from the A/D converter, a capacitance baseline of all the previews calculated capacitance value and a resistance baseline curve.

3.2.3 LabView program execution step

The system had to be carefully examined so that it would be possible to get a full understanding on how the program worked.
When the program is executed, a down count begins. When it reaches it's end value it triggers off a pulse at measure time zero. At this time a square pulse is put on an analog out port that excites the biosensor. The square pulse is a trigger pulse for the analog in port which begins to sample data from the system. A 5-bit digital out-port is activated at time zero. This controls the analog black box between LabView card and the sensor. This port controls the gain of the in signal from the biosensor. Schematics for this can be found in appendix C, figure D.1, where the switch u13 is the control of sensitivity. The input is processed and calculated, then saved in different files. The labview filing system is not flexible because it always save the files in the same catalogue C:\Capacitance Data. The baseline curve is always saved in a text file with the name "year-month-day-hour-min-sec.text", e.g. *2009-03-04-15-43-16.txt*. The file consists of three columns, first column contains all the measured capacitance values, second column contains capacitance/area, and the third contains the resistance values. The second file that is optional to save is the curve data file. The program name the file automatically as the last one, e.g. *WaveFormden 2 apr 2009 161443.txt*. The file contains three columns, the first contains the time x-axle value, the second pulse number, and the third the current y-axle value.

3.2.4 Data processing and calculation of input

Data from the inport is being filtered through a low-pass filter with a cutoff frequency of $8kHz$. The low pass-filter is of type IIR-Bessel order 2, which can be changed to 29 taps FIR filter. Each sample is given a time slot which is saved in a matrix where one column is the sample values and the other one the time slot. The result is then saved to a file in the folder "C:\Capacitance Data". The time vector that is saved and needed in the future calculation of exponential curve fit is created by a special function in LabVIEW program "Get Waveform Components", which extract the time step between two samples. Then the program iterates the sample vector trough a for-loop which gives a time slot to each sample. The pulse number is also added to the matrix which is saved to the same text file as before in a third column. A new vector is created out of the old one with the peak value from the old current vector as the first value and which then takes x number of sample points after the maximum value. //The number of points is determined either automatically or manually. The automatic vector length is determined as follows. Maximum values are compared with subsequent samples and it cuts the vector when the values are lesser than 1\3 of the maximum value. The new curve is used in a curve fitting function in labview "Exponential Fit" and the function uses the method "least square". From this exponential fitted

curve the maximum value is taken and the capacitance value is later calculated from the fitted curve. A linear fit is also made on the sampled data to get a slope value. This is used to approximate a new max value with help of formula 3.2. Capacitance values are calculated in the end by formula 3.3.

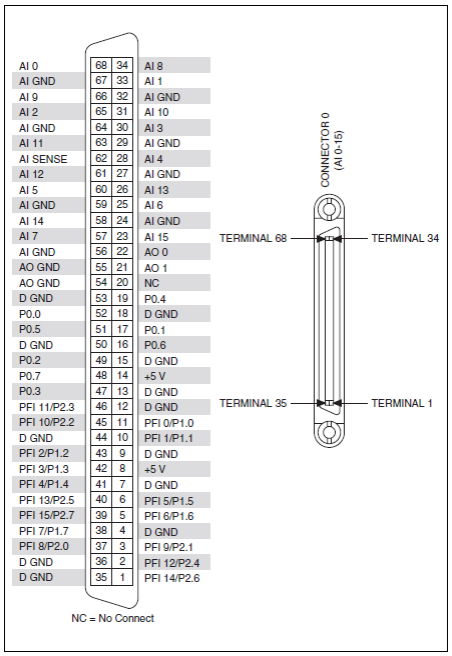

Figure A-2. PCI/PXI-6221 (68-Pin) Pinout

Figure 3.4: Pin out table from National Instruments user manual of PCI-6221 card

3.2.5 Pin configuration

All the following pins can be found in figure 3.4 A constant analog voltage between 0.4 V to 8V is put on pin 21 AO0. At the same time a pulse train is put on pin 2 PFI12\P2.4 which can be seen in figure 3.5 the signal is named DO2. There are five digital outputs for relay control 0-5V, pin 52 ”P0.0 gain100”, pin 17 ”P0.1 gain200”, pin 49 ”P0.2 gain500”, pin 47 ”P0.3 gain1000” and pin 19 ”P0.4 sensitivity X1 or X2”. The range on the analog measuring channel is -10V to +10 and is located on pin68 AI0.

3.2.6 Test result

The Labview program *Capacitive Analyzer1.7* system was tested extensively with a dummy cell consisting of a capacitance of $100nF$ and a resistance of $5kohm$ in series figure. Connected between the reference and the working node figure 3.6.

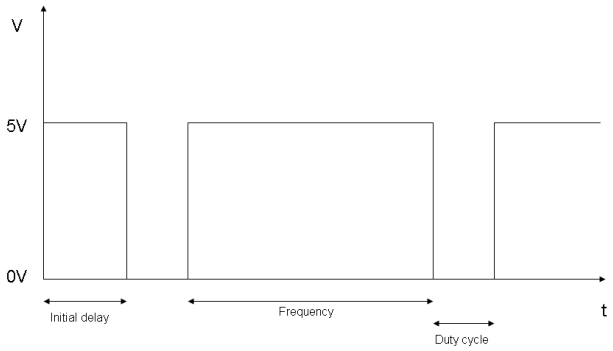

Figure 3.5: Pulse train

Between auxiliary and reference node a capacitance of $47nF$ and two resistances of $10kohm$ respective $1Mohm$ was placed in parallel to the dummy cell. The result of the test is located in appendix A. Their seems to be an error in the hardware every time the sensitivity was set to X2 because the system failed to measure. For some reason the current did not go to zero as was expected instead it stables on $0.2V$. A graph of this failed measurement can be found in appendix A figureA.1. The error which caused this kind of behavior couldn't be found that explain this failed measurement. This was consider not to be a problem because in this project we intend to build a new hardware and write new software. The old system is only a reference so it is enough to know that sensitivity X2 doesn't work. In appendix A, figure A.2 a couple of curves can be seen with different gain. They show how the analog low pass filter smoothing and filter away the peak value more and more. This because when the gain increases the measured values is putted higher up in frequency and make the result worse. In appendix A figure A.3 two curves can be seen that describe before and after the digital low pass filter. It can be seen that the signal after the low pass filter have been delayed and decreased in strength. In appendix A figure A.4 it can be seen before and after digital low pass filter at different gains and when the gain increases the difference between top value before and after the filter increases.

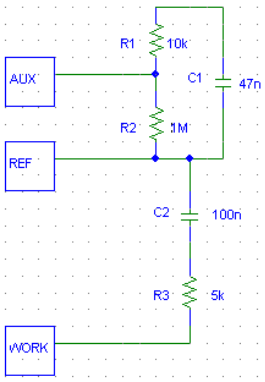

Figure 3.6: Dummy cell

Chapter 4
Previous interface box

In the old design a locally made hardware in Thailand was constructed. To be able to get a full understanding how the measuring system work the hardware had to be thoroughly investigated.

4.1 Function description of old interface box

In this section the Interface between National Instrument lab card and the bio-cell also named "BlackBox", will be presented. All the components that will be spoken off in this section can be found in appendix D, figure D.1.The black box function is to amplify down the outgoing pulse 8 times and to amplify the current response to a measurable voltage level over either resistor R18 or over R16 respective R17. An operation amplifier U12 is a voltage to voltage amplifier and it puts out the final pulse to the Biosensor. The voltage is then measured over the outgoing resistor on U12. Which resistor it will measure over is decided by the switch U13 (Sensitivity). Amplifier U7 is an instrument amplifier that amplifies the signal to measurable levels. The switches U8-U11 makes it possible to change the gain factor on amplifier U7. Since the resistance values change when the switches open and close. There are four different gains in the previous interface box 100, 200, 500 and 1000.
There are several connection pins on the black box for digital ports to operate the gain and one digital port to operate the sensitivity option. One digital clock pulse to operate the pulse to the bio cell, one analog port to give the voltage strength of the pulse and one analog out port to measure the current answer from amplifier U7.

4.2 Analysis of previous interface box

The black box was tested to see how good the implementation was and if it would be good for the new solution. A few bottle necks in the design were found. The instrument amplifier U7 (INA118P) have a bandwidth of $70kHz$ at gain 100 which is good enough but if the gain is increased to 1000 the bandwidth decrease to $7kHz$ and it is then a big risk that important data will be missed. There are two solutions presented, either to replace U7 amplifier with an amplifier which is

three times faster or replace the amplifier with an OP amplifier that have a higher bandwidth.
It was investigated if the system to regulate the gain level with relay could be replaced so that it will be possible to shrink the solution. The huge amplifying range between 100-1000 times makes it difficult to replace the relay. The concrete solution in this analysis was to use four instrument-amplifiers with different gains connected to a mux that choose which amplifier that will be active. However, this will not shrink the solution so it was decided to keep the relay solution for now. It was found that the power supply to the relay was to low. The relay need $12V$ to operate correctly but was only given $9V$. The component (74LS05) that feed the relay was only specified to drive a voltage up to $5V$. The device had five trim potentiometers installed that made it possible to trim the solution to a better performance. However the trim potentiometers had a resistance of 200-500ohm which make it hard to trim when thy only should adapt in the range of 4-15ohm. The passive resistor net filter on the outgoing analog port to the National instrument card is something that will be taken away in the new design because it is not proven that it makes the solution better. Instead it will cut away a lot of frequency that could be important to the measurement.
A MatLab model over the filter was made with approximation of infinite resistance on the input resistance on the National Instrument card. The filter begins to attenuate at $2kHz$ and a diagram of the calculation can be found in figure 4.1.
The power supply was implemented in the same box as the rest of the design which make it less robust against disturbance from the power grind plus that the transformation in itself generate noise that the measurement can pick up.

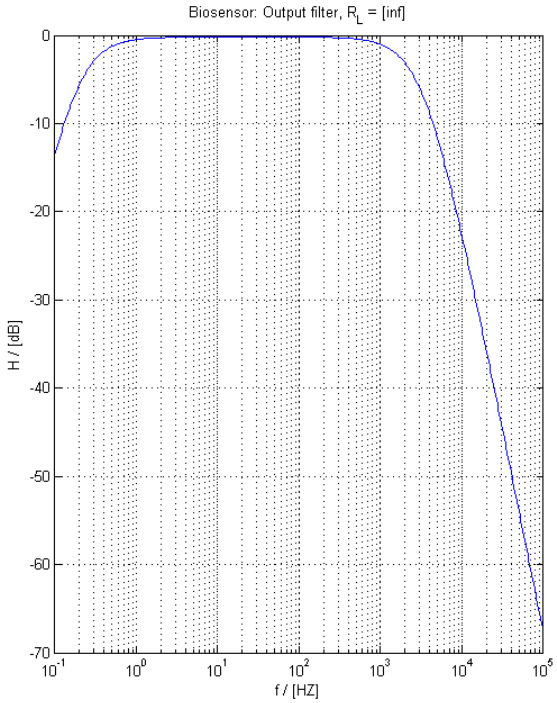

Figure 4.1: Dummy cell

Chapter 5
Tools and Test environments

An investigation of which tools and environment that would be used within the project had to be made. In this section the choices between different tools and environments will be investigated and motivated.

5.1 Development tools

There were several choices that had to be made before the tools could to be chosen. The develop environment Minimalist GNU for Windows (MinGW) with msys was chosen because it gives a Linux environment under Windows. This gives the advantage that it will be possible to run the new program with National instruments Windows drivers. To get full control over the project it was chosen to use a makefile system to build the project instead of using eclipse or codeBlocks that have project management tools. Emacs was chosen to be the editor to write the C-code in because it was developed for UNIX.
Several tools was tested in attempt to transform the Labview charts to code based form.

- Matlab
- Measurement studio
- LabWindows/CVI
- National Instruments C-library

The first choice landed on Matlab because it was a well familiar program for the developer. However, it was of no use when it was realized that National instrument did not give any support in Matlab towards clocked digital I/O in the hardware which are used to generate pulse trains.
Measurement studio is a plug-in to visual studio that makes it possible to build C-coded programs quick and even get access to all the Labview widgets. LabWindows/CVI has the same feature as Measurement studio but it is a complete programming editor so it runs without visual studio. Both programs seem to be a good choice to transform the labview program to text based code instead of the graphical code language in labview. National Instrument also gives out an ANSI

C library and header file for free because there was no full license to use Measurement Studio or LabWindows/CVI, only trial version. The final choice was to use the ANSI C library and free license editor to write the new program because it was free license and could be compiled in Linux. The benefits to use the free library file is that the project was not depending on Labview any more and the program could be developed directly in MinGW environment. By using MinGW we simplify the transition to convert to full Linux OS in the end.
A program to debug the ANSI C code was needed and there was two choices, Valgrind and GNU gdb 6.3 Pro Toolkit Insight Debugger(insight). Insight was chosen as the debugger because it has a graphical interface and is much more user friendly then Valgrind.

5.2 The OS

The first platform that was chosen was Debian which is a Linux distribution. Because of limited support from National instruments on Linux systems, a compromise between windows and Linux had to be made. It was in our interest to keep the development in a Linux environment so that it would be easier later to compile down the program to embedded Linux system. There were two different Linux environments to be driven from windows Cygwin and MinGW. MinGW was chosen because it had better support of a graphical widget package called GTK+. The compiler gcc was chosen because it is free to use and it's standard to use in Linux.

Chapter 6
Future platform

In the choice of future platform the embedded platform with a Linux kernel was preferred because if the solution was implemented with a Linux kernel the solution would be vendor independent. If the new system was implemented in Linux it would be possible for everyone to develop on the new platform because of Linux is general.
If the biosensor was implemented on an embedded system the future usability will increase. Instead of only being implemented in pharmacy factory and laboratories it would have the potential to become very usable in medicine as a diagnostic tool to survey of very specific substance in a patients blood stream. The biocell also have the potential to become an instrument to monitoring the environment of biochemical weapons or analyze your drinking water. In both cases it is necessary to have the solution in a small hand held embedded system. Embedded system is also more stable and robust than a PC system because it's internal variable is fixed and it is not disturbed by outside signals like a PC.

6.1 Criteria of new system

- Low power
- Low cost
- Open source compatible

If the biosensor will become a hand held unit like a cell phone low power is the biggest demand on the design. The embedded system should have as low cost as possible because then it is easier to get the industry interest of the biosensor. If the system is open source compatible (e.g. Linux kernel), the cost of the system will be lower. The new system should be able to handle the following tasks

1. Handle the capacitance algorithm
2. Create pulse trains
3. Save the information on file
4. Present the information
5. Simple and user-friendly graphical interface

6.2 Architecture of the embedded system

There are many different architectures to consider in this project and these are the architectures that have been analysed:

- MIPS
- DSP
- FPGA
- Power PC
- ARM

The MIPS architecture is Linux compatible and it exists in a wide range of different processors to choose from. However it has not been found any good test board to experiment with and it is in general more expensive to use than e.g. the ARM architecture.
The Digital Signal Processor(DSP) is a very interesting architecture because of it's hardware implementation of a pipelined architecture that makes it very powerful to implement signal demanding operation, for example digital filters. The DSP architecture gives some special feature:

- Hardware modulo addressing
- Separate program and data memories
- Efficient hardware implementation of DSP algorithms
- Highly parallel multiply-accumulate (MAC units)

The hardware modulo addressing system gives the system ability to use circular buffers which is good to have in e.g. FIR filter designs. The DSP processors have some hardware implemented arithmetic which could be very useful in the new measuring system. It also has a parallel MAC unit which makes the DSP processor suitable for Fast Fourier transform (FFT) implementation[8]. The DSP processor seams like a good choice to implement the signal and calculation part in the design. However, it is not ideal to handle the OS or to handle the graphical interface. A mixture of a DSP and another processor architecture would be perfect.
An Field-Programmable Gate Array (FPGA) is a platform on which it is possible to implement many different processor architectures, like MIPS and ARM. The FPGA is first of all a platform to implement hardware with a hardware programming languages, e.g. VHDL and Verilog. An FPGA will not be used as a future candidate to implement the system because it is unnecessary to first buy a platform and then buy a microprocessor platform to implement on the FPGA to drive the Operating System(OS).
Power PCs are usually used when there is a need to handle big loads of computer traffic do to it's many I/O registers. The PowerPC is a very powerful in computation of an embedded system but it is a power consuming system. Because the new system is a self contained system and we want to have the possibility to make a handheld unit, the PowerPC was ruled out[9].

The ARM architecture is the most commonly used architecture of embedded systems. ARM doesn't manufacture processors themselves, instead they license out the right to use their architecture. This have contributed to that today there are a wide range of complete embedded system that can be directly used to implement our system. The other benefit with this architecture is that it is ported to many different OS, especially to the open-source world and there exist good cross-compiler for the ARM architecture. Because of the portability to the open-source world makes it cheap to use. There are a lot of time to pre-calculate in our system and there is no need to implement a DSP as a co-processor at this time moment. Instead, the ARM architecture will satisfy the criteria of the system.

6.3 Investigation of chosen embedded hardware

An ATMEL AT91SAM9263 Evaluation Kit was chosen as the platform to port the measuring system to because it had the following features:

- LCD controller
- USB host 2xFS
- Touch-screen
- ICE interface

The Microprocessor can handle 220 million instructions per second with a clock speed of 200MHz. The evolution board has a touch screen interface on which a graphical interface of the new program could be implemented. It would then be possible to remove the PC from the system and control the measurement from the touch screen.
The embedded platform have the ability to be saved directly to a portable USB-memory through the USB interface on the chosen hardware.
With an ICE interface the developer has the advantage to program and debug the device through a hardware debugger. The chosen hardware will satisfy the criteria put up in section 6.2 "Criteria of new system". This hardware has more resources both in memory and computation speed than what is needed in the system of today. In the future the system may grow and have more complex calculations and then there are hardware resources to directly implement them.

6.4 New software development

The system was ported to line based code so it would be possible to compile the code with a conventional C compiler and later implement the code in an embedded system. First, a try to port the system with Matlab was done but it failed because of lack of support between Matlab and the National instrument (NI) hardware. At last it was decided to write the new code directly in ANSI C and C++. The first part that was written of the new code was the new temperature sensor program. It was decided to write this first because the new function could be tested continuously with the new NI card connected to simple Wheatstone

bridge. NI had given some example code how the different applications should be implemented but there was some part of the program code that was regarded unreliable. All internal error handling from the hardware was taken care of with goto commands and macros. This sort of coding can give difficult errors that are hard to find. Therefore, this coding style is to be avoided. It was decided that this part had to be rewritten to a robuster code without goto commands involved. When the decoding of the Labview started, it was realized that the program did not have any rules in which order the signals should be executed, only that the in sampling channel was trigged on the pulse signal. This problem was first apprehended when tests were made on the system presented in appendix A. Every time that one of the parameters sample rate, voltage strength or gain, was changed, the measurement failed the first time. This was because the Labview software set all the output signals at the same time which is working if the right values are already set on the static ports like gain and voltage strength. If some parameters had been changed the relay of the gain have to change position and the voltage could be changed under the measurement which make the measurement to fail. This error will correct itself when the code becomes sequential and the developer gets full control on what is happening in the program. In figure 6.1 a program cycle is described in which order the program is executed. When the program

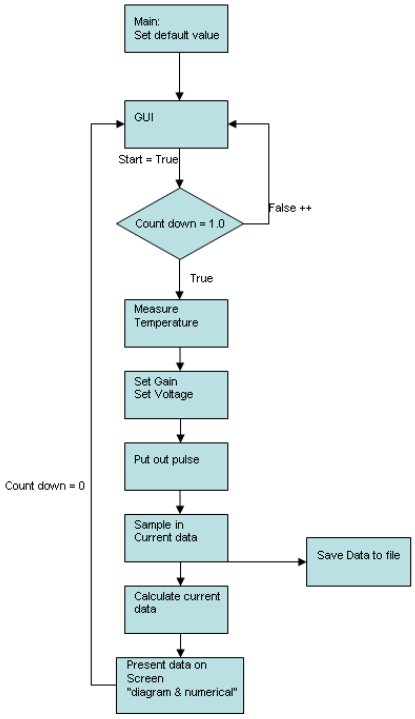

Figure 6.1: Data flow graph of program BlueBox Analyzer

starts it begins to set all the default values in the program, starts the graphical interface(GUI) and put it self in an idle state to check if the start button has been pushed. When the user presses the start button a countdown starts and when it reach its end value it first measure the temperature and then set the gain and voltage level. Following the pulse is put-out and the in sampling of the current answer begins. After the measurement is done the data are computed and saved in files that the user decides. The values are then presented numerically and in graphs in the GUI and the measurement cycle begins again until the user pushes the stop button.
The algorithm that was used in the original Labview program, presented in section 3.2.4 ”Data processing and calculation of input”, was not good enough because the line fitting algorithm from labview was imbedded in a function and couldn't be directly transferred. Therefore, completely new one hade to be made that uses more samples. This new algorithm is described more closely in the theory section 2.5 ”New calculation algorithm of capacitance”. It was implemented and tested in section 2.6 ”Validate C-code algorithm” the test of the algorithm can be found. A comparison was made between the new and old algorithm. The result is presented in section 6.4 ”Tests on system”.
In figure 6.2 the hierarchy of the program is presented with the main program at top and the GUI is an under program which have its own under program so it's able to execute the measurement cycle.
The graphical interface that was chosen is named GTK+ which is a widget standard library developed for Linux but could also be driven from windows which make it flexible.

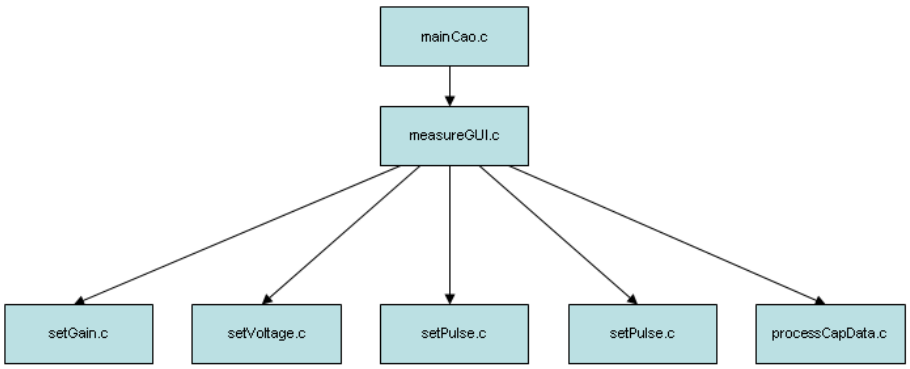

Figure 6.2: Schematic hierarchy of program BlueBox Analyzer

An instruction manual of the new program can be found in appendix C ”Instruction manual of Capacitance Analyser”.

6.5 Tests on system

Test and comparison old and new algorithm

A test was made to evaluate the new algorithm embedded in the new hardware. The experiment was made with following parameters: sample rate: $1.25MHz$, gain: 50 and voltage: $50mV$ the capacitance that was measured had the value $100nF$. The capacitance result is presented in figure 6.3. The lower curve show the old algorithm implementation and the upper curve represent the new algorithm implementation. A statistical mean value calculation on the fluctuation was made on both capacitance arrays with formula 6.1.

$$C_{fluctuation} = \sum_{j=0}^{n} \frac{\mid C_j - C_{j+1} \mid}{n} \tag{6.1}$$

The average fluctuation on the old algorithm was $0.3375nF$ and $0.0435nF$ in the

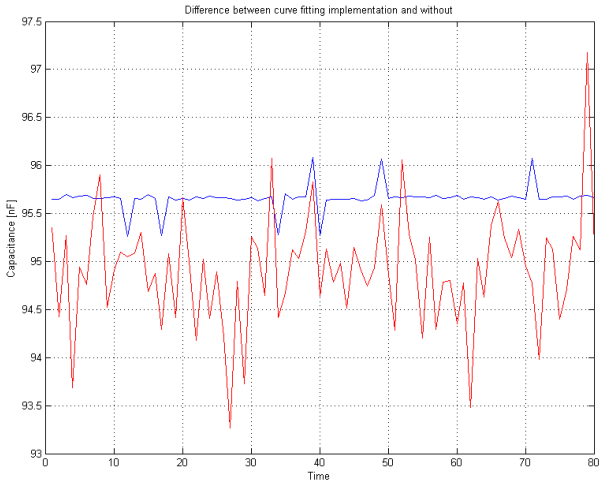

Figure 6.3: Capacitance with old algorithm red curve and blue curve new algorithm

new one which is 7 times better. This means that the new system could be used with accuracy lower than $0.5nF$ that was a criteria to be able to use the system on a capacitive biosensor cell. The difference on max and min value for both measurement was $0.8197nF$ on new algorithm and $3.9065nF$ on the old algorithm. So after this evaluation test it was decided to keep the new algorithm.
The values never reached the dummycells $100nF$ but there are some possible error sources to consider the capacitor that was measured have a error marginal on a few percent make adjustment on the gain with measurement on the new hardware.

Chapter 7

New construction

7.1 New Interface box

A new Interface between Labview card and biocell was constructed in an attempt to decrease electrical interference. By constructing a new hardware it was given full control of the hardware and a well defined construction. The improvement that was done is as fallow

- Power supply outside housing
- Virtual ground (Working node not connected to ground)
- No unspecified analog filter
- Temperature sensor

The power supply unit was put outside the housing of the sensitive measuring system to minimize the interference from the power grid plus the interference from the switching noise from the power converter.
Instead of grounding working node it was given a virtual ground as in the old construction from Lund. The benefit with a virtual ground is that it will not be sensitive towards disturbance from unwanted ground loops from the PC or other parts of the measurement system.
The analog filter on the Labview models interface box was dismissed because their was not found any good reason to keep them.
A new hardware was added that made it possible to measure the temperature in the cell. In figure 7.1 a schematic description of the new analog interface is shown. On the serial port the temperature sensor is connected.
There are future plans to put all connections to the Biosensor on the serial port to minimize cable connections. The interface has several connections to the National instrument card; it has three digital ports to set the gain level, one input digital port to open and close the pulse port, one analog inport to set the pulse strength and two analogs out ports with the signals from the biosensor and the temperature sensor.

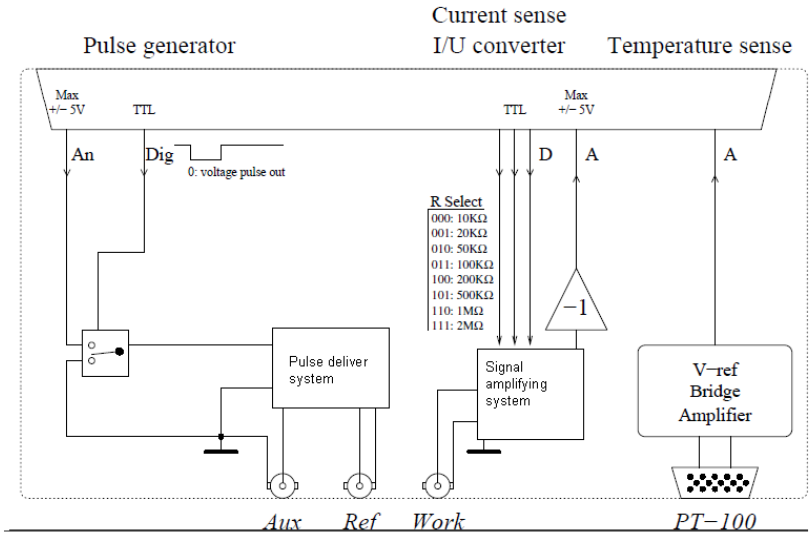

Figure 7.1: Simple schematic of Biosensor interface between The Biosensor and National instrument card

7.2 General temperature sensor

A new feature off the system was added an on request from the biotechnology institution at Lund University. The temperature sensor does not have any real function in the biosensor system yet. In a future version of the design it is intended that the measuring system will be able to compensate the measurement of temperature variation. The measuring system will only register the temperature at each measurement because there are not specified how the data should be used yet. This temperature function makes it possible for the user to develop and different solution to compensate the measurement of temperature difference.

7.3 Development of temperature sensor

To be able to achieve a smoother baseline a temperature sensor was integrated in to the system. The temperature data will later be integrated in the calculation algorithm to compensate the capacitance answer against temperature changes and to get a smoother base line. The requirements of the temperature sensor was an

accuracy of half degree, a response time of under 3 seconds and a measure interval from $0°C$ to $70°C$. The choice of the sensor was a pt-100 made of thin film technology which can endure a tough environment. The sensor has a measured range from -50$°C$ to 180$°C$ which is enough because if we go under $0°C$ the buffer solution would freeze or if the temperature go over 100$°C$ it would boil. The sensor have a response time of 2 seconds in liquid. A Wheatstone bridge was chosen to read the resistance change in the PT-100 element. The Wheatstone bridge was chosen because of it's stability, simplicity and accuracy.
The voltage is measured by National Instruments lab card pci-6221. The measured data are then been processed in Labview and displayed on the Cap Analyser front panel in Celsius. For a temperature difference of $1°C$ change corresponds to 0.39 ohm change in resistance, which results in a difference in voltage of 1.6 mV. According to formula 7.1 that is an voltage calculation over the Wheatstone bridge figure 7.2. National Instruments PCI-6221 has a 16-bit A/D converter and measure interval between-1V to +1 V. This gives a resolution of $1/2^{15} = 6.1035 \cdot 10^{-006}V$ which satisfies the requirement of accuracy of 0.5$°C$ when the difference in volts is 0.8mV. In figure 7.2 a schematic of a Wheatstone bridge can be seen.

$$V_G = (\frac{R_x}{R_3 + R_x} - \frac{R_2}{R_1 + R_2}) \tag{7.1}$$

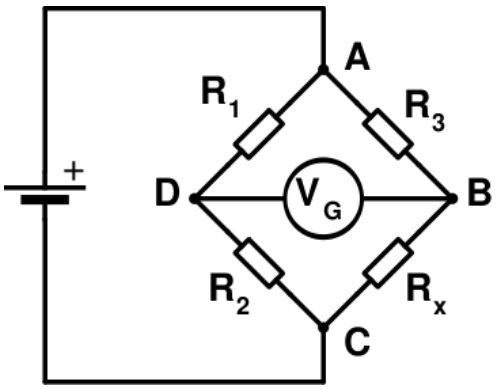

Figure 7.2: Wheatstone bridge

Chapter 8
Future Improvement

During the project more and more possibilities have come up on what could be good ideas to implement in the measuring system. Some of those ideas have been listed so the follow up project of this thesis has somewhere to begin from.

- Make Signals studies

  Make a signal study on the output signal from the analog interface electronic to verify which frequency is wanted and which we can filter away. With these studies better filter tap generation programs can be developed.
- Create better filter taps
- Improve the line fitting algorithm
- Adding better graphical tools in program

  To enhance the usability of the program better plotting tools are needed there are a extra package GTK extra that is available in Linux that would be a good tool to create better plotting function.
- Adding statistical computation on measured values

  Be able to use statistical tools to remove flyers from the baseline and smooth function to get a better baseline.
- Exchange the NI-hardware to a, embedded system

  To get ride of the dependency of National instrument an implementation of a embedded system should be done.
- Investigate the delay in the analog part

  The delay in the analog interface are changing it's delaying time depending on which gain level it have set on. If this was investigated the zero point could be set with a higher accuracy and there by get a better calculation of capacitance value.

Chapter 9
Restriction

The lack of documentation of the Labview system that was supposed to be the foundation of this master thesis led to that a lot of time in the beginning of the-work had to be put on decoding the labview code. Because there was no given or validated information of the labview system more effort had to be spent on theory behind the measurement that could support the new system. The result of this is that time had to be spent on this instead of constructing an embedded system that was intended in this master thesis.
A problem that happened in the Labview model also appeared in the new C-program with a downward spike before the new pulse is put out have been notice on the oscilloscope. However, it do not interfere with our measurement because the gate that put through the pulse voltage dose not open before $0.8V$ and the spike was never under this value. For some reason the National instrument are releasing the signal for 4ms before it puts out the new signal. The behavior stopped when the pulse stop acting as a trigger signal. But there haven't been found any solution on the problem and do to the lack of time this problem have only been noticed. This problem is not included in this Master's Thesis because this is a bug in National instruments hardware.

Chapter 10
Conclusions

10.1 Project conclusion

In this Master's Thesis a new measuring system have been developed that have the stability from the original system from Lund but have the user friendly interface from the Labview system. A temperature sensor and negative pulses have been implemented therefore completely new studies of how the capacitive biosensor reacts towards negative pulses and temperature changes.
The new system is developed in ANSI C which will make it easier to implement in a embedded system.

10.2 Personal conclusion

In this Master's Thesis I have learned the importance of good communication and organization in a bigger project. It has been a good mix of different task in this master thesis. Therefore, I have had use of many of the courses that I have been taking for example analog design, signal analyze, Numerical Analyze, C-programming, embedded systems, digital project and sensor technology. I have to say that I have learned a lot. I also have learned Labview and Linux subsystem under windows and how to build a programming project with make files which giving me a deeper understanding how bigger programs are built.
The biosensor is very sensitive of outside disturbance both in temperature changes and electrical interference so lots of effort was put on minimizing the disturbance, both trough changing the analog design, implement digital filters and change calculation model. This have shown me the strength of approaching the same problem from different angels and has given me a better confidence that everything can be solved but only if you are prepared to work for it.

References

[1] Christine Berggren, Bjarni Bjarnason, Gillis Johansson. *Instrumentation for direct capacitive Biosensors. Instrumentation science & technology, 27(2), 131-140(1999)*

[2] Charles K. Alexander, Matthew N.O Sadiku *Fundamentals of electric circuits, second edition, international edition 2003, pages 269-277*

[3] Allan J. Bard, Larry R. Faulkner *Electrochemical methods fundamentals and application, second edition, pages 19-28*

[4] Sven Knutsson, Chalmers Lindholmen institution of computer- and electronic science. *Digital signalbehandling Föreläsningsanteckningar, Uppdateringar, vecka 7, http://www.chl.chalmers.se/∼svenk/dig_sign/lektion/lek_upp7.pdf ,pages 1-5*

[5] Timothy D. Sauer. *Numerical analysis, pages 193-218*

[6] *Department of Physics, Faculty of Science, Prince of Songkla University, Hat Yai, Thailand.*

[7] Saurabh Banerjee, Nikhil Bhatwadekar, Prakash Mahajan, Pranjali Karode, Arbindakshya Mishra. *Process Analytical Technology (PAT) : Boon to Pharmaceutical Industry, http://www.pharmainfo.net/reviews/process-analytical-technology-pat-boon-pharmaceutical-industry*

[8] Digital signal processor. *http://en.wikipedia.org/wiki/Digital_signal_processor*

[9] PowerPC *http://en.wikipedia.org/wiki/PowerPC*

Appendix A

Result Labview test

A.1 Initial test on Labview system

A initial test was med on the existing Labview system to measure capacitance over a dummy cell figure 3.6. The reason why the measured values begin on pulse 19 is that ther was a glitch in one of the connection to the dummy cell.

Pulse	*Sensitivity*	*Gain*	*Volt mV*	*Sample rate*	*Cap nF*	*Res ohm*
19	X2	100	50	250k	0	3.2*10$\hat{7}$
20	X2	100	50	250k	NaN	NaN
21	X2	100	50	250k	25.3	41509
22	X1	100	50	250k	694.5	2149.01
23	X1	100	50	250k	82.2	5901.95
24	X1	100	50	250k	83.9	5796.76
25	X1	200	50	250k	82.4	6108.76
26	X1	200	50	250k	80.3	6298.68
27	X1	200	50	250k	NaN	6198.48
28	X2	200	50	250k	NaN	NaN
29	X2	200	50	250k	NaN	NaN
30	X2	200	50	250k	NaN	NaN
31	X1	500	50	250k	NaN	NaN
32	X1	500	50	250k	77.0	6693.67
33	X1	500	50	250k	77.4	6665.89
34	X1	500	50	250k	72.2	6666.95
35	X2	500	50	250k	NaN	NaN
36	X2	500	50	250k	NaN	NaN
37	X2	500	50	250k	NaN	NaN
38	X2	1000	50	250k	NaN	NaN
39	X2	1000	50	250k	NaN	NaN
40	X2	1000	50	250k	NaN	NaN
41	X1	1000	50	250k	NaN	NaN
42	X1	1000	50	250k	56.0	11811.9
43	X1	1000	50	250k	56.8	11865.6
44	X1	1000	50	250k	85.1	11885.8
45	X2	500	50	250k	NaN	NaN
46	X2	500	50	250k	NaN	NaN
47	X1	100	300	250k	NaN	NaN
48	X1	100	300	250k	71.5	7529.91
49	X1	100	300	250k	71.6	7531.47
50	X1	100	400	250k	64.0	9687.93

Pulse	*Sensitivity*	*Gain*	*Volt mV*	*Sample rate*	*Cap nF*	*Res ohm*
51	X1	100	400	250k	64.0	9689.02
52	X1	100	400	250k	64.0	9699.38
53	X1	100	200	250k	77.9	6359.91
54	X1	100	200	250k	77.9	6357.22
55	X1	100	200	250k	77.9	6336.7
56	X1	100	100	250k	78.0	4761.36
57	X1	100	100	250k	77.1	4696.98
58	X1	100	100	250k	77.1	4771.16
59	X1	100	50	500k	89.9	5982.78
60	X1	100	50	500k	77.1	5917.80
61	X1	100	50	1250k	137.5	5816.12
62	X1	100	50	1250k	140.6	5861.62
63	X1	100	50	1250k	140.7	5864.81
64	X1	100	50	125k	84.0	5775.79
65	X1	100	50	125k	81.9	5886.08
66	X1	100	50	125k	82.2	5864.04
67	X1	100	50	50k	81.7	5865.04
68	X1	200	50	1250k	NaN	6139.59
69	X1	200	50	1250k	158.9	6113.55
70	X1	200	50	1250k	157.8	6135.11

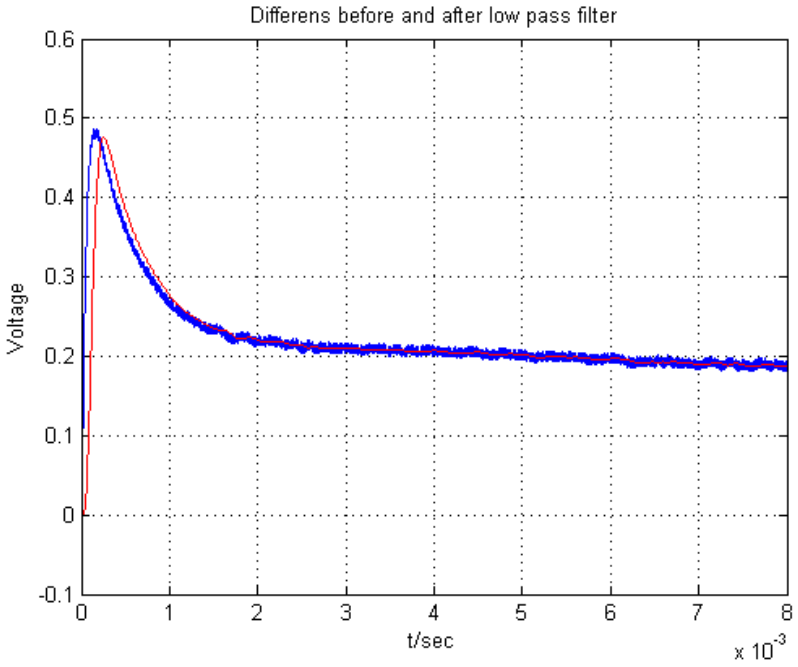

Figure A.1: Current answer from black box when sensitivity = X2

Figure A.1 show how the current answer stabilise on $0.2V$ when it should go down to zero.

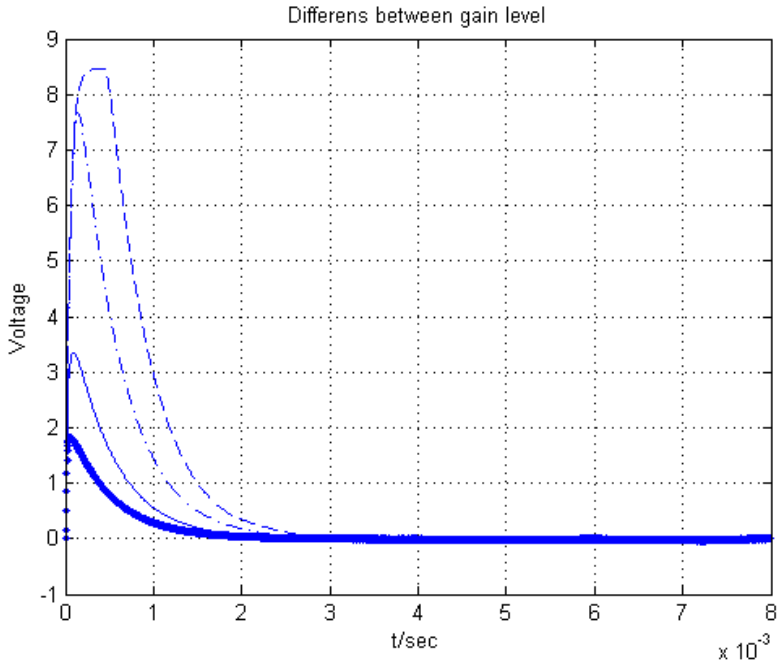

Figure A.2: Current answers with different gain. · = Gain 100, line = Gain 200, -. = Gain 500, – = Gain 1000

Figure A.2 show how different gains affect the current answer and that on gain 1000 the instrument amplifier can not amplify more then 9.5V.

In the figure A.3 it can be seen that the curve have lost to much information to calculate a accurate capacitance.

In figure A.4 make it clear that the filter delaying the current answer and that the analog instrument amplifier getting slower when the gain is increased.

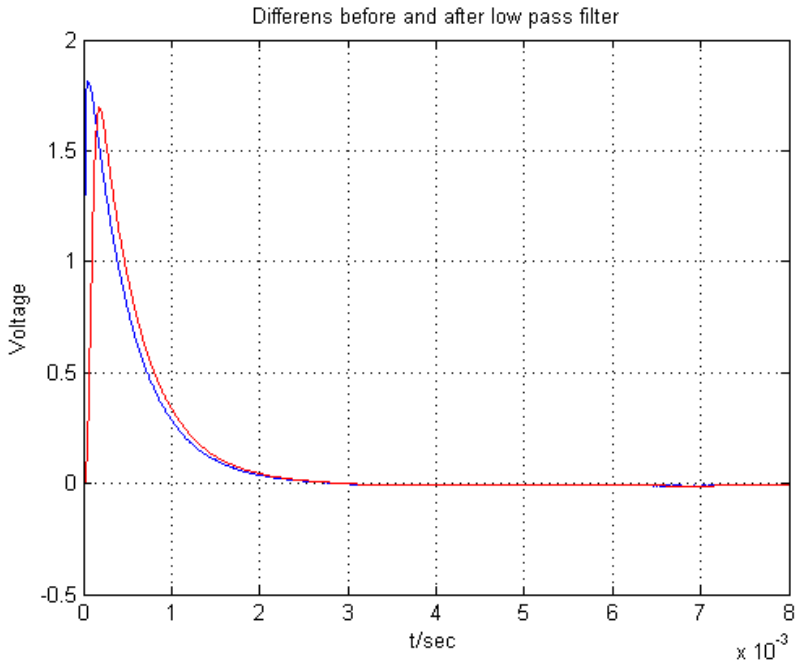

Figure A.3: Difference before and after digital filter in labview on the current answer

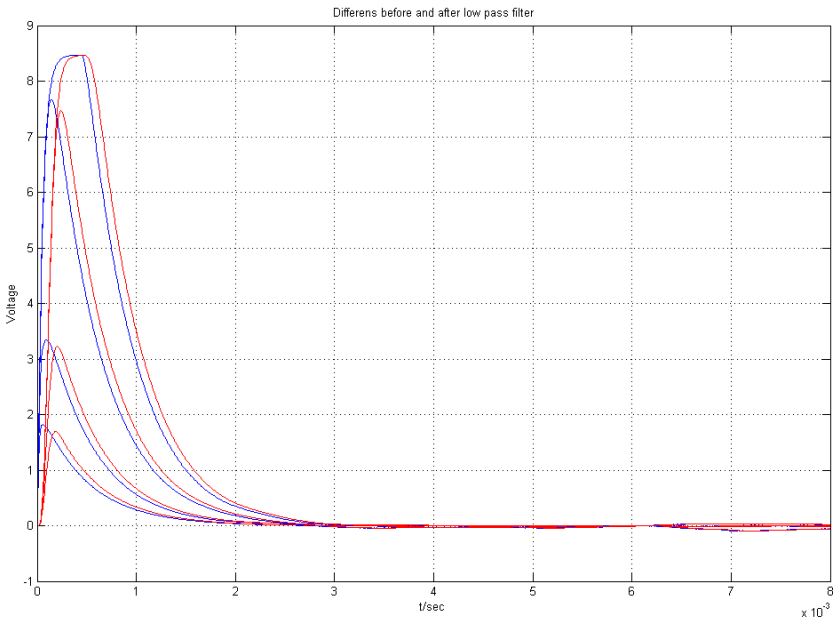

Figure A.4: Difference before and after digital filter in labview on the current answer

Appendix B
System overview

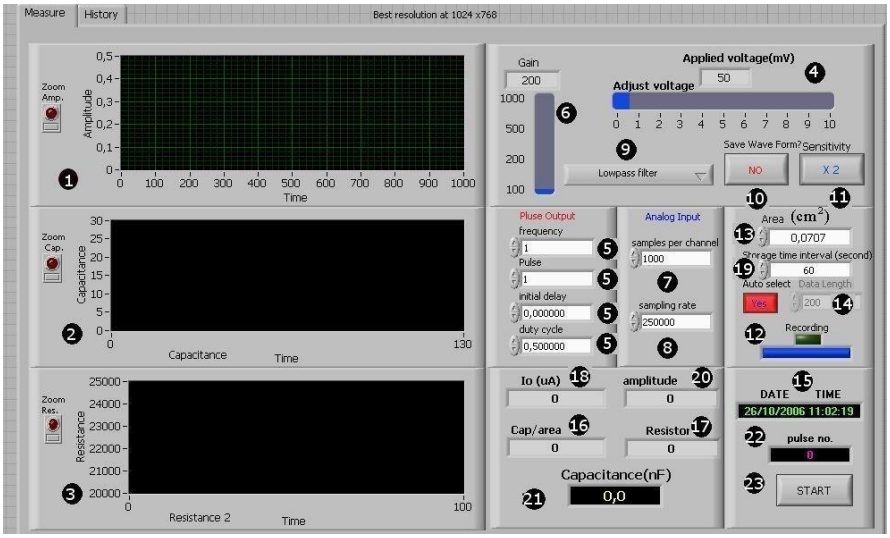

Figure B.1: CapAnalyser front panel from labview

Here on fallow a list of all the application on the LabView Panel Window. The application is numerated and corresponding to the number in figure B.1

1. Graph showing current response
2. Graph showing calculated capacitance value (Base line)
3. Graph showing estimated resistance value (Base line)
4. Voltage Strength of pulse
5. Pulse configuration frequency, pulse length and number of pulses (for details see signal study)
6. Gain of input signal from the biosensor

7. Number of samples per trigger pulse
8. Sample speed
9. Low pass filter
10. Saving the current response to file
11. Sensitivity of the system
12. Countdown to the next sampling
13. Area input value
14. how many values to load into the power response. Auto selection yes\no
15. Date and time
16. Last cap \ area value
17. Last resistance value
18. Last current value
19. Time interval between sample-tacking
20. Approximated Max value
21. Last capacitance value
22. Number of sample readings that have been done
23. Start/Stop button

Appendix C

Instruction manual of Capacitance Analyser

This is both an introduction and user manual of the new capacitance analyser program.
Start the program on clicking on the program icon named *BlueBoxCapBiosensor.exe* and a graphical interface like the figure C.1 shows. Before a measurement cycle are started some settings have to be configured. Begin to fill in the folders were the data will be saved and then fill in the file names remember to end the file with .txt. Chose gain level on the gain list button then chose Applied voltage level recommended 50mV. Then chose if the sampled in current wave should be saved. The Low pass filter is not good enough so skip this feature. The pulse settings is already optimised so they don't need to be changed. Under the Label "Analog input" options of the in sampling can be found. To get the best possible measurement change sample rate to 1250000Hz. The Auto select radio button is an option that decides how much of the in sampled data you are using in the calculation of the capacitance answer if the auto select is chosen it will take all the values from the peek value to $1/3$ of the peek value. If not the data length can be set manually but be carefully to use this option. The frontpanel of the interface box are shown in figure C.2. Several connection have to be done before measuring. First turn off the power to the interface connect the BNC contacts to respective nod on the biosensor and connect the temperature sensor to the serial port. Now the measurement can begin.

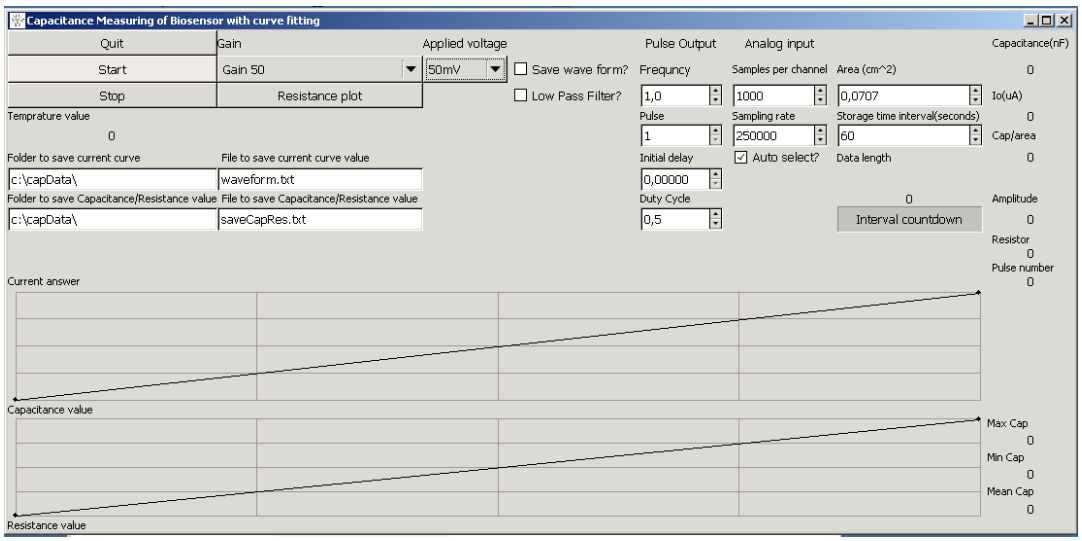

Figure C.1: BlueBoxCapBiosensor program

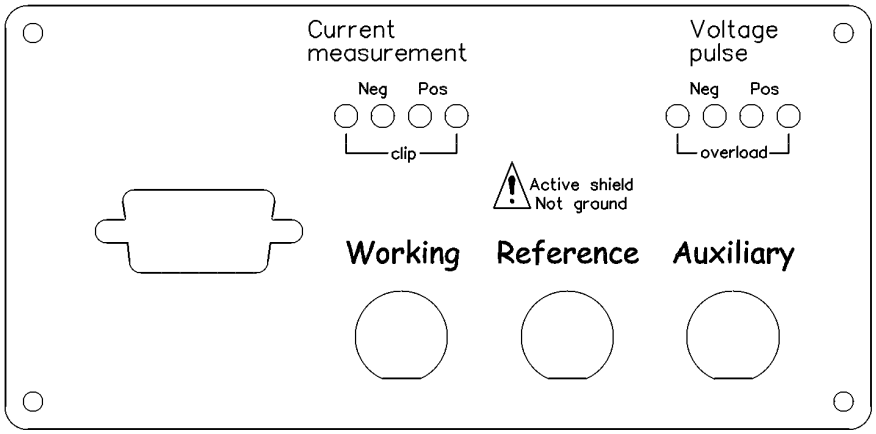

Figure C.2: Front panel of new Interface hardware

Appendix D

Old analog interface schematic

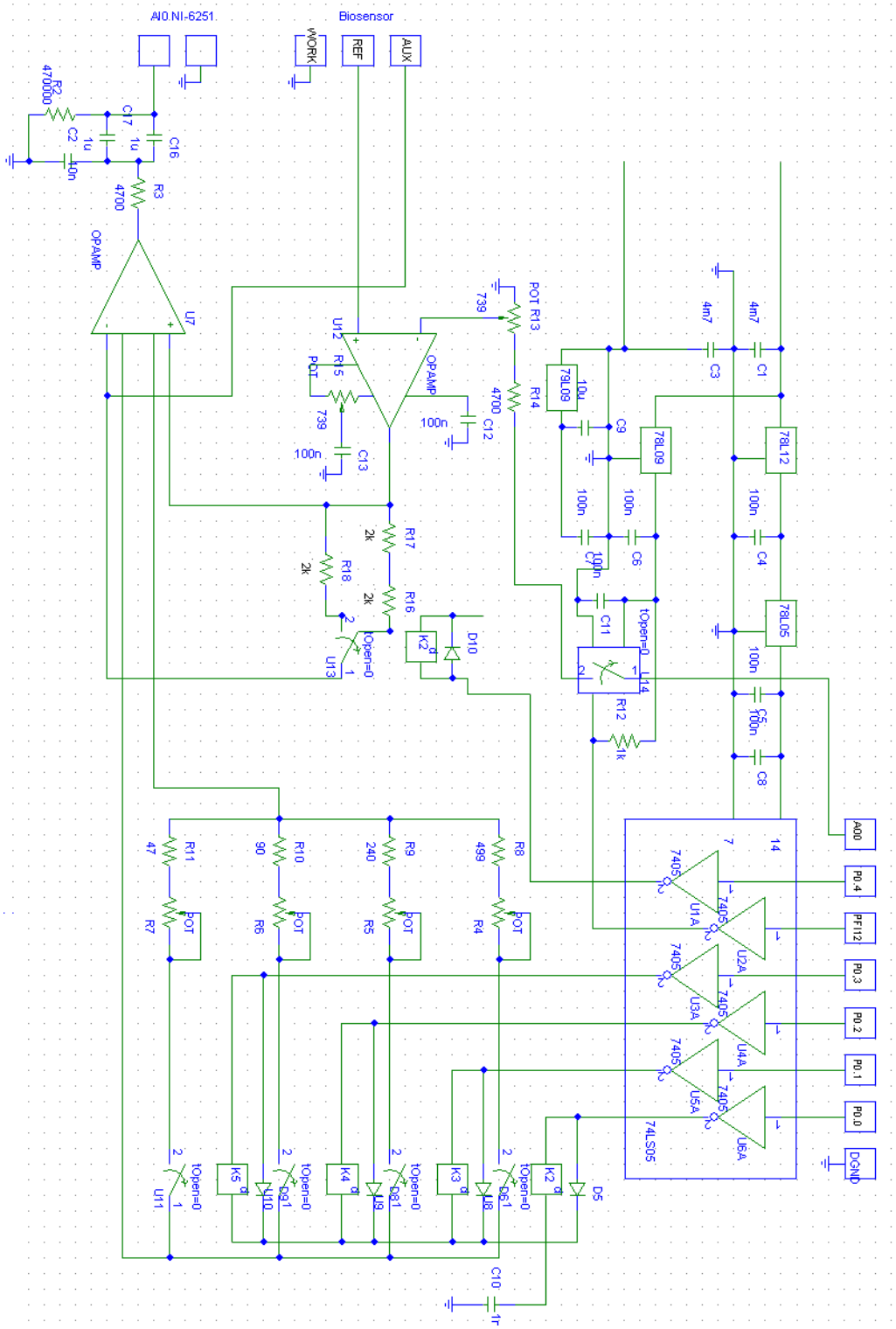

Figure D.1: Schematics of the black box